Deployment of an Ontological Framework of Functional Design Knowledge

Yoshinobu Kitamura[*1], Masakazu Kashiwase[*2], Masayoshi Fuse[*2*3], and Riichiro Mizoguchi[*1]

[*1] 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

[*2] Plant and Production Systems Engineering Division, Sumitomo Electric Industries, Ltd.

[*3] Currently, Sumitomo Wiring Systems, Ltd.

Abstract

Although importance of knowledge sharing among designers has been widely recognized, the knowledge about functionality in the conceptual design phase is hard to capture and is often scattered across technical domains. Aiming at capturing such functional knowledge easily applicable to other domains, we have developed an ontological framework for its systematic description. It includes six kinds of knowledge about functionality, that is, two types of functional models, two types of organization of generic knowledge, and two ontologies of functionality. This paper reports on a successful deployment of the framework in a production company. The Plant and Production Systems Engineering Division of Sumitomo Electric Industries uses our framework for sharing functional design knowledge of production systems since May, 2001. The empirical evaluation by the Sumitomo engineers was unanimously positive. They said that this framework enabled them to explicate the implicit knowledge possessed by each designer and to share it among team members. This paper discusses some successful use-cases in the tasks such as design review, patent application and solving a quality problem. We also discuss effects of our ontological framework as a consistent viewpoint for capturing implicit functional knowledge and as a conceptual inter-lingua among designers. Limitation of our framework is also discussed.

Keywords

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

1. Introduction

The importance of knowledge sharing among designers and engineers has been widely recognized in knowledge-intensive engineering. Although sharing of CAD data is well done by the recent CAD and computer network technologies, such data does not include knowledge about functionality which plays a crucial role in explicating how thing work. There is no common definition of functionality (Umeda and Tomiyama 1997, Chittaro and Kumar 1998, Chandrasekaran and Josephson 2000, Hubka and Eder 2001), though it at least is related to designer's intention. For example, a functional structure (Pahl and Beitz 1988) shows how to achieve the whole goal of a product for users by sub-functions of components and sub-systems. Such a product model from the viewpoint of functionality is called a functional model. Functional models represent a part of (but not all of) designer's intentions so-called design rationale (Chandrasekaran, et al. 1993).

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. For example, in design review activities, explicit description of designer's intention helps other people understand original designs effectively. Moreover, it can facilitate deep reflection of designs by the designers themselves as well.

A lot of research has been carried out on representation of functionality in Value Engineering (Miles 1961), engineering design research (Pahl and Beitz 1988, Gero 1990, Andreasen, et al. 1996, Hubka and Eder 1998, 2001), and Functional Representation research (de Kleer 1984, Sembugamoorthy and Chandrasekaran 1986, Keuneke 1991, Chittaro et al. 1993, Lind 1994, Sasajima, et al. 1995, Umeda, et al. 1996, Bhatta and Goel 1997, Malmqvist 1997, Bracewell and Wallace 2001, Deng 2002). On the other hand, practical design diagrams such as QFD (Quality Function Development), FMEA (Failure Mode and Effect Analysis) sheets and fault trees in the Fault Tree Analysis (FTA) include functional knowledge. However, there is a gap between theoretical work and daily work in companies. From the experience of two of the authors who work in a production company, engineers have been suffering from the difficulty in sharing technical (functional) knowledge among them for long years. They have been regularly writing technical reports for design review, maintenance history, FMEA sheets, etc. and

have stored a lot in databases. Unfortunately, however, it has been difficult for them to understand a report written by other engineers and hence few of such technical documents are efficiently used. The reasons include:

- It is hard to describe implicit functional knowledge systematically.
- To retrieve appropriate knowledge is hard.
- Knowledge is specific to a target product or equipment.
- Representation framework such as FMEA and FTA is task-specific.

The authors have been tackling these real problems in the industry in many years on the basis of Ontological Engineering and then established an ontological framework for functional knowledge. (Kitamura et al. 2002, Kitamura and Mizoguchi 2003a,b,c). It is based on functional ontologies which provide viewpoints and vocabulary for capturing functional knowledge in order to solve the above problems (as discussed in the next section).

This framework has been successfully deployed in the production systems division of Sumitomo Electric Industries, Ltd. This paper discusses usages and effects of the ontologies in the deployment in Section 4 after detailed analysis of the problems in Section 2 and overview of our framework in Section 3. A knowledge management software named SOFAST® has been developed in the deployment. Its architecture is mentioned in Section 5. The success factors and limitations are analyzed in Section 6. Then, related work is discussed followed by concluding remarks.

Although a lot of research on Ontological Engineering has been done in the last decade, little is known about deployment in industries. This research focuses not on generic mechanism of ontologies but on real contents of well-focused target knowledge, that is, the functional knowledge which is still so generic that it could be applied to all the artifacts. This paper shows effects of ontologies in a deployment based on fundamental ontological analysis of functional representation.

2. Ontological approach for sharing functional knowledge

A functional representation consists of descriptions of functionality of each component (or (sub-)system) and relationship among them. We claim that it is not a trivial task to clearly identify function-related concepts and relations as is explained below. We think it is one of the deep reasons of the problems in the industry mentioned in Introduction.

In Value Engineering (or in similar ways in functional representation), functionality of a component is denoted as "verb+num" style for representing component's activities (or actions) and its operands (in the terminology in (Hubka and Eder 1998)), respectively. However, such representation cannot prevent an inappropriate modeling. For example, one might describe "to weld objects" as a function of a manufacturing equipment. However, "to weld" implies not only "what to achieve" but also "how to achieve it" in which the objects are fused. In fact, the same goal can be achieved in different methods (e.g., using bolts and nuts) without fusion. To allow freedom in design and to make selection of "bolt & nut" instead of "welding" possible, the achieved function of both of the methods should be the same; say, "to join".

In (Pahl and Beitz 1988), some sets of a few (4-16) generally-valid functions are defined. However, they are too abstract to describe details of designer's intention. In fact, there are general-specific relations (so-called *is-a*, *a-kind-of*, abstraction, or specialization relations) among functions. For example, in (Hubka and Eder 1998), hierarchy of "degree of abstraction" of functions represents specialization of functions with additional conditions. The conditions, however, sometimes (not always) may include characteristics of a specific way of achievement such as "transportation by sea" (Hubka and Eder 1998) in the same manner of "welding".

This difficulty in capturing functions and their relations is a special case of a general problem treated in Ontological Engineering that is hard to distinguish *is-a* (general-specific) relation from *part-of* relation (so-called whole-part, micro-macro, decomposition, or aggregation relation). The *part-of* relation among functions represents that how a function is achieved by finer-grained functions (we call this relation "*is-achieved-by*" relation) and has been captured as the function decomposition in (Paul and Beitz 1986), the whole-part relation in (Lind 1994) and "degree of complexity" in (Hubka and Eder 1998). Nevertheless, confusion by engineers is hard to avoid as shown above examples.

These observations suggest a necessity of ontological schema for functional knowledge. An ontological schema specifies not only data structure but also conceptual viewpoint for capturing the target world (called a specification of a conceptualization (Gruber 1993)). It gives guidelines or constraints on models, which help knowledge authors describe knowledge in a consistent way. An ontological schema for functional knowledge includes a fundamental ontology for capturing functions and clear organization of concepts and relationships, which help a knowledge author detach "what to achieve" from "how to achieve it".

This paper mainly concentrates on such roles of ontologies in the knowledge capturing and organizing phase. On the other hand, the roles of ontologies in

knowledge exchange and communication in engineering domain have been investigated on different ones from ours (Lui 1992, Gruber, et al. 1992, Cutkosky 1993, Sekiya, et al. 1999).

In design literature, the systematic design approach (Pahl and Beitz 1988) provides us a basic viewpoint to capture functions, in which functions are regarded as input-output relations of a black-box. The black-boxes are connected and aggregated (or decomposed). In Ontological Engineering, such device-centered viewpoint originated from systems dynamics theory is called *device ontology*. Its examples are shown in (de Kleer and Brown 1984, Gruber and Olsen 1994, Borst et al. 1997, Kumar and Upadhyaya 1998, Chandrasekaran and Josephson 2000). For establishing ontologies for functions, such device ontology is suitable as its basis, because functions are usually considered as what components or devices have.

We have established an ontological modeling framework, whose features include;

- An extended device ontology: Refined device ontology for capturing behaviors of components (Kitamura and Mizoguchi 2003c).
- A functional concept ontology: to provide generic functional concepts representing verbs of functions in *is-a* hierarchies (Kitamura, et al. 2002).
- Conceptualization of "ways of function achievement" and their *is-a* hierarchy for detaching them from functions (Kitamura and Mizoguchi, 2003a).
- Four types of functional knowledge and ontological modeling guidelines (Kitamura and Mizoguchi 2003b).
- Integration of information of unintended use for maintenance activity (van der Vegte, et al. 2002, Koji, et al. 2003).

Such ontological commitment helps designer explicate own thought of his/her design and share it in design team as shown in the deployment in Section 4. The last feature is to solve the last problem mentioned in Introduction.

3. An Ontological Modeling Framework

Our framework for functional-knowledge modeling is shown in Figure 1. This framework is an extension of our functional modeling language FBRL (abbreviation of a Function and Behavior Representation Language) (Sasajima, et al. 1995). It shows a modeling process from a functional model of a concrete artifact to well-organized generic knowledge. It includes six kinds of knowledge about functionality.

This modeling framework is based on an extended device ontology (Figure 1(f)) as a basis of conceptualization of the functional world. The authors have extended conventional one mentioned in the previous section by redefining the concepts of "conduit" and "medium" in order to give knowledge authors richer ontological guidance and to cope with mechanical domains which seemingly do not fit device ontology(Mizoguchi and Kitamura 2000, Kitamura and Mizoguchi 2003c). The extended device ontology specifies "roles" played by participants in the physical world.

Based on the extended device ontology, "behavior" of a device is defined as the objective (independent of designer's intention) interpretation of its input-output relation considering the device as a black box. The description of the behavior is independent of the system (i.e. context) in which it is embedded. Each device is connected to one another through its input or output ports. A device plays a role as an "agent" which changes states of something input (called "operand", that is, a thing being processed by the device) such as fluid, energy, motion, force and information. The input-output relation of the behavior is, to be exact, the difference between the states of the operand at the input port and that at the output port. A

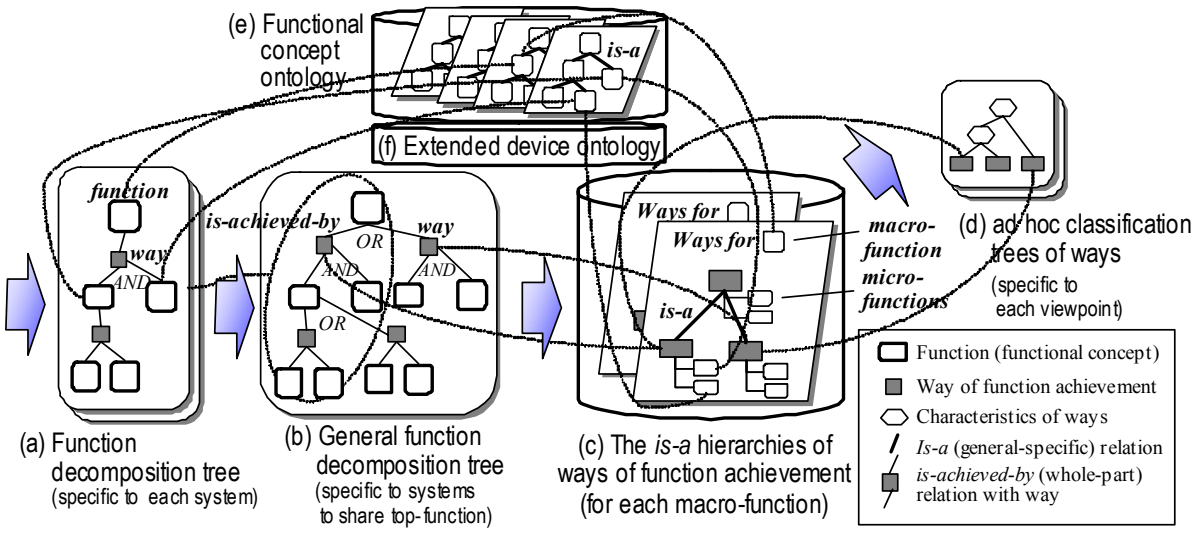

Figure 1. A framework for functional-knowledge modeling.

device can be a mechanical element, a mechanical pair, a component, an assembly, a sub-system, and a system. Those include both products and manufacturing machines.

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 for liquid, a shaft for torque, etc. One of the reasons to have the concept of conduit is to neglect the function "to transmit" common to all the devices which transmit input things to the output. A "medium" is something that holds an operand and enables it to flow among devices. Its example is steam for heat energy. In some domains, a thing playing the conduit role can play the role of medium as well. For example, while a shaft is a conduit for force and motion, at the same time, it plays the role of medium for them.

A function is defined as a teleological interpretation of a behavior under an intended goal (Sasajima, et al. 1995). Although selection of a function (i.e., interpretation of behavior) is dependent on the (sub-)system in which it is embedded, definitions of functions themselves can be done locally. Such detachment of functional interpretation from micro-macro relation is also found in (Lind 1994, Umeda, et al. 1996, Hubka and Eder 1998), but is different from (internal) behavior in (Chandrasekaran et al. 1993, Bhatta and Goel 1997). However, we explicate mapping primitives between behavior and function (called functional toppings (FTs) (Sasajima et al. 1995)) and operational conceptualization of the functional concepts.

The authors have developed an ontology of generic functions (called the functional concept ontology (Figure 1(e)) with such operational definitions (Kitamura et al. 2002). It defines about 220 concepts in four kinds of *is-a* hierarchies.

On the basis of these two ontologies, firstly, a function decomposition tree (Figure 1(a) and examples are shown in Figure 2 and 3) models a functional structure of a specific device. All functions (rounded box nodes in the tree) in the functional decomposition tree are instances of generic functions defined in the functional concept ontology. It basically represents that a required function (called a macro-function) can be achieved by the sequence of specific sub(micro)-functions. This relation is a kind of "part-of" relations or aggregation relations among functions.

In this framework, "the reason why a function can be achieved" is conceptualized as "way of function achievement". It explicates background knowledge of functional decomposition such as physical principles and theories. In Figure 1(a), a way is represented by a small black square that connects the whole function and the sub functions. Physical principles as bases of function achievement are captured as "means" in (Malqvisit 1997, Bracewell and Wallace 2001, Wilhelms 2003).

Moreover, it includes unintended behaviors (van der Vegte, et al. 2002, Koji, et al., 2003). It is needed for explicating design rationale of supplementary functions for preventing them. It is important in design review activity and equipment improvement discussed in Section 4.

Secondly, a *general function decomposition tree* (b) is composed of some function decomposition trees of similar devices having the same whole-function. It includes alternative ways of function achievement in OR relationship. It represents possible ways to achieve a specific function. An example is shown in Figure 4.

Lastly, a concrete way in a (general) function decomposition tree is generalized into a generic way (called functional way knowledge). Then, ways to achieve the same function is organized in *is-a* relations according to their principles (called an *is-a hierarchy of ways of function achievement* (c) and examples are shown in Figure 5). We distinguish the organization as an *is-a* hierarchy from the other derivative organizations dependent on viewpoints (called an *ad hoc classification tree* (d)). The ad hoc classification trees can be reorganized by a functional way server according to a given viewpoint (Kitamura and Mizoguchi 2003a). Such generic functional knowledge is somewhat similar to those in (Gero 1990; Bhatta. and Goel 1997; Umeda et al. 1997). We will discuss differences in Section 7.

The most important point in this conceptualization is that these types of trees concerning functions in Figure 1 are different from each other in spite of the superficial similarity. The function decomposition tree (a) represents *is-achieved-by* (a kind of *part-of*) relations among functions. The *is-a* hierarchies of ways (c) represent an abstraction of the key information about how to achieve the function, while the *is-a* hierarchies in the functional concept ontology (e) represent abstractions of functions themselves, that is, the concept of achievement, that is, what to achieve. Moreover, the numbers of the ways for a function are huge in nature, while the numbers of functional concepts are small.

Currently, the guidelines for building these trees are being developed and are concerned with agents and operands of functions, relations among sub-functions, and the "is-achieved-by" relations (Kitamura and Mizoguchi 2003b). They help a modeler capture functional structures based on the extended device ontology. For example, because sub-functions must contribute to achieve the macro-function clearly on the basis of the physical principles represented as the way of function achievement, a modeler should check existence of implicit functions.

4. Deployment

The ontology and the modeling framework of functional knowledge have been being deployed in over two years at the plant and production systems engineering division of Sumitomo Electric Industries, Ltd. (hereinafter referred to as SEI). The purpose is to share functional knowledge about production facilities used in the daily activities among engineers in the division.

After one-year study of our framework, the test use was started in February 2001. In May, 2001, use in real problems in daily work was started A knowledge management software named SOFAST (discussed in the next section) has been deployed since December, 2002. Currently about 50 people in three factories use the framework in daily work. The database of SOFAST software includes 103 (generic) function decomposition trees in real work. The targets are manufacturing facilities mainly used in semiconductor manufacturing processes including slicing machines for semiconductor ingots (wire-saw type and inter-blade type), polishing machines for wafers, a tension control system, an optical fiber connector adjusting machine and inspection machines.

Firstly, function decomposition trees are described for sharing understanding of the target facilities as discussed in Section 4.1 and Figure 2. It was used for improving facilities too as discussed in Section 4.2 and Figure 3. Next, general function decomposition trees were used for design review (Section 4.3) and patent application (Section 4.4 and Figure 4). Such function decomposition trees could be shared with different types of engineers and in different tasks as discussed in Section 4.5. Lastly, specific ways of function achievement in function decomposition trees are generalized and organized in *is-a* hierarchies. Such a kind of knowledge base can be used for exploring ways in conceptual design as discussed in Section 4.6.

In April, 2003, a users group for SOFAST software has been established. Current members are 13 companies, in which its test use got started.

4.1 Understanding and Sharing DRs

Among 103 function decomposition trees, Figure 2 shows a function decomposition tree of a production machine called a wire-saw. It is designed to slice semiconductor ingots by using moving wires. In Figure 2, the top function is "to split" rather than "to slice" which implies how to split. The splitting is achieved by two sub-functions; to lose combination force of a part (the kerf loss, i.e., the part lost by cutting) and to move the part away. This way of achievement is conceptualized as the removing way based on separation of the kerf loss part. The sub-functions are further decomposed into sub-functions.

Figure 2 includes possible unintended behaviors of the device(phenomena) and supplementary functions for avoiding them. For example, the cooling function for the moving wire is designed to avoid snapping of the wire due to heat caused by friction. The relation between the snapping of the wire as a possible unintended phenomena (or trouble) and the frictional heat as its cause is explicitly described. In other words, our framework provides knowledge media for such designer's intention.

Such a function decomposition tree shows designer's intention of how to achieve the goal function, which is included neither structural model nor behavioral model. This effect is basically the same as in the conventional function decomposition tree (Pahl and Beitz 1988). Main distinctive features of our framework include (1) the concept of "way of function

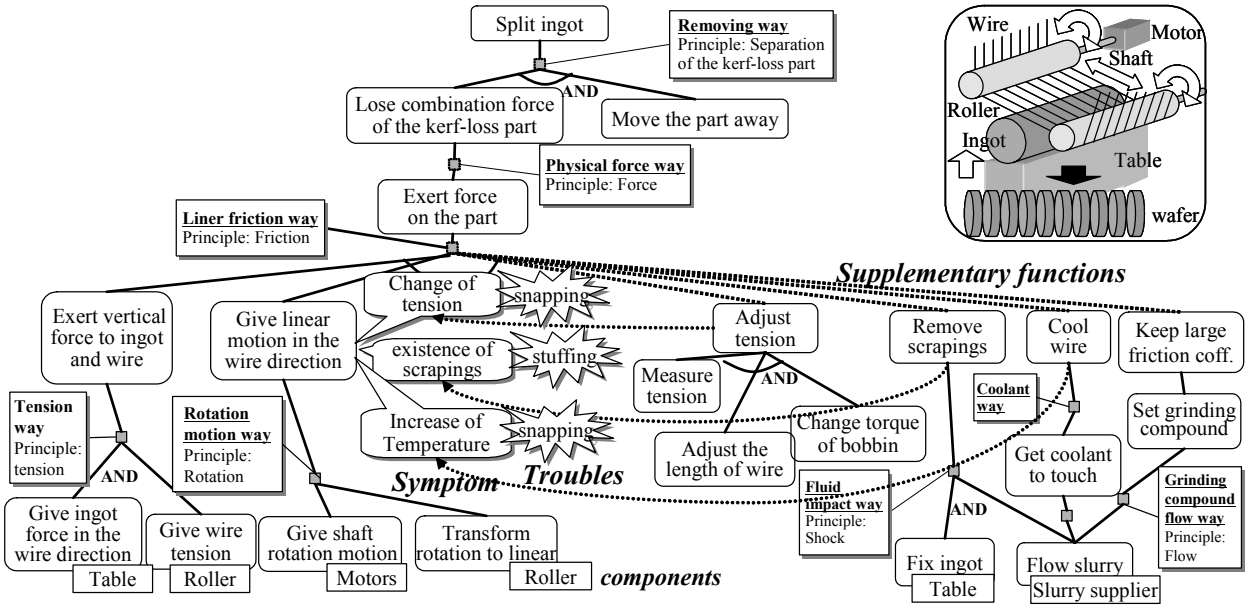

Figure 2. A function decomposition tree of a wire-saw for slicing ingots (portion).

achievement" and its relationship with functions, (2) the extended device ontology, and (3) integration of unintended behaviors explained as follows. Firstly, the concept of "way of function achievement" and definitions of relationships with functions discussed in Section 3 help knowledge authors keep functions representing "what to achieve". For example, as mentioned in Section 2, the pseudo function "to weld" can be decomposed into the "joining function" and "fusion way". To use the way of function achievement is not mandatory, since it can be left anonymous when there is no necessity to conceptualize it. Therefore, its introduction is nothing restrictive in building a function decomposition tree.

Secondly, the extended device ontology provides concepts for assigning "roles" for each object in the target world. In Figure 2, the wire can be considered as an *agent* (to exert force on ingots), an *operand* (to be moved by the roller) or a *conduit* (to transmit tension). According to semantic constraints in the extended device ontology, a possible consistent role assignment is to decompose the wire into two parts, a working wire as an agent and a transmitting wire as both medium (a sub-concept of the operand) and conduit. One of extension of the device ontology is to accept the last situation.

Lastly, integration of unintended behaviors enables us to understand the intension of the supplementary functions which often give important information. For example, another designer can understand that the reason of the existence of a supplementary function "to cool wire" in Figure 2 is to avoid snapping by removing its cause, i.e., frictional heat.

The experiential evaluation about this aspect by the SEI engineers was unanimously positive. They said that this framework enabled them to explicate the implicit knowledge possessed by each designer and to share it among team members. It was easy for designers to become familiar with the framework based on the device ontology. They said that the explication of their own implicit design knowledge helps them reflect the design and/or the target devices. This benefit gives them a strong motivation for describing the functional models.

4.2 Equipment Improvement

This section reports on a real example of making implicit knowledge about functionality of a manufacturing equipment explicit and the improvement enabled by the explicated knowledge. The target manufacturing equipment is a policing machine for wafers of semiconductor. As shown in Figure 3, the rotating disk with a weight polishes the wafer on the table with the slurry (fluid) including diamond powder as grinding powder. The rotating disk moves freely inside of an outer ring called the guide ring. The goal of improvement was 63% reduction of necessary time for a wafer. To do this, initially, an engineer tried to adjust values of working parameters of the machine such as rotating speed of the disk, the weight and amount of the slurry. After four months investigation, however, its result was still not enough for achieving the goal.

Then, in order to find another (unknown) working parameter, he described a function decomposition tree of the machine as shown in Figure 3. As the result, he became aware of additional (and implicit) function of the guide ring, that is, to put diamond powder in the slurry into grooves of the table. Thus, he changed the width of the guide ring for putting more diamonds into gaps between the wafer and the table. Eventually, the reduction of the necessary time became 76%, which is better than the initial goal. This improvement has been done within three weeks.

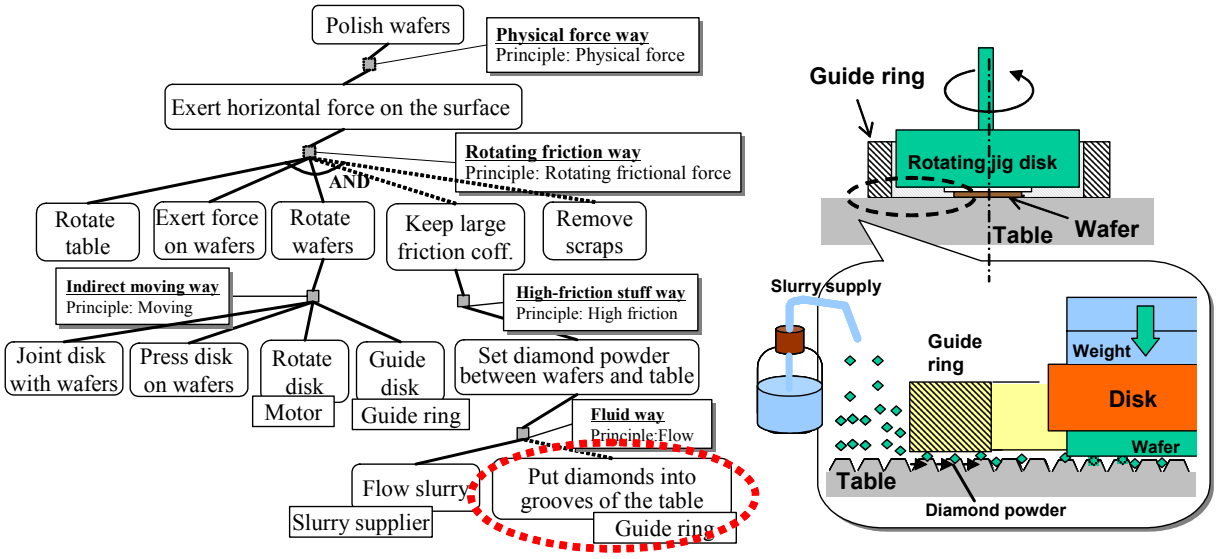

Figure 3. Function decomposition tree of a polisher for finding parameters

4.3 Design Review

The design review is a team activity to review an original design in order to doublecheck it and to explore possible alternative designs. For explaining an original design, engineers of SEI have been using a comparative table with a text which describes some alternative designs in columns and their features in rows.

The production systems division of SEI adopted the general function decomposition tree as a regular schema of documents for design review. It can show alternative ways of function achievement for each (sub) function, their features in comparison, and reasons of adoption of a specific way and of unadopted ones in one figure. In the comparative table, it is difficult to describe all alternatives exhaustively for each function. Thus, it has needed two times of redo of design review in average. After adoption of our framework, however, the times of the design reviews has been reduced to one third.

Especially, description of unintended behaviors and supplementary functions plays a crucial role in reliability design. It shows the original design takes into account what phenomena possibly happen and how to avoid them by the additional supplementary functions as shown in Figure 2..

4.4 Patent Map and Patent Application

Another usage of the general function decomposition tree is as a kind of "patent map" of applicable ways of a function. For example, Figure 4 shows a general function decomposition tree including patents for handling wafers. The italic letters are initials of names of companies which adopt each way for manufacturing wafers. The differences of working principles and features of patents are organized in each level of function decomposition. It includes possible unintended(undesirable) phenomena and troubles such as chipping of the wafers for each way. For patent application, difference (originality) of a new patent from conventional patents can be explicitly described as new ways of function achievement and/or new features of the ways.

In communication between engineers and patent attorneys for application of a new patent, it is difficult to clarify its originality and to make claims properly. In SEI, it had taken hard effort and a long period, in average, three or four weeks. When a patent application was done using the general function decomposition tree shown in Figure 4 (plus new ways of function achievement which form the new patent), the period of the patent application was reduced to just one week (i.e., one third of the usual period). Moreover, the patent claims in the patent application were increased in some case double, since using the general function decomposition tree the patent attorney found new points of difference with other patents by checking differences at each level of function decomposition.

4.5 Sharing with Different Types of Engineers and Sharing in Different Tasks

As well as communication between designers and patent attorneys, our framework was used as knowledge media for collaborative work by designers, manufacturing engineers, manufacturing equipment engineers, equipment operators and equipment maintainers. Although mutual understanding and collaboration among them has been strongly required, it rarely happened because they have own viewpoints and used different knowledge representation such as design drawing, QFD, FMEA sheets, FTA diagrams, etc. Its problem was that each representation uses own vocabulary and lack interoperability with others. The use of our framework, however, facilitated their mutual understanding and collaboration in an equipment improvement project. In the project, it turned out that the framework worked as a common vocabulary which lacked before.

Another kind of interoperability of knowledge in our framework is among different tasks (activities of

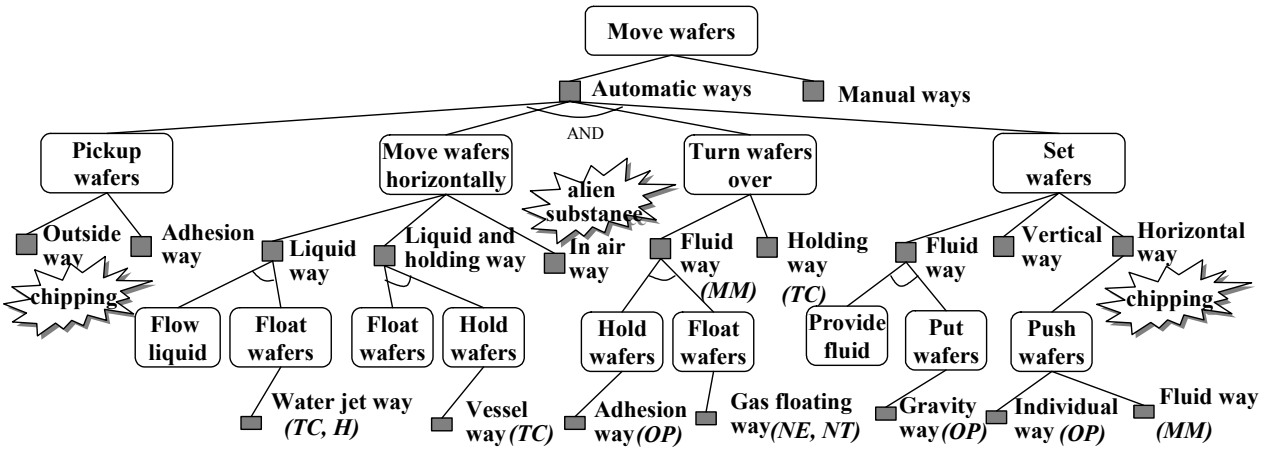

Figure 4. A patent survey for handing semiconductor wafers

engineers) such as design, solving quality problems and patent application. For example, for a sub-system for adjusting angle of cutting ingots in the wire saw shown in Figure 2, a function decomposition tree described for design review (as discussed in Section 4.3) was used for diagnosis of a problem in the equipment deployed in a manufacturing line and for improving the equipment. In conventional work style, an FTA diagram might have been newly described for the problem. Moreover, a new patent of the adjusting sub-system was applied using the same function decomposition tree plus information of related existing patents. It means that the same knowledge representation was reused in different tasks, that is, design, diagnosis, improvement and patent application, which was impossible before in SEI.

4.6 Expanding Alternative Ways of Function Achievement using Generic Ways

The specific ways of function achievement are generalized into generic ways and then organized in *is-a* hierarchies. Figure 5 shows *is-a* hierarchies of function achievement ways for split functions etc. They are generalized from specific ways used in the wire-saw shown in Figure 2 and in other cutting machines such as the water-jet cutting and the electrolysis cutting. In this organization, difference between the wire-saw and other cutting machines are explicitly represented. The wire-saw uses three ways, that is, the removing way for splitting, the physical force way for loosing combination force, and the liner fiction way for exerting force. Moreover, the ways for exerting force can be also used for other machines, e.g., washing machines. For example, in the screw-type washing machine, dirt is separated from cloth by the random friction force which is caused by a rotating screw. It suggests that these pieces of knowledge are general and applicable to different domains. Conventional organization of ways of cutting which is found in a text-book in the field mainly shows not principles but "what is used for" such as wire and blade. "The wire-saw way" found in a text book is not single way of function achievement but a composite way of the three primitive ways.

Although such an organization of ways of function achievement knowledge has not been fully deployed, a feasible new improvement of the wire-saw was found from the knowledge-base, that is, we manually found a way of using magnetic fluid which is used in the textile industry for controlling tension of the wire following thus organized knowledge. This small invention could have been done by a computer system which has a way knowledge server. This indicates the utility of our framework for general functional knowledge.

In the deployment, techniques of lighting for inspection also were systematized. Consulting the systematized generic ways in the knowledge-base, a novice engineer developed an inspection machine in three days. Such development usually needs two weeks by experts.

5. SOFAST® software

The authors have developed a knowledge management software named SOFAST® (SOFAST is abbreviation of Sumitomo Osaka-university Function Analysis and Systematization Tool. SOFAST is a registered trademark of SEI.). SOFAST is designed to support description of functional knowledge and sharing them in the intra-network. It consists of a client software and an SQL server as shown Figure 6(a). Using the client software, a user can describe a functional decomposition tree in graphical user-interface. Figure 6(b) shows its screen snapshot, in which the both main panes show a general function decomposition tree in different display styles in Japanese. As shown in the right pane, users can attach related documents including bitmap images, graphs and spreadsheets to the tree. A small window in the middle shows an overview of the whole tree for scrolling.

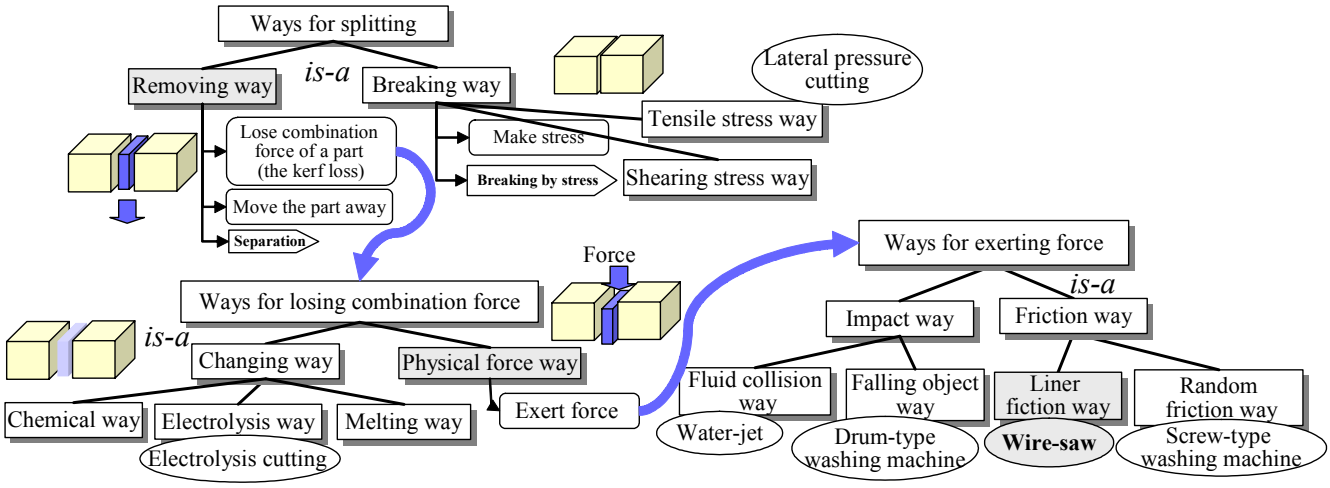

Figure 5. An example of organizing generic ways in *is-a* hierarchies.

The described models are stored in the SQL server and can be accessed by users using the client software (or using any SQL-support software) from other hosts. Because the server stores each way of function achievement at each level separately, by specifying a goal function, a user can retrieve many ways of function achievement used in different facilities or products in order to achieve the specified function. For example, 56 instances of ways of function achievement in many facilities (including different applications of the same generic way) are found from the current SOFAST database for a specific function.

Since April, 2003, we have provided SOFAST with other 13 companies in the SOFAST users group. The authors regularly hold meetings with the member-companies for lectures, training, report of their use, and discussion on further improvement.

In the current implementation of SOFAST, as discussed in Section 6.2 in detail, management capability of the *is-a* hierarchies of generic ways of function achievement is missing. The build-in vocabulary of functional concepts is not based on the functional concept ontology. The improvement of these points is being planed.

6. Discussion

6.1 Success Factors of the Deployment

The successful deployment discussed thus far is a kind of knowledge management activity. In general, difficulties of knowledge management activity include;

- Difficulty in explicating implicit knowledge,
- Difficulty in retrieve useful knowledge, and
- Lack of motivation for writing own knowledge.

Firstly, it is difficult to explicate own implicit knowledge. Functional knowledge is intrinsically not objective but subjective. Without a guideline, novice modelers would be puzzled to describe functional models. The functional ontologies give conceptual rules or guidelines for capturing the target world, i.e., conceptual and subjective functions. Especially, the extended device ontology gives users hints for interpreting how a device works consistently as discussed in Section 4.1. The concept of "way of function achievement" also helps modelers solve confusion between "what to achieve" and "how to achieve it". Clear distinction between a general-specific hierarchy (the *is-a* relations) and a whole-part hierarchy (*is-achieved-by* relations) helps knowledge authors 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 often.

Next, although difficulty in retrieval is sometimes treated as a technical issue of information search, we believe that the retrieval problem of functional knowledge is due to dependence of contents on "how to achieve functions" and "how to use information". When functional terms used are composite of how to achieve and what to achieve like before, the terms are very domain- and equipment-dependent and hence the generality is low. This has caused the low retrievability of knowledge. This issue can be avoided by the concept of "way of function achievement" as discussed above because the detachment of it from functional concepts makes the functional terms very general.

"How to use information" is also important in the knowledge management context. The knowledge found has to be ready for use in the task at hand. To do

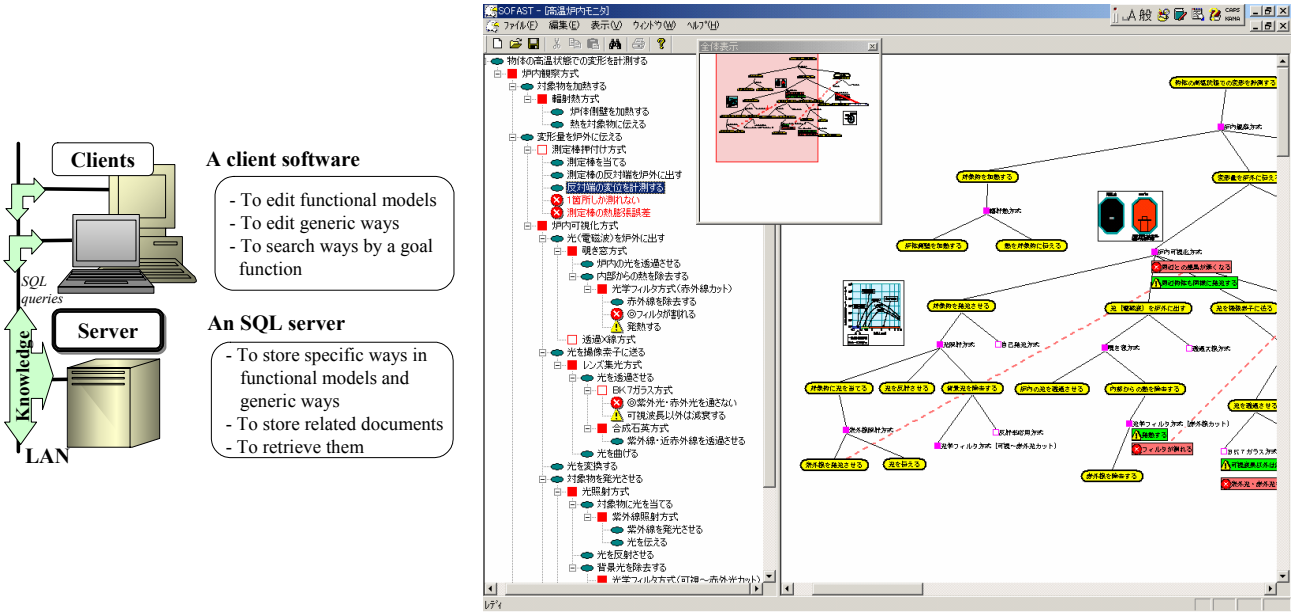

Figure 6. A knowledge management software SOFAST.
(a) Its architecture (left) and (b)Screen snapshot of the client software (right)

this, well-prepared knowledge representation needs to be available to which the knowledge should fit. This issue is partially resolved by our functional ontology framework which is an integration of knowledge about functional structure representing intended behaviors and unintended behaviors at an appropriate level of abstraction, that is, functional level. This enables the same model(representation) to be used in different tasks as discussed in Section 4.5. Such applicability in multiple tasks reduces effort of knowledge authors. Furthermore, it augments interoperability among the task-dependent knowledge representation via our framework.

Lastly, in general, knowledge authors have no effective motivation to write their own knowledge and share it with others. In the deployment, however, engineers say that they themselves can get benefit from writing functional models of their own equipment, since it gives them a chance of reflection and good stimuli which lead them to in-depth understanding of the equipment. This has been enabled by our modeling framework as a knowledge medium which externalizes the engineers' understanding which had been left implicit in an appropriate abstraction level with consistent guidance. Its real example was shown in Section 4.2. In general, the micro-macro hierarchy of the function decomposition tree enables the designer to explore the possible alternatives (for conceptual design) and/or causes of the problem (for problem solving) for each function systematically. For example, the fault tree analysis (FTA) for problem solving tends to be difficult to enumerate all possible causes without clear understanding of function structures.

6.2 Limitation

The main point of our framework is that it adopts the ontological approach for controlling contents of functional models. It, however, has disadvantage including less freedom of functional representation than ad hoc functional modeling and needs of training for writing functional models compliant with functional ontologies. The former issue became a problem especially in selecting a functional concept for a component. There could be domain-specific vocabulary and different terms for the same concept. We cannot claim completeness of the concepts in our functional concept ontology in its nature and understand there could be necessity to extend our ontology. We are currently investigating on two-level terms for easier use: functional concepts and domain-specific vocabulary which allows knowledge authors to use rather free terms which we prepare beforehand.

The latter issue is, in other words, it is difficult to enforce ontological commitments on knowledge authors. The authors are currently establishing stepwise guidelines for describing functional knowledge for easier commitments to the ontologies (Kitamura and Mizoguchi 2003b). Moreover, automatic checking of violation in the functional models against the ontologies is under investigation.

As the results of these limitations, in fact, a part of functional models described thus far follows the functional ontologies not completely in the deployment. Current vocabulary used in SOFAST is not fully based on the functional concept ontology. The current main usage of SOFAST is to describe functional decomposition trees of each production facilities and general function decomposition tree for similar facilities. Sharing of ways of function achievement in SOFAST relies not on the *is-a* hierarchy of generic ways but on searching specific ways of function achievement by specifying a goal function. Nevertheless, discrimination of *is-a* relations among ways of function achievement from *is-a* relations among functions have helped engineers a lot avoid confusion. Such advanced issues of our framework are to be deployed by implementing new functionalities for supporting them in SOFAST together with advanced training.

Besides production systems and facilities as discussed in this paper, the ontologies have been applied to modelling of a power plant (Kitamura, et al. 2002), 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.

7. Related work

7.1 Engineering Ontologies

Among types of ontologies, we concentrate on not the task ontology of design activities (such as (Chandrasekaran 1990)) but the domain ontology of artifacts to be designed. There is a lot of work on ontologies on engineering domain (de Kleer and Brown 1984, Forbus 1984, Lui 1992, Gruber, et al. 1992, Cutkosky 1993, Gruber and Olsen 1994, Borst et al. 1997, Kumar and Upadhyaya 1998, Salustri 1998, Horváth et al. 1998, Sekiya, Tsumaya and Tomiyama 1999, Chandrasekaran and Josephson 2000). One of remarkable work was done by Borst, et al. (1997), in which the PhysSys ontology was proposed as a sophisticated lattice of ontologies for engineering domain. However, the device ontology in PhysSys is weak as the same manner as the other conventional ones in that it does not have enough

concepts for understanding the *ontological roles* all the participants play. We extended such conventional one by redefining "conduit" and "medium" as an extended device ontology. Our refinement of the device ontology gives the modeller a more detailed guideline to capturing the target devices.

Moreover, PhysSys ontology includes no ontology for function from the teleological viewpoint. Chandrasekaran and Josephson (2000) clarify meanings of the concept "function" based on ontological consideration and propose 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. Other classifications of functions are found in (Chittaro and Kumar 1998, Hubka and Eder 2001, Deng 2002) as well. Other ontological consideration on functionality is found in (Kumar and Upadhyaya 1998, Salustri 1998).

Recently, in the emerging Semantic Web, ontologies play a crucial role in giving interoperability to web resources (e.g., Davies, et al. 2003). The knowledge of unintended behaviors discussed in this paper is can be regarded as different knowledge resource based on an ontology different from the functional ontology. The authors are currently investigating design knowledge transformation based on such multiple ontologies in the semantic web.

7.2 Functional representation

There are quite amount of research on functional representation (de Kleer 1984, Sembugamoorthy and Chandrasekaran 1986, Gero 1990, Keuneke 1991, Vescovi, et al. 1993, Chittaro et al. 1993, Lind 1994; Umeda et al. 1996, Bhatta and Goel 1997, Malmqvist 1997, Chandrasekaran and Josephson 2000, Bracewell, et al. 2001, Deng 2002, Rajan, et al., 2003). We focus not on purpose function but on technical functions in the terminology in (Hubka and Eder 1998, 2001).

The main point of our research is to clarify several relationships related to functionality, that is, the *is-a* hierarchy of functions, the *is-achieved-by* (*part-of*) relations among functions, and the *is-a* hierarchy of ways of function achievement. We detach a part of the conditions for specialization in (Hubka and Eder 1998, 2001) (see discussion in Section 2) and thus describe them as specific attributes of a way of function achievement in *is-a* hierarchy of the way of function achievement (Kitamura and Mizoguchi 2003a,b).

Although capturing of feature of function decomposition is also found as "means" in (Malmqvist 1997), it is not generic knowledge but a model specific to a product. In (Bracewell and Wallace 2001), generic knowledge of single functional decomposition called "means" (without explicit organization among them) is discussed. We define *is-a* relations among the conceptualized generic ways of function achievement, and investigate how to organize them.

The design prototypes proposed by Gero (1990) include structural decomposition as well as function decomposition. In the FBS modeling framework (Umeda, et al. 1997), a function prototype includes the physical feathers of behavior 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 behavior.

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.

We define functional concepts using operational information called functional toppings (FTs) such as focus on operands and necessity of operands, and then enable us to define intention-rich functional concepts (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. The recent effort for a standard taxonomy of engineering functions in NIST Design Repository Project (Hirtz, et al. 2002) is well established, however, lacks operational relationship with behaviors.

De Kleer defines function as a causal pattern between variables (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 as FTs for representing intention. Keuneke (1981) defines types of functions such as ToMake. Our FTs include them. Although we adopt neither the process ontology (Forbus 1984) nor the bond graph theory (Rosenberg and Karnopp 1983), the functional concept ontology includes similar functions in the flow-based functional modeling approaches (Chittaro et al. 1993, Lind 1994).

The teleological interpretation specified by FTs is similar to "means and ends" in (Lind 1994), F-B relationship in (Umeda, et al. 1996), and "aims-means" in (Hubka and Eder 1998). The last axis includes design requirements as well. 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".

In IDEAL in (Bhatta and Goel 1997), generic teleological mechanisms (GTM) are used (modified) for design different context based on analogy. In our approach based on the limited set of functional concepts, designers can explore explicit *is-a* hierarchies of the ways of function achievement.

The TRIZ (TIPS) theory gives some patterns (or strategies) of inventions based on contradiction between two physical quantities (Sushkov et al. 1996). We concentrate not on design strategies but on a modelling schema. The TRIZ theory also pays attention on physical principles (effects), though we make a clear relationship between physical principle and the functional structures.

7.3 Unintended behaviors

Information of unintended behaviors in our models is found in other formalism such as FMEA sheets and fault trees in FTA. One of the benefits of our framework is tight integration such information with functional structures. Such integration helps designers explore possible causes systematically for each function at each grain-size.

In order to capture a larger set of failure modes systematically, there is research on advanced FMEA (e.g. Steven et al., 1999, Hata et al., 2000) using behavioral models to simulate device behaviors. Model-based diagnosis community uses sophisticated qualitative reasoning to identify faulty components (e.g. De Kleer and Williams, 1987). The diagnosis using hierarchical functional models instead of behavior models are proposed in (Chittaro, et al., 1993, Larsson, 1996). These approaches based on deviation from "intended" behavioral models, however, cannot deal with a part of causative chains of faults as pointed out in (Davis 1984, Böttcher 1995, Kitamura and Mizoguchi 1999).

8. Concluding remarks

The successful deployment of an ontological modeling schema for functional knowledge has been reported. After discussion on the current problems in industry expected roles of ontologies, six types of real usages in the deployment, effects of ontologies, their success factors were discussed. One of the main success factors is to clarify three types of relationships related to functionality (i.e., the *is-a* relation of functions, *is-achieved-by (part-of)* relations among functions, and the *is-a* relation of ways of function achievement) and to provide four types schemata of functional knowledge. Although engineers mainly use (general) function decomposition trees in the deployment, such discrimination helps engineers reflect their own knowledge.

As discussed in Section 6.2, support for deeper commitment on functional ontologies is under investigation. The authors are planning to develop support functionality in SOFAST and to deploy it in daily work. Other 13 companies in the SOFAST users group will help us improve the software.

The modeling schema shown in this article for unintended behaviors is a simplified version. We are currently investigating more detailed ontological schema aiming at explicit representation of design rationales of supplementary functions (Koji, et al. 2003). In our collaborative work with Delft University of Technology, we are extending framework for including user actions as well (van der Vegte, et al. 2002). The interoperability among different tasks is limited in a sense that the same representation is used for all tasks. Dynamic transformation of representation of such models is under investigation.

The authors believe that Ontology Engineering contributes to knowledge systematization of domain knowledge by providing "theory of contents" of knowledge, that is, how to capture knowledge and to organize it applicable in other contexts (Mizoguchi and Kitamura 2000). An ontology provides fundamental guidelines for capturing the target world and for describing it in computer systems. This research can be regarded as a successful example of this research direction.

Acknowledgements

The authors would like to thank Yusuke Koji, Mariko Yoshikawa, Tomonobu Takahashi, Masaru Takahashi, and Kouji Kozaki for their contributions to this work. Special thanks go to Shuji Shinoki and engineers in plant and production systems engineering division, Sumitomo Electric Industries, Ltd. for their cooperation of the deployment.

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*, 294-300.

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

Böttcher, C., 1995, "No fault in sructure? - how to diagnose hidden interactions", *Proceedings of IJCAI-95*, pp.1728-1733.

Bracewell, R. H., Wallace, K. M., 2001, Designing a Representation to Support Function-Means based Synthesis of Mechanical Design Solutions, In *Proc of ICED 01*.

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*, 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*, 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.

Chittaro, L. and Kumar, A. N., 1998, Reasoning about Function and its Applications to Engineering, *Artificial Intelligence in Engineering*, 12, 331-336.

Cutkosky, M.R., et al., 1993, PACT: An Experiment in Integrating Concurrent Engineering Systems, *Computer*, January, 28-37.

Davies, J., Fensel, D., van Harmelen, F. (eds), 2003, *Towards the Semantic Web - Ontology-driven Knowledge Management* (John Wiley & Sons)

Davis, R., 1984, Diagnostic reasoning based on structure and behavior, *Artificial Intelligence*, 24, 347-410.

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.

De Kleer, J., and Williams, B.C., Diagnosing Multiple Faults, *Artificial Intelligence*, 32, 97-130

Deng, Y. M., 2002, Function and Behavior representation in conceptual mechanical design, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 16, 343-362.

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

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*, 89-104.

Gruber, T. R., Tenenbaum, J. M., Weber, J. C., 1992, Toward a Knowledge Medium for Collaborative Product Development, In *Proc. of Artificial Intelligence in Design '92*, 413-432.

Gruber, T. R., 1993, A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220.

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

Hata, T., Kobayashi, N., Kimura, F., and Suzuki, H., 2000, Representation of Functional Relations among Parts and Its Application to Product Failure Reasoning, *Proceedings of CIRP International Seminar on Design.*

Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., Wood, K. L., 2002, A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts, *Research in Engineering Design*, 13, 65-82.

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*.

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

Kitamura, Y., and Mizoguchi, R., 1999, An Ontological Analysis of Fault Process and Category of Faults, *Proceedings of Tenth International Workshop on Principles of Diagnosis (DX-99)*, 118-128.

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

Kitamura, Y., and Mizoguchi, R., 2003b, Organizing Knowledge about Functional Decomposition, *Proc. of the 14th International Conference on Engineering Design (ICED 03).*

Kitamura, Y., and Mizoguchi, R., 2003c, Ontology-based systematization of functional knowledge, *Journal of Engineering Design* (Taylor & Francis) to appear.

Koji, Y., Kitamura, Y., Mizoguchi, R., 2003, Towards modeling design rational of supplementary functions in conceptual design, *Technical Report AI-TR-2003-1*, I.S.I.R., Osaka University.

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),* 213-218.

Kumar, A. N. and Upadhyaya, S. J., 1998, Component-Ontological Representation of Function for Reasoning about Devices, *Artificial Intelligence in Engineering*, 12, 399-415.

Larsson, J. E., 1996, Diagnosis based on Explicit Means-ends Models, *Artificial intelligence*, 80, 29-93.

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 (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* (McGraw-hill)

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, (Springer-Verlag)

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

Pahl, G., and Beitz, W., 1988, *Engineering design - a systematic approach* (The Design Council)

Rajan, J. R., Stone, R. B., Wood, K. L., 2003, Functional Modeling of Control Systems, In *Proc. of ICED 03.*

Rosenberg R. C. and Karnopp, D. C., 1983, Introduction to Physical System Dynamics. (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*, 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 (eds.), *Knowledge Intensive Computer Aided Design*, 73-90, (Kluwer Academic Publishers, Boston)

Sembugamoorthy, V. and Chandrasekaran, B., 1986, Functional representation of devices and compilation of diagnostic problem-solving systems, In *Experience, memory and Reasoning,* 47-73.

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

Steven, K., Peder, F., and Ishii, K., 1999, Advanced Failure Modes and Effects Analysis of Complex Processes, *Proceedings of the ASME Design for Manufacturing Conference.*

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, Soc. of Japanese Value Engineering (In Japanese).

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.

Umeda, Y., and Tomiyama, T., Functional Reasoning in Design, *IEEE Exert*, March/April, 42-48, 1997.

van der Vegte W.F, Kitamura, Y., Mizoguchi, R., and Horváth, I., 2002, Ontology-Based Modeling of Product Functionality and Use - Part 2: Considering Use and Unintended Behavior, *Proceedings of The Third International Seminar and Workshop Engineering Design in Integrated Product Development (EDIProD 2002)*, 115-124.

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*, 626-633.

Wilhelms, S, 2003, A Conceptual Design Support System using Principle Solution Elements, *Proc. of ICED 03.*