

# A Designer Support Environment for Cooperative Design

Motoyuki Takaai, Hideaki Takeda, and Toyoaki Nishida

Nara Institute of Science and Technology, Ikoma, Japan 630-0101

## SUMMARY

The designer proceeds with his work by collecting and combining knowledge about the design object; in so doing, the knowledge possessed by the designer is organized and extended. In cooperative design performed by multiple persons, this organized knowledge is shared among them. This study proposes a designer support environment in which design knowledge is organized by using conceptual systems (ontologies) in the special fields related to the design object. The proposed environment includes workspaces to store individual knowledge and software agents that perform knowledge organization procedures within their own workspaces or across different workspaces, thus supporting the collection, organization, and processing of design knowledge, as well as cooperative design. In addition, the prototype Designers Amplifier system is described, and experimental results obtained with the prototype system in the case of a bicycle rack design are presented. © 1999 Scripta Technica, Syst Comp Jpn, 30(8): 32–39, 1999

**Key words:** Cooperative design; organization of design data; ontology; software agent.

## 1. Introduction

Design is a high-level intelligent production activity performed by engineers supported by certain knowledge to achieve a goal. When a new task is given to a designer, he collects knowledge related to the design object and design methods by all means, including experience, technical

books, and computer networks. Then, the data related to the current design task are combined and adapted properly. Thus, the designer's knowledge expands while it is being organized. Cooperative design with multiple persons involved becomes possible through sharing of such organized knowledge.

This study deals with design knowledge using a conceptual system (*ontology*) of the design object, and a computer environment called *Designers Amplifier* is offered to support designers. The proposed system supports organization of design-related information rather than design itself. *Designers Amplifier* is intended to be a space where cooperative designers store their ontologies and design data. Then the stored information is refined by software agents and is transformed into a handier format. Specifically, *Designers Amplifier* offers the following support: (1) Communication between cooperative designers. Particularly, deep-level co-use of design-related knowledge is ensured through sharing of ontologies between designers. (2) Collecting of relevant knowledge by various means. (3) Design-related fragmentary data (designers' notes, protocols of proceedings, etc.) are organized by using ontologies, which allows their reuse.

Such organization of design information integrates different kinds and stages of design. For example, more systematic collection of functions and structures at the stage of concept design may contribute to a search for similar solutions in the subsequent stages of design.

In Section 2, the organization of design knowledge is explained, and Section 3 presents workplace and software agents of *Designers Amplifier*. Section 4 outlines the configuration of *Designers Amplifier* as a designer support environment, and, in Section 5, an example of *Designers Amplifier* implementation is given.

## 2. Organization of Design Knowledge

In this section, the organization of design knowledge is considered, beginning from the data representation format, and the ontology-based organizational method is explained.

### 2.1. Representation format of design knowledge

Design is development of the technology that is necessary to manufacture an artificial object according to specified requirements. In the course of design, loosely represented requirements are transformed into detailed specifications which can be represented as shown in Fig. 1.

The top level in the diagram involves a representation aimed at human comprehension (media level). The intermediate level involves a formal representation in terms of a grammar that can be processed by computer. The bottom level involves a model representation which can be thought of as a description according to a specific model to be interpreted by a computer. The design process may be considered as the process of obtaining an appropriate algorithm for manufacture, represented at the model level, from design requirements represented at the media level, as shown by the curve in the diagram.

In particular, the transformation from the media level to the formalization level can be treated as organization of design knowledge. In this context, the authors have proposed *Designers Amplifier*, an environment that supports designers through organization of information represented at the media level by special-field ontologies.

### 2.2. Ontologies in *Designers Amplifier*

In this study, design information is organized by linking information using ontologies. A designer proceeds with his task based on his own concepts in his special field,

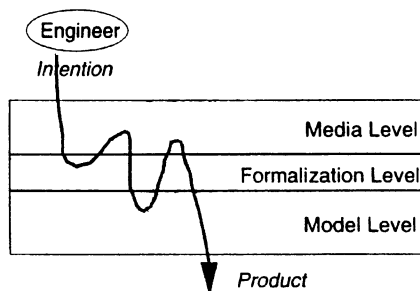


Fig. 1. Levels of informational representation.

from general concepts that can be found in technical books through particular ones specific to an individual designer or workgroup. In this sense, design ontologies contain common parts that can be shared easily and designer-specific parts.

*Designers Amplifier* provides a workspace for individual designers to store ontologies. This means that every designer can build his own ontologies in his own special way. However, *Designers Amplifier* supports ontology sharing by concept exchange in the workspace.

Ontologies are used in this study, for the following purposes:

1. To organize information represented at the media level.
2. To clearly define and refine designers' concepts.
3. To share ontologies among designers, thus contributing to mutual understanding.
4. To ensure intelligent processing of searches and other procedures by using relationships between concepts in the ontology framework.

This study focuses on building and using design object ontologies by cooperative designers. Frame-based ontologies [1] have been developed to support the aforementioned functions of *Designers Amplifiers*. The structural elements of ontologies are as follows.

- *Class*: describes general concepts. *Class* includes a name, a representation of the concept at the media level (*Expression*), and an attribute (*Slot*). The attribute includes three elements, namely, the attribute name, the class to which it belongs, and the attribute value.
- *Super-Sub Class*: represents hierarchical relations between classes.
- *Connection*: represents various associations between classes.

Documents and other data at the media level are assigned to *Expressions* of ontology *Classes* to be organized.

This study uses so-called domain ontologies [7] rather than task ontologies because the proposed system is not intended to support the design task itself but to support the organization of information in the course of design, as explained above.

## 3. Workplaces for Individual Designers

In cooperative design, knowledge, experience, and other specific information possessed by individual designers are of great importance. Individual designers collect, utilize, and organize information in their own specific way.

In the course of cooperative design, such information is exchanged and shared among designers. As will be explained below, with *Designers Amplifier*, these activities are modeled by software agents [4] that operate within or between individual workplaces.

### 3.1. Design knowledge sharing by software agents

Cooperative design progresses as individual designers supplement one another with special knowledge and ontologies. However, individual knowledge and ontologies, and even common knowledge in the field, change fast, particularly for the most advanced technologies. In addition, time limits for new designs are increasingly strict, and workgroups are arranged more flexibly. Thus architectures that provide flexibility with respect to networks, such as using mobile agents, prove more effective than architectures that use concentrated management of data indexes, such as mediators [6].

With *Designers Amplifier*, as shown in Fig. 2, every designer is provided with a workplace where design information, ontologies, and agents are stored. This is the basis for collecting design information from various sources, exchanging data with other designers, and process available information.

Here are examples of the way designers can be supported using software agents in *Designers Amplifier*:

1. Collection of necessary data by wandering around the workplace.

If necessary information cannot be specified clearly, iterative search is narrowed by using certain cues. Such

collection is easily implemented through software agents that are capable of moving between different sources.

#### 2. Information watch

If a designer needs to monitor the latest information, this can be easily realized by placing a software agent at a relevant source and instructing it to report on changes.

### 3.2. Agents in *Designers Amplifier*

Agents act according to instructions from designers or other agents. The agents can move from one place to another. In addition, they can distribute information acquired in a workplace. With the current modification of *Designers Amplifier*, agents provide the following features:

- Agents exist at workplaces.
- Agents can maintain the current state.
- Agents can move to other places together with the states maintained.
- Agents can replicate themselves.
- Agents can exchange information with other agents at the same place.
- Agents operate according to instructions from other agents or users, or they can act autonomously, depending on information available at the workplace and at their own states.
- Agents can acquire data in the workplace and rearrange or distribute it.
- Agents are capable of self-destruction.

In this framework, agents act on behalf of designers to collect design information from various sources, to exchange data with other designers, or to process available information. Using these functions, multifunction information flows can be easily implemented; for example, agents can be created to report about changes of information stored in other workplaces.

## 4. Designer Support Environment *Designers Amplifier*

As stated in Section 2, ontologies are important elements of design. However, design ontologies are not fixed; they are developed and modified as necessary. The designer support environment *Designers Amplifier* is intended to support such construction of ontologies.

Ontologies are constructed by extracting concepts offered by individual designers and structuring these concepts. With *Designers Amplifier*, ontologies are stored at individual designers' workplaces, and these workplaces are used to extract, structure, and share concepts. In so doing, information stored in a workplace is organized by assigned

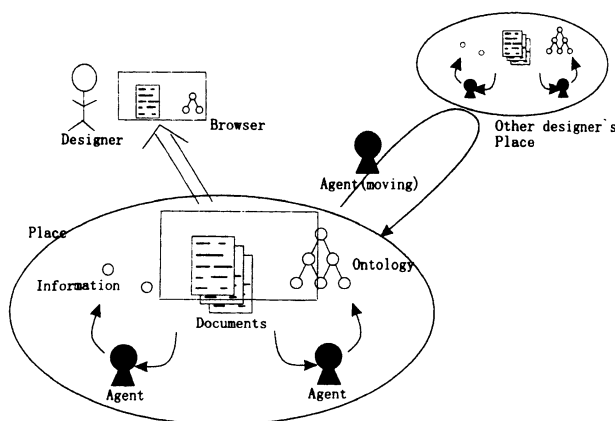


Fig. 2. Workplace and agents of *Designers Amplifier*.

software agents according to relevant ontologies. In addition, mobile agents can retrieve ontologies from other places and bring ontologies to other places.

The designer can see the design data and ontologies stored at a workplace, in full or in part, by using a browser. The same browser is used to give instructions to agents. In this context, this paper deals with methods of organizing and searching design information at workplaces. As for ontology integration, several methods are reported in Ref. 5.

Figure 3 presents the relationships between designer, ontology, and design information at the workplace. The designer directly edits concepts included in the ontology. The organizational agent organizes design information using the ontology. Normally, it is not an easy task to construct an ontology. However, while referring to design information, the designer develops and refines his own concepts. *Designers Amplifier* supports the construction of an ontology by searching design information, identifying similar words, and other procedures performed by agents.

Figure 4 shows how agents organize documents stored in a workplace of a prototype system. The organizational agent organizes a set of documents stored in the workplace, while a collocation data calculation agent and word weight calculation agent calculate collocation parameters and word weights, respectively. Based on these data, the search agent searches for documents. A word similarity calculation agent calculates word similarity primarily by means of collocation data.

Search results, collocation data, and similarity data help the designer to proceed with design and to construct the ontology. The functions of the aforementioned agents are explained below.

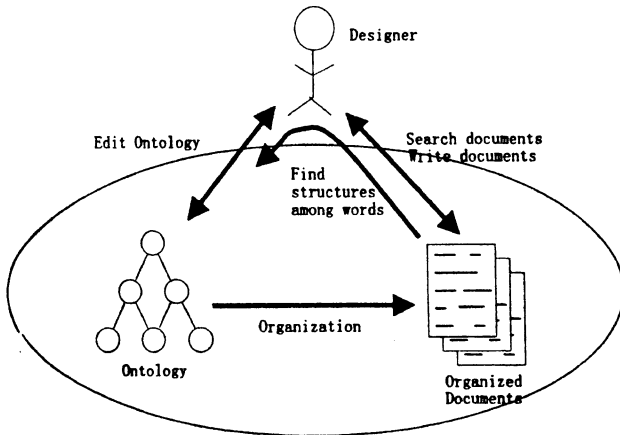


Fig. 3. Relationships between designer, ontology, and design information.

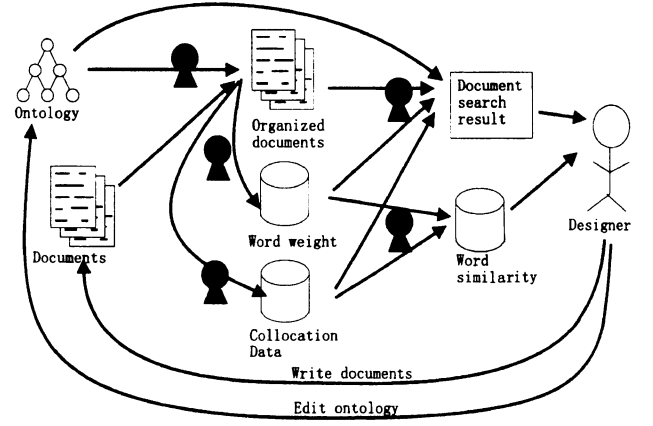


Fig. 4. A set of document organization agents.

#### 4.1. Design information organizational agent

The design information organizational agent uses *Expressions* of ontology *Classes*. With the prototype system, organization is performed by finding words related to relevant concepts and establishing links to concepts. In the current stage, word inflections and other details are disregarded so that words in a document are found through mere character string matching. This approach is perhaps too basic, but it has the advantage that the system can operate with Japanese, English, and other languages without the need for any language-specific software.

#### 4.2. Collocation data calculation agent

Assuming the existence of semantic relationships between words that appear close to each other in a text, it seems reasonable to examine collocation (co-occurrence) data for two words. The collocation data calculation agent performs the following calculations for words associated to ontology concepts.

Consider document  $d$  in a document set  $D(d \in D)$ . Denote by  $loc(d, w)$ ,  $loc(d, w')$  sets of locations of words  $w$ ,  $w'$ . In this case, the collocation of words  $w$ ,  $w'$  in the set  $D$  is defined as the collocation matrix  $M$  that has the following elements:

$$M_{ww'} = \sum_d \sum_{l \in loc(d, w), l' \in loc(d, w')} m(w, l, w', l') \quad (1)$$

Here  $m(w, l, w', l')$  is the collocation weight when  $w$ ,  $w'$  appear at locations  $l$ ,  $l'$ , respectively. In the prototype system, the following expression is used, treating location  $loc$  at the sentence level (word  $w$  appears in the  $l$ -th sentence):

$$m(w, l, w', l') = \begin{cases} 0 & (|l - l'| > 10) \\ (10 - |l - l'|)/10 & (|l - l'| \leq 10) \end{cases} \quad (2)$$

The collocation data calculation agent stores collocation matrices in the workplace.

### 4.3. Word weight calculation agent

The word weight calculation agent calculates word weights (importance) in organized texts by using the vector space model [2]. Words are weighted by the product of the term frequency  $tf$  (relative frequency of word occurrence in a document) and the inverse document frequency  $idf$  for a document set, that is,

$$tf_{dw} \cdot idf_d \quad (3)$$

Here  $tf_{dw}$  is the occurrence frequency of word  $w$  in document  $d$ , and  $idf_d$  is the inverse of the number of documents that contain word  $w$  in document set  $D$ . Generally,  $idf_d$  is expressed as follows:

$$idf_d = \log(N/n_w) \quad (4)$$

Here  $N$  is the size of document set  $D$ , and  $n_w$  is the number of documents that contain word  $w$ .

### 4.4. Design document search agent

The design document search agent is used to search for necessary documents among multiple documents stored in a workplace. With the prototype system, search agents offer two kinds of search: (1) collocation-based search, and (2) ontology-based search. In both cases, words specified by the user, as well as related words, are used as search keywords to perform intelligent search.

#### 4.4.1. Collocation-based search

When a document is searched, collocated words are added to the specified keywords, which provides an allowance for incompleteness of the initial keyword set. In particular, the vector  $\sum_w Q_w \cdot M_w$  ( $M_w$  is the  $w$ -th column vector of collocation matrix  $M$ ) is derived from the keyword set vector  $Q$  (the  $w$ -th component is 1 if  $w$  is a keyword; otherwise it is 0). This vector is added to the initial keyword vector  $Q$  to obtain a new search vector. This new vector is refined by multiplying each element by the respective word weight so that low-importance words are eliminated. The search result (appropriate documents) is expressed as the cosine of the angle between the keyword vector of docu-

ment  $d$  (this vector is found from the word occurrence frequency) and the search vector.

#### 4.4.2. Ontology-based search

This kind of search is performed using relationships between search word concepts and ontology concepts. With the prototype system, all concepts located below search word concepts (*Sub Class*) are involved in the search. In other words, with  $C$  denoting the set of concepts corresponding to search word set  $W$ , and  $sub(c)$  ( $c \in C$ ) denoting a function oriented toward lower level concepts, a new search word set is obtained by adding the set of lower level concepts  $\{sub(C), sub \circ sub(C), \dots\}$  to  $C$ . As with collocation-based search, words are first weighted using a vector space model.

### 4.5. Word similarity calculation agent

The similarity between words  $w_1, w_2$  is defined as the cosine of the angle between collocation vectors  $M_{w1}, M_{w2}$ . This approach is intended to be refined by using collocations between high-importance words; thus, in the prototype system, collocation vectors are modified by weighting corresponding words.

## 5. Example of Designers Amplifier Implementation

Described below are experiments with the prototype system of *Designers Amplifier* in the case of a protocol related to bicycle rack design [3]. This protocol includes design discussion represented as unorganized text data. The purpose of the experiments was to verify the following features of *Designers Amplifier*:

1. Organization of protocol data
2. Extraction of design knowledge from protocol data
3. Reuse of design knowledge by searching protocol data

### 5.1. Protocol of bicycle rack design

The protocol consists of text data (about 1500 lines) related to discussions of bicycle rack design by three designers (not especially experienced in bicycle design). Specifically, the discussion topic was the design of a rack for a mountain bicycle to carry a backpack.

A part of the protocol is given in Fig. 5. The discussion structure may be summed up as follows:

1. General design guidelines



window on the lower right is a concept editing window where a detailed definition of the concept is displayed. In our example, the concept *clip* is defined by the *clip* expression and has a *cost* slot (belonging to the *money* class).

### 5.3. Browsing/searching of organized documents in the prototype system

Figure 6 presents search results in the prototype system of *Designers Amplifier*. The designer can select the required concept (*clip*) in the ontology editing window, thus instructing the search agent to perform the search; results are displayed in the left part of screen. All documents found are listed in the left upper window, and individual documents are displayed in the left lower frame. The documents have been organized, and concepts that appear in the documents have been associated with the ontology. Concepts in displayed documents are provided with buttons; using these buttons, the designer can immediately confirm a concept in the right window. In our example, the word *clip* has been associated with the concept *clip*. The search uses a collocation-based search agent. The collocation data for *clip* used in the search are described below.

### 5.4. Collocation data calculation

Figure 7 presents concepts collocated with *clip*; the concepts were extracted from collocation data in documents related to the bicycle design protocol.

frame 10.9	rigidity 1.9
bag 10.0	rotate 1.8
pack 5.6	belt 1.7
thing 5.0	structural 1.7
bike 5.0	narrow 1.6
wire 2.6	bracket 1.6
outside 2.4	pivot 1.6
strap 2.4	click 1.6
back 2.4	rock up 1.6
slip 2.3	cover 1.5
tube 2.2	seat tube 1.4
snap 2.0	rack 1.4
come off 2.0	wide 1.2
little hook 2.0	original 1.1
connectors 2.0	integral fastening system 1.1

Fig. 7. Collocations for word *clip*.

Words in the diagram represent collocated concept names, and figures stand for collocation values calculated as explained in Section 4.2. Among the concepts listed, many offer some relation to *clip*, particularly attachment-related words such as *wire*, *strap*, and *little hook*. This suggests that such a search is helpful for the designer in constructing his ontology.

### 5.5. Experiments using protocol

Experiments were conducted to prove that the prototype system provides designers with useful information by organization of the design protocol and generation of collocation data.

The following scenario was used in the experiments: the parts of the protocol related to rack-to-bicycle attachment were searched for the purpose of confirmation. The subjects were familiarized with another protocol concerned with the same design task (bicycle rack) but none of them had seen the protocol used in the experiments. The method supported by prototype system was compared to another search algorithm, namely, character string search, by applying the UNIX *less* command to raw data.

Experiments were carried out in the following way:

1. Time limit: 1 hour.
2. The subjects searched six to seven parts related to the rack-to-bicycle attachment method.
3. We prepared three important parts of the protocol which explain the method to attach the rack to the bicycle. In *less*-based search, they found only one of the important parts, whereas all three important parts were found adequately when using the prototype system. With the prototype system, a typical procedure was as follows: (1) For an attachment-related word, for example, *strap*, collocations were examined to find other attachment-related words (*clip*, *screw*, *bracket*). (2) From ontology relationships, appropriate words for search were extracted. Among the documents found using these search words, the three or four top entries were read to check whether they were relevant to the discussion of attachment methods.

The experiments proved that organization of design information by *Designers Amplifier* is efficient for knowledge reuse.

## 6. Conclusions

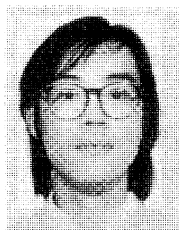
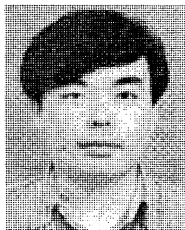
This paper has discussed the designer support environment *Designers Amplifier*, which offers comprehensive support for design activities by work groups and individuals in the framework of workplaces and software agents.

The environment provides for storing design data and ontologies at workplaces and using software agents to implement exchange and sharing of individual design knowledge among cooperative designers. In addition, it was shown that organization of design knowledge by ontologies makes possible flexible search for and extraction of design knowledge, thus supporting designers in reusing design information and ontology construction.

## REFERENCES

1. Gruber TR. Ontolingua: A mechanism to support portable ontologies. Technical Report KSL 91-66, Stanford University, Knowledge Systems Laboratory, 1992.
2. Salton G. Introduction to modern information retrieval. New York: McGraw-Hill; 1983.
3. Takeda H, Yoshioka M, Tomiyama T, Simomura Y. Analysis of design processes by functional evaluation process. In: Cross N, Christaans H, Dorst K, editors. Analyzing design activity. Chichester: John Wiley & Sons; 1995. p 187–209.
4. White J. Mobile agents white paper. <http://www.genmagic.com/agents/Whitepaper/whitepaper.html>
5. Takaai M, Takeda H, Nishida T. Distributed ontology development environment for multi-agent systems. Working Notes for AAI-97 Spring Symposium Series on Ontological Engineering, 1997, p 149–153.
6. Takeda H, Iino K, Nishida T. Agent organization and communication with multiple ontologies. *Int J Coop Inf Syst* 1995;4:321–337.
7. Mizoguchi R, Ikeda M. Ontology in engineering—basics of content-oriented studies. *Jinko Chinoshi* 1997;12:559–569. (in Japanese)

## AUTHORS (from left to right)



**Motoyuki Takaai** (nonmember) graduated from Osaka Prefecture University in 1993. He completed the first term of his doctorate at Nara Institute of Science and Technology in 1995, and is now in second term. His research interests include knowledge sharing and reuse and designer support environments. He is a member of ISCIE and the Japanese Society for Artificial Intelligence.

**Hideaki Takeda** graduated from the University of Tokyo in 1986, and in 1988 he completed the M.E. program. In 1991, he completed his doctorate and was employed as a researcher by the Japan Systems R&D Center. In 1992 he was a doctoral fellow at the Norway Institute of Technology. In 1993 he became an assistant at Nara Institute of Science and Technology, and since 1995 an assistant professor. He has a D.Eng. degree. His research interests include knowledge sharing and reuse, design theory, and intelligent CAD. In 1995 he received the Paper Award from the Japan Society for Artificial Intelligence. He is a member of the Japan Society for Artificial Intelligence, the Japan Society for Precision Engineering, and AAI.

**Toyooki Nishida** (member) graduated in 1977 from Kyoto University. In 1979 he completed his M.E. program, and since 1980 he has been a professor at Nara Institute of Science and Technology. He has a D.Eng. degree. From 1984–88 he was a researcher at Yale University. His research interests include knowledge sharing and re-use, intelligent media, and qualitative reasoning. He received the 1988, 1989, and 1995 Paper Award from the Japan Society for Artificial Intelligence. In 1990 he received the 30th Anniversary Award of the Information Processing Society of Japan. His publications are *Introduction to Natural Language Processing*, *Qualitative Reasoning*, and others. He is a member of the Japan Society for Artificial Intelligence, the Information Processing Society of Japan, the Cognitive Society of Japan, the Japan Society for Software Science and Technology, the Language Processing Society of Japan, AAI, ACL, and IEEE.