Developing Ontology-based Applications using Hozo

Kouji Kozaki[*1], Yoshinobu Kitamura[*1] and Riichiro Mizoguchi[*1]

[*1] The Institute of Scientific and Industrial Research, Osaka University

[*1] 8-1 Mihogaoka, Ibaraki, Osaka, 567 -0047 Japan

Tel: +81-6-6879-8416, Fax: +81-6-6879-2123,

E-mail: {kozaki,kita,miz}@ei.sanken.osaka-u.ac.jp

Keywords: Ontology, Knowledge-based Systems, development support system

Abstract. We have developed an environment for building/using ontologies, named Hozo, based on both of a fundamental consideration of an ontological theory and a methodology of building an ontology. Since Hozo is based on an ontological theory of a role-concept, it can distinguish concepts dependent on particular contexts from so-called basic concepts and contribute to building reusable ontologies. This paper presents an outline of the functionality of Hozo and an example of how it is used for creating ontology-based applications.

1. Introduction

Recently, ontology is expected to contribute to knowledge sharing and reuse. It is, however, difficult to develop a well-organized ontology because the principles of ontology design are not clear enough. Therefore, a methodology for ontology design and a computer system supporting for ontology design are needed. Our research goal is development of a methodology for ontology design and supporting environment based on the methodology. And we aim to develop ontology-based applications using the system.

Building an ontology requires a clear understanding of what can be concepts with what relations to others. An ontology focuses on “concepts” themselves rather than “vocabulary”, and its design is not the problem of how to represent but that of identifying the inherent conceptual structure of the target world. Therefore, many researchers have discussed ontological theories and methodologies for building ontologies. However, there are not many tools which reflect the results of these research results. We have developed an environment for building/using ontologies, named Hozo, based on both of a fundamental consideration of an ontological theory and a methodology of building an ontology. Since Hozo is based on an ontological theory of a role-concept, it can distinguish concepts dependent on particular contexts from so-called basic concepts and contribute to building reusable ontologies.

This paper presents an outline of the functionality of Hozo. We focus on how to develop ontology-based applications using it, with an example of developed application.

2. Hozo

2.1. Outline of Hozo

Hozo is composed of four modules: Ontology Editor, Ontology Manager, Ontology Server and Onto-Studio (Fig.1).

Ontology Editor provides users with a graphical interface(Fig.2), through which they can browse and modify ontologies by simple mouse operations. Its feature is the functionality to treat “role concept” and “relation” on the basis of fundamental consideration [Kozaki 00].

Ontology Manager helps users for distributed ontology development. It manages ontologies as components of a

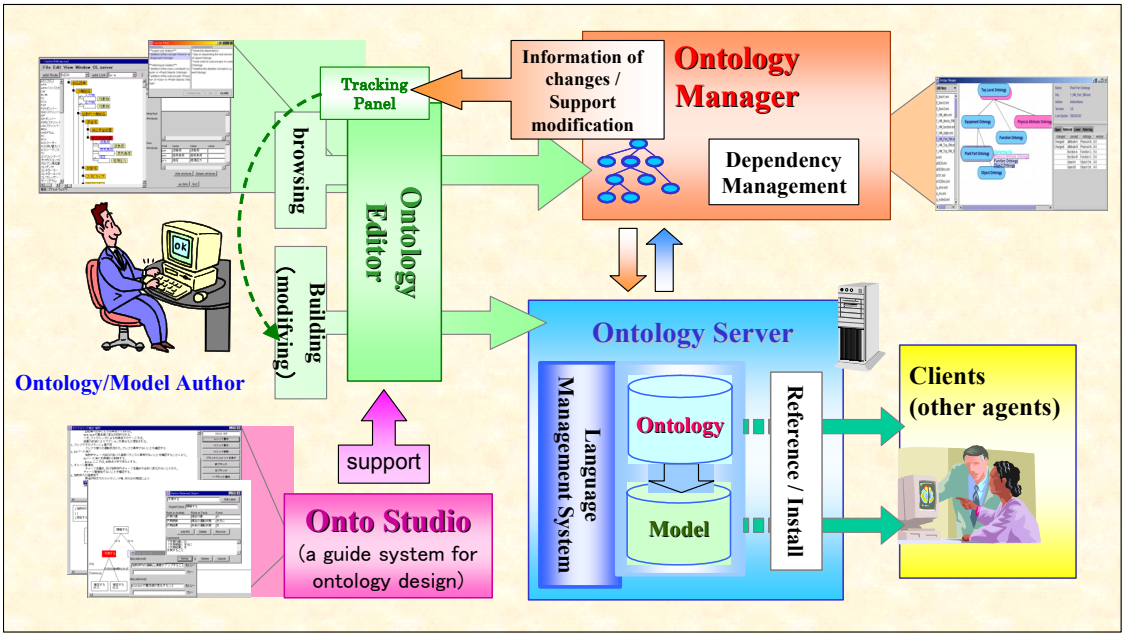


Fig.1 Hozo: an Environment for Building/Using Ontologies

target ontology-based on their dependencies. And when a change of some ontology influences others, it supports modification of influenced ontology for keeping its consistency [Sunagawa 03] .

Onto-Studio is based on a method of building ontologies, named AFM (Activity-First Method) [Mizoguchi 95]. It helps users design an ontology from technical documents. It consists of 4 phases and 12 steps [kozaki 02].

Ontology Server manages the storage and use of ontologies and models. Models are built by choosing and instantiating concepts in the ontology and by connecting the instances by defining specific relation among them. Hozo also checks the consistency of the model using the axioms defined in the ontology.

The latest version of this ontology editor is published at the URL: http://www.hozo.jp .

2.2. Ontological theories of Hozo

Several ontology development environments have been already developed [Farquhar 96, Noy 01, Staab 01]. Most of the tools are based on a frame-based knowledge representation language with an additional functionality for writing axioms. Hozo is similar to them in that sense, but is different from them in some respects:

1. Clear discrimination among a ***role-concept*** (husband role), a ***role-holder*** (husband) and a ***basic concept*** (man) is done to treat “Role” properly.
2. Management of the correspondence between a ***wholeness concept*** and a ***relational concept***.

In this section, we outline these ontological theories.

1) Basic concept, role concept and role holder

John Sowa introduces the *firstness* and the *secondness* of concepts [Sowa 95]. The former is roughly defined as a concept which can be defined without mentioning other

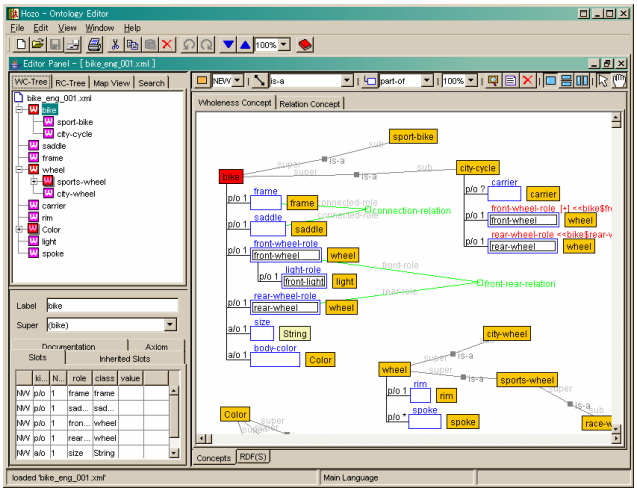


Fig.2. A snapshot of Ontology Editor

concepts. Examples include ion, a man, a tree, etc. The latter is roughly defined as a concept which cannot be defined without mentioning other concepts. Examples include wife, husband, student, child, etc. We call concepts of the *secondness* type except artifacts ***role-concepts*** in this paper. Based on his theory, we identified three categories for a concept. That is, a *basic concept*, a *role-concept*, and a *role holder*.

A ***role-concept*** represents a role which a thing plays in a specific context and it is defined with other concepts. On the other hand, a ***basic concept*** does not need other concepts for being defined. An entity of the basic concept that plays a role such as husband role or wife role is called a ***role holder***. For example in "a bicycle", its wheel plays the role as a front wheel ("a front wheel role") or a role that steers its body ("a steering role"), which is defined as a *role-concept*. A wheel that plays these roles is called "a front wheel" and "a steering wheel", respectively, which are *role holders*.

In ontological theory the role concept is very important and discussed by many researchers [Guarino 98]. We discuss how to organize role concepts and suggest a framework for organizing role-concepts in their hierarchy according to their context dependencies[Sunagawa 04].

2) Relational concept and wholeness concept

There are two ways of conceptualizing a thing. Consider a "brothers" and a "brotherhood". "The Smith brothers" is a conceptualization as a *concept*, on the other hand "brotherhood between Bob and Tom" is conceptualized as a *relation*. On the basis of the observations that most of the things are composed of parts and that those parts are connected by a specific relation to form the whole, we introduced **"wholeness concept"** and **"relational concept"**.

The former is a conceptualization of the whole and the latter is that of the relation. In the above example, the "brothers" is a *wholeness concept* and the "brotherhood" is a *relational concept*. Because a *wholeness concept* and a *relational concept* are different conceptualizations derived from the same thing, they correspond to each other. Theoretically, every thing that is a composite of parts can be conceptualized in both perspectives as a *wholeness concept* and a *relational concept*.

2.3. Creating Ontology-based Applications

Hozo provides several functions to develop applications based on ontologies and instance models which is built by its Ontology Editor. It helps the users to develop ontology-based applications by the following two ways.

1) Export of ontologies and models into different formats for another application

Hozo can translate the ontologies and models into different formats/languages (hierarchical text, XML/DTD, DAML+OIL, RDF(S), and OWL) that make them portable and reusable. Users develop their own applications to import these ontologies and models utilizing existing tools which speak these languages.

Hozo is used as a program to build and manage data of ontologies and models for the applications consistently. The use process is supported by its functionality, such as

getConceptList( ) : to get the list of all defined concepts
getInstanceList( ) : to get the list of all defined instances
findConcept(String *lbl*) : to find the concept whose label is *lbl*
findInstance(String *lbl*) : to find the instance whose label is *lbl*
getLabel( ) : to get the label of the concept
getSlots( ) : to get the slots of the concept

Table.1 Some methods of HozoAPI

building ontologies and models, dependency management of them, and checking the consistency of the model using the axioms defined in the ontology.

2) Access using HozoAPI

The operational functions which Hozo provides are open to the public as API implemented in Java, it is named HozoAPI. Using the API other systems can use some functionalities of Hozo. It has about 30 functions, necessary for applications to use ontology and instance models. Table.1 shows some methods of the API. The users can implement their systems easily to use these basic functions for operation of ontologies and models.

3. An Example of Ontology-based Application

In this section, we give an example of development of an ontology-based application with Hozo. We describe how ontology is built and used in the project of Structuring Nanotechnology Knowledge[NMC].

3.1. The project of Structuring Nanotechnology Knowledge

The research of nanotechnology is extended in various domains, and each domain intertwines with each other closely. Therefore, sharing the knowledge in common among different domains contributes to facilitate research in each domain through cross fertilization.

In this back-ground, the Structuring Nanotechnology Knowledge project, which is a NEDO (Japanese New Energy and Indus-trial Technology Development Organization) funded national project[NMC], has been carried out. The goal of the project is to build a material-independent platform for supporting development of innovative nano-materials, which is called "Structured Knowledge Platform for Nano-materials and Products" (Fig.3). It is not a database, a set of simulation tools or a knowledge base, but is an integrated environment composed of structured knowledge supported by advanced IT.

3.2. The Role of Ontology

Among many factors, the authors have been involved in building ontology of nanotechnology and its application to knowledge systematization. The key issues of knowledge structuring include how to harmonize different terminologies and viewpoints of the respective domains and how to interface end users with the platform.

Ontology of nanotechnology plays a role of glue for seamless connection among different domains, users and the platform, since it provides a conceptual infra-structure of nanotechnology and a unified framework in which functional knowledge for conceptual design of

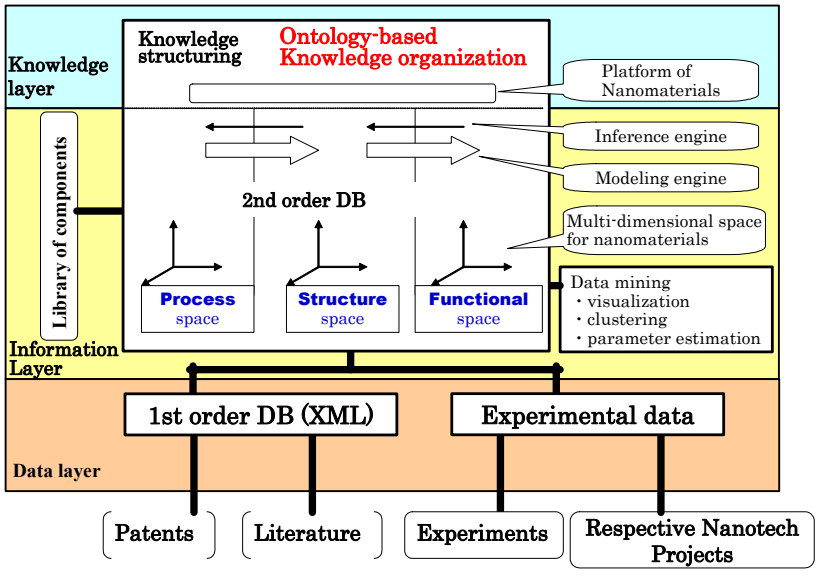


Fig.3. The Structured Knowledge Platform for Nano-materials and Products [NMC]

nanotechnology-made materials and devices and their realization processes.

3.3. Connection between Resources

The Structured Knowledge Platform consists of many resources as follows: concept design systems for "Process", "Structure" and "Function" of nano-materials, simulators, electronic textbook, scientific papers/patents database, and so on. These are developed as web-based systems. But the domain and grain size of each resources are different. Therefore, we need a consistent framework to interoperate them with each other.

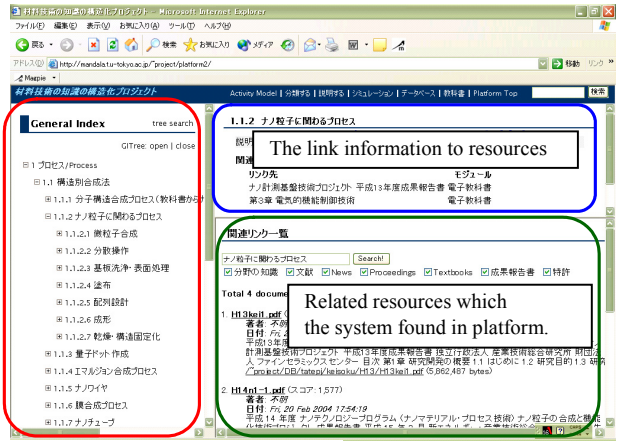
We plan to connect resources through common vocabulary which is defined in our project. We outline the mechanism of the connection as follows.

1) Develop an ontology of nanotechnology as common vocabulary in the project.
2) Connect keywords which appear in resources with the common vocabulary. And then, build a collection of links about the concepts which each vocabulary implies.
3) Connect between resources to follow the links through common vocabulary

3.4. Implementations using Hozo

We developed the system using Hozo with the following steps:

1) Experts of nanotechnology built a preliminary ontology of nanotechnology based on keywords which are extracted from textbooks, papers and patents. We call the ontology "General Index". It is constructed from the name of concepts in an is-a hierarchy. The number of the concepts is about 2,300.
2) We inputted the data of General Index into Hozo and made it to be accessible through internet. Experts accessed it and add "link information" between concepts in General Index and these in their resources in the platform using Ontology Editor of Hozo. The "link information" consists of its name, the kind of link and the hyper link to the resources. They are managed by Ontology Server of Hozo and edited by different experts in a distributed environment.
3) Hozo exports General Index with "link information" in a simple hierarchical XML format.
4) The system to show General Index read the exported XML data and shows the hierarchy of General Index in web-browser. Figure 4 shows a snapshot of the system. General Index is presented in a tree-format on the left side, and the user can brows its hierarchy by mouse operation. When he/she selects a concept in it, then the system shows "link information" edited by experts and related resources which the system found in the platform. This prototype system is implemented by JavaScript.



The hierarchy of General Index

Fig.4. A snapshot of the system to show General Index

This system is used as an index page of the Structured Knowledge Platform for Nano-materials and Products. We plan to improve it and develop a knowledge portal for the platform.

4. Conclusion and Future work

We discussed an environment for ontology development, Hozo, concentrating mainly on how to develop ontology-based application using it. Hozo helps developers to develop ontology-based applications in the following

two ways:

1) Translation of ontologies and models into different formats (hierarchical text, XML/DTD, DAML+OIL, RDF(S), and OWL) for another application

2) Access to Hozo using API implemented in Java

We also discussed an example of ontology-based application developed using Hozo in the project of Structuring Nanotechnology Knowledge. It is developed by the way of above 1). Some applications are built using Hozo, such as idea creation support system for materials design [kozaki 03], an interface system for an oil-refinery plant operation [Mizoguchi 00].

We have identified some room to improve Hozo through its extensive use. The following is the summary of our future plan:

-Ontological organization of various role-concepts.

-Management of ontologies and instance models by version control, updating and reusing.

-Improvement of ontology development method based on ontological theory of role-concept.

-Augmentation of the axiom definition and the language.

-Import from different formats (RDF(S), OWL .etc.)

-Gradable support functions according to a user's level of skill.

Acknowledgments

This work was supported by the New Energy and Industrial Technology Development Organization (NEDO). We are grateful to Dr E. Watanabe, Ms. Y. Yamagata and Mr. T. Hayashi for their valuable discussions.

References

[Guarino 98] Guarino, N.: Some Ontological Principles for Designing Upper Level Lexical Resources. Proc. of the First International Conference on Lexical Resources and Evaluation, Granada, Spain, 28-30, May 1998.

[Farquhar 96] Farquhar, A., Fikes, R. and Rice, J.: The Ontolingua Server: a Tool for Collaborative Ontology Construction, Proceedings of the 10th Banff Knowledge Acquisition Workshop, 1996

[Kozaki 00] Kozaki, K., et al: Development of an Environment for Building Ontologies which is based on a Fundamental Consideration of "Relationship" and "Role":PKAW2000, pp.205-221, Sydney, Australia, December, 2000

[Kozaki 02] Kozaki K.. et al.: Hozo: An Environment for Building/Using Ontologies Based on a Fundamental Consideration of "Role" and "Relationship", Proc. of the 13th International Conference Knowledge Engineering and Knowledge Management(EKAW2002), pp.213-218, Siguenza, Spain, October 1-4, 2002

[Kozaki 03] Kozaki K.. et al.: Systematization of Nanotechnology Knowledge Through Ontology Engineering - A Trial Development of Idea Creation Support System for Materials Design based on Functional Ontology, The Second International Semantic Web Conference (ISWC2003), Sanibel Island, FL, USA, Oct. 2003

[NMC] Nanomaterial Center, Institute of Engineering Innovation, School of Engineering, The University of Tokyo, http://mandala.t.u-tokyo.ac.jp/

[Mizoguchi 00] Mizoguchi, R., Kozaki, K., Sano, T., and Kitamura, Y.: Construction and Deployment of a Plant Ontology, 12th International Conference on Knowledge Engineering and Knowledge Management, Juan-les-Pins, French Riviera, October, 2000.

[Noy 01] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, and M. A. Musen: Creating Semantic Web Contents with Protege-2000, IEEE Intelligent Systems 16(2), 60-71, 2001

[Sowa 95] John F. Sowa: Top-level ontological categories, International Journal of Human and Computer Studies, 43, pp.669-685, 1995

[Staab 01] S. Staab, R Studer, H. P. Schnurr and Y. Sure: Knowledge Processes and Ontologies, IEEE Intelligent Systems 16(1), 2-9, 2001

[Sunagawa 03] Sunagawa, E., Kozaki, K., Kitamura, Y., and Mizoguchi, R.: An Environment for Distributed Ontology Development Based on Dependency Management, in Proc. of the 2nd International Semantic Web Conference (ISWC2003), pp. 453–468, 2003

[Sunagawa 04] Sunagawa, E., Kozaki, K., Kitamura, Y., and Mizoguchi, R.: Organizing *Role-concepts* in Ontology Development Environment: Hozo, AI Technical Report (Artificial Intelligence Research Group, I. S. I. R., Osaka Univ.), AI-TR-04-1, 2004