Hozo: An Environment for Building/Using Ontologies Based on a Fundamental Consideration of “Role” and “Relationship”

Kouji Kozaki[*], Yoshinobu Kitamura[*], Mitsuru Ikeda[*], and Riichiro Mizoguchi[*]

[*] The Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka, 567 -0047 Japan
Tel: +81-6-6879-8416, Fax: +81-6-6879-2123,
E-mail:{kozaki,kita,ikeda,miz}@ei.sanken.osaka-u.ac.jp

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.

Introduction

Building an ontology requires a clear understanding of what can be concepts with what relations to others. Although several tools for building ontologies have been developed to date, few of them were based on enough consideration of an ontological theory. We argue that a fundamental consideration of these ontological theories is needed to develop an environment for developing an ontology. We discuss mainly “role concept” and “relationship”, and consider how these ontologically important concepts should be treated in our environment. On the basis of the consideration we have developed an environment for building and using ontologies, named “Hozo”. This paper presents an outline of the functionality of Hozo. The next section discusses a *role-concept* and *a relation concept* in Hozo. Section 3 outlines the architecture of Hozo. Section 4 presents the implementation of Hozo and examples of its use. Next we discuss conclusions and some future work.

A consideration of “Role” and “Relation”

What is a role? : 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 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*.

Dependency analysis of role-concepts

There are various roles dependent on the whole, a relation, a task or a domain, and roles in artifacts, and so on. For building an ontology, it is important to discriminate among a role concept, a role holder and a basic concept. To give a guideline for such discrimination, we organized domain concepts which role concepts dependent on. In this paper we extracted 5 top-level categories and about 300 domain concepts from technical documents about oil-refinery plant operation [Mizoguchi 00]. Those top-level categories are as follows:

- **Device**: components of the plant.
- **Target object**: objects which a device targets in processing.
- **Attribute**: attributes of devices or target objects.
- **Domain activity**: behaviors and functions of devices.
- **Condition/Feature vocabulary**: vocabulary of condition and feature of devices or objects.

We do not claim the concepts listed in Table 1 are exhaustive. However, we carefully analyzed our domain, oil-refinery plant operation, and came up with domain concepts which role concepts dependent on from each categories, and classified them into 27 concepts in all. Although they might look domain dependent, the authors believe the dependency on the oil refinery domain is small, which is partially demonstrated by the concepts which are not from the oil-refinery plant domain.

Target object

- functions which the object receive:
e.g.) remained ingredient, combustion gas, reflux object, exhaust gas, drinking water

- a name of the whole device which has the object as input/output:
e.g.) decomposition device material, radiator water

- a name of place (a part of a device):
e.g.) side reflux, top reflux

- roles dependent on functions of the object:
e.g.) cooling medium, solvent medium, diluting medium, catalyst (catalytic agent), cleaner (cleaning material)

- roles against a device:
e.g.) input object, output object, raw materials

- time (temporal ?) position in a production process:
e.g.) intermediate product, finished product

(a)

Device

- its physical relationship with other devices (structure):
e.g.) pre flash drum, front wheel, rear wheel

- functions which the device has:
e.g.) heating furnace, draw pump, steering wheel, stay bar

- a name of the device which it is attached to:
e.g.) bypass valve, radiator hose

- target attribute for its function:
e.g.) liquid level control valve

- target object for its function:
e.g.) crude dram, off-gas compressor

- way of achievement which was applied to the device:
e.g.) atmospheric-pressure distillation device, reduced-pressure distillation device

(b)

Table 1. The domain concepts which role concepts dependent on. (part)

Relation 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 **“relation 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 *relation concept*. Because a *wholeness concept* and a *relation 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 *relation concept*.

Hozo

We have developed an environment, named “Hozo[1]”, for building/using ontologies based on fundamental ontological theories. “Hozo” is composed of “Ontology Editor”, “Onto-Studio” and “Ontology Server”(Fig.1). The ontology and the resulting model are available in different formats (Lisp, Text, XML/DTD,DAML+OIL) that make it portable and reusable.

Ontology Editor provides users with a graphical interface, through which they can browse and modify ontologies by simple mouse operations. It treats “role concept” and “relation” on the basis of fundamental consideration discussed in section 2 [Kozaki 00]. This interface consists of the following four parts (Fig.2):

1. ***Is-a* hierarchy browser** displays the ontology in a hierarchical structure according to only *is-a* relations between concepts.

2. **Edit panel** is composed of a *browsing panel* and a *definition panel*. The former displays the concept graphically, and the latter allows users define a

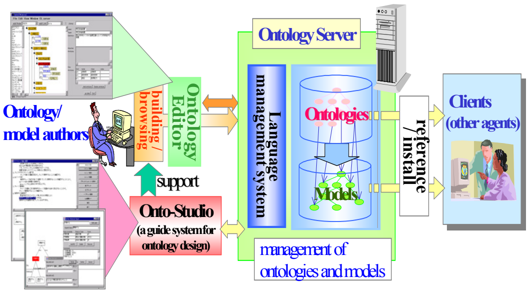
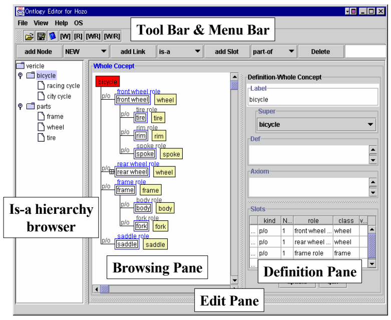


Fig. 1. The architecture of Hozo



	kind	N...	role	class	v...
...	p/o	1	front wheel ...	wheel	
...	p/o	1	rear wheel ...	wheel	
...	p/o	1	frame role	frame	

Fig. 2. A snapshot of Ontology Editor

[1] “Ho” is a Japanese word and means unchanged truth, laws or rules in Japanese, and we represent “ontologies” by the word. “Zo” means to build in Japanese.

selected concept in the *is-a* hierarchy browser.

3. **Menu bar** is used for selecting tools
4. **Tool bar** is used for selecting commands

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. Figure.3 shows the skeletal building process of ontologies using Onto-Studio. It consists of 4 phases and 12 steps. The followings outline these 4 phases.

1. ***Extraction of task-units***: In this phase, users extract **task-units** which contain only one process from technical documents which are written in natural language.
   (1) Divide technical documents into small **blocks** to extract vocabulary easier.
   (2) Extract **task-units** which contain only one process from these blocks.
   (3) Make each task-unit a flow chart which is called **concrete task-flow**.
2. ***Organization of task-activities***: In this phase, users specify input/output of task-activities and organize task-activities.
   (4) Conceptualize **task-activities** from verbs in task-units.
   (5) Organize task-activities in an *is-a* hierarchy.
   (6) Define role-concepts, called **task-activity roles**, which appear in input/output of these task-activities.
3. ***Analysis of task- structure***: In this phase, users analyze flow of task-activities, specify flow of objects from input to output, and define task-context-roles.
   (7) Generalize concrete task-flows to obtain **general task-flows**.
   (8) Describe **object-flows**, which clearly express relations between inputs and outputs of task-activities, in the general task-flows obtained above.
   (9) Define **task-context roles** on the basis of these object-flows. By task-context roles, we mean role-concepts dependent on the whole process of a task.
   (10) Extract **domain terms** which play the task-context role.
4. ***Organization of domain concepts***: In this phase, users organize domain concepts extracted in phase 3.

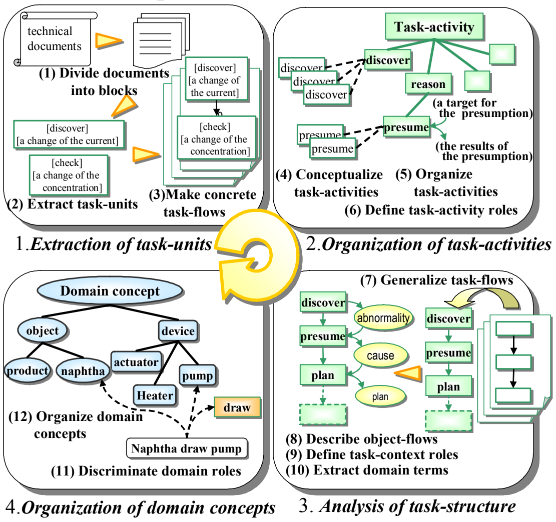
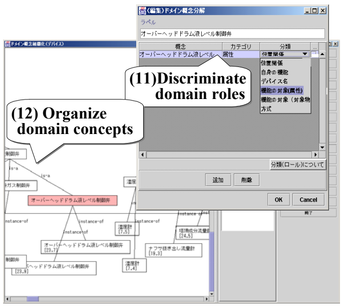

technical documents
(1) Divide documents into blocks
[discover] [a change of the current]
[check] [a change of the concentration]
(2) Extract task-units
(3)Make concrete task-flows
1.Extraction of task-units
Task-activity
discover
discover
discover
discover
reason
(a target for the presumption)
presume
presume
presume
(the results of the presumption)
(4) Conceptualize task-activities
(5) Organize task-activities
(6) Define task-activity roles
2.Organization of task-activities
Domain concept
object
device
product
naphtha
actuator
pump
Heater
draw
(12) Organize domain concepts
Naphtha draw pump
(11) Discriminate domain roles
4.Organization of domain concepts
(7) Generalize task-flows
discover
abnormality
presume
cause
plan
plan
discover
presume
plan
(8) Describe object-flows
(9) Define task-context roles
(10) Extract domain terms
3. Analysis of task-structure

Fig. 3. The building process of ontologies using Onto-Studio.


(編集)ドメイン概念分解

ラベル

オーバーヘッドドラム液レベル制御弁

概念	カテゴリ	分類
オーバーヘッドドラム液レベル	属性	位置関係

ドメイン概念細分化(デバイス)
位置関係
自身の機能
デバイス名
機能の対象(属性)
機能の対象（対象物
方式
分類(ロール)について
追加
削除
OK
Cancel
(11)Discriminate domain roles
(12) Organize domain concepts
is-a
is-a
オーバーヘッドドラム液レベル制御弁
instance-of
instance-of
instance-of
instance-of
温度計 [7.5]
塔頂成分流量計 [24.5]
オーバーヘッドドラム液レベル制御弁 [23.7]
ナフサ抜き出し流量計 [19.3]
温度計 [7.4]
ヘッドドラム液レベル制御弁 [23.9]

Fig. 4. A snapshot of Onto-Studio

(11) Discriminate between roles dependent on domain concepts and basic concepts.

(12) Organize domain concepts in an *is-a* hierarchy.

In practice these steps are not done in a water fall manner. Users can go back and forth during the process. In each step Onto-Studio provides users with graphical interfaces to help them perform suggested procedures. For example, Figure.4 shows a window to help users discriminate domain roles from basic concepts.

Implementation and application

The current version of Ontology Editor has been implemented in Java2 (JDK1.3) and been used for five years not only by our lab members but also by some researchers outside [Jin 99, Inaba 00, Barros 01,Kitamura 01].

Here we give more detail of the plant ontology [Mizoguchi 00]. The plant model contains a remarkable fact that multiple names are used to denote the same entity. Let us take an example shown in Fig.5 in which two controllers exist: Level controller (LC29) and flow controller (FC29). Both controllers use the same control valve (VFC29) as an actuator. The control valve VFC29 is called by a different name depending on which controller the operator focuses on. In Hozo, this example is represented that the basic concept "control valve" plays multiple roles depending on the context (Fig.6).

Role concept analysis and its use in helping users extract role concept from a set of domain concepts have been investigated on the basis of our experience in the development of a plant ontology described above. In order to see the performance of Onto-Studio, we restructured the plant ontology from the same technical documents we used at the first time. As a result, we extracted 355 task-units and restructured a task ontology which is consists of 36 task-activities. Based on the task ontology, we obtained 5 general task-flows and extracted 356 domain concepts. A domain ontology consists 190 basic concepts which were discriminated from the role concepts. As a consequence of this restructuring, we identified about 20 errors in role concept extraction in the original ontology. This result suggests that Onto-Studio can provide a good support in building an ontology.

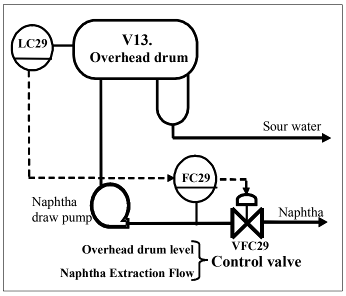
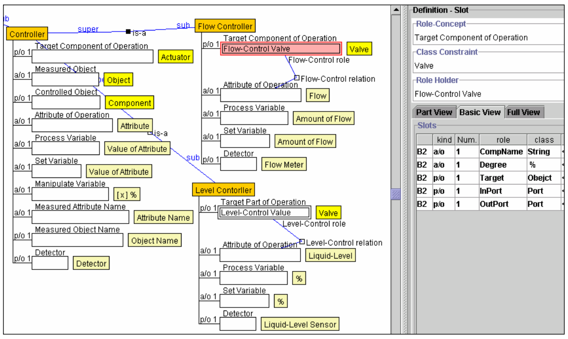


Fig.5. Cascaded control of LC and FC



Role-Concept
Target Component of Operation

Class Constraint
Valve

Role Holder
Flow-Control Valve

Slots

	kind	Num.	role	class
B2	a/o	1	CompName	String
B2	a/o	1	Degree	%
B2	p/o	1	Target	Obejct
B2	p/o	1	InPort	Port
B2	p/o	1	OutPort	Port

Fig.6. A snapshot of the plant ontology definition about *Controller*

Conclusion and Future work

We discussed an environment for ontology development, Hozo, concentrating mainly on how its Hozo treats *role-concepts* and *wholeness/relation concepts*. Several ontology development environments have been already developed. 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 ***relation concept***.
3. A guide system for building an ontology based on task/domain role concept.

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

- Ontological organization of various role-concepts.
- Augmentation of the axiom definition and the language.
- Gradable support functions according to a user's level of skill.
- Improvement of Onto-Studio by applying in more practical examples.

References

[Barros 01] Barros, B., Mizoguchi, R., et al.: A Platform for Collaboration Analysis in CSCL: An ontological approach, Proceedings of AIED01, San Antonio, Texas, May 19-23 2001

[Kitamura 01] Kitamura, Y., Sano, T., Namba, K. and Mizoguchi, R.: A Functional Concept Ontology and Its Application to Automatic Identification of Functional Structures, Advanced Engineering Informatics (Artificial Intelligence in Engineering), to appear, 2002.

[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

[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.

[Inaba 00] Inaba, A., Thepchai, S., Ikeda, M., Mizoguchi, R., and Toyoda, J.: An overview of "Learning Goal Ontology", Proc. of ECAI2000, pp.23-30, Berlin, Germany, 2000

[Jin 99] Jin, L., Chen, W., Hayashi, Y., Ikeda, M., Riichiro Mizoguchi, R., et al. :An Ontology-Aware Authoring Tool - Functionalstructure and guidance generation -, Proc. of AIED'99

[Mizoguchi 95] Mizoguchi, R., Ikeda, M., Seta, K. et al.: Ontology for Modeling the World from Problem Solving Perspectives, Proc. of IJCAI-95, pp. 1-12, 1995.

[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.

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