Model Generation in Design

Takashi Kiriyama, Tetsuo Tomiyama, Hiroyuki Yoshikawa

Department of Precision Machinery Engineering
Faculty of Engineering, The University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo 113, Japan
Internet: kiriyama@zzz.pe.u-tokyo.ac.jp
Tel: 03-3812-2111 ext. 6481
Fax: 03-3812-8849

1 Introduction

In designing a mechanical system, a designer uses various models of a design object. Each of the models represents a specific aspect of the design object, *e.g.*, aspects of dynamics, hydraulics, controlling, oscillation, strength, etc. The designer refines the design object by adding properties to the models, and evaluates behaviors by calculation using the models. At the same time, since the models represents the identical object, the designer must pay attention to the consistency of the models.

Although generation and consistency management of aspect-dependent models (aspect models) are central elements to be achieved by CAD, conventional CAD (Computer Aided Design) systems cannot support these tasks very well. Some systems can automatically generate, for instance, an FEM (Finite Element Method) model from a geometric model, but it is a fortunate case in which known algorithms work well to convert a model to another and the system does not need to know about the *deep* reason why the conversion works. Because the system lacks knowledge about physical laws to generate a different aspect model, in most cases the designer must build aspect models separately. ICAD (Intelligent CAD) systems are expected to serve as an integrated modeling environment in which aspect models are automatically generated and their consistency are maintained.

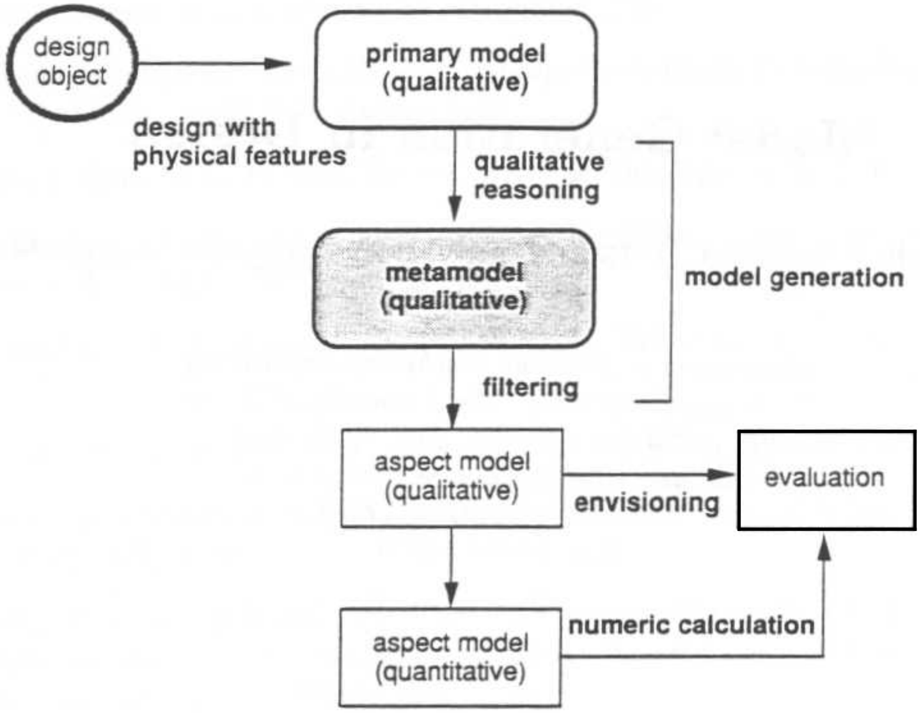

Figure 1: Aspect Model Generation

Our approach toward integrated design object modeling for ICAD is based on the use of knowledge about physical laws [7]. As depicted in Figure 1, the designer builds a *primary model* in the system using *physical features* [8] as building blocks. A physical feature is a combination of physical phenomena and structural elements that appear in mechanical design. A primary model represents the behavior and structure of the design object that the designer intends to embody. From the primary model, the system creates a *metamodel* by reasoning out all physical phenomena that can appear on the design object. The metamodel potentially includes all qualitative aspect models that the system have to deal with. Since the metamodel includes all kinds of physical phenomena, it is not suitable for the designer to understand behaviors of the design object from information that the metamodel provides. So we need aspect model generation in order to obtain behavior the designer is interested in. Aspect model generation from the metamodel is a process to filter physical phenomena and properties relevant to an aspect.

The metamodel plays a role to represent dependencies between physical phenomena and preconditions for these physical phenomena to occur in order to allow the system to maintain consistency among aspect models. When the designer changes a quantitative property of an aspect model, the change is propagated through the

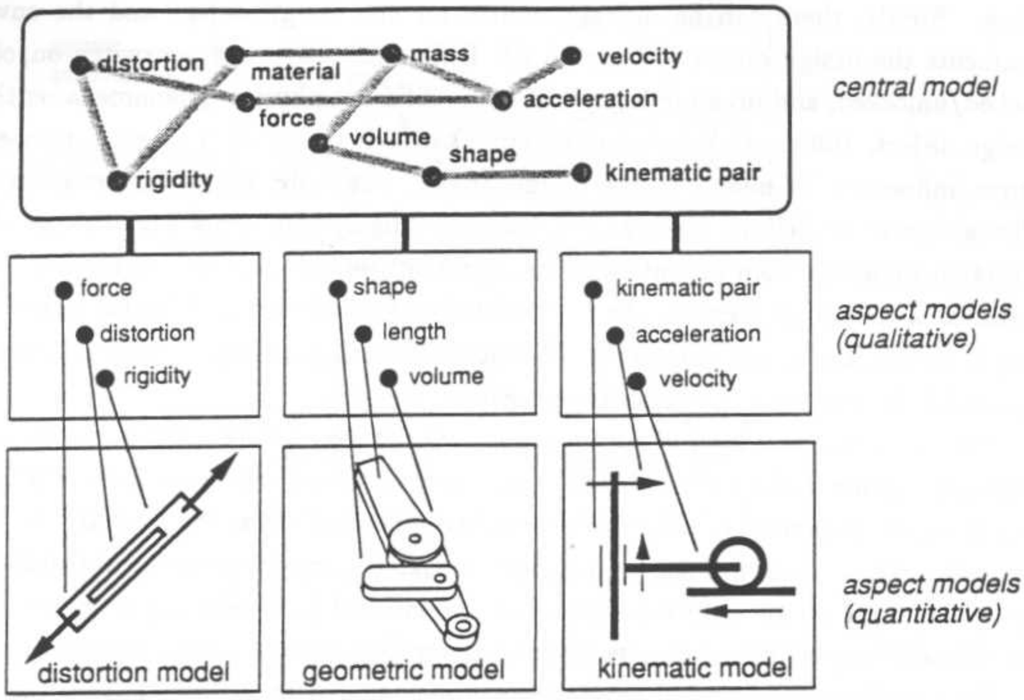

Figure 2: Consistency Management

dependency network in the metamodel to related aspect models. Figure 2 depicts the idea of dependency management among aspect models.

Refinement of the design object is prompted by the results of evaluations of aspect models. The gap between specifications and evaluations of the current design object guides the designer for further refinement. If the primary behavior needs to be changed, a new primary model is used to generate aspect models again.

In the rest of this paper, we discuss generation and consistency maintenance of aspect models based on the metamodel. In Section 2, we describe representation of the metamodel in a multiple-world environment. In Section 3, we illustrate how qualitative and quantitative aspect models are generated and used for evaluation from the central model during the course of design with an example of design of motors. Implementation of the metamodel is described in Section 4, related work is discussed in Section 5, and Section 6 concludes the paper.

2 Organizing Aspect Models in a Multi-World Environment

Since aspect models are generated from different assumptions, they are not always compatible with each other. There are two different types of assumptions for modeling. Firstly, there can be different states for the design object and the environments the design object is situated in. Different states, such as switch on/off, locked/unlocked, and pressure high/low, cause different physical phenomena on the design object. Difference between environments, such as high/low external temperature, influences differently to the design object. Secondly, there can be different abstraction in modeling. Each aspect the modeling system deals with has an abstraction mapping from an entity to its representations in the aspect model. For instance, a structural member can be modeled as a solid object, a flexible object, a fragile object, and a heat conductor. We should note that abstractions of a member as a solid object and as a flexible object cannot be mixed.

Let us call the first kind of assumptions about internal states and environments *state assumption*, and the second kind of assumptions about abstraction of the design object *aspect assumption*. The metamodel is generated from the primary model by adding these assumptions. An aspect model is derived from the metamodel by specifying a set of aspect assumptions. Relationship among aspect models is maintained by specifying consistent state assumptions for the aspect models.

We use the multiple-world environment implemented on ATMS [1] to organize aspect models under assumptions. Each physical phenomenon in the central model is labeled with a set of assumptions it depends on. Since an aspect model is a collection of physical phenomena, it is justified from a set of assumptions, *i.e.*, an environment in ATMS terminology. When the designer focuses on an aspect of the design object in a particular state, the corresponding environment is chosen in ATMS of the metamodel to justify a valid aspect model. Figure 3 depicts modeling assumptions and the metamodel as a composite of all aspect models.

A state assumption is brought into the metamodel when a physical feature having parameters over different qualitative values or different working modes is used in the primary model. For instance, a physical feature *switch* has two working mode, on and off. Such a condition is added to the assumption list of the metamodel when the metamodel is generated from the primary model. On the other hand, an aspect assumption is introduced to the metamodel when a qualitative reasoner derives physical phenomena from the primary model.

Physical phenomena that belong to a particular aspect are marked with **aspectOf** predicates in their preconditions. For instance, *electricCurrent* is marked with *aspectOfElectricity*. When an aspect model of aspect A is generated, physical phenomena that have **aspectOfA** in their preconditions are chosen out of the metamodel. For instance, when the system generates a kinematic model, it assumes the precondition **aspectOfKinematics** holds and others not. As a result, physical phenomena depending on **aspectOfKinematics** can become active when their

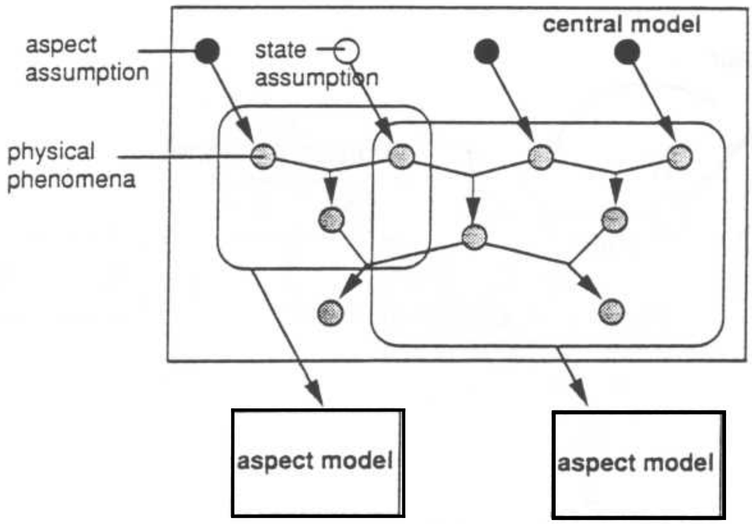

Figure 3: Assumptions for aspect models

conditions for parameters are satisfied, whereas others never become active. By choosing an **aspectOf** predicate, the system can selectively justify occurrence of physical phenomena relevant to the aspect model.

3 Design of a Motor

Let us illustrate model generation and integration with an example of design of an electromagnetic motor. A motor is basically driven by attraction and repulsion forces between a pair of magnets. The simplest structure to embody this mechanism is a single-pole motor illustrated in Figure 4. Its primary model consists of physical features attraction between different magnetic poles, a moment of force around a shaft, and rotation of a shaft by a moment of force. This motor turns half round and rests at the closest position to the permanent magnet. We can predict this behavior with a qualitative dynamic model, which is generated under an assumption that structural elements are solid objects. There is the sole operating mode of the motor, and therefore the metamodel contains no state assumption.

To make the motor continuously rotate, we have to switch polarities when the rotor aligns to the permanent magnet. Figure 5 depicts a refined single-pole motor with a commutator connected to an electromagnetic coil. Now the motor rotates continuously, but there are two problems left, *i.e.*, a problem of dynamic imbalance of force on shaft and a problem of dead points. From a qualitative dynamic model we can predict that the shaft of the motor receives a force in the direction from the rotor to the permanent magnets. If we connect quantitative analysis module to the

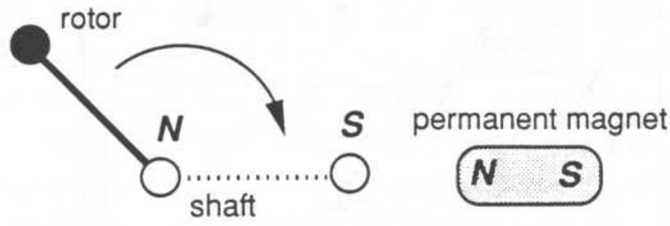

Figure 4: A single-pole motor

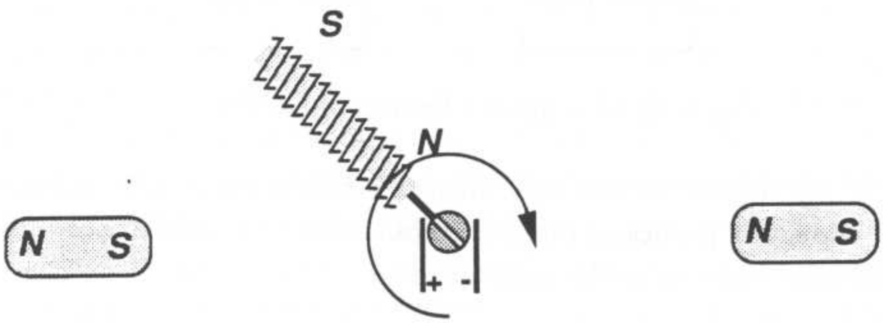

Figure 5: A single-pole motor with a commutator

qualitative dynamic model, We can numerically calculate changes of the direction and magnitude of the resultant force over the angle of revolution the shaft. From the viewpoint of dynamics, this single-pole motor turns out to be a bad design since the imbalance may shorten the life time of the mechanism. We can also qualitatively predict that the motor does not start to rotate when the rotor is aligned to the permanent magnets. Furthermore, we can find a heat problem of the electromagnetic coil using a heat model; the coil can be heated up by electric current.

To solve the first imbalance problem, we need to use a pair of poles and a pair of permanent magnets. Figure 6 shows this two-rotor motor. The qualitative aspect model about dynamics predicts that the shaft receives forces from two permanent magnets, and numeric calculation shows that they compensate each other. Thus the motor does not have the imbalance problem. But the problem of dead points is still left.

Figure 7 depicts a further improved three-pole motor. Each coil alters its magnetic pole when it passes by the permanent magnets. We can predict with a qualitative dynamic model that this motor has no dead points. But, on the other hand,

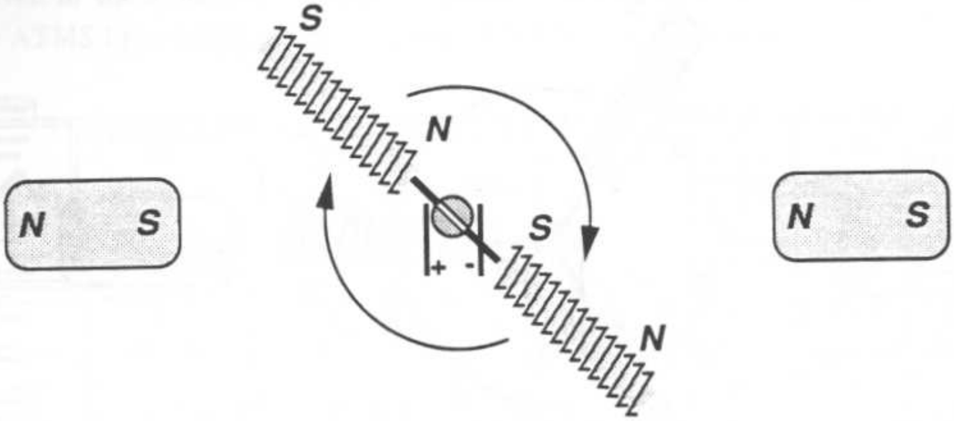

Figure 6: A two-pole motor

a quantitative aspect model shows that the resultant force is small but not equal to zero. Depending on degree of the imbalance, the designer can either use this structure as the final solution or continue to refine the structure. Yet another problem is heat generation. The problem is found by taking assumptions that the coil is a heat generator and that the coil can be burnt down by heat. Heat generation is proportional to the product of voltage and intensity of the electric current, and the strength of attraction and repulsion between a rotor and a permanent magnet are proportional to the current intensity. Therefore when the motor operates with heavier load, the motor must be supplied larger electric current. It causes greater heat generation, and in the worst case the coil is burnt down by the heat.

4 Design Object Modeling Environment

4.1 Definition of physical phenomena

Based on the idea of organizing aspect models in a multi-world environment, we implemented an integrated design object modeling system in Smalltalk-80[1]. We use Qualitative Process Theory [3] to derive physical phenomena by qualitative reasoning, since it provides a useful framework to store definitions of physical phenomena. **AspectOf** predicates are represented as preconditions of views. A qualitative aspect model consists of views and processes that occur over mechanical elements.

We use multiple inheritance and delegation mechanisms in order to represent physical phenomena. A concept about physical phenomenon is organized into a

[1] Smalltalk-80 is a Registered Trademark of Xerox Corp.

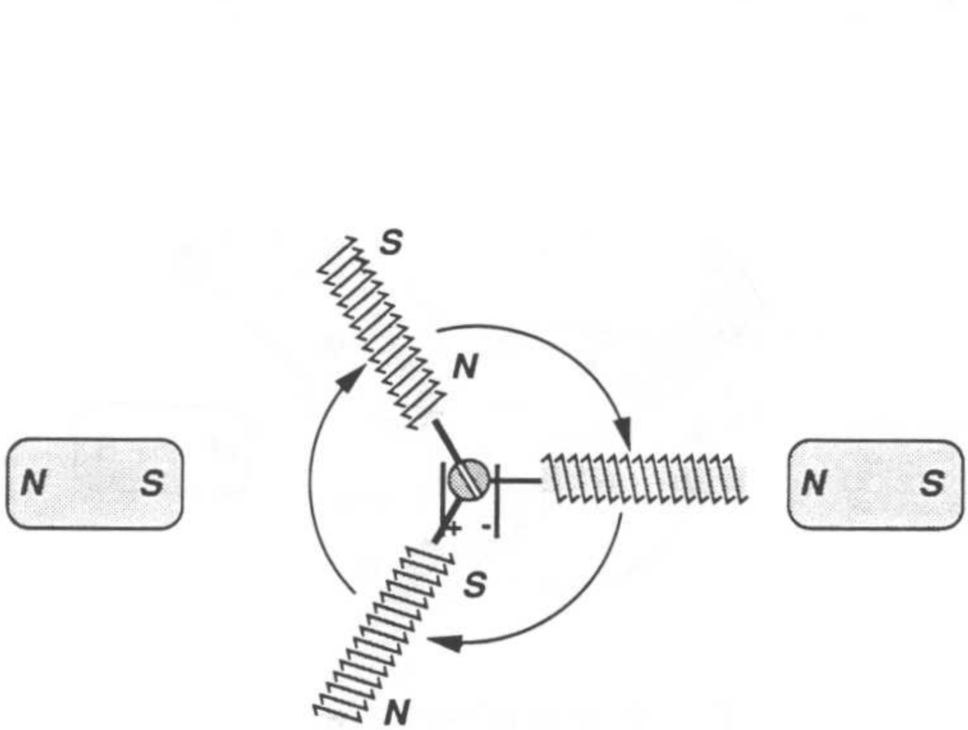

Figure 7: A three-pole motor

multiple inheritance hierarchy, so that it can have more than one super concepts. Since there are different systems of hierarchy depending on the viewpoint of organizing concepts, we need multiple inheritance to treat them. Delegation [10] is a mechanism to allow an object to use properties of others. For instance, the designer can instantiate a box made of metal to cover a mechanism, and later he can unify the box with an instance of electric path so that the box also works as a part of ground circuit. Since the delegation mechanism dynamically changes class-instance relationships, we do not call definitions of views and processes classes. Instead, we call them prototypes. An instance created from a prototype has the original prototype in its delegation list. If the instance is given a new prototype from which it delegates, the new prototype is added to the delegation list.

4.2 Model Generation and Envisioning

A metamodel is created from a primary model using qualitative reasoning. The qualitative reasoning technique used here must be able to deduce all possible behaviors, thus the metamodel includes dependencies among concepts of all physical phenomena the system concerns. A qualitative reasoning method that calculates chronological state transitions from an initial situation (*e.g.* [9]) is not suitable for the purpose. To do this, we use a QPE (Qualitative Process Engine) [4] that works in the multiple world environment of ATMS [1]. When QPE generates an instance of a view or a process, ATMS marks it with assumptions for it. The assumptions used by ATMS are values of parameters, relationships between parameters, and preconditions over individuals. A qualitative situation is defined by a set of assumptions believed in the situation. Active views and processes in a situation are obtained from ATMS by specifying a set of assumptions for the situation.

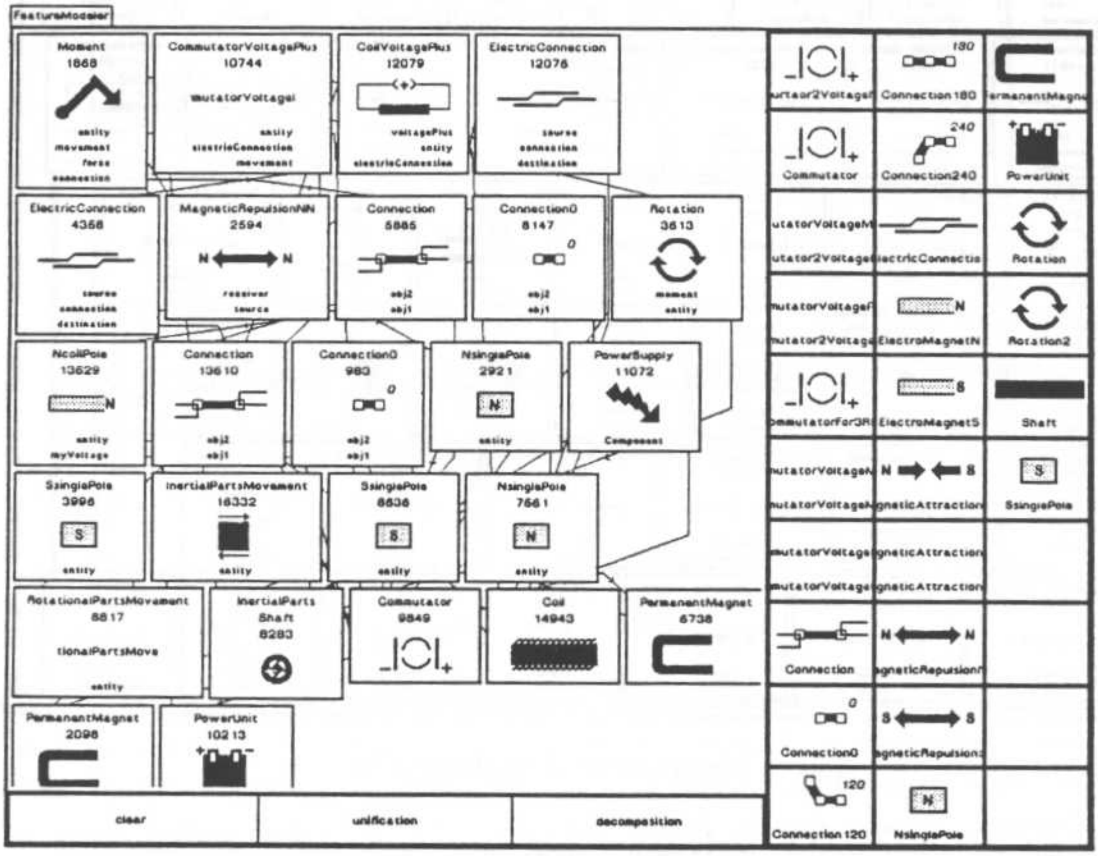

Figure 8: Primary model of a single-pole motor

4.3 Models of a Motor

Let us demonstrate the system with an example about design of electromagnetic motors. Figure 8 depicts the primary model of a single-pole motor consisting of a coil on a rotor and a pair of permanent magnets around the rotor. A commutator connected to the coil alters the directions of electric current through the coil. The designer makes the primary model of the motor by instantiating physical features such as rotation, attraction, repulsion, commutator, generation of a magnetic pole, and connection between mechanical elements.

Figure 9 depicts the metamodel of the single-pole motor created from the primary model. For instance, the primary model includes an electric current through a coil.

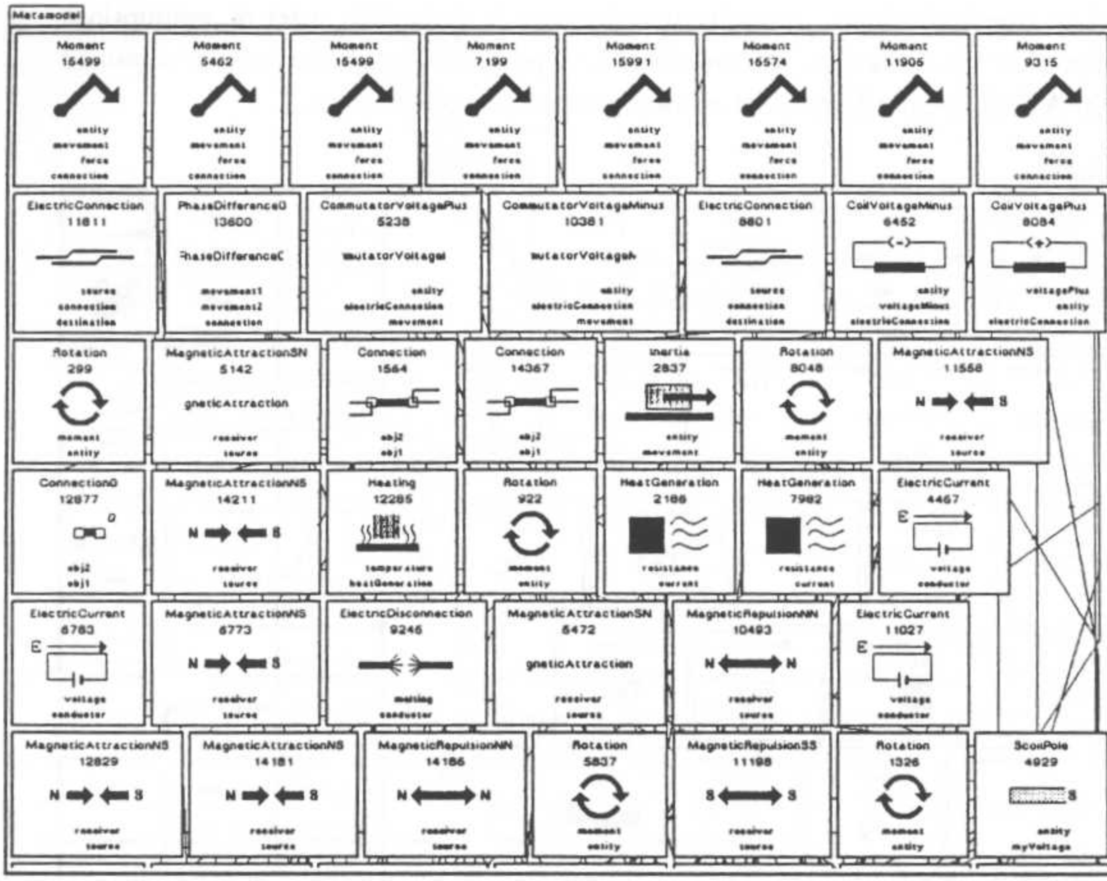

Figure 9: Metamodel of a single-pole motor

In creating the metamodel, the system reasons out that the electric current causes heating up the coil, so the coil can melt when the temperature rises to the melting point, and as a result the coil can be disconnected. A browser shown in Figure 10 is an aspect selector. A column of the browser displays a list of physical phenomena in each possible aspect. In the case of single-pole motor, there are four aspects (*i.e.*, geometric, dynamics, electric, and heat) about which the designer can generate aspect models.

Figure 11 illustrates a dynamic model of the single-pole motor, physical phenomena of which are marked with **aspectOfDynamics**. The model includes physical phenomena such as attraction, repulsion, mement of a force, and rotation. Behavior of the motor is predicted as a series of episodes shown in the lower window of the browser. The motor begins to rotate from the initial state episode1 ($0 < angle\ \theta < \pi,\ 0 = velocity\ v$). Episode2 ($0 < \theta < \pi,\ 0 < v$) to episode5 ($2\pi = \theta,\ 0 < v$) appear cyclically as the motor rotates.

There are eight forces between magnetic poles of the coil and the permanent

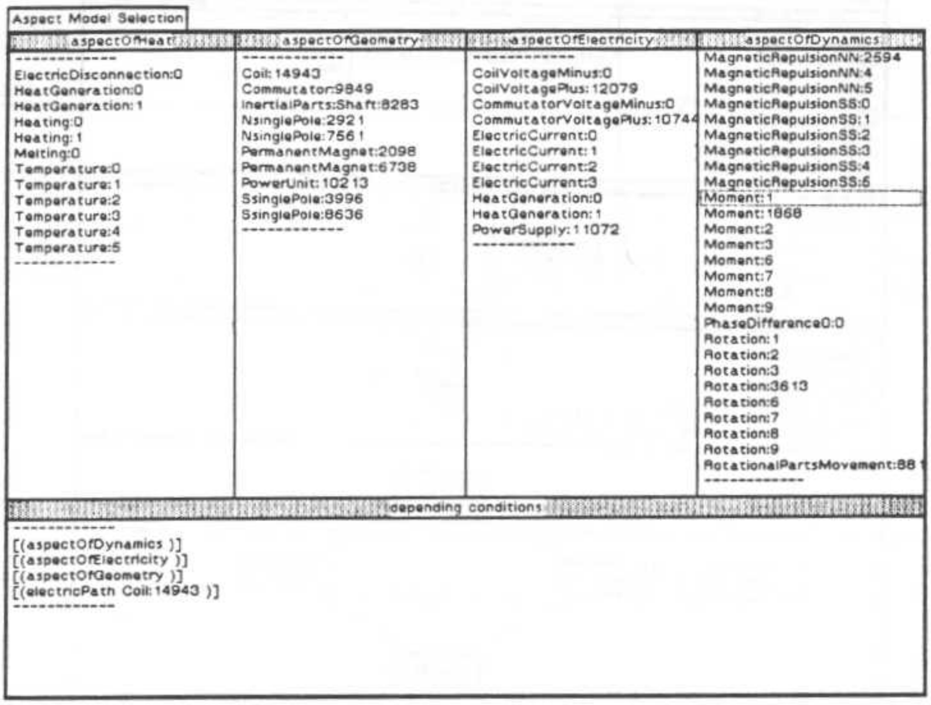

Figure 10: Aspect browser

magnets (four poles of the permanent magnets for two poles of the coil). The system allows the designer to numerically compute the magnitude and directions of the forces. To do the computation, the system needs to use information about layout of the elements of the motor. The system provides the designer with a two dimensional layout model depicted in Figure 12 to arrange coils and permanent magnets. Elements in the layout model are referred from the corresponding mechanical elements represented in the metamodel. And information about positions of the elements in the quantitative layout model is accessed from a quantitative dynamic model through the qualitative dependency network of the metamodel.

Figure 13 illustrates changes of the resultant force on the shaft over the angle of revolution between $0 < \theta\pi$. The force on the shaft tends alternatively toward left when $0 < \theta < \pi$ and right when $\pi < \theta < 2\pi$.

Figure 14 depicts the primary model of two-pole motor. The motor uses two coils for compensating forces from two permanent magnets. The graph shown in Figure 15 is the resultant force in the shaft calculated with the quantitative computation module. Although there is still some imbalance of forces due to asymmetric layout, it is smaller than that of the single-pole motor.

Figure 16 is a heat model of the two-pole motor, generated from the metamodel

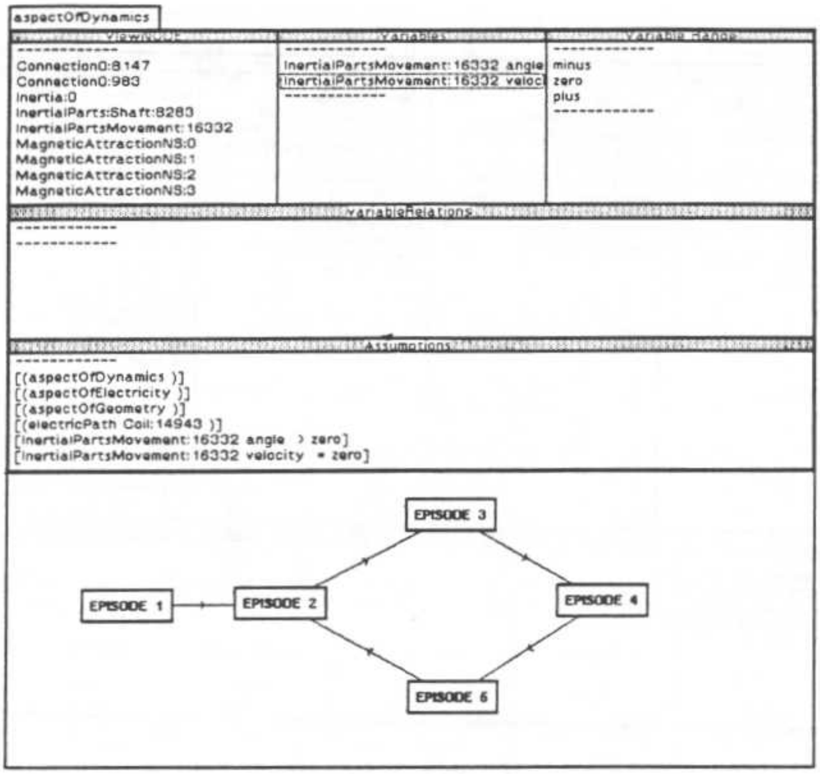

Figure 11: Dynamic model of a single-pole motor

by setting assumption **aspectOfHeat** true. The model includes concepts about temperature, heating, melting, and heat break. The model predicts that temperature of the coil can rise due to the electric current through it (episode1), and it can reach the landmark of melting point (episode2).

The coil has two states, *electricPath* and *notElectricPath*. When the temperature of the coil reaches the melting point, physical phenomena *heatBreak* changes the state into *notElectricPath*, and the electric current through the coil stops. The designer can propagate this state to the dynamic model through the metamodel mechanism. If the state is propagated to the dynamic model, the electric current does not exist any more and the motor does not rotate.

Figure 17 is the primary model of the three-pole motor. Since at least one coil is supplied with electric current by a commutator, the motor has no dead point.

5 Related Work

There is some research related to physical feature based design. Roth [11] collected physical phenomena used for mechanical design. He compiled a design catalogue from the collection to help designers to look up alternatives of physical phenomena

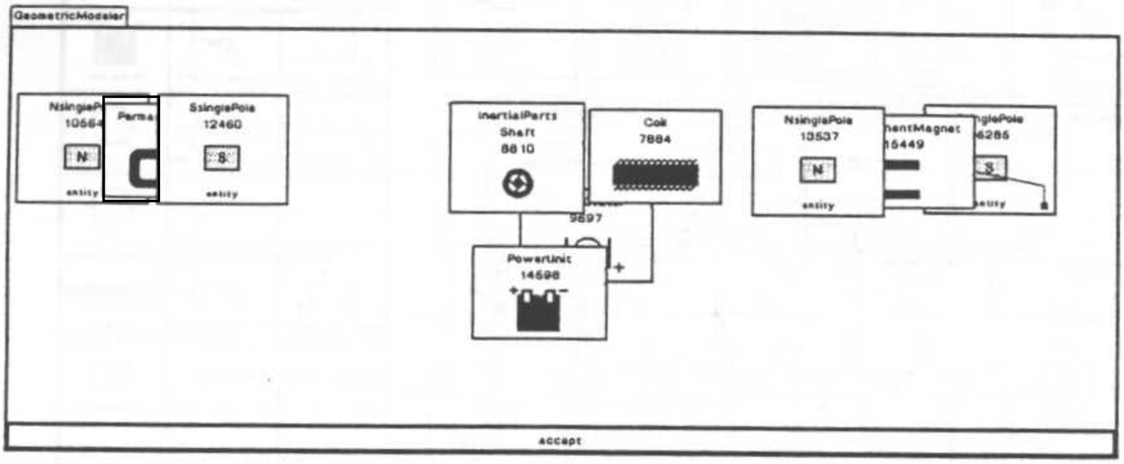

Figure 12: Layout model of a single-pole motor

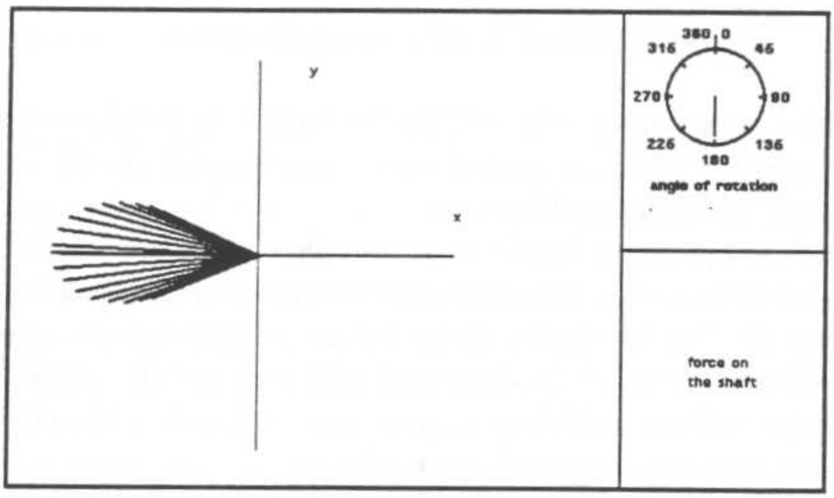

Figure 13: Quantitative dynamic model of single-pole motor

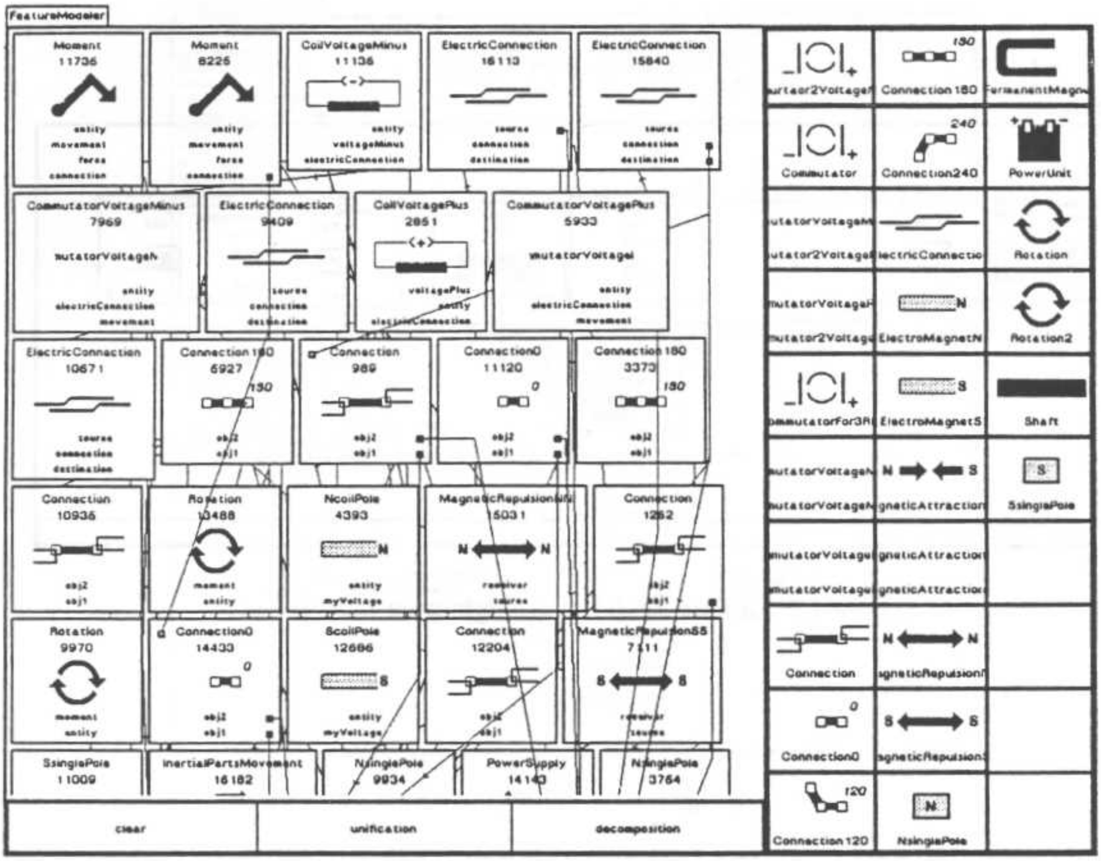

Figure 14: Primary model of a two-pole motor

for design. Although the catalogue is useful for designing behavior of a mechanism, they are not meant for CAD. Our research on physical features aims at collecting engineering knowledge in a computable form.

We have started a project to build a large-scale physical feature database. The project currently focuses on collecting physical features concerning kinematics, robotics, and classical physics. The reasons for choosing these domains are that knowledge in the domains are fundamental for mechanical engineers, and that since domain theories are already well established, we can use many textbooks and handbooks to obtain systematic description of domain knowledge.

How Things Work [6] is another large-scale database project at Stanford University. As a part of the project, Iwasaki [5] discusses multiple-level abstraction in modeling behavior. Although shifting from one grain size to another in qualitative reasoning is important, the metamodel mechanism presented in this paper can only deal with consistency in one grain size.

Falkenhainer and Forbus [2] proposes a method to generate a model most ap-

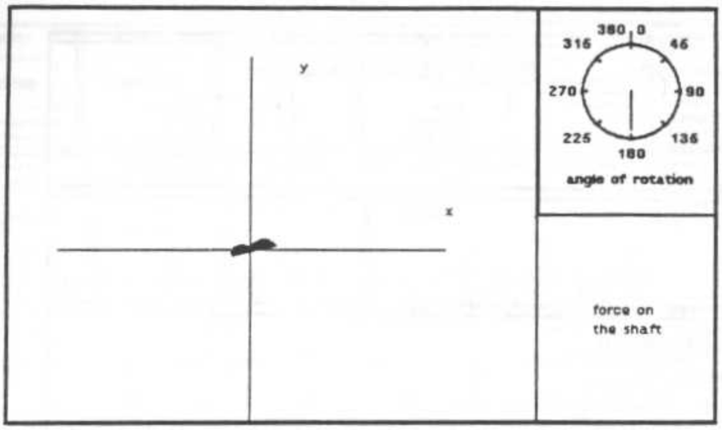

Figure 15: Quantitative dynamic model of two-pole motor

propriate to answer given questions. Their approach is suitable for generating a model that predicts expected conclusion (*i.e.* in CAI). But particularly in design, since the designer is interested in all aspects of behaviors of the design object, the system is expected to generate and take care of models of all the viewpoints, and to keep consistency among them without bothering the designer. Therefore their motivation is different from that of our research. But in modeling a large design object, we need to focus on a restricted area in order to make a treatable model. So the request of model generation is interactively given by the designer, and in such a case their approach will be useful.

6 Conclusions

In this paper we discussed the use of a multi-world environment for model generation in an ICAD system. The designer inputs the primary model using physical features, and the system converts it into the metamodel which represents assumptions for modeling and physical phenomena derived from the assumption. Aspect models generated from the metamodel is organized by ATMS. Aspect models are labeled with environments containing assumptions for modeling. There are two kinds of modeling assumptions, *i.e.*, state assumption and aspect assumption. The metamodel reserves derivation dependency among physical phenomena. The dependency is used to maintain consistency between quantitative models that are connected to qualitative models.

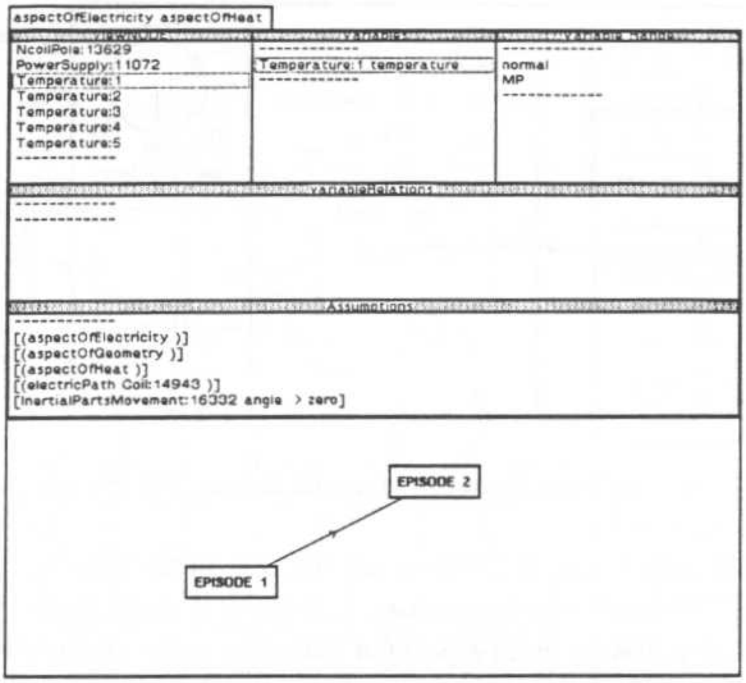

Figure 16: Heat model of a two-pole motor

References

[1] J. de Kleer. An Assumption-based TMS. *Artificial Intelligence*, Vol. 28, pp. 127–162, 1986.

[2] B. Falkenhainer and K.D. Forbus. Setting up Large-Scale Qualitative Models. In *Proceedings AAAI-88*, pp. 301–306, 1988.

[3] K.D. Forbus. Qualitative Process Theory. *Artificial Intelligence*, Vol. 24, pp. 85–168, 1984.

[4] K.D. Forbus. QPE: Using Assumption-based Truth Maintenance for Qualitative Simulation. *Artificial Intelligence in Engineering*, Vol. 3, No. 4, pp. 200–215, 1988.

[5] Y. Iwasaki. Reasoning with Multiple Abstraction Models. In *4th International Workshop on Qualitative Physics*, pp. 183–194, 1990.

[6] Y. Iwasaki, C.M. Low, and P. Nayak. DME: An Interactive Environment for Modeling Devices. Technical report, Knowledge Systems Laboratory, Stanford University, 1990.

[7] T. Kiriyama, T. Tomiyama, and H. Yoshikawa. Qualitative Reasoning and Conceptual Design with Physical Features. In *Proceedings of the 4th International

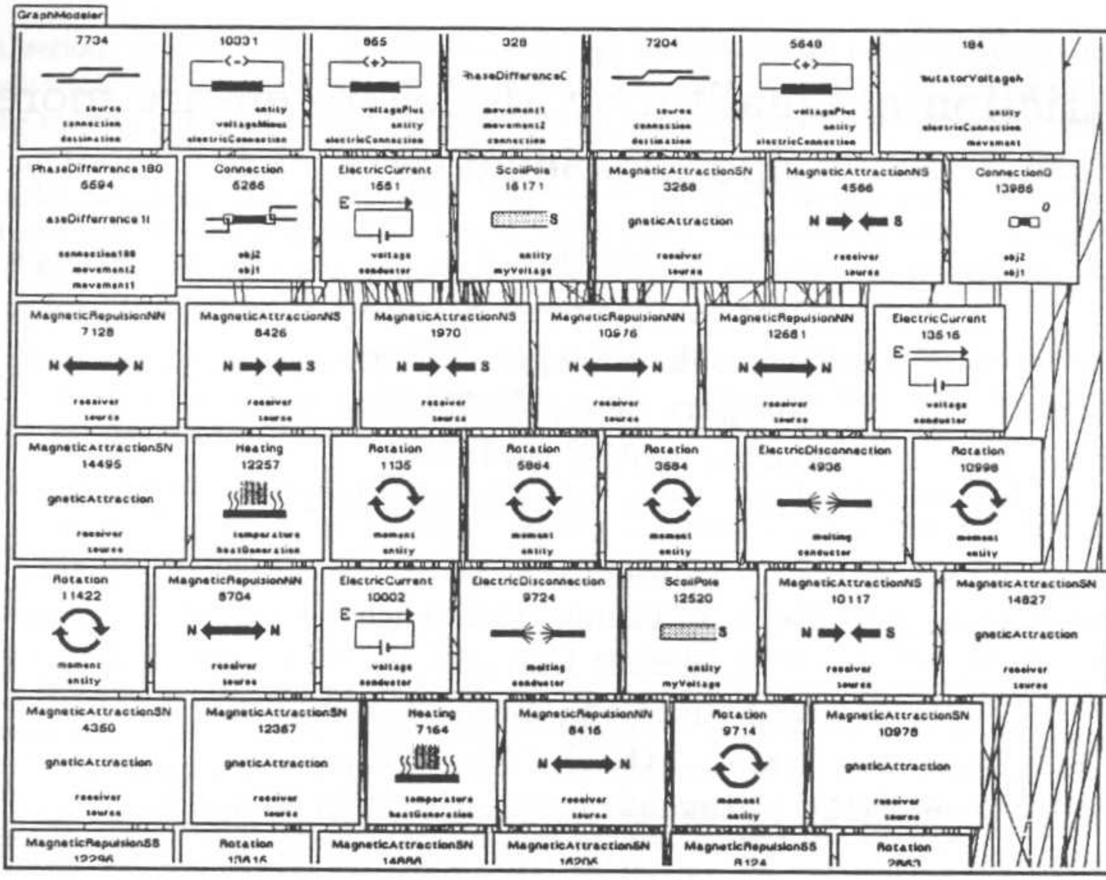

Figure 17: Metamodel of a three-pole motor

Workshop on Qualitative Physics, pp. 153–160, 1990.

[8] T. Kiriyama, F. Yamamoto, T. Tomiyama, and H. Yoshikawa. Metamodel: An Integrated Modeling Framework for Intelligent CAD. In J.S. Gero, editor, *Artificial Intelligence in Design*, pp. 429–449, Southampton, 1989.

[9] B. Kuipers. Causal Simulation. *Artificial Intelligence*, Vol. 26, pp. 289–338, 1986.

[10] H. Lieberman. Using Prototypical Objects to Implement Shared Behavior in Object Oriented Systems. In *Object Oriented Computing 1986*, pp. 189–198, 1986.

[11] K. Roth. *Konstruieren mit Konstruktionskatalogen*. Springer-Verlag, Berlin, 1982.