Proceedings of CIE 2004:
ASME 2004 Design Engineering Technical Conferences
and Computers in Engineering Conference
September 28-October 2, 2004, Salt Lake City, Utah

DETC2004-57720

COPING WITH UNINTENDED BEHAVIOR OF USERS AND PRODUCTS: ONTOLOGICAL MODELLING OF PRODUCT FUNCTIONALITY AND USE

Wilfred van der Vegte
Faculty of Industrial Design Engineering,
Delft University of Technology,
Landbergstraat 15, NL-2628 CE Delft
The Netherlands
w.f.vandervegte@io.tudelft.nl

Yoshinobu Kitamura
Institute of Scientific and Industrial Research,
Osaka University,
8-1 Mihogaoka, Ibaraki, Osaka,
567 -0047 Japan
kita@ei.sanken.osaka-u.ac.jp

Yusuke Koji
Institute of Scientific and Industrial Research,
Osaka University,
8-1 Mihogaoka, Ibaraki, Osaka,
567 -0047 Japan
koji@ei.sanken.osaka-u.ac.jp

Riichiro Mizoguchi
Institute of Scientific and Industrial Research,
Osaka University,
8-1 Mihogaoka, Ibaraki, Osaka,
567 -0047 Japan
miz@ei.sanken.osaka-u.ac.jp

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 product-use context by 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. In this paper, an ontological scheme is presented for models of the product, the user and the environment and related use processes. We present an example and discuss how supporting tools can help designers to deal with unintended use. In the example case, a modeling schema for unintended behaviors of products is extended towards unintended behaviors of users.

KEYWORDS

use process modeling, functional modeling, CAD, conceptual design, ontologies, unintended behavior

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 Computer-aided Design Engineering (CADE) 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 the common activities of functional modeling, artifact modeling and artifact-behavior modeling. Functional modeling is typically carried out in the early stages of design, where the behavior of a product according to the designer's intention is projected. In this context, *function* is the *intended behavior* of the artifact. Typically, artifact modeling concerns the geometry, morphology, structure and material of the artifact. Artifact-behavior modeling is performed to gain insight into the actual behavior that can be observed when the physical artifact is put into operation. This observable behavior (for short 'behavior') can involve not only intended but also *unintended behavior*. Often, the designer will have to deal with this unintended behavior by modifying the original artifact design, but it has to be realized that not all unintended behavior is unacceptable. In considering the *use* of the product, we do not only have to take

 Copyright © 2004 by ASME

the behavior of the product, but also that of the user and the environment of the product into account [1]. Thus, the behavior of the product is put into context with the elements with which it interacts, and which, unlike the product, are outside the direct scope of the designer's influence. Usually, the description of the functions (or intended behavior) of the product is based on assumptions about behavior of the user and the environment, i.e., intended user and environment behavior. Again, the user and the environment can act differently from what the designer intended. Hence, we will also have to take unintended behavior of the user and the environment into account.

This paper considers the issue of dealing with the range of intended and unintended behaviors in the context of the increasing deployment of knowledge-intensive systems in computer support of design, which we consider as an opportunity to facilitate the designer's work. In the research field of knowledge representation and knowledge processing, the application of ontologies has proven to be advantageous over recent years. An ontology is defined as 'a specification of a conceptualization' of the target world [2]. It consists of a system of concepts (i.e., definitions of concepts and relations among them) for describing a model of the target world. One of the roles of an ontology is as 'building blocks for models', that is, to provide controlled vocabulary and semantic constraints as a conceptual schema for models. An ontology can restrict contents of knowledge and contributes to easy authoring. Another role for inter-agents communication is to provide a shared (or common) vocabulary for different knowledge schemas, which enables agents to share knowledge and reuse it. This research focuses on the former role of ontologies as building blocks or conceptual schema.

In the research at hand, the CADE group seeks to apply ontology-based models to represent so-called design concepts that offer the product designer integrated support for artifact modeling and artifact behavior modeling, 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 [3]. FBRL is an abbreviation of Function and Behavior Representation Language, which is based on two ontologies for capturing functionalities of artifact: the extended device ontology [5] and the functional concept ontology [4]. The former provides a device-centered common viewpoint for capturing the target world in different domains including fluid-related plants, mechanical systems, and manufacturing systems [5]. The latter provides a controlled and well-defined generic vocabulary for representing functions of components [4]. The Mizoguchi Lab has established a modeling framework based on the ontologies for functional knowledge, and has successfully applied it in various application domains, such as engineering plants, mechanical systems and manufacturing processes [4, 5].

To explore the possibilities of applying the FBRL-based ontologies to product-use processes, cooperation between Delft University of Technology and Osaka University was started in 2002. The generality of the ontologies is expected to be useful as a common conceptual schema for capturing both product functionality and use process clearly and easily.

This paper reports on an explorative study into the extension of FBRL modeling towards the inclusion of (mainly qualitative) aspects of unintended behavior and product use. It covers the following research items:

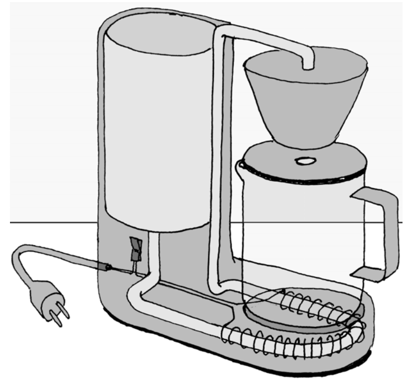


Figure 1. Schematic design of a simple coffee maker

- 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.
- 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 does not necessarily reflect a chronological order in the research activities. To illustrate the terminology and content of models, we will use a simple coffee maker as an example throughout this paper: Figure 1. For now, our research focuses on products of a similar simplicity. Possibly, more complex products – such as cars or photocopiers – will have to be decomposed before they can be dealt with. Such issues are not discussed in this paper.

1. OBJECTIVES AND ARCHITECTURE FOR A DESIGN-SUPPORT SYSTEM

The objectives of current functional-ontology modeling are (1) to provide insight into the rationale why designers applied particular design solutions by making the intended behavior explicit and (2) to provide computer-generated suggestions for alternatives based on the given functions in a product [4,5].

Extension of functional-ontology modeling towards use process modeling implies that we have to consider not only knowledge about the intended behavior (function) of the product, but *possible* (i.e., both intended and unintended) behavior of the product, the user and the environment. In our coffee maker, the coffee maker itself is the product, the human operating it is the user and all the involved non-product elements form the environment: the air that surrounds it, the water, the disposable filter, the coffee, the power socket, etc. Potentially, support for the designer can include activities concerning (1) defining intended use (including product functionality); (2) finding unintended forms of use; (3) predicting the effects of use, (4) evaluating the effects of use and (5) generating solutions for undesirable effects. Figure 2 shows a possible architecture for a modular system that would cover all of these forms of support. In its most extended form, this setup features:

- An 'enhanced function-behavior modeler' that allows the designer to create (1) a functional model of the product to describe its intended behavior (for the coffee maker this includes, for instance, heating up the water), (2) an intended-behavior model that extends the model of product func-

Copyright © 2004 by ASME

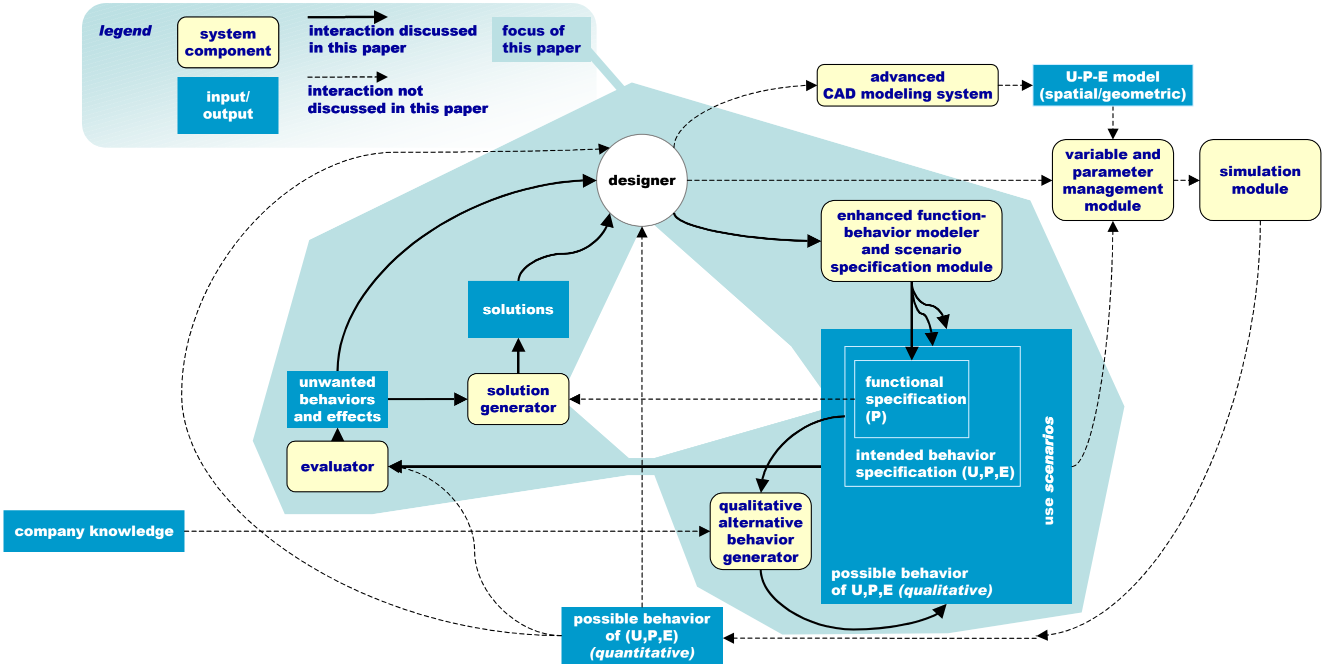


Figure 2. Information flow of working with a design-support system featuring ontology-based modeling of the use process of a product.

tions with intended behavior of the user and the environment (e.g., user operations as described in the user manual; the power socket providing the default voltage) and (3) a possible-behavior model of the product, the user and the environment to allow the designer to include forms of unintended behavior that he already anticipates (e.g., the user forgets to connect the coffee maker to the power socket);

- An alternative-behavior generator, that uses information from intended behavior, simulation systems and possibly also company knowledge to generate (forecasts of) additional unintended behavior. This information and knowledge can originate from various sources, such as simulations, insight gained from previous products, or data collected from interactive user participation sessions. Here, we will have to emphasize that practically, it will not be realistic to capture and manage knowledge about *all* possible use processes, let alone that this can be done automatically, but we do not strive to exclude any particular use at this stage;
- Simulation facilities to predict effects of use based on variables and parameters derived from (1) CAD-based (artifact) models of the product, the user and the environment and (2) the aforementioned models that describe possible forms of intended and unintended use. This may require that – like the specification of the user, the product and the environment in a CAD model – the forms of use that we want to include are also available in a quantitative form, i.e., scenarios that describe the timing and physical interventions in terms of forces, locations, displacements, etc. As an example, a scenario could specify when the user switches on the coffee maker and a simulation calculates when the coffee is ready;
- An evaluation module to assess the risk of possible behaviors and to help the designer decide whether the effects of behaviors are acceptable. For instance, if the user forgets to connect the coffee maker to the power socket, this can be considered annoying but not dangerous. Not only unintended behavior but also intended behavior may give rise to effects that call for a redesign, for instance when the designer's first estimate of the heating-element power is insufficient to attain the required water temperature.
- A solution generator, that can be considered to be an extension of FBRL's current ability to provide alternatives based on the given functions in a product. The extension would have to deal with other-than-product and other-than-intended behavior.

The collaborative work discussed in this paper focuses on the components in the 'focus' area (shaded background) that are connected by solid arrows; in other words, the extension of functional modeling towards modeling the possible behavior of the product, the user and the environment without considering quantitative aspects. The extension is based on an ontology-based scheme for the content of models that is introduced in section 3. The feasibility of further extension outside the 'focus' area in Figure 2 has not been investigated. These elements are only included as an indication of possible future extensions, with the exception of *quantitative simulation* with conceptual CAD-based user-product-environment models (U-P-E models), which is a current focus of research activities in Delft. The U-P-E models in this other line of research are based on the 'nucleus' concept that incorporates the laws of physics through built-in equation-based relations [6]. The nucleus is also considered to be the lowest-level model-building block that can be characterized in terms of function. Forthcoming collaborative work will focus on the integration of qualitative and quantitative aspects through the opportunities offered by the nucleus concept.

2. RELATED WORK

To some extent, other research dealing with computer support for considering unintended behavior and/or behavior of

Copyright © 2004 by ASME

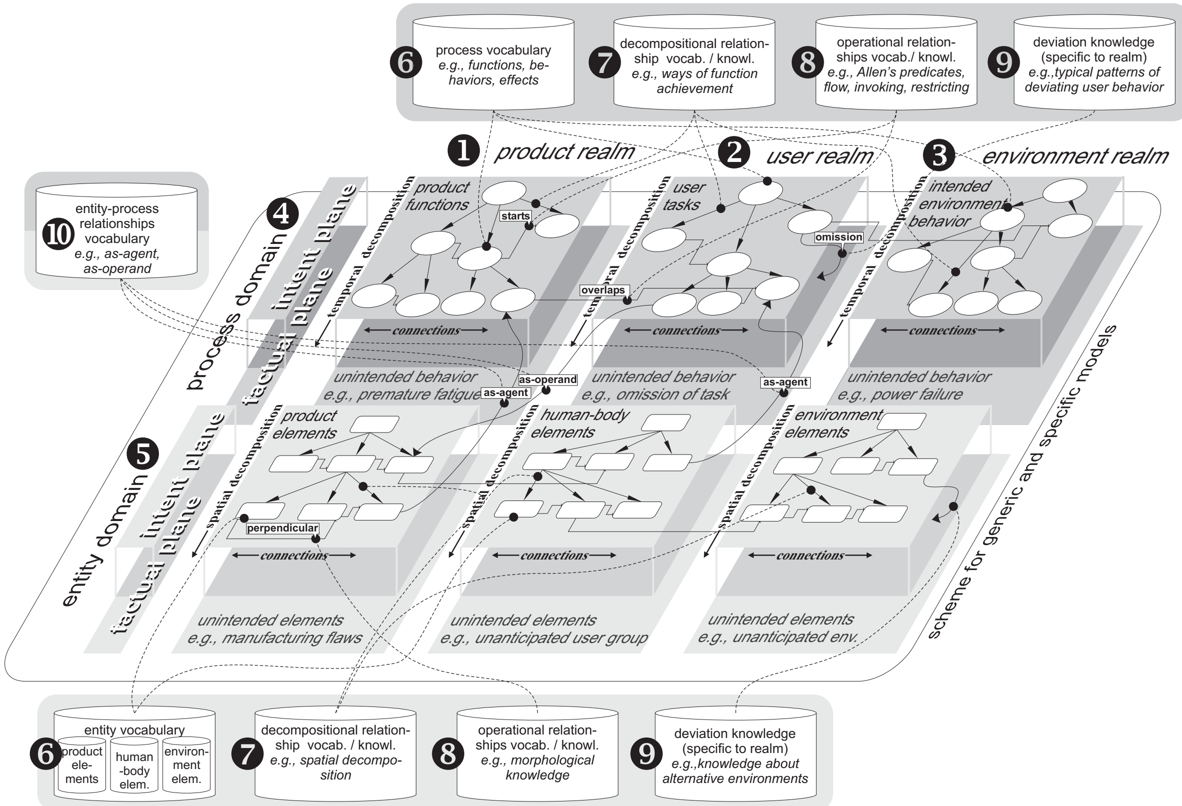


Figure 3. An ontology-based scheme for models of product, user and environment

users and the environment together with product behavior, can be considered related to our work; the extensions outside the 'focus' area in Figure 2 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 by Lee focuses on unintended behavior (failure, in particular), but it tends to concentrate on internal behavior of the product [7-9]. 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 that is presented by Goel & Chandrasekaran [10]. Furthermore, some modeling techniques have been proposed for processes that include both user actions and product functions [11-14]. 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.

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

Figure 3 shows an overview of a scheme for models of product, user and environment. This scheme represents a general structure of the models (the 'focus' part of Figure 2), i.e., major categories (sub-parts) of contents of the models and relations among them. The structure of the models shows two analogies: one between product, user and environment, and one between process and entity. Based on the analogies, products, users and environments can be structured in similar ways, using common relationships. The same similarities and common relationships apply to processes in time on the on hand and entities in space on the other hand. Examples of common relationships are decompositional relationships between larger units and their constituents, and mutual operational relations (e.g., input-output) between constituents. These analogies form the foundation for our efforts to extend FBRL towards non-product behavior and unintended behavior [15].

In the grey planes, the figure shows categories of concepts in the models as ontologies, kinds of generic knowledge, and relations 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 [4, 5].

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* (❶)*, the user realm* (❷)*,* and *the environment realm* (❸). Vertically, a model of each realm is divided over two domains; *the process domain* (temporal) (❹) and *the entity domain* (spatial) (❺). Each domain of a realm consists of two *planes*: *the intent plane*, which includes items intended by the designer and *the factual plane,* which includes unintended (alternative) items as well as the intended ones. On each plane, there are two major categories of relations among elements, that is, the decompositional (whole-part) relations and the operational relations or connections.

The planes depicted at the circumference of the scheme are ontology and generic-knowledge layers. They are generic and independent of the target product and technical domains. They include generic concepts that can be used as a vocabulary in the models and generic knowledge that can be used as building blocks for the models. We distinguish two ontologies providing typical building blocks for processes and entities, respectively (❻); ontologies providing decomposition knowledge (❼); op-

 Copyright © 2004 by ASME

erational relation knowledge (❽); deviation knowledge (❾); and knowledge about entity-process relationships (❿). In the following paragraphs we will illustrate the content of the scheme by elaborating the product realm and the process domain of the user realm.

The process domain of the product realm (top left) represents the possible behaviors of the product on the factual plane and its subset of intended behaviors related to the functions, are also represented on the intent plane. In the ontology layer in the figure, functions and behaviors in the model are instances of generic concepts that are expressed using a ‘process vocabulary’ (❻) that provides, for instance, the verbs to express functions such as ‘to give heat’. Such verbs for functions (we call them functional concepts) are operationally defined using the FBRL language as the functional concept ontology [4]. For example, a functional concept ‘to give heat’ is defined as ‘an energy flow between two mediums’ with ‘teleological focus on the destination (heat-receiving) medium of the transfer’. The former part of the definition specifies a minimum constraint on *objective behaviors*. On the other hand, the latter part specifies the *teleological interpretation* of the behavior under a goal. With such definitions, a function covers not only selected behaviors but also result of interpretation of the behaviors. Such definition of a functional concept is independent of how to realize the function, specific type of medium and function carriers (agents of function[1]). Thus, it can be mapped to many behaviors and function carriers (devices or humans as discussed in the next section) in several domains. The function in the model is a result of instantiation of such a generic concept. The functional concept ontology provides a controlled vocabulary for functions and thus contributes to generality of the models. The generic definitions are expanded for user-process modeling in the next section.

The decompositional relations between functions (or behaviors) here represent how a higher-level function can be achieved through sub-functions in a similar way to functional decomposition in [16]. As discussed in [4], such relations in this schema are instantiated from generic knowledge about the way of function achievement (❼). Discriminating function (what to achieve) from the way of function achievement (how to achieve) plays a crucial role in systematic description of functional knowledge [4], There are also operational relations to represent causal relations and temporal arrangements based on Allen’s interval logic [17] (see section 4.1, Figure 5) from operational relation knowledge (❽). The factual plane represents possible behaviors including unintended behaviors of the product, such as fatigue, derived with the help of deviation knowledge (❾). An ontological modeling schema for this factual plane of the process domain of the product realm has been elaborated in [18]. In section 5 of this paper, the same modeling schema is extended to other realms.

The process domain of the user realm (top centre) represents user actions, which can be represented using the same generic concepts that are used for product behaviors, or functions. The intent plane includes user tasks intended by the designer, while the factual plane also includes alternative user actions from deviation knowledge (❾). One approach to generate alternative user actions is through typical deviation patterns, such as ‘omission of an action’. This will be further elaborated in section 4, together with consideration of deviation knowledge for other realms/domains.

The entity domain of the product realm (bottom left) represents elements (physical things or entities) of the product on its intent plane. For the product elements, the levels of system, assembly, component and nucleus [6] can be distinguished through decompositional relations. The elements have properties such as dimension and material. Examples of the operational relations here are morphological relations to specify the interfaces between components (the wall of the reservoir is perpendicular to the bottom), and relative positions between components. A library of product elements (components) and generic categories of the relations can be realized in an ontology layer as is shown at the bottom of Figure 3 (❻).

Between the process domain and the entity domain, there are role-assignment relations described with vocabulary provided by ❿, 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.

4. EXTENSION OF FBRL TO USE PROCESS MODELING AND ITS APPLICATION IN DESIGN

In this section, we will concretize the design support that can be provided for the activities mentioned in the workflow that we introduced in section 1: (1) defining intended use (including product functionality); (2) finding unintended forms of use; (3) predicting the effects of use, (4) evaluating the effects of use and (5) generating solutions for undesirable effects. A simple coffee maker (Figure 1) will serve as an example to demonstrate the support that can be offered by system components presented in section 1. As we will indicate, some of the support can be provided by system components that already have been developed, whereas other forms will depend on future developments. The designer’s own reasoning and creativity can typically bridge such gaps in the coverage of computer support. Both the available and the future forms of support vary in level of automation, ranging from merely helping the designer to organize his ideas to aiming at computerized execution of design activities. In the following paragraphs we will indicate which level of automation applies to the presented form of support.

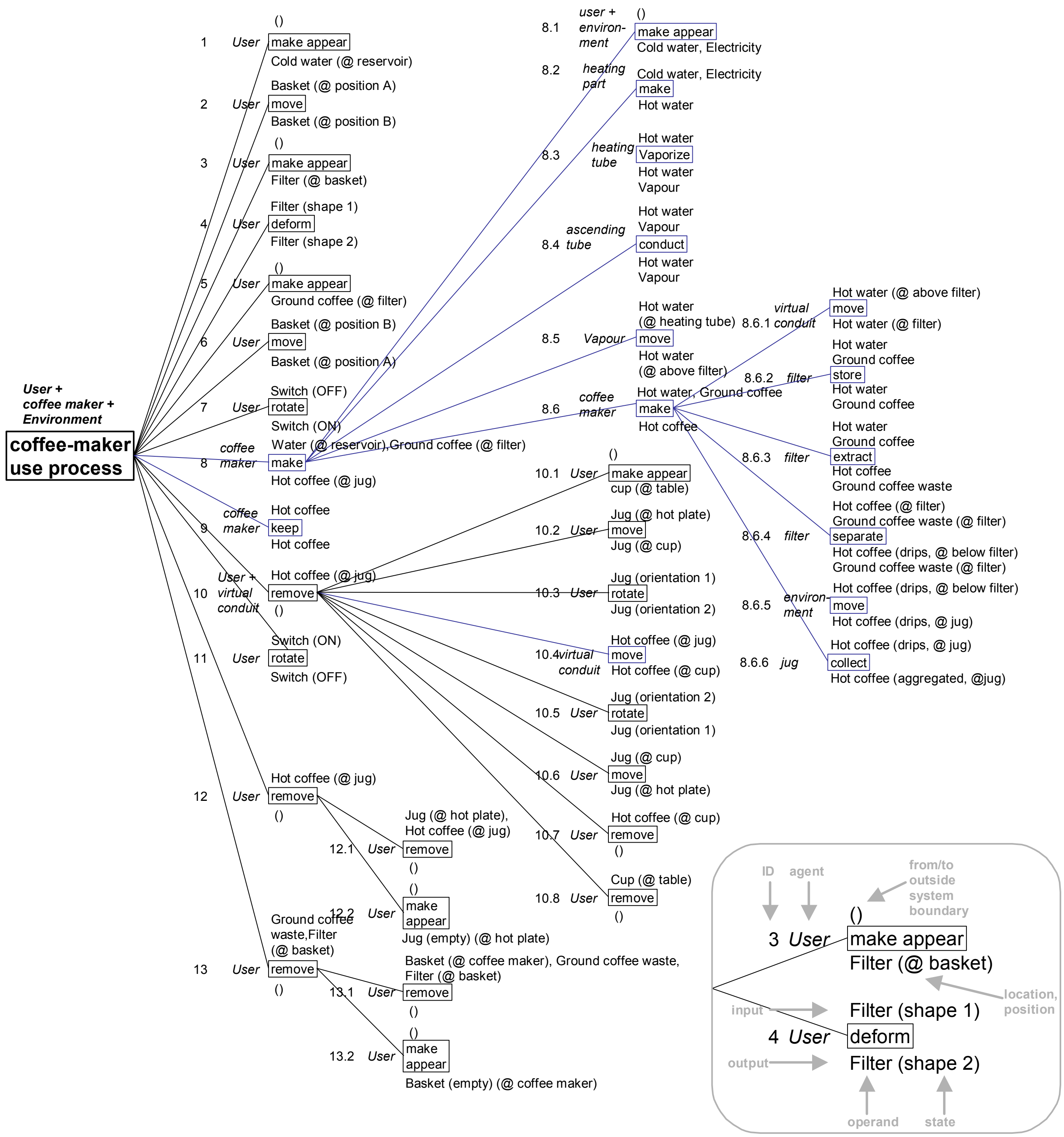
4.1. Defining intended use

Our starting point for modeling use processes is an FBRL-based functional model of the product. Such a model defines intended operation processes, i.e., processes in which the product operates autonomously – without the intervention of users – and according to the designer’s expectations.

The first step of extension towards a use process is to further specify the intended use, in other words, the intended behavior of the user and the environment that the designer expects to take place with the intended product behavior, or the ‘idealized’ use process as it can be found in a user manual. Based on the similarities and analogies elaborated in the scheme in Figure 3, we found that relatively little alteration was needed to apply the building blocks for functional models to models that also include other-than-product behavior [15]. Figure 4 shows a functional model of a coffee maker that was extended with in-

[1] 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.

Copyright © 2004 by ASME



(Note: To restrict the size of the illustration, not all items have been fully decomposed.)

Figure 4. FBRL-based model of a use process, including intended behavior of the user and the environment.

tended behavior of the user and the environment of the product. Thus, it not only covers behavior of which the product is the agent but also behavior of which the user or the environment is the agent, as a combination of the three intent planes of the process domain in Figure 3. Like a product function, such behavior can be decomposed into discrete entities. Together with product functions, user tasks and the expected behavior of the environment they constitute the intended use process. The model in Figure 4 was created using the basic principles of functional-ontology modeling, using function-describing terms from FBRL for user tasks and intended behavior of the environment. In three respects, the existing FBRL had to be extended in order to capture the knowledge and the relations concerning intended use in a process vocabulary corresponding to item ❻ at the top of Figure 3:

1. A minor extension of the function-describing vocabulary is needed to include human actions. Particular task-describing terms that have to be introduced concern human manipulation of objects in space and 'invoking' tasks to start and terminate functions of the product (switching on, switching

 Copyright © 2004 by ASME

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

The default intended relations are:

between	*m*, *m*+1	: *m* BEFORE *m* +1
	m.n, *m*.(*n*+1)	: *m.n* BEFORE *m*.(*n*+1)
	$m.n_{max}$, *m*	: $m.n_{max}$ FINISH *m*
	m, *m*.1	: *m* START *m.n* for *n*=1

Aberrations from default relations (referring to numbering in Figure 4)

Default relation to be disconnected:

1,2: **1 BEFORE 2**

Additional intended relations:

1,7: **1 BEFORE 7**

8,9: **8 STARTS 9**

8.2,8.3: **8.2 CONTAINS 8.3**
8.3,8.4: **8.3 CONTAINS 8.4**
8.4,8.5: **8.4 STARTS 8.5**
8.5,8.6: **8.5 CONTAINS 8.6**
8.6.1,8.6.2: **8.6.1 OVERLAPS 8.6.2**
8.6.2,8.6.3: **8.6.2 OVERLAPS 8.6.3**
8.6.3,8.6.4: **8.6.3 OVERLAPS 8.6.4**
8.6.4,8.6.5: **8.6.4 OVERLAPS 8.6.5**
8.6.5,8.6.6: **8.6.5 OVERLAPS 8.6.6**

8,9: **8 OVERLAPS 9**

9,10: **9 OVERLAPS 10**

8,10.2: **8 BEFORE 10.2**

9,11: **9 FINISHED-BY 11**

10.3,10.4: **10.3 OVERLAPS 10.4**
10.5,10.6: **10.5 STARTS 10.6**
10.7,10.8,10.2: **10.7 BEFORE (10.8 OR 10.2)**

13,1: **13 BEFORE 1** [refers to next coffee-making cycle]

Figure 5. Temporal relations in the example product, based on Allen's interval logic [17].

off).

2. Role assignment has to be made explicit and dynamic. The need for explicit representation of the agents/operands and dynamic role assignment to entities originates from the 'richer' content of a process with multiple agents, and where actions and functions have to be connected 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 the user inserts it.
3. Facilities to assign temporal relations have to be added. Current FBRL does not offer possibilities to include knowledge about the connecting role of temporal relations. This is due to its focus on 'steady-state' processes that are not interrupted or disturbed by external influences such as users. Figure 5 shows how Allen's interval logic is applied to specify the temporal relations in the intended use process of the coffee maker. While the arrangement and numbering of functions in Figure 4 suggest only one possible sequence for the intended use, the specification in Figure 5 allows multiple sequences to be defined as 'intended'. Thus, for instance, filling the reservoir and inserting the filter can be done in either order, if only both are completed before the coffee maker is switched on.

Currently, these modifications have not yet been implemented: the additional vocabulary has not yet been added to the functional ontology, and the means to input and represent additional role characteristics as well as temporal relations have not yet been coded into the FBRL software. However, as Figure 4 and Figure 5 show, fully computerized processing is not needed to create process models of intended use.

4.2. Finding unintended forms of use

Once the intended use of a product has been defined, we can start identifying unintended forms of use. In principle, unintended use includes all elements that find themselves on the factual planes in the ontology-based scheme in Figure 3 but not in the subsets that are formed by the intent planes.

The total set of elements on the factual plane including the intended elements comprises the *possible* use. In this paper, our focus is on finding unintended elements in the process domain, i.e., unintended behavior. Unintended elements in the entity domain, which may also lead to other forms of use, may be derived or generated in similar ways.

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

- Additions of actions to, and omissions of actions from the intended actions–this mostly applies to user tasks but it can also apply to expected functionality of the environment, for instance failing to provide electric power,
- Variations in the temporal relations 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 (in this case also unintended elements in the entity domain are included, i.e., the eggs).

Other-than-intended behavior can be modeled using the same FBRL-based building blocks that were used to describe intended behavior in 4.1. An important issue here is, that there are countless specific use processes that can be built up from forms of unintended behavior. These cannot straightforwardly be combined 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 relations and dependencies of various nature, e.g., temporal, hierarchical, semantic etc. We found that, in our case study, the knowledge from the intended use can directly be deployed to generate forms of typical unintended use. In other cases, especially for atypical behaviors, additional knowledge will be needed from simulation results or from sources that can be less easily – or even impossibly – be uncovered automatically, such as company history.

Currently, no system component has yet been developed to support the generation of unintended use forms. However, it is likely that the simplest patterns of typical unintended behavior can relatively easily be generated automatically by a computer system. Typical patterns are included as deviation knowledge in the ontology-based scheme in Figure 3.

If, like in our current example, the user carries out one par-

 Copyright © 2004 by ASME

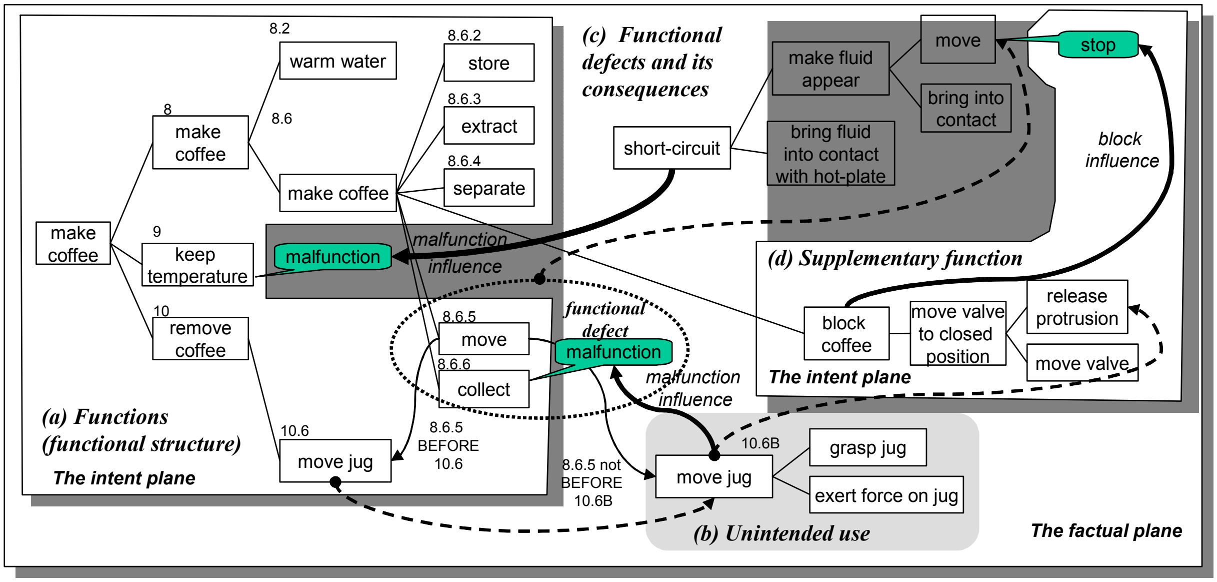


Figure 6. A coffee maker model including an unintended use, its effect and a supplementary function.

ticular 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 require assessment of all permutations of user-task sequences, or even the prediction of intermediate effects (see section 4.3). Such an automated 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' shortly after the first violation of the intended sequence. In such cases, the subsequent permutations of other actions do not have to be considered and can thus 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 [8]. For the more complex forms of unintended use, including atypical user behavior, the deviation knowledge in the ontology could be supplied manually with concepts of unintended use from non-formalized sources such as 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' [4] in functional FBRL can perhaps be applied, to generate forms of unintended use. For instance, two 'ways' of how the user can fill the reservoir of a coffee maker with water are (1) to fill it with tap water from the jug and (2) to carry the coffee machine to the tap to fill it directly.

Figure 6 shows a coffee maker model including an unintended use as it could be generated based on human reasoning by a designer. The details of the modeling schema are reported in [18]. Part (a) shows functions (intended behaviors) of the product, which include an intended user-action 'to move jug'. They are extracted from Figure 4 and simplified. As shown in Figure 5, there is a temporal constraint between the environment's function to move coffee into the jug (ID: 8.6.5) and the user's action to move the jug (ID: 10.6), because of the statement '8 BEFORE 10.2' in combination with default relationships. A possible unintended user action is to move the jug before all of the coffee has been collected. Because this is a violation on the intended constraint, the user action is unintended and thus is modeled on the factual plane in Figure 3. In Figure 6, the early jug removal forms a tree shown in part (b), which is separated from the functional structure (a). Tree (b) shows processes of unintended use in the same manner as intended behaviors. The sub-actions of grasping the jug and exerting force on it realize the displacement of the jug. Other parts named (c) and (d) of Figure 6 are discussed in the following sections.

4.3. Predicting the effects of use

Prediction of effects is expected to be more difficult to realize. As Figure 2 suggests, an interface with a simulation module can be developed to make predictions possible. However, returning to our coffee-maker example, it is not likely that current numerical simulation tools, like finite elements, bond 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. New collaborative research is currently being set up to connect quantitative simulation with user-product-environment models based on the 'nucleus' concept [6] via function carriers or ways of function realization [4] to FBRL-based functional models.

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, which is entered manually *(liquids in the immediate neighbourhood of live wires can cause short circuit)*. In absence of readily available quantitative or qualitative prediction facilities, we will have to assume for now that the designer has sufficient insight in the possible effects of a particular form of unintended

 Copyright © 2004 by ASME

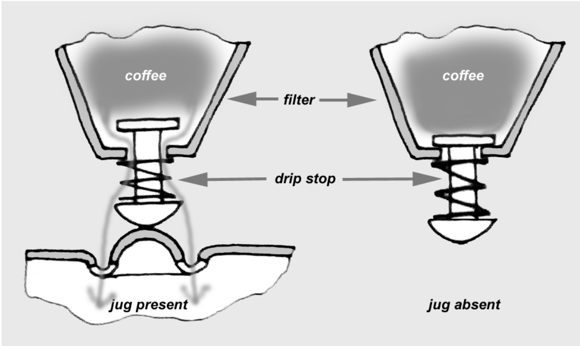


Figure 7. Drip stop

use, to deal with them in the advancement of his design work.

Figure 6(c) shows possible effects of the unintended use (the early jug removal) based on human reasoning by a fictitious designer. The unintended use obstructs the collecting function of the jug. We call unintended variations on functions such as malfunction *a functional defect*. In general, between an initial cause such as the unintended use and the functional defect, there is a cause-consequence chain that consists of multiple links, but in this case only the removal of the agent of the function, i.e., the jug, directly causes malfunction.

The functional defect also has its consequences. In this case, the malfunction of the collecting function would cause short-circuit as shown in Figure 6(c). Eventually, it causes malfunction of the temperature-keeping function of the hot plate. Basically, a functional defect propagates to other functions, the parent functions and connected functions, along the functional structure. The propagation of effects of the initial functional defect in this example forms a particular case, because it is outside the functional structure. Such process is modeled the same way as the process shown in Figure 6 (b), in that sense that the phenomena (behaviors or actions) in the processes are not directly related to the original functional structure.

The whole of these processes (b) and (c) are on the factual plane in Figure 3. Functional defects, such as the discussed malfunction and the bold causal link with the unintended use (the jug removal) or behavior (the short-circuit) indicate inter-plane relations between the factual plane and the intent plane.

4.4. Evaluating the effects of use

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, and 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. [19]) 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.

4.5. Generating solutions for undesirable effects.

Figure 7 and Figure 6(d) show 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. In Figure 6 (d), the jug removal simultaneously releases a spring-loaded protrusion that moves a connected valve to its closed position, which blocks dripping of the coffee. This new process on the intent plane serves to cut the cause-consequence chain from the unintended use (early jug removal) to the malfunction of the temperature-keeping function on the factual plane. Note that this inter-plane relation is based on the same semantic schema as the relation from the factual plane to the intent plane. We call such functions that cope with unintended use or behaviors *supplementary functions*.

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[2]. Another corrective solution could be to provide an alternative collector for coffee if the jug is missing. Such supplementary functions cut the cause-consequence chain that is caused by the functional defect. Cutting the cause-consequence chain that causes the functional defect is also possible. In that case we apply *preventive* solutions to restrain the user from early jug removal based on supplementary functions that prevent occurrence of the initial cause of the initial functional defect. Additionally, 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 (collecting) 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 solutions 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.

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 2 and reported in [4] and [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

[2] 'Corrective' refers to correcting the effects of user/environment actions, not to correcting design flaws. In other literature, 'corrective' is sometimes used in the latter context, where corrective redesign includes preventive solutions [10].

 Copyright © 2004 by ASME

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 grey background in Figure 2).

In the third place, the system could suggest function fulfillers that perform 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.

5. CONCLUSIONS AND FUTURE WORK

With minor 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. Some more substantial extensions, such as explicit representation of temporal relations 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. Since even the most advanced knowledge retrieval and data mining techniques do not seem to offer ways for automatic retrieval of such knowledge, we will have to assume that, for the coming years, it has to be collected and included 'manually' by human intervention.

Aiming at explicit representation of supplementary functions, investigation on details of modeling schema of unintended behaviors of products was done in [18]. In particular, representation of the linkage between the intent plane and the factual plane, and categories of supplementary functions, have been explored. This paper successfully uses the same framework to cope with unintended actions of users as well. Currently, an ontology for modeling unintended product behaviors is being implemented. We will extend it to unintended user actions and environment behaviors. In parallel, a user interface will have to be developed and software will have to be written to make the ontologies accessible to designers so that they can enter and model intended behaviors and unintended behaviors. Although automated generation of unintended behaviors will not be provided at this stage, we expect that the modeling possibilities will inspire designers to explore the unintended use more productively, and thus anticipate the actual usage of the product more effectively.

Future steps are (1) developing tools for generating alternative behavior and for solutions to compensate for unintended behavior and (2) providing connectivity with quantitative design support.

ACKNOWLEDGMENTS

This research has been carried out as a collaborative research supported by the Japan Society for the Promotion of Science (JSPS Fellowship ID L02536), the Netherlands Organisation for Scientific Research NWO (record B 79-84) and by Delft University of Technology. The authors wish to thank Mariko Yoshikawa for her contributions.

REFERENCES

[1] Van der Vegte, W.F., Horváth, I., 2002, 'Consideration and modeling of use processes in computer-aided conceptual design', Transactions of the SDPS - J. of Integrated Design and Process Science, **6**, No. 2, pp. 25-59.
[2] Gruber. T. R., 1993, 'A translation approach to portable ontologies'. Knowledge Acquisition, **5,** No. 2, 199-220.
[3] Sasajima, M., Kitamura, Y., Ikeda M., Mizoguchi, R., 1995, 'A representation language for behavior and function – FBRL', Expert Systems with Applications, **10**, No.3/4, pp.471-479.
[4] Kitamura, Y., Mizoguchi, R., 2003, Ontology-based description of functional design knowledge and its use in a functional way server, Expert Systems with Application, **24**, No. 2, pp. 153-166.
[5] Kitamura, Y., Mizoguchi, R., 2004, Ontology-based systematization of functional knowledge, Journal of Engineering Design, **15,** No. 3, To appear.
[6] Van der Vegte, W.F., Horváth, I., 2003, 'Use-driven product conceptualization based on nucleus modeling and simulation with scenarios', *Proc. 15th European Simulation Symposium*, Delft, The Netherlands.
[7] Kmenta, S., Ishii, K. , 1998, 'Advanced FMEA using meta behavior modeling for concurrent design of products and controls', *proc. 1998 ASME DETC*, Atlanta, GA.
[8] Hata, T., Kobayashi, N., Kimura, F., Suzuki, H., 2000, 'Representation of functional relations among parts and its application to product failure reasoning', *Proc. Int. CIRP seminar on Design with Manufacturing*, Haifa, Israel.
[9] Lee, B.H. , 2001, 'Using Bayes belief networks in industrial FMEA modeling and analysis', *proc. Annual Reliability and Maintainability Symposium*, Philadelphia. PA.
[10] Goel, A., Chandrasekaran, B. , 1989, 'Functional representation of designs and redesign problem solving', *proc. IJCAI-89*, Detroit, MI.
[11] Buur, J., Windum, J., Jakobsen, M.M., 1991, 'Man/machine interface design needs systematic methods', *proc. ICED '91*, Zürich, Switzerland, pp. 741-746.
[12] Otto, K.N, Wood, K.L., 1999, 'Customer integrated systematic design', Transactions of the SDPS, J. of integrated design and process science, **3**, no. 4, pp. 61-74.
[13] Suto, H., Kawakami, H., Katai, O., Horiuchi, T., 2000, 'Hierarchical modeling of artifacts based on interaction with humans and the environment', *proc. EDA 2000*, Orlando, FL.
[14] Janhager, J., 2003, 'Utilization of scenario building in the technical process', *Proc. ICED '03*, Stockholm, Sweden.
[15] Van der Vegte W.F., Kitamura Y., Mizoguchi R., Horváth I., 2002, 'Ontology-based modeling of product functionality and use – part 2: considering use and unintended behavior', *Proc. EDIProD' 2002*, Łagów, Poland, pp. 115-124.
[16] Pahl, G., and Beitz, W., 1988, 'Engineering design - a systematic approach', The Design Council, London.
[17] Allen, J.F., 'Maintaining knowledge about temporal intervals', Communications of the ACM, **26**, No. 12, 1983, pp. 832-843
[18] Koji, Y., Kitamura, Y., Mizoguchi, R., 2004, 'Towards modeling design rational of supplementary functions in conceptual design', *Proc TMCE 2004*, Lausanne, Switzerland, pp. 117-130.
[19] Franscheschini, F., Galetto, M., 2001, 'A new approach for evaluation of risk priorities of failure modes', Int. J. of Product Research, **39**, No. 13, pp. 2991-3002.

 Copyright © 2004 by ASME