DESIGN KNOWLEDGE MANAGEMENT BASED ON A MODEL OF SYNTHESIS

Yutaka Nomaguchi[1] and Tetsuo Tomiyama[2]
[1]*Department of Computer-Controlled Mechanical Systems, Osaka University*
[2]*Faculty of Mechanical Engineering and Marine Technology, Delft University of Technology*

Abstract: In this paper, we report the development of a design knowledge management system, called DDMS (Design Documentation Management System). By recording design documents during design, DDMS encourages a designer to externalize his/her knowledge and facilitates sharing and reuse of such externalized design knowledge in later stages. DDMS works as a front end to KIEF (Knowledge Intensive Engineering Framework), which we have been developing over years. KIEF is capable of integrating multiple design object models and of maintaining consistency among these models. DDMS automatically generates design documents after analyzing design log data and guides designers with design process knowledge based on a model of synthesis. We also illustrate an example of laser lithography design to demonstrate the features of DDMS.

Key words: knowledge management, knowledge acquisition, design rationale, synthesis, knowledge intensive engineering, design documentation

1. INTRODUCTION

Design knowledge management to support creative and effective design activities has been one of crucial issues for manufacturers. Nowadays CAD packages using design knowledge, such as design rules, have been released one after another. CATIA (Dassault Corporation) is one of them. It offers an integrated environment consisting of geometric model-based design support systems and knowledge bases, so that the designer can share and reuse

J. C. Borg et al. (eds.), *Knowledge Intensive Design Technology*

design knowledge for better design. However, these CAD systems lack a model to handle design processes which contains design rationale.

Design rationale includes two types of design information (Suzuki et al. 1996) (i) foreground information that is an explicit result of a design process, such as drawings, CAD data, and product models, and (ii) background information used to generate such foreground information. The latter is significant to understand design rationale, but usually remains unconscious and it disappears from the designer's memory after the design. This makes the acquisition of design knowledge extremely difficult, which is one of the critical issues in knowledge management (Dieng 2000).

Creating design documents is an activity to record design rationale in order to acquire design information and knowledge, and to share and reuse them. Some commercial groupwares, such as Lotus Notes (Lotus Software), integrate CAD files and documentation files so that designers can document design processes as a note during design. However, it is painful for designers to make records of all the design information during design. Besides, these groupware systems also lack a model to handle design process.

In our previous report, we described a new methodology for managing design knowledge and a knowledge management system as a new CAD system (Nomaguchi et al. 2000). Considering the nature of design knowledge stated above, we also proposed a method to acquire design knowledge, called "documentation by design," that means design documents should be composed as a by-product of design. To demonstrate the feasibility of this method, a system called DDMS (Design Documentation Management System) was developed based on the following three requirements:

1. Integration of a CAD environment (we used KIEF described below) and documentation tools,
2. Hypertext-based documentation to handle various types of design information and back-and-forth nature of design process, and
3. Automatic documentation of CAD operations.

Figure 1 depicts the architecture of DDMS that supports creating design documents during design, so that the designer can externalize his/her knowledge into documents, to retrieve past design cases, and to share and reuse it in the future design. However, DDMS reported in our previous paper (Nomaguchi et al. 2000) had some problems. In particular, the system lacked capabilities of organizing and structurizing the acquired design knowledge. This prevented effective retrieval and reuse of past design cases. To solve these problems, this paper introduces a *metaprocess model* to organize and structurize the acquired knowledge and information about design processes. This metaprocess model was derived from a knowledge operation model of synthesis which we developed in a project called "Modeling of Synthesis"

conducted between 1996 and 2001, funded by JSPS (Tomiyama et al. 2000). Based on the metaprocess model, we expanded the capabilities of DDMS, so that it can facilitate:

1. to explicitly capture the design process, and
2. to automatically translate design process information into document description in a natural language

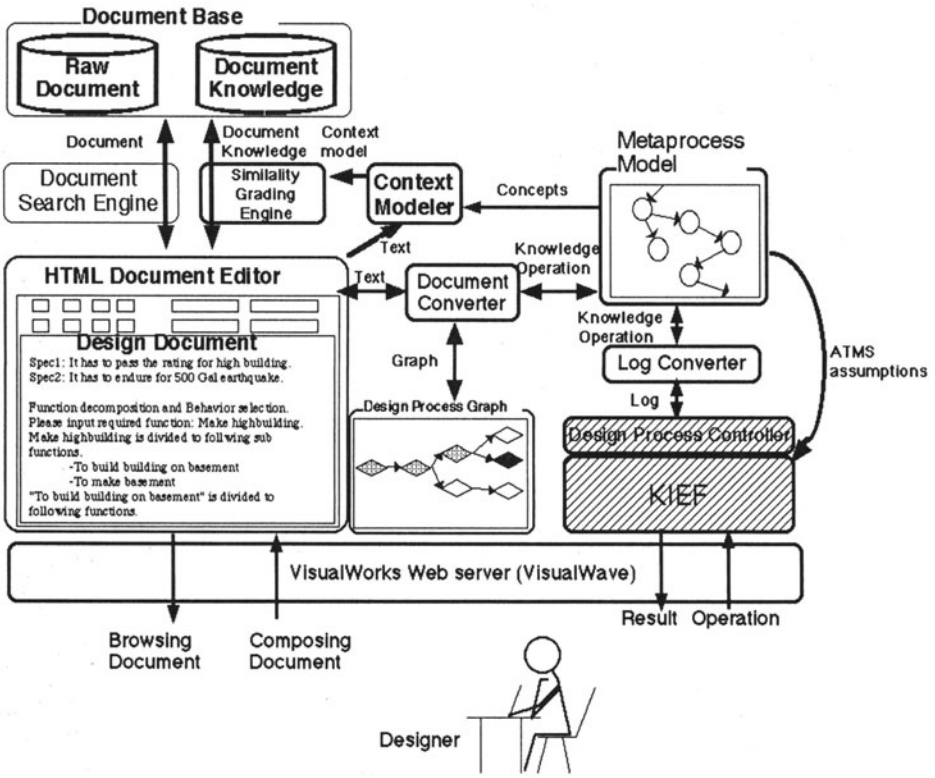

Figure 1. The architecture of DDMS

This paper describes methods (Chapters 2 and 4) and a fundamental theory of the metaprocess model (Chapter 3) needed to build and improved version of DDMS and illustrates its implementation and a design example (Chapter 5).

2. THE KNOWLEDGE INTENSIVE ENGINEERING FRAMEWORK AND THE MODEL OF SYNTHESIS

This section provides a summary of our previous research efforts that formed the basis of this work. First, we describe KIEF (Knowledge Intensive

Engineering Framework) that we have been developing over years. Second, we report some research results of the "Modeling of Synthesis" project that is aimed at establishing a scientific model of synthesis (Tomiyama et al. 2000). This project aimed at establishing a scientific model of synthesis.

2.1 Knowledge Intensive Engineering Framework

Knowledge intensive engineering (Tomiyama et al. 1996) is a new style of engineering to assist engineering activities in various product life cycle stages based on intensive use of the accumulated engineering knowledge. KIEF has been developed to support this concept. Interested readers are invited to refer to (Yoshioka et al. 1996; Sekiya et al. 1997; Sekiya et al. 1999; Yoshioka et al. 1999). Here, we only describe some important components of KIEF.

Figure 2 depicts the core architecture of KIEF. First, KIEF is equipped with the pluggable metamodel mechanism (Yoshioka et al. 1997) to support management of multiple design object models based on their dependency at conceptual level. The pluggable metamodel mechanism maintains relationships among the concepts used in different modelers and, offers the designer an effective design support environment. For instance, data transfer between different modelers is automatically taken care of by the mechanism.

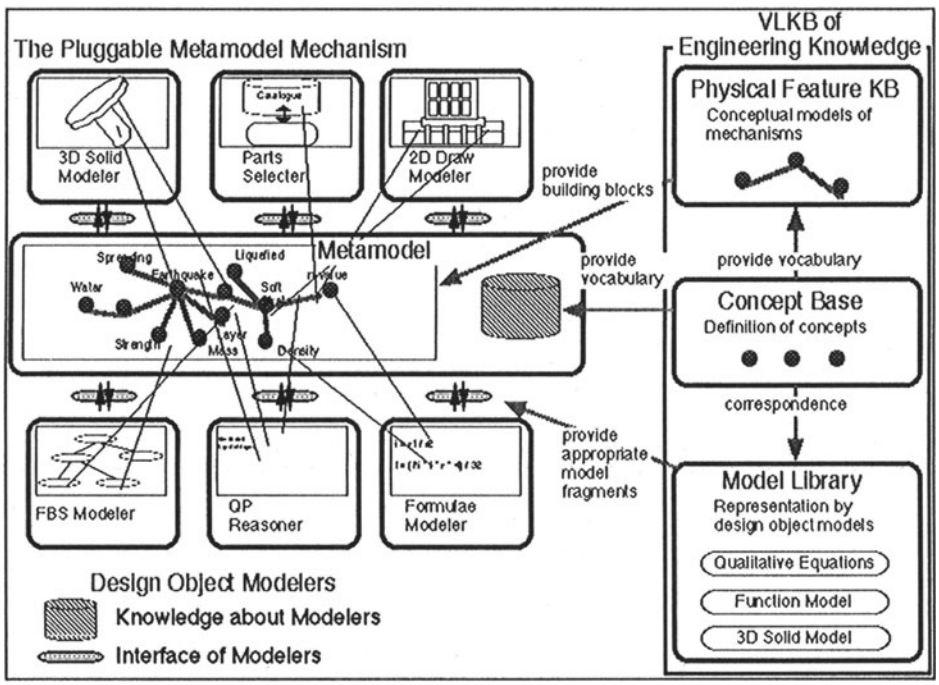

Figure 2. The core architecture of KIEF

Second, VLKB (Very Large-scaled Knowledge Base) accumulates various kinds of knowledge. The core of VLKB is the Concept Base (Ishii et al. 1995; Sekiya et al. 1997) that defines the ontology of physical concepts such as entities, relations, attributes, and physical phenomenon. These concepts form a vocabulary for the metamodel.

Although KIEF itself was carefully designed and implemented with a user-friendly interface, still the system operation is difficult for many designers. This is primary due to the fact that integrated subsystems have their own interfaces and underlying concepts. One of the motivations of this research is to implement more unified KIEF's interface that incorporates the notion of design processes (Ranta et al. 1995).

2.2 Knowledge Operations and Design Vocabulary

The "Modeling of Synthesis" project aimed at establishing a scientific model of synthesis. We analyzed knowledge operations in design processes to be performed on a design process controller of design (see Figure 4).

From the observation of designer's activities, we identified seven different knowledge operations, *Knowledge/Information Acquisition*, *Knowledge/Information Reorganization*, *Information Confirmation*, *Conflict Resolution*, *Knowledge/Information Revision*, *Solution Synthesis*, *Object Analysis*, and these operations do not come in a particular order (Tsumaya et al. 2001).

When we describe actual design activities with the knowledge operations, however, we confronted some problems. The biggest problem was that one design activity in a design process corresponded to more than one operation. This happened due to the difference in the concept granular size between a design activity and a knowledge operation. Because of this, we introduced an intermediate abstract level between them, called *design vocabulary* and composed of standardized terms about design to represent each design activity at more general and detail level. Table 1 lists terms of design vocabulary and knowledge operations into which the terms categorized. The design vocabulary is useful to describe design activities more intuitively.

Table 1. Knowledge operation and design vocabulary (Tsumaya et al. 2001)

Knowledge operation	**Design vocabulary**
Knowledge/Information Acquisition	Investigation, knowledge/information acquisition, problem indication
Knowledge/Information Reorganization	Arrangement of knowledge/information, knowledge/information reorganization, making concrete, drafting
Information Confirmation	Information confirmation

Knowledge operation	**Design vocabulary**
Conflict Resolution	Conflict resolution
Knowledge/Information Revision	Strengthening of the constraint, knowledge/information revision
Solution Synthesis	Suggestion, idea selection, improvement, decision, association
Object Analysis	Evaluation, trial manufacture, experiment, estimation, numerical analysis, derivation

2.3 Design Process Knowledge

Tomiyama and ten Hagen (Tomiyama and ten Hagen 1987) identified two different types of design knowledge, *viz.*, knowledge about design objects and knowledge about design processes. The latter is typically meta-level (or action-level) knowledge and controls object level reasoning activities. For instance, the Design Simulator developed by our group has a two-level reasoning architecture, one at the design object level for actual design and the other to control the design (Takeda et al. 1990).

In this research, we followed this idea of having a meta-level reasoning system. The meta-level knowledge is design process knowledge that was formalized in the form of the knowledge operations identified in the previous section. This design process knowledge (rule) has two slots, *condition*, which contains the conditions of the design object to trigger the application of the knowledge, and *operation*, which contains operations to the design object. For example, design process knowledge of *evaluation of a design solution* is described as follows.

- **Condition:** a new design solution is suggested.
- **Operation:** introducing a knowledge base related to the new design solution (*Knowledge/Information acquisition*), deriving behaviors of the design solution (*Object analysis*), and comparing them with the required specifications (*Information confirmation*).

Figure 3 shows the design process knowledge of *set specification*. Each of Lisp-liked operators is an operation of KIEF. The design process knowledge facilitates to navigate the designer on KIEF to perform proper design activities.

Condition:

(isEmpty: (getCurrentRequirement))

There is no specification in the KIEF system.

Operation:

(setq RelatedModelers (relatedModelers:type: () #requirement))

Suggesting modelers used to describe requirements

(setq SelectedModeler (selectOneFrom:message: (RelatedModelers '=' (<modelers>))
'select one modeler for requirement.'))

Selecting one modeler.

(setq NewModeler (makeModelerOn: (SelectedModeler '=' <modeler>)))

Building a model on the selected modeler.

(setq NewRequirements (getModelerInformation: (NewModeler '=' <modeler>)))

Describing requirements on the model.

(addInformation: (NewRequirements '=' (<specifications>)))

Add specifications to design information in the KIEF system.

Figure 3. The design process knowledge Set specification

2.4 Design Process Controller

We implemented a design process controller based on the model of synthesis. Figure 4 depicts the design process controller implemented at the meta-level of KIEF. The design process controller suggests candidate rules whose condition matches with the condition of the design object, so that the designer can select a rule from candidates to perform the design. By doing so, the design process controller navigates the design process on KIEF. The operations of rules are recorded as a log of design on KIEF, so that they can be replayed afterwards.

These features facilitate design knowledge acquisition through design, which is one of crucial issues for design knowledge management. However, the log of design process knowledge is no more than a log. It is difficult for the designer to understand a log generated in this way and not worthwhile to share and reuse such logs. To solve this problem, DDMS plays a crucial role through its hypertext-document-based approach.

3. METAPROCESS MODEL

DDMS categorizes design information into the following four categories:

1. information about the design object,
2. information about the design process,

3. information about the designer's intent, and
4. information referred in the design.

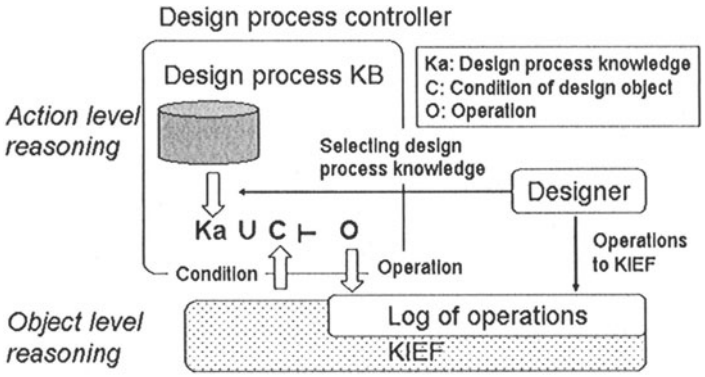

Figure 4. The design process controller of the model of synthesis

Our previous paper (Nomaguchi et al. 2000) reported that this categorization is useful in organizing design information in hyper-text documents. However, DDMS does not provide designers with guidelines to describe design process. This is one of the reasons why it is difficult for a usual designer to compose documents during design on DDMS, although DDMS semi-automatically generates document descriptions from the operations on KIEF.

In this section, we propose a revised model to document design processes in DDMS.

As some research groups stated (Shipman III et al. 1997), a design process has a back-and-forth nature, and thus hypertext-based documents are useful to describe it. This means that it is crucial to document explicit derivation and dependence relationships between design information operated in the design process. In this research, we use the knowledge operation model to describe information derivation and dependence relationships within a design process.

To do so, we first categorize design information into three information types; *internal information* (abbreviated with I in this paper) that exists inside the designer's thinking, *external information* (E) such as one obtained from the Web, and *artifacts* (A). Internal information is categorized further into four types; *viz.*, *design knowledge* (K), *requirements* (R), *design object description* (D1) that is primary an attribute of the design object, such as shape and topological relations between components, and *design object related description* (D2) that is such secondary information derived from D1 as behavior of the design object and physical phenomena which would occur

in the design object. Additionally, we introduce three operations to these information types; *viz.*, *add (a)*, *modify (m)*, and *refer (r)*.

We can now define that information derivation is a process such that *added (or modified) information is derived from referred information.* Information derivation is defined based on the type of referred, added, or modified information as Table 2 shows. For example, *making concrete* of the design vocabulary is defined as reference to internal information and adding/modifying the design object description.

Table 2. Information derivation and knowledge operation ('/' means and, '|' means or)

Knowledge operation	**Design Vocabulary**	E	I					A
			R	D1	D2	K		
Knowledge/information acquisition	Investigation	r					a	
	Knowledge/information acquisition		a		r			
	Problem indication					a	r	
Knowledge/information reorganization	Arrangement of knowledge/information						r/m	
	Knowledge/information reorganization					m	r	
	Making concrete				a | *m*		r	
	Drafting				a | *m*		r	
Information confirmation	Confirmation		r/m	r/m	r/m			
Conflict resolution	Conflict resolution						r/m	
Knowledge/information revision	Strengthen of the constraint		m	r	r			
	Knowledge/information revision					m	r/m	
Solution synthesis	Suggestion		r	a		r		
	Idea		r	a				
	Selection			r/m				
	Improvement		r	m	r			
	Decision						r/a	
	Association			r/a				
Object analysis	Evaluation		r	r	r		a	
	Trial manufacture			r				a
	Experiment				a			r
	Derivation			r		r	a	
	Numerical analysis			r		r	a	

These definitions in Table 2 define the relationships between the knowledge operation model and the information derivation. Using this table, a design process can be explicitly modeled with knowledge operations and

the used information. We call this model of a design process a *metaprocess model.*

DDMS describes a design process based on the metaprocess model in hypertext format with the following features that work as a guideline to compose hypertext documents.

- We call a chunk of hypertext document *section.* A section thus contains descriptions of the design vocabulary with design information.
- We use a hyperlink of hypertext document as information about derivation and dependence relationships. In addition, a hyperlink can be used to refer to such information as figures, Web pages, and models built and used during design activity.

4. ACQUISITION OF DESIGN PROCESS KNOWLEDGE

One of the most crucial problems of design knowledge management is acquisition of tacit and implicit thought processes of the designer, which results in missing design information. In our research, KIEF serves as an integrated CAD environment with the reasoning framework of the model of synthesis. DDMS records knowledge operations on KIEF and builds a metaprocess model without losing design information.

In this work, the designer designs by manipulating design object models in KIEF through DDMS. The additional reasoning framework aids the designer by suggesting possible actions to take (see Section 2). Every action of the designer on DDMS is recorded associated with operations to KIEF and forms a design log (see Section 2.4). DDMS also automatically records and organizes information referred, added, or modified when a knowledge operation is performed (see Section 3). In this section, we describe the methods to generate a metaprocess model (Section 4.1) and to convert it to a design document (Section 4.2). Figure 5 depicts the whole process to generate a design document with our method.

4.1 Generating a Metaprocess Model

A metaprocess model describes the information derivation relationships with seven design information types. We define rules to generate a metaprocess model from the log of KIEF.

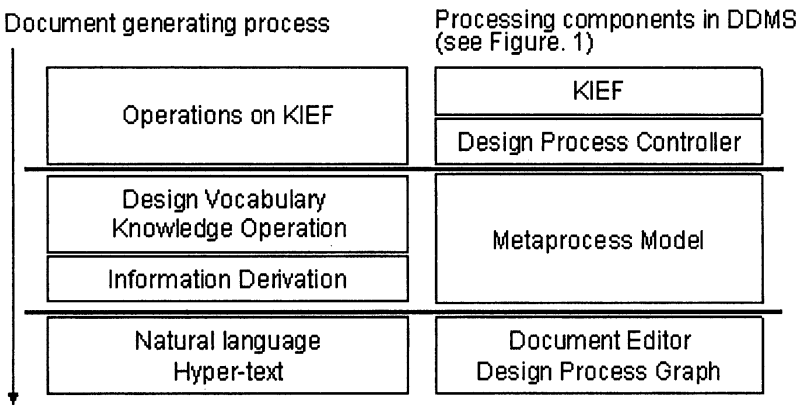

Figure 5. Document generating process

The rules to generate a metaprocess model contain two steps; (i) extracting the log of knowledge operations on KIEF and converting them into corresponding components of the design process knowledge, and (ii) explicating the information derivation of the knowledge operations. Table 3 shows an example of the rules that transfers a log generated by design process knowledge *Set specification*. This rule generates the information derivation which expresses design information <specification> in a log of *Set specification*.

Table 3. A rule of Set specification to generate the information derivation]

The logs generated by the design process knowledge	**The information derivation generated by the rule**
(setq RelatedModelers (relatedModelers:type: () requirement)) (setq SelectedModeler (selectOneFrom:message: (RelatedModelers '=' (<modelers>)) 'select one modeler for requirement.')) (setq NewModeler (makeModelerOn: (SelectedModeler '=' <modeler>))) (setq NewRequirements (getModelerInformation: (NewModeler '=' <modeler>))) (addInformation: (NewRequirements '=' (<specifications>)))	**Design Vocabulary**: Problem indication **Information derivation**: **Refer**: none **Add**: <specification>

4.2 Interpreting a Metaprocess Model into Document

Next, we need rules to transfer the information derivation descriptions in a metaprocess model into a document description written in a natural language such as English and Japanese. The rules convert information derivation descriptions into terms in the design vocabulary. Table 4 depicts example rules.

Table 4. Examples of rules to transfer the information derivation into English

Design vocabulary	**English description**
Problem indication	<add: R> is added as a problem of <refer: D2>.
Knowledge/information organization	To solve the design problem <refer: I>, you can use the knowledge of <add: K>.
Suggestion	The designer suggests <add: D1> as a solution of <refer: R> by using of knowledge <refer: K>.
Experiment	The designer performed the experiment by using of <refer: A>. <add: D2> is revealed as a result.

These transfer rules generate design process descriptions in a document in a natural language every time KIEF performs an operation. They also generate hyperlinks to the information, if the information (expressed in brackets "< >" in Table 4) is in non-text format, such as a model, a figure, and a Web page. While this feature reduces the designer's burden significantly, the system must not disturb the designer to freely compose the document, because our aim here is to acquire the designer's tacit and implicit knowledge by supporting design documentation. This implies that DDMS should function as a word processor, too, with which the designer can compose design documents during design.

5. IMPLEMENTATION

Based on the discussions above, we have developed DDMS. Figure 1 shows the architecture of the system. The system was developed in VisualWorks 5i.2 (a registered trademark of Cincom systems, Inc.), a version of Smalltalk-80, and VisualWave that is an add-on tool for development/deployment of HTML (Hyper Text Markup Language)-based Web applications, so that documents composed with DDMS can be published to geographically distributed designers who wish to access to knowledge of KIEF.

DDMS contains a document editor running on top of KIEF. The designer can use modelers plugged into the pluggable metamodel system of KIEF, including the FBS modeler to support functional design (Umeda et al. 1996), a qualitative reasoning system that analyses behaviors of a product qualitatively (Kiriyama et al. 1992), and other modelers such as a 3D solid modeler. These modelers can uniformly be operated from DDMS that can record operations to them.

5.1 The Functions of DDMS

DDMS's fundamental function is an HTML (Hyper Text Markup Language) text editor, which allows the designer to compose texts in HTML at anytime. In addition, DDMS has the following functions.

1. *Recording design process*: Knowledge operations on KIEF during design are automatically converted into a natural language by a document converter of DDMS, and recorded on the document preserving step-by-step and back-and-forth natures of design processes with hypertext techniques.
2. *Design process management*: The design process controller manages the design process with ATMS (Assumption-based Truth Maintenance System) (de Kleer 1986) embedded in the metamodel mechanism of KIEF. The designer can use this feature on the document editor. When the designer switches a section to another in order to work on an alternative, the metamodel mechanism justifies only information operated in the new section and its ancestral ones.
3. *Document knowledge base*: DDMS has a knowledge base to store documents. The designer can retrieve documents with a retrieval method that is based on the design context (Nomaguchi et al. 2000).
4. *Recording history of information reference*: DDMS records information referred to during design, such as files, Web pages, and models generated in the design process. It also links to them in the design document as hyperlinks of HTML for future reuse. The hyperlinks are automatically generated.
5. *Documenting the designer's intent*: DDMS supports the designer to document the designer's intent. DDMS prompts the designer to enter his/her intent in a natural language every time the designer performs an operation.
6. *Knowledge publishing on Web*: Documents produced by DDMS are coded in HTML, so that they can be published through a commercially available Web browser.

5.2 Design Session

We illustrate a design example of a laser lithography machine, and demonstrate the power of DDMS. The designer can perform the design on DDMS with object modelers integrated in KIEF and under the support of design process knowledge in the design process controller of the model of the synthesis. The design process on DDMS is automatically interpreted into a document in natural language.

5.2.1 Describing required specification

The designer describes a required functional specification "to perform stereo-lithograph of a product" on the FBS modeler, and a required attribute specification that is described as an inequality "Layer thickness of Fabricated product should be less than 100 μm" by operating design process knowledge *Set specification*. The log of the operations which are performed by this knowledge is converted into information derivation *Problem indication* and finally converted into document descriptions, viz., "Functional specification: Perform(stereo-lithograph)of(product) is added as a required specification," and "Attribute specification 'Layer thickness of Fabricated product should be less than 100' is added as a required specification."

5.2.2 Suggestion

Next, the designer suggests a solution for the functional requirement by reusing a past design case's model by operating design process knowledge *Solution synthesis*. The log of the operations which are performed by this knowledge is converted into information derivations, i.e., *Knowledge/information organization*, and *Suggestion*. The set of the information derivations is then converted into document description, i.e., "To solve the design problem 'Functional specification: Perform(stereo-lithograph)of(product)', you can use the knowledge of 'FBS modeler'. The designer suggests 'LaserPhotoforming SimpleTableControllingPF ...' by using knowledge of 'FBS modeler'."

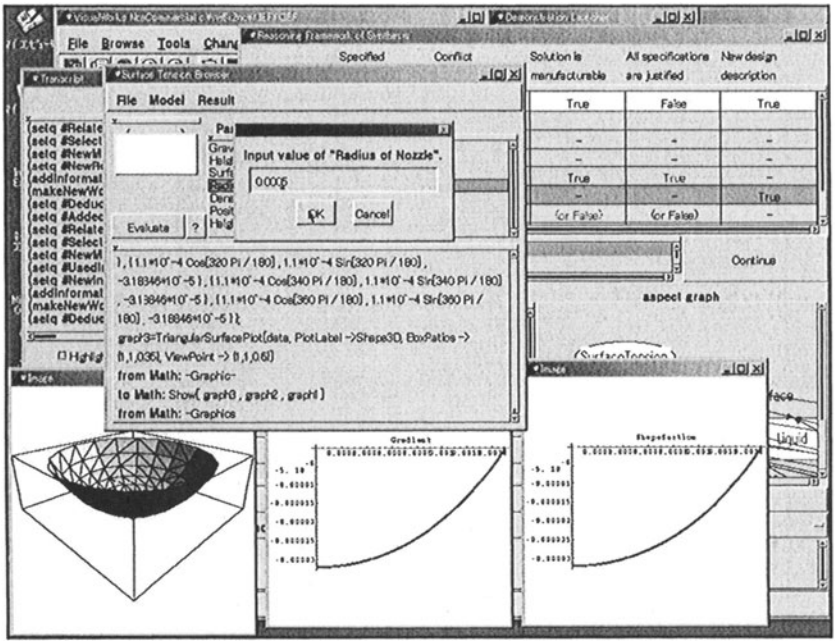

Figure 6. The laser lithography design on DDMS

5.2.3 Analysis

The designer selects the design process knowledge *Solution analysis* to analyze the behavior of the design object. The log of the operations which are performed by this knowledge is converted into information derivations, i.e., *Derivation*, *Knowledge/information organization*, and *Numerical analysis*. The set of the information derivations is converted into document description as follows "Qualitative analysis derived that unpredicted phenomena 'Stick' on 'LiquidSurface Nozzle FabricatedProduct' would occur on the design object. To solve the design problem, you can use the knowledge of 'SurfaceTensionModeler…'. The result of analysis 'Radius of Nozzle = 0.005' is added by 'SurfaceTensionModeler'". After document description was automatically generated, the designer added the annotation "To keep the liquid surface from sticking to a fabricated product, I analyzed the shape of liquid surface, and noticed that the radius of nozzle should be 0.005 mm." Figure 6 depicts the analysis of the shape of liquid resin on DDMS.

In this way, the designer can compose a design document with the support of DDMS during he/she performs the design.

6. DISCUSSION

In this section, we compare DDMS with related works (see Table 5).

CATIA is a commercial CAD package based on a geometric modeling system and a knowledge base system. This means that CATIA has knowledge management capabilities about geometric knowledge of design objects. However it does not have explicit notion of design process management. Lotus Notes is a famous groupware system that manages diverse types of design documents, although it does not have a model of design process for design process management, either.

Table 5. The comparison between our approach and others

	DDMS	Concept Base	Design Rationale: Action-base	Design Rationale: Argumentation-base	Design Rationale: Model-base	Lotus Notes	CATIA
Handling design process information	-	x	x	x	x	x	x
Modeling design object	x	-	x	-	-	-	x
Modeling design process	-	-	x	x	-	-	x
Integrated CAD environment	-	x	-	-	-	-	x
Documentation by design	-	-	-	-	x	-	x
Handling diverse design information	-	x	-	x	-	-	x
Structuring design knowledge	-	-	x	-	-	-	x
Preciousness of query of design knowledge	-	-	-	-	-	x	x
Feasibleness of query of design knowledge	-	-	-	-	-	x	x
Web-based knowledge publishing	-	x	-	-	-	x	x

From the viewpoint of design research, research in design rationale should be mentioned. Garcia classifies attempts to capture design rationale into three categories, i.e., model-based rationale, argumentation-based rationale, and action-based rationale (Garcia et al. 1992). These methods have both pros and cons. Our approach combined these three methods to capture design rationales taking advantages of each of them. First, it is model-based, because we use the metamodel concept to reflect the designer's mental model. Second, it is argumentation-based, because DDMS allows geographically distributed designers to interact with each other by documenting design. Third, it is action-based, because DDMS records the designer's actions during design.

Concept Base (Justsystem Corporation) has a powerful mechanism to manage documents and they have been actually employed in many enterprises in Japan. Concept Base retrieves documents by the statistical

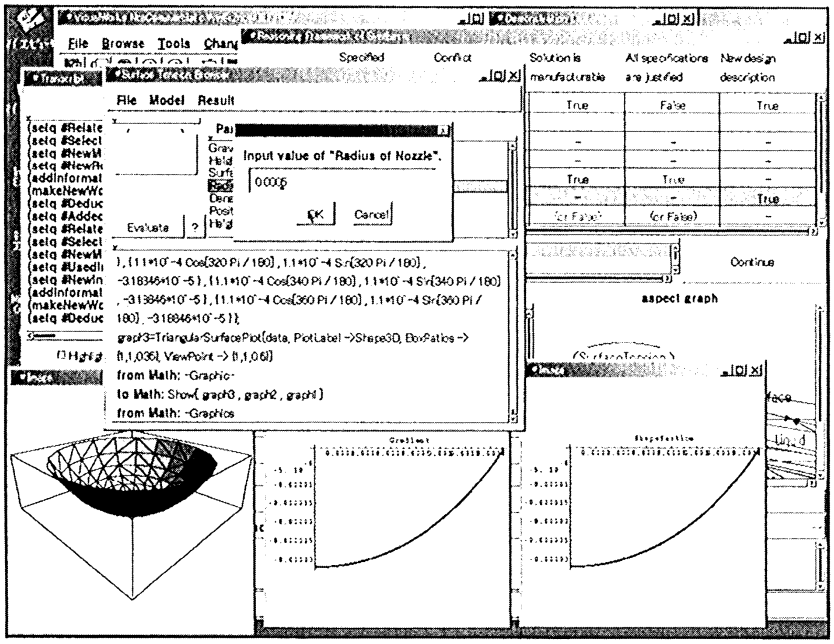

Figure 6. The laser lithography design on DDMS

5.2.3 Analysis

The designer selects the design process knowledge *Solution analysis* to analyze the behavior of the design object. The log of the operations which are performed by this knowledge is converted into information derivations, i.e., *Derivation*, *Knowledge/information organization*, and *Numerical analysis*. The set of the information derivations is converted into document description as follows "Qualitative analysis derived that unpredicted phenomena 'Stick' on 'LiquidSurface Nozzle FabricatedProduct' would occur on the design object. To solve the design problem, you can use the knowledge of 'SurfaceTensionModeler...'. The result of analysis 'Radius of Nozzle = 0.005' is added by 'SurfaceTensionModeler'". After document description was automatically generated, the designer added the annotation "To keep the liquid surface from sticking to a fabricated product, I analyzed the shape of liquid surface, and noticed that the radius of nozzle should be 0.005 mm." Figure 6 depicts the analysis of the shape of liquid resin on DDMS.

In this way, the designer can compose a design document with the support of DDMS during he/she performs the design.

Nomaguchi, Y., Yoshioka, M. and Tomiyama, T, (2000) "Document-based Design Process Knowledge Management for Knowledge Intensive Engineering," Proceedings of the Fourth IFIP Working Group 5.2 Workshop on Knowledge Intensive CAD, pp. 163-185.

Ranta, M., Mäntylä, M., Umeda, Y., and Tomiyama, T. (1996) "Integration of Functional and Feature-Based Product Modeling The IMS/GNOSIS Experience," Computer-Aided Design, Vol. 28, No. 5, pp. 371-381.

Sekiya, T. and Tomiyama, T. (1997) "Case Studies of Ontology for the Knowledge Intensive Engineering Framework," " In T. Tomiyama, M. Mäntylä, and S. Finger (eds.): Knowledge Intensive CAD-1, Chapman & Hall, London, pp. 139-156.

Sekiya, T., Tsumaya, A. and Tomiyama, T. (1999) "Classification of Knowledge for Generating Engineering Models," in S. Finger, T. Tomiyama, and M. Mantyla (eds.): *Knowledge Intensive Computer Aided Design*, Kluwer Academic Publishers, Boston, Dordrecht, London, pp. 73-90.

Shipman III, F.M., and McCall, R.J., (1997): "Integrating different perspectives on design rationale: Supporting the emergence of design rationale from design communication," AI EDAM, Vol. 11. No. 2, pp.141-154.

Suzuki, H., and Kimura, F. (1996) "Modeling Information in Design Background for Product Development Support", Annals of the CIRP, pp. 141-144.

Takeda, H., Veerkamp, P., Tomiyama, T. and Yoshikawa, H. (1990) "Modeling Design Processes," AI Magazine, Vol. 11, No. 4, pp. 37-48.

Tomiyama, T. and ten Hagen, P.J.W.: "Organization of Design Knowledge in an Intelligent CAD Environment," in J.S. Gero (ed.): *Expert Systems in Computer-Aided Design*, North-Holland, Amsterdam, (1987), pp. 119-147.

Tomiyama, T., Umeda, Y., Ishii, M., Yoshioka, M., and Kiriyama, T. (1996) " Knowledge systematization for a knowledge intensive engineering framework," In T. Tomiyama, M. Mäntylä, and S. Finger (eds.): Knowledge Intensive CAD-1, Chapman & Hall, London, pp. 300-314.

Tomiyama, T., Tsumaya, A., Hew, K.P., Kiriyama, T., Murakami, T., Washio, T. Takeda, H., Umeda, Y. and Yoshioka, M, (2000) "A Model of Synthesis from the Viewpoint of Knowledge Operations," Proceedings of Third International Symposium on Tools and Methods of Competitive Engineering, pp. 163-185.

Tsumaya, A., Nomaguchi, Y., Yoshioka, M., Takeda, H., Murakami, T. and Tomiyama, T. (2001), "Verification of a Model of Synthesis – The Methods for Verification and Results," the Proceedings of International Conference on Engineering Design ICED 01, pp. 229-236.

Umeda, Y., Ishii, M., Yoshioka, M. and. Shimomura, Y. and Tomiyama, T. (1996) "Supporting Conceptual Design Based on the Function-Behavior-State Modeler," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 10, No. 4, pp. 275-288.

Yoshioka, M., Oosaki, M. and Tomiyama, T., (1996) "An Application of Quality Function Deployment to Functional Modeling in a Knowledge Intensive Design Environment," In T. Tomiyama, M. Mäntylä, and S. Finger (eds.): Knowledge Intensive CAD-1, Chapman & Hall, London, pp. 300-314.

Yoshioka, M., and Tomiyama, T. (1997) "Pluggable Metamodel Mechanism: A Framework of an Integrated Design Object Modelling Environment," in A. Bradshaw and J. Counsel (eds.), Computer Aided Conceptual Design '97, Proceedings of the 1997 Lancaster International Workshop on Engineering Design CACD '97, Lancaster University, Lancaster, UK, pp. 57-70.

Yoshioka, M., Shamoto, Y., and Tomiyama, T. (1999) "An Application of the Knowledge Intensive Engineering Framework to Building Foundation Design," in S. Finger, T. Tomiyama, and M. Mantyla (eds.): *Knowledge Intensive Computer Aided Design,* Kluwer Academic Publishers, Boston, Dordrecht, London, pp. 197-212. Panzarasa P and Jennings NR (2001) The organisation of sociality: A manifesto for a new science of multi-agent systems, *in Proc 10th European Workshop on Multi-Agent Systems (MAAMAW-01)*, Annecy, France.

Resnick M (1994) *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*, MIT Press, Cambridge, MA.

Schön DA and Wiggins G (1992) Kinds of seeing and their functions in designing, *Design Studies* **13**(2): 135-156.

Simon HA (1969) *The Sciences of the Artificial*, MIT Press, Cambridge, MA.

Smith G and Gero JS (2000) The autonomous, rational design agent, *in* H Fujii (ed.), *Workshop on Situatedness in Design*, Artificial Intelligence in Design'00, Worcester, MA, pp. 19-23.

Smith G and Gero JS (2001) Situated design interpretation using a configuration of actor capabilities, *in* JS Gero, S Chase and MA Rosenman (eds), *CAADRIA 2001,* Key Centre of Design Computing and Cognition, University of Sydney, Sydney, pp. 15-24.

Suwa M, Gero JS and Purcell T (1999) Unexpected discoveries and s-inventions of design requirements: A key to creative designs, *in* JS Gero and ML Maher (eds), *Computational Models of Creative Design IV,* Key Centre of Design Computing and Cognition, University of Sydney, Sydney, Australia, pp. 297-320.

Sycara K (1988) Resolving goal conflicts via negotiation, *in Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, St. Paul, MN, pp. 245-250.

Vaario J and Ueda K (1996) Self-organization in manufacturing systems, *in* K Stelson and F Oba (eds), *Proceedings of the 1996 Japan/USA Symposium on Flexible Automation*, Boston, MA, pp. 1481-1484.

Weber M (1968) *Economy and Society: An Outline of Interpretive Sociology*, Bedminster Press, New York.