ONTOLOGY-BASED MODELING OF PRODUCT FUNCTIONALITY AND USE
PART 2: CONSIDERING USE AND UNINTENDED BEHAVIOR

W.F VAN DER VEGTE*, Y. KITAMURA, R. MIZOGUCHI**, I HORVÁTH***
*Delft University of Technology, Faculty of Industrial Design Engineering
e-mail: {w.f.vandervegte, i.horvath}@io.tudelft.nl
**Osaka University, Institute of Scientific and Industrial Research
e-mail: {kita,miz}@ei.sanken.osaka-u.ac.jp

Keywords: use-process modeling, CAD, conceptual design, ontologies, unintended behavior, FMEA

Abstract: *the function-behavior representation language FBRL was originally devised for modeling and knowledge management of intended product behavior. This paper explores its potential for application to other-than-intended behavior in a use context, introducing consideration of the user and the environment. We found that slightly adapted building blocks from as-is FBRL can be applied to behavior that is unintended and/or not performed by the product. To support anticipation of unintended behavior in design, special attention has to be paid to the knowledge that connects product functions, user actions and environment behavior. We distinguish typical and atypical forms of unintended use. Some forms of typical unintended use can be directly derived from the intended use. Yet, most forms of unintended use require additional knowledge, e.g., from user observations. To include such knowledge, subsequent effort has to be put into its systematization*

1. INTRODUCTION

There is a growing need for improved support of modeling and forecasting life-cycle processes in computer-aided conceptual design of various kinds of products, ranging from consumer appliances to manufacturing systems. Probably the most crucial phase in a product life-cycle is the stage in which the product is used by users or customers, and intended to fulfill its assumed functions. One of the issues in research on this topic studied in the Integrated Concept Advancement (ICA) group at Delft University of Technology, is how to include knowledge related to the use stage of products (artifacts) in computer-aided conceptual design, as a supplement to functional modeling, artifact modeling and artifact-behavior modeling. This issue has to be considered in the context of the increasing deployment of knowledge-intensive systems in computer support of design.

In the research field of knowledge representation and knowledge processing, the application of ontologies has proven to be an advantageous paradigm over recent years. In the research at hand, the ICA group seeks to apply the ontology paradigm to so-called design concepts that offer integrated support to the product designer for modeling artifact geometry and artifact behavior, in close connection to the artifact's function and its intended use by humans. An example of mature design-support oriented research based on a common ontological foundation is the development of several components and tools that started in the mid-1990s with the conception of FBRL [1] as discussed in the first part of this twofold paper. The Mizoguchi Lab has also successfully applied the ontology paradigm in various other application areas, such as intelligent educational systems and diagnostic systems, and it is seeking to extend the currently covered areas of application. To explore the possibilities of applying FBRL-like ontological principles to product-use processes a cooperation between Delft University of Technology and Osaka University was started in 2002.

This second part of the twofold paper reports on a first explorative study into the extension of FBRL modeling towards the inclusion of unintended behavior and product use. It covers the following research items:

- Setting out the objectives and proposing a tentative architecture for a design-support system featuring ontology-based modeling of the use process of a product.

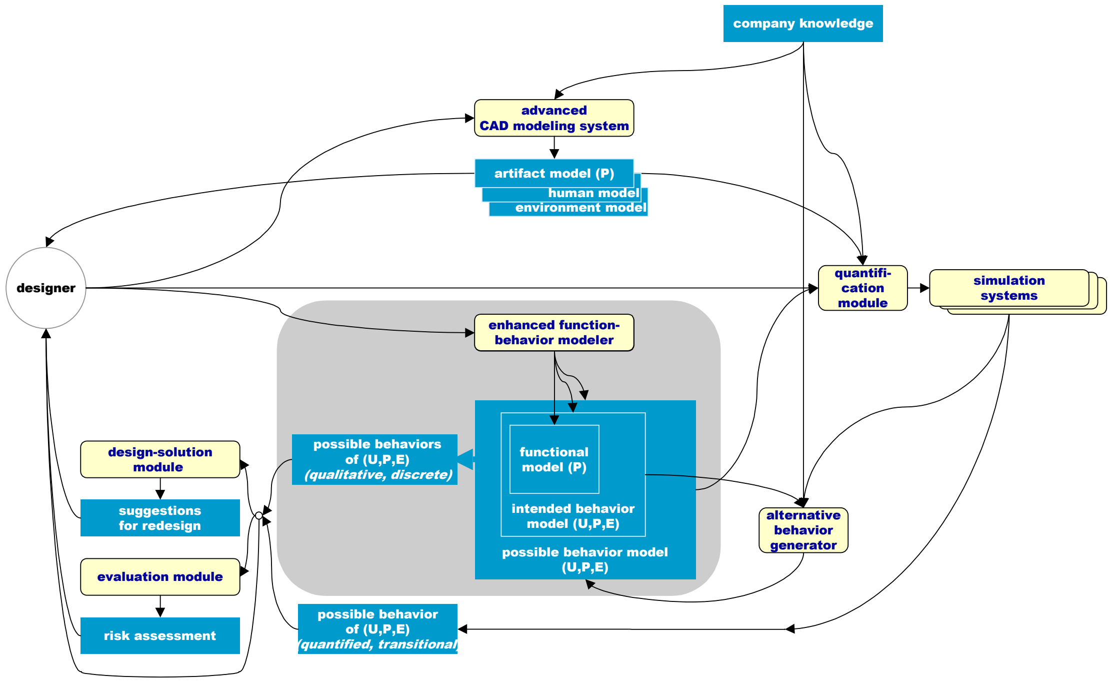

Figure 1. *A tentative architecture for a design-support system featuring ontology-based modeling of the use process of a product.*

- Exploration into extending the FBRL-based family of tools and techniques to use-process modeling.
- Exploration into concrete forms of design support based on such tools.

The following sections will discuss the above items on a more detailed level. The order of presentation in this list does not necessarily reflect a chronological order in the research activities.

2. OBJECTIVES AND ARCHITECTURE FOR A DESIGN-SUPPORT SYSTEM

The objectives of current functional-ontology modeling are (a) to provide insight into the rationale why designers applied particular design solutions by making the intended behavior explicit and (b) to provide computer-generated suggestions for alternatives based on the given functions in a product [2].
An extension of functional-ontology modeling towards use-process modeling, however, will have a more specific focus on forecasting the different use processes that are possible if behavior of the user and the environment have to be taken into account as well [3]. Unlike the product design, the user and the environment are not likely to be changed by the designer. A product may sometimes not behave as was intended by the designer[1] on its own behalf, but the risk that a user or an environment does not behave as was intended, and may cause effects that are harmful or otherwise unwanted, is even more plausible. Moreover, in many cases unintended behavior of the product itself is likely to be prompted by the user or the environment. Therefore, we want to offer assistance in managing the knowledge about *possible* (i.e., intended and unintended) use processes so that designers can anticipate them. This knowledge can originate from various sources, such as simulations, insight gained from previous products, or data collected from interactive user participation sessions. Needless to say, it will not be realistic to capture and manage knowledge about *all* possible use processes, but we do not strive to exclude any particular use a-priori.
Figure 1 outlines a tentative setup for a design-support system that includes ontology-based modeling of the use process. In this setup, an 'enhanced function-behavior modeler' managing (a) a functional model, (b) an intended-behavior model and (c) a possible-behavior model of the product, the user and the environment, is envisaged as the starting point for further developments. Such further developments can include (1) an alternative-behavior generator, that uses information from intended behavior, company knowledge and simulation systems to generate (forecasts of) unintended behavior, (2) a quantification module to facilitate integration of the otherwise qualitative system with quantitative design support as it is offered in CAD models and simulations, (3) an evaluation module to assess the risk of possible behaviors and to help the designer decide whether it calls for a redesign and (4) a design-solution module as an extension of FBRL's current ability to provide alternatives based on the given functions in a product. The extension would deal with other-than-product and other-than-intended behavior.
The present functional-ontology modeler is the starting point for the enhanced functional modeler depicted in figure 1. The exploratory work that we

[1] In this document, the terms 'intended' and 'unintended' refer to the designer, *not* the user.

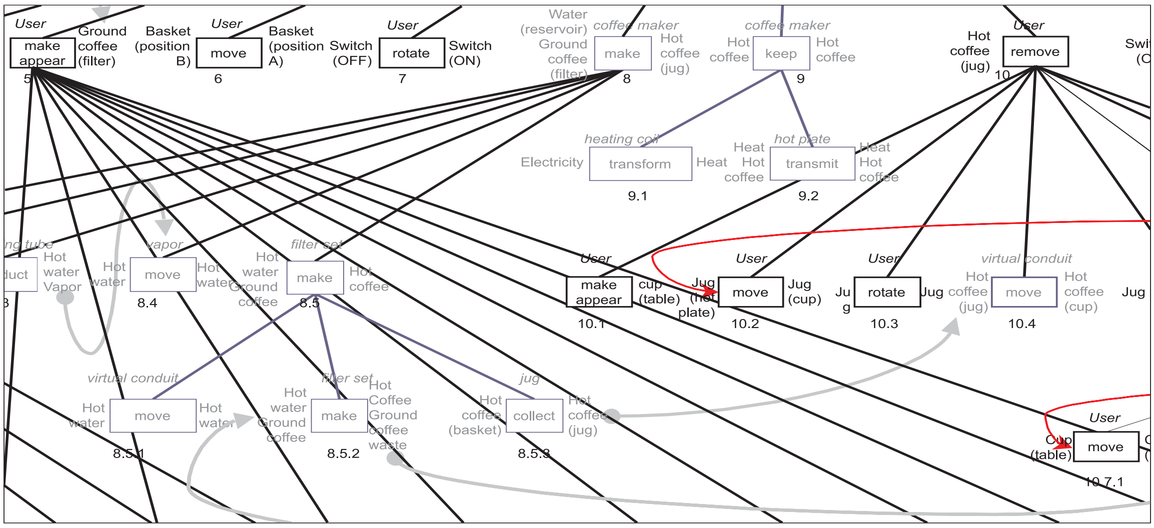

Figure 2 *Detail of an FBRL-based model of a use process, including intended behavior of the user and the environment. For an impression of the complete model, see figure 3.*

discuss in this paper focused on the central area (rounded rectangle in the background), i.e. the extension of functional modeling towards modeling the possible behavior of the product and the user and the environment.

3. RELATED WORK

To some extent, other research dealing with computer support for considering unintended behavior and/or behavior of users and the environment together with product behavior, can be considered related to our work, although the extensions outside the central area in figure 1 are typically not included. Work in the area of computer-aided failure-mode and effects analysis (FMEA) as presented by Kmenta & Ishii, Hata et al. and Lee focuses on unintended behavior (failure, in particular), but it tends to concentrate on internal behavior of the product [4], [5], [6]. Lee's work even involves inclusion of knowledge in an ontology, but the knowledge handled concentrates on probability calculation. A focus on the product's internal behavior is also found in a knowledge-based approach to redesign presented by Goel & Chandrasekaran [7]. Furthermore, some modeling techniques have been proposed for processes that include both user actions and product functions (e.g. by Buur et al., Otto & Wood, Suto et al. [8], [9], [10]). In the models presented in that area, unintended behaviors, if included at all, are 'hard-wired' into a fixed logical scheme that is intended to capture a selected subset of possible use processes. Our research pursues a more open attitude towards handling unintended behavior.

4. EXTENSION OF FBRL TO USE-PROCESS MODELING

The functional ontology of the coffee maker that we discussed in the first part typically applies to intended operation processes, i.e. the product operating autonomously – without the intervention of users – and according to the designer's expectations. In a typical use process, additional attention has to be paid to involvement of the user and the environment, as well as unintended behavior of the product, the user and the environment.

Our hypothesis was that relatively little alteration was needed to apply the building blocks for functional models to models that also include (1) other-than-product behavior and (2) other-than-intended behavior. First, we extended the functional model of the coffee maker with intended behavior of the user and the environment of the product. In that case, we have to consider not only behavior of which the product is the agent[2] but also behavior of which the user or the environment is the agent. Like a product function, behavior of which the user or the environment is the agent can be decomposed into discrete entities. For user behavior, these discrete entities are usually referred to as *actions*. Intended user actions, i.e. user actions that are anticipated by the designer of a product, can be referred to as *tasks*. Together with product functions and expected behavior of the environment, tasks form the starting point for a decomposition of the intended use process. Typically, such a model comprises the idealized use process that is, for instance, prescribed in a user manual. The model could be created using the basic principles of functional-ontology modeling (figure 2, figure 3). Only a limited number of concepts typical to product use had to be added in order to create an integrated model of the intended use and functioning of the example product:

- some function-describing terms have to be added, specifically related to human manipulation of objects in space.
- attention has to be paid to representation of invoking functions and terminating functions of the product (switching on, switching off), and

[2] The term 'agent' is used here to indicate the actor, or grammatical subject of an action. For the grammatical object, the term 'operand' is used.

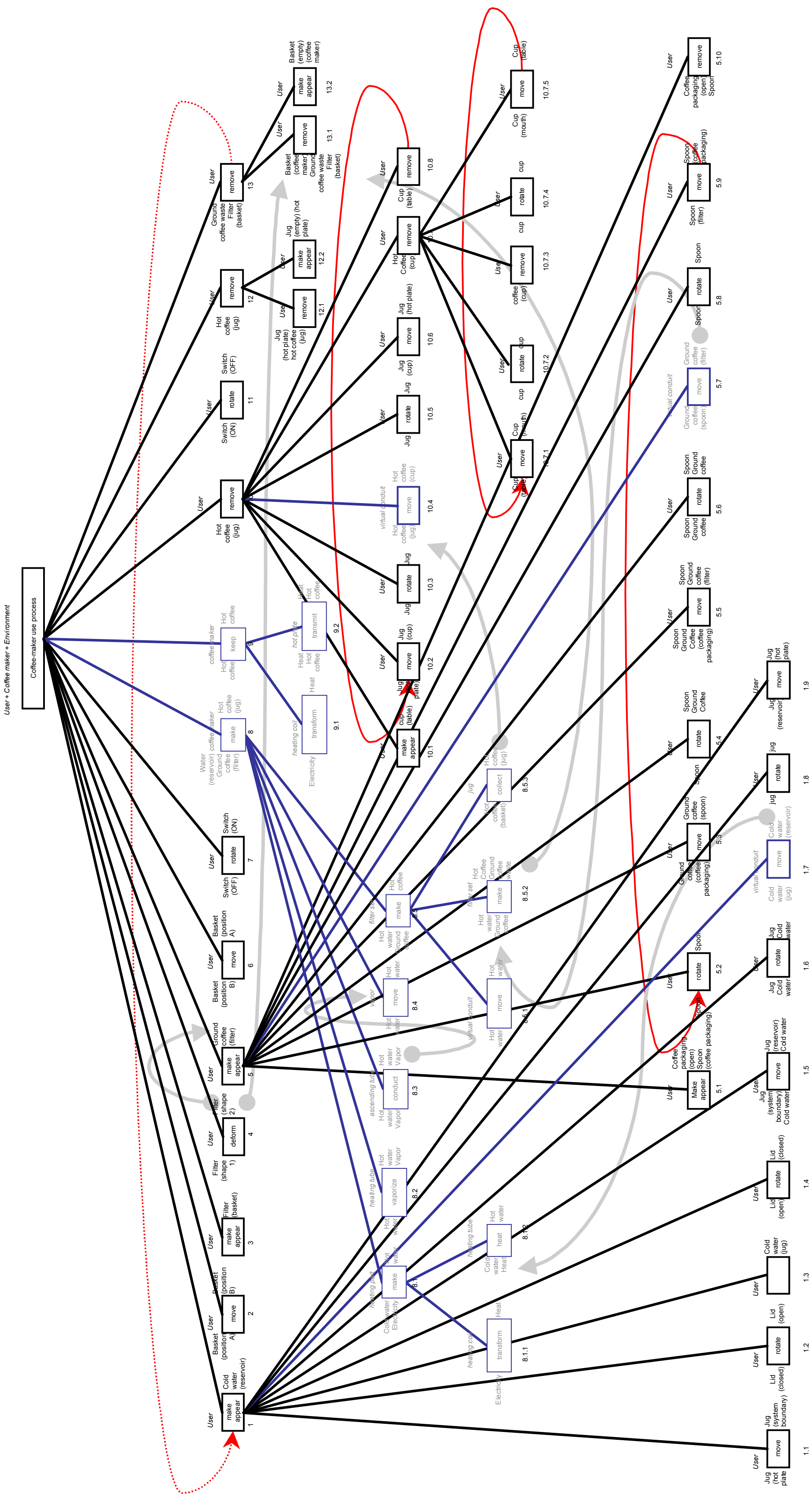

Figure 3. *FBRL-based model of a use process, including intended behavior of the user and the environment.*

- inclusion is needed of product functions merely offering passive support to user actions, such as sliders to guide the displacement of a component.

For other-than intended behavior, we identified four *typical* patterns of deviating from intended use, all defined in terms of user tasks:

- additions of user tasks to, and omissions of user tasks from the intended tasks,
- variations in the temporal relationships between tasks,
- variations in the decomposition of tasks into subtasks,
- user acting on operands other than the designer intended,
- variations in detailed descriptors of tasks (such as locations, orientations, shapes).

These typical deviation patterns cover only a small subset of possible unintended behavior. In addition, the number of possible *atypical* deviations that might be worth considering is infinite, including forms of completely aberrant behavior – such as using the hot plate of a coffee maker for frying eggs.

Other-than-intended behavior can be modeled using the same building blocks. An important issue here is, that there are countless specific use processes built up from forms of unintended behavior. These cannot be combined straightforwardly into one model of one use process. Hence, we focused on the knowledge that connects the building blocks of use processes, intended and unintended, to each other. This knowledge can be inventoried in the form of relationships and dependencies of various nature, e.g., temporal, hierarchical, semantic etc. We found that, to some extent, the knowledge from the intended use can directly be deployed to generate certain simple forms of typical unintended use (see section 6). In other cases, 'advanced' additional knowledge will be needed – for instance from company history, or from simulation results. This is especially true for atypical behaviors.

With respect to the capabilities of the existing functional ontology to cover the knowledge that connects the building blocks, it may is obvious that support for including 'advanced' additional knowledge is currently missing. For the more straightforward forms of connecting knowledge, we found that there are three items that require special attention when extending the scope of FBRL: (a) explicit representation of agents and operands, (b) dynamic role assignment to entities and (c) explicit specification of temporal relationships.

- Explicit representation of the agents/operands and dynamic role assignment to entities can connect actions and functions in which entities participate in different roles. In regular FBRL, the agent is always (a component of) the product. Role assignment is a fixed property for a certain component or entity. When considering the user and the environment, more flexibility is required. For instance, the filter of a coffee maker performs the agent role in the coffee-making process, but it is operand in the connected user action that is performed beforehand, i.e. when it is inserted by the user.
- The connecting role of temporal relationships may be apparent, but current FBRL does not offer possibilities to include this kind of knowledge. This is due to its focus on 'steady-state' processes that are not interrupted or disturbed by external influences such as users. A common technique to specify temporal relationships is to employ interval logic [11]. Figure 4 shows how this technique can be applied to specify the temporal relationships in the intended use process of the coffee maker.

Tasks and functions are recursively specified according to the format *m.n* with *n* a sub-task or sub-function of *m*, *n = 1, .., nmax.*

The default intended relationships are:
m.n BEFORE *m.(n+1)*
m.nmax FINISH *m*
m START *m.n* for *n=1*

Non-default intended relationships (see figure 2 and figure 3 for references):

```
1 BEFORE 7
1.3 BEFORE 1.5
1.4 BEFORE 1.9
1.6 STARTS 1.7
1.8 STARTS 1.9

5.2 OVERLAPS 5.3
5.6 OVERLAPS 5.7
5.9 STARTS (5.2 OR 5.10)

8 STARTS 9
8.1.1 STARTED-BY 8.1.2
8.1 CONTAINS 8.2
8.2 CONTAINS 8.3
8.3 STARTS 8.4
8.4 CONTAINS 8.5
8.5.1 OVERLAPS 8.5.2
8.5.2 OVERLAPS 8.5.3

9 FINISHED-BY 11
9.1 EQUAL 9.2

10.3 OVERLAPS 10.4
10.5 STARTS 10.6
10.7 BEFORE (10.8 OR 10.2)
10.7.2 OVERLAPS 10.7.3
10.7.4 STARTS 10.7.5
(10.7.5 FINISHES 10.7) OR (10.7.5 BEFORE
10.7.1)

13 BEFORE 1 [refers to next coffee-making
cycle]
```

Figure 4. *Temporal relationships in the example product, based on interval logic* [11].

5. AN ONTOLOGY-BASED SCHEME FOR THE CONTENT OF MODELS

Figure 5 shows an overview of a scheme for models of product, user and environment. This scheme represents a general structure of the models (the center part of the figure), i.e., major categories (sub-

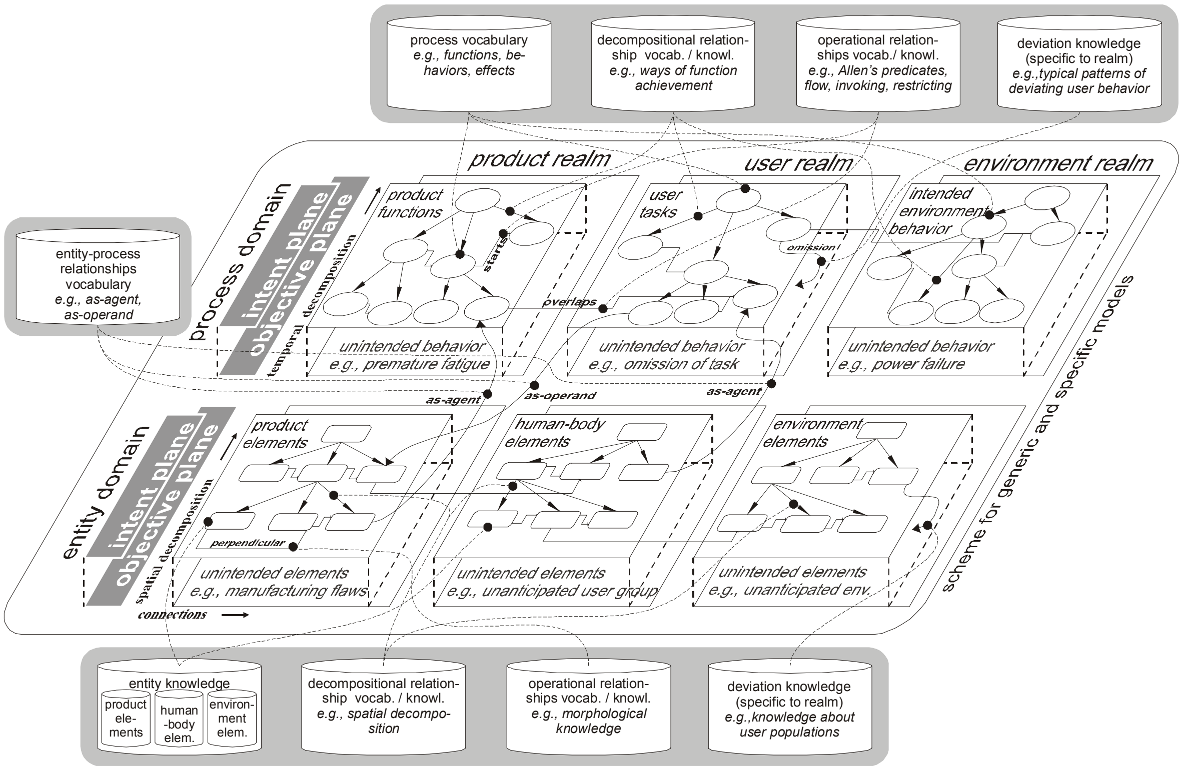

Figure 5. *An ontology-based scheme for models of product, user and environment.*

parts) of contents of the models and relationships among them. The structure of the models shows two analogies: one between product, user and environment, and one between process and entity. A model for a specific product is described in the common structure based on the similarities. In the gray planes, the figure shows categories of concepts in the models as ontologies, kinds of generic knowledge, and relationships between the models and the ontologies. One of the utilities of ontologies is to give the model author a consistent viewpoint for capturing the target world by providing vocabulary in the models. This was discussed in the first-part paper.

A model for a product in a use context consists of the six parts shown in the central area of the figure. Horizontally, it is subdivided over three realms; *the product realm, the user realm,* and *the environment realm*. Vertically, a model of each realm is divided over two domains; *the process domain* and *the entity domain*. Roughly speaking, the former is the temporal domain, while the latter is the spatial domain. Each domain of a realm consists of two *planes: the intent plane* which includes ones intended by the designer and *the objective plane* which includes unintended (alternative) ones as well as the intended ones. On each plane, there are two major categories of relationship among elements, that is, the decompositional (whole-part) relations and the operational relations.

The gray planes are ontology and generic-knowledge layers. They are generic and independent of the target product and technical domains. They include generic concepts which can be used as a vocabulary in the models and generic knowledge which can be used as building blocks for the models. The three realms, product, user and environment, have the same modeling structure, which are two domains consisting of two planes with two kinds of relationships. They also share many of the generic concepts and generic knowledge on the ontology layers. In the following paragraphs, we only explain the product realm and the process domain of the user realm.

The process domain of the product realm (the upper left planes) represents behaviors of the product on the objective plane and the functions (intended behaviors) on the intent plane. Functions and behaviors in the model are instances of generic concepts in the ontology layer as shown as the 'process vocabulary' in the figure. The decompositional relations between functions (or behaviors) here represent achievement relations between a macro function and a sequence of micro(sub)-functions. Typically, they are instances of generic knowledge about the way of function achievement as discussed in the first-part paper. The operational relations here represent temporal or causal relations. The objective plane represents behaviors, including unintended behaviors and/or faulty behaviors.

The process domain of the user realm (the center-upper planes) represents user actions, which can be represented using generic concepts for product behaviors, or functions, as discussed in the previous section. The intent plane includes user actions intended by the designer called user tasks, while the objective plane includes alternative user actions as well. The alternative user actions can be generated through the typical deviation patterns discussed in

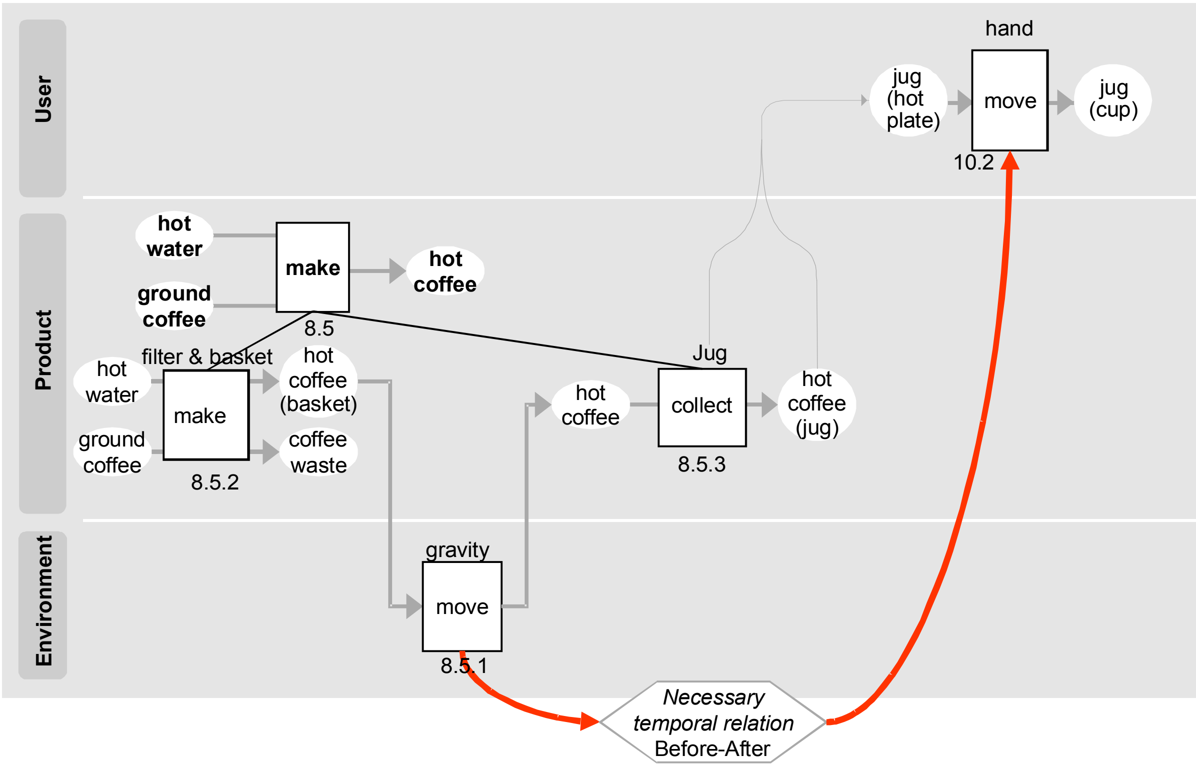

Figure 6. *Part of a use process with a necessary temporal relation that has to be observed by the user.*

the previous section, such as 'omission of an action'. We can also consider such deviation knowledge for other realms/domains as shown by examples in figure 5.

The entity domain of the product realm (the bottom left planes) represents elements (physical things or entities) of the product on the intent plane. The product elements correspond to device, system, subsystem and components, among which there are spatial decompositional relations. The elements have several properties, such as dimension and material. Examples of the operational relations here are morphological relations about contacts (connections) and relative positions between components. We can build a kind of library of the product elements (components) and generic categories of the relations as an ontology layer shown in the bottom plane.

Between the process domain and the entity domain, there are role-assignment relations such as 'as-agent' and 'as-operand'. For example, an 'as-agent' relation represents that a product element performs a function as an agent, while an 'as-operand' relation represents that the element is affected as operand by a user action. Such relations are defined in the ontology for the vocabulary of entity-process relations.

6. DISCUSSION: FEASIBILITY OF DESIGN SUPPORT FOR UNINTENDED PRODUCT USE

The explorative work we discussed in the previous subsections focused on how to capture other-than-product behavior and other-than-intended *in a model* as an extension of the knowledge models applied in FBRL. Only models are of little help for designers. In this subsection we discuss how a future design support system can possibly employ these knowledge models to the benefit of product designers. We will do this based on a process-deviation example concerning unintended coffee-maker use.

The partial model in figure 6 shows a selected set of product functions with intended environment and user behavior, which are connected through a temporal relationship that is intended by the designer. After all the brewed coffee has been transferred to the jug, the user can take out the jug to start coffee consumption. If the user violates the intended order by removing the jug before all of the coffee has been collected, the jug is no longer present for its function of collecting the remaining newly-brewed coffee (figure 7). Gravity to move the coffee downward is still present, so the coffee will end up somewhere else. Common knowledge about coffee makers tells us that the coffee will land on the hot plate, where it might leak to the product interior, possibly causing short-circuit between live wires. This is most probably an undesired situation that the designer of a new coffee maker should be aware of.

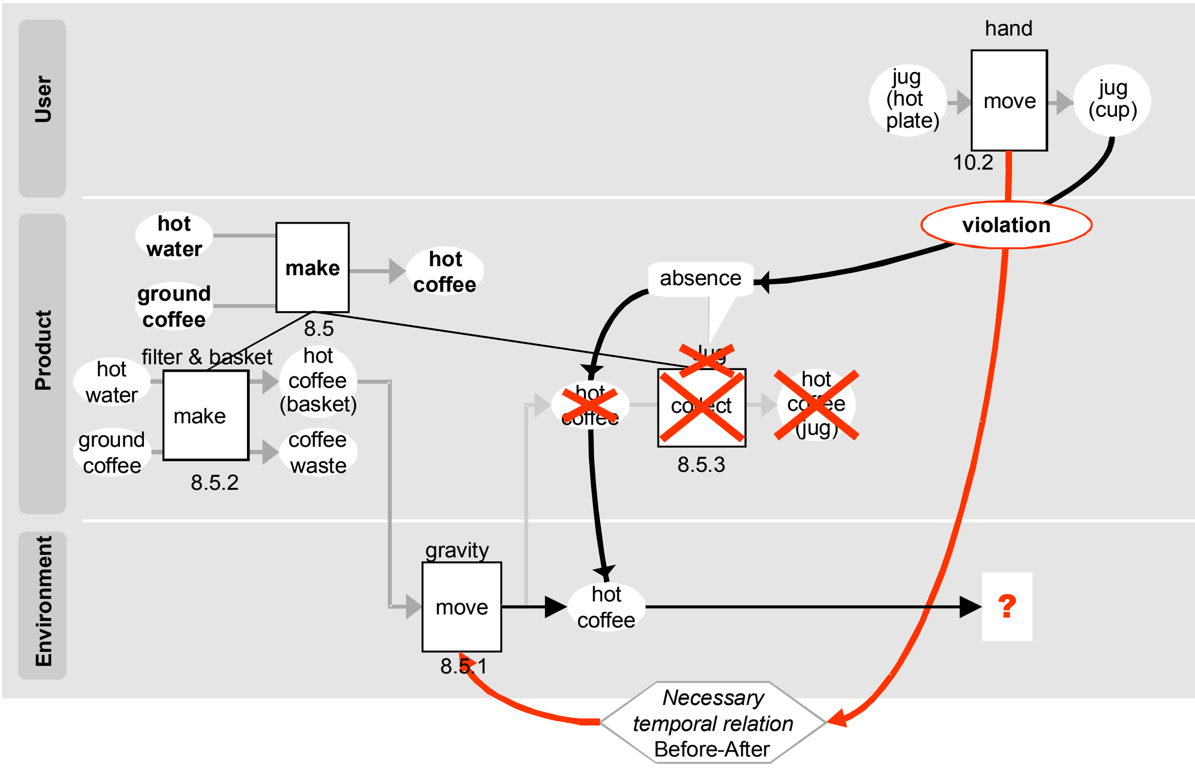

Figure 7. *Violation of the intended temporal order of user tasks.*

How can a knowledge-intensive system based on a use-oriented extension of FBRL support the designer? Based on the knowledge content discussed in sub-section 4, we identified four subsequent areas for concrete support that can be considered for inclusion for which we will briefly discuss the feasibility of actual computer support, as well as the applicability of the scheme in figure 5, in the following subsections. These are: (1) finding possible forms of unintended use, (2) predicting effects of unintended use, (3) evaluating the severity of the effects of unintended use and (4) generating solutions to deal with unintended use.

6.1. Finding possible forms of unintended use

The simplest patterns of typical unintended behavior can relatively easily be generated by a computer system. Typical patterns are included as deviation knowledge in the ontology (figure 5). If, like in our current example, the user carries out one particular task too early, or omits just one task, the violation can be considered 'simple'. More complex violations of the temporal order are likely to include multiple violations, and the need to assess all permutations of user-task sequences, or even the prediction of intermediate effects (see next item in this list). Such an assessment would be a considerable computational challenge. Yet, not every possible combination of typical unintended behaviors is meaningful. Exclusion mechanisms could help to reduce the number of possible use patterns. For instance, it is not possible for the user to move an operand from A to B if this operand is not available at A. A system that keeps track of the states produced by previous actions and functions should be able to exclude use patterns that include such actions. Another practical observation is, that in many cases, things appear to 'go wrong' from the first violation of the intended sequence. In such cases, the subsequent permutations of other actions do not have to be considered and can be excluded beforehand. Thus, generation of 'straightforward' violations may already help the designer. Such an approach is very similar to the generation of failure modes in computer-aided FMEA [5]. For the more complex forms of unintended use, including atypical user behavior, the deviation knowledge in the ontology could be supplied with concepts of unintended use from company experience, historical data, user-panel testing results, etc. The typical unintended use pattern of modifying the decomposition of tasks into subtasks is also present as deviation knowledge. To find particular forms of 'decompositional' deviation, a setup similar to the 'ways' in functional FBRL can perhaps be applied, to generate forms of unintended use. For instance, two 'ways' how the user can fill the reservoir of the coffee maker with water are (a) to fill it with tap water from the jug and (b) to carry the coffee machine to the tap to fill it directly.

6.2. Predicting the effects of unintended use

Prediction of effects is expected to be more difficult to realize. As it was already indicated in figure 1, the system could possibly interface with simulation systems to make predictions possible. Returning to our example in figure 7, it is not likely that current numerical simulation tools (finite elements, bond

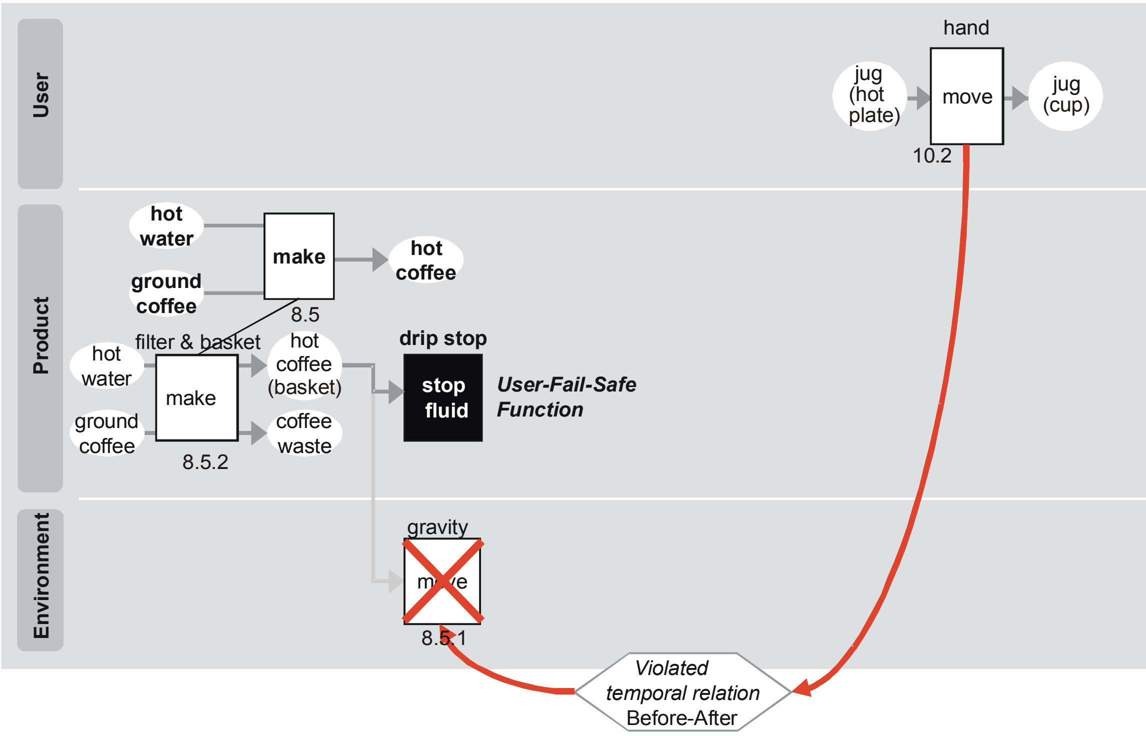

Figure 8. *A possible solution to deal with the unintended behavior.*

graphs, etc.) can predict the occurrence of coffee leaking through the hot plate and causing short-circuit. Such real-life behaviors involve multiple domains of physics – in this case fluid dynamics, thermodynamics and electricity. Perhaps the ongoing development of multiple-physics simulations will open more possibilities to handle such complex behavior in the near future [12]. Alternatively, the failure behavior in the example could be predicted qualitatively, based on spatial and temporal reasoning *(if the jug is absent, the hot plate is the first component to be reached by the coffee)* combined with company knowledge *(liquids in the immediate neighborhood of live wires can cause short circuit)*. The development of any of these prediction methods is a major research project in itself. Thus, we will assume for now that the designer has sufficient insight in the possible effects of a particular form of unintended use, to decide whether it should be dealt with in a redesigned product.

6.3. Evaluating the effects

Some forms of unintended use and their effects do not have to be considered a problem that is to be solved in a redesign. In other cases, the unintended use is likely to occur too frequently, the effects can be harmful and/or irrevocable. To assist in the decision making involved here, *risk-priority numbers* (RPN) as used in FMEA (e.g. [13]) or similar techniques may be useful. Although RPN calculations can be performed by a computer, the input is based on a human assessment – e.g. based on a particular failure mode, it has to be determined if the user is likely to be 'discomforted', 'dissatisfied' or 'very dissatisfied'. Even proposed setups for computer-aided FMEA do not include computer support for this decision-making process. Therefore, for now, we will not further elaborate on this potential area of computer support.

6.4. Generating solutions

Figure 8 shows the added functionality of a familiar solution dealing with the problems caused by early jug removal. It is the 'drip stop' of which several varieties can be found in coffee-makers, since this type of unintended use was recognized in the 1970s. Obviously, this is not the only possible answer to the problem. From the viewpoint of design strategy, it is a *corrective* remedy, in that it handles the possible harmful effects 'where things go wrong', i.e., it stops gravity from moving the coffee downward[3]. Another corrective solution could be to provide an alternative collector for coffee if the jug is missing. Conversely, *preventive* solutions to restrain the user from early jug removal, are also possible. And more radical solutions can be found by finding *evasive* remedies. Such remedies present completely different ways for the user to obtain the coffee from the coffee maker, in which no jug has to be removed at all – for instance by replacing the jug by a second reservoir with a coffee tap. This type of coffee maker is actually on the market for professional use. Computer support at the strategic level of deciding for corrective, preventive or evasive would probably be complicated without considering the further consequences for the design, i.e. at a lower level of abstraction. Three levels of abstraction can be distinguished after the strategic level.

---

[3] Note that 'corrective' refers to correcting the effects of user or environment actions, not to correct design flaws. In other literature, 'corrective' is sometimes used in the latter context, where corrective redesign includes preventive solutions [7].

In the first place, the system could indicate which actions, functions – or effects thereof – can be targeted for preventive and corrective remedies. Typically, for preventive and evasive remedies, this is the unintended user action and, in case of corrective remedies, it is the unwanted effect, or one or more actions in the chain leading to it.

In the second place, the system could provide a function description for the added preventive or corrective product behavior. The system has to find a function that can change the undesirable effect, e.g. the presence of coffee on the hot plate into a target state that is *not* undesirable. This target state is usually not specified concretely, thus, to start with, it can be any state other than the undesirable state. In the regular FBRL-based framework, function is a teleological interpretation of changes between two states known as input and output. Using generic definitions of functional concepts ('process vocabulary' in figure 5), the system could suggest functions to achieve the negative or opposite state. For example, the non-presence (i.e. absence) of coffee at the output port can be achieved by a function 'to stop fluid' to be applied to coffee at its input port, making it impossible for gravity to perform the function 'to move'. Other alternatives would be 'to absorb fluid' or 'to vaporize fluid'. Pairs of undesirable states and 'negative' functions can be stored as a chunk in the same form as the 'way of function achievement' ('knowledge layer' or areas with gray background in figure 5).

In the third place, the system could suggest function fulfillers that manifest the behavior in question (in case of a drip stop, e.g., a valve, or more specifically a spring-operated valve). This could be achieved through a hierarchy of more specific ways of achievement and/or entity knowledge.

7. CONCLUSIONS AND FUTURE WORK

With some adaptations, the current FBRL technique can be used to represent use processes, including unintended behavior, with building blocks that can be arranged in a model that is similar to a regular FBRL model of product functions (figure 5). Some more substantial extensions, such as explicit representation of temporal relationships and roles, will facilitate the applicability in a design-support system. If we want to support the anticipation of the more complicated forms of unintended behavior, including atypical forms of use, we need to find ways of capturing and managing diverse forms of knowledge, such as results from user observations and simulations, company experience and perhaps even cognitive human behavior, and to include such knowledge in the ontology.

Our subsequent research will focus on the realization of the needed extensions in FBRL and on systematization of the diverse forms of knowledge that can be used to improve design support, so that other-than-product behavior and other-than-intended behavior can be anticipated more effectively. Tools for generating alternative behavior and for solutions to compensate for unintended behavior are planned for the medium term, whereas connectivity with quantitative design support is expected to be a long-term issue.

Acknowledgements: This research has been carried out as a collaborative research supported by the Japan Society for the Promotion of Science (JSPS Fellowship ID L02536) and by Delft University of Technology. The authors wish to thank Mariko Yoshikawa for her contributions.

References

[1] Sasajima, M., Kitamura, Y., Ikeda M., Mizoguchi, R., *A representation language for behavior and function – FBRL*, Expert Systems with Applications, Vol.10, No.3/4, 1995, pp.471-479.

[2] Mizoguchi, R., Kitamura, Y., *Foundation of knowledge systemization: role of ontological engineering*, in: Roy, R. (Ed.): Industrial knowledge management - a micro-level approach. London: Springer, 2001, pp.17-36.

[3] Van der Vegte, W.F., Horváth, I., *Consideration and modeling of use processes in computer-aided conceptual design*, Transactions of the SDPS - Journal of Integrated Design and Process Science, Vol. 6, No. 2, 202, pp. 25-59.

[4] Kmenta, S., Ishii, K., *Advanced FMEA using meta behavior modeling for concurrent design of products and controls*, proc. of 1998 ASME DETC, Atlanta, 1998 (CD-ROM).

[5] Hata, T., Kobayashi, N., Kimura, F., Suzuki, H, *Representation of functional relations among parts and its application to product failure reasoning*, Proc. of International CIRP seminar on Design with Manufacturing, Haifa, Israel, 2000.

[6] Lee, B.H., Using Bayes belief networks in industrial FMEA modeling and analysis, proc. of Annual Reliability and Maintainability Symposium, Philadelphia, 2001.

[7] Goel, A., Chandrasekaran, B., *Functional representation of designs and redesign problem solving,* proc. of IJCAI-89, Detroit, Michigan, 1989.

[8] Buur, J., Windum, J., Jakobsen, M.M., *Man/machine interface design needs systematic methods*, proc. of ICED, Zurich, 1991.

[9] Otto, K.N, Wood, K.L., *Customer integrated systematic design, Transactions of the SDPS*, Journal of integrated design and process science, vol. 3, no. 4, 1999, pp. 61-74.

[10] Suto, H., Kawakami, H., Katai, O., Horiuchi, T., *Hierarchical modeling of artifacts based on interaction with humans and the environment*, proc. of EDA 2000 Conference, Orlando, Florida, 2000.

[11] Allen, J.F., *Maintaining knowledge about temporal intervals*, Communications of the ACM, vol. 26, No, 12, 1983, pp. 832-843

[12] Mahoney, D.P., *Multiphysics analysis*, Computer Graphics World, 23 (6) , 2000, pp. 44-46, 50, 52.

[13] Franscheschini, F., Galetto, M., *A new approach for evaluation of risk priorities of failure modes*, International Journal of Product Research, Vol. 39, No. 13, 2001, pp. 2991-3002.