Reasoning about Models across Multiple Ontologies

Takashi Kiriyama
Research into Artifacts, Center for Engineering
The University of Tokyo
Yayoi 2-11-16, Bunkyo-ku, Tokyo 113, Japan
kiriyama@zzz.pe.u-tokyo.ac.jp

Tetsuo Tomiyama
Department of Precision Machinery Engineering
Faculty of Engineering, The University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo 113, Japan
tomiyama@zzz.pe.u-tokyo.ac.jp

Abstract

One of the central issues of intelligent CAD is building a modeling mechanism that deals with various models. The world in which a model is built, which call ontology, may vary over domains of modeling, abstraction levels, and granularities. This paper discusses integration of design object models over multiple ontologies. The key idea is the use of meta-level model about relationships among models.

Introduction

One of the central issues of intelligent CAD is building a modeling mechanism that can handle multiple models. In the modeling environment the designer represents the concept of the design object, and the system generates relevant models for evaluations. The system deals with models in an integrated way such that models are built and maintained consistent by using the same knowledge source.

Usefulness of such a modeling environment depends on its flexibility in adding new models. The system must be able to incorporate new models without restructuring the model management mechanism. In other words, models must be *pluggable* into the modeling environment. The pluggability is enabled by having explicitly described knowledge about dependencies among models. The dependencies include causal relationships among physical phenomena represented in models. This kind of knowledge serves in propagating qualitative changes in a model to another. And, models are related to ech other across difference in abstraction, approximation, granularity, and symbolic/numeric/diagrammatic [Iwasaki *et al.*, 1992] representations.

We have proposed a framework of design object modeling called the metamodel mechanism. A metamodel is a symbolic representation of physical phenomena, attributes, and their dependencies. By referring to the metamodel, the metamodel mechanism generates aspect-dependent models. Quantitative changes of an aspect model are propagated through the dependency network of the metamodel. Changes that cause qualitative change of aspect models are reflected to the metamodel, and the metamodel mechanism updates aspect models. Figure 1 depicts a metamodel and aspect models.

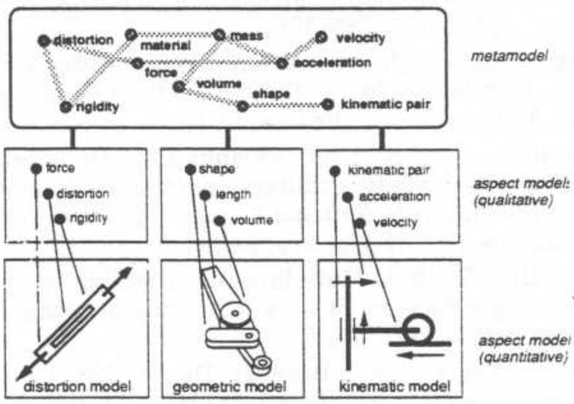

Figure 1: metamodel

In the previous work [Kiriyama *et al.*, 1991; Kiriyama *et al.*, 1992] we showed that the metamodel mechanism could deal with qualitative aspect models. From a structural model created by the designer, the metamodel mechanism reasoned out all possible physical phenomena using physical laws. The physical laws were represented based on the framework of Qualitative Process theory[Forbus, 1984]. The metamodel was a causal network among individuals and processes. Aspect models were generated from the metamodel by selecting physical phenomena relevant to the aspect. Justifications for physical phenomena are represented by using the ATMS [de Kleer and Brown, 1986]. If the user asks for consistency maintenance, the metamodel mechanism propagates a qualitative state from an aspect model to others.

Although benefiting from comprehensive representation of physical phenomena, the metamodel of the previous work also inherits limitations of Qualitative Process theory concerning multiple ontologies. For instance, since a process is not describe in terms of time, aggregation of state transitions to a longer-term process can not be treated. And components at different granularities should not be mixed, otherwise a component and its subcomponents may interact each other. Also, preconditions are not changed during envisioning, though they should rely on the qualitative state in relevant domains. In short, Qualitative Process theory as a single representational scheme is not sufficient to deal with relationships among physical phenomena across multiple ontologies.

Another issue is dealing with different levels of in abstraction and approximation. If two physical phenomena are reasoned out from the same structural model, their causal dependency can be found by qualitative reasoning. However, if they belong to models of different abstractions, approximations, granularities, or scopes, their dependency is not found by reasoning about causality among physical phenomena [Weld, 1992; Falkenhainer, 1992]. In order to make aspect models pluggable into the metamodel, the metamodel must at least be able to represent (and reason out if possible) different ways of modeling physical phenomena and structural components.

In this paper, we address reasoning about relationships among physical phenomena including causalities, structural and behavioral granularities, and parametric constraints. The metamodel is used for representing such dependencies. The rest of this paper is organized as follows. Firstly, we discuss dimensions of modeling. Secondly, we describe the metamodel for dealing with multiple ontologies. Then, examples about modeling an electromagnetic motor and a hydrofoil boat illustrate transformation of data across ontologies.

Dimensions of ontologies

Modeling is mapping from facts in a real world to a conceptual space. For instance, properties of a boat in Figure 2 are mapped to various conceptual spaces. The boat is represented as a solid model by focusing on its shape. The shape can also be represented as a simplified solid model. Another representation of the shape is a surface model. The simplest model of the boat for evaluating the strength is a beam model. Part of its structure is represented by a structural model. A finite element model is a detailed model for evaluating the strength.

A model in a conceptual space is represented using a set of concepts. Behind a model there is a

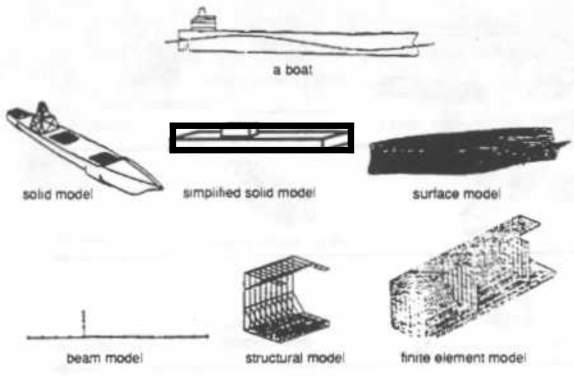

Figure 2: Models of a boat

theory of modeling that maps a real world to a model in a conceptual space. For instance, the simplified solid model of the boat in Figure 2 consists of boxes. The background theory of this model maps an object to a combination of boxes. We define the ontology of a background theory bases on as its conceptual space and mapping between a real world and the conceptual world. (Figure 3). There are dimensions that characterize differences of ontologies. They include domains, abstraction, approximation, simplification, granularity, and scope.

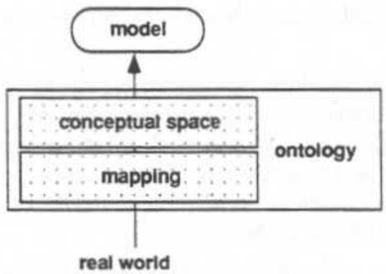

Figure 3: Ontology

Domains

There are various domains of modeling such as geometry, structure, material, control, electromagnetics, distortion, flow, heat, and strength. A domain of modeling is characterized by a set of physical phenomena and attributes. A strength model, for instance, consists of phenomena and attributes such as stress, compression, shape, and stiffness.

Abstraction, approximation, and simplification

Figure depicts relationships among abstraction, approximation, simplification, and equivalent

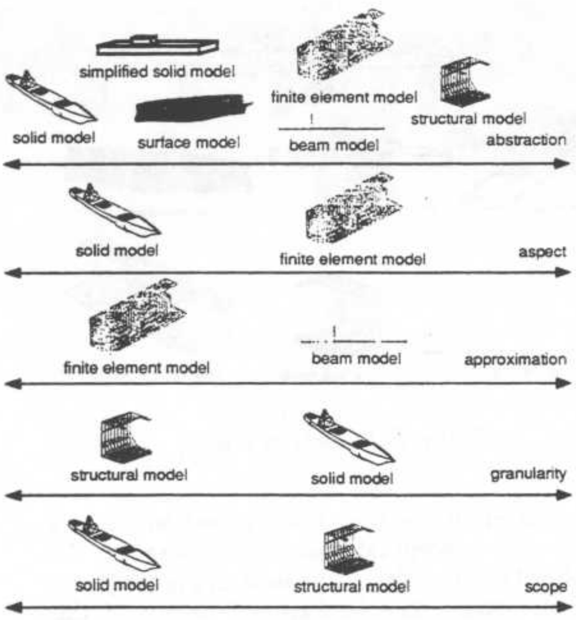

Figure 4: Dimensions of ontologies

transformation. Abstraction maps a fact about a physical world to a representation in the conceptual space. Even for one domain, there are more than one ways of abstractions. For instance, the shape of the boat in Figure 2 can be abstracted as a CSG model or as a boundary representation model.

Data of some models are derivable from other models, nevertheless the abstractions of both models are different. We call such relationship between models approximation. For instance, qualitative representation is an approximated model of quantitative representation.

Eliminating details of a model is called simplification. Simplification differs from approximation in the sense that it transforms a model within the same conceptual space. For instance, substitution of a curved surface to a plain face is simplification of geometry.

Granularities

Objects and physical phenomena can be modeled at different levels of resolution. Structural and behavioral resolution is called granularity. For instance, in order to calculate forces working on each link of a robot arm, a skeleton model consisting of links and joints is enough. At this granularity elementary entities are joints and links. If the loss of torque at a transmission is interesting, mechanisms of the joint is looked closer.

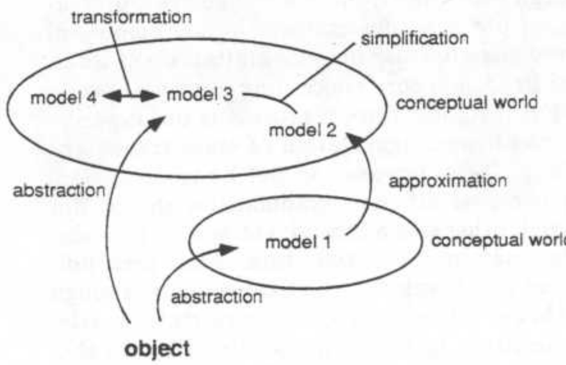

Figure 5: Abstraction, approximation, and simplification

Scopes

A model represents only a restricted area of a real world. The focused area is called the scope of the model. In finite element analysis, for instance, positions at which the stress is critical are focused on. Since influence from the outside of the scope is treated as boundary conditions, it reduces the size and computational cost of the model.

Modeling

Granularity

In building a metamodel from a structural model, the qualitative reasoner takes into account interactions among all combinations of physical phenomena. Since components at different granularities may interact each other, a structural model should not take a fixed level of granularity but hierarchical structural representation to reason out behaviors. However, a component and its subcomponents should not be paired to justify a physical phenomenon. For instance, an electric motor can be attracted by a permanent magnet of another motor but not by the one in it. Therefore, a qualitative reasoner should take into account interactions between a pair of components that are not in a super/sub relationship.

In order to avoid wrong pairing of components, we use contradictories of the ATMS. Belief of existence of an component in a model is represented with an assumption of an ATMS. The qualitative reasoner makes a conjunction node of two components in a super-sub structural relationship to be inconsistent. Contradictories between a pair of nodes of such components avoid physical phenomena on them to become active. The qualitative reasoner can remove such physical phenomena from the metamodel by checking their environment lists. Incompatible pairs are depicted in the structural hierarchy of components in Figure 6.

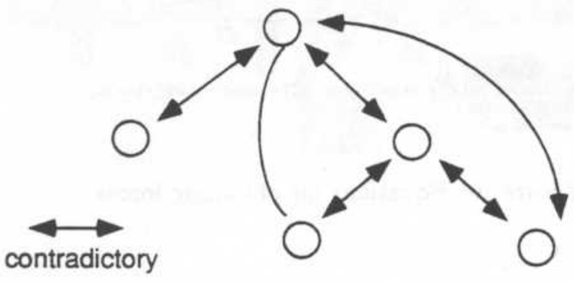

Figure 6: Contradictories between assumptions of entities

Abstraction

We use delegation [Lieberman. 1986] to represent a mapping from a component to its abstract representation. Delegation is a mechanism to make an object to share the same properties of a prototype.[1] For instance. an electric motor may be delegated by the prototype of energy transducer. This will adds it properties such as a transduce ratio and an efficiency ratio. It may be delegated at the same time by other prototypes such as a cylinder. or a mass. By delegation. super class hierarchy among prototypes is inherited and parameters are instantiated for the component.

Implementation of delegation in the metamodel mechanism is straightforward. A component has a superclass list from which it inherits parameters. Delegating a prototype to a component adds the prototype to the superclass list. And it duplicates parameters of the prototype for the component. After all. the component behaves as if it is an instance of a psedo-class which is defined as a common subclass of delegating prototypes.

Quantitative models may represent the same quantity in different units and coordinate systems. An example is a transformation matrix that converts vectors in one coordinate system to another. In the metamodel. there is a link between such representations. A metamodel may have any number of coordinate systems as abstractions of a space. There are links between them that are associated to transformation matrices. If transformation is needed. the associated matrix is obtained by referring to the link between two coordinate systems.

Approximation and simplification

Generally speaking. models can not be approximated or simplified regardless of the purpose of applications. For instance. for structural analysis any object may be approximated as a simple rectangular plate in an arbitrary orientation. But appropriateness of approximation depends on the precision required for the analysis. Therefore approximation must be done by the user or the the application. and the metamodel can represent how the approximation is done. In some cases. hierarchy among concepts can be used to show possible approximations or simplifications. For instance. prototype of rectangle plate is a subclass of prototype of plate. Thus a component delegated from a rectangle plate may be simplified as a plate.

[1] In this paper prototype means concept about a component.

Scope

In the metamodel. the scope of an aspect model is represented as a set of components. In transferring data from an aspect model to another. the metamodel is used as a reference model to identify corresponding components shared between aspect models.

Transformation across ontologies

Modeling a boat

In this section we illustrate integration of quantitative models of shape and force through an example about a hydrofoil boat. The shape of the hydrofoil boat is modeled by a solid modeler. The geometric information is used by a finite element mesh generator. To set up a finite element model. the metamodel mechanism calculates forces working on structural members.

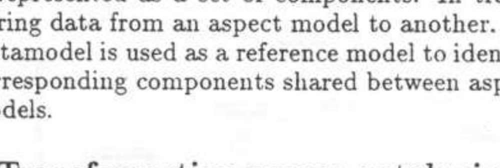

Figure 7: Solid model of a boat

The metamodel mechanism uses DESIGNBASE[2] as a plug-in solid modeler. Figure 7 depicts an assembled solid model of a hydrofoil boat. The

[2] DESIGNBASE is a trademark of Ricoh Company Ltd.

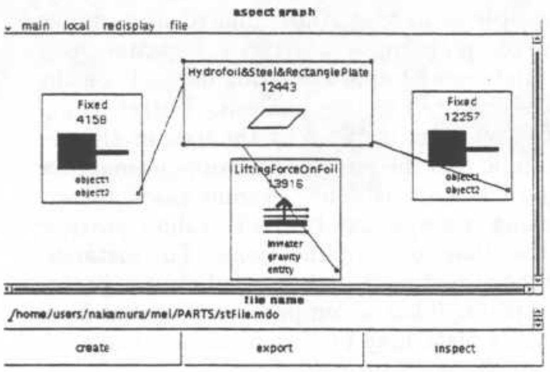

Figure 8: Aspect model of finite element analysis

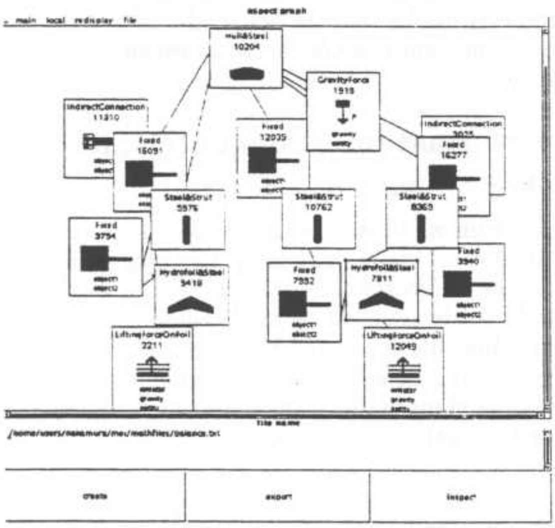

Figure 9: balance

boat has two hydrofoils supporting the hull in the front and the rear. The solid model consists the hull, the two hydrofoils, and the three struts connecting the hull and hydrofoils. The metamodel has symbolic representation of components corresponding to each part of the solid model. Using the correspondence between the metamodel and the solid model, the metamodel mechanism obtains geometric information from DESIGNBASE. For instance, if volume of a component is referred by an aspect model, the metamodel interprets it as a request to DESIGNBASE. Then the interface to DESIGNBASE issues a command for obtaining the volume, and the result is returned through the metamodel to the sender of the request.

MODIFY, a finite element mesh generator, represents a shape as connected thin shells. Using information about geometry, property of material, and boundary conditions, it creates a finite element mesh. To do so, the metamodel mechanism has to deal with delegation, coordinate transformation, calculation of weight, and equation solving. In order to generate a shell model for MODIFY, the designer explicitly specifies the two hydrofoils to be modeled as shells. This approximation is represented in the metamodel by making the hydrofoils to be delegated by prototype of plate. Figure 8 is a part of the metamodel relevant to MODIFY. Since the designer does not specify the hull to be delegated by a shell, MODIFY does not generate a mesh for the hull. It means that in this case specifying approximation also determines the scope of analysis.

DESIGNBASE has a global coordinate system for locating parts. MODIFY, on the other hand, locates a part in the local coordinate system of its connected part. The transformation between the global and local coordinate systems is represented in the metamodel as a link between them. The link is associated to a 4×4 transformation

```
zv = {0,0,0}
F = {0,0,-100}
n = {0,0,-1}
g = {0,0,0}
f1 = a n
f2 = b n
s1 ={-10,0,-1}, s2 = {20,0,-1}
eqs ={vectorproduct[s1,f1] + vectorproduct[s2,f2] + vectorproduct[g,F] == zv,
     f1 + f2 + F == zv }
ans = Solve[eqs, {a,b}]
```

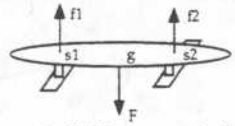

Figure 10: Equations for obtaining forces

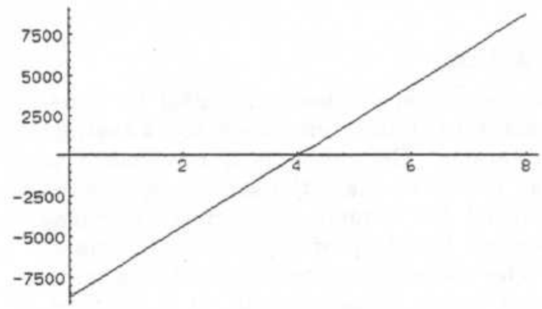

Figure 11: Shared force diagram of a hydrofoil

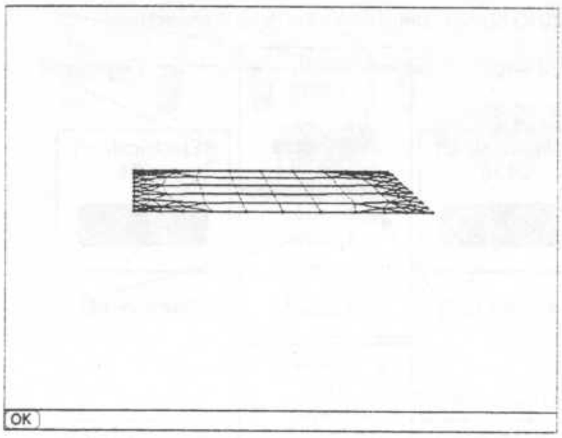

Figure 12: Generated mesh of a hydrofoil

matrix. When a solid part in DESIGNBASE is transformed to a shell in MODIFY, the position is transformed by the metamodel mechanism by referring to the transformation matrix.

To set up a model for finite element analysis, the metamodel mechanism calculates forces working on the hydrofoils. First it qualitatively reasons out that the hull is supported by forces from hydrofoils (Figure 9). It then uses the solid model in DESIGNBASE to obtain positions at which the gravity and the supporting forces are working on the hull. Since it is not yet specified that the forces are balanced, the designer has to add a constraint that the supporting forces should compensate the gravity force. By using knowledge about balance, it generates a set of equations of the boat's balance. The equations, shown in Figure 10, are then solved by Mathematica[3] to obtain quantitative values of the forces.

Then the system approximates the hydrofoil as a beam. From the beam model, the system calculates the distribution of shared force over the hydrofoil. The result is presented to the user as shown in Figure 11. MODIFY takes into account the degree of importance of areas in a part in deciding the size of mesh. From the shared force diagram, the designer knows that the both ends of the hydrofoil should be examined precisely whereas the middle are less important. The desiger specifies this analysis condition to control mesh generation. Figure 12 shows generated mesh for analysing distorsion of a hydrofoil. Mesh at each end is finer than that at the middle. Its boundary conditions are prepared by the metamodel.

[3]Mathematica is a trademark of Wolfram Research Inc.

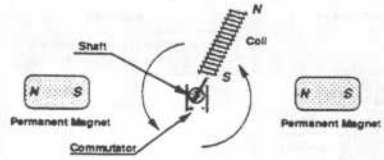

Figure 13: An electric motor

Models a motor at different granularities

Modeling an electric motor in Figure 13 illustrates how a metamodel deals with difference in granularities. The entire motor is modeled as a device that generates a moment of force. Also the motor can be modeled as an assembly of permanent magnets, electromagnetic coils, a commutator, and a shaft. In the assembled model, behavior of the motor is represented as a sequence of state transitions. Rotation of the motor in the aggregated model is justified from a state transition at the decomposed level.

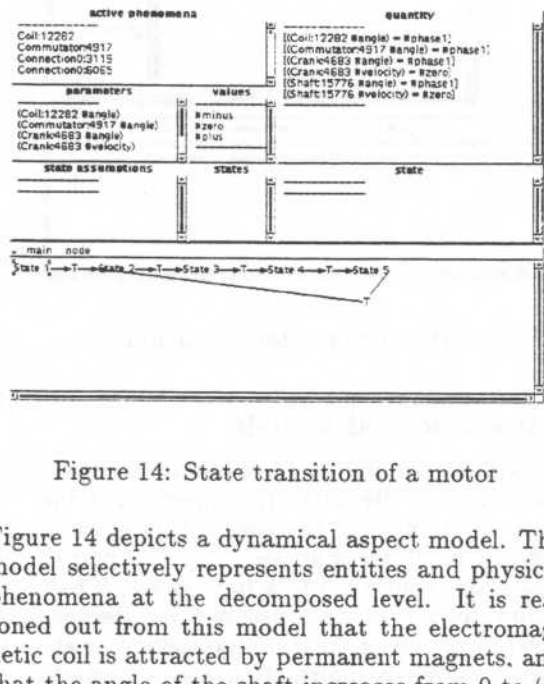

Figure 14: State transition of a motor

Figure 14 depicts a dynamical aspect model. The model selectively represents entities and physical phenomena at the decomposed level. It is reasoned out from this model that the electromagnetic coil is attracted by permanent magnets, and that the angle of the shaft increases from 0 to $(0, \pi)$, π, $(\pi, 2\pi)$, and $2\pi (= 0)$. The user can make the metamodel mechanism to assume that this sequence of transitions is believed by all other aspects. Figure 15 shows the dynamic aspect model of the entire motor. The model represents a physical phenomenon rotation of the motor. This rotation is believed if the state transition at the lower level is assumed.

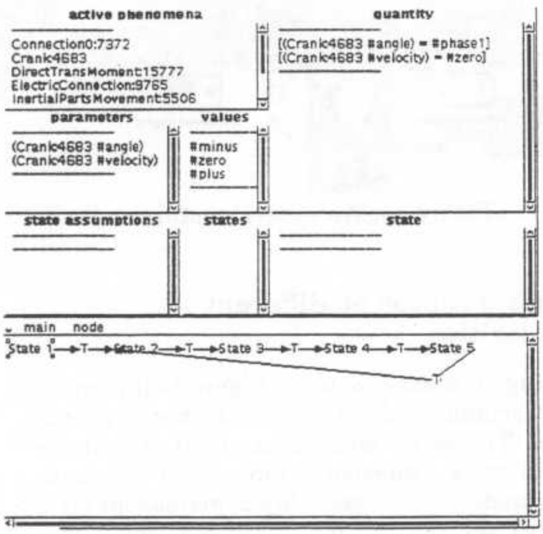

Figure 15: Rotation of the motor at the upper level

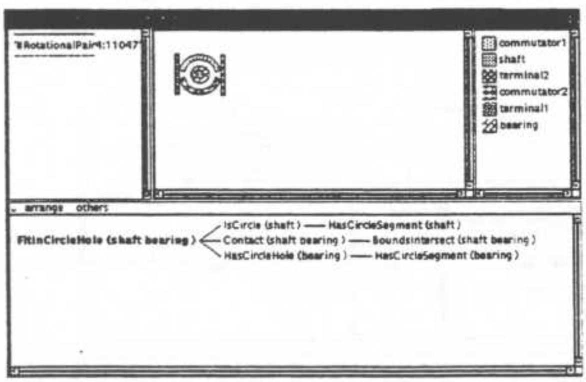

Figure 16: Commutator of a motor

Spatial and logical models

The electric motor in the example above has a commutator that alternates the direction of the electric current through the coils. This commutator can be modeled as an assembly of a pair of terminal plates connected to a battery and a pair of commutator plates connected to coils. Reasoning about behaviors of the commutator at this abstraction level needs information about contacts between the plates. To find spatial relationships between components, the metamodel mechanism uses a planar spatial model.

Figure 16 depicts a spatial model of the commutator. Shapes of the components are represented in two ways; one is boundary representation that uses lines and arcs, and the other uses bitmap images. By using the boundary representation, the

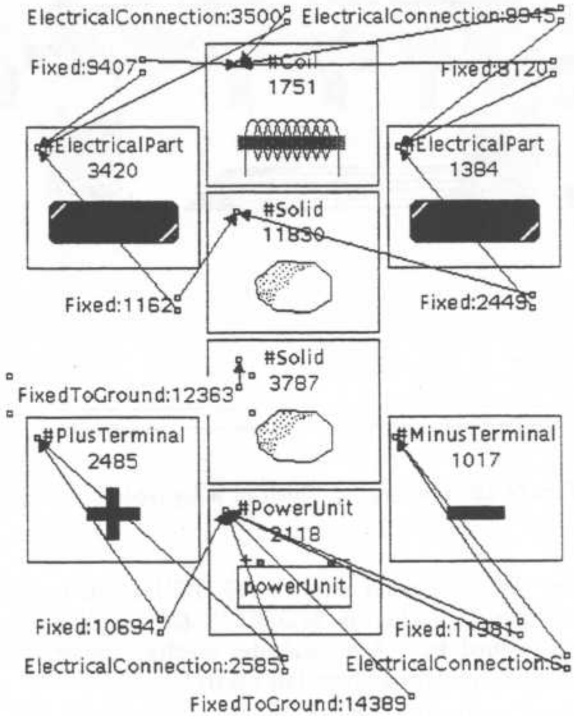

Figure 17: Primary model of a commutator

spatial modeler detects that the shaft in the center fits in the hole of the bearing. Since the bearing is connected to the frame, the shaft is found to have one degree of freedom around the axis. In the metamodel, the shaft is delegated by the concept of rotatable object. Since prototype of rotatable obhect has a parameter of angle, a new parameter of the angle of the shaft is instantiated.

Qualitative reasoning about the commutator finds that a physical phenomenon contact occurs between a terminal plate and a conductor plate. And from it existence of an electric path through the two plates is reasoned out. To determine if the physical phenomena occur, the spatial reasoner is used. In the spatial modeler, the angle of the shaft is incremented from 0 to 2π by a small step (e.g., $1°$). As the angle increases, the commutator plates connected to the shaft rotates around the shaft. At each angle, the spatial modeler creates a bitmap image of layout. Using the bitmap connections between components are checked. The picture in Figure 18 is a summary of the result of spatial reasoning. At angle 0 and π, contacts between terminal plates and commutator plates start or cease. The spatial reasoner interprets this information such that the parameter of angle needs landmarks at 0 and π. Introducing the two landmarks divides the quantity space of the

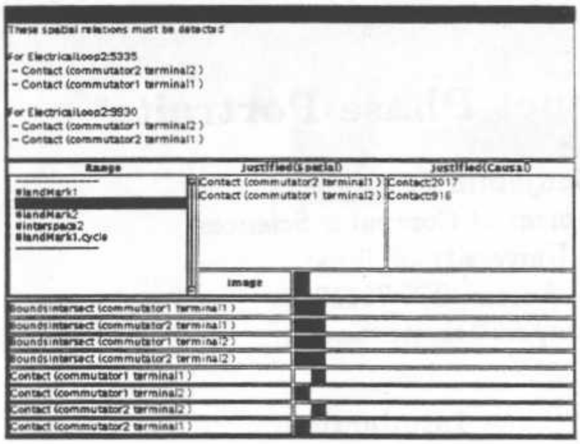

Figure 18: Spatial relationships at each angle

angle into two landmark values and two intervals. The spatial graph makes a mapping from qualitative values of angle to contacts. Since contact between battery plates does not ocur at any angle, it is removed from the metamodel. The metamodel mechanism translates the mapping to justifications in logical representation. By incorporating the justifications, qualitative reasoner can reason out rotation of the motor.

Conclusions

In order to integrate aspect models, the metamodel mechanism transforms facts represented in an aspect model to another. In addition to causal relationships among physical phenomena, the metamodel must represetn other kinds of relationships such as differences in the level of abstraction, approximation, granularity, and correspondences between symbolic and spatial representations. In this paper we discussed reasoning about models across multiple ontologies.

Acknowledgements

The authors are grateful to people who have shared this research. Masahiko Nakamura implemented the pluggable metamodel mechanism and the example of hydrofoil boat design. Shigeru Yoneda made the example of multiple granularity reasoning about an electric motor. Mikio Matsumoto, Masaki Ishii and Koji Yamazaki built the spatial and logical qualitative reasoning system.

References

de Kleer, J. and Brown, J.S. 1986. Theories of causal ordering. *Artificial Intelligence* 29:33–61.

Falkenhainer, B. 1992. Modeling without amnesia: Making experience-sanctioned approximations. In *The Sixth International Workshop on Qualitative Reasoning about Physical Systems.* 44–55.

Forbus, K.D. 1984. Qualitative process theory. *Artificial Intelligence* 24:85–168.

Iwasaki, Y.; Tessler, S.; and Law, K. 1992. Diagrammatic reasoning in problem solving. Technical Report KSL92-12, Knowledge Systems Laboratory.

Kiriyama, T.; Tomiyama, T.; and Yoshikawa.H., 1991. Model generation in design. In *Fifth International Workshop on Qualitative Reasoning about Physical Systems.* 93–108.

Kiriyama, T.; Tomiyama, T.: and Yoshikawa.H., 1992. Building a physical feature database for integrated modeling in design. In *The Sixth International Workshop on Qualitative Reasoning about Physical Systems.* 124–138.

Lieberman, H. 1986. Using prototypical objects to implement shared behavior in object oriented systems. In *Object Oriented Computing 1986.* 189–198.

Weld, D. 1992. Automatic selection of bounding abstractions. In *The Sixth International Workshop on Qualitative Reasoning about Physical Systems.* 36–43.