Transfer of Problem-Solving Strategy Using the Cognitive Visual Language

Jim Davies, Ashok K. Goel and Nancy J. Nersessian

Abstract— **We present Covlan, a visual language intended to model the diagrammatic memories of human beings, and Galatea, a computational model of analogical visual problem solving. Galatea and the experimental participants modeled in it show that 1) problem-solving procedures can be effectively represented with a visual language, and 2) the successful transfer of strongly-ordered procedures in which new objects are created requires the reasoner to generate intermediate knowledge states and mappings between the intermediate knowledge states of the source and target cases. We describe Galatea and Covlan, a model created with it, and related work.**

Index Terms—**Artificial Intelligence, Cognitive Science, Visual Languages**

I. Introduction

Visual Languages have been created for fields such as programming, diagrammatic reasoning, and cognitive modeling. In this work we describe Covlan, the Cognitive Visual Language, which is intended to describe visually-represented episodes of problem-solving procedures in human beings.

Some domains are inherently non-visual, but might be visually represented all the same. For example, effectively connecting a battery to some wires might be represented, among other ways, functionally (the battery needs to be physically touching the metal of the wire to conduct electricity) or visually (the image of the wire is adjacent to the image of the battery.) Even though other kinds of knowledge and representations might be used to reason about these domains, human beings appear to experience visual imagery when reasoning about them. Experimental evidence indicates that visual knowledge often plays an important role in human problem solving [7, 2, 11]. There is also documentary evidence for visual reasoning in scientific problem solving (e.g. [12]). Further, psychological evidence suggests that *analogical* problem solving is facilitated by animations [13], diagrams [1] as well as visually evocative phrases in stimuli [9]. These results suggest that visual representations have an important function in cognition, and that problem-solving procedures might be usefully represented with a visual language. Our hypothesis, then, is that problem-solving procedures can be effectively represented using a visual language.

Covlan is a model of the symbolic visual memories people have of problem-solving procedures. We have designed and implemented Galatea, a computer program that uses analogs represented in Covlan to infer problem-solving solutions to new problems [4]. To support Galatea as a model of human visual problem-solving, we modeled four experimental participants who solved a visual analogy problem, one of which we will describe here.

Dr. David Craig ran 34 participants in an analogical transfer experiment [3]. Participants were shown a problem-solving solution with a laboratory, presented with text and a diagram. They were asked to solve an analogous problem with a weed-trimmer, presented with text only. Of these, 17 participants (in three conditions) correctly described the analogous solution. All participants were asked to draw a diagram to illustrate their suggested solutions (See Figure 1 for the participant stimuli). A laboratory clean room strategy is transferred by adding redundant doors to a weed-trimmer arm so that it can pass through street signs. The analogous solution is to design an arm with two latching doors, so that while one is open to let the sign pass, the other stays closed to support the arm and trimmer (See Figure 2 for one participant's data). Participants produced diagrams describing their solutions to the problems. We modeled four of these experimental participants in Galatea: L14, L15, L16, and L22. We will describe in detail our model of one of these participants, L14, in this paper, and briefly describe the results of modeling the other three. We use our ability to model these participant data as an evaluation of Covlan and Galatea as a cognitive theory.

Manuscript received March 31, 2005.

J. Davies is a postdoctoral fellow at the School of Computing, Queen's University, Kingston, Ontario, Canada, K7L 3N6. (1-613-532-8828; fax: 1-613-533-6513; e-mail: jim@jimdavies.org).

A. K. Goel is an associate professor in the College of Computing, Georgia Institute of Technology, Atlanta, Georgia, USA. (404-894-4994; e-mail: goel@cc.gatech.edu).

N. J. Nersessian is a professor in the College of Computing, Georgia Institute of Technology, Atlanta, Georgia, USA. (404-894-1232; e-mail: nancyn@cc.gatech.edu).

II. Galatea

A. Overview

Analogical problem solving involves several steps: An intelligent agent (e.g., a human or an artificial intelligence program) starts with an unsolved problem, the *target*, and *retrieves* a similar *source*, or *base*, episode of a solved problem. Then the elements of the source are *mapped* to the elements of the target. This means finding alignments between the sub-parts of the problems. Then the source's solution is *transferred* to the target, perhaps with some adaptation. Then the solution is *evaluated* and finally *stored* in memory. Most analogical AI systems model analogical mapping. Galatea models the transfer stage of analogical problem solving.

The modeling architecture used to model L14, one of the experimental participants, is an implemented computer program called Galatea. The issue is how an analogical problem solver might represent its diagrammatic knowledge of the source case and target problem, and how might it transfer the relevant problem-solving steps from the source to the target?

Galatea represents a source case as a series of knowledge states starting from the initial knowledge state and ending in the final or goal knowledge state. A knowledge state is represented diagrammatically in the form of shapes, their locations, sizes, and motions (if any), and the spatial relationships among the shapes.

Succeeding states in the series of knowledge states are related through visual transformations such as move, rotate, scale and decompose. Each transformation relates two knowledge states. Transfer works by applying, step by step, each transformation in the source case to the knowledge states of the target case.

B. Knowledge Representation: Covlan

Galatea describes visual cases using Covlan, which consists of knowledge states, primitive elements, primitive relations, primitive transformations, general visual concepts, and correspondence and transform representations. In Covlan, all knowledge is represented as propositions relating two elements with a relation.

Knowledge States: Knowledge states in Covlan are symbolic images, or *s-images*, which contain visual elements, general visual concepts, and relations between them. Knowledge states are represented by a series of *s-images*, connected with transformations.

Visual Transformations. An *s-image* in the sequence is connected to other *s-images* before and after it with visual transformations. Transformations, like ordinary functions, take arguments to specify their behavior.

These transformations control normal graphics transformations such as translation (*move-to-location*), and rotation (*rotate*). In addition there are transformations for adding and removing elements from the *s-image* (*add-element, remove-element*). Certain transformations (*start-rotating, stop-rotating, start-translation, stop-translation*) are changes to the dynamic behavior of the system under simulation. For example, *rotate* changes the initial orientation of an element, but in contrast *start-rotating* sets an element in motion.

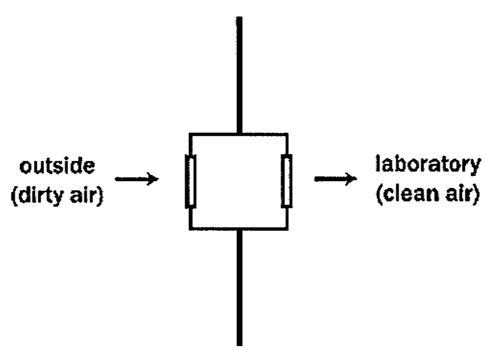
Primitive Elements are the visual objects in a diagram. The element types are *rectangle, circle, arrow, line,* and *curve.* Each element is represented as a frame with attribute slots, such as *location, size, orientation,* or *thickness.* A particular example of an element is referred to as an *element instance.*

General Visual Concepts. These act as slot values for the primitive elements as well as arguments for the visual transformations. The concepts are *location, size, thickness, speed, direction, length, distance, angle,* and *direction.* Each concept has several values it can take. For example, the *size*

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.

Figure 1: The stimuli shown to L14 in the experiment.

In the space below, try to design a weed-trimmer pole that can pass through sign posts. Draw a diagram to illustrate what you're thinking.

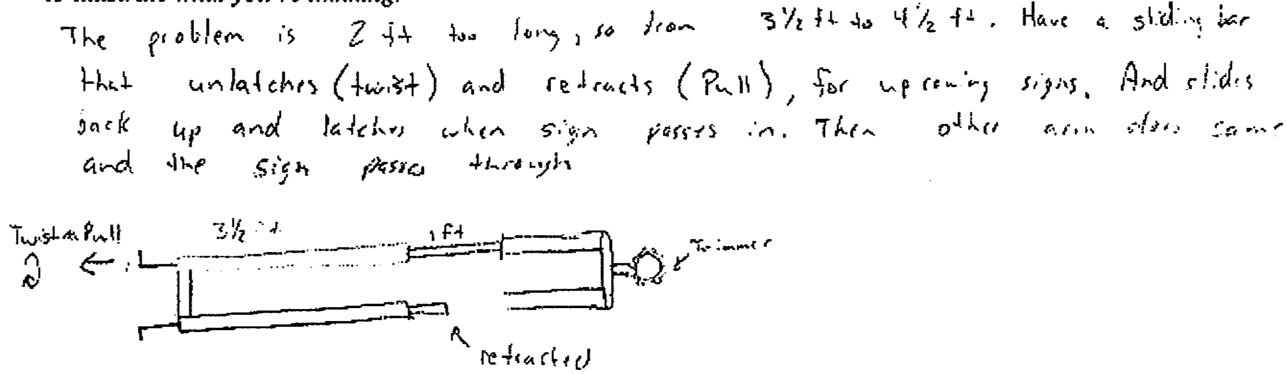


Figure 2: What L15 drew on the experiment sheet.

can be *small, medium,* or *large,* and *thickness* can be *thin, thick* or *very-thick. Location* specifies an absolute qualitative location in an *s-image* (*bottom, top, center,* etc.)

Primitive Visual Relations. This class of symbols describes how certain visual elements relate to each other and to the values taken by general visual concepts. The visual relations are *touching, above-below,* and *right-of-left-of.* The motion relation is *rotation.*

Correspondence and Transform Representations. The knowledge of which objects in one *s-image* correspond to which objects in another is a *mapping,* which consists of a set of alignments between objects. Different sets of alignments compose different mappings. The i^{th} *s-image* in the source and the i^{th} *s-image* in the target have a *correspondence* between them; each correspondence can have any number of *mappings* associated with it (determining which mapping is the best is the "mapping problem.") The correspondence and mapping between the initial *s-images* (i=1) in the source and target is given as part of the input to Galatea; the system generates the subsequent correspondences and mappings for the subsequent *s-images.*

Similarly, successive *s-images* in a series have *transform-connections.* These are needed so that Galatea can track how visual elements in a previous knowledge state change in the next.

C. Algorithm

Following is the control structure for Galatea's transfer of problem-solving procedures from a source case to the target problem. Figure 3 shows the *s-image* structure for L14's problem and solution. The algorithm below references Figure 3.

The solution procedure (for the source, and then for the target) is that the doorway mechanism gets replicated, and then moved to the correct positions. Two walls are created to complete the vestibule, and finally they are placed in the correct position so that the vestibule is complete.

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 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.* The model of L14 involves five transformations (see Figure 3). The first transformation is *replicate.* 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 adds 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 and the top and bottom walls.
3. **Identify the objects of the transformations.** The object of the transformation is what *object* the transformation acts upon. For L14's first transformation, this object is the parts of the door in the first *s-image* (we'll call it *door-set-l14s1*).
4. **Identify the corresponding objects in the target problem.** In the target, the trimmer arm's door mechanism is the corresponding object.
5. **Apply the transformation with the arguments to the target problem component.** A new *s-image* is generated for the target problem to record the effects of the transformation. *Replicate* takes two arguments: some *object* and some *number-of-resultants.* In this case the *object* is *door-set-b1s1* (b1s1 means "base one, *s-image* two") and the *number-of-arguments* is two. The *replicate* is applied to the first L14 *s-image,* with the appropriate adaptation to the arguments: The mapping between the first source and target *s-images* indicates that the *door-set-b1s1* maps to the *door-set-l14s1,* so the former is used for the target's object argument. The

number *two* is a literal, so it is transferred directly. Galatea generates *door-set1-l14s2* and *door-set2-l14s2* in the next *s-image.*

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. It takes *connection-sets-set-b1s3* as the *connection/connection-set* argument. This is a set containing four connections. Galatea uses a function to recursively retrieve all connection and set proposition members of this set. These propositions are put through a function which creates new propositions for the target. Each proposition's relation and literals are kept the same. The element instance names are changed to newly-generated analogous names. For example, *door1-endpoint-b1s3* turns into *door1-endpoint-l14s3.*

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.*

The inputs to this transformation are *nothing* (a literal denoting that there is not any thing in the previous *s-image* that is being modified), the connection set *connection-sets-set-b1s3,* the source *s-image lab-base1-simage2,* the current and next target *s-images l14-simage2* and *l14-simage3,* the mapping *l14-simage2—l14-simage3-mapping1,* and the rest of the memory.

6. **Map the original objects to the new objects in the target case**. 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 door-sets, as well as their component objects. Galatea does not solve the mapping problem, but a mapping from the correspondences of the first *s-image* enables Galatea to automatically generate the mappings for the subsequent *s-images.*

7. **Map the new objects of the target case to the corresponding objects in the source case**. Here the parts of the door set in the target *s-image* are mapped to the parts 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 which parts of the target *s-image* the later transformations would operate on.

8. **Check to see if goal conditions are satisfied**. If they are, exit, and the solution is transferred. 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.

We can now evaluate what made L14's data (Figure 2) differ from the stimulus drawing (Figure 1): 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 to keep the spinning trimmer blade from hitting the vestibule. The entire drawing is rotated ninety degrees 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 design of the trimmer: to describe length. Finally, L14 includes some mechanistic description of how the trimmer would work.

Of these seven differences, our model successfully re-creates four of them. The *rotation* of the source is modeled by a rotation in the target start *s-image.* In the *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

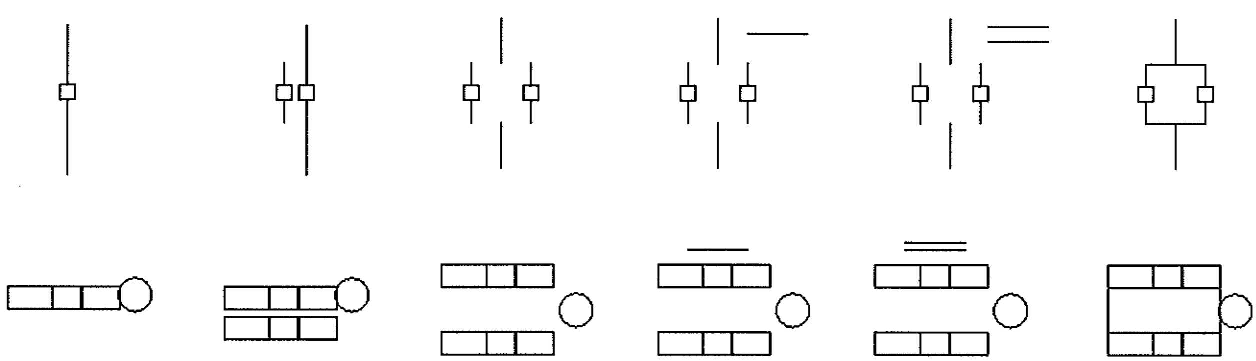


Figure 3: The Galatea Model of L14

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 transfer. 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 trimmer head *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, 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*.

Though we do not have the space to report the details here, we implemented three other models from the Craig et al. experiment. The four in total best represented the difficulty and variety of the participants we could have modeled. In all cases, our models accounted for the majority of the differences between the participant drawings and the stimulus given, supporting Covlan and Galatea as models of human thought.

III. Related Work

Though much work on visual languages involves visual programming aids, there are some visual languages meant to model the internal visio-spatial representations in people's minds. Liu's PI system [10] represents objects and operations in the Euclidean geometry domain, but is not intended as a cognitive model of human visual thought. GeoRep [8] uses a set of "primitive shapes" that resemble Covlan's visual elements. It is intended to be a model of human visual reasoning. Erwig and Schneider [6] represents changes for visual objects, but not in terms of actions taken on a visual system. Their system allows queries with respect to what things have happened (e.g. has a tornado ever passed through Iowa?). Covlan, to our knowledge, is the first to explicitly represent sequences of visual operations. Covlan also manages the complexity involved with, for example, adding objects and keeping track of what is done with them. And with regard to the systems that use visual languages, Galatea is the first to transfer visually represented problem-solving procedures.

IV. Conclusions

Our hypothesis, as stated in the introduction is that problem-solving procedures can be effectively represented using a visual language. We have presented the Covlan, a visual language designed to describe problem-solving situations and Galatea, the artificial intelligence system that uses Covlan representations to transfer sequences of visual actions to a target problem. Our hypothesis was supported by the evidence described above, and we had an unexpected discovery for a total of two claims.

First, problem-solving procedures can be effectively represented using a visual language. There are seven models written in Covlan and Galatea that support this claim. We described the model of L14 in this paper. In addition we modeled three additional participants from the Craig *et al.* experiment, a historical example from the scientific thinking of Maxwell [5], the fortress/tumor problem [4] and the cake/pizza problem [4]. Each of these models uses analogical reasoning to solve a problem using only visual knowledge. The fact that four of these models are based on human experimental participant data lend support to the idea that the claim might apply to human problem solving, as well as artificial analogy systems, although more empirical research would be needed to substantiate this. As shown above, most of the differences between source and target, as displayed in the participant data, were accounted for in our models.

In the course of building the models of Galatea, we discovered that the successful transfer of strongly-ordered procedures in which new objects are created requires the reasoner to generate intermediate knowledge states and mappings between the intermediate knowledge states of the source and target cases. Galatea shows why, in detail, this is so. Components of the problem are *created* by the operations, and these components are acted on by later operations. For L14's problem, for example, the door set must be replicated before the two sets can be moved in relation to one another. When the reasoner transfers the second operation of moving the door sets, how does it know what the corresponding objects are in the target? It must have some mapping to make this inference. And since one of the door sets did not exist in the start states of the problems, this mapping cannot be given as input with the initial mapping. The new knowledge state with the duplicated door set must be generated, and then a mapping must be made on the fly between it and the second knowledge state of the source case.

In conclusion, Covlan appears to be a good start for a full cognitive language of visual representations of procedural strategy.

Acknowledgment

Thank you to Dr. David Craig for the use of his experimental data.

References

[1] Beveridge, M. & Parkins, E. Visual representation in analogical problem solving. *Memory & Cognition.* 15(3) (1987) 230--237

[2] Casakin, H., Goldschmidt, G.: Expertise and the use of visual analogy: Implications for design education. *Design Studies.* (1999)

[3] Craig, D. L., Catrambone, R., Nersessian, N. J.: Perceptual simulation in analogical problem solving. In *Model-Based Reasoning: Science, Technology, & Values.* New York: Kluwer Academic / Plenum Pubishers. (2002) 167–191

[4] Davies, J., Goel, A. K.: Visual case-based reasoning I: Transfer and adaptation. *Proceedings of the 1st Indian International Conference on Artificial Intelligence.* Springer. (2003)

[5] Davies, J., Nersessian, N. J., Goel, A. K.: Visual models in analogical problem solving. *Foundations of Science, Special Issue on Model-Based Reasoning: Visual, Analogical, Simulative.* By Magnani, L. and Nersessian, N.J. (Eds.) (in press)

[6] Erwig, M., Schneider, M. *Journal of Visual Languages and Computing, Vol. 14*, No. 2, 181-211 (2003)

[7] Farah, M. J.: The neuropsychology of mental imagery: Converging evidence from brain-damaged and normal subjects. In *Spatial Cognition-Brain Bases and Development.* Erlbaum (1988)

[8] Ferguson, R. W., & Forbus, K. D. GeoRep: A flexible tool for spatial representation of line drawings, *Proceedings of the 18th National Conference on Artificial Intelligence*. Austin, Texas: AAAI Press. (2000)

[9] Gick, M. L. & Holyoak, K. J. Schema induction and analogical transfer. *Cognitive Psychology.* 12. (1980) 306--355

[10] Liu, Zhiqing, Semantics Design of a Visual Language for Constructing and Animating Geometric Objects. In the 1999 IEEE Symposium on Visual Languages, Tokyo, Japan, September (1999)

[11] Monaghan, J. M., Clement, J.: Use of computer simulation to develop mental simulations for understanding relative motion concepts. *International Journal of Science Education.21(9).* (1999) 921--944

[12] Nersessian, N. J. Maxwell's `Newtonian aether-field'. In Nersessian, N. J. *Faraday to Einstein: Constructing Meaning in Scientific Theories.* Kluwer, Dordrecht. (1984). 68-93.

[13] Pedone, R., Hummel, J. E. & Holyoak, K. K. The use of diagrams in analogical problem solving. *Memory & Cognition.* (2001)