Journal of Engineering Design, Vol. 15, No.4, pp. 327-351, August 2004

Ontology-based systematization of functional knowledge

Yoshinobu Kitamura and Riichiro Mizoguchi
The Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka, Japan, Tel: +81-6-6879-8416, Fax: +81-6-6879-2123
{kita,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, we first overview the state of the art of ontological engineering which we believe is able to make a considerable contribution to resolving these difficulties. We then discuss our enterprise aiming at systematization of functional knowledge used for synthesis. We discuss ontologies that guide conceptualization of artefacts 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 extended device ontology. This paper particularly discusses the extended device ontology and its application in the mechanical domain. 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 artefacts, design know-how used in the conceptual design phase is left implicit because of its subjectivity and implicitness. As discussed in the knowledge management research, 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 rarely been shared within any community. Such a framework should be general enough for being shared by people in different domains 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 skilfully in any sense. Nevertheless, the conventional knowledge technology is not good enough to realize it, which has been shown by the failure of expert systems. However, ontological engineering, which is the successor to knowledge engineering, has been expected to overcome some difficulties the conventional knowledge engineering cannot solve.

The main objective of this paper is to propose an innovative framework for systematization of functional knowledge and its applications to engineering knowledge management through Ontological Engineering (Sowa 1995, Guarino 1997, Mizoguchi and Ikeda 1997, Smith and Welty 2001). Explication of conceptual structure behind the design knowledge using ontological engineering 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 section briefly overviews the ontological engineering and describes the scope of our enterprise. Section 3 describes the skeletal plan of our ontology building for knowledge systematization. Section 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 section 5. In order to demonstrate the utility of the knowledge systematization, an overview of application systems is described in section 6. Section 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.

Although the 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[1].

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 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, it should be more than information retrieval with powerful retrieval functions. It essentially needs content-oriented research because knowledge should be carefully organized and represented to be sharable by many people of different viewpoints. 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 Ontology and design (Literature review)

In this section, we briefly discuss ontologies in engineering domain for design. Such ontologies can be categorized into "task" ontology and "domain" ontology according to the target knowledge. The task ontology for design represents the process of designing activities. Such design task ontologies have been explored as problem-solving method (PSM) research in knowledge engineering community or design methodology research in the engineering design community. For example, pioneer work of analysis of design task have been done by Chandrasekaran (1990). In the design community, there are many generic models of designing and design processes such as (Yoshikawa 1981, Takeda et al. 1990; Gero 1990, 2002, Hubka and Eder 1998). Here, we go into neither their details nor other work, because we concentrate on the domain ontology.

Domain ontology for design is concerned with things to be designed (we call it a "design target" here). When we consider design of only physical systems, the design target can be a product in the case of product design or a manufacturing process in the case of manufacturing design. The domain ontology generally aims at representation of the design targets themselves (such as structure and shape) and/or temporal changes of their physical attributes (so-called behaviour and function).

A part of the general top-level ontology discussed by Guarino (1997) and Sowa (1995) such as mereology gives a basis of the domain ontology for design for representation of physical things. Guarino et al. (1997) points out that STEP (ISO 10303), a very-large domain ontology for engineering domain, lacks clear meaning of modelling entities and necessity of a formal ontology.

Among the domain ontologies specific to the engineering products, the device-centred ontologies have been developed well. It originates from the system dynamics theory. In this ontology, the target is represented as a composition of connected components which have inputs and outputs. For example, Gruber and Olsen (1994) implemented an ontology similar to ours in Ontolingua as the Component-Assemblies Theory. Such a device-centred ontology is a basis of many model-based systems for artefacts for problem-solving not only design but also such as diagnosis. We adopt this type ontology and extend it in Section 4.

As well as the device ontology, the process-centred ontology have been developed such as one proposed by Forbus (1984). In this ontology, a physical process plays a crucial role to change attributes of entities. Physical entities in the target world just participate in the process.

Bond graph is a theory for describing a system domain-independently in the field of system dynamics (Rosenberg and Karnopp 1983). It introduces the

[1] Machine learning is not considered as a method for knowledge *accumulation*. It is just to extract a bunch of knowledge at once which also suffers from knowledge accumulation when combine knowledge learned by other learning methods.

concept of “flow” that represents amount of something which flows in the system and the concept of “effort” which has capability to cause the flow. Bond graph represents the behaviour of the system by combining each behavioural model built in terms of effort and flow.

Borst et al. (1997) propose the PhysSys ontology as a sophisticated lattice of ontologies for engineering domain. It consists of meleology, topology, systems theory, component, process (based on the bond graph) and mathematics for engineering domain (EngMath), which supports multiple viewpoints on a physical system. It forms a basis for a model component library for physical systems.

Although the PhysSys ontology is well-elaborated, it includes no ontology for function from the teleological viewpoint. Needless to say, function plays a crucial role in design. Because the concept of function is highly conceptual and thus functional modelling tends to be ad hoc, explicit ontological commitments are crucial for functional modelling and authoring of functional knowledge. Moreover, from our point of view, the device ontology in PhysSys is weak in that it does not have enough concepts for understanding the *ontological roles* all the participants play. The aims of our extension of the device ontology in Section 4 include these two issues.

One of the pioneering work for such deeper ontological commitments is found in (de Kleer and Brown 1984, de Kleer 1984) for qualitative simulation and causal explanation of how things work. They introduce the concept of “function” as a causal pattern for both keeping context-independency of behavioural models (the No-Function-In-Structure principle (de Kleer and Brown 1984)) and teleological analysis (de Kleer 1984). Moreover, they introduce the concept of “conduit” for causal explanation based on the “mythical time” instead of just “connection”. It clarifies the roles of the simple pipes.

Salustri (1998) emphasizes importance of such ontological commitments for mapping logical structures of description languages to the domain of design knowledge. The identified commitments deal with aspects of description language, of function and behaviour, of mereology based on the device ontology, and of context-sensitivity.

Chandrasekaran and Josephson (2000) clarify meanings of the concept “function” based on ontological consideration. They start with a simple ontology similar to the device ontology and discuss two types of functions, that is, device-centric function and environment-centric function. Although we share this distinction and the attitude towards the ontological analysis with them, we concentrate only on the device-centric viewpoint in this paper.

Horváth 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.

The authors and their colleagues have been involved in research on functional knowledge modelling based on Ontological Engineering for years. Our main goal is to propose a fundamental modelling framework of functional knowledge for sharing by engineers. We proposed a modelling language of function and behaviour called FBRL (Sasajima et al. 1995) and an ontology of functional concepts (Kitamura et al. 2002). The ontology has been applied for automatic identification of functional models (Kitamura et al. 2002) and functional knowledge modelling (Kitamura and Mizoguchi 2003). In this paper, we report philosophical and fundamental issues of such work. Especially, we discuss an extended device ontology as a basis of the modelling framework. Although the extended device ontology has been introduced in (Kitamura and Mizoguchi 2003), this paper discusses the details of definitions and its application to the mechanical domain. We also report deployment of our framework in a production company.

Another motivation of ontologies for design is to exchange models (or knowledge) in different forms and/or from different viewpoints. Liu proposed modelling based on the combination of the device ontology and the charge-carrier ontology that focus on movement of free electrons (or holes) in electronic devices at the microscopic level. (Lui 1992). PACT aims at integrating design tools as distributed agents using KQML and KIF (Cutkosky 1993). Sekiya, Tsumaya and Tomiyama (1999) have been developed the knowledge intensive engineering framework (KIEF) with the pluggable metamodel mechanism for integrating design tools using ontologies of modelling elements in design tools. However, in this paper, we do not deal with this issue.

2.3 Ontology and knowledge systematization

The next topic is how to use ontology. Among many possibilities, the authors believe its use for knowledge systematization is one of the most promising (Mizoguchi and Kitamura 2000). 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.

Let us investigate design knowledge in terms of the two dimensions such as the target knowledge and the way of investigation. The former has two values: product(design object) knowledge and process knowledge. The latter has two values such as domain-dependent and domain-independent ways. Concerning design knowledge systematization, the conventional research on design knowledge is highly domain-dependent and has investigated such knowledge by analytical and quantitative methods. Knowledge base construction for both types of knowledge has been ad-hoc and domain-dependent. On the other hand, there are some domain-independent investigations of design knowledge. 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 process in terms of mathematics to model the input-output relation of design process ignoring the internal sub-processes, and hence it does not cover the dynamic aspects of design. Tomiyama and his colleagues (Tomiyama 2000) have been investigating the research on design process modelling using logic and artificial intelligence and have come up with a deeper understanding of design process, that is, "design process is an abduction". Although this brief observation is not intended to present a thorough overview of the design research, it at least shows that there remains one major issue left untouched, that is, domain-independent investigation of knowledge about design objects. The authors have been mainly attacking this issue sharing the philosophy with other domain-independent research *to reveal the inherent nature of design*.

GENIAL project within the Global Engineering Networking (GEN) initiative also discusses ontologies in the Common Semantic Models for knowledge systematisation (Gausemeier et al. 1997). GNOSIS - Knowledge Systemization for Post-Mass-Production Manufacturing project within the IMS program which we have been involved in also aims at knowledge systematization.

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 section.

3. Functional Ontology and Knowledge Systematization

No one would disagree that the concept of 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 operand that can possess it and of its realization method. The claim has a strong justification that the concept of a function originally came from the user requirements which is totally operand- and behaviour-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 operands 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 et al. 1993, Chittaro et al. 1993, Lind 1994, Sasajima, et al. 1995, Umeda et al. 1996, Hubka and Eder 1998, 2001, Chandrasekaran and 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 a cam and 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 modelling process of artefacts 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 and 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.

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. This ontology is under development and not discussed in this paper. Extended device ontology is developed to provide a common viewpoint which supports to realize consistent interpretation of artefacts. This ontology is discussed in the next section. These two ontologies collectively work as a substrate on which we can build consistent knowledge in upper layers.

The functional concept ontology specifies functional concepts as an instance of the concept of "*function*" defined in the device ontology. Their definitions scarcely depend 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 the abstract part library also belong to this class of knowledge called *general concept layer*.

Way of function 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 the way of function achievement way looks similar to functional decomposition knowledge like that discussed in (Pahl and Beitz 1988), the former is much richer than the latter in that it consists of four kinds of hierarchies of different roles and principles (Kitamura and Mizoguchi 2003). 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 the 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.

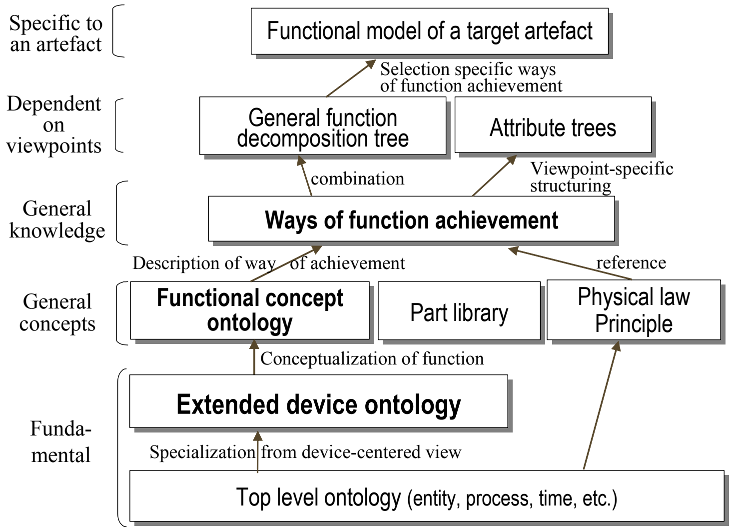

Figure 1. Hierarchy of ontology and knowledge of function

4. Device Ontology

4.1 What is device ontology?

Concerning modelling of artefacts, there exist two major viewpoints: Device-centred and Process-centred views. The device ontology, e.g., one proposed by de Kleer and Brown (1984), specifies the device-centred view of artefacts. Device-centred view regards any artefact as composition of devices which process input to produce output which is what the users need. Process-centred view applied to an artefact, e.g., one proposed by Forbus (1984), concentrates on phenomena occurring in each artefact (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-centred 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 operand 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 operands) in the extended device ontology. The discrimination between behaviour and function is discussed in chapter 5. We exclude static behaviour such as *to support* by concentrating only on dynamic behaviours.

4.2.1 Device and operand

Things that exist in the device ontology world are grouped into two categories: *Device* and *Operand.* 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 *operand* 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

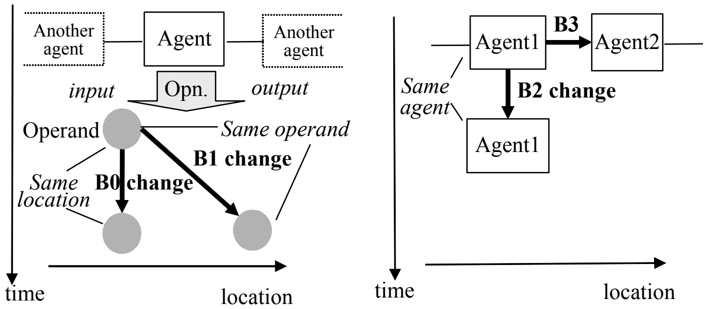

Figure 2. Four different definitions of behaviour

processed by the device. Note that the label of this concept "operand" represents "target" of activity of a device in the same sense in (Hubka and Eder 1998). Examples of an *operand* include substance like fluid, energy like heat, motion, force, information, etc. An *operand* has attributes whose values change over time. A *device* is something that operates on an *operand* that goes through the device. The state change of an *operand* is realized by the difference between the states of the *operand* at the input port and that at the output port.

4.2.2 Conduit and medium

A *conduit* is defined as a special type of a *device* that can be considered as it transmits an *operand* 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 *operand* 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 domains, a 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 Behaviour

We identified four kinds of definitions of *Behaviour*. Figure 2 illustrates simplified situations for behaviour definition. ***B0 behaviour*** is defined as the change of an attribute value of an *operand* 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 behaviour is defined as the change of an attribute value of an *operand* 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 behaviour. The key difference between B0 and B1 is that while B0 behaviour concentrates on the location of the observation rather than identity of the observed *operand*, B1 behaviour on the identity of the observed *operand* rather than the location.

B2 behaviour 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 behaviour of "to fan", "a shaft is twisted" is an inner behaviour of "to transmit torque", etc. This behaviour is based on an answer to the question such as "what motion is the device performing?" and is not the behaviour 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 behaviour is defined as any behaviour to another *device*. The important aspect here is *B0* and *B1 behaviours* are concerned with *operands* rather than *devices*.

All the definitions above share that behaviour 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 behaviour. Another definition of behaviour, which looks very similar to B1 behaviour at first glance, is found in (Chandrasekaran et al. 1993) where function is defined as *B1 behaviour* and the behaviour corresponding to the function is defined as series of *B1 behaviour* of sub-devices of smaller-grain size. That is, in his definition, the difference between function and behaviour is relative to the grain size, which is different from our principle that function is a teleological interpretation of behaviour. In (Bhatta and Goel 1997), the internal behaviour and the output behaviour correspond to the Chandrasekaran's behaviour and our B1 behaviour, respectively.

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 *operand.* 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

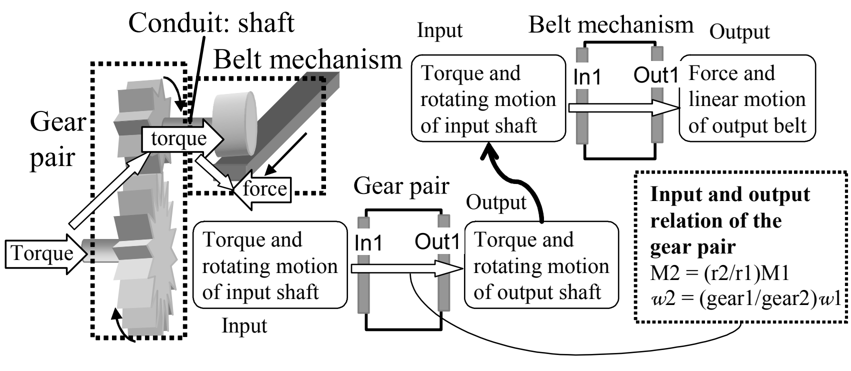

Figure 3(a) A model of a pair of gears at the "mechanism" level

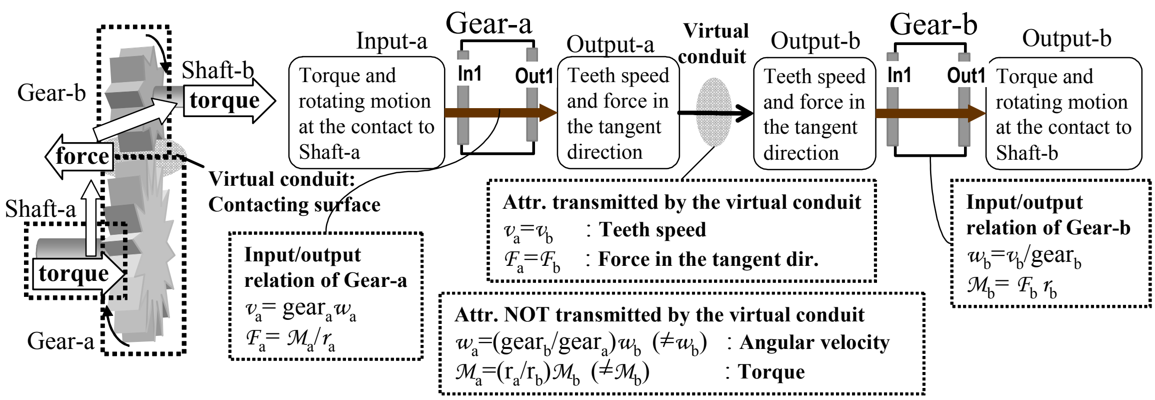

Figure 3(b). A model of a pair of gears at the "mechanical element" level

Table 1 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)
Operand	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.

change from one end to the other end. Let us take an example shown in figure 3(a). 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 3(b). 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 artefacts 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 1 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) In same cases, *medium* does not flow through a *device*, but it allows *operands* to flow by transmitting it through their connection (we call this modelling "non-flowing medium modelling").
(3) Force and motion are *operands* 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 kinematical pair.
(6) *Conduit* is not unique but of variety.

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 1 are based on our research experiences on plant, production process (Mizoguchi and Kitamura 2000) and mechanical system domains for years.

5. Functional Ontology

We now have obtained a framework for building a functional model of an artefact. 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 function achievement* and *method of function achievement* together with functional concept ontology.

5.1 Definition of function

We define a ***Function*** of a device as "teleological interpretation of ***B1 behaviour*** under a given goal" (Sasajima et al. 1995). This tells us a function of a device is determined context-dependently, though *B1 behaviour* 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 behaviour and hence structure.
3. To formulate a function 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-behaviour-function hierarchy

Figure 4 shows Structure-Behaviour-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 artefacts. 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 ***behaviour***, we mean *B1 behaviour*. What is obtained by teleological interpretation of *B1 behaviour* under a given goal is called ***(Base) function***. The term "base" is used to discriminate it from meta-function introduced later.

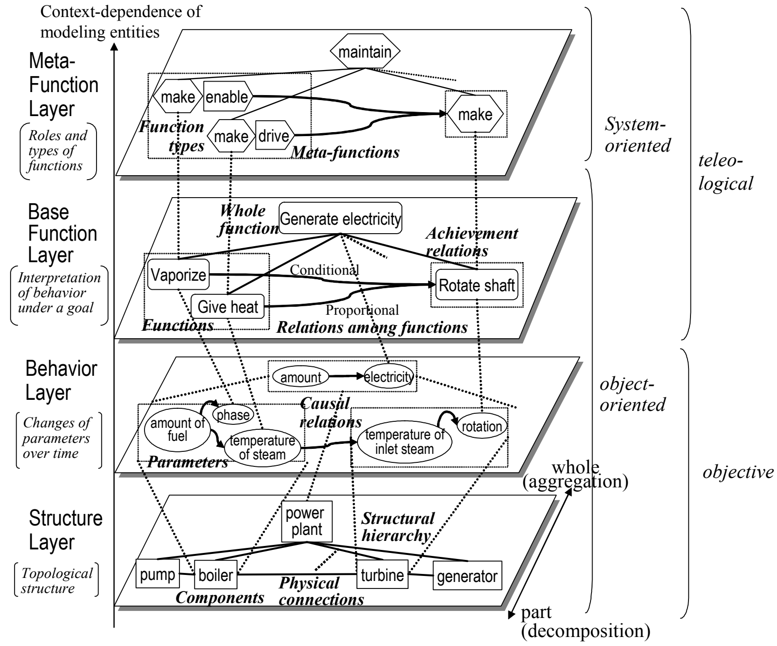

Figure 4. Hierarchy of a target object (a power plant).

A function is achieved by performing(achieving) a series of sub-functions which is called a ***method of function 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 function achievement*** that is considered as reference to the essential property of structure and behaviour that achieve the function.

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 *operands* 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*.

5.3 Function and behavior representation language: FBRL

FBRL: Function and Behavior Representation Language (Sasajima et al. 1995) is designed to ground functional concepts onto behaviour 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, behavioural definition in terms of attributes of *operands* and functional toppings (FT) that enable a system to map a structural and behavioural model to a functional concept. FT is composed of four items:

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

5.4 Functional concept ontology

Functional concept ontology defines three kinds of functional concepts introduced in 5.2 (Kitamura et al. 2002). 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 that for force and motion. Figure 5 shows *is-a* hierarchies of functional concepts for energy (5a) and 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 behavioural 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 behaviour. 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 *operands (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 et al. 2002). The definitions of function types and meta-functions have been discussed in (Kitamura et al. 2002).

6. Roles and Effects of Functional Ontology

6.1 Roles and effects of the extended device ontology

The extended device ontology views an artefact as something that receives input, process and outputs *operands.* The *operand* 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 artefacts produced in other domains. It is not an easy task to build models of a lot of artefacts 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. For example, the concept of a *conduit* helps us consistently recognize devices by taking it as the boundary between the devices. In the

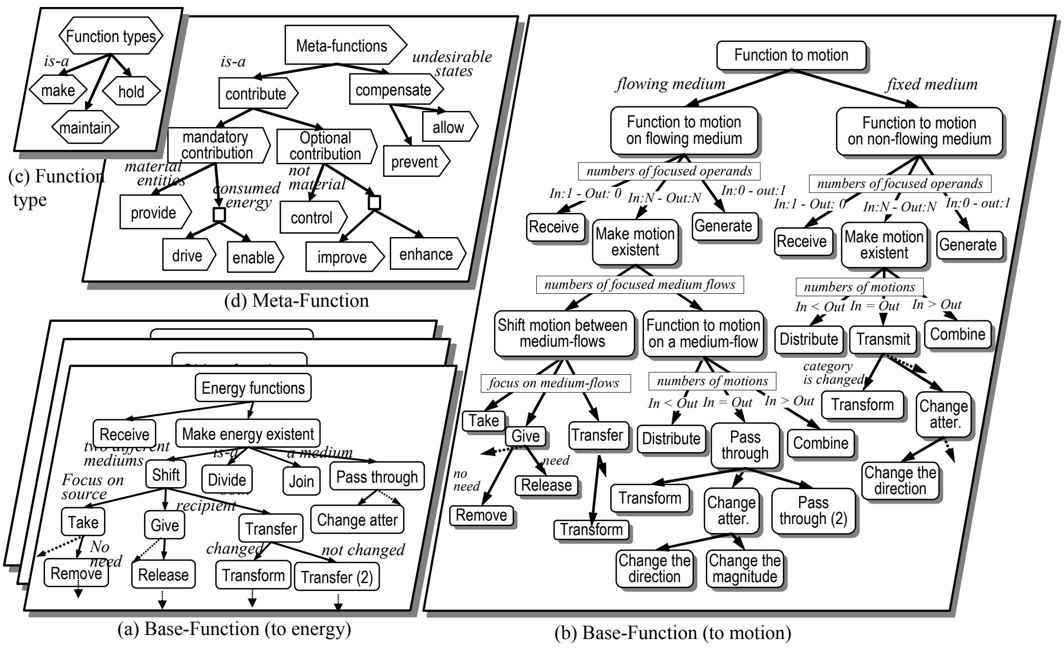

Figure 5. Functional concept ontology (portion)

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 neighbouring devices that are connected to each other only through attributes of an *operand*. On the contrary, composability of the inter-device operation ontology is low due to its high dependence on neighbouring devices.

The extended device ontology provides a unified framework in which one can build compatible models in various domains including mechanical systems and common vocabulary. In fact, terms at the higher level abstraction are common. Although motion has a special attribute of direction not common to others, most of the rest can be treated by a common framework.

Two kinds of ontologies: the extended device ontology and inter-device operation ontology can successfully distinguish two different types of terms such as *to transmit(force)* and *to move(a box)* where the former is based on the (extended) device ontology and the latter on the inter-device operation ontology. Both are incompatible. Figure 5b represents a functional concept hierarchy in the mechanical system domain. These functional concepts are bit unfamiliar to experts in that domain. We found that most of familiar terms in mechanical system domain belong to the inter-device operation ontology which is incompatible to the device ontology, and hence to the model/knowledge in other domains.

6.2 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 satisfied by them. Its utility is summarized as follows:

- **Functional model description**: The functional concept ontology gives generic classes for instances of functions that appeared in the functional model of a specific product. Thus, all functions in the models have explicit meanings based on the extended device ontology. Because functional models represent a part of intentions of designers (so-called design rationale (Chandrasekaran et al. 1993)), explicit representation of functional models helps engineers' mutual understanding. This type of utility is discussed in Section 6.3.
- **Functional knowledge description**: The ways of function achievement in the product models can be generalized into generic pattern knowledge of how to achieve a function. It is called functional way knowledge and consists of a goal function (the macro-function), sub-functions, the relations among sub-functions such as temporal relations and description of physical principle. The classes of functional concepts are used as constraints on sub-functions to be achieved. The well-defined and well-organized functional terms contribute to increase of the sharability and reusability. (Kitamura and Mizoguchi 2003)
- **Explanation generation**: Concept/terms in the functional concept ontology are used as words in explanations generated. Thanks to the operational definitions of them, the selection of words and determination of the abstraction level at which the explanation is generated can be done flexibly (Sasajima et al. 1995).
- **Specification of the inference space**: Problem solving such as design and diagnosis can be done in the functional space that is necessary for taking into account requirements represented in terms of functional concepts. The inference space is specified by the functional concepts organized in the functional concept ontology. Because the functional decomposition knowledge is represented in terms of such functional terms, automatic function decomposition can be done which is used for functional structure improvement system (Kitamura and Mizoguchi 1999).
- **Automatic identification of functional structure**: The ontology enables automatic identification of functional structures of a given structure and device model with the help of meta-functions (Kitamura et al. 2002).

The above types of use and possible utility of our ontologies are investigated by us. Indexing design cases by the functional concepts can be other use (explanation is omitted here):

6.3 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 Ltd., Japan, for sharing functional knowledge of devices used in the daily activities among engineers in the division. The test use and the deployment were started in February 2001 and May 2001, respectively. They have described about 103 functional models of production machines. Currently, about 50 people in two factories use the framework in daily work. As an example, figure 6 shows the function decomposition tree of a wire saw for cutting ingots. 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:

- Descriptions are specific to the target objects
- Knowledge is task-specific
- Vocabulary is not consistent or common
- Much knowledge is left implicit
- To retrieve appropriate report(knowledge) is hard

These are caused by deeper causes:

- There is no principle for representing and organizing functional knowledge
- Every engineer has his/her own viewpoint and there is no common way of how to view a device
- There no common vocabulary representing function
- 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 (named SOFAST(R)) in this deployment is a server of functional models and function achievement way knowledge.

Basically, for a target production facility (in general, it can be a product also), the usage of our framework is categorized into (1) to communicate with other designers about the target facility using its (general) function decomposition tree, (2) to explore causes of a problem of the facility using its function decomposition tree, and (3) to redesign (improve) the target facility using its function decomposition tree and general functional way knowledge. The following give summary of remarkable results in each type of usage in the deployment.

As one of the first usage, the models of ways of function achievement were used as knowledge media for collaborative work by people having different viewpoints such as manufacturing engineers, manufacturing equipment engineers, equipment operators and equipment maintainers. Although mutual understanding and collaboration among them was strongly required, it never happened. The use of our framework, however, enabled them understand and collaborate with each other in a facility improvement project. It turned out that the framework worked as a common vocabulary which lacked before.

In design review activities, the general function decomposition trees are used as required documents (i.e., the designer must submit a tree of the target device) for discussion. As the result, the times of the design reviews has been reduced to one third.

As one of the second usage, a designer was not able to solve a problem of low quality of semiconductor wafers after 4-month investigation. By exploring causes of the problem in the model of ways of function achievement with a clear description of physical principles, he found a solution for the problem within 3 weeks. The reasons of this success can be considered as follows;

- To write a function decomposition tree of the target machine with explicit physical principles makes the designer’s understanding clearer.
- The micro-macro hierarchy of the function decomposition tree enables the designer to explore the possible causes of the problem for each function systematically. The fault tree

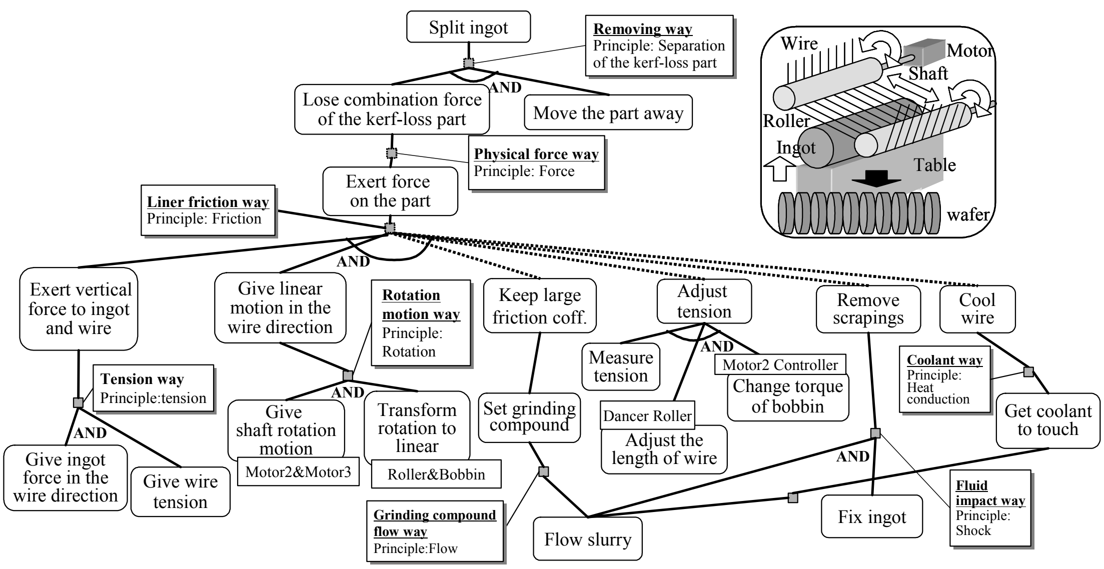

Figure 6. A function decomposition tree of a wire-saw for slicing ingots (portion).

analysis (FTA) tends to be difficult to enumerate all possible causes without clear understanding of function structures.

As one of the last usage, a feasible new improvement of the wire-saw was found from the knowledge-base by adopting the way of using magnetic fluid for controlling tension of the wire. This can be done by applying a way originating from the textile industry to the semiconductor industry. This indicates feasibility of our framework for general functional knowledge.

The success factors include:
- Extended device ontology enables users to be consistent in interpreting how a device works.
- Clear distinction between functional concept (goal) and way (its realization method) makes the knowledge highly domain-independent
- Functional concept ontology provides a rich set of well-defined functional terms
- 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. Discussion and related work

We presented the literature review of ontologies for design in Section 2.2 and definitions of functions in Section 3. Thus, here, we give the remaining notes on related work and limitation of our work.

7.1 Domain ontologies for design

Among types of ontologies discussed in Section 2.2, we concentrate on not the task ontology of design activities but the domain ontology of artifacts to be designed. On the basis of the conventional device ontology (de Kleer and Brown 1984, Gruber and Olsen 1994), we extended it for dealing with the mechanical domain. Mortenesen (1999) reports on a negative observation on the applicability of device ontology to mechanical elements. We gave wider definition to the concept of "conduit" introduced by de Kleer and Brown (1984) for the virtual conduit at the mechanical element level. We define the concept of "medium" in mechanical domain precisely. As discussed section 6.1, our refinement of the device ontology gives the modeller a more detailed guideline to capturing the target devices.

Although we do not adopt the process ontology in this paper, our functional concept ontology includes the flow-based functions that are found in the flow-based functional modelling approaches (Chittaro et al. 1993, Lind 94).

In comparison with the work by Horváth et al (1998), 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. In this sense, the "conduit" as conceptualization of connection corresponds to the nucleus. The Horváth's group and we are doing collaborative research on the integrated modelling scheme based on our ontologies and the nucleus theory.

7.2 Definition of function

There are quite amount of research on functional representation (de Kleer 1984, Sembugamoorthy and Chandrasekaran 1986, Gero 1990, Chandrasekaran et al. 1993, Chittaro et al. 1993, Lind 1994; Umeda et al. 1996, Bhatta and Goel 1997, Chandrasekaran and Josephson 2000). We focus on not purpose functionh but technical functions in the terminology in (Hubka and Eder 1998, 2001). The features of our definitions of functional concepts can be summarized as follows;
(1) Detachment the macro-micro relation from the teleological interpretation,
(2) Detachment the ways of function achievement from functions themselves.
(3) Operational grounding of designer's intention with behaviour, and
(4) Introduction of "meta-functions".

Concerning the first point, we discriminate between the macro-micro hierarchy (the *is-achieved-by* relation among functions, aggregation relation of functions) and the teleological interpretation (the behaviour-function relation). Lind (1994) emphasizes this discrimination and calls them the whole-part relation and the means-ends relation, respectively. Umeda et al. (1996) also share it with us. Hubka and Elder (1998, 2001) distinguish "degree of complexity" from "aims-means" relation. The former corresponds to our micro-macro relations. The latter is among design requirements, black box representation of the Technical Systems (TS), its function structure, its organ structure, and its component structure. Although our modelling framework does not include explicit representation of the design requirements, we focus on teleological (subjective) interpretation of objective behaviours. In (Gero and Kannengiesser 2002), dynamic changes of the design context such as the requirements are coped with. Andreasen et al. (1996) identify several structures including not only "functional oriented structure" but also "product life oriented structure" for so-called DFX: Design for "something".

On the other hand, as pointed out in Section 4.2, in (Sembugamoorthy and Chandaraseakaran 1986), functions are defined as a kind of hierarchical abstraction of behaviour at more microscopic level. In the SBF model (Bhatta and Goel 1997), the internal behaviour means decomposition of behaviour and functions. Chandrasekaran and Josephson (1996) point out the importance of implementation-independent functional models and then propose representation of function as effect.

The above discrimination helps us to keep the functional concepts representing "what is achieved" as the second point. For example, "to weld steel" is not just a function but a function implying a specific way of achievement, that is, fusing the target operand. Such concepts should be decomposed into "joining" function and "fusion" way.

The *is-a* hierarchy of the functional concepts in the functional concept ontology represents abstraction of teleological interpretation of "what is achieved". In (Hubka and Eder 1998), the hierarchy of "degree of abstraction" of functions represents specialization of functions with additional conditions. The conditions sometimes (not always) may include characteristics of a concrete way of function achievement such as "transportation by sea". We detach such conditions and thus describe them as specific attributes of a way of function achievement. In our framework, the *is-a* hierarchy is organized according to not derivative features but physical principles of the achievement (Kitamura and Mizoguchi 2003).

By the third point, we mean that the functional concepts can be automatically mapped from behavioural models by additional teleological information called FTs (Kitamura et al. 2002). Many "verb+noun"-style functional representations lack such operationality. For example, standard sets of verbs (i.e., functional concepts) proposed for value analysis in (Tejima et al. 1981) have no machine understandable definition of concepts.

FTs represent designers' intention including focus on operands and necessity of operands, and then enable us to define intention-rich functional concepts. In some of research (e.g., the SBF model (Bhatta and Goel 1997)), function is just "intended" behaviour, thus there is no compositional difference between behaviour and function. De Kleer defines function as a causal pattern between variables in his early work on teleological analysis (de Kleer 1984). In the FBS model (Umeda et al. 1996), the functional symbol in natural language in the verb+noun style represents intention of designers. We try to identify operational primitives for representing intention. Keuneke (1991) defines types of functions such as ToMake. Our FTs include them.

We also introduced a new type called meta-function. The CPD in CFRL (Vescovi et al. 1993) represents causal relations among functions. Lind (1994) categorizes such relations into Connection, Condition and Achieve. Rieger and Grinberg (1977) identify "enablement" as a type of the causal relation between states and action. Hubka and Eder (2001) categorize the assisting functions into the auxiliary functions, the propelling functions, the regulating functions and so on. The meta-functions are results of interpretation of such causal relations between functions under the role of the agent function for the target functions without mention of the operands associated with components. The consolidation theory (Bylander & Chandrasekaran 1985) tries to capture the general rationales of consolidation of components. While we share the goal, their consolidation rules are simpler than ours and depend heavily on topological relations (e.g., series and parallel) between the limited behavioural primitives.

7.3 Ways of function achievement

In design literature such as (Paul and Betiz 1988), patterns of function achievement so-called design catalogs can be found. However, they mainly concentrate on concrete mechanical pairs. In (Gero 1990; Bradshaw and Young 1991; Bhatta and Goel 1997; Umeda et al. 1996), similar ideas to our idea of way of function achievement for general functions are discussed. The major differences between the two include explicit description of "way" and generic knowledge based on a functional concept ontology.

Firstly, our ways of function achievement are explicit conceptualization of the feature of achievement such as theory and phenomena at the behavioural level. Such functional knowledge is compliant with the observations found in the research on design processes (Takeda et al. 1990) in which it is claimed that function decomposition is not done solely in the functional space but also by going back and forth between the functional, behavioural and structural spaces. Although capturing of feature of function decomposition is also found in (Malmqvist 97), it is not generic knowledge but a model specific to a product. The design prototypes proposed by Gero (1990) include structural decomposition as well as function decomposition. In the FBS modelling framework (Umeda et al. 1996), a function prototype includes the physical feathers of behaviour realizing the function as well as generic function decomposition. Our description of ways tries to maximize its generality by pointing partial (and abstract) information of structure and behaviour.

Secondly, our functional knowledge is based on the functional concept ontology (Kitamura et al. 2002). Use of generic functional concepts in *is-a* hierarchies facilitates reuse of the knowledge in different domains. In IDEAL in (Bhatta and Goel 1997), generic teleological mechanisms (GTM) generalized from case-specific SBF models are used (modified) in design for different context based on analogy. In our approach based on the limited set of functional concepts, the ways of function achievement for specific function are organized in the *is-a* hierarchy. Designers can explore them in several abstract levels explicitly.

The TRIZ theory gives some patterns (or strategies) of inventions based on contradiction between two physical quantities (Sushkov et al. 1995). We agree on the importance of cross-domain knowledge which is pointed out in the TRIZ theory and thus reusability of functional knowledge is one of our goals. However, we concentrate not on design strategies but on a modelling schema of such generic knowledge and the design target. The TRIZ theory also pays attention on physical principles (effects), though we make a clear relationship between physical principle and the functional structures.

7.4 Limitation and trade-off

The main point of our work is the ontological approach for the space in functional level. All conceptual entities in the functional models are instances of predefined classes in the functional concept ontology. In other words, the functional concept ontology specifies the space of functions and limits it. This approach enables us to avoid ad hoc modelling and obtain consistent functional models. On the other hand, it may reduce the freedom of functional representation in comparison with hand-written functional models. The cost of modelling depends on generality (reusability) of functional concepts defined in the ontology. Currently, we have defined 274 concepts including intermediate concepts and duplication of functions on different target operands. The ontologies have been applied to modelling of a power plant, an oil refinery plant, a chemical plant, a washing machine, a printing device, and manufacturing processes. Their models include changes of thermal energy, flow rate, and ingredients of fluid, force and motion of operands. The current functional concept ontology can describe simple mechanical products, though it does not cover static force balancing and complex mechanical phenomena based on the shape of operands. The modelling framework currently does not cope with human's philological process, movement of human's body (so-called therblig in Industrial Engineering), business processes and software processes.

We cannot claim completeness of the concepts in our functional concept ontology. Although one might think that the set of functional concepts is huge, not the set of function but of the set of ways of function achievement is very large. In fact, in Value Engineering research (Tejima et al. 1981), 158 verbs are proposed as a standard general set for representing functions of artefact. Although it includes functions for human sense as well, we concentrate on functions changing physical attributes.

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 function decomposition. The idea of "way" of function achievement plays a key concept for systematization.

We 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 authors would like to thank Dr. Toshinobu Kasai, Dr. Kouji Kozaki, Mr. Kouki Higashide, Mr. Toshio Ueda, Mr. Toshinobu Sano, Mr. Masaru Takahashi, Ms. Mariko Yoshikawa and Mr. Tomonobu Takahashi for their contributions to this work. Special thanks go to Dr. Masayoshi Fuse, Mr. Masakazu Kashiwase and Mr. Shuji Shinoki, Sumitomo Electric Industries, Ltd. for their cooperation of the deployment. The authors are grateful to Mr. Wilfred van der Vegte and Prof. Imre Horváth, Delft University of Technology and Prof. Mitsuru Ikeda for their 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

Andreasen, M.M., Hansen C.T. and Mortensen, N. H., 1996, The Structuring of Products and Product Programmes, In *Proc. of the 2nd WDK Workshop on Product Structuring*, 15-43.

Bhatta, S. R. and Goel, A. K., 1997, A Functional Theory of Design Patterns, In *Proc. of IJCAI-97*, (San Francisco CA: Morgan Kaufman), 294-300.

Borst, P., Akkermans, H., and Top, J., 1997, Engineering Ontologies, *Int'L Journal of Human-Computer Studies*, 46(2/3), 365-406.

Bradshaw, J. A. and Young, R. M., 1991, Evaluating Design Using Knowledge of Purpose and Knowledge of Structure. *IEEE Expert*, 6(2), 33-40.

Bylander, T., and Chandrasekaran, B., 1985, Understanding Behavior Using Consolidation, In *Proc. of IJCAI-85* (San Francisco CA: Morgan Kaufman), 450-454.

Chandrasekaran, B., 1990, Design Problem Solving: A Task Analysis, *AI Magazine*, 11(4), 59-71.

Chandrasekaran, B., Goel, A. K., and Iwasaki, Y., 1993, Functional Representation as Design Rationale, *Computer*, 26(1), 48-56.

Chandrasekaran, B. and Josephson J. R., 2000, Function in Device Representation, *Engineering with Computers,* 16(3/4), 162-177.

Chittaro, L., Guida, G., Tasso, C. and 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.

Cutkosky, M.R., Engelmore, R. S., Fikes, R.E., Genesereth, M.R., Gruber, T.R., Mark, W.S., Tenebaum, J.M., Weber, J.C. 1993, PACT: An Experiment in Integrating Concurrent Engineering Systems, *Computer*, 26(1), 28-37.

De Kleer, J. and 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.

Forbus, K. D., 1984, Qualitative Process Theory, *Artificial Intelligence*, 24, 85-168.

Gausemeier, J., Seifert, L., Joosten, M., Kesteloot, P. and Monceyron, J. L., 1997, GENIAL: The logical Framework for the Systematisation of Internal and External Engineering Knowledge, In Proc. of ICED 97.

Gero J.S., 1990, Design Prototypes: A Knowledge Representation Schema for Design, *AI Magazine,* 11(4), 26-36.

Gero J. S. and Kannengiesser, U., 2002, The Situated Function-Behaviour-Structure Framework, In Proc. of Artificial Intelligence in Design '02, (Dordrecht, Kluwer Academic Publishers), 89-104.

Gruber, T., and Olsen G., 1994, Theory Component-Assemblies, Ontology Server, http://www-ksl.standford.edu

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

Guarino, N., Borgo, S., Masolo, C., 1997, Logical modelling of Product knowledge: Towards a well-founded Semantics for STEP, In *Proc. of European Conf. on Product Data Technology*.

Horváth, I., Vergeest, J. S. M., Kuczogi, G., 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.

Hubuka, V. and Eder, W. E., 1998, *Theory of Technical Systems*, (Berlin: Springer-Verlag).

Hubka, V. and Eder, W. E., 2001, Functions Revisited, In *Proc. of ICED 01*, (London: Professional Engineering Publishing), pp. 69-76.

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

Kitamura, Y., and Mizoguchi, R., 1999, Towards Redesign based on Ontologies of Functional Concepts and Redesign Strategies, *Proc. of the 2nd International Workshop on Strategic Knowledge and Concept Formation*, (Iwate, Japan: Iwate Prefectural University), pp.181-192.

Kitamura, Y., Sano, T., Namba, K., and Mizoguchi, R., 2002, A Functional Concept Ontology and Its Application to Automatic Identification of Functional Structures, *Advanced Engineering Informatics*, 16(2), 145-163.

Kitamura, Y., and Mizoguchi, R., 2003, Ontology-based description of functional design knowledge and its use in a functional way server, *Expert Systems with Application*, 24(2), 153-166.

Kozaki, K., Kitamura, Y., Ikeda, M., and Mizoguchi., R., 2002, Hozo: An Environment for Building/Using Ontologies based on a Fundamental Consideration of "Role" and "Relationship", In *Proc. of the 13th International Conference Knowledge Engineering and Knowledge Management (EKAW2002),* (Berlin: Springer), 213-218.

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

Liu, Z., 1992, Integrating Two Ontology for Electronics, In *Recent Advances in Qualitative Physics*, 153-168 edited by B. Faltings and P. Struss, (Cambridge: MIT Press)

Malmqvist, J., 1997, Improved function-means trees by inclusion of design history information, *Journal of Engineering Design,* 8(2), 107-117.

Miles, L.D., 1961, *Techniques of value analysis and engineering* (New York: McGraw-hill)

Mizoguchi, R., and Ikeda, M., 1997, Towards Ontology Engineering, *Proc. of The Joint 1997 Pacific Asian Conference on Expert systems / Singapore International Conference on Intelligent Systems,* (Singapore: Nanyang Tech. University), 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. and 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, (Berlin, Springer-Verlag)

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

Pahl, G., and Beitz, W., 1988, *Engineering design - a systematic approach*, edited by K. Wallace (London: The Design Council)

Rieger, C., and Grinberg, M., 1977, Declarative representation and procedural simulation of causality in physical mechanisms. In *Proc. of IJCAI-77,* (San Francisco CA: Morgan Kaufman), 250-256.

Rosenberg R. C. and Karnopp, D. C., 1983, Introduction to Physical System Dynamics. (New York: McGraw-Hill).

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., and Mizoguchi, R., 1995, FBRL: A Function and Behavior Representation Language, *Proc. of IJCAI-95*, (San Francisco CA: Morgan Kaufman), pp.1830 – 1836.

Sekiya, T., Tsumaya, A., and Tomiyama, T., 1999, Classification of Knowledge for Generating Engineering Models - A case study of model generation in finite element analysis -, in Finger, S., Tomiyama, T., and Mäntylä, M. (eds.), *Knowledge Intensive Computer Aided Design*, 73-90, (Boston: Kluwer Academic Publishers)

Sembugamoorthy, V. and Chandrasekaran, B., 1986, Functional representation of devices and compilation of diagnostic problem-solving systems, In *Experience, memory and Reasoning,* edited by J. L. Kolodner and C. K. Riesbeck, (Mahwah, NJ: Lawrence Erlbaum Assoc.), 47-73.

Smith, B. and Welty, C., 2001, Prof. of the Second International Conference on Formal ontology and Information Systems: FOIS2001 (New York: ACM Press)

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

Sushkov, V.V. Mars, N.J.I., and Wognum, P.M., 1995, Introduction to TIPS: a Theory for Creative Design, *Artificial Intelligence in Engineering*, 9(3), 177-189.

Takeda, H., Veerkamp, P., Tomiyama, T., and 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, (Tokyo, 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., and 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. and Chandrasekaran, B., 1993, CFRL: A language for specifying the causal functionality of engineered devices. In *Proc. of AAAI-93*, (Menlo Park, CA: AAAI Press), 626-633.

Yoshikawa, H., 1981, General design theory and a CAD system, *Man-machine Communication in CAD/CAM*, eds. Sata, T. and Warman, E.A., North-Holland, Amsterdam, 35-58.