INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 05 MELBOURNE, AUGUST 15-18, 2005

ONTOLOGY-BASED TRANSFORMATION FROM AN EXTENDED FUNCTIONAL MODEL TO FMEA

Yusuke Koji, Yoshinobu Kitamura and Riichiro Mizoguchi

Keywords: design knowledge management, failure knowledge, ontology, functional representation

1 Introduction

Conceptual design knowledge concerning with functions of products and phenomena unintended by designers is scattered around various forms such as the functional model and the FMEA sheet, which are dependent on engineering tasks. Aiming to promote sharing of such knowledge among designers, this paper proposes a knowledge transformation system from extended functional models to FMEA sheets based on ontology engineering. The extended functional model includes both knowledge about functional structures and possible faulty behaviors. In order to transform the models, the system uses two ontologies for the two knowledge models, each of which defines fundamental concepts in each model to capture the target world from its own viewpoint. The system transforms the knowledge models using ontology mapping knowledge, which specifies the correspondences (mapping) between concepts in the ontologies. This paper discusses the contents of the ontologies, the ontology mapping knowledge, and the trasformation system.

2 Background and Approach

In conceptual design, it is important to develop functional structures which reflect the rich experience in the knowledge from previous design failures. Especially, if a designer learns possible abnormal behaviors from a previous design failure, he or she can add an additional function which prevents the abnormal behaviors and faults. To do this, it is a crucial issue to share such knowledge about possible faulty phenomena and how to avoid them. In fact, a part of such knowledge is described in FMEA (Failure Mode and Effect Analysis) sheets, function structure models for systematic design [1] and fault trees for FTA (Fault Tree Analysis).

The problems on which we focus in this paper here are that such knowledge scatters around several knowledge forms such as functional structures and FMEA sheets, and that interoperability among them is low. For example, although an FMEA sheet includes knowledge about both the possible faulty behaviors and the function preventing them, sharing (interoperation) of knowledge between FMEA and the functional structures is rarely realized in practice. Therefore, designers must describe similar knowledge repeatedly in different forms, and labor to update the contents. Moreover, it is not easy to find out how to avoid faults efficiently from a lot of documents in many forms. One of the deep reasons of the low interoperability is that each model such as a functional model, an FMEA sheet and a fault tree depends on the engineering task (problem solving) such as functional design, reliability analysis and trouble-shooting, respectively.

In order to promote knowledge sharing about faults and function, we propose an ontology-based knowledge transformation on the basis of ontological engineering in artificial intelligence. An ontology in this system defines fundamental concepts in a specific knowledge form to capture the target world from its own viewpoint. The trasnformation system uses an ontology of a source model, an ontology of a distitaition model and ontology mapping knowledge, which bridges the gap between them. This paper proposes a system to transform an extended functional knowledge model into FMEA sheets as a task-dependent knowledge model. The extended functional model reported in our previous paper [2] includes both knowledge about functional structures and possible faulty behaviors independently of engineering tasks such as conceptual design and reliability analysis. We have developed the extended functional ontology, the FMEA ontology and an ontology mapping knowledge between concepts in these ontologies.

These ontologies are based on our previous research efforts. We have established a knowledge modeling framework for functional models based on a functional ontology, which consists of an ontology about device and function and an ontology about functional concepts [3]. It has been successfully deployed in a manufacturing company [4]. Then, it has been extended to the extended functional model [2], which includes faulty behaviors as well. This paper discusses the extended functional ontology, which is an extension of the functional ontology in [3].

The rest of this paper is structured as follows. The next section overviews the transformation system. Then, the extended functional ontology and the FMEA ontology are discussed in Section 4 and 5, respectively. Section 6 shows an ontology mapping knowledge between these ontologies. Current implementation of the system is mentioned in Section 7. Section 8 discusses the benefits of our framework and the transformation system. Then, related work is discussed followed by some concluding remarks.

3 Overview of a knowledge transformation based on ontologies

Figure 1 shows an overview of our knowledge transformation system. When this system is given an extended functional model as input, the system firstly extracts FMEA-related knowledge from the given model and then transforms such knowledge into FMEA sheets as output. This system consists of the following four components.

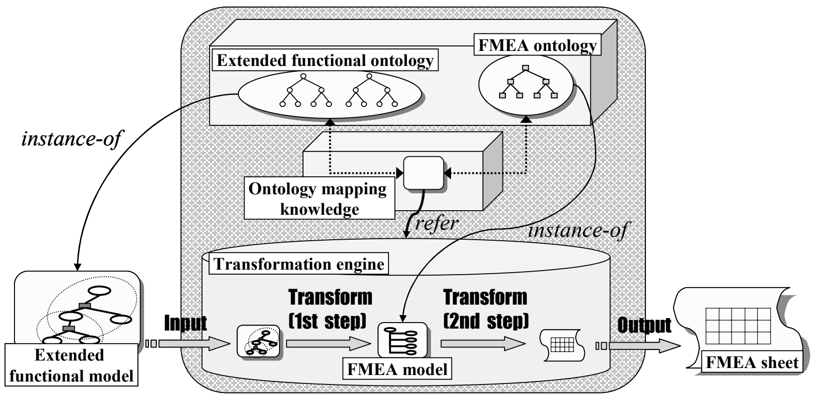

Figure 1. An overview of knowledge transformation system

1. Extended functional ontology: The ontology defines concepts in the extended functional models [2]. It includes “device”, “function”, “undesirable phenomena” and “malfunction”. A knowledge author describes the extended functional model based on the ontology. This ontology provides a knowledge author with constraints, guideline and vocabulary for description of the models. The ontology defines classes of concepts and then the model consists of instances of the classes defined in the ontology.

2. FMEA ontology: An ontology defines concepts in FMEA sheets. In this framework, this ontology specifies the target model of transformation from an extended functional model.

3. Ontology Mapping Knowledge: An ontology mapping knowledge specifies correspondence between similar concepts in the ontologies in components 1 and 2. For example, the “failure mode” concept in the FMEA ontology corresponds to the concept of “change of a device which causes an undesirable state of a function” in the extended functional ontology. A mapping knowledge shown in this paper is valid for (ideally) all FMEA and functional models. Given such mapping knowledge beforehand, our system can perform model transformations. This component is a main part of this paper.

4. Transformation Engine: Based on the ontology mapping knowledge, this transformation engine transforms a part of knowledge described in an extended functional model into an FMEA model as a task-dependent model. The transformation process consists of the following two steps: As the first step, an extended functional model is transformed into an FMEA model. Then, as the second step, the representation of the FMEA model is formed in a suitable form for practice (for example, the FMEA sheet form). Such two-step transformation enables us to separate how to transform contents of models from how to represent them.

4 Extended functional ontology

Generally, functional models deal with only functions, which are phenomena intended by a designer. However, when a designer redesigns an artifact for improvement of quality of the design such as stability, precision and reliability, he or she often adds a function which prevents possible undesirable states, that is, phenomena unintended by a designer (we call such function “supplementary function”). In order to represent design rationale of a supplementary function, it is necessary to deal with unintended phenomena, that is, to represent a process to cause undesirable state prevented by the supplementary function. Our extended functional model can represent a part of such heterogeneous knowledge in a unified framework.

An ontology which specifies descriptions of such extended functional models is called “extended functional ontology”. This ontology is extended version of the functional ontology reported previously [3], [5]. In this section, an overview of the extended functional ontology is presented using an example of an extended functional model.

Figure 2 shows a part of an extended functional model of a wire-saw. The wire-saw is a manufacturing device to split (cut) an ingot using a moving wire. This model represents a functional structure of the wire-saw and a cause-consequence chain to cause an undesirable state which is avoided by “cooling function” which is one of the supplementary functions in the wire-saw. The explanation of these two parts of the model is as follows.

Firstly, the functional structure of the wire-saw is represented as a function decomposition tree shown in the right part of Figure 2. A function decomposition tree shows the whole (macro) function is achieved by a sequence of sub (micro) functions (called “is-achieved-by” relations). In the example of the wire-saw in Figure 2, the splitting function as the whole-function of the wire-saw is decomposed into sub-functions such as “to lose binding force of the kerf-loss part” and “to move the part away”. Such functional structure is similar to the German-style function decomposition in [2], the whole-part relation in [6] and “degree of complexity” in [7].

In addition to such typical function decomposition model, in order to explicate the knowledge used when the designer decomposes a whole function into a sequence of sub functions, the concept of “way of function achievement” have been introduced as conceptualization of background knowledge of functional decomposition such as physical principles and theories. The concept of “way of function achievement“ is similar to a feature of function decomposition which is called “means” in [8]. In Figure 2, a way of function achievement is denoted by a small black square that connects the whole-function and the sub-functions. For example, the way which achieves the “to split ingots” is the “removing way”.

Another part of the extended functional model is represented as *avoided unintended phenomena* (Terms in italics font are defined in ontology.) By the *avoided unintended phenomena*, we mean a series of unintended phenomena which constitute a cause-consequence chain and are avoided by a supplementary function. In the wire-saw, frictional heat is caused by the friction as side-effect of “make a frictional force”. This frictional heat leads to increase of temperature of the wire, and finally the snap of the wire is possibly happen. Such cause-consequence chain to the snap of the wire is conceptualized as *a causative chain*. This causative chain is represented as a *causative chain tree* shown in the left part of Figure 2. The most up-stream phenomenon in a *causative chain* is shown as a leaf node in the causative chain tree and is called *trigger phenomenon*. The most down-stream phenomenon is shown as the top-node of the *causative chain tree* and is called a *causative*

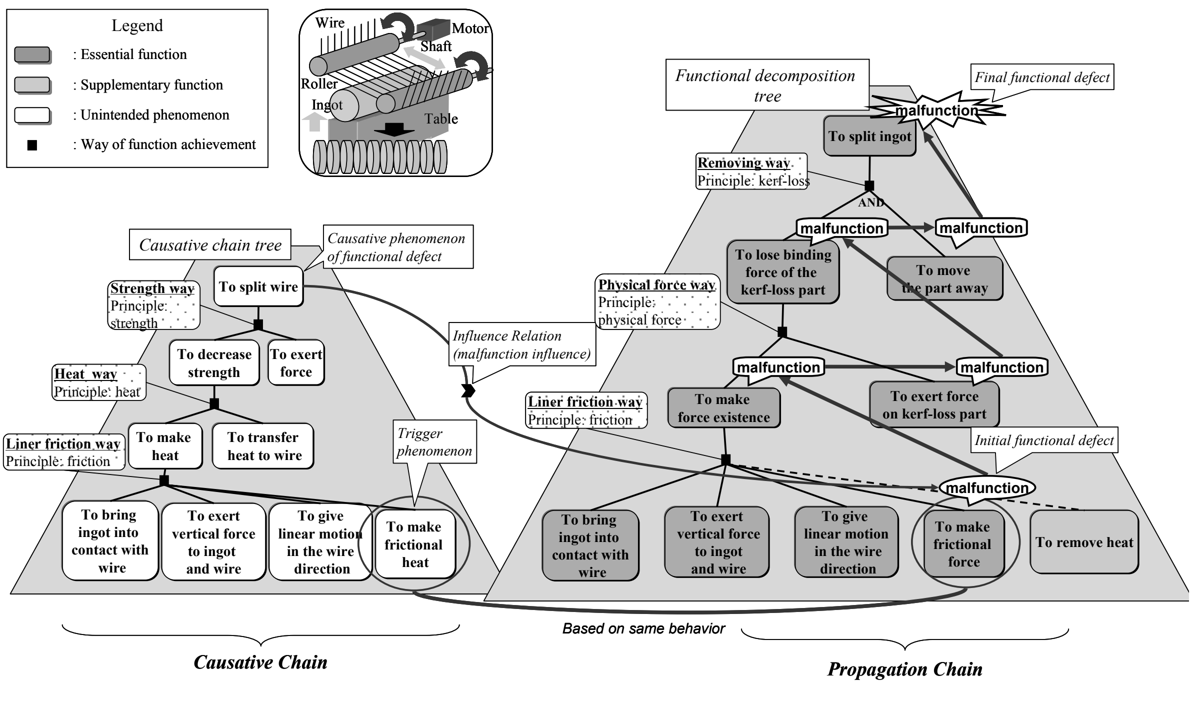

Figure2. An example of the extended functional model, which is the model of the cause-consequence chain avoided by “to remove heat” integrated with the function decomposition tree of the wire-saw

phenomenon of function defect. This causative phenomenon is a direct cause to change a function to an undesirable state. This undesirable state of the function is called *function defect*.

Then, effects of the function defect are propagated to the other functions, and eventually, the whole function, that is, “to split ingots” is changed to malfunction. Such process is called a *propagation chain* and is shown by attaching a balloon to the upper right of each function. In a series of function defects which constitute a propagation chain, the function defect which is located in the most up-stream is called an *initial function defect*. And the function defect which is located in the most down-stream is called a *final function defect*. Thus, a model of *avoided unintended phenomena* consists of a *causative chain* and a *propagation chain*.

In our functional ontology, a function of a device is defined as the teleological interpretation of its “behavior” under the intended goal [9]. The “behavior” of a device is defined as the objective (without designer’s intention) interpretation of its input-output relation as a black box based on a device-centered ontology. On the other hand, unintended phenomena are phenomena which occur in a system and are unintended by designer. In the extended functional ontology, in addition to functions, this unintended phenomenon is defined as a goal-oriented interpretation of its “behavior” under the pseudo (unintended) goal. Since both concepts of a function and an unintended phenomenon are related to an objective behavior (phenomenon), a *causative chain* can be modeled in the same manner of a function decomposition tree, as shown in Figure 2. Moreover, it is possible to use the same verb concepts and ways of achievement for the modeling of both the function decomposition tree and the causative chain tree. For example, we describe “to split wire” as the causative phenomenon in Figure 2, and the same concept of “to split” is used as the whole function “to split ingots” in the function decomposition tree.

Besides these concepts, we conceptualized a relation which represents an influence of an unintended phenomenon on a function to change it to a function defect as *influence relation*. In Figure 2, the link between “to split wire” and “to make frictional force” represents an *influence relation*. This *influence relation* represents that the “to make frictional force” function is changed to malfunction by breaking the agent of the function, that is the wire, and it causes a malfunction of the required function finally. The influence relation can also be used to describe which phenomenon a supplementary function stops to prevent the *avoided unintended phenomena*.

5 FMEA ontology

5.1 Common knowledge between an extended functional model and an FMEA sheet

FMEA (Failure Mode and Effect Analysis) is a technique widely used for prevention and detection of failure troubles in industry. In this paper, we especially focus on functional FMEA which is used in the conceptual design phase.

In functional FMEA, possible undesirable states of functions in a system or an artifact are enumerated as failure modes, and then concepts such as causes of the undesirable states, ultimate effects to the system and additional function are considered. Finally, such concepts are described as an FMEA sheet. In the example of the wire-saw, an FMEA sheet of the wire-saw might include “snap of wire” as a failure mode, “frictional heat” as its cause, and “malfunction of the whole system” as its effect. That is, these concepts such as a failure mode, an effect and a cause represent a part of cause-consequence chain in a system, and hence a part of FMEA sheet is similar to *avoided unintended phenomena* presented in section 3. In other words, an extended functional model of a system includes knowledge about faults included in a FMEA. Such knowledge is a part of what is transformed by our knowledge transformation system.

5.2 FMEA ontology

In this system, the FMEA ontology is used for transformation of the extended functional ontology to an FMEA sheet. Therefore, the FMEA ontology includes only necessary concepts for this purpose. In the FMEA ontology, a concept representing a row of an FMEA sheet is defined as *Row of FMEA sheet*. This *Row of FMEA sheet* consists of *component*, *failure mode*, *cause* and *effect* etc. Besides, sub concepts of these classes are defined. For example, as sub concepts of *effect, malfunction* and *hypofunction* are defined.

6 Ontology mapping knowledge

6.1 *corresponding-to* relation

There are several types of relationship between concepts in different ontologies, such as *super-sub* relation, *whole-part* relation and *corresponding-to* relation. This paper concentrates on the *corresponding-to* relation, since our purpose here is to transform common knowledge described in models based on each ontology. By the *corresponding-to* relation, we mean a relationship between two concepts each of which essentially corresponds to each other even if names or slots of these two concepts are different.

The ontology-mapping knowledge between ontologies consists of such *corresponding-to* relations among concepts in the ontologies. According to the mapping knowledge, our system can transform a model based on an ontology to a model based on another ontology.

6.2 *corresponding-to* relations between the extended functional ontology and the FMEA ontology

In order to identify *corresponding-to* relations in ontologies, it is necessary to adequately understand the ontologies and to choose suitable concepts. In this section, focusing on necessary relations to transform models, *corresponding-to* relations between the extended functional ontology and the FMEA ontology are presented. Firstly, each concept constituting a row of FMEA sheet are investigated, and then each concept in the extended functional ontology which corresponds to the concept in FMEA ontology are identified. Figure 3 shows correspondences of main concepts in the FMEA ontology and the extended functional ontology.

(1) Failure mode: Although there are various definitions of “failure mode” in FMEA, they are summarized as “state of failure” such as “the manner by which a failure is observed” [10]. In FMEA sheets in practice, however, “inefficient“, “vibration“ and “fracture“ are found for example. These correspond to *function defect*, *trigger phenomenon* and *causative phenomenon of function defect* in the extended functional ontology, respectively. The meaning of “failure mode” is thus unstable. In order to transform an extended functional model to an FMEA sheet, an stable ontology alignment is needed. Since many of failure modes found in practical FMEA sheets which we have investigated indicate a change of a device to an undesirable state which is a cause of a function defect, we connect *corresponding-to* relation from *causative phenomenon of function defect* in the extended functional ontology to “failure mode” in the FMEA ontology.

(2) Component: “Component” in FMEA means a component which changes to a failure mode. In the extended functional ontology, this corresponds to a “device” which performs a function to be *initial function defect*. Hence, a *corresponding-to* relation is connected from such a device to a component.

(3) Cause: “Cause” in FMEA means a cause of a failure mode. In a manner similar to the “failure mode”, the concept of “cause” is used with some meanings in practice. For example, “wear”, “vibration”, “crack” and “structural defect” are described as causes. In these examples, “wear” and “vibration” correspond to “trigger phenomenon” in the extended functional ontology. “Crack” and “structural defect” correspond to *(effect of) causative phenomenon of function defect*. Since *causative phenomenon of function defect* is mapped to “failure mode” as mentioned previously, a *corresponding-to* relation is connected from “cause” to *trigger phenomenon*, which is an upstream phenomenon of the causative phenomenon of function defect.

(4) Effect: “Effect” in FMEA means how the whole system is eventually changed as an effect of a failure mode. Since “failure mode” corresponds to *causative phenomenon of function defect*, a propagation chain, which is an effect of the causative phenomenon in the extended functional ontology, is regarded as a correspondence concept to “effect”. Since all phenomena concerning a propagation chain are too much for FMEA sheets and not necessary, two concepts, that is, *initial function defect* and *final function defect* in a propagation chain are connected to “effect” by a *corresponding-to* relation. The system displays such two concepts as “initial function defect -> final function defect”.

(5) Action: “Action” in FMEA means adding function to prevent or mitigate “effect”. Since a supplementary function in the extended functional ontology is *corresponding-to* “action”, “action” and *supplementary function* are connected by a *corresponding-to* relation.

(6) Severity and Detection: “Severity” in FMEA is a number indicating how severe the “effect” is. “Detection” in FMEA represents how to detect the failure mode or effect. Currently, an extended functional model does not include such information. Therefore, there is no corresponding concept in the extended functional ontology.

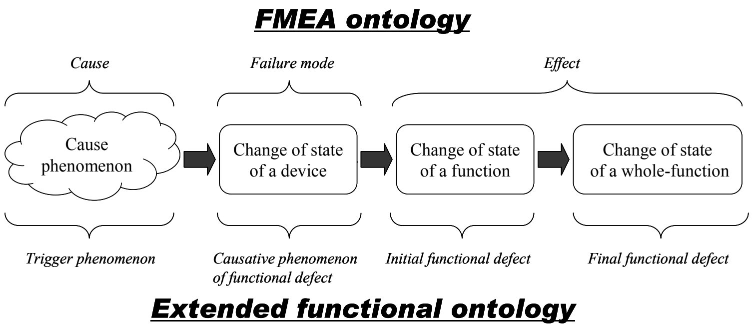

Figure 3. Correspondences of main concepts in the FMEA ontology and the extended functional ontology

7 Implementation of knowledge transformation system

We implemented this system in an environment for building/using ontology named HOZO [11]. The ontologies and the ontology mapping knowledge are described using an ontology editor (OE) of HOZO. Figure 4 shows snapshots of components of the system. Figure 4 (a) shows the extended functional ontology, and Figure 4 (b) shows the FMEA ontology. Figure 4 (c) shows the ontology mapping knowledge between the extended functional ontology and the FMEA ontology. Based on the extended functional ontology, a model in Figure 2 is redescribed using the model editing function of OE. Figure 4 (d) shows such the model.

The transformation engine is implemented using the API of OE, which provides a set of Java-functions in order to access ontologies and ontology-based models stored in Hozo. As mentioned in Section 2, the knowledge transformation consists of the following two steps. As the first step, the system gets necessary information for the transformation from the extended functional models according to the ontologies and the ontology mapping knowledge using API functions to access the knowledge in OE. Then, it makes instances of the corresponding concepts in the FMEA ontology as a FMEA model. In the current implementation, the second step is realized using the W3C XSL Transformations (XSLT) technology. The FMEA model is exported as a XML file by the system and is displayed as the FMEA table by a web browser using a XSLT style sheet prepared previously.

Figure 4 (e) shows an example of an FMEA sheet transformed by the system as output. The second row shows the result of transformation from the extended functional model shown in Figure 2. In the same manner, other rows are transformed from other extended functional models. In Figure 4 (e), as an item of failure mode, a label of a phenomenon connected to the failure mode is displayed without modification, since a transformation of verb form of functional concepts into noun form is omitted, although, generally, nouns such as “fracture”, “misalignment” and “loose” are described as items of a failure mode. This transformation is under investigation.

In this paper, we choose a *causative phenomenon of function defect* as a correspondence concept to “failure mode”. However, when our system is used in practice, a concept corresponding to failure mode in practically used FMEA sheet may be different from *causative phenomenon of function defect*. In such a case, a suitable FMEA sheet can be generated by preparing another ontology mapping knowledge with suitable *corresponding-to* relations.

The framework and the transform system are being used in collaborative research with JAXA (Japan Aerospace Exploration Agency) in which it is verified that our system’s output cover a crucial part of FMEA sheets used in JAXA practically.

8 Benefits of our framework

Our extended functional model and the knowledge transformation system can facilitate engineers’ activities such as design, reliability analysis and redesign. This section discusses merits of our framework in such engineering phases.

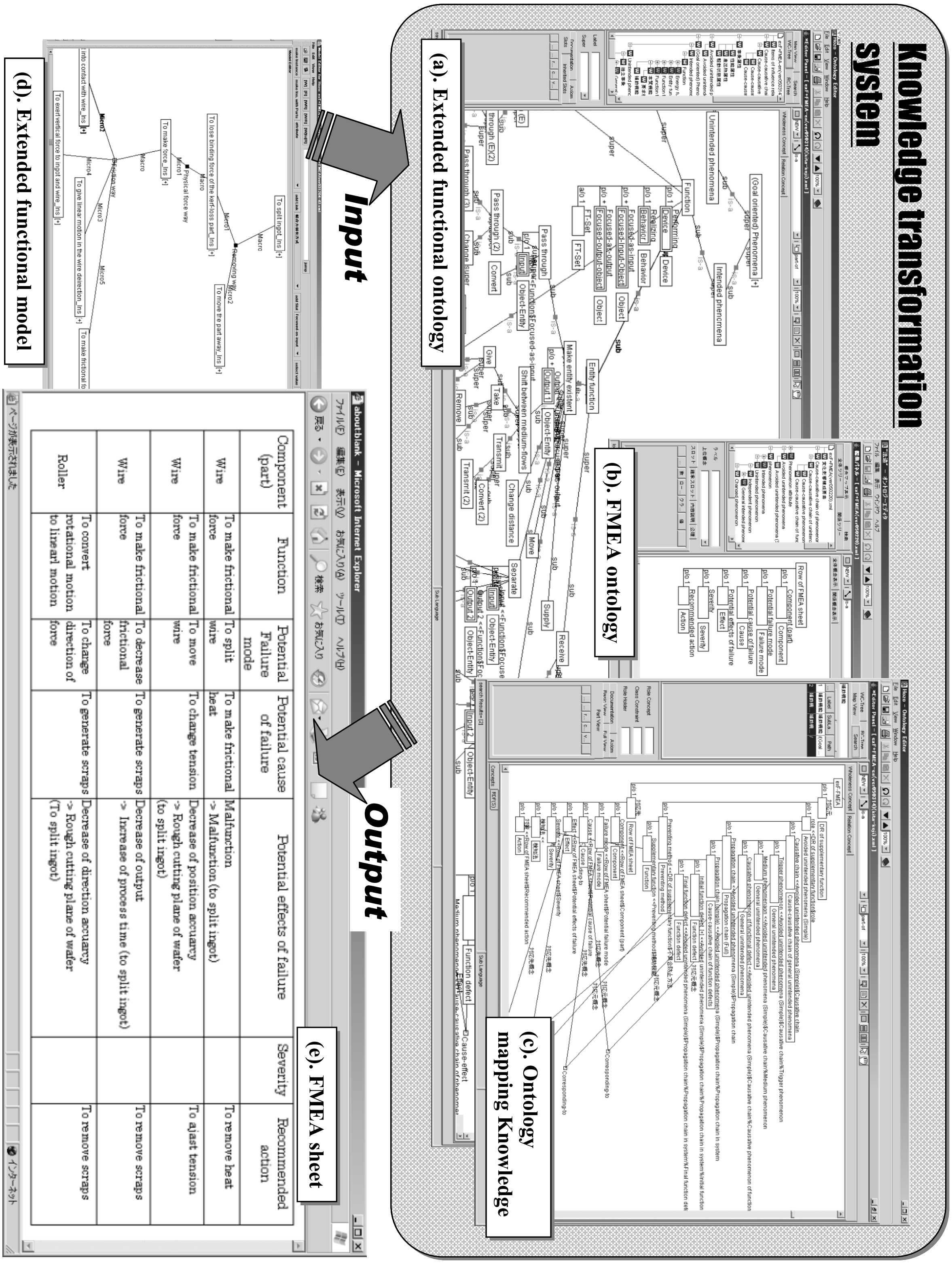

Figure 4. Snapshots of components of knowledge transformation systems

(a) Design phase: Knowledge about functionality of an artifact plays a crucial role in the conceptual design. Especially, the product model from the functional viewpoint (so-called a functional model) describes a part of designer's intention so-called design rationales (DR) [12]. Our extended functional models explicate such DR. Especially, the extended functional model can deal with both functions and unintended phenomena, and hence DRs of supplementary functions can be explicated. Thus, our framework helps designers describe both what is prevented by the supplementary functions and how to prevent it systematically. Such explicit representation helps designers enumerate possible faults and explore available solutions exhaustively. In practice, supplementary functions to retain quality of a design such as stability, precision and reliability are attached to many artifacts as the result of the constant effort for improvement of design. Thus, in order to improve such quality of the design, it is important to reuse DR of the supplementary functions in the past design.

One of the features of systematic description is that the conceptualization of "way of function achievement" in our framework helps us separate "how to achieve a function" from "what is achieved". For example, "to weld steel" is not just a function but a function implying a specific way of achievement, i.e., fusing the target operand. "Welding" should be decomposed into "joining" function and "fusion" way. The same benefit can be gained in the case of unintended phenomena. It helps us keep "how to happen unintended phenomena" from "what is happened".

Another benefit can be brought by a knowledge base of generic way of function achievement [3]. The designer can explore possible deeper phenomena which cause an unintended phenomenon by consulting the knowledge base.

In addition, our models can be used to share a DR to other designers in team activities such as a design review. In general, there are many possible functional defects in an artifact. One of the important goals of a design review is to double check whether possible defects have been considered and how to avoid them. Our framework provides a representational schema of the *avoided unintended phenomena* and the preventing methods as a knowledge medium for such discussion. As the result, other designers can understand whether the designer has sufficiently considered possible functional defects.

In the example of the wire-saw, our models can represent that there are two possible causes of the malfunction of the exerting-force function, that is, "frictional heat" and "extra tension". Moreover, the "to flow slurry" function achieves not only the removing-heat supplementary function but also achieves another supplementary function, that is, "to remove scrapings". Removing scrapings prevents the final functional defect "the decrease of precision of split ingots" which is possibility caused by the scrapings. That is, this model represents that "to flow slurry" has multiple roles.

(b) Reliability analysis phase: Designers can get FMEA sheets in which a failure mode, a cause and an effect are filled semi-automatically by transforming extended functional models described in the design phase. This enables designers to reuse a part of design knowledge in the reliability analysis phase. Thus, it can avoid designer's load of describing similar knowledge repeatedly in different forms.

When a designer reads an FMEA sheet described by another designer, it is sometimes difficult to understand a reason why the action prevents the corresponding effect. Since the transformed FMEA sheet is linked to the original extended functional model, it helps user understand the intention of the action referring to the detailed cause-consequence chain in the extended functional model.

(c) Redesign phase: When a designer performs FMEA of an artifact and finds a high severity of an effect of a function defect in reliability analysis phase, it is necessary to go back to the design phase and then to redesign the original design. In such redesign phase, designers can smoothly start using relationship between an FMEA sheet and an extended functional model.

An extended functional model plays a crucial role in redesign. For example, it is easier for a designer to notice another better method to prevent a functional defect of a required function. This is because the other unintended phenomena to be possibly blocked by the supplementary function are represented in the model of the *avoided unintended phenomena*.

As an improvement, a drastic change of the design is sometimes required instead of such addition of a supplementary function. In a function decomposition tree, some of such drastic changes can be done by a replacement of way of function achievement. In such a replacement, it is necessary to consider effects of the change of the way of function achievement to other functions, such as whether the supplementary function attached to the original way of function achievement is needed or not in the new way of function achievement. Since, in an extended functional model, relationships between functions are explicated, it helps the designer to check such effects.

In the example of the wire-saw, a supplementary function “to flow slurry” is attached to “friction way” for two purposes, that is, removing the frictional heat and removing scrapings. If the “friction way” for the making-force function is replaced by an alternative way, say “inner-diameter blade way”, it can be understood whether the flowing slurry is required or not. In this case, the flowing slurry is still required. Its reason is that the scrapings still cause to decrease the precision of wafers, although the frictional heat does not cause to break the inner-diameter blade.

Moreover, such changes of the extended functional models in the redesign phase are automatically reflected in the FMEA sheet by our transformation system. Therefore, it is possible to reduce load of updating the contents repeatedly.

Our transformation system promotes knowledge circulation among the design phase, the reliability analysis phase and the redesign phase. In a similar manner to the transformation to an FMEA sheet, it is possible to extend the transformation system to other task-dependent models. Such transformations by our system enable dynamic adaptation of the product knowledge to engineering tasks. Therefore, it can improve communication among task practitioners from many design viewpoints across various design phases.

9 Related work and discussion

Abnormal behavior modeling

The formalism of FT (Fault Tree) is apparently similar to our model of unintended behaviors. However, fault trees tend to be ad hoc and some phenomena in the middle of a causal chain tend to be omitted. Our models of unintended behaviors can be consistently described and can decrease such omissions, since the ways of function achievement in the causative chain tree clarify physical principles which explain why the super-phenomena can be achieved (caused) by the specified sub-phenomena. It help designer to be aware of a gap of the causal chain. Described ways of function achievement can be accumulated as general knowledge in a knowledge-base. This knowledge-base also helps engineers describe consistent models. Moreover, although propagation chains of functional defect have a close relation to the functional structure of the product, such a relation cannot be represented in FT.

In the research area on reliability of artifacts and failure physics, general phenomena appeared in the causative chains has been investigated in [13]. A categorization of generic faulty phenomena such as SCWIFT (abbreviation of Stress, Corrosion etc.) has been identified [13]. In our work, such phenomena are organized as generic ways of function achievement for unintended behaviors. Such a generic organization and its integration with ways for intended phenomena are under investigation.

Model Transformation (generation of abnormal behavior model)

In order to generate an FMEA sheet from a functional model automatically, a method called FMAG [14] is proposed. In FMAG, how functions fail (i.e., how malfunction occurs) is described as inference rules. Then, such rules are used to reason out possible failure modes and effects from a functional model. It, however, mainly deals with the change of a function to failure modes and dose not deal with a reason why the failure mode is caused, which is a causative chain in our model. Therefore, it dose not generate causes of FMEA sheets.

In order to capture a larger set of failure modes systematically, there is research on advanced FMEA [15] using behavioral models to simulate device behaviors. However, the models used in such research cannot deal with the causative chain of a functional defect. Thus, DRs of many supplementary functions to block a certain unintended phenomena in a causative chain cannot be represented sufficiency.

In the model in [16], some relations such as support and fixation that constrain motions of components can be represented as functional constraints. In our framework, they are described as a functional structure of kinetic functions of components.

In model-based diagnosis community, product models of normal (intended) behavior also are used [17]. These models can deduce the propagation chain of a functional defect. However, the causative chains cannot be described, since the causative chain is independent of the normal behaviors. Moreover, Davis, R. [18] and Böttcher, C. [19] pointed out that typical model-based diagnostic system cannot deal with an unusual interaction between components like "leakage". Our model can deal with such interaction by using the propagation chain off the functional structure and the influence relation.

The diagnosis using hierarchical functional models instead of behavior models are proposed in [20]. Although the models are similar to our model from the viewpoint of using functional model, they cannot represent the causative chain neither.

Ontology based transformation

As a methodology based on ontologes for knowledge sharing between design and diagnosis tasks, DAEDALUS knowledge engineering framework [21] is proposed. In this framework, an ontology filter is used for model transformation in order to select necessary information from the original model. On the other hand, our system uses the ontology mapping knowledge, which can include more complex relations such as a corresponding relation between multiple concepts in an ontology and a single concept in another ontology. Besides, ontologies in DAEDALUS define highly abstract concepts only such as components, states and event and then are much simpler than our ontologies.

Christel, D. and Parisa, G. propose transformation of product models based on ontologies [22] for knowledge sharing. In this study, two ontologies for two software systems and a generic ontology are constructed as a case study. However, it mainly aims at exchange of product data such as geometry between software systems rather than task-oriented transformation of functional knowledge in our research.

Supplementary function

Some kinds of functions similar to supplementary functions are proposed in [1]. They distinguish between main and auxiliary functions. The former are defined as those sub-functions that serve the overall function directly. On the other hand, the latter are defined as those sub-functions that serve the overall function indirectly. They point out that the auxiliary functions have supportive or complementary characters. Hubka, V. and Eder, W. E. [7] define additional functions that allow (ensure or support) the working functions to fulfill their tasks. They point out that the working functions and the additional function are loosely equivalent to the main functions and the auxiliary (secondary) functions, respectively.

We define the supplementary functions from the viewpoint of not only its essentiality but also influences on unintended phenomena. In the former sense, supplementary functions are similar to the auxiliary functions and the additional functions. The supplementary functions contribute to the whole (macro) functions in the inessential manner by avoiding a functional defect of required function that possibly occurs without such a function.

Limitation

Our model does not include explicit representation concerning shape and time. Therefore, our model can not deal with an error of timing for example. Temporal representation was discussed in [23]. Besides, a modeling cost is high. A designer has to describe many nodes and links to make a model based on our framework. However, once a designer describes a model, the model can be used for multiple different purposes such as redesign and reliability assessment. Thereby, this limitation can be reduced at some level.

10 Conclusion

We have proposed a knowledge transformation system which aims at support of sharing of various kinds of design knowledge such as functional knowledge and knowledge about faulty behaviors. For this purpose, we have established ontologies specifying knowledge models used in various forms, and described ontology mapping knowledge specifying correspondence relations between concepts in the ontologies.

In the same manner of FMEA sheet, the system can transform an extended functional model into other task-dependent models such as FTA using other ontologies for the task-dependent models and ontology mapping knowledge between ontologies. Transformation for FTA is under investigation.

As a long-term goal, we aim at an extension of this system in order to transform an FMEA sheet into an extended functional model inversely. When a generated FMEA sheet by the system is updated by a designer, it is reflected on the original extended functional model semi-automatically. Such inverse transformation enhances knowledge circulation among the design phase, the reliability analysis phase and the redesign phase. This is under investigation.

Acknowledgements: The authors would like to thank Mr. Yoshio Tsutsui and Dr. Yoshikiyo Kato, Japan Aerospace Exploration Agency (JAXA), for their cooperation in the application of our methods in a space system.

References

[1] Pahl, G., and Beitz, W., “Engineering Design - a Systematic Approach”, The Design Council, 1988.

[2] Koji, Y., Kitamura, Y., and Mizoguchi, R., ”Towards Modeling Design Rationale of Supplementary Functions in Conceptual Design”, Proc. of the 5th international Symposium on Tools and Methods of Competitive Engineering (TMCE2004), 2004, 117-130, Lausanne, Switzerland, April 13-17.

[3] Kitamura, Y. and Mizoguchi, R., “Ontology-based description of functional design knowledge and its use in a functional way server”, Expert Systems with Application, Vol 24, Issue 2, 2003, 153-166.

[4] Kitamura, Y., Kashiwase, M., Fuse M., and Mizoguchi, R., ” Deployment of an Ontological Framework of Functional Design Knowledge”, Advanced Engineering Informatics, Vol. 18, Issue 2, 2004, 115-127.

[5] Kitamura, Y., and Mizoguchi, R., “Organizing knowledge about functional decomposition”, Proceedings of 14th International Conference on Engineering Design (ICED-03)

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

[7] Hubka, V., Eder, W.E., ”Theory of Technical Systems”, Springer-Verlag; Berlin, 1998.

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

[9] Sasajima, M., Kitamura, Y., Ikeda, M., and Mizoguchi, R., “FBRL: A Function and Behavior Representation Language”, Proceedings of the fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), 1995, 1830 – 1836.

[10] Military Standard, Procedures for Performing a Failure Mode, Effects and Criticality Analysis, MIL-STD-1629A, 1980.

[11] Kozaki, K., Kitamura, Y., Ikeda, M., and Mizoguchi R., “Hozo: An Environment for Building/Using Ontologies based on a Fundamental Consideration of “Role” and “Relationship””, Proc. of EKAW2002 (2002), 2002, 213-218.

[12] Chandrasekaran, B., Goel, A. K., and Iwasaki, Y., “Functional Representation as Design Rationale”, IEEE Computer, Vol. 26(1), 1993, 48-56.

[13] Shiomi, H., “Failure Analysis and Diagnosis”, JUSE Press, Tokyo, 1977.

[14] Teoh, P.C. and Case, K., "Modelling and Reasoning for Failure Modes and Effects Analysis Generation", Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 218, 2004, pp 289-300, ISSN 0954-4054 .

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

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

[17] De Kleer, J., and Williams, B.C., “Diagnosing Multiple Faults”, Artificial Intelligence, Vol. 32, 97-130

[18] Davis, R., “Diagnostic reasoning based on structure and behavior”, Artificial Intelligence, Vol. 24, 1984, 347-410.

[19] Böttcher, C., “No fault in sructure? - how to diagnose hidden interactions”, Proceedings of IJCAI-95, 1995, 1728-1733.

[20] Chittaro, L., Guida, G., Tasso, C. and Toppano, E., “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, Vol. 23(6), 1993, 1718-1751.

[21] Burton H. Lee., “Using FMEA models and ontologies to build diagnostic models”, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 15, 2001, 281-293.

[22] Christel D. and Parisa C., “Product Data Exchange Using Ongologies”, Artificial Intelligence in Design (AID2002), Cambridge, UK, 2002, 617-637.

[23] van der Vegte W.F, Kitamura, Y., Mizoguchi, R., and Horvath, I., “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), Lagow, Poland, 2002, 115-124.

For more information please contact:
Koji Yusuke,
The Institute of Scientific Industrial Research, Osaka University,
8-1, Mihogaka, Ibaraki, Osaka, 567-0047, Japan.
Tel: +81-6-6879-8416,
Fax: +81-6-6879-2123,
E-mail: koji@ei.sanken.osaka-u.ac.jp,
URL: http://www.ei.sanken.osaka-u.ac.jp/~koji/