Ontological Engineering: Foundation of the next generation knowledge processing

Riichiro Mizoguchi

ISIR, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047 Japan
Email: miz@ei.sanken.osaka-u.ac.jp

Abstract. Ontological engineering as a key technology of the next generation knowledge processing is discussed. After a brief introduction to ontological engineering with my speculation about its potential contribution, three major results of the practice of ontological engineering in my lab are presented. Then, paradigm shift in information processing is discussed followed by a future directions in the Web intelligence context.

1. Introduction

In AI research history, we can identify two types of research. One is "Form-oriented research" and the other is "Content-oriented research". The former investigates formal topics like logic, knowledge representation, search, etc. and the latter content of knowledge. Apparently, the former has dominated AI research to date. Recently, however, "Content-oriented research" has attracted considerable attention because a lot of real-world problems to solve such as knowledge sharing, facilitation of agent communication, meta-data, semantic web, large-scale knowledge bases, etc. require not only advanced formalisms but also sophisticated treatment of the content of knowledge before it is put into a formalism.

Formal theories such as predicate logic provide us with a powerful tool to guarantee sound reasoning and thinking. It even enables us to discuss the limit of our reasoning in a principled way. However, it cannot answer any of the questions such as what knowledge we should prepare for solving the problems given, how to scale up the knowledge bases, how to reuse and share the knowledge, how to manage knowledge and so on. In other words, we cannot say it has provided us with something valuable to solve real-world problems.

In expert system community, the knowledge principle[Feigenbaum, 77] proposed by Feigenbaum has been accepted and a lot of development has been carried out with a deep appreciation of the principle, since it is to the point in the sense that he stressed the importance of accumulation of knowledge rather than formal reasoning or logic. This has been proved by the success of the expert system development and a lot of research activities have been done under the flag of “knowledge engineering”. However, the author is not claiming the so-called rule-base technology is what we need for future knowledge processing. Rather, in order to adapt to the rapid change of the situation, treatment of knowledge should be in-depth analyzed. Advanced knowledge processing technology should cope with various knowledge sources and elicit, transform, organize, and translate knowledge to enable the agents to utilize it.

Although importance of such "Content-oriented research" has been gradually recognized these days, we do not have sophisticated methodologies for content-oriented research yet. In spite of much effort devoted to such research, major results were only development of KBs. We could identify the reasons for this as follows:

a) It tends to be ad-hoc, and

b) It does not have a methodology which enables knowledge to accumulate.

It is necessary to overcome these difficulties in order to establish the content-oriented research or content technology. Ontological Engineering has been proposed for that purpose. It is a research methodology which gives us design rationale of a knowledge base, kernel conceptualization of the world of interest, semantic constraints of concepts together with sophisticated theories and technologies enabling accumulation of knowledge which is dispensable for knowledge processing in the real world. The author believes knowledge management essentially needs content-oriented research. It should be more than information retrieval. The content technology should be more sophisticated and powerful to realize the true knowledge management.

The objective of this paper is to discuss how ontological engineering[Mizoguchi 97] has emerged and how it will contribute to the future knowledge processing together with a brief history of the author's research activities on those topics.

2. What is an ontology and what is ontological engineering?

Ontological engineering is a successor of knowledge engineering which has been considered as a technology for building knowledge-intensive systems. Although knowledge engineering has contributed to eliciting expertise, organizing it into a computational structure, and building knowledge bases, AI researchers have noticed the necessity of a more robust and theoretically sound engineering which enables knowledge sharing/reuse and formulation of the problem solving process itself. Knowledge engineering technology has thus developed into “ontological engineering” where “ontology” is the key concept to investigate. Roughly speaking, ontology consists of **task ontology**[Mizoguchi 95a] which characterizes the computational architecture of a knowledge-based system which performs a task and **domain ontology** which characterizes the domain knowledge where the task is performed. By a task, we mean a problem solving process like diagnosis, monitoring, scheduling, design, and so on. The idea of task ontology which serves as a theory of vocabulary/concepts used as building blocks for knowledge-based systems might provide us with an effective methodology and vocabulary for both analyzing and synthesizing knowledge-based systems.

An ontology is understood to serve as a kernel theory and building blocks for content-oriented research. Definitions of an ontology are presented below:

a) In philosophy, it means *theory of existence*. It tries to explain what exists in the world and how the world is configured by introducing a system of critical categories to account for things and their intrinsic relations.

b) From an AI point of view, an ontology is defined as "explicit specification of conceptualization" [Gruber].

c) From a knowledge-based systems point of view, it is defined as "a theory (system) of concepts/vocabulary used as building blocks of information processing systems" [Mizoguchi 95a].

d) Another definition [Gruber]: Ontologies are agreements about shared conceptualizations. Shared conceptualizations include conceptual frameworks for modeling domain knowledge; content-specific protocols for communication among inter-operating agents; and agreements about the representation of particular domain theories. In the knowledge sharing context, ontologies are specified in the form of definitions of representational vocabulary.

e) A compositional definition: An ontology consists of concepts with definitions, hierarchical organization of them, relations among them (more than *is-a* and *part-of*), and axioms to formalize the definitions and relations.

f) Yet another definition: An ontology is an explicit specification of objects and relations in the target world intended to share in a community and to use for building a model of the target world.

Why ontology instead of knowledge? Knowledge is domain-dependent, and hence knowledge engineering which directly investigates such knowledge has been suffering from rather serious difficulties, such as domain-specificity and diversity. Further, much of the knowledge dealt with in expert systems has been heuristics that domain experts have, which makes knowledge manipulation more difficult. However, in ontological engineering, we investigate knowledge in terms of its origin and elements from which knowledge is constructed. An ontology reflects what exists out there in the world of interest or represents what we should think exists there. An ontology is essentially designed to be objective and shared by many people. Hierarchical structure of concepts and decomposability of knowledge enable us to identify portions of concepts sharable among people. Exploitation of such characteristics makes it possible to avoid the difficulties knowledge engineering has faced with. The following is a list of the merits we can enjoy from an ontology:

a) *A common vocabulary*. The description of the target world needs a vocabulary agreed among people involved.

b) *Explication* of what has been often left implicit. Any knowledge base built is based on a conceptualization possessed by the builder and is usually implicit. An ontology is an explication of the very implicit knowledge. Such an explicit representation of assumptions and conceptualization is more than a simple explication. Its contribution to knowledge reuse and sharing is more than expectation considering that the implicitness has been one of the crucial causes of preventing knowledge sharing and reuse.

c) *Systematization* of knowledge. Knowledge systematization requires well-established vocabulary/concepts in terms of which people describe phenomena, theories and target things under consideration. An ontology thus contributes to providing a backbone for the systematization of knowledge.

d) *Standardization.* The common vocabulary and knowledge systematization bring us more or less standardized terms/concepts. Standardization has to be taken not as restriction of free exploration of research mind but as a minimum set of shared terms/concepts among human and computer agents who can communicate with each other thanks to them.

e) *Meta-model functionality.* A model is usually built in the computer as an abstraction of the real target. And, an ontology provides us with concepts and relations among them which are used as building blocks of the model. Thus, an ontology specifies the models to build by giving guidelines and constraints which should be satisfied. This function is viewed as that at the metalevel. This functionality suggests us the possibility of an "ontology-aware" authoring tool which can be very intelligent in the sense that it knows what model it is going to help authors build.

3. Some Experiences in Ontological Engineering

3.1 Functional Ontology and Knowledge Systematization

The first topic is on systematization functional knowledge in computer-aided design(CAD)[Mizoguchi 00a]. Knowledge systematization is indeed a topic of content-oriented research and is not that of a knowledge representation such as production rule, frame or semantic network. Although knowledge representation tells us how to represent knowledge, it is not enough for our purpose, since what is necessary is something we need before the stage of knowledge representation, that is, knowledge organized in an appropriate structure with appropriate vocabulary. This is what the next generation knowledge base building needs, since it should be principled in the sense that it is based on well-structured vocabulary with an explicit conceptualization of the assumptions. This nicely suggests ontological engineering is promising for the purpose of our enterprise.

While any scientific activity which has been done to date is, of course, a kind of knowledge systematization, it has been mainly done in terms of analytical formulae with analytical/quantitative treatment. As a default, the systematization is intended for human interpretation. Our knowledge systematization adopts another way, that is, ontological engineering to enable people to build knowledge bases on the computer as a result of knowledge systematization. The philosophy behind our enterprise is that ontological engineering provides us with the basis on which we can build knowledge and with computer-interpretable vocabulary in terms of which we can describe knowledge systematically in a computer-understandable manner.

By building a framework for knowledge systematization using ontological engineering, we mean identifying a set of backbone concepts with machine understandable description in terms of which we can describe and organize design knowledge for use across multiple domains. The system of concepts is organized as layered ontologies as is seen in Figure 1.

3.1.1 Functional modeling

No one would disagree that the concept of function should be treated as a first class category in design knowledge organization. That is, function is an important member of a top-level ontology of design world. One of the key claims of our knowledge systematization is that the concept of function should be defined independently of an object that can possess it and of its realization method. If functions are defined depending on objects and their realization, few functions are reused in different domains. As is well understood, innovative design can be facilitated by flexible application of knowledge or ideas across domains.

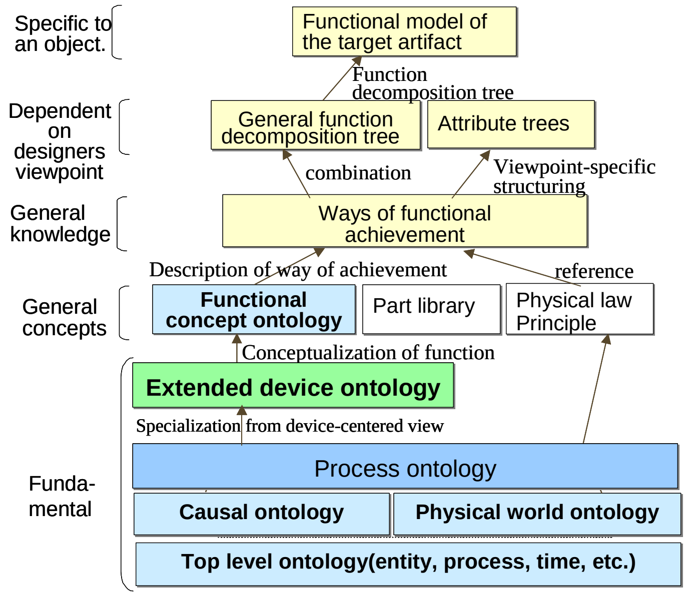
Functional representation has been extensively investigated to date [Sasajima 95] [Chandrasekaran 00] and a lot of functional representation languages are proposed with sample descriptions of functions of devices. However, because it is not well understood how to organize functional knowledge in what principle in terms of what concepts, most of the representation are ad-hoc and lack generality and consistency, which prevents knowledge from being shared. One of the major causes of the lack of consistency is the difference between the ways of how to capture the target world. For example, let us take the function of a super heater of a power plant, *to heat steam* and that of cam of cam&shaft pair, *to push up the shaft.* The former is concerned with something that comes in and goes out of the device but the latter with the other device that cannot be either input or output of the device. This clearly shows the fact that there is a difference in how to view a function according to the domain. The difference will be one of the causes of inconsistency in functional representation and non-interoperability of the knowledge when functional knowledge from different domains is put into a knowledge base.

The above observation shows that we need a framework which provides us with a viewpoint to guide the modeling process of artifacts as well as primitive concepts in terms of which functional knowledge is described in order to come up with consistent and sharable knowledge.

3.1.2 Hierarchy of functional knowledge and ontology

Figure 1 shows a hierarchy of functional knowledge built on top of fundamental ontologies. The lower layer knowledge is in, the more basic. Basically, knowledge in a certain layer is described in terms of the concepts in the lower layer. Top-level ontology defines and provides very basic concepts such as time, state, process and so on. Causal ontology specifies actions and causality against teleology. Physical world ontology specifies 3D space and entity to give axiomatic physical world with a state-based modeling reflecting a special world of design in which an entity(artifact) is created from nothing. These two ontologies contribute to “Symbol grounding” of higher-level concepts, that is, functional concepts. On top of these three, process ontology is introduced to specify natural processes or phenomena. Every device utilizes several natural phenomena to realize its functions.

Device ontology imposes a frame or viewpoint on an event to introduce a more engineering perspective. That is, it introduces the concepts of a black box equipped with input and output ports. Device ontology defines fundamental roles such as *agent, object, conduit and medium.* Although process ontology is more fundamental than device ontology, there are some cases where process ontology is directly employed to model real world events/phenomena instead of device ontology. Typical cases are found in modelling chemical processes for which device ontology is not appropriate. In summary, five ontologies(Top-level, causality, physical world, process and device



ontologies) collectively work as a substrate on which we can build consistent knowledge in layers.

Functional concept ontology specifies functional concepts as an instance of *function* defined in device ontology. The definitions are scarcely depends on the device, the domain or the way of its implementation so that they are very general and usable in a wide range of areas. Theories and principles of physics and abstract part library also belong to this class of knowledge called *general concept layer*.

Functional achievement way knowledge is such knowledge that represents various ways of achievement of a function. This knowledge is about *how*(in what way) a function is achieved, whereas the functional concept is about *what* the function is going to achieve. In other words, the former is formulated in terms of *whole-part relation* and the latter in terms of *is-a relation*. Although functional achievement way knowledge looks similar to functional decomposition like that discussed in [Pahl 88], the former is much richer than the latter in that it consists of four kinds of hierarchies of different roles and principles(*is-a* hierarchy, attribute tree, functional decomposition tree and general functional decomposition tree). The inherent structure of such knowledge is organized in an *is-a* hierarchy from which the other three structures are derived according to the requirement. The *is-a* structure is carefully designed identifying inherent property of each *way* to make it sharable and applicable across domains. One of the key issues in knowledge organization is clear and consistent differentiation of *is-a relation* from other relations such as *part-of*, *is-achieved-by*, etc. keeping what is the inherent property of the target thing in mind.

3.1.3 Roles and effects of functional ontology

The extended device ontology views an artifact as something that inputs, process and outputs *objects.* The *object* is something processed by the device during it goes through a device and hence it never be another device that cannot go through a device. This ontology imposes a proper viewpoint from which one can successfully model a mechanical system in a way consistent with those models of engineering artifacts produced in other domains. It is not an easy task to build models of a lot of artifacts in a consistent way. “A gear pair changes torque”, “A cam shrinks a spring” and “A cam pushes up a rod” are inconsistent with each other in the hidden computational models. While the first one is based on the extended device ontology, the latter two are based on a different ontology, say, inter-device operation ontology. The organization of knowledge including these models will lose consistency.

The extended device ontology allows us to build interoperable models and provides us with a guideline for modelling process by its ***role-assignment*** functionality which is the very source of consistency in functional knowledge organization. For example, the concept of a *conduit* helps us consistently recognize devices by taking it as the boundary between the devices. In the mechanical system domain, a shaft and a wire, which play the role of *conduit* in the mechanism level, enable us to identify each mechanism composed of mechanical elements. Models designed based on the extended device ontology has a high composability thanks to its localized description, that is, its independence of neighboring devices that are connected to each other only through attributes of an *object*. On the contrary, composability of inter-device operation ontology is low due to its high dependence on neighboring devices.

3.1.4 Use of functional concept ontology

Functional concept ontology provides us with necessary and sufficient operational terms used for representing functional knowledge/model together with constraints to be satisfied by them. The following is the list of our work on use of the ontology through to evaluate it. All the activities are of new type and different from the conventional knowledge base technology. No problem solving knowledge is treated. Instead, objective and fundamental knowledge is analyzed and modeled to enable a knowledge-based system to articulate the domain and hence to in-depth understand the fundamental knowledge to provide useful knowledge with designers. This is what we need intelligent functional knowledge management in CAD community.

a) **Explanation generation** at the functional level[Sasajima 95].
b) **Functional model description** of specific artifacts of many kinds[Kitamura 99a]
c) **Description of ways of functional achievement**: 104 ways for 26 functions found in five different artifacts(a washing machine, a printing device, slicing machines for ingot of semiconductors (using wire or rotating blade), and an etching device)
d) **Specification of the inference space** for functional reasoning[Kitamura 99b,00].
e) **Way knowledge server** for designer support.

3.2 Plant ontology for multi-agent plant operator support system

This section describes an activity of ontology construction and its deployment in Oil-refinery plant which has been done under the umbrella of Human-Media Project for five years, which is a MITI(Japanese Ministry of International Trade and Industries) funded national project, is intended to invent an innovative media technology for happier human life in the coming information society. Our ontology construction activities have been done in the project named “Development of a human interface for the next generation plant operation” running as a subproject of Human Media project[Mizoguchi 99, 00b].

The interface for oil-refinery plant operation has been developed intended to establish a sophisticated technology for advanced interface for plant operators and consists of Interface agent: IA, Virtual plant agent: VPA, Semantic information presentation agent: SIA, Ontology server: OS and Distributed collaboration infrastructure: DCI. The last two are mainly for issues related to system building, while the first three are related directly to interface issues. OS has been developed employing ontological engineering.

3.2.1 The role of a plant ontology

Any intelligent system needs a considerable amount of domain knowledge to be useful in a domain. The amount of knowledge necessary often goes large, which sometimes causes difficulties in the initial construction and maintenance phases. As described above, one of the methods we adopted to cope with such problems is ontological engineering. Roughly speaking, the essential contribution of the plant ontology is making shared commitment to the target plant explicit, and hence terminology is standardized within the community of agents. By agents, we also mean human agents, operators, to share such a fundamental understanding about the plant. This enables the system to communicate with operators using the terms stored in Ontology server: OS. It is the second major role of OS in the current implementation of the interface system which is discussed below.

In message generation, we need to pay maximal attention to word selection to make operators’ cognitive load minimum in message understanding. After an intensive interview with domain experts, we found human operators use different terms to denote the same thing depending on context. When we first noticed this fact, domain experts apologized for this seemingly random fluctuation of word usage, since they did not know the reason why they use terms that way and they were used to collaboration with computer engineers who do not like neat adaptation and tend to compel their idea of “this is what a computer can do, so accept it”. They kindly declared that they would soon determine a unique label for each thing. But, we were different from such computer engineers. Instead of accepting their proposal, we carefully analyzed the way of their word usage and finally came up with that it is not random except a few cases. Many of the wording have good justifications which have to be taken care of in the message generation.

The reasons why we employed distributed collaboration architecture with multiple agents include making the whole system robust and easy to maintain. As is well known, however, these merits are not free. We need a well-designed vocabulary for describing message content as well as a powerful negotiation protocol. Although the latter is of importance, it is out of the scope of this article. DCI is responsible for enabling collaborative problem solving by multiple agents with the help of OS. It is one of the key factors that domain-dependent knowledge be isolated in OS so that DCI can be as general as possible.

3.2.2 Plant ontology

The plant ontology we built consists of several hierarchical organizations of concepts such as *operation task, plant components, plant objects, basic attributes* and *ordinary attribute*. Because of the space limitation, only domain ontology is discussed here. The key issue in the design of an ontology is clear distinction essential categories from view-dependent concepts.

There exist two major things in the plant domain: ***Plant components(devices)*** and ***plant objects*** to be processed by the ***devices***. Domain concepts also have role concepts like task ontology does. To say precisely, many of the domain concepts are role concepts. The first things we have to do when designing a domain ontology is discrimination of roles concepts from essential categories (or basic concepts), i.e., view- or context-independent concepts. Let us first take ***plant object***. The top-level categories of ***plant object*** are ***view-independent object*** and ***view-dependent object***. The former includes LP gas, gasoline, naphtha, etc. which are categories persistent in

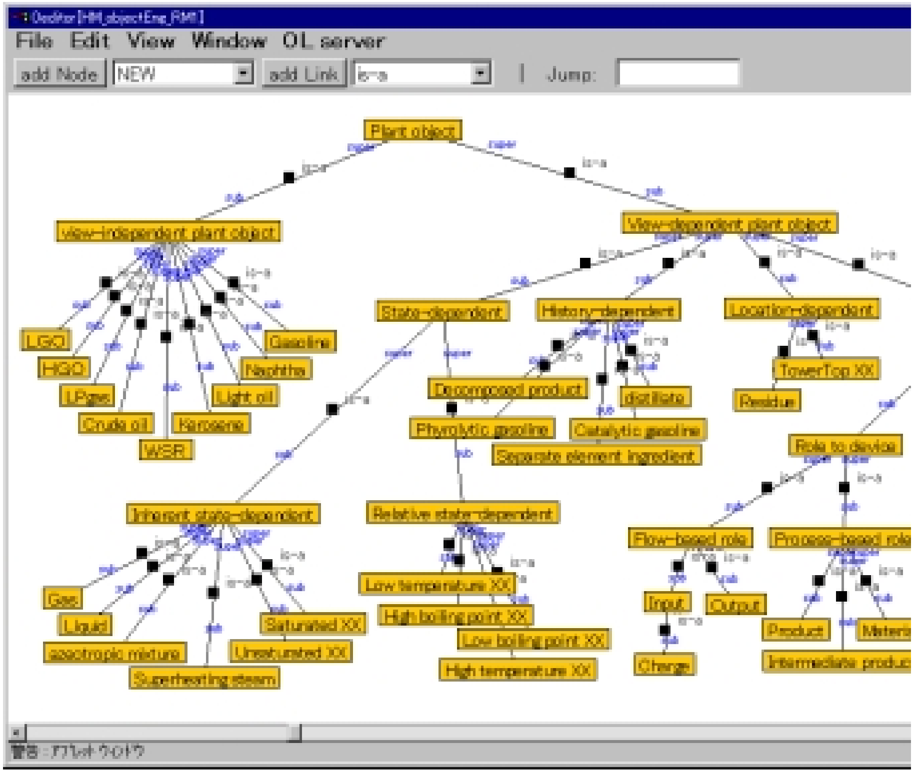


Figure 2 A Portion of Plant Object is-a Hierarchy in the Plant Domain Ontology.

any situation. The latter includes ***tower-head ingredient, liquid, distillate, input, intermediate product, raw material, fuel***, etc. All are view- or context-dependent. The major task needed was categorization of such dependency. Figure 2 shows a portion of ***plant object*** *is-a* hierarchy. The major categories of ***view-dependent plant object*** are ***state-dependent, location-dependent, history-dependent and role-dependent*** objects***.*** ***state-dependent*** objects has ***inherent state-dependent*** and ***relative state-dependent*** objects as its sub-concepts. The former includes ***liquid, gas, superheating steam*** etc. and the latter ***low temperature ingredient***, ***low boiling point ingredient***, etc.

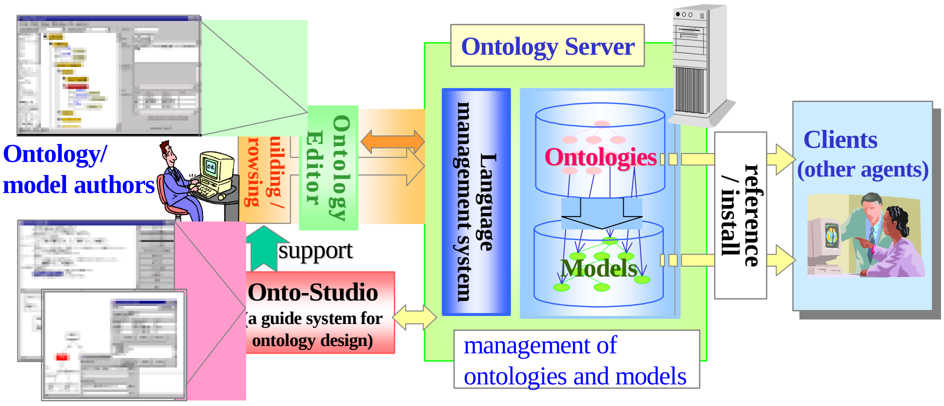
Attribute also needs careful treatment. Most of the attributes people think so are not true attribute but role attribute. Let us take an example of height. It is a role attribute whose basic attribute is length. Height, depth, width and distance are role attributes. Just like a man is called a husband when he has got married. The true attribute is called basic attribute. Examples of basic attribute include length, area, mass, temperature, pressure, volt, etc. Role attribute includes height, depth, input pressure, maximum weight, area of cross section, etc. Needless to say, these attributes are also decomposed into several sub-concepts.

We finally built an ontology which contains about 400 concepts which are approved by the domain experts and the coverage is around the normal pressure fractionator of a full-scale refinery plant. The model of the target refinery plant is built by instantiating the appropriate concepts in the ontology and connect them. The number of instances generated is about 2000. The ontology and the model of the target plant is stored in OS and served to other agents in the total prototype system. Evaluation was done by experts and we have got a favorable result.

3.3 Hozo: An integrated ontology development/use environment

Building an ontology requires a clear understanding of what can be concepts with what relations to others. An ontology thus focuses on “concepts” themselves rather than “representation” of them. Although several systems for building ontologies have been developed to date, they were not based on adequate consideration of an ontological theory which is why most of them are yet another KR languages. We believe that a fundamental consideration of ontological theories is needed to develop an environment for developing ontologies. 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 designed and have developed an environment for building and using ontologies, named “Hozo”.

“Hozo”[Kozaki 00] is composed of “Ontology Editor”, “Onto-Studio” and “Ontology Server”(See Figure 3). The Ontology Editor provides users with a graphical interface, through which they can browse, build and modify ontologies by simple mouse operations. This system manages attributes between concepts organized in an *is-a* hierarchy. The Onto-Studio is based on a method of building ontologies, named AFM (Activity-First Method)[Mizoguchi 95b], and it helps users design an ontology from technical documents. The Ontology Server manages the ontologies and models. Ontology Editor in Hozo have been extensively used in my lab for four years



and it was used to develop the plant ontology described above together with Ontology Server.

Because Hozo is implemented in Java and the ontology editor is an applet, it can work as a client through Internet. Hozo manages ontologies and models for each developer. For each ontology in Hozo, only its author can modify it, and the other users can only read and copy those developed by others. It lets share ontologies among users without explicit version control.

Models are built by choosing and instantiating concepts in the ontology and by connecting the instances. Hozo also checks the consistency of the model using the axioms defined in the ontology. The ontology and the resulting model are available in different formats (Lisp, Text, XML/DTD) that make them portable and reusable. The axiom contains constraints which part-concepts or attributes should satisfy, and

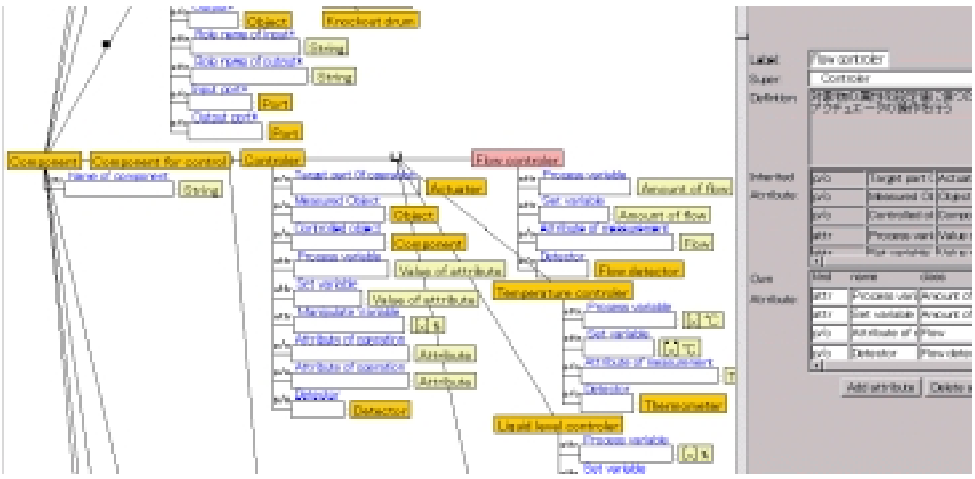


Figure 4 A snapshot of the plant ontology definition.

relations among the part-concepts. For example, these are constraints on the part-concept such as “any teachers must have a teaching certificate” in a “school”, and “the size of wheels are from 10 inch to 30 inch” in a “bicycle”. Another example is a constraint on the relation such as there must be a connection relation between a wheel and a frame in a “bicycle”. Figure 4 depicts a snapshot of plant ontology definition using Ontology editor where *Flow controller* as a subclass of *controller* is defined. A string above a blank rectangle represents a role name of an instance which should be put in the rectangle followed by colon and an dark rectangle which is a class constraints of the instance.

4. Paradigm shift

At the expert system time, people’s expectation was to build a stand-alone problem solving system which has a knowledge base of a domain expert’s heuristics to perform a specific task with similar or higher performance than that of the expert. That is, the main focus was put on dealing with subjective and specific knowledge for problem solving. However, situation has been changing. Most of the salient activities such as Web document search in the internet, Electronic commerce(EC), Electronic Data Interchange(EDI), Knowledge management(KM), STEP, XML tag design, etc. do require almost opposite characteristics to *knowledge*, that is, objective, general and sharable *knowledge* which is not necessarily tuned to problem solving.

We can summarize the major trends in the following four kinds of paradigm shifts in computer technology:

a) From Computer-centered to Human-centered
b) From Processing-centered to Information-centered
c) From Form-oriented to Content-oriented
d) From Centralized control to Distributed control

The first is based on deep reflection on the long history of computer-centered research which have never been good in human-computer interaction aspects, since such technology forces a human to approach machines/systems or allow, at best, addition of an ad-hoc interface on top of each system. We need to change paradigmatically to come up with an innovative and essentially better human interface technology. The concepts of Human-centered and Information-centered technologies are key concepts of such an enterprise. The true man-machine systems which are what we need in the coming information age do require an open architecture involving humans who need computers to help them facilitate their daily activities.

The second reflects what we learned from expert system development in which *processing-centered* approach has been dominant. That is, an expert system tries to solve a problem instead a human. The problem was it is not what people really need. What people need is an intelligent life-long partner who helps them in many aspects to amplify their capability.

The third is a topic related to artificial intelligence(AI) research where so-called form-oriented basic research has been extensively conducted. It has been trivial from the beginning that no intelligent system can function without a reasonable amount of knowledge. Nevertheless, form-oriented research has dominated AI research. Content-related activities are mainly knowledge base construction. Although huge amount of such activities have been conducted to date, they are “development” rather than “Research”, since they are ad hoc, heavily domain-specific and hardly accumulatable. We need content-oriented “Research” to make an essential contribution to intelligent system building.

The fourth is related to system architecture issues. It is an infrastructure of building a large-scale robust system which is often difficult to build and maintain. Typical distributed control systems include a multi-agent system in which agents collaborate with each other without a priori specification of interaction between them unlike the conventional centralized control systems. This paradigm of system design makes it easier to build a large-scale system provided a powerful negotiation protocol.

Web Intelligence, or Web-based intelligent systems should be something like a partner of a human by being compliant with the above paradigm shift.

5. Web Intelligence and ontological engineering

When we accept the paradigm shift, WWW technology is going to bring us a kind of revolution to a knowledge base building. Conventionally, a knowledge base has been something to design and build upon request. However, WWW and semantic web technologies facilitate automatic building of knowledge resources so that a huge knowledge base virtually exists out there, and hence the problem to solve has become not to build a knowledge base from scratch but to collect appropriate web pages out of already existing WWW knowledge resources, to reorganize and to merge them. Enabling technologies are XML, RDF(S) and DAML+Oil in Semantic Web[SW].

Semantic web has been devised to make web pages machine interpretable and hence to change the WWW from flood of irrelevant information to a huge useful knowledge source. The goal is good. The problem, however, is how we can make use of the web pages retrieved which still include irrelevant pages and need more elaboration for use of specific purposes. Although ontology will be extensively used in Semantic web activities, the major use is limited to exploiting super-sub relation between concepts for the purpose of the intelligent retrieval of relevant web pages. It is true that retrieval is one of the key technologies in Semantic web. We look further ahead and envision critical contribution of ontological engineering to the next generation knowledge processing. That is, systematic development and sophisticated processing of semantic tags which will definitely overflow all over the world and will need sophisticated ontologies that are something more than a hierarchical organization of concepts to process them appropriately.

6. Conclusions

We have discussed the ontological engineering, its successful applications and future directions to go mostly from the application-oriented viewpoint. Academic perspectives of ontological engineering include fundamental theories and common top-level ontology. Both perspectives suggest ontological engineering will play a critical role as content technology for the next generation knowledge processing.

Acknowledgement

The author is grateful to Dr. Yoshinobu Kitamura and Mr. Kouji Kozaki for their contributions.

Reference

[Feigenbaum, 85] Feigenbaum, E.A.: "The art of artificial intelligence – Themes and case studies of knowledge engineering" Proc. of 5th IJCAI, pp.1014-1029, 1977.

[Gruber] Gruber, T., *What-is-an-ontology?* http://www-ksl.stanford.edu/kst/what-is-an-ontology.html.

[Chandrasekara 2000] Chandrasekaran, B. and John R. Josephson, "Function in Device Representation," to appear in *Journal of Engineering with Computers*, Special Issue on Computer Aided Engineering. http://www.cis.ohio-state.edu/~chandra/Function-in-device-representation.pdf

[Kitamura 99a] Kitamura, Y., and Mizoguchi, R., Meta-functions in Artifacts, Papers of 13th International Workshop on Qualitative Reasoning (QR-99), 136-145, 1999

[Kitamura 99b] Kitamura, Y., and Mizoguchi, R., Towards Redesign based on Ontologies of Functional Concepts and Redesign Strategies, Proc. of the 2nd International Workshop on Strategic Knowledge and Concept Formation, pp.181-192, 1999

[Kitamura 00] Kitamura, Y., Sano, T., Mizoguchi, R., Functional Understanding based on an Ontology of Functional Concepts, The Sixth Pacific Rim International Conference on Artificial Intelligence (PRICAI 2000), pp.723-733, Springer-Verlag, 2000

[Kozaki 00] Kozaki, K. et al.: Development of an Environment for Building Ontologies which is based on a Fundamental Consideration of "Relationship" and "Role": Proc. of the Sixth Pacific Knowledge Acquisition Workshop (PKAW2000), pp.205-221 ,Sydney, Australia, December 11-13, 2000.

[Mizoguchi 95a] Mizoguchi R. et al., Task Ontology for Reuse of Problem Solving Knowledge Knowledge Building & Knowledge Sharing 1995(KB&KS'95) (2nd International Conference on Very Large-Scale Knowledge Bases), Enschede, The Netherlands, pp.46-59.

[Mizoguchi 95b] Mizoguchi, R., et al.: Ontology for Modeling the World from Problem Solving Perspectives Proc. of IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, pp. 1-12, 1995.

[Mizoguchi 97] Mizoguchi, R., and Ikeda, M. 1997, Towards ontology engineering. In *Proc. of PACES/SPICIS '97*, 259-266.

[Mizoguchi 99] Mizoguchi, R., et al.: Human media interface system for the next generation plant operation, Proc. of the IEEE SMC99, IEEE Systems, Man and Cybernetics Society, V-630-635, 1999.

[Mizoguchi 00a] Mizoguchi, R., and Kitamura, Y., Foundation of Knowledge Systematization: Role of Ontological Engineering, Industrial Knowledge Management - A Micro Level Approach, Rajkumar Roy Ed., Chapter 1, pp.17-36, Springer-Verlag, London, 2000

[Mizoguchi 00b] Mizoguchi, R. et al., Construction and Deployment of a Plant Ontology, The 12th International Conference on Knowledge Engineering and Knowledge Management - Methods, Models and Tools -, EKAW2000, pp.113-128, 2000.

[Pahl 88] Pahl, G., and Beitz, W., Engineering design - a systematic approach, The Design Council, 1988.

[Saasajima 95] Sasajima, M.; Kitamura, Y.; Ikeda, M.; and Mizoguchi, R. FBRL: A Function and Behavior Representation Language. Proc. of IJCAI-95, 1830-1836, 1995.

[SW] http://www.semanticweb.org/