

A Web-Based Compositional Modeling System for Sharing of Physical Knowledge

Yumi Iwasaki, Adam Farquhar, Richard Fikes, James Rice
Knowledge Systems Laboratory, Department of Computer Science
Stanford University
Stanford, CA 94305 USA

Abstract¹

This paper describes a compositional modeling system called CDME (Collaborative Device Modeling Environment) for constructing domain theories of physical systems, composing models of devices, and simulating their behavior. We have implemented the system with the goal of encouraging sharing as well as the collaborative construction of knowledge bases describing physical domains. To maximize the chance of sharing and reuse of knowledge, CDME is implemented as a collection of network services on the World Wide Web. Knowledge is represented at three distinct levels: the physical, ontological, and logical. We describe the levels of representation, and how the system enables knowledge sharing at each level.

1 Introduction

Compositional modeling [1] is an effective method for automatically formulating a behavior model of a complex physical system. In compositional modeling, a system is provided with a knowledge base about the physical world, including knowledge of the behavior of a variety of devices and physical processes. The system formulates a model of a given situation by selecting pieces of knowledge that are applicable and composing them together.

The compositional modeling approach can lower the cost of analyzing physical systems if the cost of constructing the knowledge base can

be amortized over many physical devices whose models can be quickly and easily formulated using the knowledge base. Thus, the advantages of compositional modeling hinge on one's ability to reuse knowledge.

Reuse and sharing is important in the task of representing a sufficient quantity of knowledge to make a viable compositional modeling system. Since compiling a large quantity of knowledge covering a wide variety of types of physical phenomena and devices requires a large amount of resources, one would like to be able to do so collaboratively, allowing a community of users to reuse and share existing knowledge bases or new ones constructed by others. The desire for sharing knowledge has lead some in the qualitative reasoning community to design a modeling language called CML (Compositional Modeling Language) in an attempt to unify similar languages used by different groups [2, 3]. This effort also reflects the awareness in the larger AI community of the need for knowledge reuse and sharing [4].

We have implemented a compositional modeling system called CDME (Collaborative Device Modeling Environment) for compositional modeling with the explicit goal of enabling collaborative construction as well as sharing of knowledge bases. We took two avenues towards facilitating knowledge sharing:

- The knowledge used by the system is represented at three levels: the physical, ontological and logical levels. Knowledge can be shared at each level.
- We made an explicit decision to implement the system as a network service. This allows the system to be used by a large community of users. More importantly, this allows a library of knowledge bases to be constructed collaboratively and in a modular fashion. The knowledge, as well as the result of inferences made by the system, can be shared.

¹This project is supported in part by DARPA and the National Institute for Standards and Technology, Rapid Development, Exploration and Optimization (RaDEO) program, under cooperative agreement 70NANB6H0075.

This paper describes the multiple levels of representation in CDME and how sharing and collaborative construction of a knowledge base is supported at each level.

2 CDME, a Collaborative Device Modeling Environment

CDME automatically formulates the behavior model of a given device and simulates its behavior. To predict the behavior of a physical system in some domain, CDME requires knowledge, encoded in CML, about the physics of the domain. Such a knowledge base, called a *domain theory* in CML consists mainly of *model fragment* definitions, which are descriptions of classes of relevant objects, phenomena and systems. The domain theory of chemical processing plants, for example, might include descriptions of physical phenomena such as mass and heat flows, boiling, evaporation, and condensation; it would also include chemical reactions, the effects of catalysts, and models of components such as reaction vessels, pumps, controllers, and filters. A specific system or situation being modeled is described in a *scenario*. A scenario specifies the objects that exist in a given situation to be analyzed, their relations, and the initial conditions.

Given a scenario, CDME formulates a model by identifying model fragment instances whose operating conditions are satisfied in the scenario, and extracts a mathematical equation model from those instances. The equations are solved to produce a behavioral description. Because the conditions under which the model fragments hold are explicit in the domain theory, the system is able automatically to construct additional behavior models that describe the device as it moves into new operating regions. The functionality of CDME is similar to previous compositional modeling systems [1, 5-8], and the behavior of its direct predecessor is described in detail in [9].

2.1 Multi-layered representation

A modeling system such as CDME requires a rich repository of knowledge about a subject domain in order to be useful. In developing the system as well as its user interface, we often experienced a conflict between the desire to ensure reusability of the knowledge and the need to provide a rich environment in which users can express knowledge efficiently. Using a formal language such as the first-order predicate logic and making sure that all concepts used are logically defined is an important step towards ensuring knowledge can be reused. However, a

formal logical language is not the most convenient language in which to state knowledge. The fine granularity and context-insensitivity of logical representations, the very characteristics that make formal representations suitable as vehicles for knowledge sharing, make such a representation inappropriate to use for stating a large amount of domain-specific knowledge.

To allow the efficient compilation of a large amount of knowledge, a language needs both semantic constraints and syntactic tools, such as those offered by a number of object-oriented representation languages, to help in the efficient capture of the structure of the knowledge. If knowledge is represented in a specialized language, however, the community of people that can use and understand the language is likely to be smaller, and consequently, the ability to share it will be reduced.

Though using a common knowledge representation formalism goes a long way towards enabling sharing, it is not the only issue. Equally important is making one's *ontology* explicit. By ontology, we mean the set of words used in describing a domain along with their definitions. In order for knowledge to be communicated accurately, one must state what the words are and what they mean. The ontology is also a useful body of knowledge to be shared in its own right since the set of concepts and their relations comprising an ontology reveal one's conceptual model of the domain of discourse.

We, therefore, have conflicting requirements: on the one hand, the desire to maximize the chance for reuse and sharing, in different forms and portions, and, on the other, the desire to enable the efficient encoding of a large amount of knowledge. To satisfy these requirements, we implemented CDME with three levels of representation: the physical, the ontological and the logical levels. Each level has its own representation language, which is independently designed to facilitate sharing. We built a web-based user interface for the physical and ontological levels to support encoding of knowledge at the respective level. Table 1 shows the three levels with the representation language, the authoring tool, and the types of objects manipulated. The following sections describe each level in detail.

2.2 Physical Level

The physical level is the level at which knowledge about physical phenomena and objects can be stated by users constructing domain theories. Using CML as the representation language, one defines the classes

Level	Objects manipulated	Language	Authoring tool
Physical	model fragments, entities, quantities	CML	CML Editor
Ontological	concepts concept definitions	Ontolingua	Ontolingua Editor
Logical	logical symbols axioms	KIF	

Table 1: Levels of Representation in CDME

of objects and physical phenomena along with their attributes that must be considered in modeling the domain.

CML is a general declarative modeling language for representing knowledge of physical domains for compositional modeling. CML was designed 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 [5] and include the languages used for the Qualitative Physics Compiler [6, 7], compositional model formulation [1] and the Device Modeling Environment [8]. CML has been formulated by researchers involved in these projects to provide a clean redesign of these languages. CML is described in more detail in [2, 3].

The basic representational unit in CML is the *model fragment*. Each model fragment describes one aspect of a component's behavior or a physical process. Each model fragment is a self-contained assertion whose applicability to a given situation is decided separately to generate the model of an entire situation. Following is an example of a model fragment definition of contained-fluid.

```
(defModelFragment Contained-Stuff
:subclass-of (physical-object)
:participants
  ((sub :type substance)
   (ctnr :type fluid-container))
:conditions
  ((> (amount-of-in sub ctnr) 0))
:quantities
  ((pressure
    :dimension pressure-dimension)
   (mass
    :dimension mass-dimension))
:consequences
  ((= mass (amount-of-in sub ctnr))))
```

To facilitate sharing of CML knowledge bases, CML is fully translatable to the Knowledge Interchange Format (KIF). KIF is the proposed specification for a standard format for first-order predicate logic developed for the purpose of knowledge interchange [10], and

indeed CDME automatically translates CML into KIF. Though one could simply use KIF, CML and the CML Editor in CDME provide more structure than pure KIF to simplify stating knowledge about physical phenomena.

To support collaborative construction of knowledge bases in CML, we have implemented a web-based CML editor as part of CDME. Figure 1 shows some user interactions with the editor. Once a domain theory is constructed and a scenario to be analyzed is given, a user can push the "Compose Model" button to instruct CDME to formulate a model for the scenario, and to simulate the model's behavior. CDME also includes facilities to help comprehension of simulation results, including causal explanations and plots of time-varying variables.

2.3 The ontological level

Having a common knowledge representation format is an important step in enabling communication. Equally important is to establish an explicit vocabulary that will be used in stating the content of knowledge. In our system, users as well as the implementers of CDME can work at the Ontological Level to construct fundamental ontologies for stating knowledge about the world.

An ontology is a formal specification that defines the representational vocabulary in some domain of discourse. In our case, the basic vocabulary of CML, including words such as *model fragment* and words naming parts of its definition (shown in **bold** in the example in Section 2.2), are all defined at the ontological level. General mathematical and physical concepts such as mathematical operators and the dimensions and units of quantities are also defined at this level. In general, concepts such as *systems*, *subsystems*, *components*, and *connections*, which are useful in describing physical domains in general are defined at the ontological level, while more domain-specific concepts such as *isothermal expansion* are defined at the physical level.

To support sharing and construction of ontologies, we have implemented the Ontolingua Editor, which provides a full-distributed

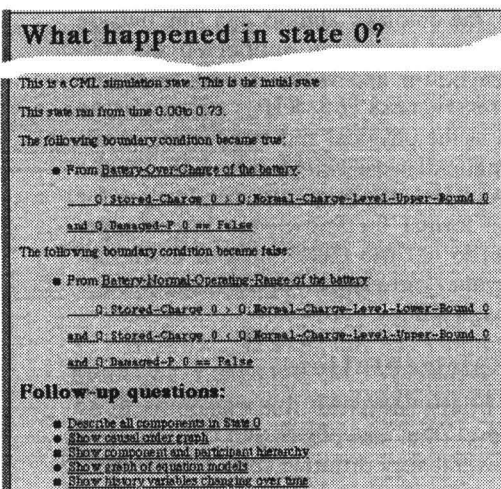
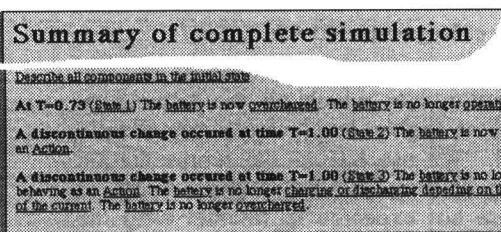
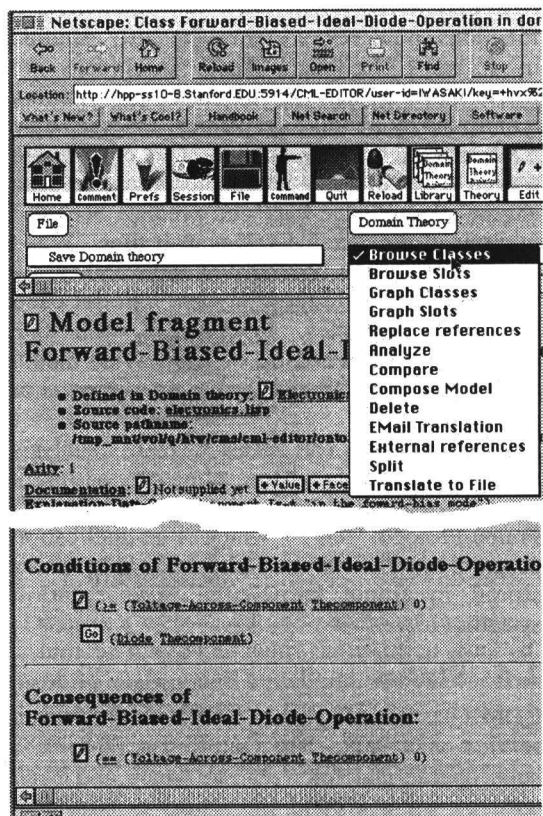


Figure 1: CML Editor: The left page shows a model fragment, *Forward-biased-ideal-diode-operation*. The user interface uses a row of icons for frequently used operations, and pop-up menus with submit buttons as labels. The two pages on the right summarize the simulated behavior and show the state of the device in State-0.

collaborative editing environment for constructing a library of ontologies [11]. To enable distributed, collaborative development of ontologies, the user-interface of the Ontolingua Editor is implemented using off-the-shelf Web-browsers as the interaction medium. Our Ontolingua Editor assists the user in the basic development tasks of browsing, creating, maintaining, sharing, and using ontologies. The Ontolingua Editor also supports modular construction of ontologies by allowing an ontology to include other ontologies. The CML Editor shares these features.

We use an ontology specification language called *Ontolingua* [12] for constructing ontologies. Ontolingua extends KIF by providing a rich set of constructs for modularizing and grouping axioms together. Ontolingua itself was designed to support sharing of knowledge, and a number of translators have been constructed between Ontolingua and various representation languages. While our ontologies are used for knowledge sharing and agent inter-operability [13], they share many properties in common with class libraries, object-oriented specifications, object-oriented

simulation components, database schema definitions, and knowledge-bases.

Ontology construction usually is a distributed and collaborative activity [14], because in many domains no single person has the expertise necessary to construct an ontology for an entire domain. Therefore, it is essential for experts to collaborate on ontology construction. Furthermore, it is important for prospective users of an ontology, who may be geographically distributed, to examine, evaluate, and critique the ontology.

The CML and Ontolingua Editors are unusual among Web services because they allow users to create and edit objects, rather than simply retrieve them. The implemented interface provides a full, distributed collaborative editing environment with over a hundred user commands, context sensitive help, context sensitive user feedback and bug-report collection, multi-level undo/redo, multi-user sessions, and so on.

2.4 The logical level

This is the level of the formal, logical representation into which all the knowledge expressed at higher levels are automatically translated. We use KIF as the representation formalism at this level. Since the users of CDME are not expected to interact with the system at this level, no special purpose authoring tool is provided at this level. However, when necessary, the ontology authors can step outside of the frame language supported by the Ontolingua Editor and write axioms using the full expressiveness of KIF. Knowledge expressed at the physical and ontological levels are automatically translated into KIF by the system. As KIF is the proposed specification for a standard format for knowledge interchange, those who do not use Ontolingua or CML can still share the knowledge in CML knowledge bases through KIF.

2.5 Implementation

We now briefly describe the transformations between levels that take place in CDME. CDME requires several very different sorts of inference. A mixture of symbolic logic and mathematics is required during model formulation. Numeric integration may be used to simulate the systems behavior and monitor for boundary conditions. Logical inference is required again to determine the state of the system after a boundary condition has been reached, so that analysis can continue in the subsequent operating region.

Figure 2 shows a schematic view of the transformations. The left side of the figure illustrates domain theory construction, while the right half illustrates model formulation and simulation. As described above, the ontological knowledge is specified in the Ontolingua language. The physical knowledge is specified in CML, employing terms defined in the ontologies. To unify these representations, CDME translates the CML definitions into Ontolingua. This provides a uniform logical level representation for both the ontological and physical knowledge. This representation is the source for a variety of computationally efficient forms.

To perform model instantiation, the unified Ontolingua representation is translated into a restricted language (REL in the figure) supported by a forward chaining rule engine whose rules are compiled into efficient executable code. Most of the model instantiation work is done by the rule engine. Some inferences, such as

reasoning about inequalities and mathematical functions, however, are not efficiently handled by forward chaining. These inferences are performed by a mixture of backward chaining (controlled by iterative deepening) and a separate mathematical solution engine. This engine also solves the equations extracted from the rule engine's knowledge-base and predicts behavior in terms of the variable values over time. When a model reaches its boundary of applicability, the rule engine produces a new model and simulation continues. In Figure 2, each state contains a representation of a behavior model and the variable values over the time interval of the state.

External agents can access CDME's analysis results, knowledge-base contents, domain theories, and ontologies by using the Generic Frame Protocol (GFP), which is an application program interface for accessing knowledge bases stored in frame knowledge representation systems [15].

2.6 Status

CDME has been implemented and available as a service on WWW. The Ontolingua Editor has been operational for over two years and been used by users around the world to construct ontologies for various purposes. The core part of CDME, the CML Editors, the model formulation and simulation mechanisms as well as some of the explanation facilities, is operational and has been used by us and a limited number of friendly users. The system is not yet robust enough for use by the general public. The devices for which we have used CDME include subsystems of micro-satellites being developed in the Aeronautics and Astronautics Department at Stanford.

Our limited experience with the system shows the beginning of knowledge reuse. Because of the modularity of domain theories, some of the fundamental ontologies (e.g. the ontologies of algebra, component connection) and some domain theories (e.g. kinematics, motors) are reused often. However, as a CML domain theory gets larger, it becomes more difficult to understand its context and implicit assumptions, and thus harder to reuse. This is a general problem with any knowledge base or collection of information. More research is needed to address the issue of making a large body of information more understandable through various types of aggregation, abstraction, and interface tools.

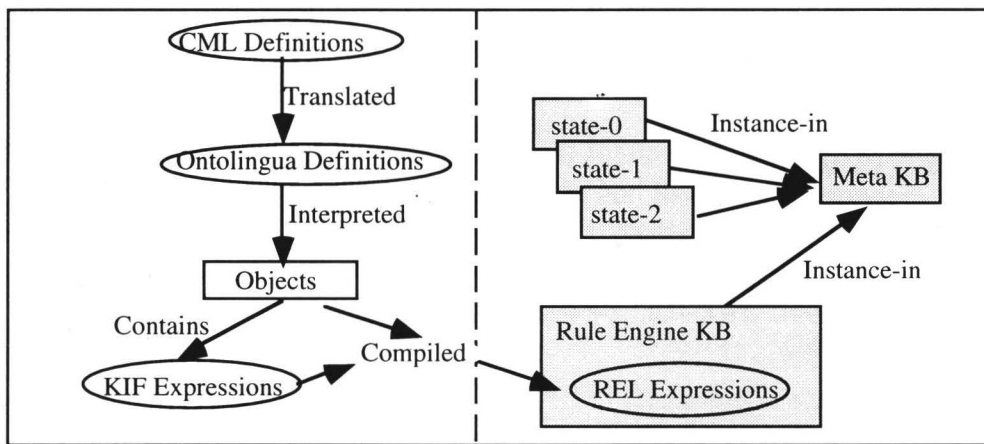


Figure 2: Transformation of knowledge in the system

3 Conclusions

CDME is a web-based compositional modeling system implemented with the explicit goal of enabling sharing and collaborative construction of knowledge bases. The system, which is accessible on the WWW, includes a full, distributed collaborative editing environment to construct CML domain theories. To encourage sharing, knowledge is represented at multiple levels. We have described the different representation levels, and how each level allows different groups of users to share and reuse knowledge in a different form.

Accessibility through the web allows a potentially large community of users not only to reuse but also to contribute to the construction of domain theories. We hope that the availability of CDME and similar services will encourage the development of a significant public library of domain theories by the research community and that it will spur further research and development in the field.

Acknowledgment

The authors would like to thank the past and the present members of the How Things Work project at the Knowledge Systems Laboratory for their assistance in designing and implementing various features of CDME.

References

1. Falkenhainer, B. and Forbus, K., *Compositional modeling: finding the right model for the job*. Artificial Intelligence, 1991. **51**(1-3).
2. Falkenhainer, B., et al. *CML: A Compositional Modeling Language*. 1994.

Technical report KSL-94-16, Knowledge Systems Laboratory, Stanford University.

3. Bobrow, D., et al. A Compositional Modeling Language. In *Proceedings of The Tenth International Workshop on Qualitative Reasoning*. 1996. Fallenleaf Lake, CA. AAAI Press.
4. Neches, R., et al., *Enabling Technology for Knowledge Sharing*, in *AI Magazine*. 1991, p. 36-56.
5. Forbus, K.D., *Qualitative Process Theory*. Artificial Intelligence, 1984. **24**(1-3).
6. Crawford, J., Farquhar, A., and Kuipers, B. QPC: A Compiler from Physical Models into Qualitative Differential Equations. In *Proceedings of The Eighth National Conference on Artificial Intelligence*. 1990.
7. Farquhar, A. A Qualitative Physics Compiler. In *Proceedings of Proceedings, Twelfth National Conference on Artificial Intelligence*. 1994. Seattle, Washington. The AAAI Press/The MIT Press.
8. Iwasaki, Y. and Low, C.M. Device Modeling Environment: An Integrated Model-Formulation and Simulation Environment for Continuous and Discrete Phenomena. In *Proceedings of Conference on Intelligent Systems Engineering*. 1992. Edinburgh, Scotland.
9. Low, C.M. and Iwasaki, Y., *Device modelling environment: an interactive environment for modelling device behaviour*. Intelligent Systems Engineering, 1993. **1**(2): p. 115-145.

10. Genesereth, M.R. and Fikes, R.E. *Knowledge Interchange Format, Version 3.0 Reference Manual*. 1992. Technical report Logic-92-1, Stanford University Logic Group.
11. Rice, J., *et al.* Using the Web Instead of a Window System. In *Proceedings of CHI 96*. 1996. ACM Press.
12. Gruber, T.R. *Ontolingua: A Mechanism to Support Portable Ontologies*. 1991. Technical report KSL 91-66, Knowledge Systems Laboratory, Stanford University.
13. Fikes, R., *et al.* *Knowledge Sharing Project Overview*. 1991. Technical report KSL-91-71, Knowledge Systems Laboratory, Department of Computer Science, Stanford University.
14. Grudin, J., *Groupware and Cooperative Work: Problems and Prospects*, in *The art of human-computer interface design*, B. Laurel, Editor. 1990, Addison-Wesley: Reading, Massachusetts. p. 171-185.
15. Karp, P.D., Myers, K.L., and Gruber, T. The Generic Frame Protocol. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*. 1995.