

Visual Models in Analogical Problem Solving

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

Abstract

Visual analogy is believed to be important in human problem solving. Yet, there are few computational models of visual analogy. In this paper, we present a preliminary computational model of visual analogy in problem solving. The model is instantiated in a computer program, called Galatea, which uses a language for representing and transferring visual information called Privlan. We describe how the computational model can account for a small slice of a cognitive-historical analysis of Maxwell's reasoning about electromagnetism.

Introduction

Both analogical and visual reasoning are known to be important in human problem solving. Visual analogy, i.e., analogy using visual information, itself is believed to be an important human problem solving strategy (Clement 1994a; 1994b; Do and Gross, 1995a; 1995b; Goldschmidt, 1995; Pedone et al., 2001). The history of science too provides examples of analogical problem solving that can be interpreted as employing visual information, such as Kekule's solution of the problem of the arrangement of the atoms in benzene through a visual analogy with a snake biting its tail (Boden, 1990) and Faraday's use of the visual representation of the lines of force in providing a solution to the problem of continuous transmission of force in space (Nersessian 1984; 1992; Gooding, 1994).

While there exist several well-developed computational models of analogy in general (e.g. Falkenhainer et al., 1990; Bhatta and Goel, 1997a, 1997b; Holyoak and Thagard, 1997), there are relatively few computational models of visual analogy (but see Thagard et al., 1992; Croft and Thagard 2000; McGraw and Hofstadter, 1993; Ferguson and Forbus, 2000). Given that existing computational models of analogy have provided useful insights into the role of mental models in recognizing and making analogies, one would expect that computational models of visual analogy similarly may provide insights into the role of visual models in visual analogy. This paper first describes a computational model of visual analogy in problem solving. The model is instantiated in a computer program, called Galatea, which uses a knowledge representation language called Privlan (Davies and Goel, 2001). Then, we describe how the computational can be extended to account for a small portion of a cognitive-historical analysis of Maxwell's reasoning about electromagnetism (Nersessian, 1984,1992, 1994b, 2001).

Visual Information in Analogy

Visual analogy employs representations of visual information. For our purposes visual information is information that is relevant to how an image appears, i.e., it is information

that describes an image. Visual representations can include descriptions of what is in the image, such as color, location, motion, connections, overlap, distance, and geometrical and topological structure. Note that this description includes both high-level symbolic representations and low-level bitmap representations. What would *disqualify* a representation from being visual is information not relevant to how it is pictured, for example, what is inside opaque objects, sounds, causal relations, teleology etc.

Our characterization of visual representations differs from that of Glasgow and Papadias (1998), who distinguish *visual* from *spatial* representations along the lines of the different what/where pathways in the brain. Visual representations describe what is in an image and spatial representations describe where those elements are and their spatial relations with one another. Both of these are visual representations on our account.

Some of our earlier computational models of analogy (e.g. Bhatta and Goel 1997a; 1997b; Griffith et al., 1996; 2000) represented *structural* knowledge, which described a system's physical composition but generally included only the information directly relevant for analyzing and simulating the causal behaviors of the system. Like a schematic, structural knowledge shows the components of the system and the connections among them, but leaves out other visual information, such as what a component wire looks like and which side of a pump is up.

Following Kosslyn (1994) two ways of representing images are: as depictive representations (or bitmaps), and as descriptive representations. A depictive representation "specifies the locations and values of points in space" (Kosslyn, 1994, p.5). Thus, a depictive knowledge representation is similar to raw data; it is uninterpreted. For example, to know that there is a line and a box in a depictive representation one needs to apply perceptual processes to it. Descriptive representations are symbolic networks that represent the contents of an image. A system must *interpret* it to turn it into a depictive representation. In graphics this is called *rendering*. In the computational system we will be discussing here all images are represented symbolically, as they are in most computational systems of visual analogy (e.g.; Croft and Thagard, 2000; Ferguson and Forbus, 2000). The visual elements are related to each other with labeled, directional links. Analogical mapping is done based on the similarity of the relations or nodes, with the different systems determining similarity in different ways.

Knowledge and Inference in Analogy

Our earlier work on **model-based analogy** (e.g., Bhatta and Goel, 1997a; 1997b) informs many of the design decisions in this work. A theme of model-based analogy is creating ontologies of useful abstractions by making claims about *what kinds of inferences* are needed and *what kinds of knowledge* are required to draw the needed inferences. This functional, top-down approach contrasts with more bottom-up architectural approaches to knowledge representation. For example, a bottom-up theory might specify that knowledge is represented as chunks or as productions. In our work, we postulate specific kinds of knowledge that need to be encoded to enable particular kinds of inferences. For example,

the Kritik system (Goel 1991a; 1991b; Goel et al., 1997;) represented knowledge of the functioning of physical devices in terms of **structure, behavior and function** models (Chandrasekaran et al., 1993; Prabhakar and Goel, 1996). The primitives of the SBF language enabled the inferences needed to retrieve and adapt previous design cases to solve new design problems. Similarly, the Ideal system, (Bhatta and Goel, 1997a) used a language of **generic physical principles** and **generic teleological mechanisms**, which are useful units of analogical transfer in creative device design. Generic teleological mechanisms provide a taxonomy of functional and causal transformations to physical devices. In contrast, the Torque system (Griffith et al., 1996, 2001) used a taxonomy of **generic structural transformations** that could be applied to physical systems. These transformations were found to be useful in modeling a protocol of a human subject solving a problem dealing with spring systems.

Following the themes of our earlier work on analogy, for our current work on visual analogy we needed both a vocabulary for expressing the knowledge content of known analogs and new problems and a vocabulary for capturing generic transformations for the analogs and the problems. That is, we needed both a language for capturing visual abstractions and a taxonomy of visual transformations. We call the resulting language **Privlan** [Davies and Goel 2001]. Privlan has major two components: **Privels** and **Privits**. Privels (for primitive visual elements) provides a vocabulary for representing visual information in an analog or a problem. Privits (for primitive visual transformations) provide a taxonomy of transformations that constitute a problem solving strategy. Since we are modeling problem solving, the transformations refer to specific actions that an agent can take. They are the steps of the problem solving procedure, rather than snapshots of how a system behaves. For example, adding a gear between wheels is a transformation, but the motion of the gears is a simulative behavior, not a transformation.

Galatea: A Computational Model of Visual Analogy

Since the images are represented symbolically, we will refer to them as **simages**. The simages are linked into a series, like a filmstrip, with each simage representing a step in the problem solving process. The simages are connected with one another with privits. Specifically, privits connect elements of a previous simage to their changed counterparts in the next.

Our first case is the classic Duncker's Fortress/Tumor problem (Duncker, 1926). This example was chosen because psychological data indicate that experimental participants used visual information in solving it (Holyoak and Thagard, 1997; Pedone et al., 2001). In this task, participants first read a story about a problem-solving situation: A general with a large army wants to overthrow a dictator who lives in a fortress. All roads to the fortress are armed with mines that will go off if many people are on them at the same time. To solve this problem he breaks up his army into small groups and has them take different roads. The groups arrive at the same time and take the fortress. The participants are then given a new problem: A patient needs radiation treatment on a tumor inside the body, but

the radiation will harm the healthy tissue it reaches on the way in. Finally, the participants are asked to solve the tumor problem. The analogous solution is to target the tumor with low-level rays coming from different directions and have them converge on the tumor. We assume that the design of the representation language used should be determined in part by the processing to be done on those representations. In both the fortress and tumor problems, the active force must be made into several weaker versions of itself and put in different locations. This suggests the abstract transformation privits **decompose** and **move**. Privels are introduced for the transformations to act upon. In this case, the army and the ray of radiation are what get decomposed, so we need a visual element that can describe both of these things. Since a rich visual representation of the army and the ray look nothing alike, the representation used must be an abstraction. One such abstraction lies in the visual similarity both share: the shape and direction of the path taken. The ray's path is filled with light and the army's path is filled with the army at different points at different times. To capture this similarity we abstracted over time and represented the paths as lines. The abstraction **line** is the first privel. This choice involves a hypothesis about how domains are transformed into abstract symbolic images. Objects with translational motion are abstracted into privels that visually resemble the path of that motion, as needed in the fortress/tumor problem. This hypothesis allows the transformation to apply to both problems and thus has implications for generalization. The system gains power in that by abstracting motion into line shapes, it can efficiently find similarities between dynamic systems. The trade off is that complex motions, particularly those with causal interactions, can be hard to represent statically and important information can be lost. For complex causal behaviors, this kind of abstraction will be less useful. Our representations are currently not dynamic because we abstract motion over time into static paths. The roads are also represented as lines, because that is their basic shape.

Continuing the process we abstracted **generic-visual-element** as the next privel. The fortress and tumor are at the end of the path lines, but the shape is not relevant to the problem. Also, the shapes of the body parts and around the tumor and mines are not important, so these things can be represented generically.

Privels are represented as frames that have attributes that take on values. A **line**, for example, can be thick or thin with respect to its attribute *thickness*. These attributes are not strongly typed, meaning that any kind of entity can fill the slots. For example, the *end-point* of a **line** could be a location such as the "center" of the image or some component of the image, like the fortress. Table 1 shows some privels.

Insert Table 1 about here

Figure 1 is a diagram of how Galatea represents problem solving solutions in memory and transfers that knowledge to new problems. The series of ovals running left to right represent simage series. The top series represents a case in memory. The first simage is the problem input, or the start state. The last simage in this series represents the final solution state of the problem. The arrows pointing left to right represent transform-

connections, which link the simages in a series. Associated with each transform-connection is a privit, which describes the action to be taken at that step. The bottom left oval represents the start state of the target problem. There is an analogy between the start state of the given problem and the source problem in memory, which provides analogical mappings between the privels of the two simages. The items in the shaded box are what get generated. According to our theory, in transferring the problem solution to the target, the same transformations are applied to the target as those in the source problem, in the same order. Each time, the objects of the transformation are the elements analogous to those in the source. Since the simages are different throughout the series, new analogies must be made at each step of the transformation. In the end, the procedure from the source has been completely applied to the target, resulting in a new simage series, as shown in the row of ovals along the bottom of Figure 1. We describe the algorithm in more detail in the next section.

Insert Figure 1 about here

Duncker's Fortress/Tumor problem

Part of the representation for the fortress/tumor problem is shown in Figure 2. The fortress problem is represented as the three bottom simages. The fortresses are represented as **generic-visual-elements** and the roads and soldier paths are **lines**. The first simage is connected to the second with a transform-connection, which is not shown. This transform-connection specifies maps between the components of the adjacent simages in the sequence. Note the maps between the *soldier-path1* and the four soldier paths in the second simage. This map is associated with the decompose privit. The other maps are not shown.

Insert Figure 2 about here

The top left simage is the initial state simage for the tumor problem. The tumor is a **generic-visual-element** and the body parts are too. The system starts with these and an analogy between the first simage in each series. That analogy specifies the maps between the components. Pictured is the map between the *left-body1* and *left-road1*. The other maps are not pictured.

The system outputs what is in the darkly shaded box. To get the second tumor problem simage, the system finds the first privit for the fortress problem, **decompose**; finds what it is acting on, *soldier-path1*; and applies the same privit to the corresponding component of the first tumor simage. The result of this transformation is put in a new simage, on the top middle. **Decompose** makes the resultant lines thinner.

The next transformation is **move**, which moves the soldier paths in the source to the different roads. To transfer this, an analogy is used between the second simages of the two problems. **Move** applies to the soldier paths, so in the tumor problem it applies to the rays.

Galatea creates a new simage, and the new rays are placed there, moved to their new positions.

Algorithm

The control structure for our visual analogical transfer theory is as follows. We will describe the transfer of the first privit as a running example. The process in the abstract can be seen in Figure 1.

1. Identify the first simages of the target and analog problems.

2. Identify the privits and associated arguments in the current simage of the source analog. This step finds out how the source problem gets from the current simage to the next simage. In our example, the privit is **decompose**, with “four” as the *number-of-resultants* argument (not shown).

3. Identify the objects of the privits. The object of the privit is what object the privit acts on. For the **decompose** privit is the *soldier-path1* (the thick arrow in the bottom left simage.)

4. Identify the corresponding objects in the target analog. The *ray1* (the thick arrow in the top left simage) is the corresponding component of the source analog's *soldier-path1*, as specified by the analogical map between the simages (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 privit is applied to all of them in the next step. If the privit *arguments* are components of the source simage, then their analogs are found as well. Otherwise the arguments are transferred literally.

5. Apply the privit with the arguments to the target analog component. A new simage is generated for the target problem (top middle) to record the effects of the privit. The **decompose** privit is applied to the *ray1*, with the argument “four.” The result can be seen in the top middle simage in Figure 2. The new rays are created for this simage.

6. Map the original objects to the new objects in the target problem. A transform-connection and mapping are created between the target problem simage and the new simage (not shown). Maps are created between the corresponding objects. In this example it would mean a map between *ray1* in the first top simage and the four rays in the second top simage. The privit is associated with the map, as shown in the Figure, so the target problem itself can be used as a possible source analog in the future.

7. Map the new objects of the target problem to the corresponding objects in the source problem. In this case the rays of the second target simage are mapped to soldier paths in the second source simage. This step is necessary for the later iterations (i.e. going on to another transformation and simage). Otherwise the system would have no way of knowing which parts of the target simage the later privits would operate on.

8. Check to see if goal conditions are satisfied. If they are, exit, and the problem is solved. If not, and there are further simages in the source series, set the current simage equal to the next simage and go to step

9. If there are no further simages, then exit and fail.

Maxwell Case

Our next project was to see how much of our primitive visualization language could be applied to another example and to what extent it would need extension. We applied it to a piece of the example of the model construction process James Clerk Maxwell used in deriving the electromagnetic field equations. The interpretation we employed is taken from Nersessian's cognitive-historical analysis of James Clerk Maxwell's problem solving (Nersessian, 1984; 1992; 1994a; 1994b; 2001). We will present it only in very broad terms and refer you to Nersessian's extensive research for the details.

The general problem Maxwell was dealing with was the mathematization of the electromagnetic field concept. In Maxwell's model of electromagnetism, the ether between magnets swirl into vortices, which are all spinning in the same direction. The spinning causes the vortices to shorten, pulling the magnets together. Maxwell constructed this vortex-fluid model through an analogy with continuum mechanics.

Insert Figure 3 about here

Figure 3 is drawn from Maxwell's description of the aether and the vortices. We do not assume his mental model had this level of detail. In thinking about how electricity relates to magnetism he needed to consider multiple vortices and their interaction. We hypothesize that a generic cross section as drawn in Figure 4 approximates his mental model at this stage of problem solving. Making topological changes of this kind to imagined physical systems has been shown in our earlier work to be useful in problem solving. (Griffith et al., 2000; Griffith et al., 1996).

Insert Figure 4 about here

Since the vortices were all assumed to be spinning in the same direction, Maxwell found a problem in the model: Friction would cause the vortices to slow or stop. Figure 5, drawn by Maxwell, shows his solution to this problem. He introduced what he called "idle wheel particles" spinning in the opposite direction between the vortices. This model was used in the further derivation of the mathematical laws of the electromagnetic field. Although we

will not go into it here, Nersessian mounts a sustained argument for the generativity of the models in Maxwell's derivation in her work.

Insert Figure 5 about here

In developing the computational model of visual analogy, the next issue we was that of how Maxwell got the idea to put in the idle wheel particles. We hypothesize that Maxwell used a visual analogy drawn from another model in memory to obtain the notion of the idle wheels and then transfer the notion to the vortex model. Maxwell noted that in machine mechanics such problems are solved with "idle wheels" (1855-6). But gear systems and continuum mechanical systems, such as the vortex fluid model, are quite different. We hypothesize that understanding the cross-sectional model of the vortices generically as "spinning wheels" enabled Maxwell to retrieve his knowledge of gear systems which in turn enabled him to generate the abstraction of "dynamically smooth connectors" and instantiate it as "idle wheels" between vortices in the model.

Insert Figure 6 about here

Figure 6 summarizes Nersessian's analysis of the process through which Maxwell created the analogy. The vortices in the initial vortex model were abstracted into generic spinning wheels. Then, the abstraction was used as a probe to retrieve the gear system model, which was perceptually similar to it. This model contained the notion of fly wheels acting between gears to keep them moving. The fly-wheel mechanism was abstracted to the generic notion of dynamically smooth connector. From there it was specified into the idle wheel particles in the new vortex idle-wheel model. Galatea models the transfer of the solution from the generic abstraction to the vortex/ vortex idle wheel model.

We have reason to think that Maxwell used visual reasoning in this episode because he used visual language, drew visual representations, and explicitly discussed the analogy when describing the system, e.g., "We have obtained a point of view from which we may regard the relation of an electric current to its lines of force as analogous to the relation of a toothed wheel or rack to wheels which it drives." (Maxwell, 1861-62, p472).

For Galatea to solve the problem we need to represent the generic spinning wheels model and the vortex problem in such a way that Galatea can analogically transfer the solution. To do this requires expanding the vocabulary of privels and privits. Figure 7 is a diagram of the input to Galatea for the Maxwell case. S1, S2, and S3 refer to simages for each series. The new privel **circle** represents the generic spinning wheels pictured in the cross section of the vortices (Figure 4). The idle wheels are represented as circles in Galatea. In his drawing of the solution Maxwell exaggerated the deviation of the vortices from circles, rendering them as hexagonal cross sections in order to emphasize the packing of the idle wheel particles between them. However, in the mathematical analysis he treated the vortices as rigid pseudo spheres and a generic cross section of these would be

approximately circular, as we have drawn. We used **circle** because of this and because it was already a part of the ontology of Galatea. There is an analogy between the two first simages. Mapping is enabled by the visual abstraction: Even though spinning vortices will not always look like circles, as discussed above, generically they approximate circles, facilitating the analogy.

Since the idle wheel particles are added to the model, we introduced the privit **add-component**. As shown in Figure 7, the first two simages in the spinning wheels generic model are connected with an **add-component** privit, which adds the dynamically smooth connector, which looks like a small **circle**. Its exact location is unspecified, since you can have an attribution of what is in an image without knowing exactly where it is (different what/where pathways in the brain.) To complete the problem solution a new transformation **put-between**, which places an object between two other objects, is introduced. This procedure is split into two privits because they are more primitive. Both the Fortress and Maxwell cases have been fully implemented.

Insert Figure 7 about here

Other Systems

Galatea uses visual analogies for solving new problems. Other computational models of visual analogy have focused on different tasks.

Galatea does not generate the analogical mapping, but other systems that create mappings with visual information, and, thus, show that it can be done. The VAMP systems are analogical mappers (Thagard et al., 1992). 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 (Holyoak and Thagard 1997), a constraint satisfaction connectionist network. The radiation problem mapping was one of the examples to which VAMP.2 was applied. Croft and Thagard (2000) created a computational model DIVA which does analogical mapping using ACME. What it maps are three-dimensional representations in hierarchically organized scene graphs. Things in the graph can be associated with behaviors, so it can represent dynamic systems.

The Structure Mapping Engine, or SME (Falkenhainer et al. 1990) which finds the best mapping of elements between two domains has been applied to visual knowledge in a system called MAGI (Ferguson and Forbus 2000). This system takes visual representations and uses SME to find examples of symmetry and repetition in a single image.

Like Galatea, MAGI, DIVA, and the VAMPs use visual knowledge. But unlike Galatea their focus is on creating the mapping rather than on transfer of a solution procedure.

MAGI's and Galatea's theories are compatible: a MAGI-like system might be used to create the mappings that Galatea uses to transfer knowledge. The theory behind the VAMPs is incompatible because they use a different level of representation for the images.

LetterSpirit is the only other system that does a limited kind of problem solving (McGraw and Hofstadter 1993). It is primarily a model of analogical transfer. LetterSpirit 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. Like Galatea, the analogies 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. A mapping is created for the input character. For example, the seed letter may be interpreted as an "f" with the crossbar suppressed. When the system makes a lower-case "t," by analogy, it suppresses the crossbar.

However, unlike Galatea, it is not at all clear that LetterSpirit is applicable to other domains (such as the fortress/tumor problem), in part because there is little distinction between its theory and the implementation that works for letters. In contrast, one can see how Galatea might be applied to the font domain: The stylistic guidelines in LetterSpirit, such as "crossbar suppressed" would be a visual transformation in Galatea: a transformation of removing an element from the image, where that element was the crossbar and the image was a prototype letter "f." This transformation could be applied to the other letters one by one. We conjecture that our theory, when fully developed, will have more generality than LetterSpirit. It has already been applied to two different domains, and the nature of the privels are sufficiently abstract so as to allow for the representation of more examples.

Conclusions

We have described a preliminary computational model of visual analogy in problem solving. The model is consistent with our understanding of the available psychological data on the Duncker radiation problem and is instantiated in Galatea, an operational computer program. We have also described how adding primitives to the knowledge representation language, Privlan, allows an account of a small portion of a cognitive-historical analysis of Maxwell's reasoning about electromagnetism.

Galatea and Privlan constitute a research theme for analyzing and modeling visual analogy in problem solving. The privels (primitive visual elements) of Privlan provide a vocabulary of symbolic abstractions of visual information. Visual analogs are represented as a sequence of symbolic images, where the knowledge content of each image is expressed in the language of privels. Privlan's privits (primitive visual transformations) similarly provide a taxonomy of generic visual transformations. Transitions between symbolic images are expressed in the language of privits. Galatea shows that this representation and organization of visual information enables analogical transfer of problem solving strategies from a known analog to new problems. Further, the application of Galatea to account for

Maxwell's case study shows that it is possible to incrementally augment Privilan to accommodate new domains without necessarily revising Galatea's computational process of analogical transfer.

References

Bhatta, S. R. and Goel, A. K., 1997a, A functional theory of design patterns. In the Proceedings of the International Joint Conference for Artificial Intelligence. pp. 294—300.

Bhatta, S. and Goel, A. K., 1997b, Learning Generic Mechanisms for Innovative Design Adaptation. *Journal of Learning Sciences*, 6(4):367—394.

Boden, M., 1990, *The Creative Mind: Myths and Mechanisms*. Basic Books: London.

Chandrasekaran, B., Goel, A. and Iwasaki, Y., 1993, Functional Representation as a Basis for Design Rationale. *IEEE Computer*, 26(1):48-56.

Clement, J., 1994a, Imagistic simulation and physical intuition in expert problem solving. *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum, Hillsdale, NJ. pp. 201—206.

Clement, J., 1994b, Use of physical intuition and imagistic simulation in expert problem solving. In: *Implicit and explicit knowledge*. Tirosh, D., Ed., Norwood, NJ: Ablex Publishing Corp.

Croft, D. and Thagard, P., 2000, Dynamic imagery: A computational model of motion and visual analogy. Unpublished manuscript.

Davies, J., and Goel, A. K., 2001, Visual analogy in problem solving. *Proceedings of the International Joint Conference on Artificial Intelligence 2001*.

Duncker, K., 1926, A qualitative (experimental and theoretical) study of productive thinking (solving of comprehensible problems), *Journal of Genetic Psychology*, 33:642—708.

Do, E. Y., and Gross, M. D., 1995a, Drawing analogies: Supporting creative architectural design with visual references. In: *3rd International Conference on Computational Model of Creative Design (HI '95)*, Do, E., Gross, M. D. and Gero, J., eds., pp. 37-58, Australia.

Do, E. Y., and Gross, M. D., 1995b, Drawing analogies: Finding visual references by sketching. In: *Proceedings of the Association for Computer Aided Design in Architecture*

(ACADIA 95), 1995 National Conference for Computing in Design, Kalisperis, L. and Kolarevic, B., eds., Seattle, pp. 35—52.

Falkenhainer, B., Forbus, K. D., and Gentner, D., 1990, The Structure mapping engine: algorithm and examples. *Artificial Intelligence* (41):1—63.

Ferguson, R., and Forbus, K. D., 2000, GeoRep: A flexible tool for spatial representation of line drawings. *Proceedings of the Seventeenth National Conference on Artificial Intelligence*. AAAI Press/MIT Press. Cambridge, MA. pp. 510--516.

Glasgow, J. and Papadias, D., 1998, Computational imagery. In Thagard, P. *Mind Readings*. Cambridge, MA: MIT Press. pp. 157—205.

Goel, A. K., 1991a, Model Revision: A Theory of Incremental Model Learning. *Proc. Eighth International Conference on Machine Learning (ICML-91)*, Chicago, June 1991, Los Altos, CA: Morgan Kaufmann, pp. 605—609.

Goel, A. K., 1991b, A Model-Based Approach to Case Adaptation. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, Chicago, August 1991, Hillsdale, NJ: Lawrence Erlbaum, pp. 143—148.

Goel, A., Bhatta, S. and Stroulia, E., 1997, Kritik: An early case-based design system. In: *Issues and Applications of Case-Based Design*. Maher, M. L. and Perl, P., eds., Erlbaum, Hillsdale, NJ., pp. 87—132.

Goldschmidt, G., 1995, Visual displays for design: Imagery, analogy and databases of visual images. In: *Visual Databases in Architecture: Recent advances in design and decision making*, Koutamanis, A., Timmermans, H. and Vermeulen, I., Eds, Avebury: Brookfield. pp. 53—76.

Gooding, D. C., 1994, *Experiment and the Making of Meaning*, Dordrecht, Kluwer Academic Publishers.

Griffith, T. W., Nersessian, N. J., and Goel, A., 1996, The role of generic models in conceptual change. *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, San Diego. pp. 312—317.

Griffith, T. W., Nersessian, N. J., and Goel, A., 2000, Function-follows-form transformations in scientific problem solving. *Proceedings of the Twenty-Second Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum Associates, Mahwah, New Jersey. pp. 196—201.

Holyoak, K. J. and Thagard, P., 1997, The analogical mind. *American Psychologist*, 52(1):35—44.

Kosslyn, S. M., 1994, *Image and Brain: The Resolution of the Imagery Debate*. MIT Press, Cambridge, MA.

Maxwell, J. C., 1861-2, On physical lines of force. In *The Scientific Papers of J. C. Maxwell*, Niven, D. ed. Cambridge: Cambridge University Press, 1890. Reprinted in 1952, New York: Dover Publications Vol. 1, pp. 451-513.

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.

Nersessian, N. J., 1984, *Faraday to Einstein: Constructing Meaning in Scientific Theories*. Kluwer, Dordrecht, pp. 68-93.

Nersessian, N. J., 1992, How do scientists think? Capturing the dynamics of conceptual change in science. In: *Cognitive Models of Science*. Giere, R. N., ed., University of Minnesota Press. Minneapolis, MN, pp. 5—22.

Nersessian, N. J., 1994a, Opening the black box: Cognitive science and the history of science. In: *Constructing Knowledge in the History of Science*. *Osiris*, Thackray, A., ed., 10:194—214.

Nersessian, N. J., 1994b, Abstraction via generic modeling in concept formation in science. Georgia Institute of Technology Cognitive Science Technical Report 94/22. To appear in: *Idealization and Abstraction in Science*. Jones, M. R. and Cartwright, N., eds., Amsterdam: Editions Rodopi, in press.

Nersessian, N. J., 2001, Maxwell and ‘the method of physical analogy’: Model-based reasoning, generic abstraction, and conceptual change. In: *Reading Philosophy of Nature: Essays in the history and philosophy of science and mathematics to honor Howard Stein on his 70th Birthday*. D. Malamet, ed., LaSalle, IL: Open Court, in press.

Pedone, R., Hummel, J. E., and Holyoak, K. J., 2001. The use of diagrams in analogical problem solving. *Memory and Cognition*, 29:214—221.

Prabhakar, S. and Goel, A., 1996, Learning about novel operating environments: Designing by adaptive modelling. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, Special Issue on Machine Learning, 10:136-142.

Thagard, P., D. Gochfeld, and S. Hardy, 1992, Visual analogical mapping. *Proceedings of the 14th Annual Conference of the Cognitive Science Society*. Hillsdale, Erlbaum, pp.522—527.

Table 1: Some privels in Galatea	
Privel name	Attributes
Generic-visual-element	Location, size
Line	Location, thickness, start-point, end-point
Circle	Location, size
Box	Location, height, width, orientation

Figure 1: The things outside the shaded box are given to Galatea: a complete source problem, an incomplete target problem, and the analogy between them. Galatea completes the analogical transfer and stores the new simage sequence for the target problem.

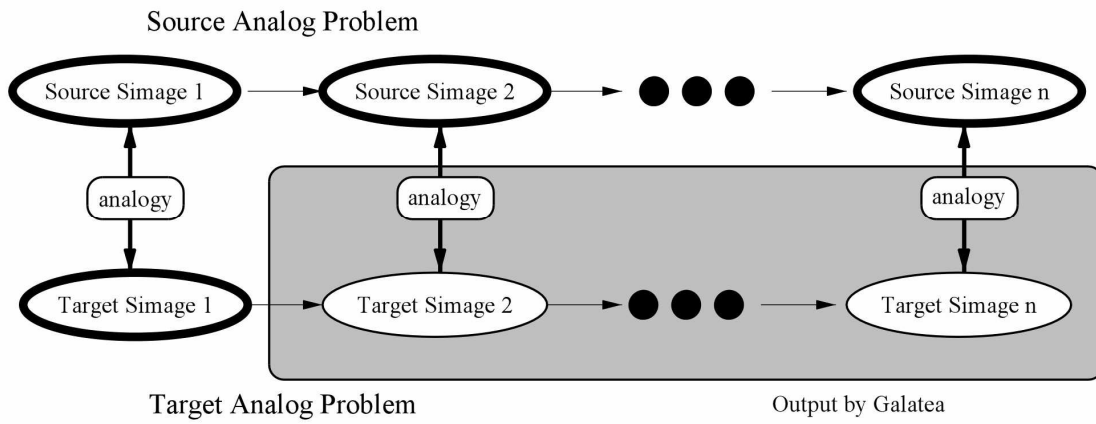
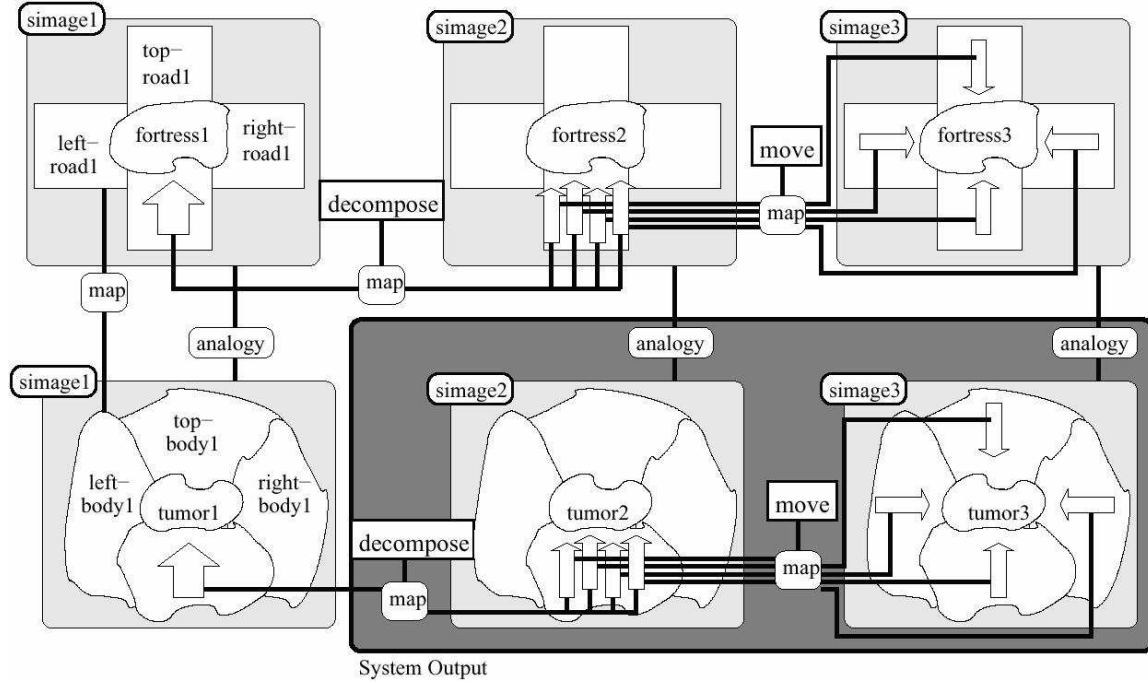


Figure 2: The fortress/tumor problem representatio



n

Figure 3: Several vortices packed together.

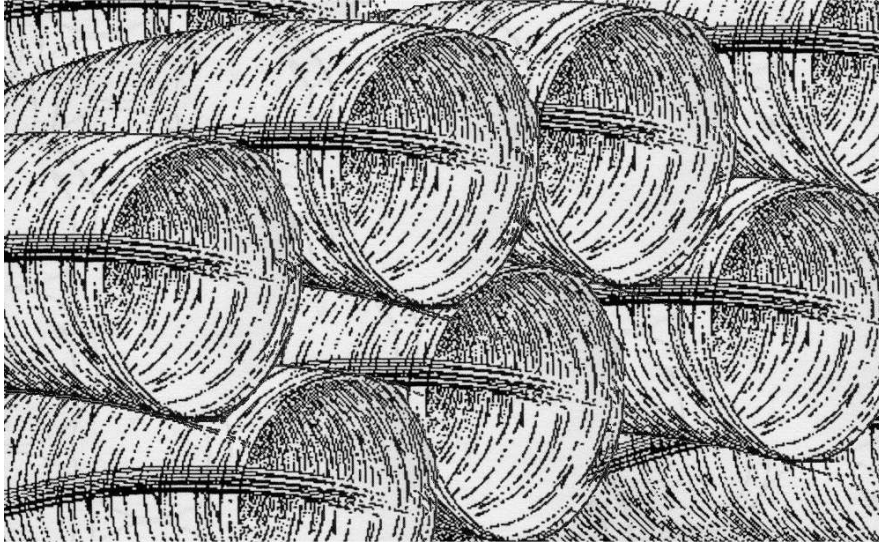


Figure 4: Cross-section of vortices.

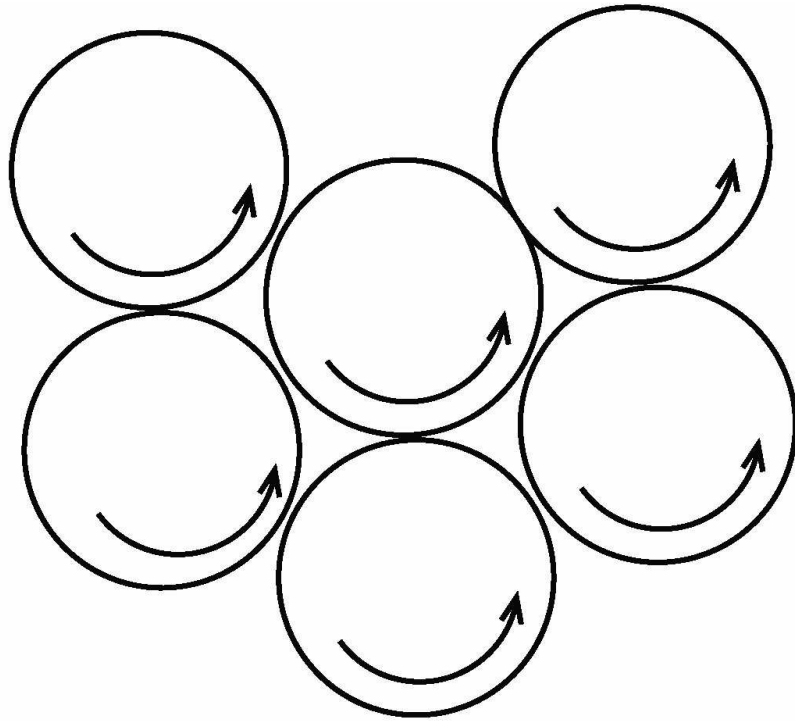


Figure 5: Maxwell's drawing of the wheels in the vortices (p489, Maxwell, 1861-2).

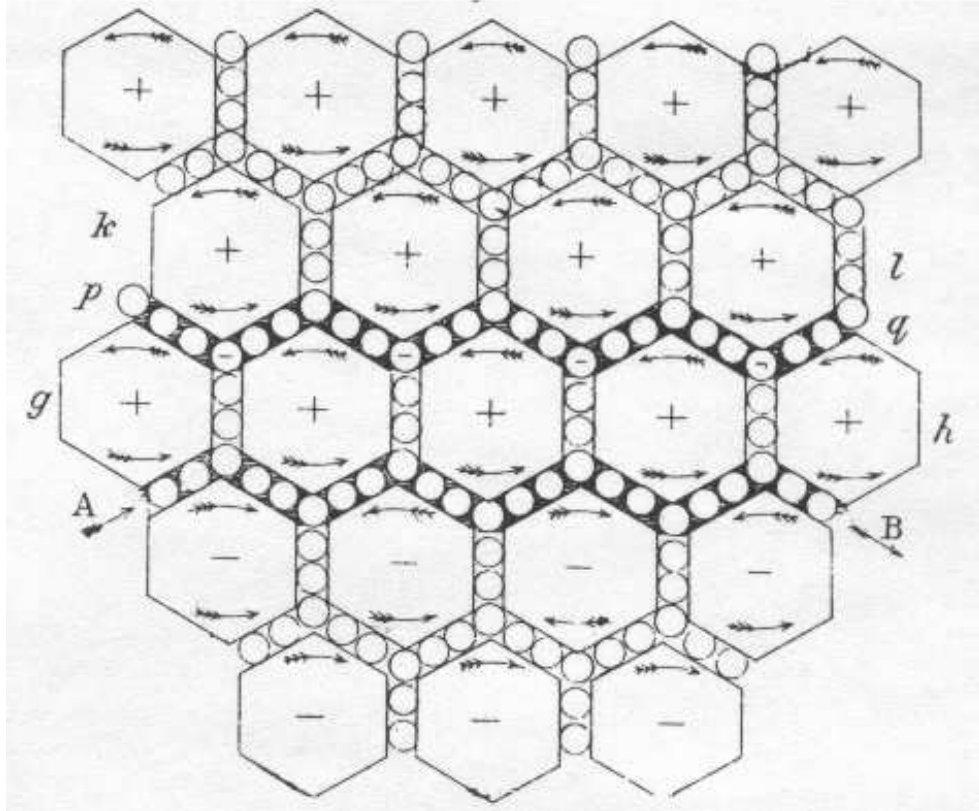


Figure 6: The analysis of how Maxwell transferred the idea of the dynamically smooth connectors from the gear system model to the vortex idle wheel model through the use of a generic abstraction.

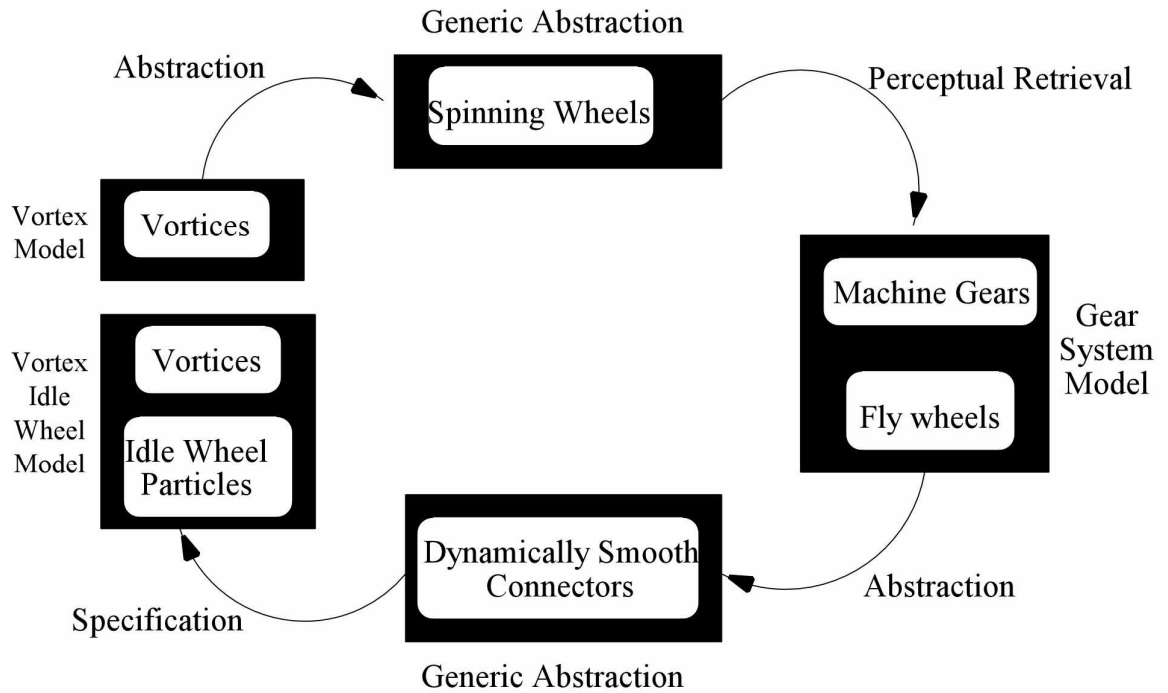


Figure 7: The source and target analogs for our implementation of the Maxwell case. The top simage series represents the source analog. The shaded area represents the output of Galatea as a result of the analogical transfer.

