UC Merced
Proceedings of the Annual Meeting of the Cognitive Science Society

Title
Function-Follows-Form Transformation in Scientific Problem Solving

Permalink
https://escholarship.org/uc/item/5hn346c1

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 22(22)

Authors
Griffith, Todd W.
Nersessian, Nancy J.
Goel, Ashok

Publication Date
2000

Peer reviewed

eScholarship.org
Powered by the California Digital Library
University of California

Function-Follows-Form Transformations in Scientific Problem Solving

Todd W. Griffith
Computer Science Department
Bucknell University
Lewisburg, PA 17837
(570) 577-3721
tgriffth@bucknell.edu

Nancy J. Nersessian and **Ashok Goel**
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0280
(404) 894-5612
{nancyn,goel}@cc.gatech.edu

Abstract

This paper presents a pattern of reasoning called "function-follows-form" (FFF) uncovered through a study of scientific problem solving. In the study we modeled eleven "think-out-loud" problem-solving protocols taken by John Clement (1989). Our work involved computationally modeling the reasoning processes of eleven scientists each attempting to solve the same problem about springs. We describe experiments with two computational systems, ToRQUE and ToRQUE2, which were used to model salient reasoning found in the protocols, and we show how the use of function-follows-form reasoning patterns enables exploration and conceptual change.

Introduction

Our research identifies and elucidates a pattern of reasoning we call function-follows-form (FFF) reasoning. We have shown that this pattern of reasoning plays an important role in exploratory problem solving, and may lead to significant change to a subject's mental models. Here we present specification of FFF resulting from experiments with two successive computational systems called ToRQUE and ToRQUE2. The study involved modeling the problem solving of eleven scientists each attempting to solve the same problem about springs. We used "think-out-loud" protocols collected by John Clement (1989) and performed experiments testing the fidelity of our computational model with the protocols.

This research represents a melding of disciplines with the goal of understanding complex scientific problem solving. We have combined techniques from history and philosophy of science, cognitive psychology, and artificial intelligence to study the problem solving of scientists. The focus of our effort discussed here was to capture the salient aspects of problem solving for each of the scientists in the form of a general competence model, encoded in a computational system (i.e. ToRQUE2).

Background

As a first attempt at developing an interpretation of scientific problem solving Nersessian and Greeno (1992) examined an extensive expert problem-solving protocol obtained in a "think-out-loud" interview conducted by John Clement (1989). In particular, they were interested in the second protocol (S2), because it exhibited many of the characteristics of James Clerk Maxwell's problem-solving practices in the construction of the electromagentic field concept. As they interpret this protocol, the subject uses what they call "constructive modeling" to satisfy himself that his initial answer to a problem was the correct answer. They saw this process as primarily one of arriving at a model that is of the same kind with respect to the salient features of the spring problem. They argue that while this example is much more constrained than historical cases of scientific discoveries, it is still complex enough to require dealing with the many quite difficult modeling issues historical discoveries present.

Clement's own analysis of S2's reasoning focuses on a process he calls modeling via "bridging analogies". He characterizes this process as one in which the subject "produces models via a successive refinement process of hypothesis generation, evaluation, and modification or rejection" (p.358, Clement 1989). It is the specific nature of the construction and "successive refinement" process that led Nersessian & Greeno to interpret S2's reasoning as a form of constructive modeling, and subsequently led to our computational theory of generative modeling (Griffith *et al* 1996, Griffith *et al* 1997, Griffith 1999).

The Problem

According to Clement, S2 was a computer scientist who had some training in physics. He had also passed comprehensive examinations in mathematics in the area of topology, which is highly significant to our interpretation of the protocol session.

In the protocol, S2 is asked to solve the following problem about springs:

> "... a weight is hung from a spring. The original spring is replaced with a spring made of the same kind of wire; with the same number of coils; but with coils that are twice as wide in diameter. Will the spring stretch form its natural length more, less, or the same amount under the same weight? (Assume the mass of the spring is negligible compared to the mass of the weight.) Why do you think so?"

In our interpretation, S2 began the problem-solving session with an intuitive understanding that the stretch of a spring is due to its flexibility. Then he derived a new understanding that a spring maintains constant slope when stretched through torsion in the spring's wire. So, although this is a more modest outcome of scientific reasoning than evidenced in historical cases, for S2 it was an instance of highly creative problem solving leading to conceptual change. To find a satisfactory explanatory model for the problem solution, S2 had to generate a novel representation of how a spring works. He did so by generating a series of successive models through what we call FFF transformations.

Methodology

This research focuses on investigating the reasoning processes found in all eleven protocols in order to place S2's creative problem-solving in a context. In so doing, we highlight the reasons that lead to his discovery of torsion as a central causal element in the function of a spring. The additional protocols show scientists attempting to solve the spring problem. All the scientists were expert problem solvers, though none were experts in the domain. The protocols were modeled in two sets. The first set of five protocols (S1-S5) was used to build and refine the ToRQUE system. The second six protocols (S6-S11) were used to evaluate the refined ToRQUE2 system. Experiments were conducted at each stage of development in order to evaluate hypotheses with respect to the methods and knowledge used by the subjects. The first set of experiments, used to refine the systems performance with respect to the first five protocols, involved the ablation and reconfiguration of tasks, methods, and knowledge in order to determine what aspects of the system enabled accurate modeling of the first five subjects. The second set of experiments were also ablation and reconfiguration experiments. For these experiments the system was left unchanged but was "reconfigured" to account for each of the remaining six subjects. This means that reasoning elements such as tasks, methods, or knowledge structures were removed or reordered but not added, and that no reimplementation was done on the ToRQUE2 system during the testing phase. Both sets of experiments looked at the choice of knowledge structures and reasoning methods used by the system, as well as the ordering and availability of knowledge and methods. The system was evaluated based on its ability to accurately model the salient reasoning of subjects.

Ontologies for Function-Follows-Form

Function-Follows-Form transformations are based upon a series of ontological commitments with respect to the control of processing and the representation of knowledge, each of which is based upon past computational results. The language for the control of processing is called the task, method, knowledge language which is based upon a TMK architecture, while the language for representing physical systems is called the structure, behavior, function (SBF) language which was first developed as part of the theory of adaptive modeling. A reasoning packet comprises patterns from each of these languages.

The TMK Language: The Task, Method, Knowledge (TMK) architecture is a theory of control of processing that was first developed by Goel & Chandrasekaran (1992) in an analysis of the methods used for addressing complex tasks. This work was continued in (Goel *et al* 1994, Punch, Goel, & Brown 1995, Goel *et al* 1996). The theory posits that high-level tasks such as conceptual design can be broken down into a hierarchy of methods and subtasks. Each task or subtask may have one or more methods that can be applied to solve the task. It also posits that each method specifies the sub-tasks that it spawns and control information for the ordering of those sub-tasks. Using multiple methods enables the architecture to account for a variety of reasoning strategies for addressing any one task, where a *strategy* is some sub-hierarchy of the task-method tree whose root is a method.

One advantage of the TMK architecture is that knowledge can have a direct effect on which method is selected to accomplish a particular task. For the purpose of modeling multiple subjects this feature is particularly important. In general each subject has different initial knowledge conditions. This means that one wants the system to be able to select different methods based on that knowledge in order to exhibit different reasoning traces. The TMK architecture allows for this kind of variation. The goal from a modeling perspective is to correctly specify the knowledge structures, reasoning strategies, and ordering of strategies, such that for any initial knowledge condition the TMK model is able to accurately account for the reasoning.

The SBF Language: As an initial attempt to address the issues from the Maxwell case and the Clement protocols, we attempted to model the Clement protocols using a computational theory of device design called "adaptive modeling" (Goel 1991b, 1996). This attempt led to the development of new design considerations and ultimately to a new computational theory of scientific problem solving. The theory of "adaptive modeling" takes its name from the perspective it adopts on conceptual device design. Conceptual design generally refers to the preliminary phase of the design process. The problem-solving task in this phase takes a specification of the functions of the desired device as input. It has the goal of giving a high-level specification of a structure for the device as output, where the structure can deliver the desired functions.

Kritik and IDeAL are operational knowledge systems that instantiate the theory of adaptive modeling, enable experiments with it, and provide well-defined AI languages. Built in the late eighties, Kritik integrated case-based and model-based reasoning for modeling evolutionary design of simple physical devices (Goel 1989, 1991a, 1992; Goel & Chandrasekaran 1989, 1992). The specific hypothesis in the Kritik experiments was that since the design task is a function → structure mapping, the inverse structure → function map of old designs may guide the adaptation of an old design to achieve a new functional specification. The structure → function map of a device design in Kritik is specified as a Structure → Behavior → Function model. In an SBF model of a device the behavior mediates between function and structure: it captures teleological and compositional knowledge of a device, and provides a functional and causal explanation of the how the structure of the device delivers its functions.

The IDeAL system builds upon the Kritik system in several significant ways. Perhaps the most significant contribution of the IDeAL system is the addition of a theory for cross-domain analogy called model-based analogy (MBA) (Bhatta 1995, Bhatta & Goel 1993, Bhatta & Goel 1997). This theory enables the system to apply abstract information that it learns in one domain such as that of electric circuits to another domain such as that of heat exchangers.

Kritik and IDeAL both focus on the task of conceptual design. In design the goal is the description of some artifact that serves a particular purpose, i.e., it has some desired function. For this reason the Kritik and IDeAL systems focus on functionally driven transformation processes. The task in scientific discovery is often one in which changes to the function are only realized after a structural change has taken place. The ToRQUE systems make use of this kind of transformation – form-based or "function-follows-form" transformations (see Griffith *et al* 1997, 1999).

In this research we have identified a series of transformational knowledge patterns that can be used to accomplish form-based transformations. We have called these patterns generic structural transformations (GSTs) because they are generic with respect to the models to which they may be applied and because they are first applied to the structure of the model and then propagated to the behavior. We have described two strategies for carrying out form-based transformations. The first is called Structure-Based Model Transformation (SBMT) and the second is called Limiting Case Analysis (LCA).

Function-Follows-Form Reasoning Packets

One important task in artificial intelligence is identifying patterns of reasoning that are generic to a variety of problems. In this research we have identified several reasoning patterns using the TMK language. These reasoning patterns are packets of tasks, methods, and knowledge that frequently appear together. The most promising of these TMK reasoning packets is the FFF packet, which appears to be a general process used by expert reasoners to solve exploratory problems.

One important issue in both the historical and protocol studies is to find the function of a particular physical system given its form. For example, in the S2 case the task is to find the amount the spring will stretch given the diameter. Thus far we have developed a computational system, TORQUE2, that models S2's discovery of torque in springs. A key computational characteristic of TORQUE2 is its application of structural transformations to the structural and topological elements of SBF models to generate new models. To achieve FFF transformations TORQUE2 uses the GST knowledge structures. After retrieving an initial source analog via model-based analogy, TORQUE2 evaluates the model by attempting to reduce the differences between the target and the analog model. This evaluation process involves retrieving generic models of physical principles (GPPs) which can explain away the differences or applying GSTs to transform the target or source models. These adaptations bring new knowledge to the task that may lead TORQUE2 towards or away from the initial goal. TORQUE2 discovers the GPP of torque while attempting to reduce the differences between a circular and an imaginary square coil.

> *S2 protocol: line 121:* Now that's interesting…Just looking at this it occurs to me that when force is applied here [end segment], you not only get a bend on this segment, but because there's a pivot here [referring to a connection in the hexagonal coil], you get a torsion effect.. around there..[a center segment]

Through TORQUE2 we have established that in the S2 case "function-follows-form" transformations play a significant role in the exploratory process. We hypothesize that "function-follows-form" transformations also play a significant role in Maxwell's exploration of electromagnetism.

In the following sections we present the function-follows-form reasoning packet by showing the task pattern, method pattern, and knowledge patterns that are used to carry out the reasoning, which taken together form a reasoning packet.

Function-Follows-Form Task Pattern

In (Griffith 1999) we hypothesize that the ordering of high-level reasoning strategies proceeds from a strategy of model-based search through a process of analogy and, failing that, to processes of transformation. We are also hypothesizing that the FFF reasoning packet is used only under certain conditions. The conditions under which a method takes place is a part of its task pattern. The task pattern for FFF can be defined formally with respect to the models in memory, the target model, problem description. and the solution.

The formal task pattern for FFF is: given (1) a target model that is an element of a set of models available to the agent, (2) some problem with respect to that target model, and (3) that no solution can be generated using a search method or an analogy method, return a new model that contains a solution to the problem, such that using analogical transfer from this new model back to the target model provides a solution to the problem. The task pattern defines the problem to consist of input (the problem and the target model) and output (the solution to the problem). It also defines the situation in which the task is performed – in this case, after attempting model-based search and model-based analogy. We see this task pattern in several of the subjects including S2, S6, and S8. We also see this pattern in Maxwell's reasoning.

Function-Follows-Form Method Pattern

The method pattern for the FFF transformation is found in the SBMT hierachy. The method pattern in a TMK reasoning packet includes the hierarchy of subtasks that the method spawns, the ordering of these subtasks, and the knowledge that the method acquires during its processing. These aspects of the method are shown in Figure 1. The dashed lines indicate the subtasks that are spawned and the solid black lines indicate that a method or procedure is selected. The gray lines show the ordering of subtasks. The rectangles are subtasks in the method hierarchy. The single-line ellipses are methods, and the double-line ellipses are procedures. Memory is indicated by the star or seal figure and the plus sign is a specific piece of knowledge that gets retrieved from memory.

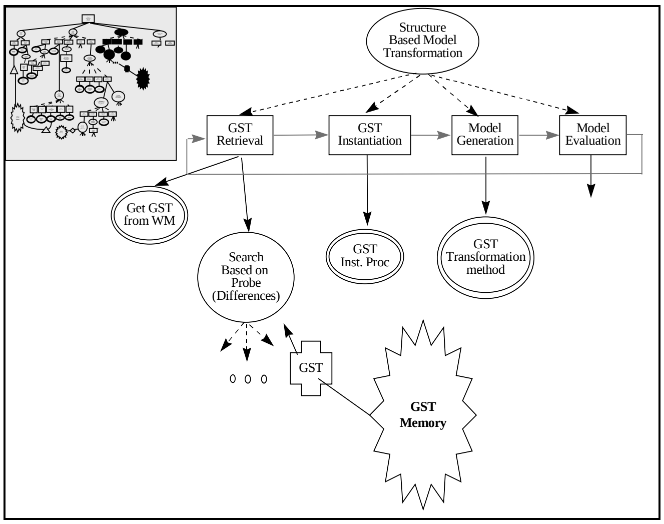

Figure 1: Function-Follows-Form Method Pattern

The pattern shows that SBMT is a memory-based process in which the retrieval of particular GSTs occurs when no GSTs are available in working memory. The process then instantiates the retrieved GST for the particular problem-solving situation, and attempts to generate a new model by applying the GST to the model. The process ends with an evaluation of the model that could result in the recognition of GPPs or in recursive application of the SBMT method.

Function-Follows-Form Knowledge Patterns

The knowledge patterns for the FFF reasoning packet include SBF models as well as GSTs. GSTs are the active knowledge element in FFF. GSTs contain indexing information that allows them to be retrieved based on differences between analog models and information that indicates when they can be applied to a model. Most importantly they contain the processing information for transforming the structure (including geometry and topology) as well as the behavior of SBF models.

In Figure 2 we see one application of the function-follows-form reasoning packet. This reasoning packet shows how the 3D-to-2D GST is applied to the spring model. First, the geometry of the spring is transformed, which results in changes to the structure. The changes lead to behavior changes in the spring. Each new component's behavior is consolidated to form a new behavioral pattern.

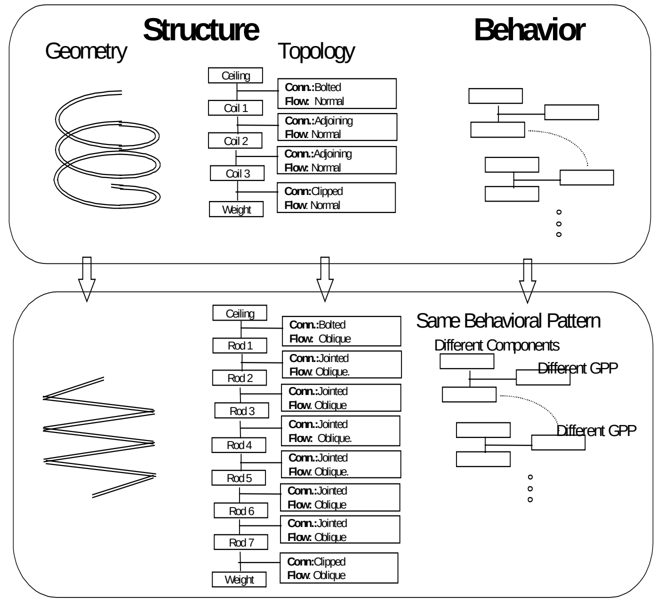

Figure 2: Function-Follows-Form in S2 Protocol

In Figure 3 we see how the function-follows-form reasoning packet is applied to our Maxwell's model construction. The top figures represent one stage in Maxwell's reasoning about the electromagnetic aether. He envisioned the aether as composed of a group of fluid vortices. The rotating circles is our represention of a cross section of a set of vortices packed together in the aether. The bottom figure shows Maxwell's representation of his model of the aether after the application of a function-follows-form transformation that changes the structure by adding "idle wheel particles" to solve the problem of friction between the vortices. The structural changes in the form of the aether result in behavioral changes. It is this model of the aether that Maxwell uses to construct the equations for electromagnetic interactions.

The significant point here is that the FFF reasoning packet was first discovered with respect to the Clement protocols, and then identified as potentially significant in interpreting Maxwell's case.

The ToRQUE System

In this section we show how the computational model instantiated in ToRQUE2 captures the salient reasoning processes of subjects by presenting a walkthrough of the steps taken by the ToRQUE2 system when configured with our interpretation of S2's initial knowledge state.

The primary task of the ToRQUE2 system is to solve a problem. The problem in this situation can be characterized as finding a relationship between a structural concept (Cs) (e.g. diameter) in the model and a behavioral concept (Cb) (e.g. amount of stretch).

To achieve exploration in TMK requires a working memory of target models (WMT), analogs (WMA), and GSTs (WMGST). As an agent addresses its task they may come to a point where they do not know how to proceed. Past reasoning stored in working memory allows the agent to pick a GST that is related to the reasoning at hand or to reasoning that has occurred recently. This serves to constrain the randomness of the selection of a GST. In ToRQUE2 working memory is captured in a data structure which has a last-in-first-out (LIFO) structure. Figure 4 shows two snapshots of working memory structures. The snapshot labeled (A) shows the WM during the first model-based analogy process prior to attempting any transformations. Snapshot B shows what transformations are placed on the structure when the circular coil becomes the target model. The transformation structure between A and B is the transformations performed between these snapshots. All the transformations that are retrieved are ordered and placed onto this WMGST structure. Thus one can think of this structure as using the last transformation which the agent was thinking about but did not apply. Not all transformations can be used on all models so many transformations may be rejected prior to being applied, e.g., a circle-to-square transformation is only possible if the target model is circular. Also, previously explored target models are removed from the structure such as when a coil retrieves a spring as an analog.

The exploration process proceeds through the interaction of Model-Based Analogy (MBA) and Structure-Based Model Transformation (SBMT) with the working memory structures, WMA and WMGST. MBA retrieves a set of analog models to solve the particular problem one of which is selected and the rest of which are placed in WMA. The answer that is produced from these analogs is evaluated by attempting to reduce the differences between it and the target model. One method of reducing these differences is to apply SBMT to the source or target analogs. Similarly GSTs are indexed and retrieved by these differences and one GST is applied while the remaining are placed in the

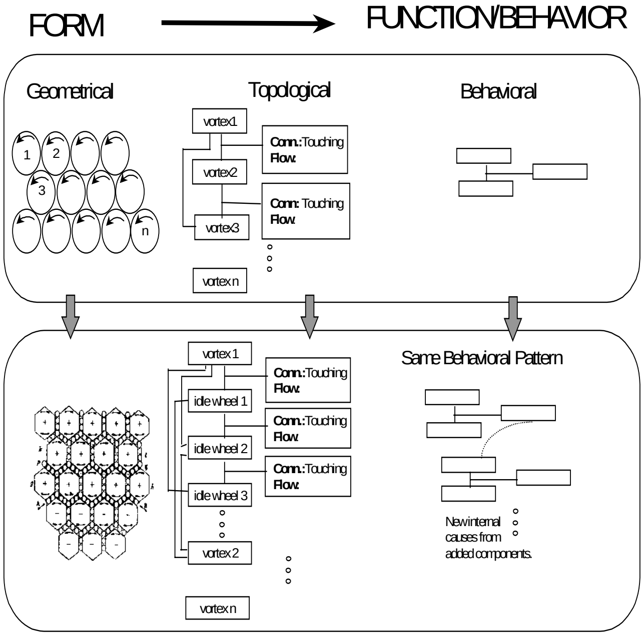

Figure 3: Hypothesized Function-Follows-Form in Maxwell

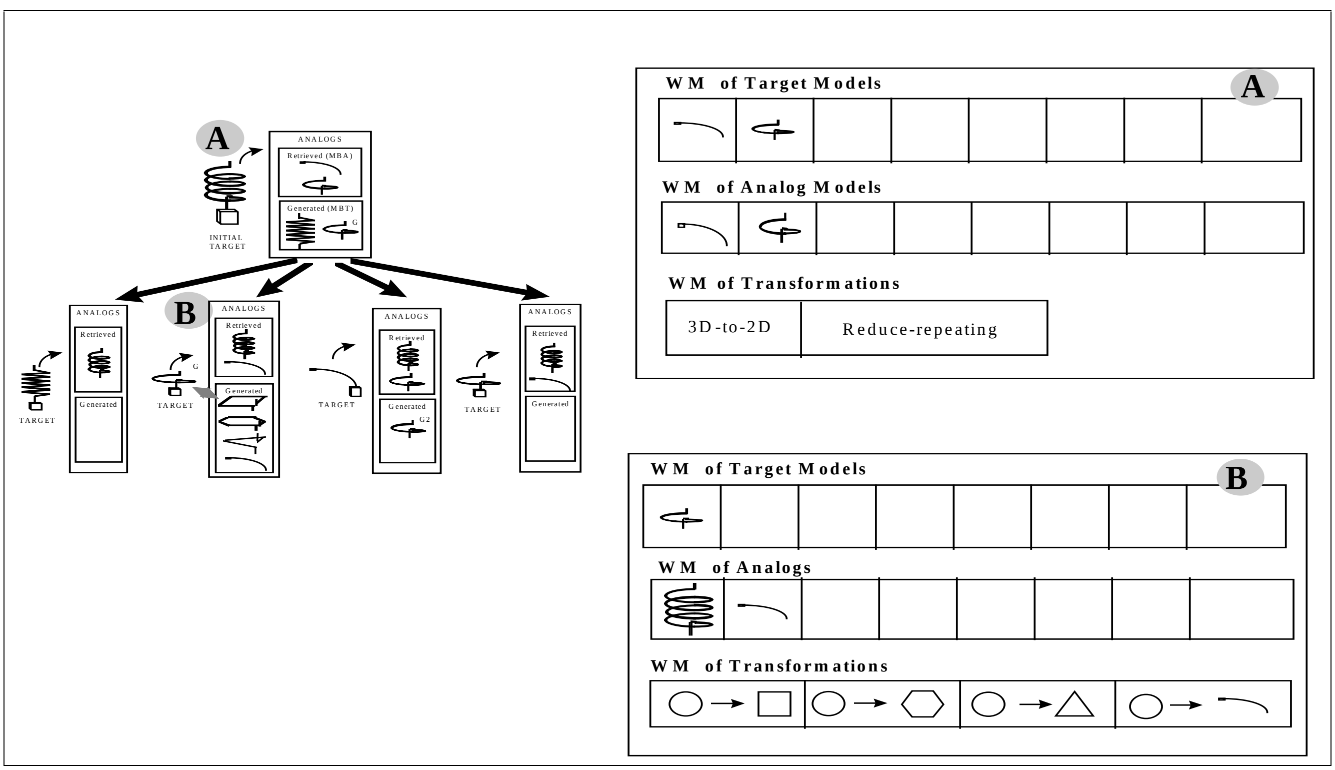

Figure 4: Snapshot of Working Memory Structures at Two Places in the Program State

WMGST structure. As reasoning progresses, a collection of transformations are placed into WM. In this way WM is not being used as a repository for knowledge that is currently being addressed, but as a repository for knowledge which has been retrieved but which has not been considered.

The left portion of Figure 4 depicts the models that are initially retrieved using the spring model and the models that are generated after transforming the initial target model in various ways. After the system retrieves an analog model it then evaluates that model by attempting to reduce the differences between itself and the analog model. These differences are used as indexes into a memory of generic structural transformations (GSTs). The SBMT process then applies the retrieved transformations to the target model to generate additional models. Notice that the models that are generated may be similar to retrieved analog models. These models, however, are not identical and so we have marked the generated coil model with a G. The figure shows the models that are retrieved as analogs for the spring model. These models were retrieved as functional analogs to the spring because they each supply a restorative force. Generic models such as GPPs are knowledge abstractions that can reduce the differences between two models by recognizing that the features of the analog model are also present in the target model.

One significant outcome of the ToRQUE2 experiments is that ToRQUE2 is able to model the competences exhibited by the test subjects (S6-S11) to a surprising degree of accuracy without changing anything except for the starting knowledge conditions. This means that the system could model the test subjects:

- without additional knowledge structures,
- without additional reasoning strategies,
- without altering the control architecture, and
- without altering the ordering of the strategies

Altering the starting knowledge conditions includes one or more of the following:

- removal of knowledge structure,
- removal of reasoning strategy, or
- removal of an index to a knowledge structure.

This means that the ToRQUE2 system is a representative instantiation of a general competence model for the spring problem. This means our model covers a representative subset of the possible knowledge and strategies one might used to solve the spring problem, such that it can account for both paths to the solution and paths to failure by the scientists. This lends support to our claim that "funtion-follows-form" transformations enable S2's conceptual change, because it is one configuration of a representative problem-solving model.

Acknowledgements

We would like to thank John Clement for generously supplying us with all of the "think-out-loud" protocols.

References

Bhatta, S.R. & Goel, A.K. (1993) Learning Generic Mechanisms from Experiences for Analogical Reasoning. In *Proc. Fifteenth Annual Conference of the Cognitive Science Society*, Boulder, Colorado, July 1993, Hillsdale, NJ: Lawrence Erlbaum.

Bhatta, S. R. & Goel, A. K. (1994) Model-Based Discovery of Physical Principles from Design Experiences. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Special Issue on Machine Learning in Design, 8(2), May 1994.

Clement, J. (1989) Learning via Model Construction and Criticism: Protocol Evidence on Sources of Creativity in Science, In *Handbook of Creativity: Assessment, Theory and Research*, Glover, G., Ronning, R., & Reynolds, C. (Eds.), chapter 20, New York, NY:

Goel, A., K. (1989) Ph.D. Dissertation, *Integrating Case-Based Reasoning and Model-Based Reasoning for Adaptive Design Problem Solving*, Department of Computer and Information Science, The Ohio State University.

Goel, A., K. (1991a) A model-based approach to case adaptation. In *Proc. of the Thirteenth Annual Conference of the Cognitive Science Society*, Chicago.

Goel, A., K. (1991b) Model revision: A theory of incremental model learning. In *Proc. of the Eighth International Conference on Machine Learning*, Chicago.

Goel, A.K. & Chandrasekaran, B. (1992) Case-Based Design: A Task Analysis. In C. Tong and D. Sriram (eds) *Artificial Intelligence Approaches to Engineering Design, Volume II: Innovative Design* (San Diego: Academic Press).

Griffith, T.W. (1999) A Computational Theory of Generative Modeling in Scientific Reasoning, Atlanta, Georgia Institute of Technology, Ph.D. Dissertation, GIT-COGSCI99.

Griffith, T.W., Nersessian, N.J., & Goel, A.K. (1996) The Role of Generic Models in Conceptual Change, *Proceedings of the Cognitive Science Society, 18* (Hillsdale, NJ: Erlbaum).

Griffith, T.W., Nersessian, N.J., Goel, A.K., and Clement, J. (1997). Exploratory Problem-Solving in Scientific Reasoning. *The Nineteenth Annual Conference of the Cognitive Science Society*, Stanford University, Lawrence Erlbaum

Nersessian, N.J. (1992) How Do Scientists Think? Capturing the Dynamics of Conceptual Change in Science, In *Cognitive Models of Science*, ed. R.N. Giere. Minneapolis, MN: University of Minnesota Press.

Nersessian, N.J. (1995) Should Physicists Preach What They Practice? Constructive Modeling in Creating Scientific Understanding, *Science & Education*, vol. 4.

Nersessian, N.J. & Greeno, J. (1992) Constructive Modeling in Scientific Discovery (unpublished manuscript).

Punch, W.F., Goel, A. K., & Brown, D.C. (1995). A Knowledge-Based Selection Mechanism for Control with Application in Design, Assembly and Planning. *International Journal of Artificial Intelligence Tools*.

Stroulia, E. & Goel, A.K. (1992) Generic Teleological Mechanisms and their Use in Case Adaptation, In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society* (Hillsdale, NJ: Lawrence Erlbaum)

Stroulia, E. & Goel, A.K. (1995) Functional Representation and Reasoning in Reflective Systems. *Journal of Applied Intelligence, Special Issue on Functional Reasoning, 9(1)*.