An Application of the Knowledge Intensive Engineering Framework to Architectural Design

Masaharu Yoshioka
Research and Development Department, National Center for Science Information Systems, Japan
Otsuka 3-29-1, Bunkyo-ku, Tokyo, 112-8640, Japan
Telephone: +81-3-3942-7337. Fax: +81-3-5395-7064
e-mail: `yoshioka@rd.nacsis.ac.jp`

Yasuhiro Shamoto
Shimizu Corporation
Shibaura 1-2-3, Minato-ku, Tokyo, 105-8007, Japan.
Telephone: +81-3-5441-1111 Fax: +81-3-5441-0361
e-mail: `shamoto@sit.shimz.co.jp`

Tetsuo Tomiyama
Research into Artifacts Center for Engineering, The University of Tokyo
Komaba 4-6-1, Meguro-ku, Tokyo, 153-8904, Japan.
Telephone: +81-3-5453-5888 Fax: +81-3-3467-0648
e-mail: `tomiyama@race.u-tokyo.ac.jp`

Abstract

Because of increasing concerns about environmental problems and other issues, engineers are requested to evaluate their products from ever 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) system 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 architectural design to demonstrate the power of the system. To do so, we analyze knowledge that is used in architecture design and discuss how to implement the knowledge on KIEF. Finally, we show some results of this application and discuss the capability of KIEF.

Keywords

Design, Model integration, Knowledge base system, Computer Aided Design, Architectural design

1 INTRODUCTION

Engineers design and evaluate products from various aspects with computational tools. Since these tools deal with the same product, a change of the product, based on evaluation results from one aspect, affects other aspects. To deal with these dependencies among tools, it is necessary to have a framework to integrate various aspects. Speaking of mechatoronics design, design models such as a geometric model, kinematics model, electric circuit analysis model, 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 various product life cycle stages based on intensive use of various kinds of engineering knowledge (Tomiyama *et al.,* 1996). The Knowledge Intensive Engineering Framework (KIEF) is a computational framework to support this engineering process by integrating computational modeling system for each activity.

This paper describes the flexibility of KIEF by applying 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 KIEF. In Section 3, we show this framework can be used as a kernel of intelligent CAD in various domains, even though this system is originally developed for mechanical engineering domain. To evaluate this idea, we demonstrate the result of applying KIEF to architectural 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

Since knowledge intensive engineering deals with various engineering activities over a product life cycle, KIEF requests to handle large scale knowledge base. However, when the amount of knowledge stored in the knowledge base increases, it is very difficult to maintain the consistency among the knowledge. From the discussion about the frame problem, we think that the most difficult problem is exceptional case handling.

Therefore, we separate the ontological definitions of concepts and the definitions of causal relationships among concepts for the knowledge base of the KIEF system (Yoshioka *et al.,* 1998). We compartmentalize the large scale knowledge base to moderate sized knowledge bases for particular domains and maintain the relationships among the knowledge bases by using ontological definitions. When a conflict happens among different knowledge bases, the system detects this problem and asks the user to solve the conflict.

KIEF employs a concept dictionary (Ishii *et al.,* 1995) as an ontological knowledge base. The concept dictionary is a knowledge base about physical concepts such as entities, relations, and physical phenomena. For integrating compartmentalized knowledge, we use the pluggable metamodel mechanism (Yoshioka and Tomiyama, 1997) to integrate multiple design object modelers (Figure1). In the pluggable metamodel mechanism, a design object is represented by a network of physical concepts defined in the concept base. The mechanism uses this concept network, called a metamodel, for maintaining the relationships among the concepts used in the modelers. Therefore, we can use these modelers as compartmentalized knowledge base systems and maintain the consistency by using the pluggable metamodel mechanism.

In addition to that, we support building conceptual network models, with physical features that are knowledge about mechanisms and physical phenomena related to the mechanisms.

Building and evaluation of a product in KIEF proceeds as follows.

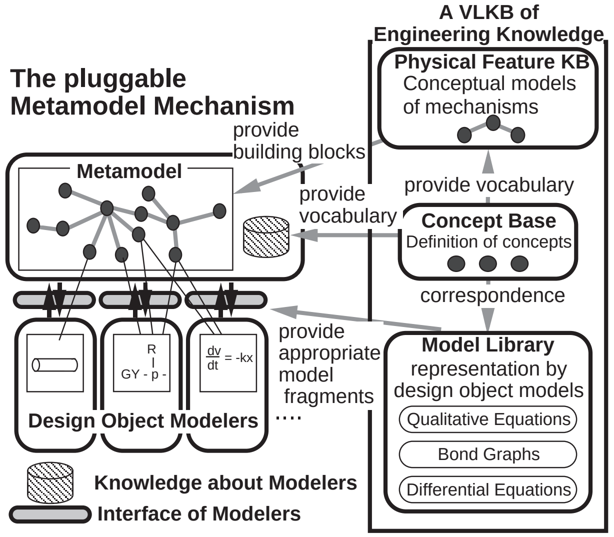


Figure 1 Knowledge Intensive Engineering Framework

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.
2. Reason out possible physical phenomena that may occur on the product.
   We use physical feature as knowledge that represents causal relationships between the structure of the product and the physical phenomena. KIEF reasons out possible physical phenomena with this knowledge.
3. Evaluate with other modelers.
   The user generates and evaluates the product with other modelers. KIEF assists the modeling process and maintains the consistency among them.

3 APPLICATION TO ARCHITECTURAL DESIGN

KIEF was originally developed for mechanical engineering design. However, since KIEF has a flexible framework to integrate different knowledge bases, we can extend the application domain by changing the knowledge base of KIEF to serve as a kernel of intelligent CAD in that domains.

To validate this concept, we apply KIEF to an architectural design domain that is different from mechanical engineering design. In this section, we describe an analysis of the architectural design domain and discuss a strategy to construct the knowledge base. Since we have not applied KIEF to large scale problems, this discussion includes the scaling up problem of KIEF. Then, we illustrate an example of architectural design with KIEF.

3.1 Knowledge in Architectural Design

In this research, we interviewed a researcher of a construction company to acquire knowledge about architectural design. Since the main research area of the researcher was foundations of buildings, most of the knowledge collected from him was knowledge for designing building foundations.

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

1. 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.
2. 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.
3. 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.

1. 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.
2. 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.
3. 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 An Architectural Design Support System on KIEF

Based on the discussion about the knowledge for architectural 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 “SoftCrayStrata”, “SoftSandStrata”, and “StiffCrayStrata”. 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.

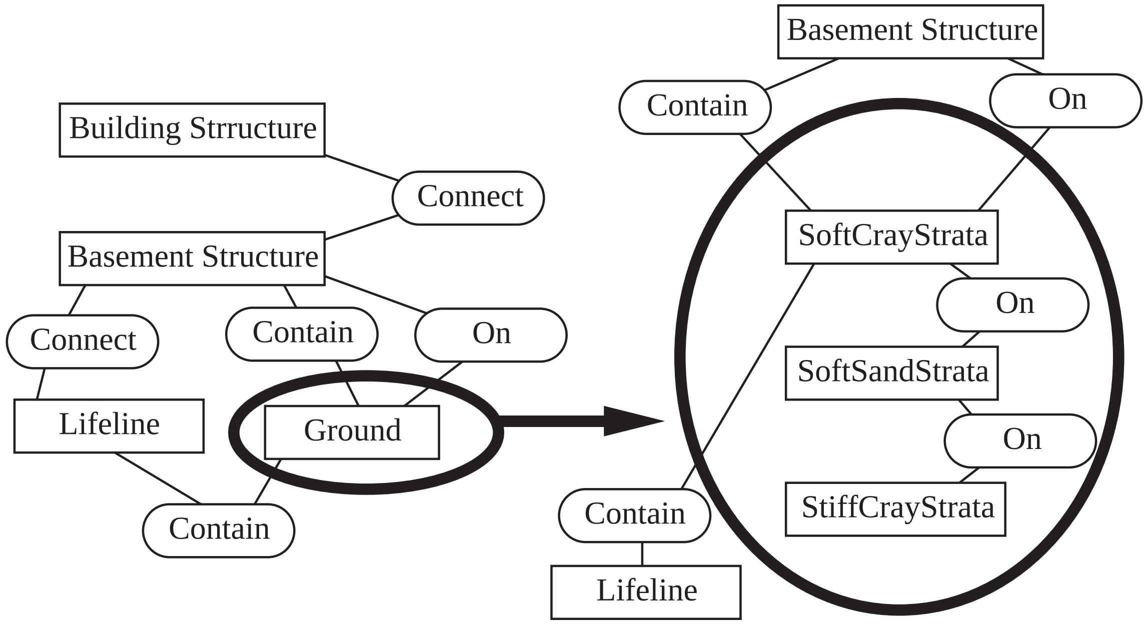


Figure 2 Relationships between Different Granularity Models

For supporting selection of an appropriate analysis method from two or more methods, we define knowledge about analysis methods. Table 1 shows the scheme of the knowledge and Table 2 shows an example of the definition for earthquake. Selection of 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 method. Then, KIEF checks whether or not information about each input attribute existing in the metamodel or not.
2. Select appropriate methods by the user.
   The user selects an appropriate method by using the descriptions about analysis method and the information about the existence of the input attribute information.
3. Suggest another analysis method to generate information about input attribute.
   If there is no input attribute information in the model, KIEF suggests another analysis method to obtain information by using output attribute definitions in the knowledge about analysis methods.

Table 1 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 2 Example Definition for Earthquake

phenomenon name	earthquake	
analysis method	description1	Use default variable This method can be used for simple design. Set 2 levels for the earthquake Maximum acceleration on the ground: Level1 250 Gal, Level2 500 Gal Maximum velocity on the ground: Level1 25cm/s , Level2 50cm/s
	input attribute	none
	output attribute	Maximum acceleration on the ground, Maximum velocity on the ground
	description2	Use earthquake wave data for design Use appropriate earthquake wave data for the place This method is used for complex design.
	input attribute	place
	output attribute	earthquake wave data(transition data of acceleration and velocity of the structure and the ground)

3.3 Example of Architectural Design

We demonstrate how KIEF works with an example of architectural design. First, the designer inputs an initial model (Figure 3) and adds detail information for Ground with the part-unit editor (Figure 4).

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 5) and evaluates whether or not the problems are critical.

When the phenomena are negligible, the designer specifies these phenomena to be neglected and prop-

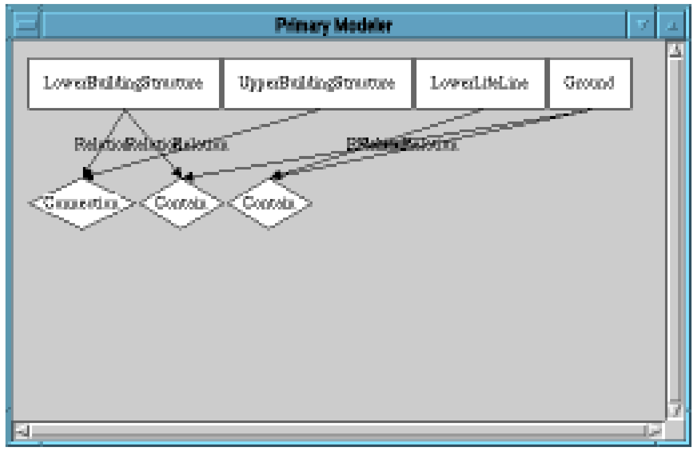


Figure 3 Initial design model

agates this decision to the higher level. In this case, since quick sand is a negligible phenomenon, other phenomena caused by quick sand like liquefaction settlement are also neglected in the higher level modeling (Figure 6). 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. After finishing the design, the designer generates a plan for construction process on a manufacturing process design modeler (Figure 7).

Since it takes time to construct the whole building, there is a chance that undesired natural phenomena happen (e.g., earthquake, typhoon, and so on). This modeler supports to evaluate these intermediate stages of the building under-construction. If the designer builds a model of the building just in the same way as the real construction takes place, models for intermediate building are automatically built. Therefore, the designer can evaluate these intermediate stages in the same way as the building after finishing construction.

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 method. 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 architectural design.

From the view point of the scaling up KIEF, we have to deal with complex design object models. For managing this type of complexity, the architectural designers set multiple granularity level for one design object and manage the relationship among them. However, this type of knowledge management can be

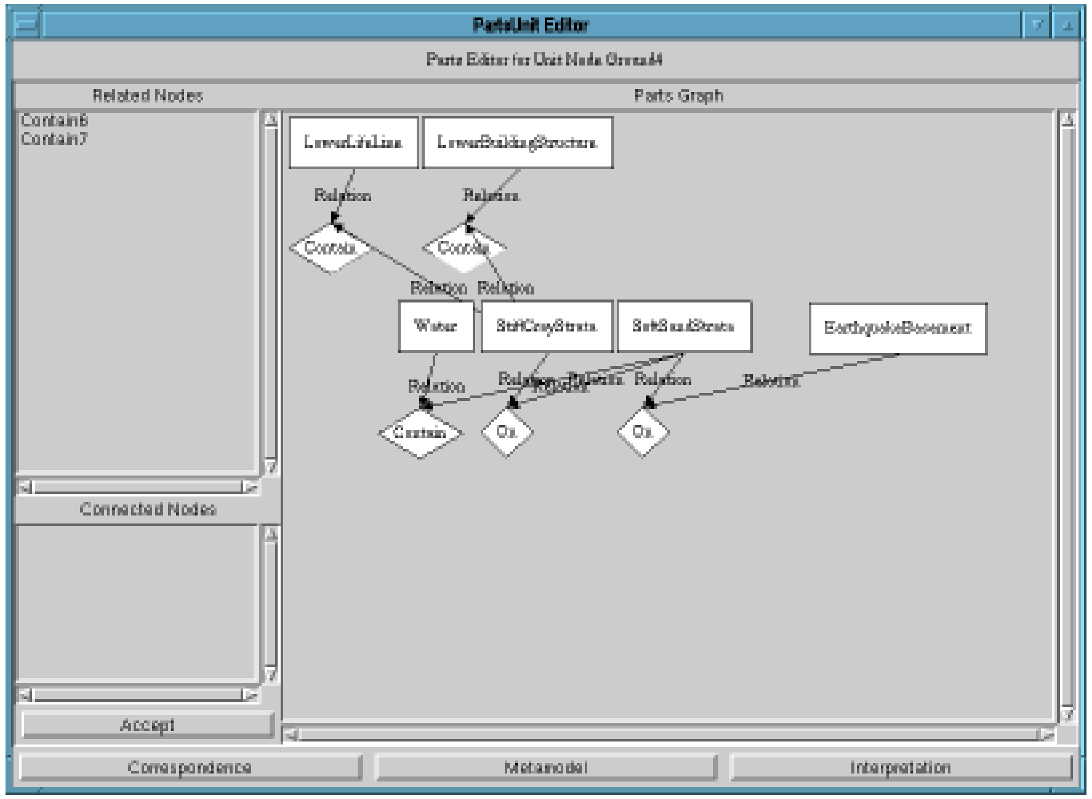


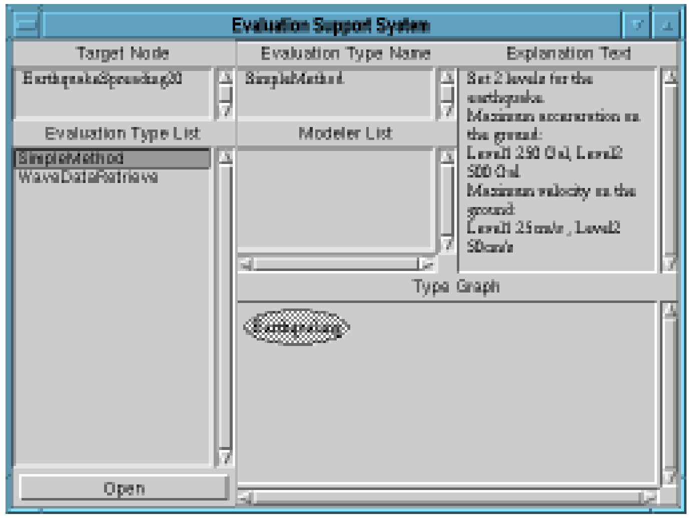
Figure 4 Decomposition of Ground

seen in other engineering domains. Therefore, we believe that granularity handling mechanism of KIEF is also useful not only for architectural 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 like mechanical design, ship design, and so on. 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



Target Node: EarthquakeSpreading20

Evaluation Type Name: SimpleMethod

Explanation Text:
Set 2 levels for the earthquake.
Maximum acceleration on the ground:
Level1 250 Gal, Level2 500 Gal
Maximum velocity on the ground:
Level1 25cm/s , Level2 50cm/s

Evaluation Type List:
- SimpleMethod
- WaveDataRetrieve

Modeler List:

Figure 5 Suggestion of Evaluation Method

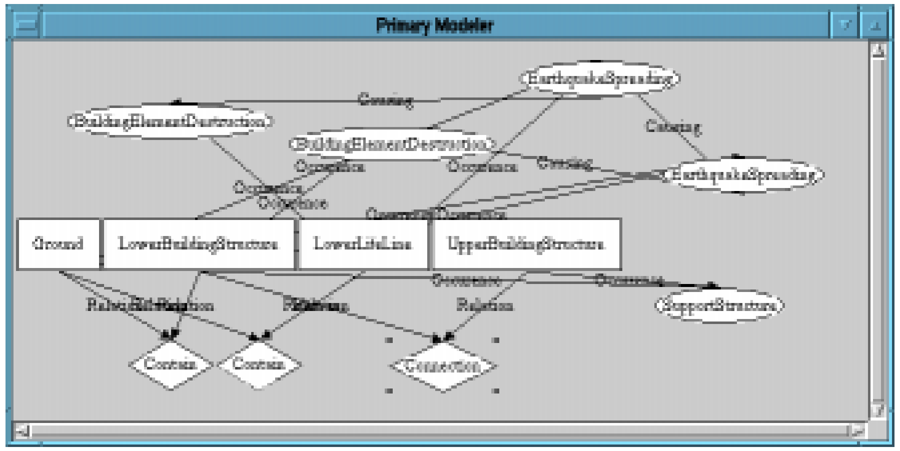


Figure 6 Physical Phenomenon Occurred on the Top Level

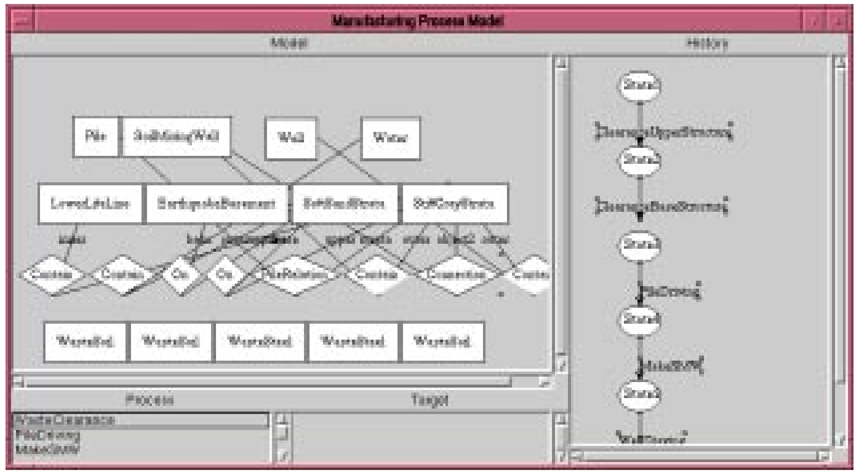


Figure 7 Manufacturing Process Designer

(Knowledge Query and Manipulation Language)(T. Finin *et al.*, 1992), they do not address the interaction between designers and the system in design process. 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 these 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 KIEF to architectural design. In this application, knowledge about causal relationship is described in KIEF 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

Ömer Akin et al. 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,, 1998.

M.R. Genesereth. 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, 1992.

Thomas R. Gruber. Ontolingua: A mechanism to support portable ontlogies. Technical report KSL91-66, Knowledge Systems Laboratory, Stanford University, Stanford, 1992.

M. Ishii, T. Sekiya, and T. Tomiyama. 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, 1995.

James G. McGuire, Daniel R. Kuokka, Jay C. Weber, Jay M. Tenenbaum, Thomas R. Gruber, and Gregory R. Olsen. SHADE: Technology for knowledge-based collaborative engineering. *Concurrent Engineering: Research and Applications (CERA)*, Vol. 1, No. 2,, September 1993.

Architectural Institute of Japan, editor. *Recommendations for Design of Building Foundations*. Architectural Institute of Japan, 1988.

T. Finin *et al.* Specification of the kqml agent-communication language. Technical report EIT TR 92-04, Enereprise Integration Technologies, Palo Alto, California, 1992.

ISO TC184/SC4. *ISO 10303-1 Industrial Automation Systems and Integration - Product Data Representation and Exchange-, Part1: Overview and Fundamental Principles*. 1994.

T. Tomiyama, Y. Umeda, M.Ishii, M. Yoshioka, and T. Kiriyama. Knowledge systematization for a knowledge intensive engineering framework. In T. Tomiyama, M. Mäntylä, and S. Finger, editors, *Knowledge Intensive CAD-1, Preprints of the first IFIP WG 5.2 Workshop on Knowledge Intensive CAD-1*, pp. 33–52. Chapman & Hall, 1996.

M. Yoshioka, T. Sekiya, and T. Tomiyama. Design knowledge collection by modelling. In *PROLAMAT 98*, 1998.

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