Ontology-based systematization of functional knowledge

Riichiro Mizoguchi
The Institute of Scientific and Industrial Research
Osaka University, 8-1 Mihogaoka, Ibaraki, Osaka, Japan
miz@ei.sanken.osaka-u.ac.jp

Abstract

It has been recognized that design knowledge is scattered around technology and target domains. One of the two major reasons for it is that different frameworks (viewpoints) for conceptualization of design knowledge are used when people try to describe knowledge in different domains. The other is that several key functional concepts are left undefined or even unidentified. In this paper, I first overview the state of the art of ontological engineering which I believe is able to make a considerable contribution to resolving these difficulties. I then discuss our enterprise aiming at systematization of functional knowledge used for synthesis. We discuss ontologies that guide conceptualization of artifacts from the functional point of view. The framework for knowledge systematization is based on an extended device ontology and a functional concept ontology built on top of the device ontology. The utilization of the systematized functional knowledge in several application systems is also discussed together with its advantages.

Keywords
Ontology, Knowledge systematization, Functional knowledge, Knowledge sharing, Design

1. Introduction

Design is a creative activity that translates a requirement specification at the functional level into a set of attribute values of concrete things. Although advancement of computer and AI technologies has enabled easy access to information related to structure and/or shape of artifacts, design know-how used in the conceptual design phase is left implicit because of its subjectivitiy and implicitness. As discussed in the knowledge management, such subjective, and hence implicit knowledge is highly required to be made explicit to share within a community. The same applies to design community and it is expected that design knowledge sharing will improve the design process drastically. In order to make it happen, however, we need to resolve some big problems. One of them is to devise a framework for capturing and describing conceptual design knowledge that has never been shared within any community. Such a framework should be general enough for being shared by people in different communities and should enable consistent description of such knowledge in a computer interpretable form. However, it is a challenge to make it possible to describe subjective design knowledge in a general and sharable form. In fact, we see many examples that fail to organize knowledge in such a way.

The above issue is recognized as that of knowledge engineering because it is deeply related to how we can deal with knowledge. Needless to say, design process is inherently knowledge-rich, and hence a computer which tries to facilitate the process should be able to utilize knowledge skillfully in any sense. Nevertheless, the conventional knowledge technology is not good enough to realize it. However, ontological engineering that is the latest technology in knowledge engineering has been demonstrating its usefulness in overcoming some difficulties the conventional knowledge engineering has.

The main objective of this paper is to overview the current state of the art of Ontological Engineering (Guarino, 1997)(Mizoguchi & Ikeda, 1997)(Smith & Welty, 2001)(Sowa, 1995) with an example of its use for systematization of functional knowledge. Among several merits of ontological engineering, systematization of knowledge is one of the most promising because an ontology provides us with a system of fundamental concepts and relations among them in term of which we can describe all pieces of knowledge we are interested in. Explication of conceptual structure behind the design knowledge contributes to interoperation between knowledge and to sharing/reuse of knowledge by providing a firm basis. Among various kinds of design knowledge, we concentrate on functional knowledge. The next chapter overviews the knowledge engineering that has developed into ontological engineering and describes the scope of our enterprise. Chapter 3 describes the skeletal plan of our ontology building for knowledge systematization. Chapter 4 discusses an extended device ontology that plays a crucial role in our framework. On the basis of the device ontology proposed, functional ontologies are discussed in detail in chapter 5. In order to demonstrate the utility of the knowledge systematization, an overview of a deployment system is described in chapter 6. Chapter 7 discusses related work followed by concluding remarks.

2. Ontological Engineering and Knowledge Systematization

2.1 From knowledge engineering to ontological engineering

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 systematization, knowledge sharing, facilitation of agent communication, media integration through understanding, 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 how to systematize knowledge, what knowledge we should prepare for solving the problems given, how to scale up the knowledge bases, how to reuse/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 proposed by Feigenbaum(1977) 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". The authors are not claiming the so-called rule-base technology is what we need for future knowledge processing. Rather, treatment of knowledge should be in-depth analyzed further to make it sharable and reusable among computer and human agents. 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 a knowledge base. We could identify the reasons for this as follows:

1) It tends to be ad-hoc, and
2) 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. Ontological Engineering has been proposed for that purpose.

Ontological Engineering 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. Taking knowledge management as an example, I believe it essentially needs content-oriented research. It should be more than information retrieval with powerful retrieval functions. We should go deeper to obtain the true knowledge management.

An ontology, which is a system of fundamental concepts, that is, a system of background knowledge of any knowledge base, explicates the conceptualization of the target world and provides us with a solid foundation on which we can build sharable knowledge bases for wider usability than that of a conventional knowledge base. Knowledge engineering has thus developed into ontological engineering.

2.2 What is an ontology?

There are many interpretations about what an ontology is in spite of the fact that it is understood to serve as a kernel theory and building blocks of content-oriented research. In fact, hot discussions are often done in many meetings on ontology. This section presents some of the definitions of an ontology.

2.2.1 Some definitions

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

(2) From AI point of view, an ontology is defined as "explicit specification of conceptualization" (Gruber, 1993) which is widely accepted in AI community. "Conceptualization" here should be interpreted as "intensional" rather than "extensional" conceptualization contrary to that defined in (Genesereth & Nilsson, 1987).

(3) From knowledge-based systems point of view, it is defined as "a theory(system) of concepts/ vocabulary used as building blocks of information processing systems" by R. Mizoguchi. Knowledge bases are necessary when one builds a model of problem solving in the world. In such a context of problem solving, ontologies are divided into two types: Task ontology for problem solving process and domain ontology for domain where the task is performed.

(4) Another given by Gruber(Gruber, 1993)
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. A very simple case would be a type hierarchy, specifying classes and their subsumption relationships. Relational database schemata also serve as ontologies by specifying the relations that can exist in some shared database and the integrity constraints that must hold for them.

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

2.2.2 Typical questions

(1) How is an ontology different from a knowledge base?

Let me cite a phrase found in the email archive of ontology:

Date: Wed, 26 Feb 1997 12:49:09 -0800 (PST)
From: Adam Farquhar axf@HPP.Stanford.EDU
………

① Does it express the consensus knowledge of a community of people?
② Do people use it as a reference of precisely defined terms?
③ Does it express the consensus knowledge of a community of agents?
④ Is the language used expressive enough for people to say what they want to say?
⑤ Can it be reused for multiple problem solving episodes?
⑥ Is it stable?
⑦ Can it be used to solve a variety of different sorts of problems?
⑧ Can it be used as a starting point to construct multiple (sorts of) applications including: a new knowledge base, a database schema, an object-oriented program?

The stronger the 'yes' answer is to these questions, the more 'ontological' it is.

The above opinion is based on that there is no clear boundary between ontology and knowledge. It is a reasonable understanding when we think of Cyc(Lenat, 1995) whose upper part is definitely an ontology and the whole seems to be a knowledge base. The above opinion is somewhat misleading, though many of AI researchers accept it, since it does not try to capture an essential property of an ontology which is something related to concepts rather than vocabulary and is something related to what exists in the target world of interest. My answer to the question is that we need to introduce a concept of relativity when we understand an ontology. I mean, a clear differentiation of an ontology from a knowledge base should come from its role, that is, an ontology gives you a system of concepts which are used to build a knowledge base on top of it, and hence an ontology can be a specification of the KB builder's conceptualization of the target world and is a meta-thing of a conventional knowledge base.

(2) How an ontology is different from the class hierarchy in object-oriented paradigm?

They are similar. The developmental methodology of an ontology and that of an object hierarchy is also similar to each other in the upper stream. In the lower stream, however, the former concentrate on declarative aspects and the latter on performance-related aspects. Thus, the essential difference between the two lies in that the ontology research exploits declarative representation, while the OO paradigm is intrinsically procedural. In OO paradigm, the meaning of class, relations among classes, and methods are procedurally embedded and they are implicit. The ontology paradigm, on the other hand, descriptions are made declaratively in most cases to maintain formality and explicitness.

(3) What's new? How is it different from taxonomy of concepts?

An ontology contains a taxonomy as its component. So, it partially implies a taxonomy. In general, a new term is rarely totally new. Rather, it is usually coined by extending existing terms. The term "ontology" is not an exceptional case. It is a new term and concept including existing concepts such as "taxonomy", "common vocabulary", "upper model", etc. by adding formality, richer relations, explicit representation of stuff usually left implicit.

(4) Are upper model and domain ontology compatible with each other?

An ontology should be use-specific in engineering settings. People believe that a general and neutral ontology is of useless. A typical example is task ontology(Mizoguchi, 1995) where domain knowledge has to be organized so as to fit into the task model the task ontology specifies. In such cases, neutral domain ontology cannot apply to any problem without adjusting to the task structure. This discrepancy suggests us a potential difficulty which we might face with. People involved in upper ontology development advocate the importance of general and use-neutral ontology which is hard to be accepted by people who have to deal with real world problems. Ideally, however, many use-specific ontologies could find some of the essential concepts in use-independent ontology which should be inherited by many real applications. We need more effort to harmonize the both activities believing we will come up with several possible upper ontologies one of which is really agreeable to many people.

(5) What is the computational semantics of an

ontology? Is it just a set of labels?

This is one of the most crucial points of the roles an ontology plays. Contrary to that an ontology sometimes looks just a set of labels, it has deeper computational semantics. I have proposed the following three levels of ontologies.

Level 1: A structured collection of terms. The most fundamental task in ontology development is articulation of the world of interest, that is, elicitation of concepts and identifying so-called *is-a* hierarchy among them. These are indispensable to things to be an ontology. Typical examples of ontologies at this level include topic hierarchies found in internet search engines and tags used for metadata description. Little definitions of the concepts are made.

Level 2: In addition to that at the level 1 ontology, we can add formal definitions to prevent unexpected interpretation of the concepts and necessary relations and constraints also formally defined as a set of axioms. Relations are much richer than those at the level one. Definitions are declarative and formal to enable computers to interpret. The interpretability of an ontology at this level enables computers to answer questions about the models built based on the ontology. Many of the ontology building efforts aim at those at this level.

Level 3: The ontology at this level is executable in the sense that models built based on the ontology run using modules provided by some of the abstract codes associated with concepts in the ontology. Thus, it can answer questions about runtime performance of the models. Typical examples of this type are found in task ontologies(Mizoguchi, 1995)(Breuker, 1994) (Chandrasekara, 1998).

(6) What are the concrete merits an ontology can provide?

This question is also very important. The following is an enumeration 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. The fundamental role of an ontology contributes to it.

(b) Explication of what is left implicit.
In all of the human activities, we find presuppositions/assumptions which are left implicit. Typical examples include definitions of common and basic terms, relations and constraints among them, and viewpoints for interpreting the phenomena and target structure common to the tasks they are usually engaged in. Any knowledge base built is based on a conceptualization possessed by the builder and is usually implicit. An ontology is an explication of such implicit knowledge. An explicit representation of assumptions and conceptualization is more than a simple explication. Although it might be hard to be properly appreciated by people who have no experience in such representation, 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) Data structure
An ontology in a database is the conceptual schema. In this sense, an ontology provides us with a data structure appropriate for information description and exchange.

(d) 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 backbone of systematization of knowledge.

(e) Standardization
The success of the modern industries has been achieved thanks to standardization of various components. We cannot avoid standardization in the successful knowledge processing research and activities in the real world. We do need something comparable to “bolts” and “nuts” in our community.

(f) Design rationale
Typical examples to be explicated include intention of the designers of artifacts, that is, part of design rationale. An ontology contributes to explication of assumptions, implicit preconditions required by the problems to solve as well as the conceptualization of the target object which reflects those assumptions. In the case of diagnostic systems, fault classes diagnosed and range of the diagnostic inference, in the case of qualitative reasoning systems, classes of causal relations derived, etc.

(g) Meta-model function
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.

(h) Theory of content
In summary, an ontology provides us with “a theory of content” to enable research results to accumulate like form-oriented research avoiding ad-hoc methodologies which the conventional content-centered activities have been suffering from.

2.2.3 A brief survey of the research on ontology

This subsection is devoted to a brief survey of the current ontology research.

(A) Theory

N. Guarino and J. Sowa have been independently conducting research on theories of ontology. They both incorporate the results obtained in philosophy as principles to design the top-level ontology. Many of the practitioners have negative attitude to the top-level ontology under which they are required to put their ontology because they believe no use-independent ontology is useful. In the case of building an ontology for a large scale knowledge base, however, the validity of the knowledge base necessarily be justified in terms of wider rage of tasks, that is, it needs to show its generality rather than task-specific utility. Compliance with the principled top-level ontology provides a good justification. Thus, top-level ontology is important.

Sowa's ontology is based on J. S. Peirce's idea of the firstness which is defined without assuming any other things like human, iron, etc., the secondness which is defined depending on other things like wife, teacher, etc., and thirdness which provides an environment or context where the secondness works like family, school, etc. He introduces two important concepts, continuant and occurrent in addition to the three and obtains 12 top level categories by combining the seven primitive properties(Sowa, 1995).

Guarino's top level ontology is more extensively incorporates philosophical consideration. It is designed based on mereology(theory of parts), theory of identity, and theory of dependency. His ontology consists of two world: An ontology of Particulars such as things which exist in the world and Universals which include concepts we need when we describe Particulars(Guarino 1997).

(B) Standard Upper Ontology(SUO, 2001)
IEEE LTSC(Learning Technology Standard Committee has started to design the standard upper ontology. This is one of the most philosophical ontologies among those in AI community. Discussions on what an identity is, which is better, 3D(3D space with time) or 4D(including time as the 4th dimension) modeling, multiple ontologies or monolithic ontology, etc. The problem is that it is very hard to come to an agreemen.

(C) Machine readable dictionary: MRD
Development of machine-readable dictionaries has been done extensively in natural language processing community where ontology has also been discussed as the upper level model of the words/concepts structure. Typical examples include WordNet (WN, 1993), EDR(Yokoi, 1995), EuroWordNet(EWN, 2001). Cyc(Lenat, 1995) is not a dictionary, but is a huge common sense knowledge base whose upper level structure is an ontology.

(D) Semantic Web(SW, 2001)

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(XML 2001), RDF(S)(RDF 2001) and DAML+Oil(SW 2001) in Semantic Web.

One of the key technologies employed in Semantic web is DAML+Oil, a markup language for ontology representation. Although it is being developed energetically, there is no principle on how to design an ontology, since ontology design is intentionally detached from their project. In Semantic web, ontology design is understood as the users' job, that is, there will be no control over what ontologies might be designed.

There are two kinds of ontologies in computer science: Light-weight ontology and heavy-weight ontology. The former is what we see often in WWW and/or Semantic web such as a topic hierarchy of a search engine, a set of terms defined so as to enable a computer interpret them. The ontologies found in semantic web belong to this kind. They are mainly vocabulary rather than concepts. On the other hand, heavy-weight ontology is what we have discussed in this paper and ones discussed by Guarino and Sowa such as a hierarchy of concepts designed intended to relate them to real individuals existing in the world relying on the results obtained in Philosophy. The former is nothing to do with knowledge systematization because it has little principle in ontology design and does not pay attention to the semantics of links or to achievement in philosophy.

2.3 Two basic questions

(1) Why ontology instead of knowledge?
It is true that knowledge is domain-dependent, and hence knowledge engineering which directly investigates such knowledge has been suffering from a rather serious difficulty caused by its specificity and diversity. However, ontology is different. In the ontology research we investigate knowledge in terms of its origin and elements from which knowledge is constructed. Hierarchical structure of concepts and decomposability of knowledge are exploited to deeply investigate primitives of knowledge as well as background theories of knowledge which enables us to avoid the difficulties knowledge engineering has faced with.

(2) Isn't Ontological engineering yet another application-oriented research?
While ontological engineering deals with

Table 1 Current state of the art of design knowledge systematization.

	Knowledge about design objects	Knowledge about design process		
		Logical aspects	Mathematical aspects	System aspects
Domain dependent	Conventional domain-oriented engineering	NA	NA	Ad hoc system
Domain independent	**None1**	Tomiyama's project	General Design Theory	**None2**

domain-specific knowledge, it tries to establish theories and technology for "accumulating" knowledge within reasonable size of stratified domains utilizing ontologies. It is such a branch of AI research that investigates basic theories and technology to treat real world knowledge and is such an enterprise that denies the simple dichotomy of AI research: "Basic research" and "Application research".

2.4 Ontology and knowledge systematization

We have overviewed ontology research thus far. The next topic of this paper is how to use ontology. Among many possibilities, I believe its use for knowledge systematization is one of the most promising. This 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 concept with an explicit conceptualization of the assumptions. This nicely suggests ontological engineering is promising for the purpose of our enterprise.

While every 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 consumption. Our knowledge systematization adopts another way, that is, ontological engineering to enable people to build systematized knowledge bases for computer consumption. 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.

Table 1 represents my understanding of the current state of the art of design knowledge systematization. Basically, the conventional research on design knowledge is highly domain-dependent and has investigated such knowledge by analytical and quantitative methods. Knowledge base construction has been ad-hoc as is seen in the upper row of Table 1. Yoshikawa (1981) initiated the research on General Design Theory (GDT) to overcome the difficulties caused by the domain-dependence of the research activities. GDT is mainly concerned with the static nature of design in terms of mathematics, and hence it does not cover the dynamic aspects of design, that is, design process. Tomiyama and his colleagues (Tomiyama 2000) have been investigating the research on design process modeling using logic and artificial intelligence and have come up with a deeper understanding of design process, that is, "design process is an abduction". Table 1 shows there are two major issues left untouched. In our project, we have been mainly attacking the issue depicted as **None1**. Although it is mainly concerned with *knowledge* rather than *process*, our research shares the philosophy with Yoshikawa and Tomiyama in the sense that *investigating synthesis from a domain-independent view to reveal the inherent nature of synthesis*.

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 the next chapter.

3. Functional Ontology and Knowledge Systematization

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. Contrary to the possible strong disagreement from the people working in the mechanical design, the claim has a strong justification that the concept of a function originally came from the user requirements which is totally object- and behavior-independent, since common people have no knowledge about how to realize their requirements and are interested only in satisfaction of their requirement by a device built. Another justification is reusability of functional knowledge. If functions are defined depending on objects or their realization method, few functions are reused in and transferred to different domains. As is well understood, innovative design can be facilitated by flexible application of knowledge or ideas across domains.

3.1 Functional representation

Functional representation has been extensively investigated to date (Keuneke 1991; Chandrasekaran, Goel & Iwasaki 1993; Chittaro, Guida, Tasso & Toppano 1993; Lind 1994; Sasajima, Kitamura, Ikeda & Mizoguchi 1995; Umeda, Ishii, Yoshioka, Shimomura & Tomiyama 1996; Chandrasekaran & Josephson 2000) 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 cause 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. However, conventional research of function is not mature enough to propose such a framework and only a few functional concepts are identified to date (Pahl & Beitz 1988; Chittaro et al. 1993; Lind 1994). Although value engineering (Miles 1961; Tejima et al. 1981) has a long history of research on functional terms and has come up with a rich set of functional vocabulary, they are for human consumption and no operational definition is made.

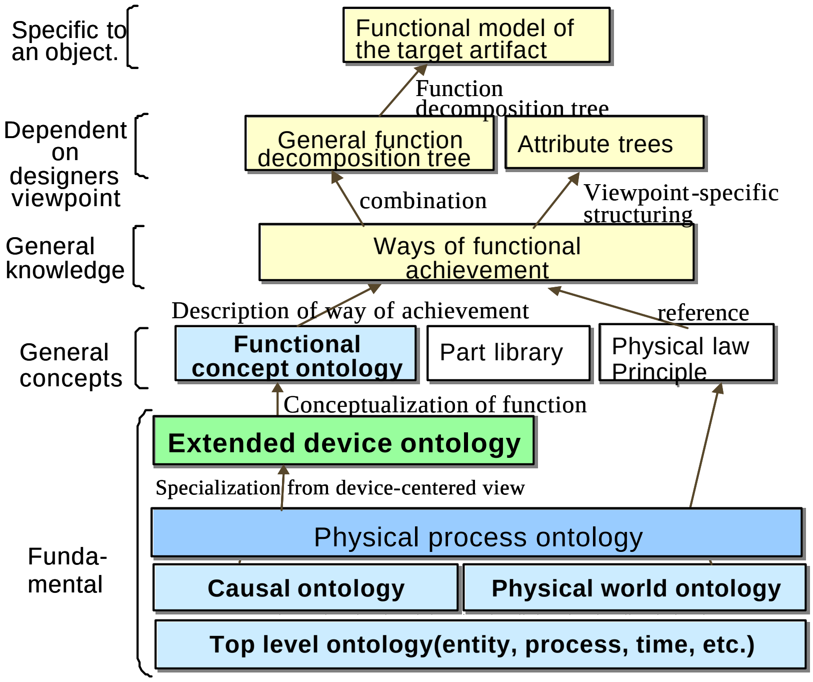

Specific to an object.
Dependent on designers viewpoint
General knowledge
General concepts
Funda-mental
Functional model of the target artifact
Function decomposition tree
General function decomposition tree
Attribute trees
combination
Viewpoint-specific structuring
Ways of functional achievement
Description of way of achievement
reference
Functional concept ontology
Part library
Physical law Principle
Conceptualization of function
Extended device ontology
Specialization from device-centered view
Physical process ontology
Causal ontology
Physical world ontology
Top level ontology(entity, process, time, etc.)

Figure. 1 Hierarchy of ontology and knowledge.

3.2 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 scratch. These two ontology contributes to "Symbol grounding" of higher-level concepts, that is, functional concepts. Although these two topics are very important, they are omitted in this paper due to the space limitation. On top of these three, physical process ontology is introduced to specify physical(natural) processes. A very primitive definition of *process* is done in the top-level ontology to cover any sequence of events in terms of state and its change. The term "Process" is confusing, since it has three different meanings; sequence/course/stage, something found in the phrase "Burning process" and processing. The first is one defined in the top-level ontology, the second is covered by the physical process ontology introduced here and the last is covered by device ontology discussed in chapter 4. These five ontologies(Top-level, causality, physical world, physical 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 a device, a 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.*

Way of functional achievement is knowledge about *how*(in what way) a function is achieved, whereas the functional concept is about *what* the function is going to achieve. Although functional achievement way knowledge looks similar to functional decomposition like that discussed in (Pahl & Beitz 1988), the former is much richer than the latter in that it consists of four kinds of hierarchies of different roles and principles. 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. The next chapter is devoted to the intensive discussion on device ontology that is the key topic of our knowledge systematization. All the concepts introduced and discussed in the rest of the paper are based on the extended device ontology.

4. Device Ontology

4.1 What is device ontology?

Concerning modelling of artifacts, there exist two major viewpoints: Device-centered and Process-centered views. The device ontology, e.g., one proposed by de Kleer and Brown (1984), specifies the device-centered view of artifacts. Device-centered view regards any artifact as composition of devices which process input to produce output which is what the users need. Process-centered view applied to an artifact, e.g., one proposed by Forbus (1984), concentrates on phenomena occurring in each artifact(device) to obtain the output result with paying little attention to the devices existing there. Device ontology imposes a frame or viewpoint on an event to introduce an engineering perspective. That is, it introduces the concepts of a black box equipped with input and output ports. Although physical process ontology, which specifies the process-centered view, 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

The major difference between the two is that while device ontology has an ***Agent***, which is considered as something that plays the main actor role in obtaining the output, process ontology does not have such an *Agent* but has ***participants***, which only participate in the phenomena being occurring. Needless to say, such an *Agent* coincides with a device in device ontology.

Device ontology specifies the roles played by the elements that collectively constitute a device. The concept of role is a hot topic in ontological engineering because an object plays different roles in different situations, and the fact has been a major source of failure in conceptualization of the world. For example, a man plays many roles such as husband, father, son in his family. These roles are defined in the family context, and hence they are specified by family ontology. Thus, device ontology can be said as a role specification system for the elements we recognize in a device in general.

Our claim is not that device ontology enables all kinds of description of all kinds of artifacts but that we should appreciate the potentials of device ontology and we do our best to extend it if possible to extend its applicability without losing its advantages.

4.2 Extended device ontology

This section presents the key concepts(roles played by objects) in the extended device ontology. The discrimination between behavior and function is discussed in chapter 5. We exclude static behavior such as *to support* by concentrating only on dynamic behaviors.

4.2.1 Device and object

Things that exist in the device ontology world are grouped into two categories: *Device* and *Object.* A *device* has input and output ports through which it is connected to another device(precisely speaking, not a device but conduit which is explained later in detail). A *device* consists of other devices of smaller grain size and usually is organized in a whole-part hierarchy of sub-devices. An *object* is something that can be considered as that it goes through a device from the input port to the output port during which it is processed by the device. Examples of an *object* include substance like fluid, energy like heat, motion, force, information, etc. An *object* has attributes whose values change over time. A *device* is something that operates on an *object* that goes through the device. The state change of an *object* is realized by the difference between the states of the *object* at the input port and that at the output port.

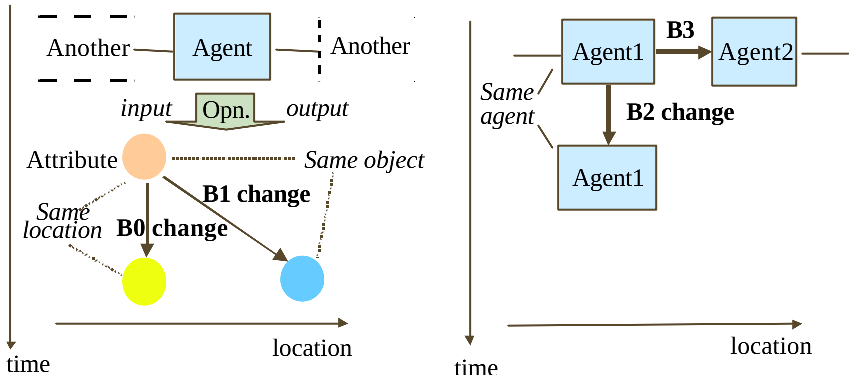


Figure 2 Four different definitions of behavior.

4.2.2 Conduit and medium

A *conduit* is defined as a special type of *device* that can be considered as it transmits an *object* to output port without any change in an ideal situation. Examples include a pipe, a shaft, etc. We exclude conduit from device.

A *medium* is something that holds an *object* and enables it to flow among devices. For example, steam can play the role of a medium because it can hold heat energy. In some domain, conduit can play the role of medium. For example, while a shaft is a conduit for force and motion, at the same time, it plays the role of medium for them.

4.2.3 Behavior

We identified four kinds of definitions of *Behavior*. Figure 2 illustrates simplified situations for behavior definition. ***B0 behavior*** is defined as the change of an attribute value of an *object* at the same location over time. Typical example is increase of the temperature of fluid at some observation point over time. Note that what is observed is a different thing at any time. This is exactly same as the observation of a real phenomenon and coincides with what numerical simulation obtains.

B1 behavior is defined as the change of an attribute value of an *object* from that at the input port of a device to that at the output of the device. For example, the increase of the temperature of steam occurred during it goes through a super-heater is B1 behavior. The key difference between B0 and B1 is that while B0 behavior concentrates on the location of the observation rather than identity of the observed *object*, B1 behavior on the identity of the observed *object* rather than the location.

B2 behavior is defined as the change of something inside of a device rather than input/output ports. The “something” could be *motion* of a part of the device or inner state of the device. For example, “rotation of fins in a fan” is an inner behavior of “to fan”, “a shaft is twisted” is an inner behavior of “to transmit torque”, etc. This behavior is based on an answer to the question such as “what motion is the device performing?” and is not the behavior of the device of smaller grain size but that identified by peeping into the device with a violation of the “black box principle” of device ontology.

B3 behavior is defined as any behavior to another *device*. The important aspect here is *B0* and *B1 behaviors* are concerned with *objects* rather than *devices*.

All the definitions above share that behavior is a conceptualization of the change of attribute values in the spatio-temporal space over time. The differences come from the way of treatment of the location in the spatio-temporal space and the target of the operation to be interpreted as behavior. Another definition of behavior, which looks very similar to B1 behavior at first glance, is found in (Chandrasekaran et al. 1993) where function is defined as *B1 behavior* and the behavior corresponding to the function is defined as series of *B1 behavior* of sub-devices of smaller-grain size. That is, in his definition, the difference between function and behavior is relative to the grain size, which is different from our principle that function is a teleological interpretation of behavior.

4.3 Modelling of a mechanical systems based on the extended ontology

In the extended device ontology, we view motion and force in mechanical systems as an *object*. That is, a mechanism as a device is considered as a thing that changes attributes(direction, amount, etc.) of motion and force.

There are two levels of grain sizes in mechanical systems: mechanism level and mechanical element level. By a mechanical element, we mean a gear, a shaft and so on and by a mechanism, we mean a complex of elements like a gear pair. Identification of a mechanism is done by identifying a conduit and by considering it as the boundary between mechanisms. A conduit at the mechanism level is a shaft or a wire, since they just transmit force or motion without change from one end to the other end. Let us take an example shown in Figure 3a. The gear pair is modelled as a device that accepts torque and angle velocity as input through a shaft as a conduit and outputs them with different values determined according to the ratio of gear numbers. The output is put on the shaft and transmitted to the belt mechanism that changes the rotation motion to horizontal motion. These two mechanisms constitute a larger mechanism.

Treatment of elements is as follows: A conduit at the element level is virtual and is defined as conceptualization of the mechanical pair(of elements), that is, the contact point/line/face that locates at the boundary between the connecting elements. A wheel is modelled as a device that has a line-contact conduit that can transmit only tangent velocity and force at the surface and has fixed(embedded) connection to a shaft around its central part that transmits everything. A gear in a gear pair is modelled in the similar way as a wheel. Let us take an example shown in Figure 3b. A gear is modelled as a device that accepts number of rotations and torque and outputs tooth number velocity(number of teeth per time) and tangent force that is obtained by dividing the torque by radius of the gear. This model neglects the boundary between teeth, though it causes no problem in our goal. Note here that input torque is not transmitted by a gear or a wheel as a mechanical element, since torque can be changed according to the ratio of the number of teeth of the connecting gears or to the ratio of the radiuses of the wheels.

The inherent property of a conduit in the device ontology is that it transmits all attributes that medium holds. It is true for pipe in the plant domain and a shaft in the mechanical domain. But, a wire transmits only pulling force and the virtual conduit introduced in the element level of mechanical domain is not the case. It transmits only limited attributes depending on the pair of the elements. This is the key extension of our device ontology.

4.4 Applicability of the extended device ontology to different domains

The contributions of the extended device ontology are two fold: it provides us with (1) a machine understandable framework for modelling artifacts in different domains with a consistent viewpoint and (2) appropriate vocabulary in terms of which we can describe differences between models in different domains. Table 2 shows comparison among models in plant domain and mechanical system domain. The differences are summarised as follows: In the mechanical system domain,

(1) A *conduit* is degenerated to *medium*. In other words, the *conduit role* and *medium role* are played by a single thing at the same time.
(2) *Medium* does not flow through a *device*, but it allows *objects* to flow by transmitting it through their connection.
(3) Force and motion are *objects* processed by a *device*.
(4) *Conduit* at the mechanical element level is virtual.
(5) What is transmitted by a *conduit* is limited depending on the types of the kinematic pair.
(6) *Conduit* is not unique but of variety.

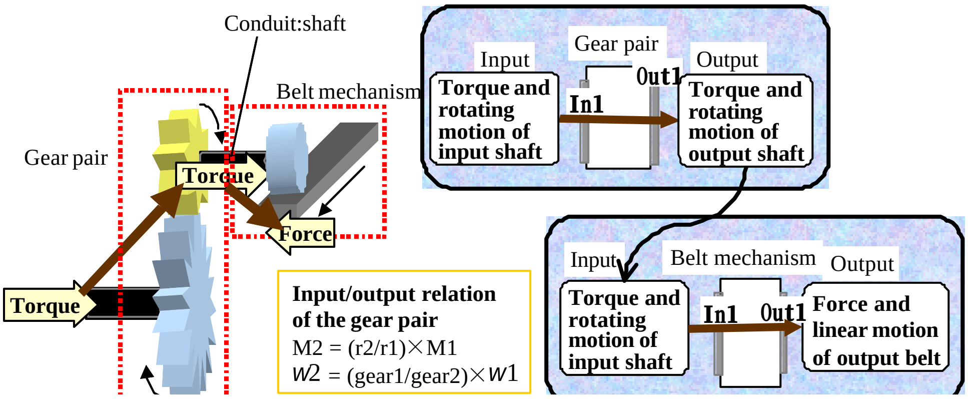

Conduit:shaft
Belt mechanism
Gear pair
Torque
Force
Torque
Input/output relation of the gear pair
M2 = (r2/r1)×M1
W2 = (gear1/gear2)×W1
Gear pair
Input
Output
Torque and rotating motion of input shaft
In1
Out1
Torque and rotating motion of output shaft
Input
Belt mechanism
Output
Torque and rotating motion of input shaft
In1
Out1
Force and linear motion of output belt

Figure.3a A model of a pair of gears at the mechanism level

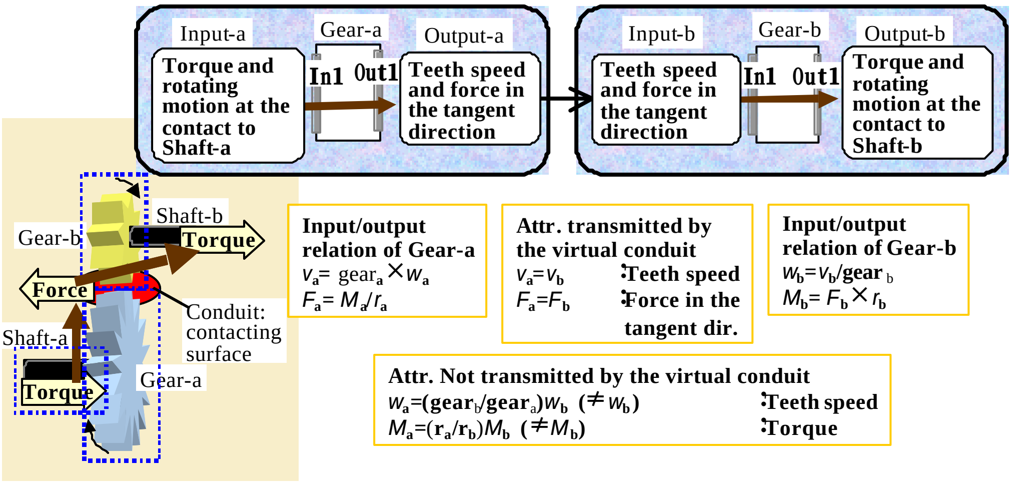

Input-a
Gear-a
Output-a
Torque and rotating motion at the contact to Shaft-a
In1
Out1
Teeth speed and force in the tangent direction
Input-b
Gear-b
Output-b
Teeth speed and force in the tangent direction
In1
Out1
Torque and rotating motion at the contact to Shaft-b
Gear-b
Shaft-b
Torque
Force
Shaft-a
Torque
Conduit: contacting surface
Gear-a
Input/output relation of Gear-a
v_a= $gear_a \times w_a$
$F_a = M_a/r_a$
Attr. transmitted by the virtual conduit
$v_a=v_b$:Teeth speed
$F_a=F_b$:Force in the tangent dir.
Input/output relation of Gear-b
$w_b=v_b/gear_b$
$M_b= F_b \times r_b$
Attr. Not transmitted by the virtual conduit
$w_a=(gear_b/gear_a)w_b$ $(\neq w_b)$:Teeth speed
$M_a=(r_a/r_b)M_b$ $(\neq M_b)$:Torque

Figure 3b A model of a pair of gears at the mechanical element level.

Table 2 Comparison among key concepts in plant and mechanical system domains.

	Plant: Energy	Plant: Entity	Mechanical: Mechanism level	Mechanical Mech. element level
Device	Boiler, turbine, etc.	Boiler, Distiller, etc.	Mechanism (Gear pair, Cam&shaft, etc.)	Mech. Element (gear, shaft, etc.)
Conduit	Pipe	Pipe, Belt conveyer	Shaft and wire	Contact (Surface, line, point)
Object	Heat energy	Fluid, stuff, etc.	Force and motion	Force and motion
Medium	Fluid (water, steam,etc.)	Fluid, tool, or nothing	Shaft and wire	Contact
Function	generate, give, rob, cool, etc.	divide, distil, separate, process, etc.	change No. of rotation, change kind of motion, etc.	change speed, transmit force and motion, etc.

In spite of these differences, the extended ontology captures essential properties of models in the two domains with explication of their differences. These results shown in Table 2 are based on our research experiences on Plant, production process (Mizoguchi & Kitamura 2000) and mechanical system domains for years.

5. Functional Ontology

We now have obtained a framework for building a functional model of an artifact. The next things we need are well-defined fundamental concepts in terms of which we can describe functional knowledge. In this chapter, we introduce several categories such as *base function, meta-function, way of functional achievement* and *method of functional achievement* together with functional concept ontology.

5.1 What is a function

We define a ***Function*** of a device as "teleological interpretation of ***B1 behavior*** under a given goal" (Sasajima et al. 1995). This tells us a function of a device is determined context-dependently, though *B1 behavior* is constant independently of the context. Considering that, in most cases of design, the context of a device is determined by the goal to be achieved by the device, function of a device is determined goal- or purpose-dependently. This reflects the reality that definition of function tends to be context-dependent and hence, in many cases, functional knowledge about design is hardly reusable. The major goals of our research include to give a framework for organizing functional concepts in a reusable manner and to define them operationally as an essential step towards knowledge systematization. By operationally, we mean a computer can make use of functional knowledge in the reasoning tasks of the functional modelling, understanding of the functional structure of a device and revising it. What we have to do for these goals are as follows:

1. To define functional concepts independently of its realization so as to make them reusable.
2. To devise a functional modelling method to enable a modeller to relate such reusable functional definitions to specific application problems, that is, to get functional concepts **grounded** onto the behavior and hence structure.
3. To formulate a functional decomposition scheme to obtain efficient functional knowledge for design.
4. To identify categories of functional concepts for systematization of functional knowledge.
5. To provide rich vocabulary for reasoning in the functional space.

The following is an overview of our work on ontologies of function.

5.2 Structure-behavior-function hierarchy

Figure 4 shows Structure-Behavior-Function hierarchy. It looks similar to Figure 1 but is different from Figure 1 which is an abstraction hierarchy of concepts related to function of artifacts. By structure, we mean topological relations among components(devices). ***Structure*** of a device constitutes a hierarchical structure according to the grain size that is shown in the lowest plain in Figure 4. By ***behavior***, we mean *B1 behavior*. What is obtained by teleological interpretation of *B1 behavior* under a given goal is called ***(Base) function***. The term "base" is used to discriminate it from meta-function introduced later.

A function is achieved by performing(achieving) a series of sub-functions which is called a ***method of functional achievement***. On the other hand, a conceptualization of the principle or intended phenomena or structure that gives justification of why and how the method achieves the function is called ***way of functional achievement*** that is considered as reference to the essential property of structure and behavior that achieve the function. The functional decomposition based on the way of functional achievement is compliant with the observations found in the research on design processes (Takeda et al. 1994) in which they say functional decomposition is not done solely in the functional space but done by going back and forth between the functional, behavioral and structural spaces during which some portions of the artifact are determined in each of the spaces simultaneously.

Note that whole-part hierarchies in the different layers do not always correspond to each other. Although the typical functional structure is one analog to the structural hierarchy, it could have many other different hierarchies according to the viewpoint to organize functional components.

Meta-function is a conceptualization of type of a base function and inter-dependency between them. While a *base function* is concerned with the change of *objects* in the domain, meta-function is concerned with *base functions*. ***Meta-function*** as inter-dependency between base functions is defined as teleological interpretation of causal relation between *base functions*.

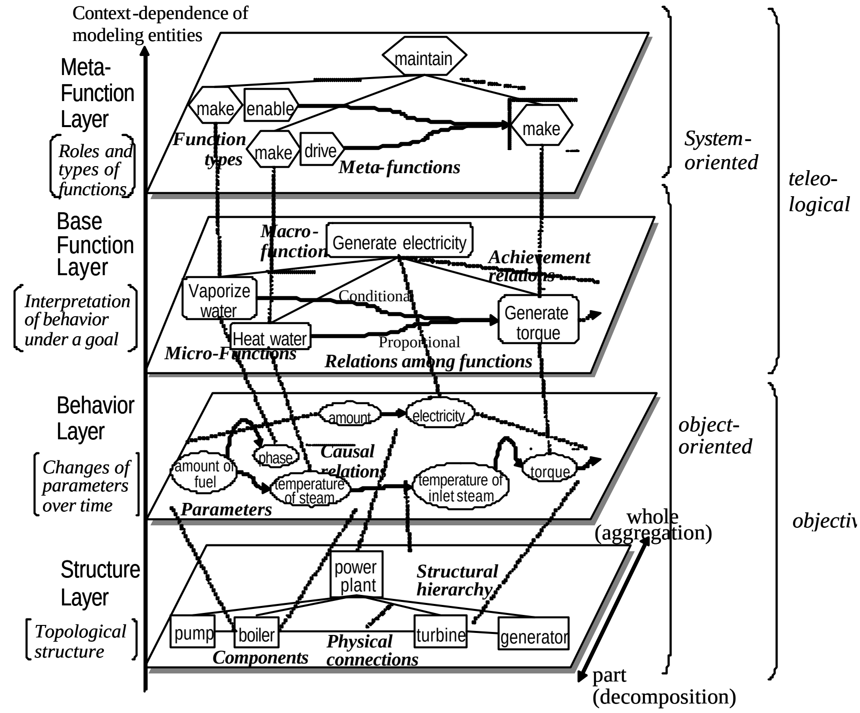


Figure 4. Hierarchy of target objects (Power plant).

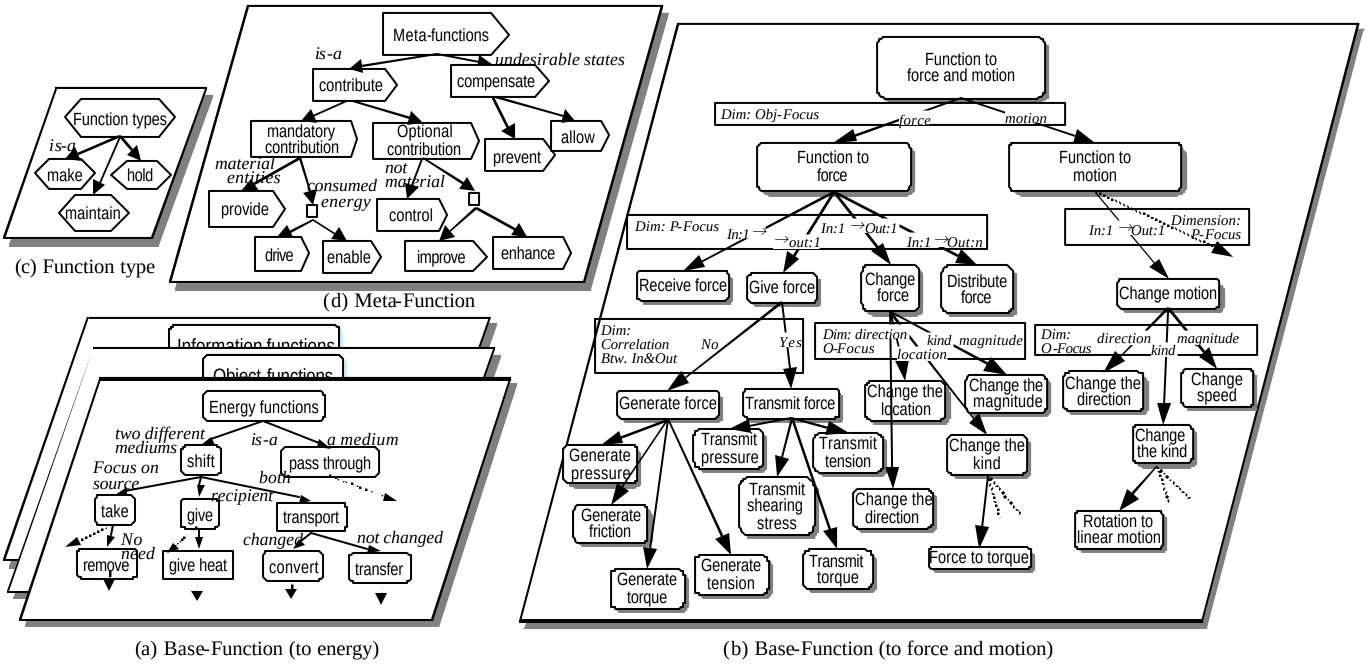
5.3 Function behavior representation language: FBRL

FBRL: Function Behavior Representation Language (Sasajima et al. 1995) is designed intended to ground functional concepts onto behavior and structure of a device. It is a language for representing a base function based on our extended device ontology. It consists roughly of input and output ports for device connection, behavioral definition in terms of attributes of *objects* and functional toppings(FT) that enable a system to map a structural and behavioral model to a functional concept. FT is composed of four items:

(1) *Obj-Focus* specifies *objects* to focus on
(2) *O-focus* specifies attributes of the *object* to focus on
(3) *P-Focus* specifies port to focus on
(4) *Necessity* specifies the necessity of *objects*

5.4 Functional concept ontology

Functional concept ontology defines three kinds of functional concepts introduced in 5.2 (Kitamura & Mizoguchi 1999a, 2000). Figure 5 shows a portion of *is-a* hierarchy of those concepts. Base function consists of four kinds of functions such as function for substance, that for energy, that for information and



(c) Function type

(d) Meta-Function

(a) Base-Function (to energy)

(b) Base-Function (to force and motion)

Figure 5. Functional concept ontology (portion)

that for force&motion. Figure 5 shows *is-a* hierarchies of functional concepts for energy(5a) and force&motion(5b). Each concept in the hierarchies are defined using FBRL and hence operational. For example, an energy function, *To take*, is defined as a behavioral constraint: *to shift* of energy and P-Focus on the port of energy provider. *To remove* is defined as that of *to take* with an additional FT, the energy taken is *unnecessary* as *Necessity* FT. These definitions demonstrate high independence of their implementation, while function is clearly related to structure and behavior. As is also shown in Figure 5b, functional concepts that mainly appear in mechanical system domain are defined in the same way as others. Note here that because the definitions are based on the extended device ontology, all the functional concepts(*verbs*) have *objects(force and motion)* rather than *devices*(mechanisms and mechanical elements) as their grammatical object.

The functional concept ontology is defined using Hozo, an environment for ontology development and use (Kozaki, Kitamura, Ikeda & Mizoguchi 2000).

The function types represent the types of goal achieved by the function (Keuneke 1991). Keuneke proposes some function types including "*To Prevent*" which represents to "keep a system out of an undesirable state of objects". However, because it focuses on changes of objects associated with the component, the objective of the function is implicit, that is, another function would be affected by the state. Therefore, we redefined the function type as "*To Make*", "*To Maintain*", and "*To Hold*" and redefined "*To Prevent*" as a kind of a meta-function as below.

Meta-function (denoted by *mf*) represents a role of a *base function* called an *agent function* (f_a) for another *base function* called a *target function* (f_t) (Kitamura & Mizoguchi 1999a, Kitamura, Sano & Mizoguchi 2000). We have defined the eight types of *meta-functions* as shown in Figure 5d (an *is-a* hierarchy). We begin definition of *meta-functions* with the condition where there is a causal relation from the focused parameter of f_a to that of f_t. If the goal of f_t is not satisfied when f_a is not achieved, the f_a is said to have a *mandatory contribution* for the f_t. Although we can intuitively say that f_a has a *To Enable meta-function* for f_t in such a case, the authors define a narrower meaning of *To Enable* by excluding the cases of *To Provide* and *To Drive* as follows.

Firstly, when a function f_a generates such an object (or energy) that will be a part of the focused entity of f_t (called *material*), the function is said to perform a *meta-function* "to provide material" for f_t. When a function f_a generates or transfers such an energy that intentionally consumed by f_t (called *driving energy*), the function is said to have the meta-function "to drive f_t". Lastly, *To Enable* meta-function is used for changing a necessary condition for f_t excepting the cases of *To Provide* and *To Drive*. What we mean by this weak definition is that the conditions such as the existence of the material and that of the driving energy are too obvious to be said to enable a function.

Furthermore, a function f_a having positive effects on the undesirable side effect of a function f_{t1} is said to have a meta-function "*To allow* the side-effects of f_{t1}". On the other hand, if a serious trouble (e.g., faults) is caused in a function f_{t2} when a function f_a is not achieved, the function f_a is said to have a meta-function "*To prevent* malfunction of f_{t2}". The details of definitions and examples are shown in (Kitamura and Mizoguchi 1999a).

Generally speaking, one cannot claim the theoretical completeness of something like functional ontology. What one can do is to provide convincing discussion with empirical studies. We already know there is another type of meta-function such as *To synchronize* in timing-related domain.

6. Roles and Effects of Functional Ontology

Functional concept ontology provides us with necessary and sufficient operational terms used for representing functional knowledge/model together with constraints satisfied by them. Its utility has been summarized in:

??**Functional model description**: (Sasajima et al. 1995, Kitamura et al. 2000).
??**Functional knowledge description**: (in 6.2)
??**Explanation generation**: (Sasajima et al. 1995).
??**Specification of the inference space**: (Kitamura & Mizoguchi 1999b).
??**Automatic identification of functional structure**: (Kitamura et al. 2000)

6.1 Deployment of the research result

The ontology and the systematization framework of functional knowledge are currently being deployed at production systems division of Sumitomo Electric Industries Inc., Japan, for sharing functional knowledge of devices used in the daily activities among engineers in the division. The preliminary result is very successful.

Engineers in the production systems division have been suffering from the difficulty in sharing and reusing knowledge among engineers in charge of different devices for long years. They have been regularly writing a technical report for design review, maintenance history, etc. and have accumulated a lot. Unfortunately, however, it has been difficult for them to understand a report written by other engineers who are in charge of different devices, and hence few of the reports are used. The reasons include:
(1) Descriptions are specific to the target object

(2) Knowledge is task-specific
(3) Vocabulary is not consistent or common
(4) Much knowledge is left implicit
(5) To retrieve appropriate report(knowledge) is hard

These are caused by deeper causes:
(1) There is no principle for representing and organizing functional knowledge
(2) Every engineer has his/her own viewpoint and there is no common way of how to view a device
(3) There no common vocabulary representing function
(4) There is no guideline for representing functional knowledge with little domain-dependence.

Our functional ontology and the framework for knowledge systematization is a solution to overcoming all the difficulties. In fact, engineers in the production systems division liked our framework very much and are happy to use it to represent their knowledge about devices they take care of. The system we build in this deployment is a server of function achievement way knowledge. Even in the current situation where no system is implemented, engineers who have some experiences for knowledge representation by our framework are sure that the system will help them a lot in managing and serving appropriate knowledge about the way of function achievement.

The success factors include:
(1) Extended device ontology enables users to be consistent in interpreting how a device works.
(2) Clear distinction between functional concept (goal) and way (its realization method) makes the knowledge highly domain-independent
(3) Functional concept ontology provides a rich set of well-defined functional terms
(4) Clear distinction between a general-specific hierarchy(*is-a* tree) and a whole-part hierarchy (*is-achieved-by* tree) enables to have consistent descriptions of functional decomposition trees and *is-a* hierarchies of ways of function achievement. This avoids the confusion between the two which has occurred very often.

7. Related work

Due to the space limitation I only present related papers with little discussion on them.

7.1 Definition of function:

There are quite amount of research on functional representation: de Kleer 1984, Chandrasekaran et al. 1993; Lind 1994; Umeda et al. 1996,Sembugamoorthy &Chandrasekaran 1986, Chandrasekaran & Josephson 1996. Our definition of function is "teleological interpretation of (B1-)behavior" and is operationalized by introducing FT: functional topping. Unlike all the other definitions, our idea of function can explain its goal (context)-dependency and that multiple functions are derived from a behavior according to the context.

7.2 Definition of functional concepts:

The functional vocabulary is discussed in Sembugamoorthy & Chandrasekaran 1986; Umeda et al. 1996, Tejima et al. 1981, Pahl & Beitz 1988; Chittaro et al. 1993; Lind 1994. The features of our definitions of functional concepts can be summarized as (1) use of mapping primitives with behavior and (2) independence of implementation. By the former, we mean that the functional concepts are defined by additional information to behavioral models using functional toppings (FTs) as primitives and then the information specifies the mapping from behavior to function. It enables us to ground the functional concepts on behavior and then to realize the behavior-function mapping.

7.3 Ontology of artifacts:

The importance of explicit specification of concepturalization (i.e., ontology) in artifact modeling is widely recognized in literature such as (Abu-Hannna & Jansweijer 1994; Borst, Akkermans & Top 1997; Horvath, Kucgozi & Vergeest 1998; Salustri 1998; Chandrasekaran & Josephson 2000).

Bond graph is a theory for describing a system domain-independently in the field of system dynamics (Rosenberg & Karnopp 1983). Although it is elaborated well, it still lacks the expressive vocabulary and device ontology to specify viewpoint in modelling an artifact.

Horvath et al. (1998) discuss design concept ontologies for comprehensive methodology for handling design concepts in conceptual design, which include structure and shape as well as functionality. For example, in the structural view, the connected entities are specified by positional, morphological, kinematical, and functional descriptors. We concentrate on functionality and then categorize connections among devices according to their functions, that is, kinds of transmitting force or motion in the mechanical domain.

7.4 Ways of functional achievement:

In Bradshaw & Young 1991; Bhatta & Goel 1997; Umeda et al. 1997, similar ideas to our idea of way of functional achievement is discussed. The major differences between the two include explicit description of "way", organization in *is-a* hierarchies based on principles of ways and a functional concept ontology.

8. Concluding remarks

We have discussed ontological engineering and its application to systematization of functional knowledge. We have shown that the extended device ontology contributes to consistent model building of artifacts and knowledge base building sharable across domains and to explication of the differences of functional knowledge in different domains which are seemingly incompatible with each other. The layers of ontologies thus help us systematize functional knowledge. Another contribution of this research can be summarized as a framework of systematization of design knowledge about functional decomposition. The idea of "way" of functional achievement plays a key concept for systematization. The benefits of such functional knowledge in a concurrent design team were shown using a scenario. All the systems described in this paper have been implemented.

I have been conducting this ontology-based knowledge systematization project for five years. It had looked a bit challenging because we got started from the very fundamental issues which seem to be very far from real world problems. But, I have made it. We are currently in the deployment phase of the achievement. My industrial partner, Production Systems Division of Sumitomo Electric Industries, has started to use our ontology-based framework for sharing engineering knowledge and some other types of applications in May, 2001.

I have advocated the importance of "Content-oriented" research rather than form-oriented research that have dominated AI research to date. The research described in this paper is a result in this direction. Knowledge processing never loses its importance and requires more sophisticated treatment of knowledge rather than inference mechanisms. To cope with the high demand on advanced knowledge processing, we need in-depth understanding about the nature of knowledge and viewpoints to model the target world, framework of knowledge description supported by solid foundation of conceptualization, and so on. Otology engineering for systematizing knowledge will become more important in the coming years.

Acknowledgements

The author would like to thank Yoshinobu Kitamura, Toshinobu Kasai, Kouki Higashide, Toshio Ueda, Toshinobu Sano, Masaru Takahashi, Mariko Yoshikawa and Tomonobu Takahashi for their contributions to this work. The author is grateful to Mitsuru Ikeda for his valuable comments. This research is supported in part by the Japan Society for the Promotion of Science (JSPS-RFTF97P00701). Special thanks go to the members, Professors. Eiji Arai,. Masahiko Onosato and Hiroshi Kawakami.

Reference

ABU-HANNA, A. & JANSWIJER, W. (1994), Modeling Domain Knowledge Using Explicit Conceptualization, *IEEE Expert*, 9(5), 53-63.

BHATTA, S. R. & GOEL, A. K. (1997). A Functional Theory of Design Patterns, In *Proc. of IJCAI-97*, 294-300.

BORST, P., AKKERMANS, H., & TOP, J. (1997). Engineering Ontologies, *Int'l Journal of Human-Computer Studies*, 46(2/3), 365-406.

BRADSHAW, J. A & YOUNG, R. M. (1991). Evaluating Design using Knowledge of Purpose and Knowledge of Structure. *IEEE Expert*, 6(2), 33-40.

CHANDRASEKARAN, B., GOEL, A. K., & IWASAKI. (1993). Functional Representation as Design Rationale, *COMPUTER*, 48-56.

CHANDRASEKARAN, B. & JOSEPHSON J. R. (1996). Representing Function as Effect: Assigning Functions to Objects in Context and Out. In *Proc. of AAAI-96 Workshop on Modeling and Reasoning with Function*.

CHANDRASEKARAN, B. & JOSEPHSON J. R. (2000). Function in Device Representation, *Engineering with Computers*, 16(3/4), 162-177.

CHITTARO, L., GUIDA, G., TASSO, C. & TOPPANO, E. (1993). Functional and Teleological Knowledge in the Multi-modeling Approach for Reasoning about Physical Systems: A case Study in Diagnosis, *IEEE Transactions on Systems, Man, and Cybernetics*, 23(6), 1718-1751.

LENAT, D. (1995) http://www.cyc.com/public.html

DE KLEER, J. & BROWN, J. S. (1984). A Qualitative Physics Based on Confluences, *Artificial Intelligence*, 24, 7-83.

DE KLEER, J. (1984) How Circuits Work., *Artificial Intelligence*, 24, 205-280.

FEIGENBAUM, E.A. (1977) "The art of artificial intelligence – Themes and case studies of knowledge engineering" Proc. of 5th IJCAI, pp.1014-1029, 1977.

FORBUS, K. D. (1984). Qualitative Process Theory, *Artificial Intelligence*, 24, 85-168, 1984.

GANGEMI, A., GUARINO, N., MASOLO, C., and OLTRAMARI, A. (2001) Understanding top-level ontological distinctions. *Proc. of IJCAI 2001 workshop on Ontologies and Information Sharing*

GRUBER, T. (1993) http://www-ksl.stanford.edu/kst/what-is-an-ontology.html.

GENESERETH, M. and NILSSON, N. (1987) Foundation of Artificial Intelligence, 1987.

GUARINO, N (1997) Some organizing principles for a unified top-level ontology, Working Notes of AAAI Spring Symposium on Ontological Engineering, Stanford.

HODGES, J. (1992). Naive mechanics - a computational model of device use and function in design improvisation, *IEEE Expert*, 7(1):14-27.

HORVÁH, I., KUCZOGI, GY. & VERGEEST, J. S. M. (1998). Development and Application of Design Concept Ontologies for Contextual Conceptualization, in *Proc. of 1998 ASME Design*

Engineering Technical Conferences DETC, CD-ROM: DETC98/CIE-5701, ASME, New York.

KEUNEKE, A. M. (1991). A. Device Representation: the Significance of Functional Knowledge, *IEEE Expert*, 24, 22-25.

KITAMURA, Y., & MIZOGUCHI, R. (1999a). Meta-Functions of Artifacts, *Proc. of The Thirteenth International Workshop on Qualitative Reasoning (QR-99)*, 136-145.

KITAMURA, Y., & MIZOGUCHI, R. (1999b). 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.

KITAMURA, Y., SANO, T. & MIZOGUCHI, R. (2000). Functional Understanding based on an Ontology of Functional Concepts, *Proc. of The Sixth Pacific Rim International Conference on Artificial Intelligence (PRICAI 2000)*, pp.723-733, Springer-Velag.

KOZAKI, K.,, KITAMURA, Y., IKEDA, M., & MIZOGUCHI R. (2000). 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*), 205-221.

LIND, M. (1994). Modeling Goals and Functions of Complex Industrial Plants. *Applied artificial intelligence*, 8, 259-283.

MALMQVIST, J. (1997). Improved function-means trees by inclusion of design history information, *Journal of Engineering Design*, Vol.8, No.2, pp.107-117.

MILES, L. D (1961). *Techniques of value analysis and engineering*. McGraw-hill.

MIZOGUCHI, R., & IKEDA, M. (1997) Towards Ontology Engineering, *Proc. of The Joint 1997 Pacific Asian Conference on Expert systems / Singapore International Conference on Intelligent Systems*, pp. 259-266, also Technical Report AI-TR-96-1, I.S.I.R., Osaka University, http://www.ei.sanken.osaka-u.ac.jp/pub/miz/miz-onteng.pdf

MIZOGUCHI, R. et al. (1995) Task ontology for reuse of problem sloving knowledge, Proc. KB&KS95, Enshede, The Netherland.

MIZOGUCHI, R (1997) Ontology Engineering, Towarad establishment of theories and fundamental technologies for content-orietned research, J. of JSAI, Vol.12, No.4, pp.559-569, 1997(in Japanese)

MIZOGUCHI, R. & KITAMURA, Y. (2000). Foundation of Knowledge Systematization: Role of Ontological Engineering, *Industrial Knowledge Management - A Micro Level Approach*, Rajkumar Roy Ed., Chapter 1, 17-36, Springer-Verlag, London.

MORTENSEN, N. H (1999). Function Concepts for Machine Parts - Contribution to a Part Design Theory, *Proc. of ICED 99*, 2, 841-846.

PAHL, G., & BEITZ, W. (1988). *Engineering design - a systematic approach*, The Design Council.

RDF (2001) http://www.w3.org/RDF/

RIEGER, C., & GRINBERG, M (1977). Declarative representation and procedural simulation of causality in physical mechanisms. In *Proc. of IJCAI-77*, 250-256.

ROSENBERG R. C. & KARNOPP, D. C. (1983) Introduction to Physical System Dynamics. McGraw-Hill, 1983.

SALUSTRI, F. A. (1998). Ontological Commitments in Knowledge-based Design Software: A Progress Report, In *Proc. of The Third IFIP Working Group 5.2 Workshop on Knowledge Intensive CAD*, 31-51.

SASAJIMA, M., KITAMURA, Y., IKEDA, M., & MIZOGUCHI, R (1995). FBRL: A Function and Behavior Representation Language, *Proc. of IJCAI-95*, pp.1830 – 1836.

SW, (2001) http://www.semanticweb.org/

SEMBUGAMOORTHY, V. & CHANDRASEKARAN, B. (1986). Functional representation of devices and compilation of diagnostic problem-solving systems, In *Experience, memory and Reasoning*, 47-73.

SMITH, B. & Welty, C. (2001) Prof. of the Second International Conference on Formal ontology and Information Systems: FOIS2001, ACM ISBN 1-58113-377-4 ACM Press

SOWA, J. (1995) Distinction, combination, and constraints, Proc. of IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing.

SUO (2001) http://suo.ieee.org/

TAKEDA, H., VEERKAMP, P., TOMIYAMA, T. & YOSHIKAWA, H. (1990). Modeling design processes, *AI Magazine*, 11(4), 37-48.

TEJIMA, N. ET AL. (eds). (1981) Selection of functional terms and categorization, Report 49, Soc. of Japanese Value Engineering (In Japanese).

TOMIYAMA, T. (2000). A theoretical approach to synthesis, Proc. of 2000 International symposium on modeling of synthesis, pp.25-64.

UMEDA, Y., ISHII, M., YOSHIOKA, M., SHIMOMURA, Y., & TOMIYAMA, T. (1996). Supporting conceptual design based on the function-behavior-state modeler. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 10, 275-288.

VESCOVI, M.; IWASAKI, Y.; FIKES, R.; & CHANDRASEKARAN, B. (1993). CFRL: A language for specifying the causal functionality of engineered devices. In *Proc. of AAAI-93*, 626-633.

WN (1993) http://www.cogsci.princeton.edu/~wn/

XML (2001) HTTP://WWW.W3.ORG/XML/

YOSHIKAWA, H. (1981). General design theory and a CAD system, *Man-machine Communication in CAD/CAM*, ed. By Sata, T. and Warman, E.A., North-Holland, Amsterdam, pp.35-58.