In Proc. of the 7th International Conf. of Engineering Design in Integrated Product Development (EDIProD 2011), Wroclaw Poland, 30 June - 1 July, 2011, pp. 39-49, 2011 (invited).

FOREST: AN ONTOLOGICAL MODELING FRAMEWORK FOR PRODUCT-RELATED PROCESSES

YOSHINOBU KITAMURA, YUSUKE KOJI, RIICHIRO MIZOGUCHI
Osaka University
The Institute of Scientific and Industrial Research
e-mail: kita@ei.sanken.osaka-u.ac.jp

Keywords: Process modeling, teleology, ontology, function, fault, FMEA

Abstract: *This paper proposes a holistic teleology-based modeling framework for product-related processes named FOREST. Several processes such as functioning processes, fault-related processes and manufacturing processes can be represented as goal-oriented process trees in the three-dimensional space as an extension of an ontology-based functional modeling framework. One of the expected benefits is to help engineers externalize their process knowledge extensively from several aspects in a comprehensive manner. In this paper, we demonstrate its effect in enumerating fault processes by showing the result of an industrial experiment.*

1. INTRODUCTION

The physical attributes of products change in time in several situations in the product lifecycle. Such temporal changes, the activities (or phenomena) that cause these changes, or their sequences are intuitively called processes. The examples of such processes related with physical changes of products include functioning processes, fault-related processes and manufacturing processes. Although the modeling of such processes has been extensively investigated in the literature (e.g., function [1-12] and fault [13-15]), the modeling frameworks are rather independent of each other and then the relationships among them remain unclear.

Based on this observation, this paper proposes a holistic modeling framework named FOREST [1], which can represent several product-related processes such as functioning processes, fault-causative processes, fault-propagation processes, degrading processes, changing-mode processes and manufacturing processes [2]. The main aims are to represent those processes in a uniform modeling schema and to show the relationship among them clearly. For this, our basic idea is the goal-oriented modeling based on teleological interpretation of those processes in the three-dimensional space. We have developed an ontology-based modeling framework as an extension of our functional modeling framework [16-19] and an integrated modeling method of function and faults [20][21].

This modeling framework is expected to help engineers externalize their process knowledge extensively from several aspects as a comprehensive manner. In this paper, we demonstrate its real effect in enumerating fault processes in an industrial experiment by showing a comparison between the auto-generated FMEA sheets from the models and an FMEA sheet described by experts. Such extensive externalization of knowledge of individuals and sharing it in a design team would contribute to innovation of a product.

This paper is organized as follows. Section 2 introduces our functional modeling framework. An overview of the FOREST modeling framework is presented in Section 3. Section 4 shows its two examples. We discuss some expected benefits of the modeling framework in Section 5 and then we demonstrate its effect in an industrial experiment in Section 6. Section 7 mentions the implementation of a 3D model viewer. Then, related work is discussed followed by some concluding remarks.

2. AN ONTOLOGIAL MODELING OF FUNCTION

This section introduces our functional modeling framework [16-19] in the FOCUS[3] project. In this

---

[1] This is an abbreviation of “Functional Ontology-based Role-oriented procES Trees”.

[2] The process of design activities is intentionally out of scope of this paper. See discussion in Section 8.

[3] This is an abbreviation of “a Functional Ontology for Categorization, Utilization and Systematization” of functional knowledge. Please refer to http://www.ei.sanken.osakau.ac.jp/topics/Focus for the project overview.

framework, an artifact as the whole system is regarded as a hierarchical composite of 'devices' as black-boxes. A device is connected to other devices via ports and can consist of sub-devices. A device as an *agent* operates on (influences) target objects (called *operands*) and thus changes their physical attributes. The operand such as fluid and energy flows through the device via ports. We define *behavior* of a device as the changes in the attribute values of the operands, which are represented as pairs of physical-states as input-output relations.

When a behavior type is identified as the behavior of a device, its instances can play different functions (as roles as discussed below) according to teleological contexts, which we call *function contexts*. For example, when we identify "to exchange (transfer) heat" as the behavior type of a heat exchanger, which is described as temporal changes of the temperatures of the fluids, an instance of this behavior type can play either of the following functions; (1) "to give heat" function when the heat exchanger is used as a heater with a turbine in a power plant, (2) "to remove heat" function when it is used as a radiator with an engine in a car.

Thus, we define an (actual) *function* as "a *role* played by a (device-oriented) behavior in a teleological (function) context" [19]. The *role concept* is a technical term in Ontological Engineering, which cannot be defined without a context and the existence of a role concept depends on the existence of the context [22]. In short, function is a result of "teleological interpretation of behavior of a device under a goal" [16].

We developed an ontology of generic functions (called functional concept ontology in [17] and FOCUS/Tx recently), which represent classes for concrete functions performed by devices. This ontology can be used as a shared taxonomy for verbs in the typical "active verb-object" description of function like in [1][5][6][10]. Such taxonomies for generic functions have been proposed in the literature such as [4][5][7][10] as well. As demonstrated in this paper, this ontology can be used also for describing unintended phenomena.

A function can be achieved by a process of sub-functions. A sub-function is further decomposed into finer-gained functions in a tree-like structure called a function decomposition tree. Fig. 1 indicates its example of a wire-saw. It is designed to slice semiconductor ingots with moving wires. In the function decomposition tree, the whole (macro) function is-achieved-by the sequence of sub (micro) functions (called "is-achieved-by" relations). All functions in the function decomposition tree are instances of the generic functions defined in the functional concept ontology. Such functional structure is similar to the German-style function decomposition in [5], the whole-part relation in [4] and "degree of complexity" in [2]. We introduce the way of function achievement as follows.

The "way of function achievement" is a conceptualization of background knowledge of function decomposition such as physical principles. It represents "how to achieve the function", while function in our framework represents "what to achieve" only. For example, the top-function of the wire-saw is not "to slice ingot" which implies how to achieve a goal. So, it is conceptually separated into the "to split ingot" function and the "removal way of function achievement" as shown in Fig. 1.

Such concrete ways of function achievement are generalized into *generic ways of function achievement*, which are stored in the knowledge base (called *functional way knowledge*). The generic ways are organized in an *is-a* hierarchy and are reused for other function decomposition trees.

3. OVERVIEW OF THE FOREST FRAMEWORK

For establishing FOREST modeling framework as an extension of the functional modeling framework explained in the previous section, we have generalized the function decomposition tree into a *goal-oriented process tree* (called GPT or just 'tree' in this paper). A FOREST model consists of some

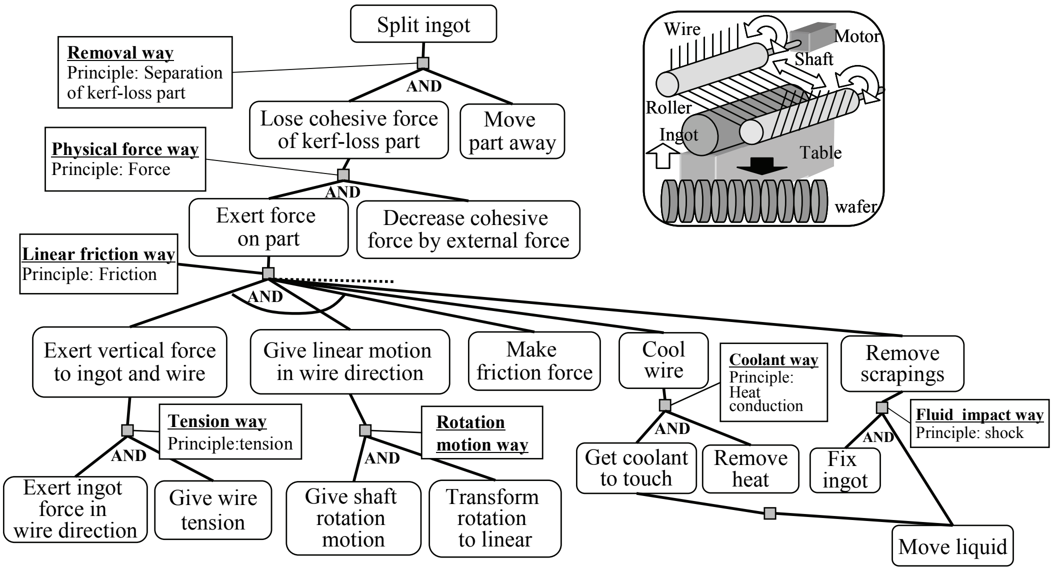

Fig.1. *A function decomposition tree of a wire-saw using our functional modeling framework [18][19]*

goal-oriented process trees in the three-dimensional space as shown in Fig. 2. Fig. 2 shows some typical kinds of GPTs with examples and their relationship.

Firstly, the device(agent)-operand relationship has been generalized into relative one. When a function decomposition tree is identified as a *primary GPT*, the devices and the operands in the tree are called the *primary devices* and *primary operands*. Here, we introduce two quasi-temporal axis; *changing-operand time* and *changing-functional-structure time*. In the time along the former temporal axis, a primary device changes its primary operands. In a typical function decomposition tree shown in Fig. 1, it is the horizontal axis. In the latter time axis, the primary devices and/or their functional structures change.

In a graphical representation shown in Fig. 2, the axes X and Y represent the *changing-operand time* and the *changing-functional-structure time*, respectively. The Z axis represents the grain-sizes of functions (the height of trees). The primary GPT is located in the position of the tree ***A*** in Fig. 2.

The tree ***C*** along the Y axis represents such a process that changes (influences) the primary devices somehow and then the functional structure shown in the tree ***A*** changes to that in the tree ***B***. For example, a process that changes behavioral modes and then functional structures change according to the behavioral modes can be located as a tree ***C***. For instance, for an air-conditioner with two modes; cooling-air and blowing-air, we can describe the functional structure of the cooling mode as a tree ***A*** and that of the blowing mode as a tree ***B***. The functions to switch the behavioral modes are represented as a tree ***C***. In addition, a manufacturing process to create the primary devices is also along the Y axis and then is located at the position of the tree ***D***.

These positions of the goal-oriented process trees represent *roles* of the processes shown in these trees in the relationship between them. These roles are based on the relative device(agent)-operand relationship, where things playing the agent-role influence things playing the operand-role.

As the second generalization, the *function context* is generated into *goal-oriented context*, in which a physical state (e.g., a set of specific attribute values) is intentionally recognized as a goal state to be achieved. The notion of ‘function’ is also generalized into a *goal-oriented process* as a result of teleological interpretation of device-oriented behavior (effect on operands) under the *goal-oriented context*. A goal-oriented process tree represents the *achievement structure* (“is-achieved-by” relation) of sub-processes which contribute to achievement of the goal under the goal-oriented context. In a function decomposition tree with a function context, its goal is intended by designers and/or users. Sub-functions in function decomposition trees contribute to achievement of the whole-function which realizes this intended goal.

As a result of this generalization, we can introduce *unintended context* under which an unintended state as a quasi-goal is achieved by its sub-phenomena as sub-processes. One of typical such processes is a causative process of a fault (*fault-causative process*). Such fault-causative process is initially caused typically by an external disturbance or an internal degrading process along time. Such initial unintended process (phenomenon) causes other unintended processes and then such propagation finally causes a faulty (abnormal) process such as a malfunctioning process. This fault-causative process can be represented as a goal-oriented process tree where the fault as an unintended state is a quasi-goal and is achieved by the causative processes. Some of them change (e.g., degrade) the primary devices and then its normal functional structure changes to an abnormal one. So, such a fault-causative tree is also located at the same

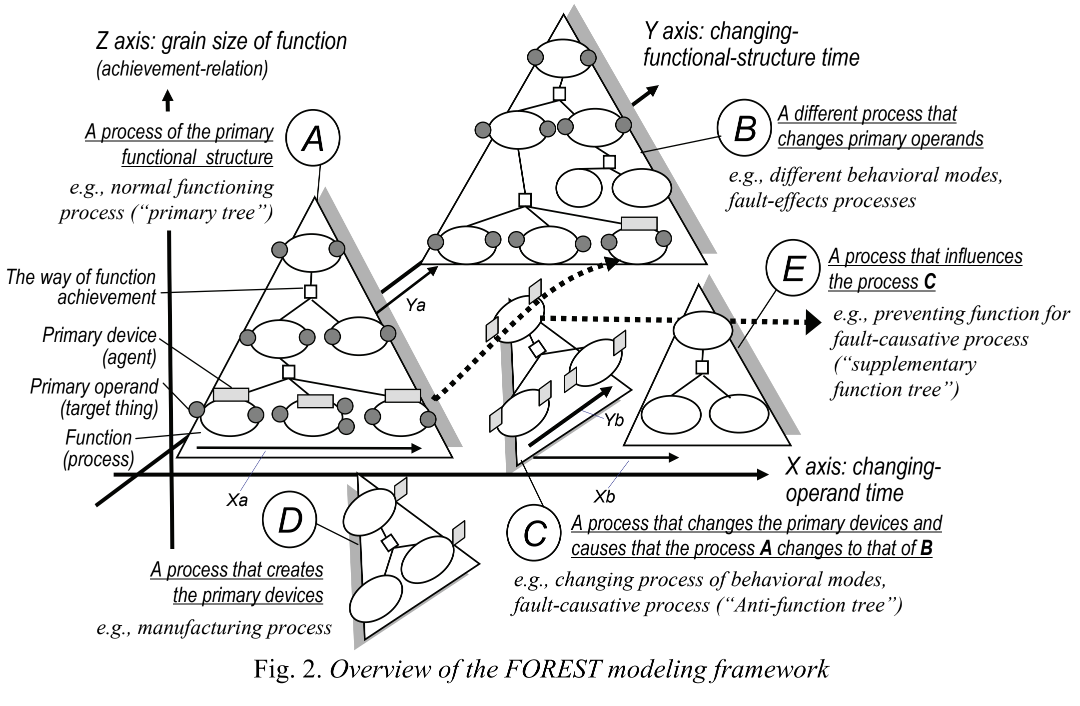

Fig. 2. *Overview of the FOREST modeling framework*

position of the tree ***C***.

We call such processes like the *fault-causative processes* that contribute to occurrence of unintended quasi-goal *anti-functions*. By term ‘anti’, we here mean that an anti-function gives (possibly) harmful influence to the achievement of the function intended by designers and/or users. Although the use of the term ‘function’ is of course somehow counterintuitive for unintended processes, the differences between function and anti-function are just the goal-contexts according to the designer’s intention and their process structures are based on the same achievement relation. On the basis of the generalization of the function context to the goal-oriented context, anti-function can be described using the same concepts defined in the functional concept ontology (FOCUS/Tx) for functions.

The second example of the processes under the *unintended context* is the faulty processes when the system works abnormally. It includes malfunction and insufficient functioning called *functional defects*. The effects of the initial faulty process propagate to other devices and eventually the final abnormal process (e.g., malfunction of the whole system) occurs (called *fault-effects process*). The final state of such the last process in the causal chain is the quasi-goal here. Such goal-oriented process trees can be located also at the position of the tree ***B***, because such process represents an achievement structure different from that of the primary GPT.

Concerning faults, many products have functions to prevent possible faults. Such function can be regarded as a process that changes something related to the fault-causative process (the tree ***C***) and then influences the process ***C***. So, we can locate such process trees at the position of the tree ***E***. Although these functions are usually included in the normal function decomposition tree (like in Fig. 1), the tree ***E*** explicitly shows its role that this process influences the anti-function process represented in the tree ***C***.

Precisely speaking, the meaning of the axis X is categorized into Xa and Xb. Xa represents the changes of the primary operands as explained above. The horizontal relationship within the trees ***A*** and ***B*** in Fig. 2 is Xa. On other hand, the horizontal inter-tree relationship between the trees ***C*** and ***E*** is Xb, which represents the influences on the goal-oriented process ***C*** which changes the primary devices in the tree ***A***. The axis Y is also categorized into Ya and Yb. Ya represents changes of the achievement structure between the trees ***A*** and ***B***. Yb represents changes within the tree ***C***. In many cases**,** it represents the changes (or degradations) of the primary devices in the primary tree ***A***. In other cases, Yb might represent other changes. This asymmetry is due to the limitation of the graphical representation in the 3D space.

4. DEMONSTRATIVE EXAMPLES

4.1. Wire-saw: functions and faults

Fig. 3 shows a FOREST model of a wire-saw as an example. Its normal functional decomposition tree shown in Fig. 1 is the primary GPT (the tree A_1) here and then is located at the position of ***A*** in Fig. 2. Note that the tree A_1 in Fig. 3 shows a portion of the function decomposition tree shown in Fig. 1.

The tree C_1 shows one of the possible *fault-causative processes* of the wire-saw as an *anti-function tree*. As a side-effect of the “make a frictional force” function, frictional heat is caused by the friction. This frictional heat would lead to increase of temperature of the wire and decrease of strength of the wire as degrading along time. Finally

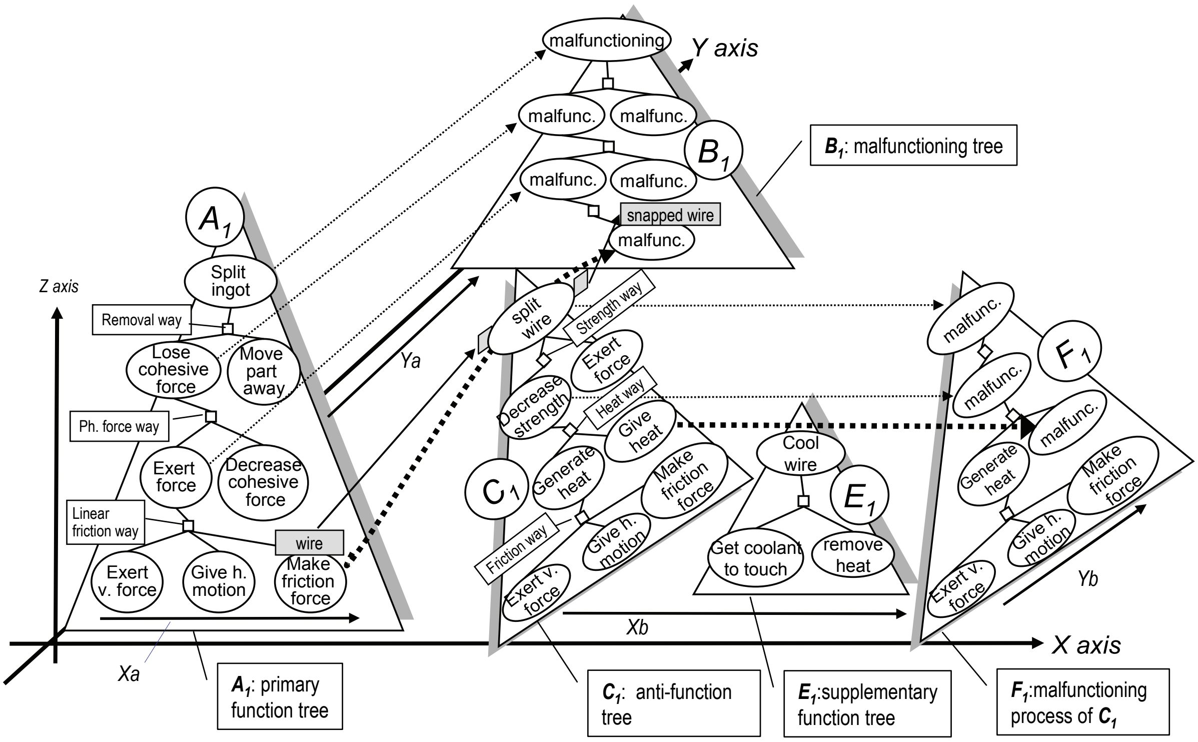

Fig. 3. *An example of a FOREST model of a wire-saw*

the snap of the wire possibly happens. If it happens, the wire malfunctions as a fault. Such causal chain to the snap of the wire is represented as a goal-oriented process tree in which "to split a wire” as the whole *anti-function* is achieved by the causative processes such as “to generate heat” and “to give heat”. As discussed in the previous section, the unintended processes are also described in terms of the concepts defined in the functional concept ontology (FOCUS/Tx), ‘split’ is, for example, used instead of ‘snap’.

This tree $\boldsymbol{C_1}$ breaks a wire as a component of the wire-saw and then the normal functioning processes of the wire-saw changes to defective processes (such as malfunctioning). So, this process affects the primary devices and then influences its functional structures. So, in Fig. 3, the tree $\boldsymbol{C_1}$ is located at the same position of the tree $\boldsymbol{C}$ in Fig. 2.

The tree $\boldsymbol{B_1}$ represents a faulty process which is caused by the process $\boldsymbol{C_1}$ as a result of degradation. In this example, the initial malfunction of the wire propagates to other functions (and devices) and eventually the whole-function, i.e., “to split ingot”, changes to malfunction. This fault-effects process is represented as a goal-oriented process with the malfunction of the wire-saw as the quasi-goal.

The wire-saw has a “cooling-wire” function for preventing the snap of the wire due to heat. This function is not mandatory for achieving the main function of the wire-saw. Rather, this function exists for preventing a possible fault and hence for more stable functioning of the wire-saw. Such a function is called *(preventive) supplementary function* [20]. The cooling function is achieved by its sub-functions such as “get coolant to touch”. Then, its functional structure can be represented as a goal-oriented process tree as shown in the tree $\boldsymbol{E_1}$ in Fig. 3. A goal-oriented process tree of this type is called a *supplementary function tree*.

The effect of this supplementary function tree $\boldsymbol{E_1}$ is to stop the fault-causative process shown in the *anti-function tree* $\boldsymbol{C_1}$. Specifically, the cooling function of the tree $\boldsymbol{E_1}$ prevents the occurrence of the “to give heat (to the wire)” process (*anti-function*) in the tree $\boldsymbol{C_1}$. In other words, this cooling-function influences the giving-heat anti-function and then the anti-function changes to malfunction. Note that this malfunction is good one for the whole function intended by designers and/or users. Thus, the process of the tree $\boldsymbol{E_1}$ influences some of the processes in the tree $\boldsymbol{C_1}$ and then they change into their malfunctioning processes, which are shown as a goal-oriented process tree $\boldsymbol{F_1}$. As a result, the transition from the primary (normal) function tree $\boldsymbol{A_1}$ to its malfunctioning process $\boldsymbol{B_1}$ is prevented.

In Fig. 1, the cooling-function is represented as a sub-function in a single function decomposition tree. As explained above, it is, however, different kind of function, which stops an anti-function and then prevents the abnormal transition of the functional structure. The FOREST model shown in Fig. 3 explicitly shows such an important role played by the cooling-function in the tree $\boldsymbol{E_1}$ and the relationship among the goal-oriented process trees.

In Fig. 3, the GPTs $\boldsymbol{C_1}$, $\boldsymbol{E_1}$ and $\boldsymbol{F_1}$ are located at the different positions on the X axis. As explained in Section 3, they represent not the changes of the primary operands (the axis Xa) but the influences on the anti-function tree $\boldsymbol{C_1}$ which influences the primary GPT $\boldsymbol{A_1}$ (the axis Xb).

4.2. Paperweight: different perspectives for function

The second example is a paperweight. Its function might be captured differently. A typical one is “to prevent possible deviant movement of a piece of paper”. Another one is to “exert vertical force on a piece of paper”. Using the FOREST modeling framework, we can clearly capture these different functions as explained below.

Figure 4 shows an example FOREST model of some processes related to the paperweight. We here regard “to write letters” as the whole-function of the primary tree with “letters (as diagram) on a piece of paper” as the primary goal. This goal is achieved collaboratively by functions of some artifacts (e.g.,

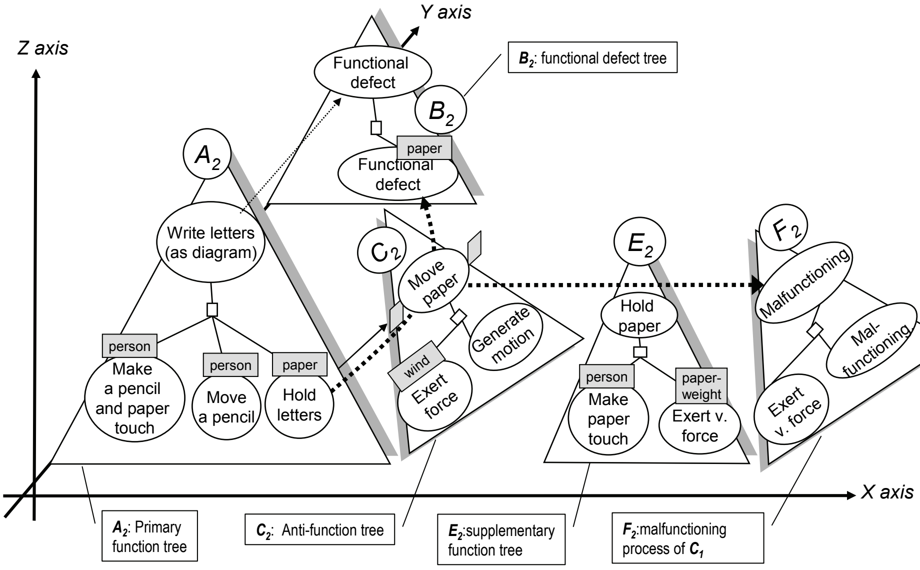

Fig. 4. *An example of a FOREST model of a paperweight*

"to hold letters (as a diagram)" of a piece of paper) and by actions of a person (e.g., "to move a pencil"). This goal-oriented process is represented as the primary tree A_2 in Fig. 4. Note that we here regard a person as a kind of a device which causes some effects on something and then treat a person in the same way of artifacts. We revisit this issue in Section 8.

The tree C_2 in Fig. 4 is an *anti-function tree* which represents a possible unintended process: "to move the piece of paper" by wind (more precisely, "to move the piece of paper from the proper position to a deviant position by external disturbance force"). Then, the tree B_2 in Fig. 4 represents possible *functional defects* which are the results of the possible occurrence of the process shown in C_2.

The tree E_2 in Fig. 4 is a *supplementary function tree* of which the top-goal is "to hold the piece of paper (at the proper position)". This top-function is achieved by "to exert vertical force (on the piece of paper)" which is performed by the paperweight.

Using this model, we can explain several functions of a paperweight including ones mentioned above. The "to prevent the possible deviant movement of a piece of paper" function captures the prevention of the occurrence of the anti-function process (the tree C_2) as a result of the supplementary function (the tree E_2). Note that this is indirect result of the physical effects caused by the paperweight and refers to a possible process of unintended phenomena, which does not occur in the normal run-time of the system.

On the other hand, the "to exert vertical force" function captures only the physical effect directly caused by the paperweight in the tree E_2, which always occurs in the run-time. This does not refer to its role as a supplementary function for preventing unintended processes.

In addition, one might consider "to help write letters" as a function of the paperweight. This function captures that the paperweight plays an assisting role indirectly for achievement of the user's ultimate goal "to write letters" which is the primary goal of the primary tree A_2 in Fig. 4, though this function does not mention how to assist its achievement explicitly.

5. BENEFITS

This section discusses expected benefits of the proposed modeling framework. The benefits discussed in Sections 5.1, 5.2 and 5.3 are general ones. Those in Sections 5.4 and 5.5 are specific to the integrated modeling of fault-related processes. The consideration on these benefits is based on the some practical difficulties in functional modeling of engineers in industry which have been faced in the authors' experience of industrial deployment of the functional modeling framework for knowledge management in manufacturing companies [19], though this deployment has been very successful and many practical benefits have been received.

5.1. Holistic modeling of processes

The proposed method provides the goal-oriented process modeling schema based on teleological interpretation of the physical behaviors as a uniform foundation for holistic modeling of the product-related processes. The FOREST model can represent several kinds of those processes using the same vocabulary with clear relationship with other processes. Such holistic modeling of different processes would contribute to the improvement of exhaustiveness of modeling of possible processes as demonstrated in the next section.

5.2. Discrimination of the processes

In the industrial experience, the engineers sometimes feel difficulties in describing functional models. One of its deep reasons is the difficulty in distinguishing the different kinds of processes. For example, functional models sometimes include (1) adjustment function which is not performed in the run-time, (2) quality improvement by redesign and (3) unintended phenomena to be prevented. The FOREST model makes the roles of the processes in the relationships between them (i.e., the role of each tree in a forest) clearer. Thus, it is expected to contribute to helping the model-authors distinguish different kinds of processes and describe their models separately.

5.3. Clear understanding of perspectives of functions

As discussed in Section 4.2, for an artifact, we can capture several functions. We believe that its main reason is that the perspectives for capturing functions are unclear and implicit. Functions of an artifact are captured in different ways according to such perspectives. In our industrial experience, there are many cases that the perspective is not consistent and confused within a functional model.

The proposed modeling framework contributes to explication of such several different perspectives for functions. As demonstrated in Section 4.2, the model shown in Fig. 4 can explain different (at least, three) perspectives for capturing functions of the paperweight. A fundamental benefit of such modeling is to help the model-authors capture function consistently with mutual dependencies between them.

5.4. Reliability analysis

The benefits discussed in this sub-section and the next sub-section is concerned with the integrated modeling of fault processes. The holistic modeling of fault processes with different processes such as a normal functioning process would contribute to exhaustive enumeration of possible faults. A designer can describe possible malfunctions of each function and their causative processes (like the tree C_1 in Fig. 3) and effect processes (the tree B_1) by referring to the primary functional tree (the tree A_1). The proposed method can represent multiple functions of a device and multiple upper-functions of a sub-function. In such complex cases, a designer might overlook possible faults or possible propagation paths without referring to an explicit functional model.

For reliability analysis, it is useful to transform the model into the conventional FMEA[13][14]-type sheets for easy understanding by designers. Such transformation requires not only transformation of the representation styles (i.e., tree and table) but also ontological alignment of the semantics of the elements in the knowledge forms. We have developed an automatic transformation method based on the ontology mappings, which can generate FMEA sheets from the integrated models of the functional processes and the fault processes [21].

Designers can get FMEA sheets for the reliability analysis by the automatic transformation from the integrated models that can be described in the design phase. It can avoid designer's workload of describing similar knowledge repeatedly in different forms. Moreover, the improvement of exhaustiveness of possible fault enumeration in FMEA can be expected as demonstrated in the next section.

5.5. Redesign for fault prevention

If a designer finds a high severity of an effect of a malfunction, it is necessary to redesign the original design. The FOREST modeling framework makes easier for a designer to notice other better methods to prevent the malfunction, because the other unintended processes to be possibly blocked by a supplementary function are explicitly described in the fault-causative process tree (an anti-function tree like the tree $\boldsymbol{C_1}$). In addition to such prevention of the fault, a designer can add such a function that stops the propagation of the effects of the fault (the tree $\boldsymbol{B_1}$). Such a function is called a *corrective* supplementary function for faults. In either case above, a designer can systematically investigate all the unintended processes explicitly described in the trees such as $\boldsymbol{B_1}$ and $\boldsymbol{C_1}$ as candidates of the processes to be stopped.

Moreover, instead of addition of a supplementary function discussed above, fault prevention could be done also by finding an alternative way of function achievement in the primary functional model (the tree $\boldsymbol{A}$). For example, in the functional structure of a wire-saw shown in the tree $\boldsymbol{A_1}$ in Fig. 3, the "exerting-force" function is achieved by the "linear friction way" using friction phenomena. Alternatively, it can be achieved also by the high-pressure fluid way (e.g., water jets). If this alternative way is adopted, there is no frictional heat as the initial cause of the fault shown in the tree $\boldsymbol{C_1}$.

In addition, the models can be used for sharing the design rationale of (re)designs with other designers in the design review activity. Our framework provides a knowledge medium for the unintended phenomena and the preventing methods. As a result, other designers can double check whether the designer has sufficiently considered possible faults and preventing methods.

6. INDUSTRIAL DEMONSTRATION

This section demonstrates the effect on the possible fault enumeration discussed above in the collaborative research with Japan Aerospace Exploration Agency (JAXA). The target system is an electrical power subsystem of a satellite that controls electric energy in the system using batteries and solar panels etc. We described its normal functioning process and fault-related processes for 11 initial causes of faults. Fig. 5 shows a portion of a FOREST model for the "open cell[4]" fault of a battery[5]. Then, the transformation system generates FMEA sheets. Table 1 shows a portion of one of them. Lastly, we compared the auto-generated FMEA sheets with the FMEA sheets that were previously described by domain experts of JAXA directly (referred to as the original FMEA hereafter).

As a result of the comparison, we found the auto-generated one includes some extra failure-modes and possible effects that are not written in the original one. Those fault modes and possible effects were implicit in the original. The third row of the FMEA sheet in Table 1 shows one of those failure modes and effects. As shown in the tree $\boldsymbol{A_3}$ in Fig. 5, the battery has two functions: to convert electric energy into chemical energy (the charging mode) and to convert chemical energy into electric energy (the discharging mode). The former function contributes to the two upper-functions, that are, to increase electric energy (charging for the discharging mode) and to remove (decrease) extra electric energy in order to control electric energy in the system. In the original FMEA sheet, there is only the failure mode and effects for the increasing-electric-energy function. Thus, the failure mode and effects for the removing-electric-energy function are implicit. In our model, both fault-effect processes are explicitly described as the trees $\boldsymbol{B_3}$ and $\boldsymbol{B_4}$. It is enabled by referring to the functional model shown in the tree $\boldsymbol{A_3}$ that explicitly represents the both upper-functions of the battery.

Note that, in the real satellite system, those implicit fault modes (even if it happened) and effects never cause real serious effects. Thus, redesign was not needed in this case. Even so, according to the aim of the FMEA activity, exhaustive enumeration of failure modes and effects is expected.

In summary, the holistic modeling of functional processes and fault processes can improve the exhaustiveness of fault knowledge. Our aim is not automatic reasoning of fault-related phenomena but to help designers describe their knowledge exhaustively. Integration of fault knowledge with

---

[4] The "open cell" is one of possible faults of a battery where the current does not flow through one or more units (called 'cells') of the battery.

[5] At the time of this experiment, we have used different graphical representation for the integrated model of functioning process and fault processes. Fig. 5 shows such a FOREST model that represents the same contents of the model used in the experiment.

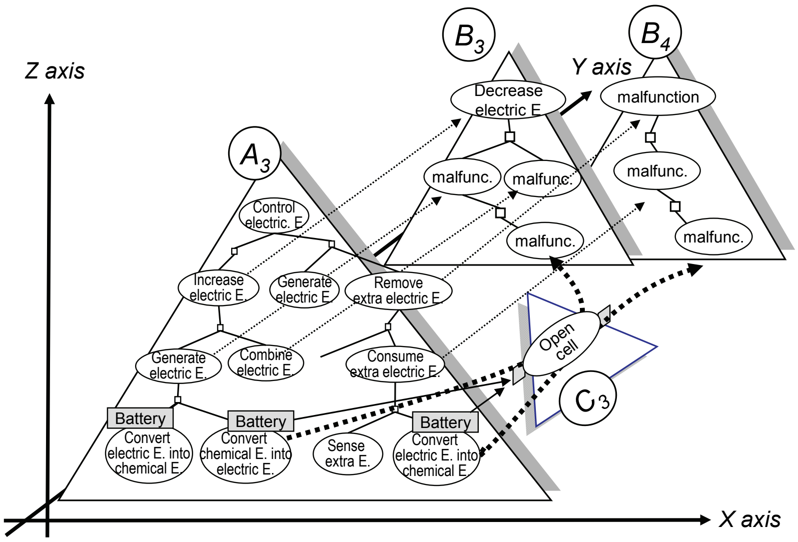

Fig. 5. *A FOREST model of the "open cell" fault of an electrical power subsystem of a satellite*

functional knowledge gives knowledge-authors good stimuli for enumerating possible faults and effects.

7. IMPLEMENTATION

The authors have developed an ontology for the FOREST modeling framework. It defines classes of concepts used in the modeling framework as an extension of those for the previous functional modeling framework. An ontology editor named Hozo[6] is used for its implementation.

A FOREST model can be described using Hozo as instances of those classes defined in the FOREST ontology.

The authors and their colleague have also developed a three-dimensional viewer for the FOREST models. Figure 6 shows its screen snap-shot. Users can view the contents of FOREST models interactively from any viewpoints (positions) in the three-dimensional space and can investigate its details by zooming up.

8. RELATED WORK AND DISCUSSION

There are some modeling languages for generic processes, where a process represents a temporal change, an activity, and/or a sequence of them. For example, Process Specification Language (PSL) [23] can represent temporal occurrence of process (called *activity*) and *object* and their temporal relations such as *before*. IDEF0 (Integration definition for function modeling) [24][25] aims at representation of systems or manufacturing processes, where the term 'function' means 'transformation'. It can represent the input-output flow among such transformation processes mainly. IDEF3 (Process Description Capture Method) [24] can represent temporal relationships between 'Unit of Behavior (UOB)'.

The first crucial difference of the FOREST modeling framework in comparison with these generic process modeling languages is explicit teleological interpretation. In the FOREST model, the sub-processes in a process tree contribute to achievement of the top-process under a specific teleological context with a goal to be achieved. So we call it a *goal-oriented* process tree. As a result, if two processes contribute to different goals, those are described in the different goal-oriented process trees.

Table 1. *An example of the auto-generated FMEA sheets in the experiment.*

Compo-nent	Function	Failure mode	Cause	Effects
Battery	To convert chemical E. into Electric E.	Malfunction	-> To open a cell	Decease of the output of "to increase electric E." function
	To convert electric E. into chemical E.	Malfunction	-> To open a cell	Malfunction of "to remove extra electric E." function

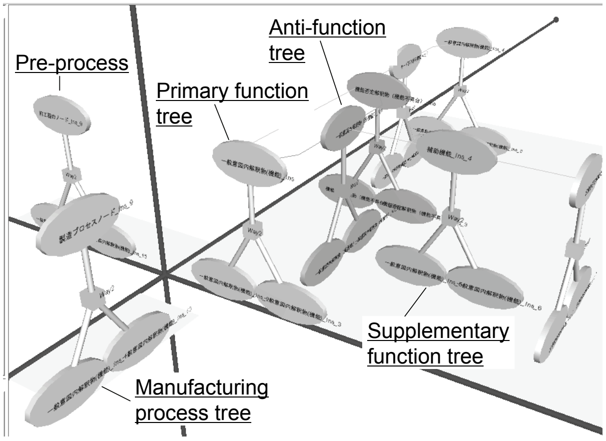

Fig. 6. *A three-dimensional FOREST model viewer*

The second difference is the structure of process models. In many process modeling, the structure of the models (e.g., process decomposition or aggregation) is typically based on temporal relationship like those in PSL and IDEF0. In a FOREST model, the structure is organized based on achievement relationship for a goal. The processes at a hieratical level directly contribute to achievement of the process at the level which is one level higher.

---

[6] http://www.hozo.jp

This is the second reason why we call it a *goal-oriented* process tree. The current proposed modeling framework has weak representation-power for temporal relations.

Such a goal-oriented process tree is, as discussed in this paper, a generalization of our functional models, which are similar to teleological structures in many functional representation methods such as [2]-[6]. This generalization enables us to describe other process such as fault-related processes in the same modeling framework. The second difference is the *way of function achievement*. Similar concepts are called "means" in [26]. One of our aims is to 'purify' function (process) concepts as "what to do" by detaching "how to do". It enables us to describe processes using the same vocabulary.

The representation style of the Fault Tree (FT) [15] is similar to that of the anti-function tree (e.g., C_1 in Fig. 3). Fault trees, however, tend to be ad hoc and some processes in a causal chain tend to be omitted. Our model can decrease such omissions, since the ways of achievement as physical principles help designer to be aware of a gap of the causal chain. Moreover, although the causal chain of functional defects and/or malfunctions has a close relationship with the functional structure, such a relationship is implicit in FT.

The models representing normal behavior of the target system and/or functional structures are used for automatic FMEA generation such as AFMEA [27] and FMAG [28], model-based diagnosis [29][30] and fault identification and propagation [31][32]. These models can automatically deduce the effect-propagation of a functional defect along with the behavioral/functional structure. However, the deep causative chains of the fault are outside of the scope of such automatic reasoning, since such causative chains are independent of the normal behavior. So, the proposed system in [21] aims not at such automatic reasoning of causal relations but at just transformation of the models written by designers. Although a designer must describe the detailed processes, the knowledge-base of the generic ways of achievement would be able to reduce the model-author's workload.

In [33], a mapping between function and failure mode is derived from the component-failure mode matrix and the function-component matrix. It can be reused for failure prevention and concept generation in the functional design in FFDM [34]. FFDM was extended to identify risk likelihood and risk consequence in Risk in Early Design (RED) [35]. The similar generic and reusable past knowledge about occurrence of failure modes can be represented as a *generic way of achievement* that is associated with its possible fault process trees and their supplementary function trees. Although the current FFDM does not address the "cause" of failure modes [34], the fault-causative process tree can help designers investigate prevention methods systematically for each step of the causal chain. Our method treats neither probability [33], risk likelihood in [35] nor severity of faults.

An ontology-based methodology for knowledge sharing between design and diagnosis tasks has been proposed in [36]. It is based on a simple ontology and an ontology filter. We use more fundamental ontologies and an ontology mapping.

Some representation frameworks on the *use processes* and/or scenarios (e.g., [37][38]) by human users with interaction with product function have been proposed. As discussed in Section 4.2, in the FOREST modeling framework, processes are distinguished rather based on the device(agent)-operand relationship. Thus, if a process of user operations changes the primary devices and then a functional structure changes, then it is represented as an independent goal-oriented tree. If not, such user activities are integrated in the primary tree or other trees. Such modeling is dependent on the decision of the primary tree, the primary device and the primary operands.

Note that this modeling framework does not cope with the processes of design activities for products and manufacturing process. These processes are to determine *specifications* of devices, functional structure or manufacturing processes. So, the design process is totally different from the product-related processes in the FOREST modeling framework. Ontological work for designing processes has been investigated in elsewhere such as [39][40].

9. CONCLUDING REMARKS

This paper has proposed a holistic modeling framework for product-related processes. It aims at the comprehensive modeling of several processes and explication of their relationship. We have demonstrated its effect in improvement of exhaustiveness of the enumeration of the fault-related processes. Such exhaustive externalization of designers' knowledge in a holistic model would contribute to reliability analysis, fault prevention and innovation of a product.

Acknowledgements: The authors would like to thank Masataka Takauchi for implementation of the 3D FOREST model viewer. Special thanks go to Mr. Yoshio Tsutsui and Dr. Yoshikiyo Kato, Japan Aerospace Exploration Agency (JAXA), for their cooperation in the application to the space system.

References

[1] Miles, L. D., *Techniques of value analysis and engineering*, McGraw-hill, 1961.

[2] Hubka, V., Eder, W.E., *Theory of Technical Systems*, Springer-Verlag, 1988.

[3] Chandrasekaran, B., Goel, A. K., Iwasaki, Y, *Functional representation as design rationale*, Computer, 26 (1), 1993, pp. 48-56.

[4] Lind, M., *Modeling goals and functions of complex industrial plants*, Applied Artificial Intelligence, 8, 1994, pp. 259-283.

[5] Pahl, G., Beitz, W., *Engineering Design - a Systematic Approach*, Springer-Verlag, 1996.

[6] Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., Tomiyama, T., *Supporting conceptual design based on the function-behavior-state modeler*, AI EDAM, 10, 1996, pp. 275-288.

[7] Kirschman, C.F., Fadel, G.M., *Classifying Functions for Mechanical Design*, Journal of Mechanical Design, 120 (3), 1998, pp. 475-482.

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

[9] Gero, J.S., Kannengiesser, U., *The Situated Function-Behaviour-Structure Framework*, In Proc. of Artificial Intelligence in Design '02, 2002, pp. 89-104.

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

[11] Stone, R. B., Chakrabarti, A. (eds.), *Special issues: Engineering applications of representations of function*, AI EDAM, 19 (2 and 3), 2005.

[12] Goel, A. K., Rugaber, S., Vattam, S., *Structure, behavior, and function of complex systems: the structure, behavior, and function modeling language*, AIEDAM, 23 (1), 2009, pp. 23-35.

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

[14] Rausand, M., Oien, K., *The basic concepts of failure analysis*, Reliability Engineering & System Safety, 58, 1996, pp. 73–83.

[15] Lee, W.S., Grosh, D.L., Tillman, F.A., Lie, C.H. *Fault tree analysis, methods, and applications - a review*. IEEE Transactions on Reliability R-34(3), 1985, pp. 194–203.

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

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

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

[19] Kitamura, Y., Koji, Y., Mizoguchi, R., *An Ontological Model of Device Function: Industrial Deployment and Lessons Learned*, Applied Ontology, 1(3-4), 2006, pp. 237-262.

[20] Koji, Y., Kitamura, Y., Mizoguchi, R., *Towards modeling design rationale of supplementary functions in conceptual design*, In Proc. of the 5th Int'l Symposium on Tools and Methods of Competitive Engineering (TMCE2004), 2004, pp. 117-130.

[21] Koji, Y., Kitamura, Y., Mizoguchi, R., *Ontology-based transformation from an extended functional model to FMEA*, In Proc. of the 15th Int'l Conf. on Engineering Design (ICED '05), 2005, 264.81.

[22] Mizoguchi, R., Sunagawa, E., Kozaki, K. Kitamura, Y., *The Model of Roles within an Ontology Development Tool: Hozo*, Applied Ontology, 2(2), 2007, pp. 159-179.

[23] National Institute of Standards and Technology, *Process Specification Language (PSL)*, http://www.mel.nist.gov/psl/index.html, 2008.

[24] Knowledge Based Systems, Inc., *IDEF Family of Methods*, http://www.idef.com/, 2010.

[25] National Institute of Standards and Technology, *Integration definition for function modeling (IDEF0)*, Federal Information Processing Standards Publications 183, 1983.

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

[27] Steven, K., Peder, F., Ishii, K., *Advanced Failure Modes and Effects Analysis of Complex Processes*, In Proc. of the ASME Design for Manufacturing Conf., 1999.

[28] Teoh, P.C., Case, K., *Modelling and Reasoning for Failure Modes and Effects Analysis Generation*, In Proc. of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 218, 2004, pp. 289-300.

[29] De Kleer, J., and Williams, B.C., *Diagnosing Multiple Faults*, Artificial Intelligence, 32(1), 1987, pp. 97-130.

[30] Chittaro, L., Guida, G., Tasso, C., 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, 23(6), 1993, pp. 1718-1751.

[31] Kurtoglu, T., Tumer, I. Y., *A graph-based fault identification and propagation framework for functional design of complex systems*, Journal of Mechanical Design, 130(5), 2008.

[32] Krus, D., Lough, K. G., *Applying Function-based Failure Propagation in Conceptual Design*, In Proc. of ASME IDETC/CIE 2007, DETC2007-35475, 2007.

[33] Tumer, I. Y., Stone, R. B., *Mapping function to failure mode during component development,* Research in Engineering Design, 14(1), 2003, pp. 25-33.

[34] Stone, R. B., Tumer, I. Y., Van Wie, M., *The function-failure design method*, Journal of Mechanical Design, 127(3), 2005, pp. 397-407.

[35] Lough, K. G., Stone, R. B., Tumer, I., *The risk in early design (RED) method: Likelihood and consequence formulations*, In Proc. of ASME IDETC/CIE 2006, DETC2006-99375, 2006.

[36] Lee, B. H., *Using FMEA models and ontologies to build diagnostic models*, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 15(4), 2001, pp. 281-293.

[37] Nagel; R. L., Hutcheson, R., McAdams, D. A., Stone, R, *Process and event modelling for conceptual design*, Journal of Engineering Design, 22: 3, 2011, pp. 145 - 164.

[38] van der Vegte, W., *Testing virtual use with scenarios*, Ph.D. Thesis, Delft University of Technology, 2009.

[39] Sim, S. K., Duffy, A. H. B., *Towards an ontology of generic engineering design activities*, Res Eng Design, 14, 2003, pp. 200–223.

[40] Gero, J. S., Kannengiesser, U., *A function-behavior-structure ontology of processes*, AI EDAM, 21, 2007, pp. 379-391.