NETWORK AND KNOWLEDGE INTEGRATION

Hideaki Takeda
Graduate School of Information Science
Nara Institute of Science and Technology
Ikoma, Nara, Japan

Toyoaki Nishida
Graduate School of Information Science
Nara Institute of Science and Technology
Ikoma, Nara, Japan

ABSTRACT

In this paper, we discuss how ontologies can represent and handle enormous and heterogeneous information in order to organize information according to our intention. We discuss ontologies from two points of view in this paper. One is how to formalize ontologies that can represent multiple definitions for concepts. We propose to formalize ontologies by aspects and discussed its nature. The other is how ontologies can work to deal with information. We show ontology-based information navigation system called IICA that can collect, classify, and extract information on WWW. Ontologies here work as background knowledge to handle enormous and heterogeneous information.

INTRODUCTION

Design is a process of creating new artifacts, especially creation of specifications of new artifacts for manufacture from given requirements. Although it is a visible and apparent process in design, there is another usually invisible but also important process in design, namely, *reorganization of knowledge*. When a new requirement is given to designers, they sometimes survey their internal knowledge like experience and expertise and sometimes collect external knowledge like books and catalogs in order to solve the given requirement. Such introduction of new knowledge and reinterpretation of existing knowledge cause designers' knowledge to reorganize to adapt current and future design tasks. In this paper we focus on this process, in particular, process of representation of unstructured information to gather from outer world.

INFORMATION FOR DESIGN

Most crucial part of information organization is not in well-structured information but in unstructured one. Designers use various kind of information from physical models to natural languages. We have usually paid attention to so-called "model", e.g., 2/3D model, some mathematical model for thermo and dynamics. Since models are well investigated and understood, there are many books and studies to handle them. But models are not sources but results of knowledge organizing formed for a long design history. Individual designers try to capture unstructured information, and some excellent and common organization of information can become models. Much of designers' work is done with unstructured information.

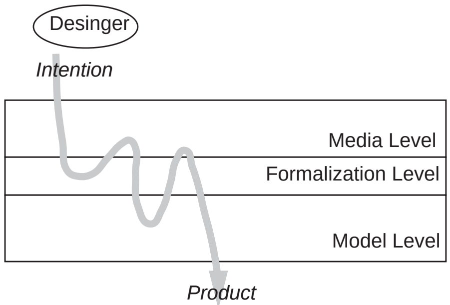

Fig. 1. Levels of Information Representation

Figure 1 illustrates levels of information to deal with. The lowest level is "model" level, i.e, information represented as models. The middle level is "formalization" level, where information is represented by some formal language. The difference between the model and formalization levels is that the former has syntax and semantics, and the latter has only syntax. The top level is "media" level where information representation is only restricted by property of medium which is represented by.

All of these three levels are needed for design because design can be completed only with mixture of some different domains. In general, the lower levels can deal with only specific domains and can offer specific solutions in those domains. On the other hand, the upper level can deal with wider domains but can offer less specific solutions. The middle level can offer interface between both levels.

Design process is performed by penetrating the media, formalization, and model levels in order (see Figure 1). When domain of design is quit new, much effort of design is done in the media level before going the lower levels. On the other hand, when domain is well investigated, the media level is relatively less important and much work is done in the model level.

We discuss the formalization level as "ontology" in this paper. Although ontology cannot capture all functions of the formalization level, it can offer one way to connect the media and model levels.

RULE OF ONTOLOGY IN KNOWLEDGE REPRESENTATION

As mentioned in the previous section, we adopt "ontology" as the formalization level. Ontology is a term in philosophy and defined as "the branch of metaphysical enquiry concerned with the study of existence itself" [1]. In AI community, an ontology is defined as "an explicit specification of conceptualization"[3] which is intended to use as base for knowledge representation. Ontology is represented as systems of symbols in computers, i.e., symbols and their relations represents concepts and their relations. Ontology commits only declarative nature of systems of concepts because its purpose is to provide consent for knowledge representation, e.g., identification of concepts. Therefore ontology is adequate as representation for the formalization level.

Since our main concern is how we can deal with various and unstructured information, we should consider how ontology can be used for dealing with such information. In the Section 4, we discuss formalization of ontology and extend it to deal with various information sources. In the Section 5, we discuss practical value of ontology by showing use of ontology for information handling like collecting, classification, and extraction.

MULTIPLE ASPECTS

In this section, we introduce *aspect* as component of ontology in order to realize multiple ontologies.

Aspects

As we mentioned in the previous section, an ontology is related to a conceptualization. Building a large single ontology means that we can model a large domain by a single conceptualization. It is one of reasons that building large ontologies is not easy in fact.

We often confuse and force to unify concepts from different conceptualization when we try to build large ontologies. For example, Figure 2 shows how concept "temple" is modeled differently. One may think temple as item in textbook of history, so "founded year", "sect", "historical events", and so on are used with "temple". One may think temple as place for religion, so "doctrine", "parishioners", "branch temples", "chief priest" are concepts related to "temple". Of course, some concepts are appeared in more than one aspect like "name" and "location".

Mixture of concepts from different conceptualization often confuse us. Some different concepts in different aspects may denote the same fact, for example "chief priest" in *aspect-for-temple-as-religious-place* and "representative" in *aspect-for-temple-as-public-organization* may be the same. On the other hand, concepts used in more than one aspect may denote different facts, for example "name" is used in common, but its meaning is either religious name or historical

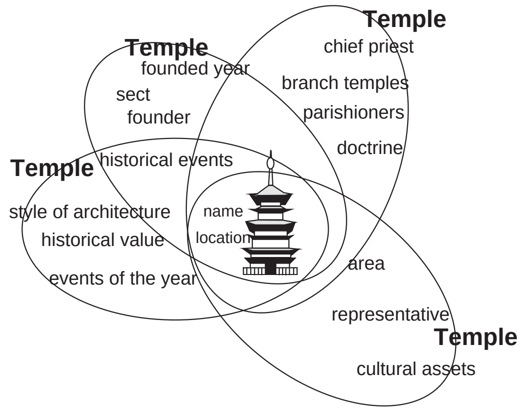

Fig. 2. What is Ontology for Temple?

name or official name with according to aspect it is used. In such cases, the more concepts are collected, the less clear are meanings of concepts.

Each concept is meaningful if and only if concept is used in proper way, that is, concept is used with concepts which come from the same conceptualization. We call this unit *aspect*. We can say that an aspect is a consistent view for conceptualization. Then ontology can be composed of some aspects.

We use various aspects, for example in engineering we use aspects like dynamics aspects, kinematics aspects. To model the common-sense world of traveling, we may use aspects like traffic aspect or geography aspect.

There are two issues on aspect. One issue is what should be in aspect. Aspect should have a vocabulary to describe phenomena in its domain. It should also have a theory which associates concepts in its vocabulary. And the theory should be consistent. In the other words, aspect is what we can conceptualize the world without inconsistency[1].

The other issue is how to compose aspects from other aspects.

We provide two types of basic connections among aspects. One is *combination* aspect. This is just integration of aspects for different domains. For example, one of the ways to build *aspect-for-travel* is to combine *aspect-for-hotel* and *aspect-for-traffic*. In this case, concepts like "railway" which is in aspect-for-traffic do not exist in aspect-for-hotel, because domain of modeling is different from each other. In aspect-for-travel, concepts like "tour" are defined using concepts from both aspects.

The other is *category* aspect. This is collection of aspects which share domains but come from different conceptualization.

[1] The other important content of aspect is *intention*, because aspect is abstraction of the target world by some *intention*. But intention itself is seldom written explicitly. We have tried to represent intention of aspect of object modeling as *function*[8].

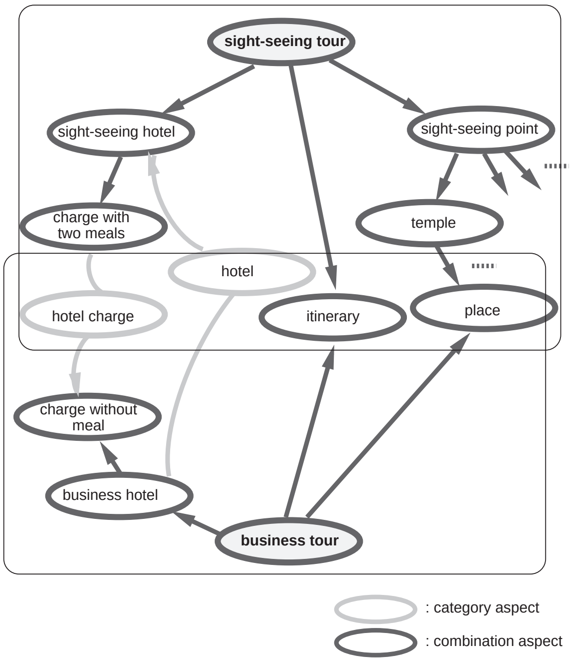

Fig. 3. Multiple and Sharable Ontologies

When a temple is modeled differently we have just shown, we can assume there is a *category-aspect-for-temple*. This aspect has some specific aspects for temple like *aspect-for-temple-as-history-textbook* and *aspect-for-temple-as-religious-place* as component. Since component aspects share domains, it is reasonable (but not mandatory) that there are relations among concepts in different component aspects. Such relations are contents of the category aspects.

Since combination and category aspects can use other combination or category aspects as component, we can construct large aspects from relatively small aspects. We call such relatively large aspects as ontologies.

Figure 3 is an example how different ontologies can be defined with sharing aspects.

Two aspects can share aspects in their constitution, or be connected by category aspects. We call these two aspects are compatible. That is, they may share or transfer information to each other.

In the following sections, we first describe aspect in a logical framework, and then in a programming framework. Finally, we show how communication between different aspects can be established.

A Logical Formalization of Aspect

Since our basic policy is to define aspects constructively, we start from defining atomic aspect and then define more complicated aspects.

We assume a first-order language L_E, and predicate *aspect* of L_E. L is a first-order language which is the same to L_E except predicate *aspect* is removed.

Aspect Theory. First we define *atomic aspect*, aspect which does not depend on any other aspects.

Definition 1 *An atomic aspect A with a consistent theory $T(A)$ of a first language L and with a unique name $aspect(A)$ satisfies the following formula.*

$$aspect(A) \leftrightarrow T(A)$$

$aspect(A)$ is an identifier of aspect which has a similar effect to the second argument of predicate `ist` in Ref. [5], and a modal operator in Ref. [6].

Then, we introduce L_E^m and L^m as modal extension of L_E and L respectively. Here we assume domain of individuals are always the same regardless of possible worlds. In the following discussion, we assume this language L_E^m.

A combination aspect is simply defined as follows.

Definition 2 *$T(A_{COM}(A_1,\ldots,A_n))$, aspect theory of combination aspect for aspect $A_1,\ldots,A_n$, is a consistent theory defined as follows;*

$$\begin{aligned} T \quad & (A_{COM}(A_1,\ldots,A_n)) = aspect(A_1) \\ & \wedge \ldots \wedge aspect(A_n) \wedge I(A_1,\ldots,A_n) \end{aligned}$$

where $I(A_1,\ldots,A_n)$ is a set of formula of language $L^m \cup \{aspect(A_1),\ldots,aspect(A_n)\}$, which means inter-aspect theory among $A_1,\ldots A_n$.

Apparently, it would cause unexpected results if some of aspect theories share predicates. We ideally assume that if the same predicates appear in some aspects, there should share some concepts in them[2]. In such cases, it should be represented by category aspects.

On the other hand, a category aspect is more complicated because it does not imply that both of aspect theories are *always* true. In order to represent a category aspect, we introduce modal operators $\Box$ (necessity) and $\Diamond$ (possibility) and assume S4 modal system. Then a category aspect for two aspects is define as follows.

Definition 3 *$T(A_{CAT}(A_1,\ldots,A_n))$, aspect theory of category aspect for aspect $A_1,\ldots,A_n$, is a consistent theory defined as follows;*

$$\begin{aligned} T \quad & (A_{CAT}(A_1,\ldots,A_n)) = \Diamond aspect(A_1) \\ & \wedge \ldots \wedge \Diamond aspect(A_n) \wedge I(A_1,\ldots,A_n) \end{aligned}$$

where $I(A_1,\ldots,A_n)$ is a set of formula of language $L^m \cup \{aspect(A_1),\ldots,aspect(A_n)\}$, which means inter-aspect theory among $A_1,\ldots A_n$.

$I(A_1,\ldots,A_n)$ is again an inter-aspect theory among $A_1,\ldots,A_2$.∎

Since we can use combination and category aspects as component of aspects, we can define hierarchical aspects using combination and category aspects. In other words, An aspect A is represented $A = f(A_1,\ldots,A_n)$ where $A_1,\ldots,A_n$ are aspects and function f is composed by A_{COM} and A_{CAT}.

Inter-aspect Relations. Then we can define relations between aspect, **inclusion** and **strict inclusion**.

Definition 4 *An aspect A is included in aspect B if and only if $aspect(B) \vdash \Diamond aspect(A)$.*

Definition 5 *An aspect A is strictly included in aspect B if and only if $aspect(B) \vdash aspect(A)$.*

[2] Of course, it is too strict in practise. In programming approach we allows the same predicates in different meanings.

Note that there are two reasons for these relations, i.e., one is composition or category relations among aspects and the other is logical implication. Strict inclusion corresponds *weaker-than* relation in Ref. [6].

Similarly, relations between formula and aspect are defined.

Definition 6 *A formula f is included in aspect A if and only if $aspect(A) \vdash \Diamond f$.*

Definition 7 *A formula f is strictly included in aspect A if and only if $aspect(A) \vdash f$.*

These definitions mean that there are two types of interpretation of aspect theories. One is represented as *strict inclusion* which is traditional way of inter-theory relation. The other is *inclusion* which takes account of all alternatives of theories. By having two types of interpretation, we can deal with both strictly a single representation and variety of representations.

Theorem 1 *If aspect A is strictly included in aspect B, then A is included in aspect B.*

Another relation is **compatibility** which is criteria two aspects are related to each other[3].

Definition 8 *Aspect A and B is compatible if one of the following condition is satisfied;*

(1) A and B is the same aspect,

(2) there exists aspect C which has both A and B as component,

(3) there exist compatible aspect A' and B' are components of A and B respectively.

Definition 9 *Formula f is compatible with aspect A if and only if there exists aspect B in which f is and B is compatible with A.*

Compatibility assure neither consistency nor translatability between aspect theories, but denotes existence of connection between aspects.

ASPECTOL: A Language for Aspects

Here we show a language of aspects called ASPECTOL (Aspect-based Ontology Description Language), which is an extension of Ontolingua-like ontology definition (see [2]). Syntax of ASPECTOL is shown in Figure 4.

Definition of an atomic aspect consists of declaration of aspect name and definitions of classes, relations, and functions. Definition of a combination aspect is definition of an atomic aspect and declaration of including aspects. Definition of a category aspect consists of a set of translation formulae. A translation formula is defined between two aspects in a category aspect, and is defined as `define-translation` which describes logical relation between concepts in both aspects. It can represent a class of direct relations between formulae in two aspects which we have discussed in the previous section. A left hand side of an implication formula is a formula of the aspect of the first argument and a right hand side is a formula of of the aspect of the second argument.

[3] Term *compatible* is borrowed from Ref.[4].

```
(define-aspect aspect-name
   Ontoligua definitions are here
   ...
)
```
(a) atomic aspect

```
(define-aspect aspect-name
  (:use componet-aspct1, ...)
  (:rename (aspect!predicate . new-name) ...)
)
   Ontoligua definitions are here
  ...
```
(b) combination aspect

```
(define-category-aspect aspect-name
  (:use componet-aspct1, ...)
  (:category-type category-type)
  (:rename (aspect!predicate . new-name) ...)
  (define-translation aspect-name
      (=> atmoic-formula1 atmoic-formula2)
    ((:query-procedure nil or t
      :inform-procedure nil or t
       translation-rule
        ...
      ))
     ...
    )
  (define-translation ...)
   ...
   Ontoligua definitions are here
)
```

(c) category aspect

Fig. 4. Definition of aspect

ONTOLOGY FOR INFORMATION NAVIGATION

The other view of ontology we discuss in this paper is how it can contribute to organize real unstructured information. We built a system called IICA (Intelligent Information Collector and Analyzer) that can gather, classify, and extract information on Internet by using ontologies.

Ontology-Based Approach for Information Navigation

Information on the Internet, in particular, WWW information is various and enormous and furthermore totally decentralized, so that there need services to guide such information. But the existing approaches are insufficient to organize information because they treat information without knowledge.

There are mainly two types of services to guide information on the Internet. The first is so-called "WWW spider" that collects and indexes WWW pages periodically so that users can ask pages by keywords, e.g., Lycos and Infoseek. The second is directory service that provides hierarchical categories of WWW pages, e.g., YAHOO!. The former is useful when users can express what they want by keywords, but it is difficult to use when they have only vague goals. On the other hand, in the latter approach, users can find their goals just by choosing categories repeatedly so that it is possible to find their goals even if their goals are vague. But it depends on how categories are well maintained.

Ontology-based information navigation proposes a new way of information navigation because information is treated as knowledge. In our approach, users can define their goals by browsing ontologies, and then the system collects and

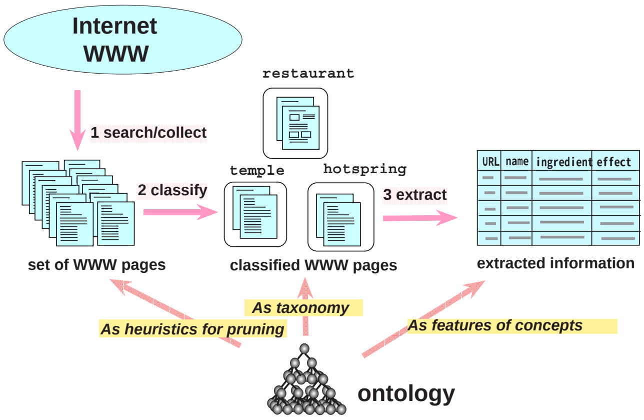

Fig. 5. Roles of ontologies in IICA

classifies WWW pages, and furthermore extract information in them. All information in these processes are provided in ontologies (See Figure 5).

Roles of ontologies

Basically ontologies represent background knowledge that human use consciously or unconsciously when they read pages. For example, suppose searching information on "artificial intelligence". If you find a page on "expert systems", you may think this pages is related to your goal even if there are no descriptions on "artificial intelligence" directly. It is because there is ontological relation between "artificial intelligence" and "expert systems", probably the former is superclass concept of the latter.

In IICA, there are four roles of ontologies, i.e., ontologies plays different roles in each step in information navigation, i.e., guide for users for choosing goals, heuristics for pruning search in information gathering, categories in information classification, and features of concepts in information extraction.

We adopt a simple structure to represent ontologies to achieve above roles. It is difficult to use strict representation like logical formula because information sources are not strict either. There are two types of concepts, i.e., class and attributive concepts, and interclass and class-attribute relations. A class concept has a name and keywords that are used as keywords in search and classification, and attributive concepts which are connected by class-attribute relation. Class concepts can be connected by interclass relation which is super-sub relation with weight to represent similarity between two concepts. There are some types of class-attribute relations like `has-some` which indicate how many values attributive concepts should hold. Attributive concept represents one which has one or more values and is represented as pattern of words or concepts (see next subsection).

Processes of Information Navigation

Firstly, users can browse concepts in ontologies and choose some of them as goals they want. Here ontologies plays guide for defining goals (See Figure 6).

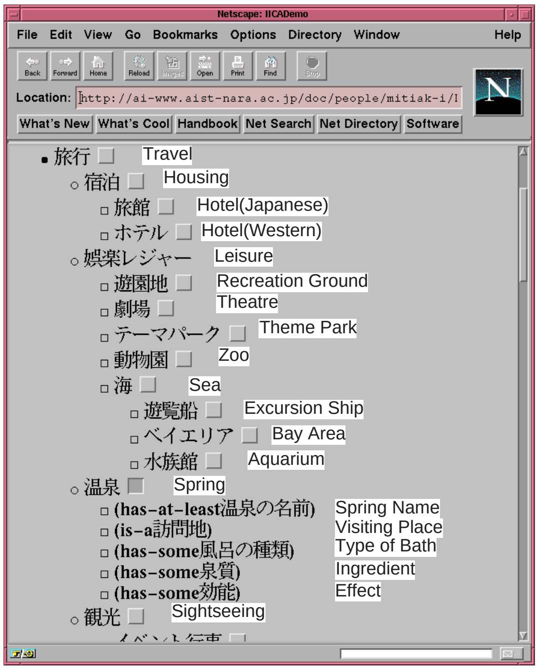

Fig. 6. Ontology as Interface

Then, IICA collects keywords of concepts specified by users and those of concepts related to these concepts. Then IICA collects links whose titles include these keywords. It is because concepts close to the given concepts in ontological relations are also important to find information, e.g., their superclass or subclass concepts may be used in pages. Ontologies are used as heuristics for pruning search secondly.

Thirdly, IICA evaluates full texts of the collected pages and classifies them into concepts in ontologies. In classification process, TFIDF method [7] is used. It is effective when there may be many pages found by search process (see Figure 7). ontologies are used as categories to classify pages.

Finally, IICA extracts information from pages by consulting attributive concepts of the given class concepts. Each concepts can have attributive concepts which are represented by patterns of texts. For example, concept "hot-spring" can have four attributive concepts;

```
(define-pclass
  (hot-spring
    ((has-at-least name)
     (has-some types-of-bath)
     (has-some ingredient)
     (has-some effect))))
```

Concept "effect" which is one of attributive concepts of "hot-spring" is represented as follows;

```
(define-concept
  (effect
    (is disease with (or "効能>" "効果" "効く"))))
(define-concept
  (disease
    (is (or "+病>" "+傷>" "+痛>"))))
```

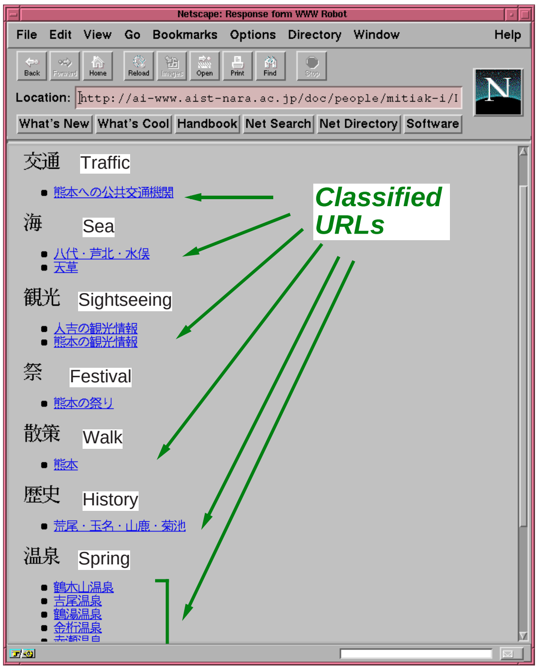

Fig. 7. Results of Information Classification

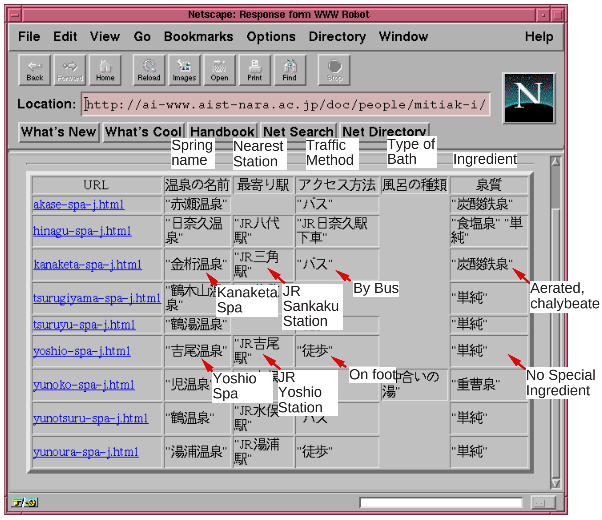

Fig. 8. Results of Information Extraction

The first rule represents that concept "effect" is concept "disease" which appears with "効能"("efficacy" in Japanese) or "効果"(noun "effect" in Japanese) or "効く"(verb "effect" in Japanese) in a single sentence. The second is that concept "disease" is a word ending either character "病" (illness) or "傷"(injury) or "痛"(ache). As results, IICA can build a table as summary of some different pages for a concept (See Figure 8). In this figure, each row contains values of attributive concepts extracted from a single page, and each column contains an attributive concept and its values in those pages. Here ontologies are used as features of concepts when IICA extracts information form pages.

SUMMARY

We discussed ontologies from two points of view in this paper. One is how to formalize ontologies that can represent multiple definitions for concepts. We propose to formalize ontologies by aspects and discussed its nature. The other is how ontologies can work to deal with information. We show ontology-based information navigation system called IICA that can collect, classify, and extract information on WWW. Ontologies here work as background knowledge to handle enormous and heterogeneous information.

It is important how we can represent and handle various information in a unified way, as we have shown, but it is merely the first step to organize information according to our intention. Further research should be done how we can externalize new organization of information from existing information.

REFERENCES

[1] A. Flew, editor. *A Dictionary of Philosophy.* Pan Books, 1979.

[2] T. R. Gruber. Ontolingua: A mechanism to support portable ontologies. Technical Report KSL 91-66, Stanford University, Knowledge Systems Laboratory, 1992.

[3] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University, August 1993.

[4] N. Guarion, M. Carrara, and P. Giaretta. Formalizing ontological commitments. In *Proceedings of AAAI-94*, pages 560–567, 1994.

[5] R. V. Guha. *Contexts: A Formalization and Some Applications.* PhD thesis, Department of Computer Science, Stadford University, Stanford, CA, 1991. (Available as Report No. STAN-CS-91-1399-Thesis).

[6] P. P. Nayak. Representing multiple theories. In *Proceedings of AAAI-94*, pages 1154–1160, 1994.

[7] G. Salton and M. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., 1983.

[8] Y. Umeda, H. Takeda, T. Tomiyama, and Y. Yoshikawa. Function, behaviour, and structure. In J. Gero, editor, *Applications of Artificial Intelligence in Engineering V*, volume 1, pages 177–194, Berlin, 1990. Springer-Verlag.