

An Application of the Knowledge Intensive Engineering Framework to Building Foundation Design

Masaharu Yoshioka¹, Yasuhiro Shamoto², and Tetsuo Tomiyama³

1. *Research and Development Department, National Center for Science Information Systems*

2. *Shimizu Corporation.*

3. *Research into Artifacts, Center for Engineering, The University of Tokyo.*

Key words: Design Object Model Integration, Computer Aided Design, Building Foundation Design

Abstract: Because of increasing concerns about environmental problems and other economic and global issues, engineers are requested to evaluate their products from a wide variety of aspects. To support this, we proposed the concept of knowledge intensive engineering, in which various kinds of knowledge are used in a flexible and integrated manner aiming at generation of more added-value. In addition, we also proposed the Knowledge Intensive Engineering Framework (KIEF) that forms a computational framework of knowledge intensive engineering. In this paper, we describe the concept of knowledge intensive engineering and KIEF. Then, we apply KIEF to building foundation design to demonstrate the power of the system. To do so, we analyze knowledge that is used for building foundation design and discuss how to implement the knowledge on KIEF. Finally, we show some results of this application and discuss the capability of KIEF.

1. INTRODUCTION

During design of a new product, engineers evaluate it from various aspects, supported by computational tools with which a model represents the product. Since these tools represent and deal with an identical product, a change of the product, based on the evaluation results from one aspect,

affects other models. To deal with these dependencies among models, it is necessary to have a framework that can integrate these tools. Speaking of mechatronics design, for instance, design models, such as a geometric model, a kinematics model, an electric circuit analysis model, and a finite element model, are required to be integrated.

We have proposed knowledge intensive engineering that is a new style of engineering to assist engineering activities in product life cycle stages based on intensive use of various kinds of engineering knowledge (Tomiyama *et al.*, 1996). The Knowledge Intensive Engineering Framework (KIEF) system is a computational framework to support knowledge intensive engineering by integrating computational modeling systems for each activity.

This paper is intended to demonstrate the flexibility of the KIEF system by describing applications to a new engineering domain that is not an original target domain of the system. In Section 2, we briefly review our previous research about the knowledge base (Ishii *et al.*, 1995), (Sekiya and Tomiyama, 1997) and the pluggable metamodel mechanism (Yoshioka and Tomiyama, 1997) that are two basic elements of KIEF. In addition, we describe the architecture of KIEF. In Section 3, we argue that KIEF can be flexibly applied to designs in other domains than mechanical design for which KIEF was originally developed. To evaluate this idea, we demonstrate the result of applying KIEF to building foundation design domain as a case study and identify research issues for this application. Section 4 compares our approach with related work and Section 5 concludes this paper.

2. THE KNOWLEDGE INTENSIVE ENGINEERING FRAMEWORK

Engineering activities in various life cycle stages generate and modify information about the product. In KIEF, we formalize these activities as operations to computational models. This system supports engineers by integrating these computational models and managing their dependency and consistency.

KIEF has two basic elements (Figure 1).

- A very large scale knowledge base (VLKB):
It gives ontological definitions of engineering knowledge that are used in various modeling systems.
- A model management mechanism called the pluggable metamodel mechanism:
It provides the data sharing capability among various modeling systems and keeps dependency and consistency among them.

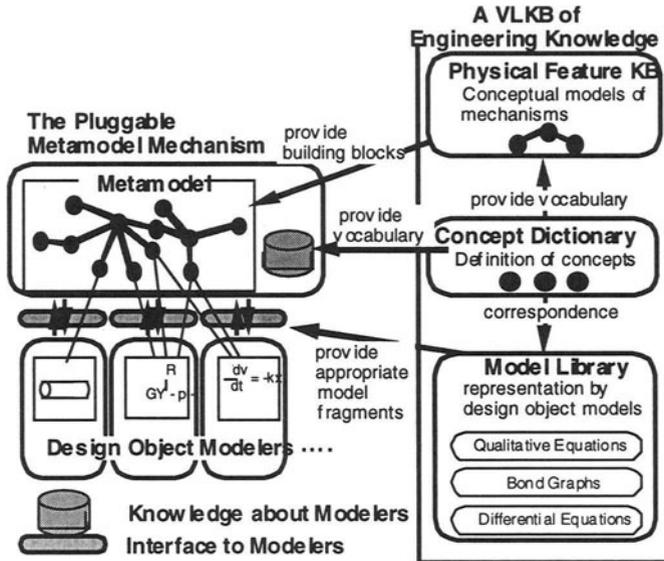


Figure 1 Knowledge Intensive Engineering Framework (KIEF) System

2.1 The Very Large Scale Knowledge Base

We have been working for developing an engineering knowledge base based on the concept of qualitative physics base (Ishii *et al.*, 1995), (Sekiya and Tomiyama, 1997). This knowledge base system has following three components.

– Concept Dictionary

This dictionary gives ontological definitions of concepts used throughout the system. We classify these concepts into five categories: entity, relation, physical phenomenon, attribute, and physical rule.

– Physical Feature Library

A physical feature is a piece of knowledge that represents a mechanism and related physical phenomena. This knowledge is used for two purposes. One is a building block to build a design object model, and the other is knowledge to reason out possible physical phenomena occurring on design object models. This is philosophically the same idea as embodiment in German systematic methodologies (VDI, 1977), (Hubka 1973). The difference is that this knowledge is a computational tool rather than just a theory or a textbook.

– Model Library

This is a library of model fragments that is used for a particular modeling system. Model fragments are associated with a physical phenomenon, attribute, and so on.

In VLKB, the concept dictionary gives vocabularies used to describe knowledge in the entire system. We define physical features with combination of concepts that are defined in the concept dictionary and we also define the model library in association with the concepts in the concept dictionary. In KIEF, the user builds a design object model by selecting concepts from the concept dictionary and defines the relations among them. To support this modeling process, a physical feature is used as a building block. The model library is used for mapping a model to a particular modeling system. Figure 2 shows an example of physical features, concepts in the concept dictionary, and the model library in the system.

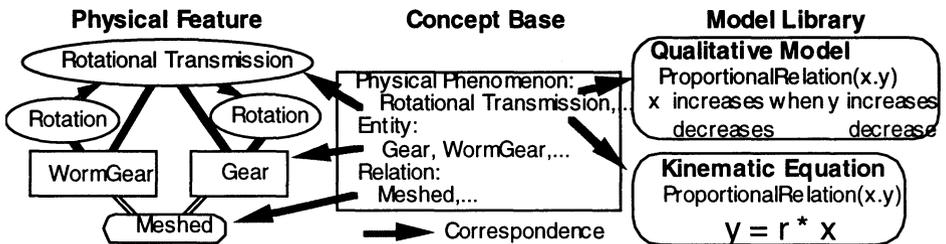


Figure 2. Examples of the Physical Feature, the Concept Dictionary, and the Model Library

When building a model for a particular modeling system, the system collects all related models from the model library. However, the assembling process of these fragments is not so easy in some cases. We also conduct a research to support this model building process (Sekiya *et al*, 1999).

2.2 The Pluggable Metamodel Mechanism

The pluggable metamodel mechanism is a computational system that integrates multiple computational modeling tools (called modelers in this paper) by using a metamodel and knowledge about each modeler. A metamodel represents relationships among physical concepts used for various design object models about an identical design object. VLKB gives definitions of the physical concepts and the relationships are described as links in the pluggable metamodel mechanism. The pluggable metamodel mechanism handles the following links.

- Causal link
This link represents a causal relationship among physical phenomena.
- Occurrence link
This link represents the relationship between a physical phenomenon and an entity.
- Attribute link
This link represents the relationship between an attribute and an entity, a relation, or a physical phenomenon.

- Relation link

This link represents the relationship between a relation and an entity. We use relation concepts associated with relation links to represent relationships among entities.

- Reason-from link

This link represents the relationship between an attribute and other concepts that are used to reason out the attribute.

The pluggable metamodel mechanism maintains relationships and consistencies among these design object models and exchanges the information among them. It integrates existing modelers by defining the data exchanging interface and the knowledge about each modeler (Table 1).

Table 1. Knowledge about a Modeler

Name	Name of the modeler
Related concepts	Concepts that are relevant to the modeler
Usable concepts	Concepts that are used for making a model on it
Available concepts	Concepts that can be reasoned out by using it
Attribute Translation Method	Attribute relationship graph and translating method

The pluggable metamodel mechanism assists to build a model for each modeler based on the information provided by metamodel and information in other modelers. A modeling process with the pluggable metamodel mechanism consists of two steps.

1. Construction of a conceptual model

- a) Abstraction of a model

Abstraction of a model is a process to map concepts in the metamodel to concepts used in a particular modeler. The pluggable metamodel mechanism first selects related concepts from the metamodel by using the related concepts definition in the knowledge about the modeler. Second, the user maps the selected concepts onto the concepts used in the modeler. For example, considering an analysis of distortion of a robot arm. The pluggable metamodel mechanism selects concepts such as force, robot arm, and connection in the first step. In the second step, designer maps the robot arm concept to a beam that is a usable concept of a distortion analysis tool.

- b) Simplification of a model

Simplification of a model is a process to neglect some concepts of a model, considering the computational cost, computability, and so on. This operation is realized as selection of concepts that can be neglected. In the example of robot arm analysis, the following operations correspond to this operation; neglectation of self-weight of

the robot arm when it is light, body of the robot if the user concentrates on the analysis of the robot arm.

2. Building a particular model

In this process, the pluggable metamodel mechanism collects the necessary information by using the attribute translation method in the knowledge about the modeler. If the mechanism cannot find the information in the metamodel, the mechanism finds other modelers that can handle this information or asks the user to input the data. In this process, the mechanism tries to automatically collect as much information as possible, but there could be some that cannot be handled automatically. In the example of the robot arm, to calculate the length of beam, the metamodel mechanism asks the user to use a solid modeler or 2D CAD to input the data. In addition, it asks to input the location of the central axis of the beam to feed the information to the beam modeler.

The pluggable metamodel mechanism stores any information to build a model. This information, as a whole, is called modeling assumption and is connected to the resulting information with “Reason-from links”. This information can be useful for checking the reliability of the computational result in later stages, and for maintaining the consistency.

2.3 The KIEF System

The KIEF system uses the pluggable metamodel mechanism as a core modeling system and VLKB as a source of knowledge. Building and evaluation of a product on KIEF proceeds as follows.

1. Create an initial metamodel by combining physical features.

The user combines physical features to build a metamodel. The pluggable metamodel mechanism models the structure of the product and physical phenomena to achieve the functionality of the product. If the design problem is a typical routine design case, he/she just combines physical features to represent the design object. If the design problem is a new one, the FBS modeler (Umeda *et al.*, 1996) supports to build an initial model by functional decomposition and to map subfunctions to physical features.

2. Reason out possible physical phenomena that may occur on the product.

We use a physical feature as knowledge that represents causal relationships between the structure of the product and the physical phenomena. A qualitative physics based reasoning system of KIEF reasons out possible physical phenomena with this knowledge.

3. Evaluate the product with other modelers.

The user evaluates the product from various aspects with modelers that deal with these aspects. KIEF also maintains the consistency of information by checking whether the modeling assumptions for a model are modified or not, every time the model is modified.

3. APPLICATION TO BUILDING FOUNDATION DESIGN

KIEF was originally developed for mechanical engineering design. However, since the system has a flexible framework to integrate different kinds of knowledge, we can apply KIEF to other domain by changing the knowledge base of KIEF.

To validate this idea, we apply the KIEF system to building foundation design domain. In this section, we describe an analysis of building foundation design and discuss a strategy to construct the knowledge base. Since we have not applied KIEF to larger scale problems, this discussion includes the scaling up problem of the system. Then, we illustrate an example of building foundation design with KIEF.

3.1 Knowledge for Building Foundation Design

In this research, we interviewed a researcher of a construction company to acquire knowledge about building foundation design.

Architectural Institute of Japan compiled a textbook that gives guidelines to design foundations of building (Architectural Institute of Japan, 1988). From the interview of the researcher, this type of textbook turned out to have the following problems.

- Knowledge is often a black box.
Descriptions of each analysis method describe only how to use it without giving theoretical basis. This may cause inappropriate use of analysis methods.
- It is intricate to select necessary analysis methods.
This textbook has a flowchart for selecting necessary analysis methods. However, in a real design process, the designers do not need to consider all of the considerations described in the flowchart.
- Information for selecting an appropriate analysis method is incomplete.
This textbook describes various analysis methods used in the building foundation design process. In many cases, two or more analysis methods are listed for one problem, but there is not enough information how to select an appropriate analysis method from them.

To solve these problems, it is necessary to consider the following issues.

- Describe causal relationships of each analysis method.
Since each analysis method is knowledge for evaluating some physical phenomena, and there exist causal relationships among these phenomena, we can describe the relationships among analysis methods with the causal relationship among physical phenomena. KIEF has a capability of describing them with the metamodel and physical features.
- Select necessary analysis methods based on an object model.
The pluggable metamodel mechanism can reason out possible physical phenomena that may occur on a product. In order to select an appropriate analysis method for possible physical phenomena, we should describe knowledge about relationships between analysis methods and physical phenomena.
- Suggest an appropriate analysis method when two or more methods are applicable to one problem.
It is necessary to have a scheme to describe features of each analysis method for selecting methods. Therefore, the knowledge about the relationships between possible physical phenomena and analysis methods should have a description for selecting an appropriate analysis method.

In addition, these analysis methods are described on different granularity levels; e.g., a building can be modeled as one mass or combination of floors, walls, and so on. Therefore, KIEF is requested to manage the consistency among different granularity levels.

3.2 A Building Foundation Design Support System on the KIEF system

Based on the discussion about the knowledge for building foundation design, we extend KIEF to deal with knowledge for selecting appropriate analysis methods and granularity modeling.

For dealing with different granularity modeling, we developed a mechanism to manage multiple metamodels for different granularity levels. In this metamodel, the designer defines relationships between one entity and its subcomponent model with the part-unit editor. Figure 2 shows an example of mapping different granularity levels. In this case, "Ground" is decomposed into "SoftClayStrata", "SoftSandStrata", and "StiffClayStrata". In addition, the designer also defines the relationships between "Ground" and the other entities in the decomposed level. By using these relationships,

the system propagates physical phenomena occurred in the decomposed level to the "Ground" level.

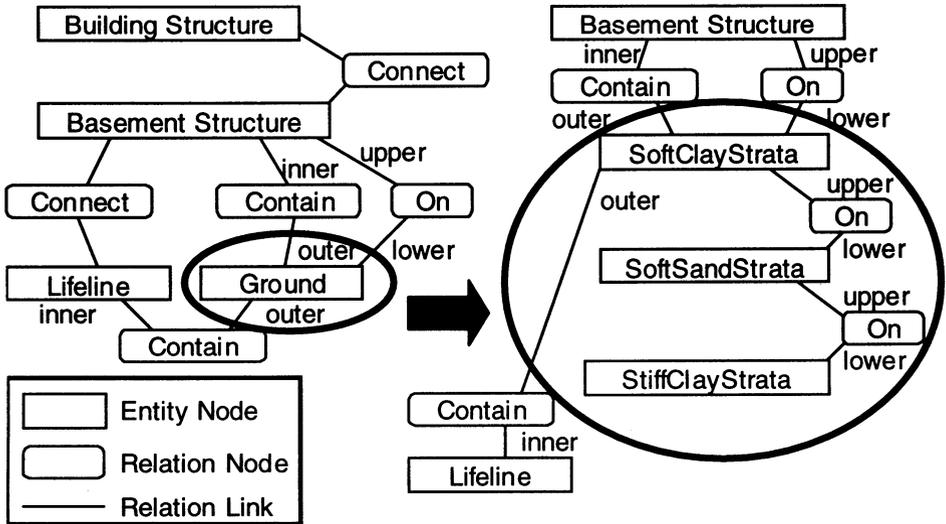


Figure 3. Relationships between Different Granularity Models

This granularity handling is also useful for other engineering domains including mechanical design. Therefore, we think this is an expansion not only for building foundation design but also of KIEF itself. In addition to this, it is better to give a guideline to make these different granularity models. However, this guideline requires understanding of all models that are used for the design and it is very difficult to make it. So we make these models based on the example at this moment.

For supporting selection of an appropriate analysis method from two or more methods, we define knowledge about analysis methods. Table 2 shows the scheme of the knowledge and Table 3 and 4 show an example of the definition for earthquake. Selection of an appropriate analysis method proceeds as follows.

1. Retrieve candidate analysis methods and check availability of the input attributes.
 KIEF retrieves candidate analysis methods by using knowledge about analysis methods. Then, the system checks whether or not information about each input attribute exists in the metamodel or not.
2. Select appropriate methods by the user.
 The user selects an appropriate method by using the descriptions about analysis methods and the input attribute information.

3. Suggest another analysis method to obtain input attribute information. If there is no input attribute information in the model, KIEF suggests another analysis method to obtain the information. This is done by comparing the concept of the information with the output attributes definitions in the knowledge about analysis methods.

Table 2 Knowledge about Analysis Methods

Phenomenon name	Name of physical phenomenon	
Analysis method (one or more)	Description	Description about the method
	Input attributes	Description about required attributes for the method
	Output attributes	Description about the result

Table 3 Example Definition for Earthquake (Simple)

Phenomenon name	Earthquake	
Analysis method SimpleMethod	Description	Use default variable This method can be used for simple design. Set Two levels for the earthquake. Maximum acceleration on the ground: Level 1: 250 Gal, Level 2: 500 Gal Maximum velocity on the ground: Level 1: 25cm/s, Level 2: 50cm/s
	Input attributes	None
	Output attributes	Maximum velocity on the ground Maximum acceleration on the ground,

Table 4 Example Definition for Earthquake (Use wave data)

Phenomenon name	Earthquake	
Analysis method WaveDataRetrieve	Description	Use earthquake wave data for design Use appropriate earthquake wave data for the place. This method is used for complex design
	Input attributes	Place
	Output attributes	Earthquake wave data(transition data of acceleration and velocity of the structure and the ground)

3.3 Example of Building Foundation Design

We demonstrate how KIEF works for an example of building foundation design. Before starting this design, we define concepts used in the building foundation design in the concept dictionary, and also define physical features to represent causal dependency knowledge of physical phenomena. Since mechanical engineering domain is so different from the building foundation design, we cannot reuse the knowledge, even though the knowledge is stored in the same knowledge base.

First, the designer selects an initial model of a typical building from the physical feature library (Figure 4). In order to evaluate a particular design case, he/she adds detail information to Ground with the part-unit editor (Figure 5).

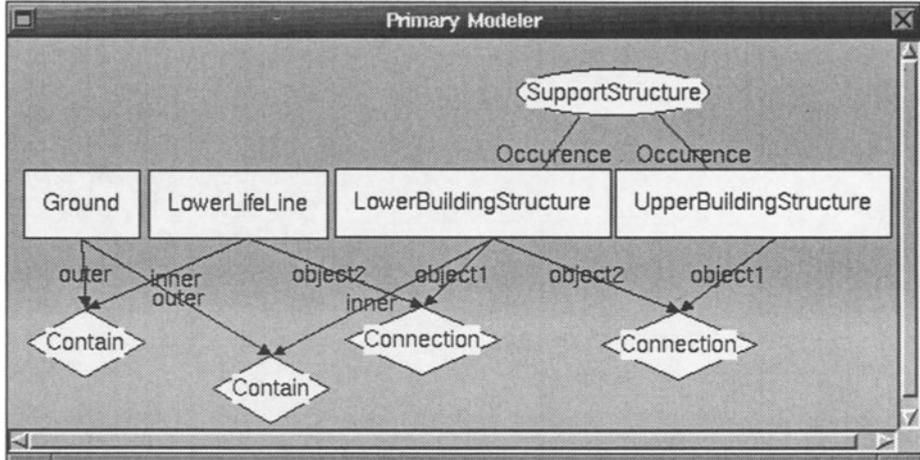


Figure 4. Initial Design Model

After building a decomposed model of Ground, KIEF reasons out the possible physical phenomena that may occur on this structure. In this case, KIEF reasons out earthquake, quick sand, liquefaction settlement, and so on. After reasoning out possible physical phenomena, the system suggests the necessary analysis methods for these phenomena (Figure 6) and evaluates whether or not the problems are critical.

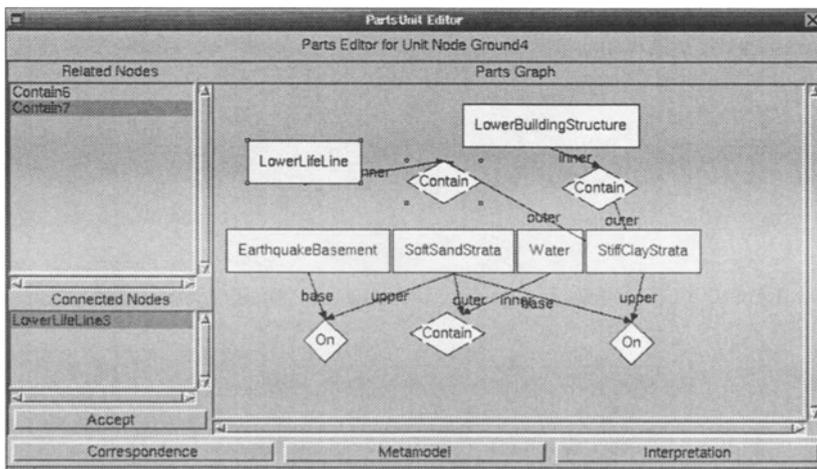


Figure 5. Decomposed Ground Unit

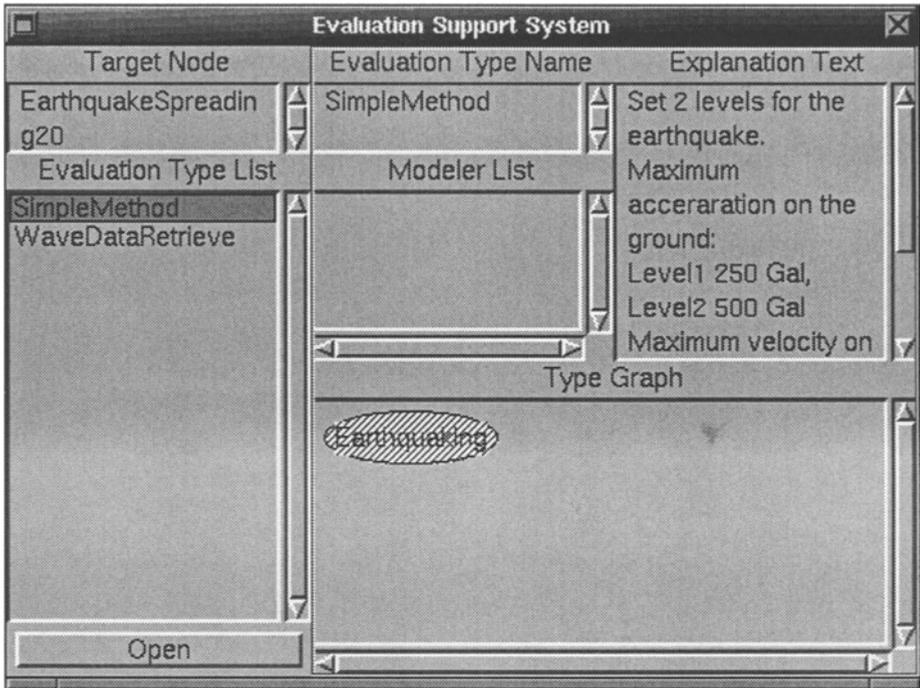


Figure 6. Suggestion of Evaluation Method

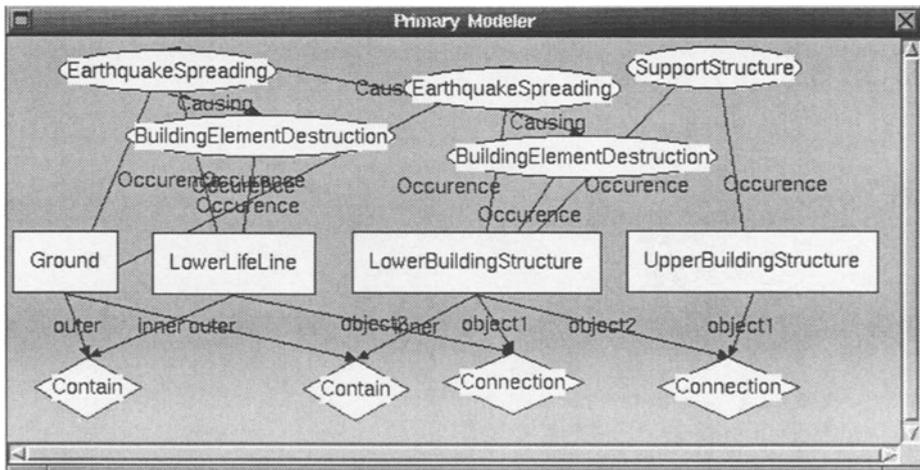


Figure 7. Physical Phenomena Occurred on the Top Level

When the phenomena are negligible, the designer specifies these phenomena to be neglected and propagates this decision to the unit level (Ground level). Figure 7 shows a propagated result. In this case, since quick sand is a negligible phenomenon, other phenomena caused by quick sand such as liquefaction settlement are also neglected. On the other hand, since

earthquake is not negligible one, parts level physical phenomena such as EarthquakeSpreading and BuildingElementDestruction are propagated to the Ground level.

Next, the designer moves on to the design of the lower building structure and then to the upper building structure, and repeats evaluation for this building foundation in the same way.

3.4 Discussion

From this example, we conclude that KIEF has a capability of describing the knowledge of a typical engineering textbook (Architectural Institute of Japan, 1988). KIEF describes the causal relationships among physical phenomena and the relationships between physical phenomena and analysis methods.

Speaking of the analysis methods, most of them are just plain texts from the textbook. However, if we implement these analysis methods as computational tools, we can integrate them with the pluggable metamodel mechanism of KIEF. From these, we conclude that KIEF can be used as a kernel for an intelligent CAD system for building foundation design.

In this experiment, we only share the reasoning system and VLKB from the original KIEF domain and knowledge about physical phenomena is not reused for this design. This problem comes from the difference between mechanical engineering design and building foundation design. In order to discuss this cross-domain design issue, we need to try many other application domains.

From the viewpoint of the scaling up of KIEF, we have to deal with complex design object models. For managing this type of complexity, the building foundation designers set multiple granularity level for one design object and manage the relationship among them. This type of knowledge management can be seen in other engineering domains, too. Therefore, we believe that granularity handling mechanism of KIEF is also useful not only for building foundation design but also for other engineering design domains.

4. RELATED WORK

STEP (TC184/SC4, 1994) is a standard for sharing and exchanging product model data. STEP also tries to include different engineering domains such as mechanical design and ship design. However, STEP only focuses on data structure for representing products and does not explicitly mention knowledge used in these application domains.

In the SHADE project (McGuire *et al.*, 1993), they also propose framework to integrate various design tools. This approach is similar to our approach in that integrating models takes place through translating their knowledge in one format. While they use KIF (Knowledge Interchange Format) (Genesereth, 1992) as the format and Ontolingua (Gruber, 1992) to define ontology, we use the concept dictionary as the most fundamental description. However, since they try to integrate these tools automatically with KQML (Knowledge Query and Manipulation Language) (T. Finin *et al.*, 1992), they do not address the interaction between designers and the system during design. Therefore, the system cannot deal with knowledge for process management (e.g., selection of appropriate methods, supporting modeling process, etc.), although an intelligent agent oriented architecture is assumed.

SEED (Akin *et al.*, 1998) is an integrated application for building design. This system has an advantage of good integrity due to their particular modeling format. However, it has a difficulty to expand a system with existing computational tools. On the other hand, the main feature of KIEF is a flexibility to integrate new tools.

5. CONCLUSION

This paper described the architecture of KIEF that can be used as a kernel of intelligent CAD. In order to verify this concept, we applied the system to building foundation design. In this application, knowledge about causal relationships is described in the system and analysis methods are defined with regard to relationships among physical phenomena. Finally, we showed some results of the application and confirmed this framework can deal with most of the knowledge in the design of building foundations.

ACKNOWLEDGMENT

The work described in Section 3 was part of an international collaborative research project, GNOSIS within the Intelligent Manufacturing Systems (IMS) research program.

REFERENCES

- Akin, Ö., Aygen, Z., Chang, T.-W., Chien, S.-F., Choi B., Donia, M., Fenves, S., Flemming, U., Garrett, J., Gomez, N., Kiliççöte, H., Rivard, H., Sen, R., Snyder, J., Tsai, W.-J., Woodbury, R., and Zhang, Y. (1998) *A software environment to support early phases in building design*. International Journal of Design Computing (<http://www.arch.usyd.edu.au/kcdc/journal/>), Vol. 1.
- Architectural Institute of Japan, editor. (1988) *Recommendations for Design of Building Foundations*. Architectural Institute of Japan.
- Finin, T., Weber, J., Wiederhold, G., Genesereth, M., Fritzson, R., Mckay, D., McGuire, J., Pelavin, R., Shapiro, S., and Beck, G. (1992) *Specification of the KQML agent-communication language*. Technical report EIT TR 92-04, Enereprise Integration Technologies, Palo Alto, California.
- Genesereth, M. R. (1992) *Knowledge interchange format*. In James Allen, Richard Fikes, and Erik Sandwall, editors, *Proceedings of the Conference of the Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers.
- Gruber, T.R. (1992) *Ontolingua: A mechanism to support portable ontologies*. Technical report KSL91-66, Knowledge Systems Laboratory, Stanford University, Stanford.
- Hubka, V. (1973), *Theorie der Maschinensysteme*, Springer-Verlag, Berlin.
- Ishii, M., Sekiya, T., and Tomiyama, T. (1995) *A very large-scale knowledge base for the knowledge intensive engineering framework*. In KB&KS'95, the Second International Conference on Building and Sharing of Very Large-Scale Knowledge Bases, pp. 123-131.
- McGuire, J. G., Kuokka, D. R., Weber, J. C., Tenenbaum, J. M., Gruber, T. R., and Olsen, G. R. (1993) *SHADE: Technology for knowledge-based collaborative engineering*. Concurrent Engineering: Research and Applications (CERA), Vol. 1, No. 2.
- ISO TC184/SC4. (1994) *ISO 10303-1 Industrial Automation Systems and Integration - Product Data Representation and Exchange-, Part1: Overview and Fundamental Principles*.
- Sekiya, T., and Tomiyama, T. (1997) *Case studies of ontology for the knowledge intensive engineering framework*, In Mäntylä, M., Finger, S., and Tomiyama T. editors, *Knowledge Intensive CAD*, Vol. 2, pp. 139-156, Chapman & Hall.
- Sekiya T., Tsumaya, A., and Tomiyama, T. (1999) *Classification of Knowledge for Generating Engineering Models - A Case study of Model Generation in Finite Element Analysis -*. In Finger, S., Tomiyama, T., and Mäntylä, M. editors, *Knowledge Intensive CAD-3*, Preprints of the first IFIP WG 5.2 Workshop on Knowledge Intensive CAD-3, Kluwer Academic Publishers, 1999.
- Tomiyama, T., Umeda, Y., Ishii, M., Yoshioka, M., and Kiriya, T. (1996) *Knowledge systematization for a knowledge intensive engineering framework*. In Tomiyama, T., Mäntylä, M., and Finger S. editors, *Knowledge Intensive CAD-1*, Preprints of the first IFIP WG 5.2 Workshop on Knowledge Intensive CAD-1, pp. 33-52, Chapman & Hall.
- Umeda, Y., Ishii, M., Yoshioka, M., and Tomiyama, T. (1996) *Supporting conceptual design based on the Function-Behavior-State modeler*. Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM), Vol.10, No. 4, pp. 275-288.
- VDI-Gesellschaft. (1977) *VDI-2222 Konstruktion und Entwicklung*, VDI-Verlag, Düsseldorf.
- Yoshioka, M., Sekiya, T., and Tomiyama, T. (1998) *Design knowledge collection by modelling*. In Jacucci, G., Olling, G. J., Preiss, K., and Wozny, M. editors, *GLOBALIZATION OF MANUFACTURING IN THE DIGITAL COMMUNICATIONS ERA OF THE 21ST CENTURY* Innovation, Agility and the Virtual Enterprise, pp. 287-298. Kluwer Academic Publishers.

Yoshioka M., and Tomiyama, T. (1997) Pluggable metamodel mechanism: A framework of an integrated design object modelling environment. In Bradshaw, A., and Counsell, J. editors, *Computer Aided Conceptual Design '97*, Proceedings of the 1997 Lancaster International Workshop on Engineering Design CACD'97, pp. 57-70, Lancaster University.