Available online at www.sciencedirect.com

ScienceDirect

Cognitive Systems Research 10 (2009) 204–215

Cognitive Systems RESEARCH

www.elsevier.com/locate/cogsys

A computational model of visual analogies in design

Action editor: Angela Schwering

Jim Davies [a,*], Ashok K. Goel [b], Nancy J. Nersessian [b]

[a] Carleton University, Institute of Cognitive Science, Ottawa, ON, Canada K1S 5B6

[b] College of Computing, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332, USA

Received 13 March 2008; accepted 19 September 2008
Available online 10 January 2009

Abstract

We present an analysis of the work of human participants in addressing design problems by analogy. We describe a computer program, called Galatea, that simulates the visual input and output of four experimental participants. Since Galatea is an operational computer program, it makes specific commitments about the visual representations and reasoning it uses for analogical transfer. In particular, Galatea provides a computational model of how human designers might be generating new designs by incremental transfer of the problem-solving procedure used in previous design cases.

Keywords: Analogy; Visual reasoning; Design; Artificial intelligence; Cognitive modeling; Transfer; Case-based reasoning; Diagrammatic reasoning

1. Introduction

Visual analogies, which are instances of analogical reasoning with visual knowledge, play an important role in design (e.g., Ferguson, 1992). In fact, on the basis of historical case studies of architectural design as well as cognitive studies of expert and novice architects, Goldschmidt and Casakin have described visual analogy as a core design strategy (at least) in architectural design (Casakin, 2004; Casakin & Goldschmidt, 1999; Goldschmidt, 2001). Further, Gross and Do (2000) have proposed CAD environments that explicitly support visual analogies (especially for architectural design). Although there appears to be a general agreement in research on design cognition that visual analogies play an important role in design, we are unaware of an information-processing model of visual analogies in design.

Let us consider a specific example of visual analogy in design to explain the goals of our work described here. Fig. 1 illustrates an input condition presented to a novice designer (experimental participant number L24) and Fig. 2 illustrates the output generated by the participant. Several aspects of the input and output illustrated in Figs. 1 and 2 are especially noteworthy (we describe the experiments in more detail later in the paper). Firstly, since the participants in this study were asked to the use the design problem and solution illustrated in the "Problem 1" part of Fig. 1 as a source for addressing the problem illustrated in the bottom half of the same figure, analogical retrieval is not a major issue in this setting. The participants in this experiment were given the source, and advised to use it. Secondly, note that the solution for the new design problem drawn by the participant (Fig. 2) is closely analogous to the drawing of the solution in the source design case (Fig. 1). (The analogy between the two drawings becomes even more apparent if the last drawing in Fig. 2 is mentally rotated clockwise by 90°.) The high-level research question for our work described in this paper, then, is this: given the source design case and an initial mapping between the representations of the source design case and the new design problem, how might participant L24 (and other participants in the study who generated similar drawings) have generated the drawing depicting his or her solution to the

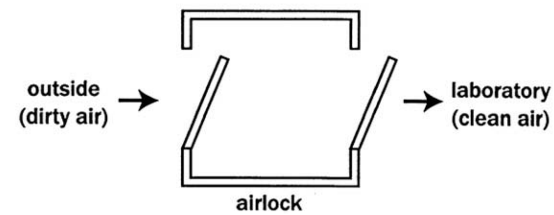
* Corresponding author. Tel.: +1 613 620 2888; fax: +1 613 520 3985.
E-mail address: jim@jimdavies.org (J. Davies).

1389-0417/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cogsys.2008.09.006

Please read the two problems below. At the bottom of the page, please try to solve **Problem 2.** *Draw a diagram to show what you're thinking. The solution to* **Problem 1** *may be helpful in solving* **Problem 2**.

Problem 1: **A computer chip manufacturer has designed a special lab for manufacturing microscopic devices. They have taken great care to seal off the lab from the surrounding environment in order to keep the air inside the lab free of dust and undesirable gases. The problem, though, is that whenever lab workers enter or leave the room, the seal is broken and contaminated air is allowed in. The company is trying to design a door that will allow workers to enter and leave the lab easily, while minimizing the amount of contaminated air that is let in.**

Solution: Have workers enter a vestibule space before entering the lab.



Problem 2: In order to trim the weeds that grow along the side of the road, the Department of Transportation has designed a weed trimmer that attaches to the end of a long pole sticking off the side of a truck. As the truck drives down the highway, the trimmer is extended about 6 feet to the right, perfectly positioned to trim the weeds at the side of the road. The problem is that while the 6-foot pole clears telephone poles and mailboxes, it is obstructed by sign posts that are positioned at the curb in certain parts of the city. The weed-trimmer pole, in fact, is exactly 2 feet too long to clear the sign posts. Although the weed-trimmer pole could be retracted or lifted out the way to clear the sign posts, this would interfere with the weed trimming. The Department of Transportation is trying to design a pole that can pass *through* the sign posts without stopping or changing the position of the trimmer.

Fig. 1. Condition 2: plan view of lab, with no walls.

new design problem by using an analogy with the drawing of the solution in the source design case?

Following Simon and his colleagues (Chase & Simon, 1973; Larkin & Simon, 1987), we assume that humans use visuospatial representations (i.e., knowledge comprised of only visual and spatial knowledge) not only externally, e.g., in the form of a drawing, but also internally. Again following Simon, we use the term "visuospatial" representations here to mean knowledge representations that capture the topology of the objects and relations in a situation but do not explicitly capture causality or teleology; such concepts are at most implicit in visuospatial representations. Building on Simon's work, Ullman, Wood, and Craig (1990) provide additional arguments about designers using visuospatial representations both externally and internally. Given these assumptions, let us return to the design solution generated by L24 (Fig. 2) to characterize the thesis of our work described here. Our data indicates that the (four) experimental participants transferred the design for the vestibule to generate the design of the weed trimmer. Since many other theories of analogy, such as SME (Falkenhainer, Forbus, & Gentner, 1990), LISA (Hummel & Holyoak, 1996), Proteus (Davies, Goel, & Yaner, 2008) and AMBR (Kokinov, 1998), might provide

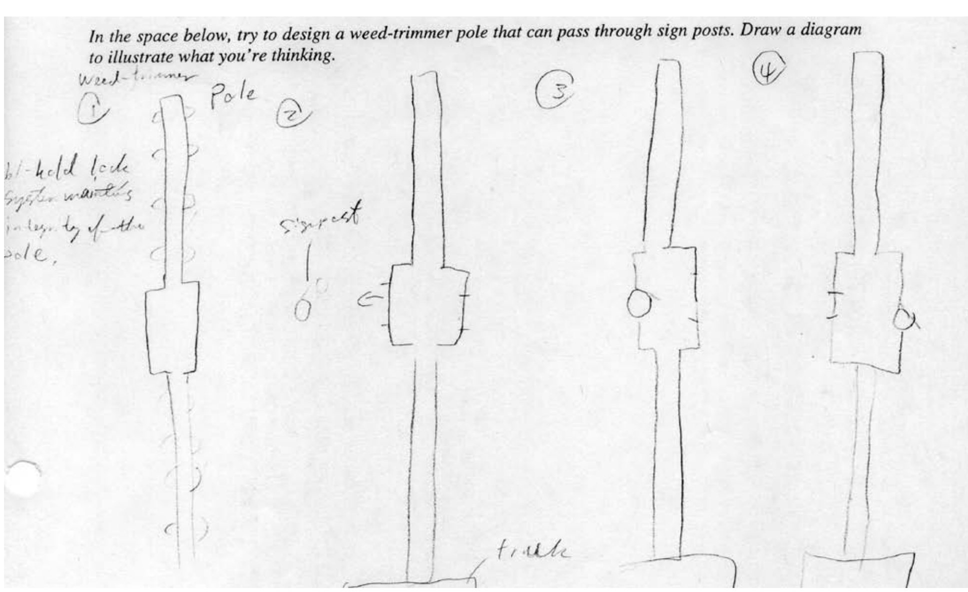


Fig. 2. Participant L24's data, scanned from what was drawn and written on the experimental sheet. L24 was in condition 2 (Fig. 1).

answers to the issue of analogical *mapping* between the two problems, we do not address it here; this work focuses on analogical transfer.

The data do not clearly indicate the information-processing mechanism that the participants used in the transfer of the design solution, but as previous research suggests (e.g., Holyoak &Thagard, 1989a, 1995) one possible mechanism is to abstract and transfer the problem-solving *procedure* from the source case to the target problem. Since both the source design case (top half of Fig. 1) and the new design problem (bottom half) have textual descriptions, we acknowledge that the participants might have built internal verbal representations of the two design problems, and may have used them to help with the analogy. Though the participants could be using different kinds of knowledge (such as visual and verbal) for transferring the problem-solving procedure, our more refined research goal is to examine the role of *visuospatial* knowledge in enabling the transfer of the problem-solving procedure from the source to the target. We want to examine whether visuospatial knowledge alone can account for transfer of the procedure, and what is the content, organization and representation of visuospatial knowledge that can support this transfer. Our high-level hypothesis is that visuospatial representation of intermediate knowledge states, organized in chronological order can enable transfer of problem-solving procedures. We hypothesize that these representations and processes can account of many elements of human participant data. We conjecture that (at least) in the context of design generation, human designers might address new design problems by abstracting and transferring visuospatially represented problem-solving procedures from source design cases.

As noted above, this conjecture is similar to that of Holyoak and Thagard (1989a, 1995). In their pioneering work on the PI model of analogical reasoning, Holyoak and Thagard proposed that humans address new problems by abstracting and transferring problem-solving procedures from familiar source cases. They also showed how the PI model provides an explanation of analogical transfer in (Duncker, 1926; Gick & Holyoak, 1980) radiation problem. Their explanation of Duncker's problem involves a problem-solving procedure that explicitly captures both causality and intent. The major difference between our thesis and that of Holyoak and Thagard's is that we hypothesize that (at least) in design, humans can usefully represent the problem-solving procedures using visuospatial representations in which causality (and intent) is (at most) implicit.

The thesis of this paper is that visuospatially represented problem-solving procedures, as mediating analogical transfer between source cases and new problems, can be used to model the transfer stage of design-by-analogy, where the source design case contains a drawing and the solution to the new design problem also needs to be in the form of a drawing. A visuospatial representation of the problem-solving procedure appears necessary because the source design solution is in the form of a drawing and because the final design solution is often presented as a series of drawings. However, as we noted above, analogical mapping may well involve alternative representations, such as verbal representations that explicitly capture causality. To this end, below we first present an analysis of the work of 15 human participants in addressing design problems by analogy. Then, we describe a computer program, called Galatea, that simulates the input and output visuospatial representations of four of the 15 participants. Since Galatea is an operational computer program, it makes specific commitments about the visuospatial representations and reasoning it uses for analogical transfer. Since we have described Galatea in detail elsewhere (see Davies & Goel, 2001, for the first publication of Galatea, Davies & Goel, 2007 for a description of the Cognitive Visual Language, Davies & Goel, 2008 for a theoretical description, and Davies et al., 2008 for a detailed description of algorithms) and due to limitations of space in this paper, here we include only a basic sketch of its working that is sufficient for the purposes of this discussion. Finally, we discuss how Galatea models the drawings generated by the human designers.

2. An analysis of design generation

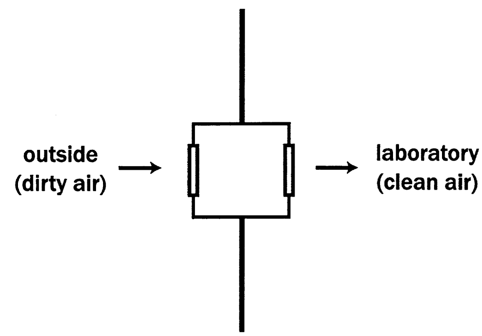
Craig (2001) describes a cognitive study of 34 novice designers (undergraduate students at the Georgia Institute of Technology). The participants in the study were shown a design source case (a laboratory clean room), containing both a design problem stated in the form of text and a design solution in the form of an annotated drawing. The study was conducted in different input conditions: Fig. 1 illustrates one input condition; Fig. 3 illustrates another input condition.

The participants in the experimental study were asked to solve an analogous design problem (a sidewalk weed trimmer); the new problem was represented with text only. The participants were encouraged to use the design case presented earlier as a source for addressing the new problem, and asked to illustrate their designs. Of the 34 participants, 15, or a little less than half of the participants in different conditions, generated the correct design solution rendered as a drawing by adding redundant doors to a weed-trimmer arm so that it can pass through street signs; if the arm contains two latching doors, then while one door is open to let the sign pass, the other stays closed to support the arm and trimmer. Figs. 2, 5 and 6 depict the work of three participants (L24, L22, and L15, respectively) in the input condition depicted in Fig. 1; Figs. 4 and 7 depict the work of two participants (L14 and L16, respectively) corresponding to the input condition of Fig. 3.

The data from this experiment are appropriate for our work several reasons: (1) it is an example of the kind of design task we are interested in investigating: cross-domain analogies involving the transfer of multi-step, *Please read the two problems below. At the bottom of the page, please try to solve* **Problem 2.** *Draw a diagram to show what you're thinking. The solution to* **Problem 1** *may be helpful in solving* **Problem 2**.

Problem 1: A computer chip manufacturer has designed a special lab for manufacturing microscopic devices. They have taken great care to seal off the lab from the surrounding environment in order to keep the air inside the lab free of dust and undesirable gases. The problem, though, is that whenever lab workers enter or leave the room, the seal is broken and contaminated air is allowed in. The company is trying to design a door that will allow workers to enter and leave the lab easily, while minimizing the amount of contaminated air that is let in.

Solution: Have workers enter a vestibule space before entering the lab.



Problem 2: In order to trim the weeds that grow along the side of the road, the Department of Transportation has designed a weed trimmer that attaches to the end of a long pole sticking off the side of a truck. As the truck drives down the highway, the trimmer is extended about 6 feet to the right, perfectly positioned to trim the weeds at the side of the road. The problem is that the 6-foot pole is obstructed by sign posts that are positioned at the curb in certain parts of the city. The weed-trimmer pole, in fact, is exactly 2 feet too long to clear the sign posts. Although the weed-trimmer pole could be retracted or lifted out the way to clear the sign posts, this would interfere with the weed trimming. And although the pole could bend over the top of the sign posts, this would be impractical since in some areas the signs are 15 feet tall. The Department of Transportation is trying to design a pole that can pass *through* the sign posts without stopping or changing the position of the trimmer.

Fig. 3. Condition 1: plan view of lab, with the vestibule.

strongly-ordered solution procedures, (2) addressing the design task involves visual knowledge and reasoning, at least for understanding the diagram in the input as well as for generating a drawing as the output, (3) solving the design task also involves non-visual knowledge (e.g., causal and functional knowledge to understand the systems described).

The 15 participants who successfully generated the correct solution for the given design problem showed many differences in the outputs they produced. Table 1 summarizes these differences.

It is possible that some participants realized the analogy but failed to find the correct answer nonetheless. Those that failed either ignored the suggestion to use the analogy or could not figure out how to effectively use it. It appears that no one who used the correct analogy in their drawing failed to find the correct answer.

3. Galatea: a computational program that performs visual analogies

We briefly summarize the salient elements of Galatea that are relevant for the present discussion. Galatea is an implementation of the constructive adaptive visual analogy theory (Davies, 2004). It uses Covlan, a Cognitive Visual Language, for representing visuospatial knowledge (Davies & Goel, 2007). The main features of this language are *primitive visual elements*, such as *rectangles* and *lines*, and *primitive visual transformations*, such as *replicate* and *add-object*. The inputs to Galatea (design source cases, new design problems) are completely visual in nature.

Galatea represents multi-step problem-solving procedures as a series of knowledge states and transformations between the states. The elements of each knowledge state

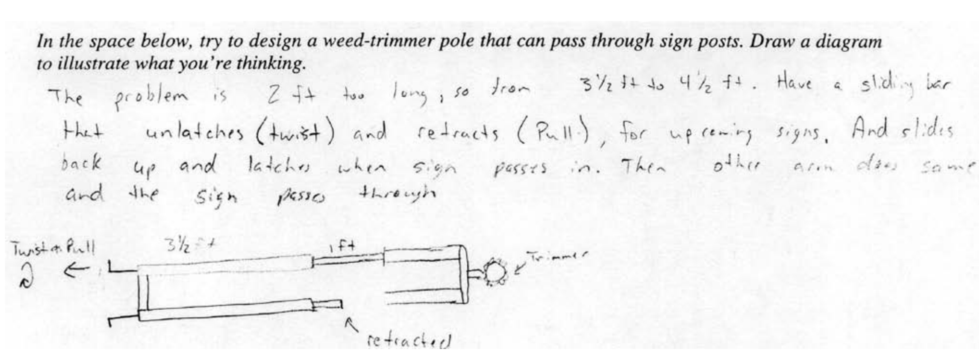


Fig. 4. L14's data, scanned from what was drawn and written on the experimental sheet. L14 was in condition 1 (Fig. 3).

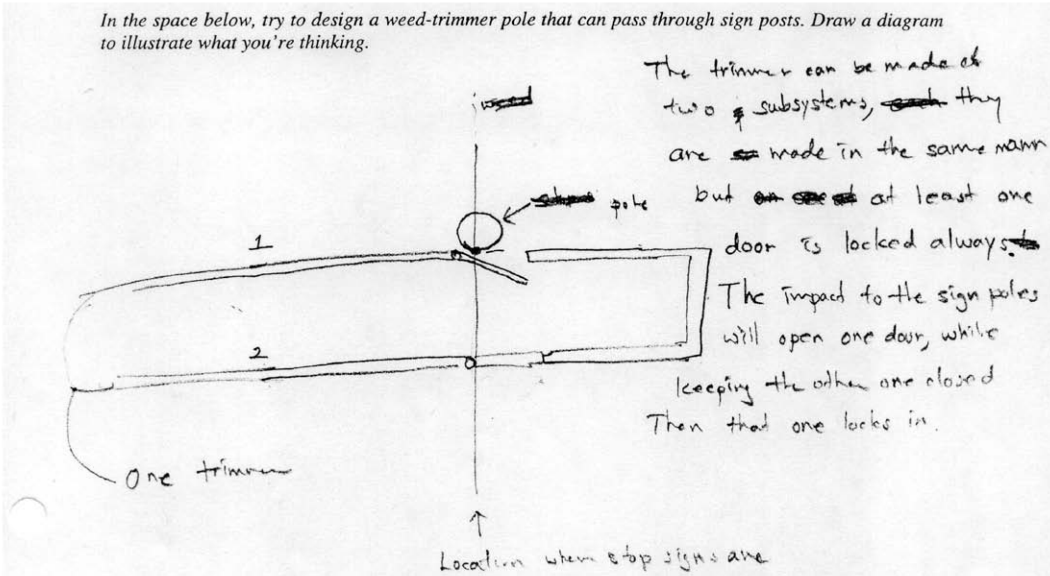


Fig. 5. Participant L22's data, scanned from what was drawn and written on the experimental sheet. L22 was in condition 2 (Fig. 1).

are *instances* of visual *elements*, and the operations are visual *transformations*. Knowledge states consist of visual knowledge represented symbolically; we call them *s-images*, or symbolic images.

We will use Duncker's radiation problem (1926) as an example because it is so well known (see Davies & Goel, 2001, for details on Galatea's model of this problem). In the fortress problem, we needed an operator that took one shape and turned it into multiple, smaller shapes. We created one that did this and called it *decompose*. This *transformation* was later used for other examples. The *elements* are defined by the slots (location, size, length, etc.), the possible values those slots can take, and the *transformations* that can be applied to them. For example, the tumor problem required an *element* that had a start and end point, so the *line element* was created.

We represented the fortress story with three *s-images*. The first was a representation of the original fortress problem. It had four roads, represented as thick *lines*, radiating out from the fortress, which was a *curve* in the *center* (*curves* are used to represent irregular shapes). We represented the original soldier path as a thick *line* on the bottom road. This first *s-image* was connected to the second with a *decompose transformation. Decompose* takes in some *primitive visual element* instance and replaces it with some number of smaller versions of it in the next knowledge state. *Transformations*, like functions, take arguments (in this case the arguments were *soldier-path1* for the *object* and *four* for the *number-of-resultants*). The second *s-image* has the *soldier-path1* decomposed into four thin *lines*, all still on the bottom road. The *lines* are thinner to represent smaller groups.

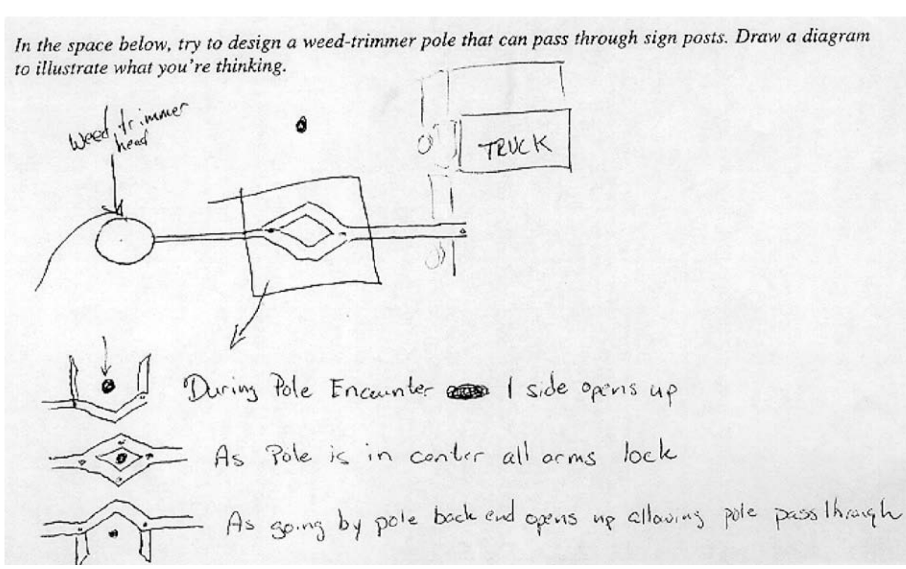


Fig. 6. Participant L15's data, scanned from what was drawn and written on the experimental sheet. Participant L15 was in condition 2 (Fig. 1).

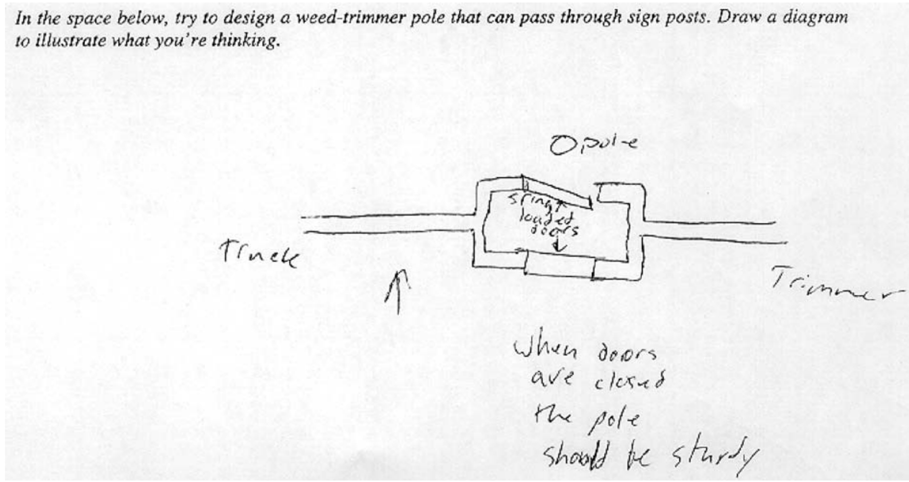


Fig. 7. Participant L16's data, scanned from what was drawn and written on the experimental sheet. L16 was in condition 1 (Fig. 3).

In the fortress/tumor example, after the *decompose* transformation generates a number of smaller armies (by transforming a thick arrow into thinner arrows), those armies must be dispersed to the various roads, in various locations in the image. This uses the *move transformation*. We represented the start state of the tumor problem as a single *s-image*. The tumor itself is represented as a *curve*. The *ray* of radiation is a thick *line* that passes through the bottom body part. From this example one can see how Galatea describes analogs visually, and incrementally transfers knowledge states, and transformations taken on them, one at a time.

4. The models of the lab/weed-trimmer problems

We used our theory of constructive adaptive visual analogy to model the work of all 15 participants in Craig's data who successfully generated the correct design solution, 4 in Galatea itself and the other 11 using pen-and-paper models based on the theory. In the case of four participants directly modeled in Galatea, we kept the reasoning architecture, the representation language, and control of processing exactly the same for each of the four participants, varying only the initial knowledge content entered into Galatea for the different participants.

To evaluate the 15 models, we look at how well the model accounts for the differences between the source problem diagram and the participant's drawn diagram (as summarized in Table 1). The image accompanying the source in the experimental stimulus is very abstract. It is so abstract, in fact, that with a different textual description it could apply equally well to the source and target problems. What this means is that if the experimental participants used the image to transfer the solution, they did not *need* to change the diagram at all. As we will see, *every* participant produced a drawing that differed in some way

Table 1
Differences observed in the outputs generated by the 15 successful participants. Difference names are on the y axis, participant numbers are on the x axis.

	L1	2	11	12	13	14	15	16	19	20	21	22	24	27	28	Total
Added objects	X	X		X	X	X	X	X	X	X	X	X	X	X	X	14
Center	X	X														2
Doors open, walls remain			X													1
Dotted object					X				X							2
Double line to line									X	X	X		X			5
Explicit simulation							X		X	X	X	X	X		X	6
Line to double line	X	X			X	X	X	X								6
Long vestibule						X						X				2
Mechanism added	X					X		X	X							4
Multiple doors									X							1
No vestibule/doors distinction		X					X								X	3
Numeric dimensions						X								X		2
Point of view change											X			X	X	3
Rectangle to line: door			X	X	X											3
Rotation	X			X	X	X	X	X		X		X		X		9
Sliding doors						X										1
Zoom					X						X			X		3
Total	5	4	2	3	6	7	5	4	6	4	5	4	3	5	4	

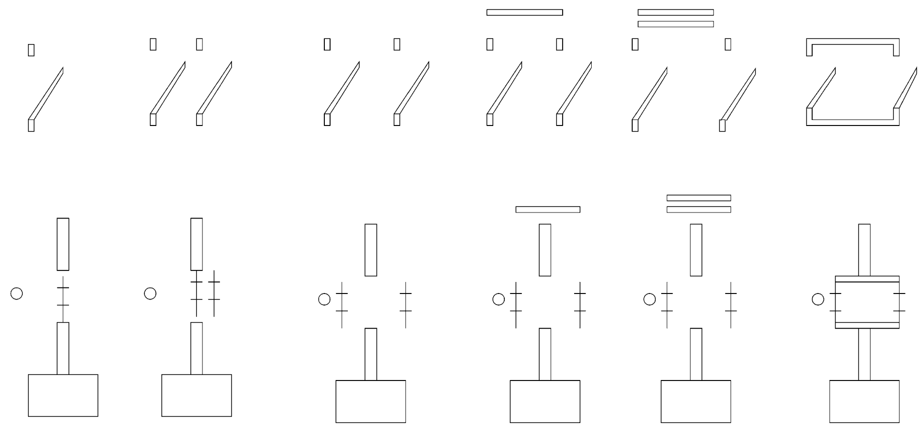


Fig. 8. The model of L24. The top series of s-images is the source, the bottom series is the target.

from the original source. These “differences,” as we will call them, between the source and target diagrams, are indicative of the variation among the participants studied.

Modeling Craig’s experimental participants involved determining the Covlan representation of the source and initial target *s-images*. Using our hypothesis about visual re-representation in analogy, we predicted the participants’ output. To evaluate the models we compared the nature of this predicted output to the differences found in the data. We will describe one participant in detail, L24, and refer you to Davies (2004) for detailed descriptions of all the models.

4.1. The model of L24

L24 was in experimental condition 2, the stimulus of which can be seen in Fig. 1. As in all the models, we represented the source analog as a series of *s-images* connected with *transformations*. This representation of the source case in condition 2 we will call *lab-base2*.

Fig. 8 has two parts. The top series of images refers to Galatea’s representation of the source problem given to L24 as stimulus (it is only a depiction for the reader’s understanding. Covlan represents *s-images* propositionally). Looking at the stimulus (Fig. 1), we see that there is only a single image. However, we conjecture that participants use this image and the text description given to create a representation of the steps taken to solve the problem. Thus there are six images in our model of the source. The first, in the top left of Fig. 8, shows the situation in its problem state. Between each picture along the top are *transformations* (not shown in Fig. 2) leading to the final picture, which has the image as given in the stimulus. Briefly, the doorway mechanism is duplicated, and then the duplicate is moved. Two walls are created, and finally they are placed in the correct positions with respect to the doorway duplicates.

The bottom set of images in Fig. 8 illustrate the model’s representation of L24’s solving of the problem. The picture on the bottom left is the initial state of the problem, including representations of the truck, blades, and pole. Double lines are turned to lines and the system is rotated. As actions are transferred from the source to the target, new states are generated, until finally, in the bottom right, we see the target problem in its final state. Our model of L24 involves five *transformations*. The first is *replicate*. It takes in the set of elements composing the door mechanism (we will call it *door-set-l24s1*[1]) and creates another identical but distinct set of elements (*door-set2-l24s2*) in the next *s-image*.

The second transformation is *add-connections* which places the door sets in the correct position in relation to the top and bottom walls. *Add-connections* adds spatial relationships to the objects it modifies. The third and fourth *transformations* are *add-component*, which add the top and bottom containment walls that complete the vestibule. The fifth *transformation*, another *add-connections*, places these containment walls in the correct positions in relation to the door sets and the top and bottom walls.

We will describe the first two *transformations* in detail. The first *transformation* in the *lab-base2* source is *replicate*, which takes two arguments: some *object* and some *number-of-resultants*. In this case the *object* is *door-set1-l24s1* and the *number-of-arguments* is *two*. The *replicate* is applied to the first L24 *s-image*, with the appropriate adaptation to the arguments: the *mapping* between the first source and target *s-images* indicates that the *door-set-b2s1* maps to the *door-set-l24s1*, so the former is used for the target’s *object* argument. The number *two* is a literal, so it is transferred directly.

As part of the *transformation updating* the reasoner automatically generates the *mapping* between *lab-base2-simage2* and *l24-simage2*. *Element* instances that are results of source *transformations* are mapped to newly-generated instances in the target. All other alignments, called *maps*, are carried over to the new *s-images* with their new names.

[1] The notation “l24s1” means that the symbol is a part of the first *s-image* of the L24 model. The same scheme is used to name other symbols in our models.

This is a crucial step, and is an important part of a claim this paper is making.

The second *transformation* is *add-connections*. The effect of this transformation is to place the replicated door sets in the correct spatial relationships with the other element instances. How does the reasoner know to which *elements* the *transformation* should be applied? The door set was replicated, and the new door set is not a part of the original input mapping. In the previous paragraph we described how the reasoner updates the mapping so that newly-generated objects have analogs. Without this inference, the reasoner will not know to which element or elements to apply the *add-connections transformation*. It takes *connection-sets-set-b2s3* as the *connection/connection-set* argument. This is a *set* containing four *connections*. The reasoner uses a function to recursively retrieve all *connections* and set proposition members of this *set*. These propositions are put through a function which creates new propositions for the target. The element instance names are changed to newly-generated analogous names. For example, *door1-endpoint-b2s3* turns into *door1-endpoint-l24s3*.

Then, similarly to the *replicate* function, horizontal target maps are generated, and the other propositions from the previous *s-image* are instantiated in the new *s-image*.

We will now examine the differences between the source picture and what L24 wrote on his or her experimental sheet (see Figs. 1 and 2). On the experimental sheet L24 described explicitly how the mechanism could work, added some objects (the truck, blades, and pole), and changed double lines into single lines. Also, the entire mechanism is rotated.

The model of L24 accounts for two of the three differences found. The added objects are accounted for with the input target representation: since these extra elements are in the first *s-image*, the reasoner carries them through all subsequently generated *s-images*. The parts of the drawings drawn as double lines in the source change to single lines in the target. This change is also accounted for with the input representation. All of these differences required no changing of the theory, just a modification of the input information. However, the line to double line difference cannot be considered completely accounted for because the model fails to capture the double line used to connect the door sections, because the single line is transferred without adaptation from the source. This could be fixed, perhaps, by representing the argument to the *add-component* as a function referring to whatever element is used to represent another wall, rather than as a *line*.

The one difference the model fails to account for is the presence of explicit simulation. This kind of information is not describable in Covlan, which is intended to describe diagram-like inscriptions rather than working mental models.

4.2. The other models

Our models (both with Galatea and pen-and paper) of the other 14 successful participants were created similarly.

Table 2
Differences accounted for by Galatea.

Participant	Differences accounted for	Percentage
L1	2/5	40
L2	3/4	75
L11	0/2	0
L12	2/3	67
L13	3/6	50
L14[a]	4/7	57
L15[a]	4/5	80
L16[a]	3/4	75
L19	2/6	33
L20	2/4	50
L21	2/5	40
L22[a]	3/4	75
L24	2/3	67
L27	2/5	40
L28	3/4	75
Total	37/67	55

[a] Implemented in the Galatea modeling architecture.

Table 2 shows that our models accounted for about half of the differences (55%). The following sections describe the four models we implemented in Galatea (L14, L15, L16, and L22). L24 described above and 10 other participants were modeled with pen-and-paper using Galatea's representations and processing. L14, L15, L16 and L22 are representative of some of the more difficult experimental participants in the study.

4.3. The Galatea model of L14

L14 received condition 1 of the lab problem (see Fig. 3). Fig. 9 shows the model of L14.

We represented the source analog with a different series of *s-images* connected with *transformations*, which we will call lab-base1. See the top of Fig. 9 for an abstract diagram of this analog.

The model of L14 involves five *transformations* (see Fig. 9). The first *transformation* is *replicate*. It takes in the *door-set-l14s1* as an argument, generating *door-set-l14s2* and *door-set2-l14s2* in the next *s-image*. The "door set" is a group of elements consisting of the door, and the two wall pieces adjacent to it.

The second *transformation* is *add-connections* which places the door sets in the correct position in relation to the top and bottom walls. The third and fourth *transformations* are *add-component*, which add the top and bottom containment walls that complete the vestibule. The fifth *transformation*, another *add-connections*, places these containment walls in the correct positions in relation to the door sets and the top and bottom walls.

We can now examine what made L14 (Fig. 4) differ from the stimulus drawing (Fig. 3): L14 features a longer vestibule in the drawing than the vestibule pictured in the stimulus. In fact, there is no trimmer arm (analogous to the wall in the lab problem) in the drawing at all that is distinct from the vestibule, save a very small section, apparently

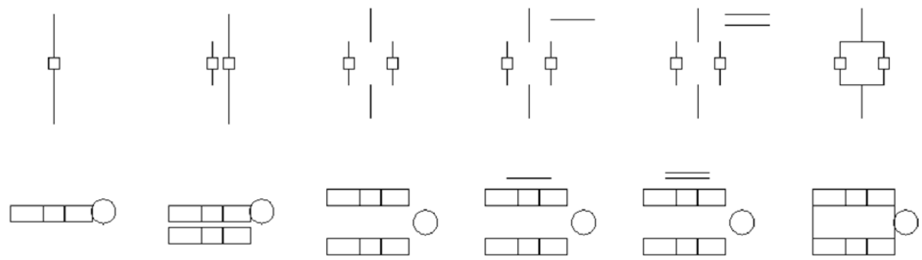


Fig. 9. The model of L14. The top series of s-images is the source, the bottom the target.

to keep the spinning trimmer blade from hitting the vestibule. The entire drawing is rotated 90° from the source. The single lines in the source are changed to double lines in the target. The doors also slide in and out of the vestibule walls. What's interesting about this modification is that it does not appear that this kind of door opening is possible with the diagram given of the lab in the source: since the door is a rectangle that is thicker than the lines representing the walls, the door could not fit into the walls. In contrast L14 explicitly makes the doors and walls thick (with two lines) and makes the doors somewhat thinner. L14 adds objects to the target not found in the source: a blade and a twisting mechanism to describe how the doors can work. L14 also included numerical parameters to describe the lengths in design of the trimmer. Finally, L14 includes some mechanistic description of how the trimmer would work.

In summary, these behaviors are: (1) long vestibule, (2) rotation, (3) line to double line, (4) sliding doors, (5) added objects, (6) numeric dimensions added, and (7) mechanisms added. Of these seven differences, Galatea successfully models four. The *rotation* of the source is modeled by a rotation in the target start *s-image*. In this *s-image*, all spatial relationships are defined only relative to other element instances in the *s-image*. Each instance is a part of a single set which has an orientation and direction. In the case of *s-image* 1 of the target, it is facing right. Since all locations are relative, there is no problem with transfer and each *s-image* in the model of L14 is rotated to the right.

The *line to double line* difference is accounted for by representing the vestibule walls with rectangles rather than with lines, as it is in the source. Because the *mapping* between the source and target correctly maps the *side1* of the rectangle to the *startpoint* of its analogous line, the rectangle/line difference does not adversely affect processing and transfer works smoothly.

The *long vestibule* difference is accounted for by specifying that the heights of the vestibule wall rectangles are *long*. In the source the vestibule wall lines are of length *medium*, but this does not interfere with transfer.

The blade *added object* is accounted for by adding a circle to the first *s-image* in the target.

Unaccounted for are the two bent lines emerging from the vestibule on the left side, the numeric dimensions and words describing the mechanism. Also, L14 shows one of the doors retracting, and the model does not. The model also fails to capture the double line used to connect the door sections for the same reason the L24 model failed in this regard.

4.4. The Galatea model of L22

L22 received condition 2 (see Fig. 1). Fig. 5 shows what L22 wrote on his or her data sheet during the experiment. Again, we represented the source analog as a series of *s-images* connected with *Transformations*. See the top of Fig. 10 for an abstract diagram of the analogs.

The model of L22 involves five *transformations* (see Fig. 10). The first *transformation* is *replicate*. It takes in the *door-set-l22s1* as an argument, generating *door-set1-l22s2* and *door-set2-l22s2* in the next *s-image*. Note that the door set replicated here is different from the door set replicated for L14. In this case, there are three connected rectangles, corresponding to the top wall, door, and bottom wall. In the case of L14, the door set is made of a single long rectangle (representing the wall) with another rectangle (representing the door) in front of it. But because *replicate* can work on any set of element instances, Galatea can accommodate the kind of doorway L22 had in mind.

The second *transformation* is *add-connections* which places the door sets in the correct position in relation to each other. Unlike for L14, there are no top and bottom walls. The third and fourth *transformations* are *add-component*, which add the top and bottom containment walls. The fifth transformation, another *add-connections*, places these containment walls in the correct positions in relation to the door sets.

The processing and adaptation of these *transformations* resembles the processing done with L14.

We can now examine what made L22 (Fig. 5) differ from the stimulus drawing (Fig. 1). The entire drawing is rotated 90° from the source. An object is added to the target that has no analog in the source: the trimmer. L22 features a proportionately longer vestibule than in the source, and has some explicit simulation diagrammed. Of these differences, all but the last were modeled by changing the nature of the start *s-image* for L22.

4.5. The Galatea model of L15

As shown in Fig. 6, L15 does not distinguish between the vestibule and the doors leading into it. The drawing

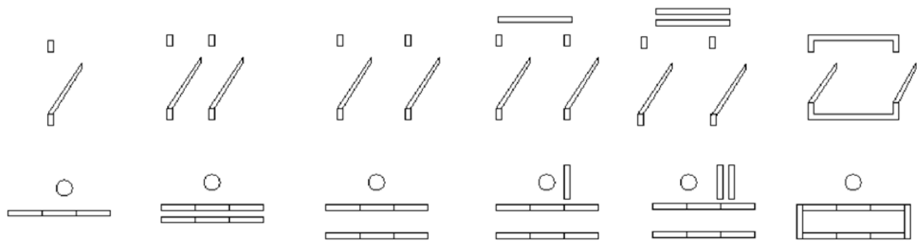


Fig. 10. The model of L22.

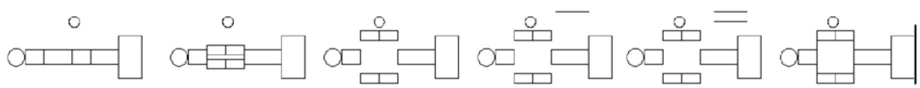


Fig. 11. The model of L15.

is rotated, and the lines depicting the walls are turned into double lines. Added objects include: truck, pole, hinges, and the trimmer head.

Most interestingly, at the bottom is a set of states, like a film strip, describing a simulation of how the pole could move through the trimmer. The observed differences were (1) rotation, (2) changing a line to a double line, (3) adding objects, (4) explicit simulation description and (5) a lack of distinction between the vestibules and the doors.

The model for L15 uses the same source analog as L22. As seen in Fig. 11 the changing of lines to double lines, the rotation and the added objects are accounted for by the input target. The no vestibule/doors distinction is accounted for by what is replicated. It does not account for the simulation, nor some of the details of the shape of the door mechanism (particularly the angle of the doors).

4.6. The Galatea model of L16

L16 (Fig. 7) was in condition 1 (Fig. 3) and features a rotated trimmer, and includes an arrow showing the direction of the motion of the truck. The pole is added, the lines are thickened to double lines, and the mechanism is described, including one door open and one shut. The observed differences were (1) rotation, (2) line changed to double line, (3) the adding of objects, and (4) a mechanism added.

The door mechanism, which includes doubled lines in the initial target, gets replicated in the second *s-image*. As in the case of L14 and others, the results of the connection *transformations* result in single line transfers. This is because the *add-component* function takes the *line* literal as an argument. Thus when Galatea transfers it, it remains a *line*, even though the rest of the structure in the target is *rectangles*.

The model can be seen in Fig. 12. Our model accounts for three of the four differences: the mechanism difference is missing for the same reasons as in models described above.

4.7. Summary of results

The models described in the previous section show how using only visual representations allows the generation of design drawings by analogy, supporting our hypothesis. The models presented accounted for many of the differences shown in the participants' drawings. Although the Galatea models were able to account for most of the differences observed, in general it failed to account for differences of the following kinds: explicit simulations, added mechanisms, numeric dimensions, and sliding doors (which only one participant exhibited). Of these, we would not expect Galatea to model explicit simulations, since simulation of the designed mechanism is beyond the intended scope of the theory. Other systems, (e.g., Forbus, 1995; Funt, 1980; Larkin & Simon, 1987; Narayanan, Suwa, & Motoda, 1994) use visual representations of physical systems to predict how the represented systems will behave.

The added mechanisms and sliding doors, however, are visuospatial information that Galatea failed to model. To account for these would require adding causal knowledge needed to invent new mechanisms. At this point Galatea has no such knowledge.

5. Related work

There are a variety of computational systems, each aiming to understand different parts of the analogical process. Though they use visual representations, MAGI (Ferguson, 1994), JUXTA (Ferguson & Forbus, 1998), VAMP.1, VAMP.2 (Thagard, Gochfeld, & Hardy, 1992), and DIVA (Croft & Thagard, 2002) are all addressing the problem of analogical mapping. They are all extensions of non-visual analogical mappers: MAGI and JUXTA are built on SME (Falkenhainer et al., 1990) and GeoRep (a visual language and inference engine, Ferguson & Forbus, 2000); VAMP.1, VAMP.1, and DIVA are all built on ACME (Holyoak & Thagard, 1989b).

Galatea transfers problem-solving solution procedures, like Prodigy (Schmid & Carbonell, 1999; Veloso, 1993;

Fig. 12. The model of L16.

Veloso & Carbonell, 1993), CHEF (Hammond, 1990), and PI (Holyoak & Thagard, 1989a). Other visuospatial problem solvers, such as Letter Spirit (McGraw & Hofstadter, 1993; Rehling, 2001) and ANALOGY (Evans, 1968), as well as non-visual ones, IDeAL (Bhatta & Goel, 1997; Goel & Bhatta 2004), ToRQUE2 (Griffith, Nersessian, & Goel, 2000), PHINEAS (Falkenhainer, 1990), and Copycat (Hofstadter & Mitchell, 1995), do not attempt to transfer problem-solving procedures.

Many of the systems described above deal with visuospatial reasoning. Though the systems use information of many kinds, including, sometimes, non-visuospatial information, the visuospatial information represented all fall under the categories of *what* is there, *where* it is and finally if and how the components of the image are related (e.g., above/below relationships).[2]

Some analogical reasoning systems use a purely symbolic or propositional representation (e.g., Galatea, GeoRep), some use a pixel or occupancy array representation (e.g., NIAL (Glasgow & Papadias, 1998), WHISPER (Funt, 1980)), some use a hybrid, such as a symbolic array (e.g., NIAL and VAMP.2), and finally one (FROB, Forbus, 1995) uses quantitative measures, such as lengths and distances. There is good reason to think that a variety of representations schemes come into play in cognition (e.g., Farah, 1988; Glasgow & Papadias, 1998; Kosslyn, 1994). In terms of visual representation, Covlan's *primitive visual elements* resemble GeoRep's "primitive shapes." Covlan's *connection* ontology allows orientation-independent transfer of operations in the cognitive modeling, which is important because many experimental participants rotated the target 90°.

Though most diagrammatic reasoning systems include ways to change visual knowledge, Covlan's *transformations* are intended to represent steps in problem-solving procedures that are reasoned about by the system. Griffith, Nersessian, and Goel's "Generic Structural Transformations" (GSTs) (2000), though not specifically visual in nature, are somewhat similar in that they are transformations that are chosen by the system to be applied to a representation in an effort to solve a problem.

6. Conclusion

Recall that our hypothesis was that visuospatial representation of intermediate knowledge states organized in chronological order can enable transfer of problem-solving procedures. We used visuospatially represented problem-solving procedures to model how designers create new solutions by transferring from old ones. When engaged in design-by-analogy, designers might generate new designs by abstracting and transferring problem-solving procedures, where the procedures are expressed in the form of visuospatial representations in which causality is (at most) implicit. In light of our models of novice designers engaged in analogy-based design, we present the following findings: first, a language of visuospatial symbols can provide a level of abstraction sufficient for common actions on concepts. For *all* the Galatea models of these participants, no core processing code was changed. Some *transformations* were added to code, and all participant differences accommodated were done through changes to the input representations only. We modeled the visuospatial input and output for the participants' data—a good start to a full cognitive model. Though people likely use non-visual as well as visual knowledge in analogical problem-solving, this work shows how visuospatial knowledge alone *could* be used. This research also investigates the possible *maximal* role of visual knowledge and reasoning for analogical problem-solving transfer. The Galatea computational model shows that under the conjecture that human participants may have generated a solution to the new design problem by transferring the problem-solving procedure for the source case, temporally organized visuospatial representation of knowledge states generated by the procedure in the source is sufficient for analogical transfer of the procedure to the new problem.

In conclusion, visual and spatial reasoning is useful for many subtasks of analogical problem-solving. Galatea shows that at least in the context of design, analogical transfer can work using only visuospatial knowledge, and other work shows this for the retrieval and mapping stages as well (Davies et al., 2008; Ferguson, 1994; Yaner & Goel, 2006), building a strong case for visuospatial analogy for problem solving.

Acknowledgements

We thank David Craig for use of his data. Davies would also like to thank Janice Glasgow for her support during the writing of this paper. Goel's work on this paper was supported in part by a NSF grant (IIS Award # 0534266) entitled "Multimodal Case-Based Reasoning in Modeling and Design."

References

Bhatta, S. R., & Goel, A. (1997). Learning generic mechanisms for innovative strategies in adaptive design. *The Journal of the Learning Sciences, 6*(4), 367–396.

[2] It could be argued that relations are a part of the "where" class of information, but "where" information is typically conceived as being a location relative to an image, rather than in relation to other visual objects.

Casakin, H. (2004). Visual analogy as a cognitive strategy in the design process: Expert versus novice performance. *Journal of Design Research, 4*(2).

Casakin, H., & Goldschmidt, G. (1999). Expertise and the use of visual analogy: Implications for design education. *Design Studies, 20*, 153–175.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology, 4*(1), 55–81.

Craig, D. L. (2001). Perceptual simulation and analogical reasoning in design. Architecture department Doctoral Dissertation, Georgia Institute of Technology. Technical Report GIT-COGSCI-2001/05.

Croft, D., & Thagard, P. (2002). Dynamic imagery: A computational model of motion and visual analogy. In L. Magnani & N. J. Nersessian (Eds.), *Model-based reasoning: Science, technology, and values* (pp. 259–274).

Davies, J. (2004) Constructive adaptive visual analogy. Doctoral Dissertation. College of Computing, Georgia Institute of Technology. Technical Report GIT-COGSCI-2004/3.

Davies, J., & Goel, A. K. (2001). Visual analogy in problem solving. In *Proceedings of the international joint conference on artificial intelligence* (pp. 377–382).

Davies, J., & Goel, A. K. (2007). Transfer of problem-solving strategy using Covlan. *Journal of Visual Languages and Computing, 18*(2), 149–164.

Davies, J., & Goel, A. K. (2008). Visuospatial re-representation in analogical reasoning. *The Open Artificial Intelligence Journal, 2*, 11–20.

Davies, J., Goel, A. K., & Yaner, P. W. (2008). Proteus: Visuospatial analogy in problem solving. *Knowledge-Based Systems, 27*(7), 636–654.

Duncker, K. (1926). A qualitative (experimental and theoretical) study of productive thinking (solving of comprehensible problems). *Journal of Genetic Psychology, 33*, 264–708.

Evans, T. G. (1968). A heuristic program to solve geometric analogy problems. In M. Minsky (Ed.), *Semantic information processing*. Cambridge, MA: MIT Press.

Falkenhainer, B. (1990). A unified approach to explanation and theory formation. In J. Shrager & P. Langley (Eds.), *Computational models of scientific discovery and theory formation* (pp. 157–196). San Meteo, CA: Morgan Kaufman.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1990). The structure mapping engine: Algorithm and examples. *Artificial Intelligence, 41*, 1–63.

Farah, M. J. (1988). The neuropsychology of mental imagery: Converging evidence from brain-damaged and normal participants. In *Spatial cognition-brain bases and development*. Erlbaum.

Ferguson, E. S. (1992). *Engineering and the mind's eye*. Cambridge, MA: MIT Press.

Ferguson, R. W. (1994). MAGI: Analogy-based encoding using regularity and symmetry. In A. Ram & K. Eiselt (Eds.) *Proceedings of the 16th annual conference of the cognitive science society* (pp. 283–288).

Ferguson, R. W., & Forbus, K. D. (1998). Telling juxtapositions: Using repetition and alignable difference in diagram understanding. In K. Holyoak, D. Gentner, & B. Kokinov (Eds.), *Advances in analogy research* (pp. 109–117). Sofia: New Bulgarian University.

Ferguson, R. W., & Forbus, K. D. (2000). GeoRep: A flexible tool for spatial representation of line drawings. In *Proceedings of the 18th national conference on artificial intelligence*. Austin, TX: AAAI Press.

Forbus, K. D. (1995). Qualitative spatial reasoning framework and frontiers. In J. Glasgow, N. H. A. Narayanan, & B. Chandrasekaran (Eds.), *Diagrammatic reasoning* (pp. 183–202). Austin, TX: AAAI Press.

Funt, B. V. (1980). Problem-solving with diagrammatic representations. *Artificial Intelligence, 13*(3), 201–230.

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

Glasgow, J., & Papadias, D. (1998). Computational imagery. In P. Thagard (Ed.), *Mind readings*. Cambridge, MA: MIT Press.

Goel, A. K., & Bhatta, S. R. (2004). Design patterns: An unit of analogical transfer in creative design. *Advanced Engineering Informatics, 18*(2), 85–94.

Goldschmidt, G. (2001). Visual analogy – A strategy for design reasoning and learning. In C. Eastman, W. Newsletter, & M. McCracken (Eds.), *Design knowing and learning: Cognition in design education* (pp. 199–219). New York: Elsevier.

Griffith, T. W., Nersessian, N. J., & Goel, A. K. (2000). Function-follows-form transformations in scientific problem solving. In *Proceedings of the 22nd annual conference of the cognitive science society*. Mahwah, NJ: Lawrence Erlbaum Associates.

Gross, M. D., & Do, E. (2000). Drawing on the back of an envelope: A framework for interacting with application programs by freehand drawing. *Computers and Graphics, 24*, 835–849.

Hammond, K. J. (1990). Case-based planning: A framework for planning from experience. *Cognitive Science*.

Hofstadter, D. R., & Mitchell, M. (1995). The copycat project: A model of mental fluidity and analogy-making. In D. Hofstadter & The Fluid Analogies Research group (Eds.), *Fluid concepts and creative analogies* (pp. 205–267). Basic Books.

Holyoak, K. J., & Thagard, P. (1989a). A computational model of analogical problem solving. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 242–266). Cambridge: Cambridge University Press.

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

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

Hummel, J., & Holyoak, K. J. (1996). Lisa: A computational model of analogical inference and schema induction. In G. Cottrell (Ed.), *Proceedings of the 18th annual conference of the cognitive science society*.

Kokinov, B. (1998). Analogy is like cognition: Dynamic, emergent, and context-sensitive. In K. Holyoak, D. Gentner, & B. Kokinov (Eds.), *Advances in analogy research: Integration of theory and data from the cognitive, computational, and neural sciences* (pp. 96–105). Sofia: NBU Press.

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

Larkin, J., & Simon, H. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science, 11*, 65–99.

McGraw, G., & Hofstadter, D. R. (1993). Perception and creation of alphabetic style. In *Artificial intelligence and creativity: Papers from the 1993 spring symposium*. AAAI Technical Report SS-93-01, AAAI Press.

Narayanan, N. H., Suwa, M. & Motoda, H. (1994). How things appear to work: Predicting behaviors from device diagrams. In *Proceedings of the 12th national conference on artificial intelligence* (pp. 1161–1167). AAAI Press.

Rehling, J. A. (2001). Letter spirit (part two): Modeling creativity in a visual domain. Indiana University, Ph.D. thesis.

Schmid, U., & Carbonell, J. (1999). Empirical evidence for derivational analogy. In M. Hahn & S. C. Stoness (Eds.), *Proceedings of the 21st annual conference of the cognitive science society*.

Thagard, P., Gochfeld, D., & Hardy, S. (1992). Visual analogical mapping. In *Proceedings of the 14th annual conference of the cognitive science society* (pp. 522–527). Erlbaum.

Ullman, D. G., Wood, S., & Craig, D. (1990). The importance of drawing in the mechanical design process. *Computer Graphics, 14*(2), 263–274.

Veloso, M. M. (1993). Prodigy/analogy: Analogical reasoning in general problem solving. *EWCBR*, 33–52.

Veloso, M. M., & Carbonell, J. G. (1993). Derivational analogy in PRODIGY: Automating case acquisition, storage, and utilization. *Machine Learning, 10*(3), 249–278.

Yaner, P. W., & Goel, A. K. (2006). Visual analogy: Viewing retrieval and mapping as constraint satisfaction. *Journal of Applied Intelligence, 25*(1), 91–105.