Proceedings of DETC'98
1998 ASME Design Engineering Technical Conferences
September 13-16, 1998, Atlanta, Georgia

98-DETC/CIE6007

MULTI DISCIPLINARY EARLY PERFORMANCE EVALUATION VIA LOGICAL DESCRIPTION OF MECHANISMS: DVD PICK UP HEAD EXAMPLE

Masanori Ozawa*
Systems & Software Research Laboratories
Research and Development Center
Toshiba Corporation
70, Yanagi-cho, Saiwai-ku, Kawasaki-shi, 210 Japan
ozawa@ssel.toshiba.co.jp

Yumi Iwasaki
Knowledge Systems Laboratory
Stanford University
Maple Avenue Stanford, CA 94305
iwasaki@ksl.stanford.edu

Mark R. Cutkosky
Center for Design Research
Stanford University
560 Panama Street Stanford. CA 94305-2232
cutkosky@cdr.stanford.edu

ABSTRACT

This paper describes an approach developed to support the early stages of designing multidisciplinary products. In the design of such products it is important for various specialists to share representations of the engineering models they use in making trade-off studies. Sharing information among diverse engineering tools at the data level is notoriously difficult and inflexible. When any of the modeling details, assumptions or constraints change, the translation must usually be rewritten. Our solution to this problem is to formalize the exchange of information among engineering specialists and their tools as a problem in communication among agents. The agents share a common ontology and a compositional modeling language (CML) in which models can be created such that conventions, constraints, and assumptions are defined formally and ultimately grounded in logic.

We have applied CML in the context of a pick-up head design problem for DVD (digital versatile disc / digital video disc) players. We describe the pick-up head design problem, the CML models we have created to address early stages of pick-up head design, and the use of the CML models by a team of specialists, each participating as an agent in a concurrent engineering exercise.

1. INTRODUCTION

In the early stages of designing products such as optical disc players it is important for the various specialists (mechanical, electrical, optical, software) to be able to create models rapidly and share them, to inform the other team members about their concerns, to address interactions among the different models, and to conduct system-wide trade-off studies.

The information sharing problem is complicated by the specialized tools and modeling conventions used by each specialist. For example, different finite element models are commonly used for elastic deformation and thermal and magnetic field analyses, circuit simulation software is used for designing the electronics, and ray tracing software may be used for the optics. The models are also interdependent. For example, in the design of pick-up heads for optical disc devices the choice of laser diode will affect the system heat transfer and component temperatures which, in turn, will affect the materials properties of elastic elements used in the servo mechanism and the field strength of permanent magnets used in the actuators.

A second problem is that the engineering tools are typically complex and require extensive, detailed information about the product geometry, materials properties, etc, with the result that models are time-consuming to develop or modify.

Much recent work has been directed toward solving these problems. For example, many computer-aided engineering software vendors now offer integrated 'toolboxes' that support multiple kinds of analysis (e.g., thermal and elastic) from a common data structure. Ongoing research is focused on making the models more semantically complete and flexible so that

---

*Visiting scholar, Center for Design Research, Stanford University.

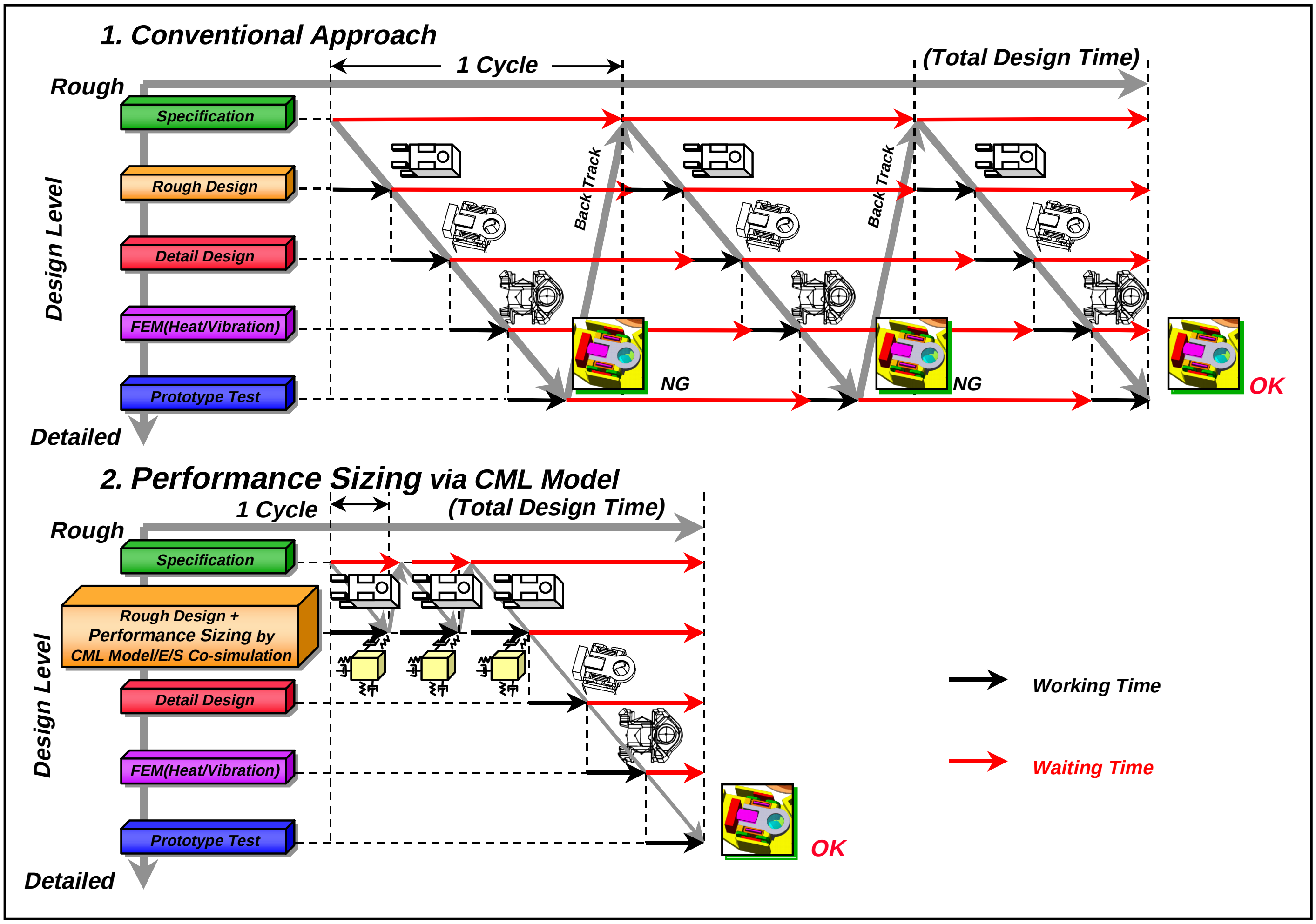

Figure 1.1: The effect of Performance Evaluation via CML model

changes in an evolving design can be accommodated without extensive rework. Examples include the product data models developed by Beam Technologies[1] and iSIGHT[2].

The ability to exchange information among different models and data structures is also improving through the adoption of more powerful and open programming interfaces (APIs) and standards for defining and exchanging software objects (e.g., CORBA[3], DCOM[4], Active X[5]) and product specifications (e.g., ISO STEP[6]).

Despite the recent advances in engineering modeling and information exchange, concurrent engineering teams today typically encounter long iteration cycles and delays resulting from an inability to effectively share information in the earliest design stages, as shown in the upper half of Figure 1.1. Our goal is to enable performance evaluation, including the effects of interactions among various physical phenomena, starting with the earliest stages of design. Our hypothesis is that if knowledge about such phenomena can be represented explicitly, it can be examined, extended and changed more easily. The result will be a concurrent engineering scenario like that shown in the lower half of Figure 1.1, in which information-associated delays are minimized.

Our approach is based on the idea the models used in early design stages should be represented in a semantically richer and more complete manner than is typically done today. The models should be grounded in a common *ontology* [Gruber, T. R., 1993] (a formal "dictionary" of terms and definitions in the domain) and built up from a common library of mathematical, physical and engineering elements (e.g., vectors, differential equations, units of measure, materials properties, physical laws) each of which is formally described and builds upon previous definitions.

In our work, the elements are defined and combined to create engineering models using a Compositional Modeling Language (CML) [Falkenhainer et al, 1994], [Bobrow et al, 1996]. CML is an object-oriented language that facilitates the

---

[1] http://www.beamtech.com/
[2] http://www.engineous.com/isight.html
[3] http://www.omg.org/
[4] http://www.microsoft.com/msdn/sdk/mactivex/docs/com/
[5] http://www.microsoft.com/activex/
[6] http://www.iso.ch/, http://www.scra.org/pdesinc.html, http://www.steptools.com/library/standard/

construction of behavior models of physical systems. The CML effort has focused on lumped-parameter systems models that are commonly used in early design stages. The purpose of these approximate models is to arrive at basic trade-offs regarding size, weight, heat, power consumption, etc. of the system components. We call this early-stage evaluation "**Performance Sizing**". We have used CML to represent several of the domains of physical behavior that are of importance in designing pick-up heads for optical disc devices. They can be used to analyze and simulate different physical phenomena (e.g., mechanical and electrical) simultaneously so that interactions among them can be addressed. The CML models also provide more complete descriptions of the components, and documentation of associated physical laws, assumptions, constraints, and dependencies than those typically found in lumped-parameter models built using commercial software.

In addition, the CML modeling environment is well suited for information sharing in a heterogeneous network environment. Different analysts and experimenters, and their associated tools acting through agents, can query and access shared CML models interactively. Therefore, models, model fragments, constraints, etc. can be shared during the early stages of design.

In the following sections we first briefly describe the current status of DVD pick up head design. Then we briefly ground our application of CML in the context of previous work on knowledge representation and agent-based engineering. Next we discuss some of the modeling issues associated with pick-up head performance and their representation using CML. We then discuss the use of CML models in an agent-based approach to concurrent engineering. We conclude with some observations about the merits and difficulties associated with building sharable models, based on our experience to date.

1.1 Current State of Pick Up Head Design

Optical disc devices have become an important data storage technology, with rapidly increasing performance. The Compact Disc, which made its debut in 1982, has 680 MB of capacity, a track pitch of 1.6 μm, and a minimum pit length of 0.83 μm. The DVD, launched in 1996, has 4.7GB of capacity, a track pitch of 0.74 μm and a minimum pit length of 0.4 μm. It is expected that the next generation of DVDs, to be launched in the near future, will have 3 - 4 times the capacity of today's devices. Along with increases in storage density, DVD players are expected to become smaller, more accurate, more reliable, less expensive, and designed and brought to market with increasingly rapid product generation cycles.

These expectations will present many technical challenges for all components of DVD players. A particularly critical component of the DVD player is the pick-up head (PUH) that positions the lens used to read the disc. For best performance, many interactions must be considered. Examples include undesirable rotational vibrations caused by eccentricity of the head and the effects of heat on the electromagnetic actuators and the material properties of the PUH structure. As discussed in the next section, a cross-disciplinary model for early stages of PUH design should help engineers to make these interactions explicit and to evaluate their effect on DVD performance.

Today, at Toshiba and other DVD manufacturers, FEM analysis is the tool of choice for evaluation of structural, vibration, magnetic, and thermal problems. However, to conduct FEM analyses we typically need the precise geometric shapes of products. The FEM models are time consuming to create and relatively difficult to modify, making them cumbersome for early design stages when many interactions are involved.

1.2 Motivation and Previous Work

The work described in this paper is being performed as part of a collaborative project between Toshiba Corporation and Stanford University. Our ultimate target is to shorten the design period. To achieve this target, we identified some problems, which make current multidisciplinary design processes inefficient. Prominent among these are:

1. The lack of a standard approach for representing specifications, functions, behaviors, and constraints among the various domains formally and in unified manner.
2. An inadequate infrastructure to support knowledge sharing among a diverse group of specialists and their computer-aided engineering tools.

These problems have been recognized for years and have been the subject of extensive research. For example, the ISO STEP effort is partly an effort to develop a standard for consistent and unambiguous representation of product models and their associated manufacturing processes. There are now many useful application protocols specialized to such industries as printed circuit boards, automobiles, shipbuilding etc. Nonetheless, it would be fair to say that the goal of a total life-cycle model for electromechanical products remains elusive. One particular limitation is that attention has been focused on the representation of static models using the EXPRESS language in ISO/STEP, without consideration of dynamic effects including state transitions and conditional behaviors.

In VLSI circuit development, the situation is more favorable. Designers often use a Hardware Description Language (HDL) such as VHDL or Verilog [Lipsett, R., Schaefer, C., and Ussery, C., 1989], [IEEE, 1995]. Using HDL, designers can describe behavioral and structural level specifications in a formal and standardized syntax. Synthesis tools and methodologies have also been developed to convert high-level specifications into physical-level designs automatically or interactively [Gajski et al, 1994], [Gajski, 1997]. However, an analogous language and specification tools are not yet available for electromechanical design. Consequently, it is more difficult to automate the low-level design of electromechanical systems and more difficult to share system specifications among different electrical, mechanical and optical designers on a team.

One attempt to overcome this limitation is the “metamodel” concept described by T. Kiriyama, T. Tomiyama, and H. Yoshikawa [1992]. The metamodel provides a framework for capturing relationships among engineering models corresponding to different domains (e.g., electrical and mechanical models of a product). The approach of Kiriyama et al, like that underlying CML, is partly based on qualitative physics and qualitative process theory [Forbus 1984].

In other work, Olsen *et al* [Olsen 1995] have argued that standards, such as ISO/STEP, for representing engineering models and products will always evolve too slowly to meet the needs of engineers working on state-of-the art multidisciplinary products. Therefore, what is needed is an environment that allows engineers to develop their own ontologies and encourages them to do it in a consistent manner that maximizes re-use of previously developed representations. It is from this line of research that the current project draws its main inspiration.

2. COMPOSITIONAL MODELING LANGUAGE

To approach the issues described in section 1.2, we used the Compositional Modeling Language (CML) to build models of PUH components and the associated domain. In this section we briefly describe CML and its characteristics that make it suitable for developing engineering models for early-stage design.

2.1 Origin and History of CML

Compositional modeling is an effective paradigm for formulating a behavior model of a physical system by composing descriptions of symbolic and mathematical properties of individual system components. CML is a general-purpose declarative modeling language for representing physical knowledge required for compositional modeling.

CML is intended to facilitate model sharing between research groups, many of which have long been using similar languages. These languages are based primarily on the language originally defined by Qualitative Process Theory [Forbus 1984] and include the languages used for the Qualitative Physics Compiler [Farquhar 1994], compositional model formulation [Falkenhainer 1991] , and the Device Modeling Environment [Low and Iwasaki 1993]. CML is an attempt to synthesize and provide a clean redesign of these languages. The specification of CML has been formulated by researchers involved in those projects.

CML was designed with efficiency, expressiveness and ease of use in mind. The language is restricted enough to allow efficient implementation of procedures to predict behavior. The language currently supports lumped-parameter ordinary differential equations that are common in engineering modeling. Finally, the language supports a variety of different approaches to representing physical phenomena; it allows the definition and use of domain theories that use components, processes, bond graphs, kinematic pairs, etc., and also supports both relational and object-oriented specification styles. CML specifies a set of top-level forms for defining models and an ontology of primitive functions, relations, and constants.

An important goal in designing CML is to support as much sharing as is reasonably possible. To facilitate sharing the content of CML knowledge bases, CML is fully translatable to the knowledge interchange format (KIF)[Genesereth and Fikes 1992].

2.2 Pattern of Use

To predict the behavior of a physical system in some domain, knowledge about the physics of the domain is captured in a general purpose *domain theory* that describes classes of relevant objects, phenomena and systems.

A domain theory in CML consists of a set of definitions, called *model fragments* or *entities*, each of which describes a piece of the domain's physics or objects. Once the domain theory has been constructed, it can be used to model different physical devices under a variety of conditions. The description of a specific system or situation being modeled is called a *scenario*. The user specifies a scenario that defines an initial configuration of the device, the initial values of some of the parameters that are relevant to modeling it, and perhaps conditions that further characterize the system. The CML implementation automatically identifies those model fragments that are applicable to the scenario. These model fragments are composed into a single model that comprises both a symbolic description and a set of governing equations. The equations may be solved or simulated to produce a behavioral description. Because the conditions under which the model fragments hold are stated explicitly in the domain theory, the system is able to assemble new models that describe the device as it moves into new operating regions.

2.3 Language Deffinition

A domain theory in CML is a finite set of the following top-level forms:

- **defEntity** for defining properties of persistent objects (e.g., resistors, containers).
- **defModelFragment** for describing the behavior of modeled entities under explicitly specified conditions. Model fragments are used to describe phenomena that arise out of the interactions of a composite set of objects (e.g., collisions or flows), or the behavior of a single object (e.g., a resistor, pump, or valve).
- **defScenario** for defining initial value problems consisting of a set of objects, their configuration, and initial values for the quantities that describe them.

The general syntax of a form is the form identifier (e.g., **defEntity**), followed by its name, followed by a series of keyword/value pairs. Some keywords are optional.

The **defEntity** and **defModelFragment** forms have been designed to support an object-oriented style of defining domain theories. Each form defines a class of objects specified by sets of static attributes and time dependent quantities. These attributes and quantities are effectively slots defined on instances of the class. Furthermore, these classes may be arranged in a hierarchy via the subclass-of clause.

The **defEntity** is used for defining properties of a persistent object that are always true. The **defModelFragment** form defines a class of phenomena, which are described by a set of objects involved, static attributes and time-dependent quantities. They also define consequences that hold only when an instance of the class is active. The **defModelFragment** form defines conditions sufficient to imply the existence of an instance, while the **defEntity** form defines only necessary consequences of an object being an instance of the class, not conditions sufficient to imply the existence of an instance.

The **defScenario** form is used for setting up problems in which the behavior of a system is to be predicted from a set of initial conditions. The main components of a scenario include the following:

- **Individuals:** The individuals clause names the set of objects that are assumed to exist initially.
- **Initially:** The **initially** clause specifies conditions that initially hold in the scenario. It may specify initial valuesof quantities, relations between quantities, time-dependent relations.
- **Throughout:** The **throughout** clause specifies conditions that hold throughout the scenario.

In addition, the forms **defRelation** for defining logical relations, **defQuantityFunction** for defining quantities used in the domain theory are provided as well as **defDimension**, **defUnit**, and **defConstantQuantity** for defining new or derived dimensions, new or derived units, and universal constants, respectively.

The full specification of the languages can be found in [Falkenhainer et al. 1994].

2.4 Collaborative Device Modeling Environment (CDME)

CDME is the web-based interface through which users can compose and interact with models in CML [Iwasaki et al, 1997]. If users define the domain theory in CML, and initial conditions or simulation conditions, etc. in a scenario, CDME will automatically interpret those definitions, convert them into an internal logical model, and prepare a procedure for numerical calculation. Currently, “Mathematica”[7] is the solver used in CDME. In addition, CDME can extract and generate equations from the CML model for use with an external solver or simulator. This architecture will be explained in section 4.2.

CDME also provides a function to show, graphically, the causal ordering of variables in a model which can help users to understand dependencies among variables during trade-off studies.

3. LOGICAL DESCRIPTION OF VIRTUAL PUH MODEL IN CML

To approach the issues described in section 1.2, we used CML to build behavioral models of the PUH, appropriate for the early design stages. The models capture several different domains of physical behavior, and the interactions among them.

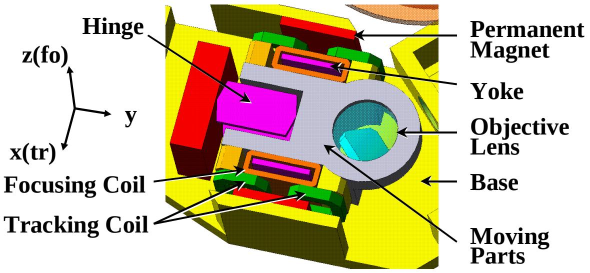

Figure 3.1: Structure of PUH

3.1 6-DOF Rigid Body Dynamics

The moving part of the PUH can be approximated as a rigid body with six degrees of freedom, suspended by a combination of linear and rotational stiffness and damping elements. Electrical coils actuate the body, typically along two orthogonal axes: focusing (fo) and tracking (tr). Without loss of generality, we can chose the *x* axis as the tracking direction and the *z* axis as the focusing direction.

The equations of motion are listed in equations (3.1.1 - 3.1.3). Equation (3.1.2) includes the gyro effect. In equation (3.1.2), an actuation force *F* caused by the electrical coils is calculated using Fleming’s law (3.1.3), where *l* is the effective length of coil in the magnetic field.

$$\left\{ \begin{array}{l} m_x \cdot \ddot{x} + c_x \cdot \dot{x} + k_x \cdot x = F_x \\ m_y \cdot \ddot{y} + c_y \cdot \dot{y} + k_y \cdot y = F_y \\ m_x \cdot \ddot{z} + c_z \cdot \dot{z} + k_z \cdot z = F_z \end{array} \right\} \tag{3.1.1}$$

$$\left\{ \begin{array}{l} I_x \cdot \dot{\omega}_x - \left(I_y - I_z\right)\omega_y \cdot \omega_z = F_z \cdot l_y - F_y \cdot l_z \\ I_y \cdot \dot{\omega}_y - \left(I_z - I_x\right)\omega_z \cdot \omega_x = F_x \cdot l_z - F_z \cdot l_x \\ I_z \cdot \dot{\omega}_z - \left(I_x - I_y\right)\omega_x \cdot \omega_y = F_y \cdot l_x - F_x \cdot l_y \end{array} \right\} \tag{3.1.2}$$

$$\left\{ \begin{array}{l} F = i \cdot B \cdot l \\ B = B_o \cdot r_{mt} \cdot r_{md} \end{array} \right. \tag{3.1.3}$$

---

[7] http://www.wri.com/

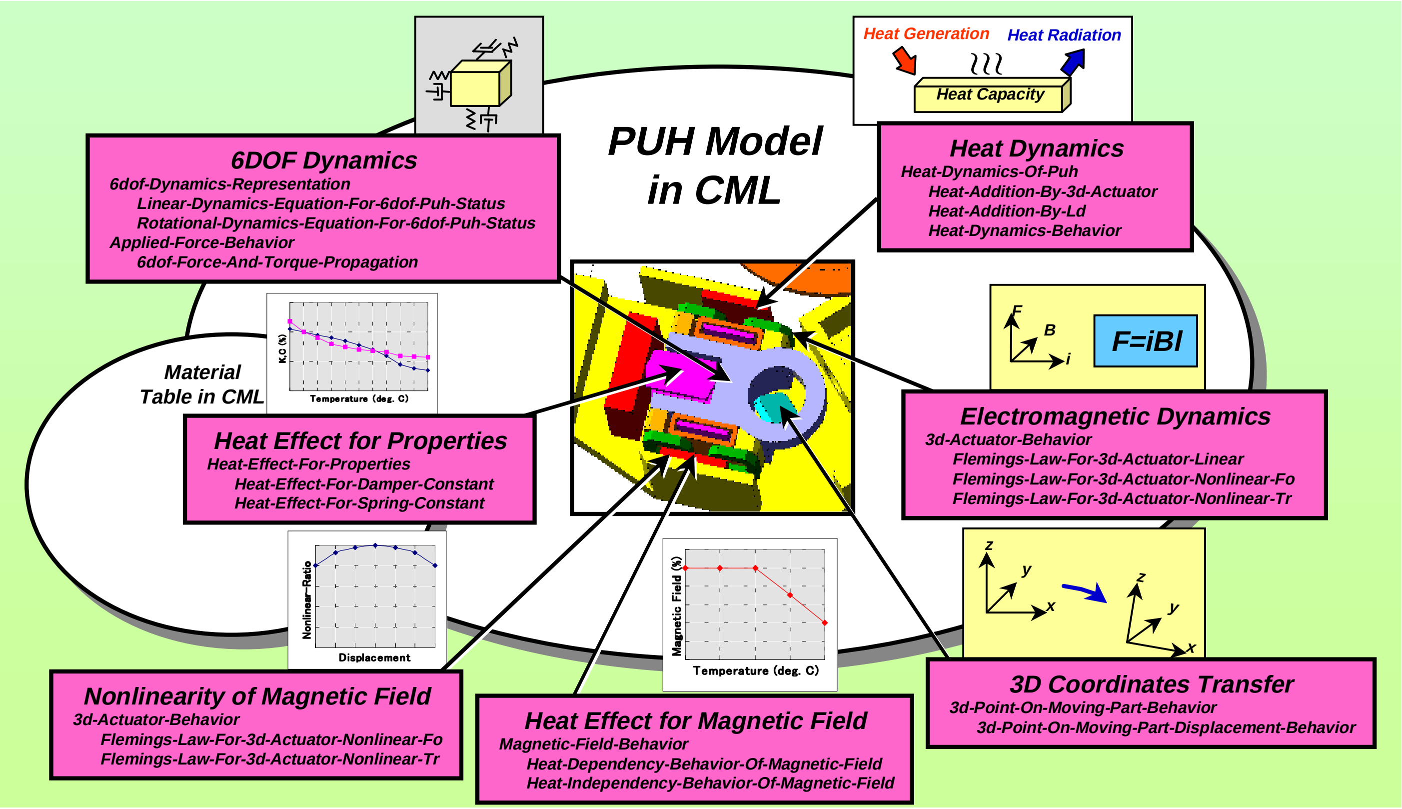

Figure 3.2: PUH Model in CML

One issue of importance is that the line of action of the electrical coils should coincide with the center of mass so that undesirable rotations are not produced. This requirement imposes constraints on the PUH geometry and tolerances.

A second concern is that the strength of the magnetic field B in equation (3.1.3) is not always constant. As explained in sections 3.4 and 3.5, B is affected by the temperature and deformations of the moving parts of the PUH. To account for these effects we define two parameters r_{mt} and r_{md}.

To express these dynamics in CML, we divided them into two model fragments. The first describes force propagation, and the second describes the dynamic response.

The force propagation behavior model fragment determines the total applied forces on the PUH system. The participants are

“`6dof-Puh-Status`”
“`3d-Force`”
“`Center-of-Gravity.`”

For this kind of model, the “C+” operator is useful for decomposing statements like

```
a = b + c + d
```

into;

```
(C+   a  b)
(C+   a  c)
(C+   a  d)
```

Thus, in the model fragment “`6DOF-Force-And-Torque-Propagation`”, the x-components of “`3d-Applied-Force-Involved`” are added to “`X-Total-Applied-Force`.” If the model has other “`3d-Force`” entities, their x-components are added in the same manner.

The dynamic behavior of equations (3.1.1-3.1.2) is expressed by the fragment “`Linear-Dynamics-Equation-For-6dof-Puh-Status`” as shown in Figure 3.3. In this model fragment, the participants are

“`3d-Mass`”
“`C-Matrix`”
“`K-Matrix`”
“`3d-Inertia-Tensor`”

When the model fragments for the applied force and linear dynamics are combined as part of a scenario in the CDME environment, the 6-DOF equations are automatically put in a form for solution by Mathematica[7].

By creating a scenario, we can run simulations to determine, for example, the effects of actuator forces that are not applied exactly through the center of gravity.

Since we are more specifically interested in the motion of the focal point of the PUH lens, we can use a coordinate transformation model fragment defined in CML as “`3d-Point-On-Moving-Part-Behavior`”.

3.2 Heat Dynamics

The temperature of the PUH is an important parameter because it affects the laser diode, as well as the various physical properties of materials, magnetic fields, etc. The temperature

```
;;; Linear-Dynamics-Equation-For-6dof-Puh-Status
(Defmodelfragment Linear-Dynamics-Equation-For-6dof-Puh-Status
                  :Subclass-Of
                  (6dof-Dynamics-Representation Model-Fragment)
                  :Participants
                  ((6dof-Puh-Status-Involved-Unique :Type 6dof-Puh-Status)
                   (3d-Mass-Involved-Unique :Type 3d-Mass)
                   (C-Matrix-Involved-Unique :Type C-Matrix)
                   (K-Matrix-Involved-Unique :Type K-Matrix)
                   (3d-Inertia-Tensor-Involved-Unique :Type 3d-Inertia-Tensor)
                   (Center-Of-Gravity-Involved-Unique :Type Center-Of-Gravity)
                 :Consequences
                  ((== 0.0
                    (+
                     (* (M-X 3d-Mass-Involved-Unique)
                      (D/Dt
                       (D/Dt
                        (X-Displacement 6dof-Puh-Status-Involved-Unique))))
                     (* (C-X C-Matrix-Involved-Unique)
                      (D/Dt
                       (X-Displacement 6dof-Puh-Status-Involved-Unique)))
                     (* (K-X K-Matrix-Involved-Unique)
                      (X-Displacement 6dof-Puh-Status-Involved-Unique))
                     (-
                      (X-Total-Applied-Force
                       6dof-Puh-Status-Involved-Unique))))
                    :
                    :
                    :
```

Figure 3.3: The description for 6DOF Dynamics

rise in the PUH is determined by the dynamics of heat generation, conduction and radiation. Recently, PUHs are becoming smaller and there is a trend toward integrating the laser diode with the PUH assembly. This integration creates particular difficulties because the laser diode both produces heat and is adversely affected by heat. As DVD devices find their way into portable and automotive applications, where the ambient temperature may be high, the issue of thermal management will become critically important.

For a preliminary analysis, the temperature change in the PUH is governed by the following equation:

$$\frac{dt_{PUH}}{dt} = \frac{\left\{ {i_{fo}}^2 \cdot r_{fo} + {i_{tr}}^2 \cdot r_{tr} - c_{rad} \cdot \left(t_{PUH} - t_{room} \right) \right\}}{c_c} \tag{3.2.1}$$

where t_{PUH} is the temperature of the PUH, and c_{rad} and c_c are constants for heat radiation and heat capacity respectively.

This equation is described in the model fragment “`Heat-Dynamics-Behavior`”.

3.3 Temperature Dependency of Material

High-polymer materials are often used in the flexure (hinges) of the PUH to provide controlled motion of the lens with respect to the base. The stiffness and damping parameters of these materials are often temperature-dependent. The changes in stiffness and damping must be accounted for when designing the servo system..

The heat dependent stiffness (*k*) and damping (*c*) are expressed by equations (3.3.1) – (3.3.3).

$$k\left(t_{PUH}\right) = k_{20} \cdot e^{\left\{-c_k \cdot \left(t_{PUH} - 20\right)\right\}} \tag{3.3.1}$$

$$c\left(t_{PUH}\right) \propto \frac{\sqrt{m \cdot k\left(t_{PUH}\right)} \cdot \tan\delta}{\sqrt{1 + \tan^2\delta}} \tag{3.3.2}$$

$$\tan\delta = c_{c1} \cdot e^{\left(c_{c2} \cdot t_{PUH}\right)} \tag{3.3.3}$$

K_{20} ≡ spring constant at 20 (c°)

c_k, c_{c1}, c_{c2} ≡ constants that are peculiar to each high-polymer material

3.4 Temperature Dependency of Magnetic Field

Magnetic fields of permanent magnets are also affected by temperature. Although the sensitivity is not high at normal room temperatures, it may be a concern in automotive applications where the ambient temperatures around the PUH exceed 80 °C and the surface temperatures of actuator coils can exceed 120 °C.

An approximate expression of the temperature dependency is given by equation (3.4.1).

$$\begin{cases} r_{mt} = 1.0 & (t_{PUH} \le t_{cr}) \\ r_{mt} = 1.0 - c_{mt} \cdot (t_{PUH} - t_{cr}) & (t_{PUH} > t_{cr}) \end{cases} \qquad (3.4.1)$$

In this equation, c_{mt} and t_{cr} are constants that are peculiar to each magnetic material, c_{mt} is the slope of the curve shown in Figure 3.2 and t_{cr} is the critical temperature at which the magnetic field starts to be weakened by heat.

For certain materials, a more accurate model may be available in the form of a look-up table or a function. Such models can be incorporated using the “Blackbox Function” of CML. Explicit conditions can be imposed on the applicability of such blackbox models.

The model fragments for the actuator, including the heat dependency of the magnetic field, are shown in Figures 3.4 – 3.6. In these models `Heat-Dependency-Of-3d-Actuator` is determined according to the `Material-Of-3d-Actuator`. Each material property is already registered in the domain theory “`material table.`” Note that

```
;;; Flemings-Law-For-3d-Actuator-Nonlinear-Fo
(Defmodelfragment Flemings-Law-For-3d-Actuator-Nonlinear-Fo
                  :Subclass-Of
                  (3d-Actuator-Behavior Model-Fragment)
                  :Documentation
                  "This calculates the force generated by focusing actuator."
                  :Consequences
                  ((==
                    (Magnitude-Of-Applied-Force
                     (Generated-Force 3d-Actuator-Involved))
                    (*
                     (Nonlinear-Ratio-For-Fo-Coil
                      (Z-Displacement 6dof-Puh-Status-Involved-Unique))
                     (Directional-Sine-Against-Magnetic-Field 3d-Actuator-Involved)
                     (Input-Current 3d-Actuator-Involved)
                     (* (Magnetic-Induction 3d-Actuator-Involved)
                      (Heat-Dependency-Of-3d-Actuator 3d-Actuator-Involved))
                     (Effective-Length 3d-Actuator-Involved))))
                  :Conditions
                  ((== (Nonlinear-Flag-For-Magnetic-Field 3d-Actuator-Involved)
                    True)
                   (== (Actuator-Purpose 3d-Actuator-Involved) "fo"))
                  :Participants
                  ((6dof-Puh-Status-Involved-Unique :Type 6dof-Puh-Status)))
```

Figure 3.4: Actuator model obeying Fleming’s Law

```
;;; Heat-Dependency-Behavior-Of-Magnetic-Field
(Defmodelfragment Heat-Dependency-Behavior-Of-Magnetic-Field
                  :Subclass-Of
                  (Magnetic-Field-Behavior)
                  :Consequences
                  ((==
                    (Material-Temperature
                     (Material-Of-3d-Actuator 3d-Actuator-Involved))
                    (Temperature-Of-Puh 6dof-Puh-Status-Involved-Unique))
                   (== (Heat-Dependency-Of-3d-Actuator 3d-Actuator-Involved)
                    (Heat-Dependency-Of-Magnetic-Material
                     (Material-Of-3d-Actuator 3d-Actuator-Involved))))
                    :
                    :
                    :
```

Figure 3.5: The description for Heat Dependency of Magnetic Field

```
;;; 3d-Actuator
(Defentity 3d-Actuator
           :Subclass-Of
           (Entity)
           :Attributes
           ((Material-Of-3d-Actuator :Type Magnetic-Material)
            (Generated-Force :Type 3d-Force :Documentation
             "Generated force on 3d-actuator by current."))
           :Quantities
           ((Input-Current :Dimension Electrical-Current-Dimension
             :Piecewise-Continuous True)
            (Resistance-Of-Coil :Dimension Resistance-Dimension
             :Piecewise-Continuous True)
            (Actuator-Constant :Dimension Force-Over-Current-Dimension
             :Documentation "force(N)/current(A)" :Piecewise-Continuous True)
            (Directional-Sine-Against-Magnetic-Field :Dimension Identity-Dimension
             :Documentation
             "directional sine between magnetic field and current"
             :Piecewise-Continuous True)
            (Magnetic-Induction :Dimension Magnetic-Induction-Dimension
             :Documentation "magnetic-induction for equ.:F=iBl"
             :Piecewise-Continuous True)
            (Effective-Length :Dimension Length-Dimension :Documentation
             "effective length of coil in the magnetic field"
             :Piecewise-Continuous True)
            (Heat-Dependency-Of-3d-Actuator :Dimension Identity-Dimension
             :Piecewise-Continuous True)
                    :
                    :
                    :
```

Figure 3.6: The description for the 3-Dimensional Actuator model fragment

dimensions, units and the continuity are made explicit. Note also the use of explicit conditions (on the applicability of part of a model) and consequences (of the model).

3.5 Non-Linearity of Magnetic Field

Ideally, the field produced by the permanent magnet should be uniform over the range of motion of the actuator coils to obtain maximum servoing accuracy. However, the field usually weakens somewhat near the edges. This effect is captured by the ratio, r_{md}, where the full strength of the field corresponds to a value of 1.0 at the center of the field.

A typical curve is illustrated in the lower left corner of figure 2. This information is entered into CML in the form of a table.

4. CML MODELS IN AN ENGINEERING SCENARIO

In this section, we show the results of a typical cross-disciplinary performance simulation and describe how the simulation is implemented with CML forming part of an agent-based concurrent engineering environment.

4.1 Result of Performance Evaluation by CML Models

Harmful vibration, caused by eccentricity of the actuator forces with respect to the center of mass, was simulated using the CML models. In this case, the sensitivity of tracking error caused by eccentricity is 2.5 (μm/mm) and its frequency is about 220 Hz. The sensitivity of tracking error to angular errors in the actuation force is 14 (μm/deg.).

Heat dynamics, material properties, and the changes in magnetic field were simulated simultaneously (Figure 4.1). The solid black line shows the temperature change calculated by equation (3.2.1). This temperature is substituted into equations (3.3.1) - (3.3.3) and (3.4.1) to simulate each property. As shown in Figure 4.1, *C*, *K*, and the magnetic field were weakened in proportion as the temperature rose.[8]

By substituting these properties into equations (3.1.1)-(3.1.3), the dynamic behavior of PUH is calculated as shown in Figure 4.2. As the temperature rises, the amplitude of indicial response is increased and frequency is lowered. These changes affect the selection of servo constants. In addition, the temperature increase lowers the second resonance of the moving parts of the PUH. Servo designers can utilize such information in the early stages of control system design. Consequently, the risk of major back tracking is reduced.

---

[8] In this simulation, we used exaggerated values for the heat generation ratio to emphasize the temperature dependency of the PUH performance. However, the effects obtained in realistic models are qualitatively the same and are observed in real PUH products.

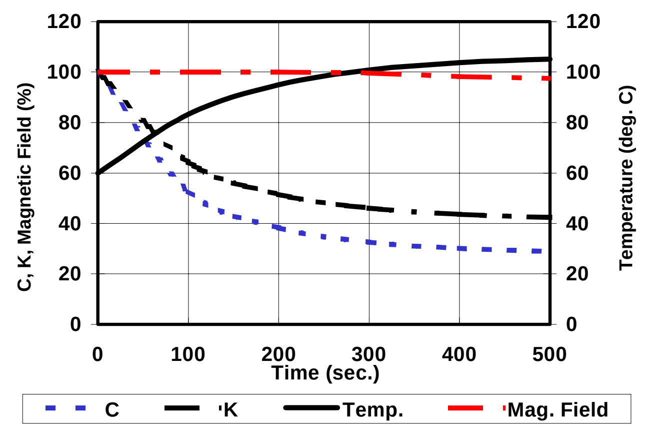

Figure 4.1: Heat Effect for Material Properties

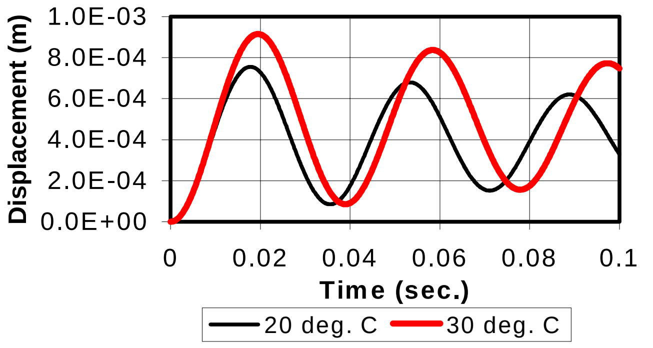

Figure 4.2: Indicial Responce

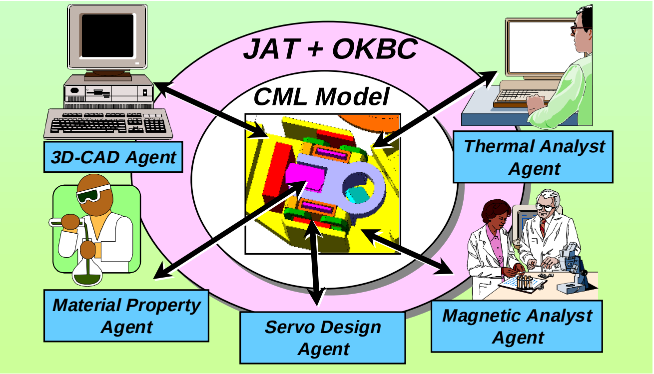

Figure 4.3: CML model in Agent Interaction System

4.2 Implementation of CML model in an Agent Interaction System

Although cross-disciplinary simulations can be performed entirely within CML/CDME, we do not anticipate that this will be the usual approach. As discussed in Section 2, CML is object-oriented, declarative language that emphasizes expressiveness and re-use of models and model fragments. It is not optimized for numerical efficiency. Moreover, we believe that most engineers will prefer to use their own specialized tools for computationally intensive analyses and simulations. The role of CML/CDME is to provide models that these engineers can interact with (view, query, post and refine).

To support interaction with CML, we have developed an agent-based framework as shown in Figure 4.3. This approach is based on the agent-interaction technology described in Cutkosky *et al* [1993].

The engineers and their tools interact through agents using the emerging standard communications language, KQML (see http://www.cs.umbc.edu/kqml/ for the current KQML standard). The communications are built on open Internet standards, TCP/IP, SMTP, and FTP.

To facilitate the wrapping of engineering tools we have developed the Java Agent Template (JAT). A description of JAT is beyond the scope of this paper, but details on the beta-release of JAT can be found at http://java.stanford.edu. Briefly, JAT is a package of programs written in the Java language that allow users to quickly create new software "agents" that communicate over the Internet. JAT facilitates especially construction of agents that send and receive messages in KQML and provides services including name registration, queuing and buffering of messages, connect disconnect and security. Agents can be stand-alone programs or applets, downloaded through a standard web browser. Interfaces have been created for programs written in C++ and Lisp as well as Java.

For interaction with CML models and scenarios, the agents use the Open Knowledge Base Connectivity (OKBC)[9] protocol from the Stanford Knowledge Systems Laboratory [Chaudhri et al, 1997]. OKBC provides access to the classes, individuals, slots, facets and so on, of the CML library. OKBC is based on

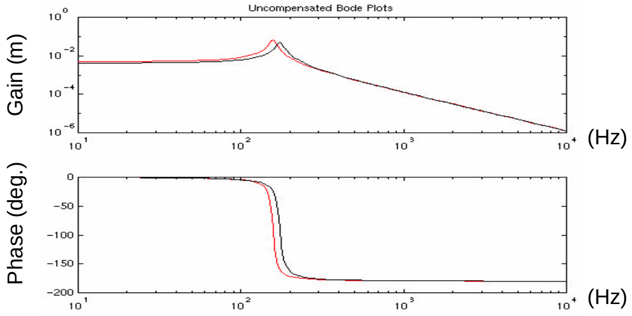

Figure 4.4: Open Loop Response

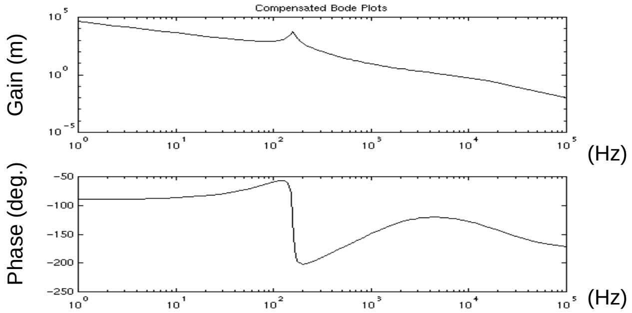

Figure 4.5: Closed Loop Response with Servo Agent

---

[9] http://ksl-web.stanford.edu/KSL_Abstracts/KSL-98-06.html
http://cml.stanford.edu/doc/release/okbc/index.html

the Generic Frame Protocol (GFP) [Karp, Myers, Gruber, 1995].

Using JAT, we have created the set of agents shown in Figure 4.3. The servo agent uses MATLAB[10] for control analysis, interacting with CML for concurrent dynamic, thermal, magnetic, and material simulation.

Figure 4.4 and 4.5 show the open and closed-loop Bode plots for the PUH.

The two different curves in Figure 4.4 show the effect of two different temperatures at 20 c° and 30 c°, respectively.

Figure 4.5 shows that by making an appropriate adjustment in the servo parameters, the servo agent can achieve identical performance in each case (the two curves are indistinguishable in Figure 4.5).

5. DISCUSSION

We have demonstrated how shareable models and scenarios can be developed in a compositional modeling language and used in an agent-based concurrent engineering scenario. In this section, we return to the motivation for our approach and draw some preliminary conclusions about its advantages and difficulties.

5.1 Reducing delays in early design stages

The hypothesis underlying our work is that the ability to construct and share models from different points of view will make it possible to shorten design cycles in the early stages of product design. By helping engineers to catch unintended side-effects and interactions in early design stages, we hope to reduce the amount of costly backtracking that is partly responsible for the delays in the upper half of Figure 1.1.

Modeling at the level of detail provided in CML is appropriate for early design stages because it does not depend on the exact geometry of the product. Of course, the accuracy of the simulation is not as high as it would be with FEM or other detailed numerical analysis tools. But in many cases, it gives designers enough information to evaluate the design alternatives. For example, as shown in Figure 4.1, 4.2, 4.4 and 4.5, CML models can provide useful information to the servo designer.

In addition, we note that the CML models are easily modified when more precise information becomes available. For example, the results of an FEM analysis can be incorporated into a look-up table that is used as a black-box model in CML for a more precise answer than the simple equations in Section 3 provide.

5.2 Reusability of CML Models

A chronic difficulty with engineering models is that they are difficult for people other than the creators to use (and sometimes even for the creators to use, after a long time has elapsed). The main advantage of creating models in CML is that more modeling information is captured and grounded in a formal framework. Once the domain theories about the target products are built up, a user can compose physical models which correspond to each particular design in a scenario. In this process, the user can inherit and reuse the basic knowledge from library. For example, many dimensions, units, mathematical operations and so on are pre-defined as hierarchical libraries that can be browsed in CDME.

CML/CDME also captures explicit information about the conditions of applicability of a model (indeed, a model fragment can have several different equations with explicit conditions governing when to use each), and the consequences of each model so that a logically consistent scenario is assured.

Another useful feature is the maintenance of causal ordering among parameters in a model. An example of part of a causal ordering graph is shown in Figure 5.1. This information can be useful when exploring design trade-offs.

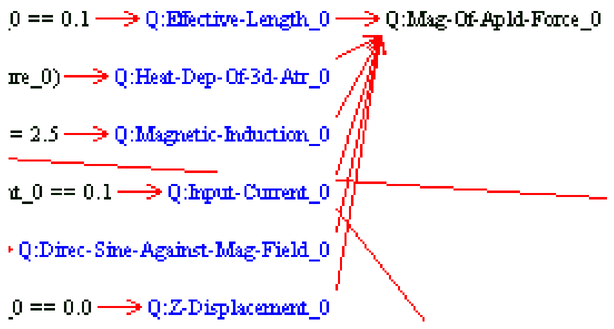

Figure 5.1: Partial View of Causal Order

Despite these advantages, we must admit some difficulty in achieving model re-use when constructing scenarios like the PUH example. For example, we found that there were pre-existing libraries in CML for spatial rigid-body dynamics. However, when considering the linear dynamics of the PUH, where only small angles of rotation are involved, the overhead associated with using these libraries was greater than the effort required to simply create some new models for linear dynamic response. On the other hand, many of the low-level definitions concerning units, dimensions, spatial transformations, and assembly/part relationships were drawn from previous CML modeling.

6. CONCLUSIONS AND FUTURE WORK

This paper describes an approach using a compositional modeling language, CML, and an interface for constructing shared models to be used in the early design stages of multidisciplinary products. We demonstrated the approach for an application involving the design of pick-up heads (PUH) of optical compact disc players. In this domain, the interactions among optical, electrical and mechanical models must be considered.

Our hypothesis is that by creating declarative models in CML, with explicit definitions, terms, conditions and consequences, the ability of other engineers to understand, and incorporate those models will be enhanced. Ultimately, this hypothesis will need to be verified empirically. However,

---

[10] http://www.mathworks.com

preliminary experience with constructing models and scenarios in CML suggests to us that model-reuse is facilitated (though still imperfect) and that it is possible for specialized engineering tools to interact with CML models if we wrap them with an agent interface. To our knowledge, this is the first industrial example of the kind of agent-based engineering interactions originally described in PACT [Cutkosky *et al*, 1993].

In our continuing work, we will develop and refine the models in CML and establish connections with more agents using the Java Agent Template. We are particularly interested in the interaction between CML models and models created using parametric 3D-CAD tools. If the geometrical parameters or properties of 3D-CAD models are dynamically accessible through the CAD API, they can easily be linked to each frame of a CML description using OKBC. This interaction would allow changes in CAD models immediately to be propagated into CML models and the interdependent models used by other engineers. In addition, parameter changes in CML could result in direct parametric modification of the CAD geometry.

Our second area of work will be to start to introduce compositional modeling in a corporate setting to evaluate whether it truly accelerates the design cycle, as we intend.

ACKNOWLEDGMENTS

The authors would like to thank the past and the present members of the Super Design Technology: Toshiba-Stanford project, Tadatoshi Ishii, and Dai Araki for their advice and assistance in this work. We gratefully acknowledge the contributions of Tony Looser, James Rice, Hideaki Takahashi, Adam Farquhar, and Richard Fikes from the Stanford Knowledge Systems Lab for their help with CML, CDME and OKBC. We also acknowledge the help of H. Jeon and B. Howley and N. Shodan with the Java Agent Template interface and MATLAB API integration.

The work at CDR has supported by Toshiba Corporation and DARPA, under contract N00014-96—1-0679.

REFERENCES

Forbus, K. D., (1984). “Qualitative Process Theory.” Artificial Intelligence, 1984, vol. 24, pp. 85-168.

Gajski, D., Vahid, F., Narayan, S., and Gong, J., “Specification and design of embedded systems” , 1994, Prentice hall.

Gajski, D., “Principles of digital design”, 1997, Prentice hall.

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

Bobrow, D., Falkenhainer, B., Farquhar, A., Fikes, R., Forbus, K., Gruber, T., Iwasaki, Y., Kuipers, B., “A Compositional Modeling Language”, 96 AAAI Technical Report WS-96-01, 1996, Fallen Leaf Lake, CA.

Cutkosky, M. R., Engelmore, R. S., Fikes, R. E., Genesereth, M. R., Gruber, T. R., Mark, W. S., Tenenbaum, J. M. and Weber, J. C. “PACT: An Experiment in Integrating Concurrent Engineering Systems” IEEE Computer, 1993, Vol. January pp. 28-38.

Reddy, S. Y., Fertig, K. W., Smith, D. E. “Constraint Management Methodology for Conceptual Design Tradeoff Studies” Proc. of ASME Design Engineering Technical Conference and Computers in Engineering Conference, 1998, Sacramento, CA.

Seki, K., Ishii, K., Esterman, M., “Robust Design for Dynamic Performance: Optical Pick-Up Example” Proc. of ASME Design Engineering Technical Conference and Computers in Engineering Conference, 1997, Sacramento, CA.

Steele, G. L., “Common Lisp: The Language”, 1990, Digital Press.

Falkenhainer, B., Farquhar, A., Bobrow, D., Fikes, R., Forbus, K., Gruber, T., Iwasaki, Y., Kuipers, B., “CML: A Compositional Modeling Language”, Technical report KSL-94-16, 1994, Knowledge Systems Laboratory, Stanford University.

Farquhar, A., “A Qualitative Physics Compiler”, Proc., Twelfth National Conference on Artificial Intelligence, 1994, Seattle, Washington, The AAAI Press/The MIT Press.

Chaudhri, V. K., Farquhar, A., Fikes, R., Karp, P. D., and Rice, J. P., "Open Knowledge Base Connectivity 2.0", Artificial Intelligence Center, SRI International, 21 July 1997.

Karp, P. D., Myers, K. L., Gruber, T., “The Generic Frame Protocol”, Proc. of the Fourteenth International Joint Conference on Artificial Intelligence, 1995, Montreal, pp768-774.

Genesereth, M. R., Fikes, R. E., “Knowledge Interchange Format, Version 3.0 Reference Manual”, Technical report Logic-92-1, Stanford University Logic Group, 1992.

Iwasaki, Y., Farquhar, A., Fikes, R., and Rice, J., "A web-based compositional modeling systems for sharing of physical knowledge", Proc. of the 15th International Joint Conference on Artificial Intelligence, AAAI Press/The MIT Press, August 1997.

Olsen, G., R., Cutkosky, M., Tenenbaum, J., M., and Gruber, T., R., “Collaborative Engineering Based on Knowledge Sharing Agreement”, CONCURRENT ENGINEERING, June 1995, Vol. 3, No. 2, pp. 145 – 159.

Gruber, T. R. (1993). “Toward principles for the design of ontologies used for knowledge sharing”, International Journal of Human-Computer Studies, special issue on Formal Ontology in Conceptual Analysis and Knowledge Representation, August 1993 (Available as technical report KSL-93-04, Knowledge Systems Laboratory, Stanford University.

Lipsett, R., Schaefer, C., and Ussery, C., "VHDL : Hardware Description and Design", Kluwer Academic Publishers, 1989.

IEEE “Standard Description Language Based on the Verilog(TM) Hardware Description Language”, 1995.