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

THE CONCEPT OF FUNCTIONAL SURFACES AS CARRIERS OF INTERACTIVE PROPERTIES

Ulf Sellgren and Sören Andersson

Abstract

The perceived use and attractiveness of a product relies on its subjective understanding by one or several human being(s) to a great extent. The attractiveness of a product depends strongly on its aesthetic appearance and how we perceive the whole product and its external aesthetic and ergonomic features with all our senses in relation to the performance we, for some reason, expect from the product. Fulfillment of the technical functional requirements and the cost constraints is often seen as an obvious prerequisite for an attractive product but it is not seen as *the* differentiator among competing product. It is increasingly important that the product also have distinctive *interactive properties* of a semiotic, ergonomic and/or aesthetic nature that distinguish the product from other products on the market. Consequently, development of many types of products benefits from an integrated and holistic treatment of both the technical and the interactive product aspects.

The fact that products are designed by someone to be perceived by someone else makes treatment of ergonomic and aesthetic requirements and implementations quite complicated and even fuzzy. A significant challenge to design research is to find ways to represent the "hard" technical requirements and the more "soft" interactive requirements, the implementations of the technical and the interactive properties, and the relations between these types of requirements and properties. This paper presents a model-based approach that addresses this challenge. Interactive and technical functional surfaces and how they fit into a general modeling principle of technical systems are elaborated on. The general modeling principle includes both technical and interactive interface models. This paper, furthermore, presents an integrated matrix-based representation of the technical and interactive properties of a technical product and relates these properties to the stated customer requirements. The presented approach is exemplified with a recent design project.

Keywords: Functional surface, interactive function, interface, model structure matrix

1 Introduction

A quality product meets or exceeds customer needs. Manufacturing companies that operate in a global business environment face challenges and opportunities to develop their business by providing a variety of high quality and attractive products to the market. Modern products are becoming increasingly complex. The statement by Lange [1] “that a product is designed by someone to be perceived by someone” (else) acknowledges the highly complex and iterative character of the design process. The perceived use of a product relies on subjective understanding of its attractiveness to a great extent.

The attractiveness of a product depends strongly on how we perceive the product with all our senses in relation to the expected performance of the product [2]. This means that the "technical" requirements and the various design constraints, such as cost, must be satisfied. But, sometimes even more important, the product must also have "something more" properties of an aesthetic and/or ergonomic nature.

The German philosopher Alexander Gottlieb Baumgarten coined the term aesthetics in the 18th century and he established aesthetics as a separate branch of philosophy dealing with the nature of beauty. The word *aesthetic* can be used as a noun with the meaning "that which appeals to the senses". The elements that contribute to the aesthetic appeal of an artifact depend upon the medium under design, such as art, music, architecture, performing arts, literature, gastronomy, information technology, and product design. Aesthetics in art, as an example, is related to the principles of symmetry/asymmetry, focal point, pattern perspective, direction of motion, and proportion. Design aesthetics involves the study of appearance and perception of shape, functions, attributes, and behaviors of products [3]. The rules of creating designs with intended appearance through form giving, materialization, and decoration have not been sufficiently explored yet [4].

Ergonomics (from Greek ergon work and nomoi laws) is the study of design objects to be better adapted to the shape of the human body and/or to the user's posture. Ergonomics is much larger than looking at the physiological and anatomical aspects of the human being. The psychology of humans is a key element within the ergonomics discipline. This psychological portion of ergonomics is often referred to as human factors or human factors engineering. Research on human-product interfaces concentrates on various concepts for both physical and virtual interfaces [5]. Non-quantifiable factors, such as user satisfaction and comfort emotional responses, are also getting emphasis in current research [4].

The *functionality* of an artifact describes and represents a part of the designer's intention or design rationale (Kitamura et al., 2002). Based on an extensive classification study of highly complex natural systems (e.g., biological systems and cosmic systems) and engineering systems (i.e., systems that are human designed and having both significant human and technical complexity) Magee and Weck [7] found that function type as originally proposed by Hubka and Eder [8], Pahl and Beitz [9], and van Wyk [10] is the only technical attribute able to differentiate among engineering systems. Function is consequently a characteristic product attribute that captures important knowledge about an existing product, component, or principal solution whenever a task involves re-design, adaptive design, or creative design by analogy.

Since the aesthetic and ergonomic properties have a major influence on the human-artifact interaction we will further on refer to them as *interactive properties*. Consequently, we can make a distinction between technical and interactive product requirements.

Development of theories, methods, and tools to manage and reduce design complexity is a fundamental challenge to design research. This challenge can be partly addressed by developing methods and representations that enable integrated treatment of "classical" technical, aesthetic, and ergonomic product aspects. That is, to find ways to understand, represent, and communicate the more "hard" and objective requirements and the more "soft" and subjective requirements, the technical and interactive properties implemented in the product, and, perhaps most importantly, the relations between the stated requirements and the implemented product properties. This paper presents a new model-based concept that addresses this challenge.

2 Customer requirements, functions, and surface implementations

2.1 Customer requirements

The inclusion of customer desires is integral to the design of quality products [11]. Several methodologies help companies include customer desires in their products. One of the most popular methodologies is Quality Function Deployment (QFD), which graphically is represented by the House of Quality [12]. In QFD, customer requirements typically refer only to the consumer or end-user requirements. End-user requirements are expectations that the end-user has on the product.

Gershenson and Stauffer [13] argued that it is necessary to consider all customers, not just end-users, from the beginning of the design process. They defined a user context and partitioned the customer requirements into four types; end-user, corporate, regulatory, and technical requirements. *End-user requirements* include the end-user' expectations about the product's capabilities, aesthetics and usability. *Corporate requirements*, which encompass business issues as well as product life-cycle issues, are of concern to the individuals involved in the related engineering and non-engineering disciplines. These individuals are also commonly the source for these requirements. *Regulatory requirements*, which include safety/health, environmental/ ecological, disposal and/or political issues, are imposed by the society. *Technical requirements*, which include such things as engineering principles, material properties and physical laws, is usually the input to the primary design phase. The *user context* characterizes the end-user of the product helping the development team to apply the four types of requirements. As such, it is not a customer requirements area.

Here, we prefer to exclude the technical requirements from the customer requirements and we thus end up with three context focused sets of customer requirements (see figure 1) that are stated at the beginning of the product development process and implemented, verified, and validated in the following stages.

Error!

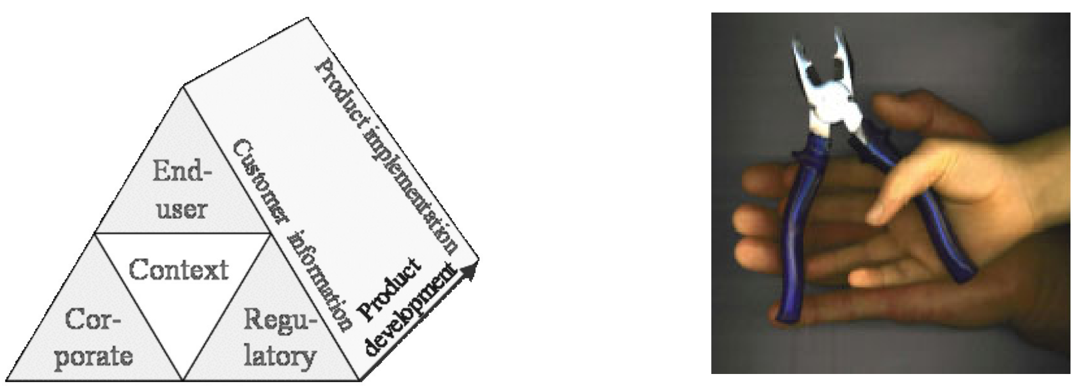

Figure 1. Requirements wedge showing the three types of context dependent customer requirements (left) and a photo that visualizes an aspect of the ergonomic context (right). Photo: Linda Rose.

2.2 Technical and interactive functions

The function tells what the design is intended to do (i.e., the purpose) and is often used for specifying design requirements, sometimes referred to as functional requirements (e.g., [14]). The stated functionality of an artifact thus describes and represents a part of the designer's intention or design rationale. Whenever a task involves adaptive design, re-design, and/or design by analogy function is, consequently, a characteristic product attribute that captures important knowledge about an existing product or component.

This purpose or function can be defined as relationships between inputs and outputs of energy, mass, and information, or as a change in the fluxes thereof (e.g. Qian and Gero [15]). Manipulation of flows (or fluxes) involves actions. Function can, thus, be characterized by two kinds of variables – action and flow – that can be classified into two taxonomy hierarchies. The action taxonomy represents a hierarchy of verbs (e.g., store, transport). In the flow taxonomy, the flow superclass is categorized into material, energy, and information subclasses. These three subclasses can be categorized further. For example, energy may be mechanical, which may be further classified as kinetic or potential, and so on.

Actions and flows can be decomposed in many different ways, but some decompositions are more convenient to use than others. Little et al. [16] refer to their set of actions and flows as a basis set. The mathematical definition of basis requires that the set spans the space and the components of the set are linearly independent. For example, in many engineering situations the eigenvectors of a dynamic is a convenient and thus attractive basis set. The set of functions and flows proposed in [16] and further elaborated on by Stone [17] is a basis set in a qualitative sense and not in a strict mathematical sense. It is further on referred to as the *Little function base set*. This set of function classes (i.e., branch, channel, connect, control magnitude, convert, provision, signal, and support) and their specializations (e.g., sense, indicate, display, and measure are signal specialization) is used in the presented research as if it was a function base set in a strict sense.

Flows are first distinguished by class, such as material, energy, and signal (Pahl and Beitz, 1996), then by basic flows and, if desired, into complement flow. Human energy and human material can both be viewed as basic flows [17]. A flow selected from the list fills the object spot of the verb-object or Function – Flow description, e.g., *Import Human Hand*.

If we have an ambition to use functional decomposition as a means to reduce complexity in product development we need a mechanism that help us to distinguish between technical functional requirements and functional requirements that are directly related to human factors.

Warell [18] divided functions into technical functions and interactive functions. With this terminology, technical functions are internal product functions while interactive functions are human-product interactions.

Technical functions are associated with the flow, transformation, and storage of energy, materials, and information in the product. A technical function can be active, for example when it involves transporting or transforming something, or passive, for example when it involves supporting something.

Interactive functions are associated with the interaction between the user and the product and communicate the usability and the attractiveness of a product [18]. They can be decomposed into *ergonomic functions, semantic functions,* and *syntactic functions*. Syntactic and semantic functions are *communicative functions*. An ergonomic function captures the relation between a product and the physical and physiological capability of the human body. The function *Import Human Hand*, in the Little function base set terminology, is an example of an ergonomic interactive function. A semantic function captures how products or parts of the product communicate their purpose to the user. Syntactic functions capture how the form of a product, or of part of a product, are perceived by humans. It is often difficult for engineers to clearly distinguish between a semantic and a syntactic function. Furthermore, semantic and syntactic functions are often interrelated and act in parallel. In the examples presented below, we will thus use the common term *communicative function* for these two types of interactive functions. Communicative functions can be treated as signal functions in the Little function base set.

3 Implementation system

A technical artifact can be viewed as an implementation system, which can be defined as a set of subsystems that are interrelated to each other and to the whole so as to satisfy a common functional requirement by implementing an intended behavior (e.g., [15]). Many subsystems are configurations of components that are developed from principal solutions or organs. The technical function of a component is often implemented as a mechanical contact relation within the component and between the component and the surrounding components of the system, or as interaction relations between the component and the environment.

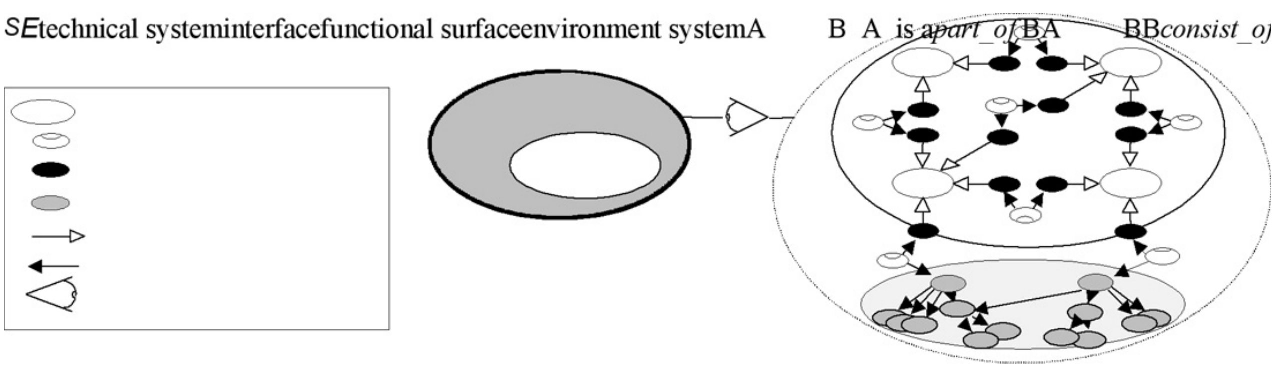

Figure 2. A system *S* as an aggregation of subsystems and interfaces in an environment *E*

Our modeling principle, which is based on the modular approach proposed by Sellgren [19], is to look at a technical system as an aggregation of subsystems and interfaces (see figure 2). The subsystems have defined functional surfaces (discussed below), which interface with other functional surfaces. A modular architecture enables easy modification of a systems model [20].

3.1 Functional surfaces

The concept of functional surfaces originate from the work performed by Tjalve [21]. *Functional surfaces* are surfaces on technical products that are *carriers of properties that enable technical and interactive functions*. What we mean by functional surfaces on a product can be exemplified by the bottle opener concept in figure 3. The most obvious functional surface on a bottle opener is the *technical functional surface* that has to fit to the bottle cap and transmit torque/force from the hand to open the cap. Another functional surface is located at the other end, where the user holds the tool and applies the force. This part has ergonomic properties. The function of the relation between that part and a gripping hand is ergonomic, and the related functional surface is thus an *ergonomic functional surface*. The rest of the tool is required to provide material that can support the load/torque that must be transferred from the grip to the front end of the tool. The form of the middle part is rather free, and it can thus be given distinctive aesthetic properties as long as the technical functional requirement of supporting the applied load is fulfilled. The surface of the middle part is thus a *communicating functional surface*.

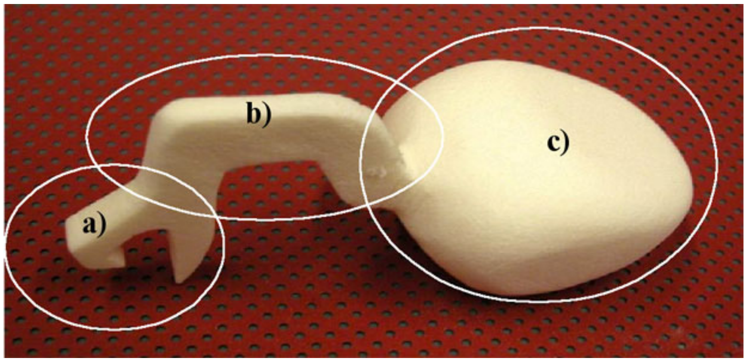

Figure 3. . Illustration of three functional surfaces on a bottle opener: a technical functional surface, a; an interactive (communicative) functional surface, b; and an interactive (ergonomic) functional surface, c. Modeled by a student in the Design and Product Realization Program at KTH. Photo: C.-M. Johannesson.

The two interactive surfaces can be shaped in many alternative ways. Figures 3 and 4 show some of the concepts produced by freshman students in the Design and Product Realization Program at KTH.

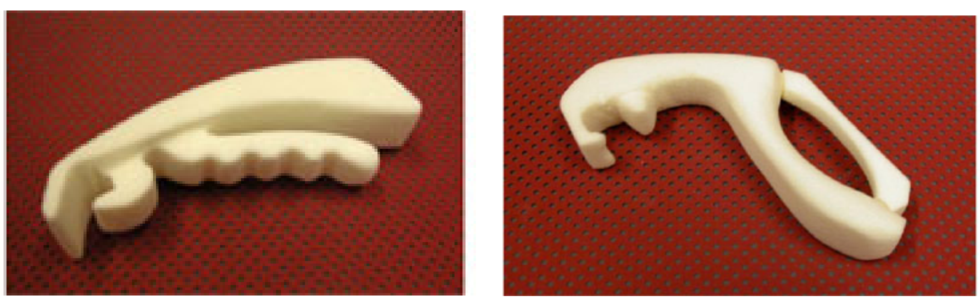

Figure 4. Two examples of bottle opener concepts developed by freshman students in the Design and Product Realization program at KTH. Photo: C-M Johannesson.

3.2 Functional interfaces

Functional interfaces of a product are the interfaces that realize or implements the different technical and interactive functions. Consequently, we can define a functional interface as *an intended interaction relation between two functional surfaces*. There are other interfaces that can be classified as accidental and/or unintended. Such interfaces, which represent physical side-effects or design errors, are more closely related to the behavior domain than to the functional and implementation domains.

This definition of a functional interface easily embraces all types of interaction relations between technical functional surfaces within the technical system as well as between the environment and the technical system. Furthermore, if we generalize the concept and represent the human side of the human–product (or man-machine) interaction as an interactive functional surface, we can also represent interactive functions as functional interfaces.

In summary, make a distinction between two types of functional interfaces:

- *Technical interface* – an intended interaction relation between a pair of technical functional surfaces in or on a technical system or in the environment.
- *Interactive interface* – an intended interaction relation between an ergonomic or communicative functional surface on a technical system and a sensory feature of a real or generic human.

4 Representations of domains and relations

Different representations of customer requirements, functional requirements, and the implementation structure can be important tools to manage complexity in the product development process. The architecture of a technical implementation system can for example be represented in several ways. A virtual reality (VR) representation is attractive for communication purposes, but it lacks strict and complete representation. A graph-based representation allows the properties of the system to be captured formally and completely, but it is difficult to communicate and it is not suitable for large problems. A matrix-based representation such as a product-based design structure matrix (DSM) [22] provides a compact, complete, and clear representation of a complex system, but can be difficult to communicate to non-experts. Both the graph and the DSM may show causal (i.e. directed) as well as non-causal relations. Consequently, a combination of a graphical/symbolic representation and a matrix-based representation of subsystems and interfaces is generally preferred. Such a combined approach provides both a strict and complete representation of the model architecture of a system and an easily understandable illustration of the related systems models [20].

Since the aim of the presented research is primarily to represent the relations between the customer, functional, and implementation domains, matrix representations are chosen. That is, the customer requirements (CR), as well as the derived functional requirements (FR) are represented as vectors, and they are related with a matrix, referred to as CFM in figure 5. Further more, the implementation structure is combined with the environment and human systems in an extended DSM-alike implementation-structure matrix (ISM in figure 5). The ISM and CFM matrices are then connected with a function-implementation matrix (FIM).**Error!**

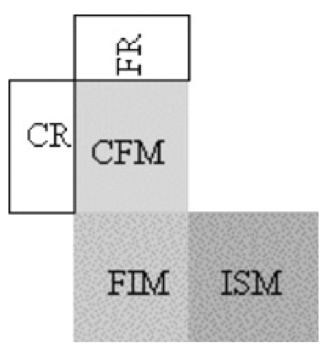

Figure 5. A matrix representation that relates customer requirements, functional requirements, and implemented features.

4.1 Representation of customer and functional requirements

A structured and condensed version of the list of requirements for the bottle opener is shown on the left in figure 6. The customer requirements are decomposed into end-user requirements, corporate requirements, and regulatory requirements. The structuring of the list of requirements is preliminary and it can most certainly be improved. The condensed result of the analysis of the functional requirements for the bottle opener is shown on the right-hand side of figure 6. The functional requirements are divided into interactive and technical functions. The interactive functions are further divided into ergonomic and communicative functions, and the technical functions are divided into active and passive functions. The basic functions are then expressed in the “Little base sets”. The focus in the presented work has been on the interactive functions.

Customer requirements (CR)	Functional requirements (FR)
End-user requirements	**Interactive requirements**
Good ergonomic shape	**Ergonomic requirements**
Good grip	Import Human Hand
Soft handle	Import Human Force
Friction grip	Secure Human Hand
Good opening performance	Stop Human Hand Slipping Motion
Low human force	**Communicative requirements**
Good grip on cap	Signal Aesthetic Appeal
Adapted to all recyclable glass bottles	Signal Cap Removal
Adapted to all standard caps	Signal Company Brand
Easy retrieval	Signal Design Innovation
Open storage	**Technical requirements**
Pocket storage	**Active requirements**
Aesthetically attractive	Secure Cap
Innovative look	Separate Cap
Alternative colors	Secure Wall Attachment
Corporate requirements	Separate Wall Attachment
Ecological	Separate Dissimilar Material
Environmentally friendly material	**Passive requirements**
Recyclable/reusable material	Secure Technical Features
Life-cycle issues	
Easy to disassemble/recycle	
Brand recognition	
Recognizable and consistent company-look	
Safety requirements	
No-brittle material	
Regulatory requirements	
Non-toxic material (Norm)	
Safe for children over three years of age (Norm)	

Figure 6. A condensed list of customer (left) and derived functional requirements (right) for the bottle opener

A basic functional requirement may be fully or partially related to one or several customer requirements. These relations, which represent the actual structuring and decomposition of the problem space, is represented with the integration matrix, i.e. the CFM in figure 7.

4.2 Extended implementation structure matrix

One of the strengths of DSM is its applicability to large and complex systems. Researchers in the area of engineering design (e.g. Wood et al. [23]) have argued that the DSM is a convenient and

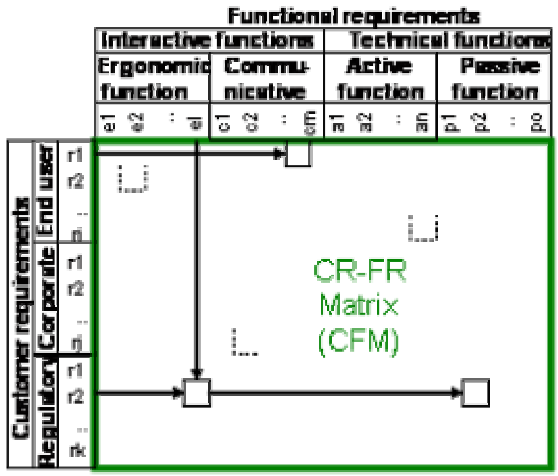

Figure 7. A condensed list of customer (left) and derived functional requirements (right) for the bottle opener

reasonably complete representation for many engineering tasks that require an integrated treatment of product architecture, modularity and technical interface aspects. With the aim to support efficient configuration of complex models and to enable navigation in system models, a representation of the architecture of behavior models, referred to as the model structure matrix (MSM) has been developed [20]. The MSM, which is a model-based DSM, provides a compact representation of a complex behavior model and its building blocks (i.e., subsystem models and interface models. The interface models are off-diagonal terms in the MSM matrix. The causality at each interaction is an internal property of the related interface.

Error!

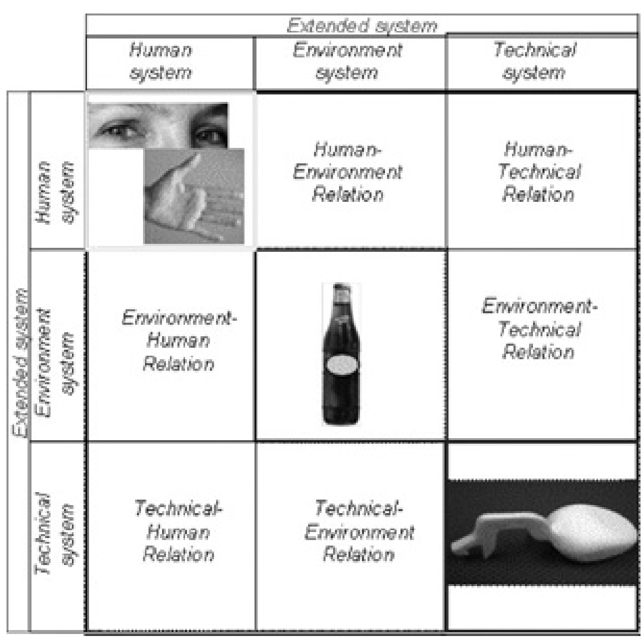

Figure 8. Principal structure of the extended ISM representation of the bottle opener

An implementation structure matrix (ISM), which targets the implementation domain, has a similar structure as an MSM. The principal structure of an extended ISM is shown in figure 8. The technical ISM is extended with submatrices representing the environment, the human (i.e., the person uses the opener and other significant humans such as potential customers), and the interactions between the human, the environment, and the technical system.

We will now represent the model of the bottle opener in figure 3 with an ISM. The three "subsystems" in this case are the *FrontPart*, the *MidPart*, and the *RearPart*. We assume that the functional surfaces are included in (features of) the model of each part. The different parts have functional surfaces as outlined above, plus the new technical functional surfaces generated when we divided the tool into three separate parts. The interfaces between the two pairs of internal technical functional surfaces are rigid connections between the related section surfaces. In this case, the other functional surfaces are more interesting. As can be seen in figure 9, the three functional surface models mentioned earlier are not related to anything. But we know that the functional surface on the front part has a technical function related to a bottle cap. The functional surface of the rear part is related to the hand of the user who is gripping the tool. The functional surface of the middle part has a communicative function.

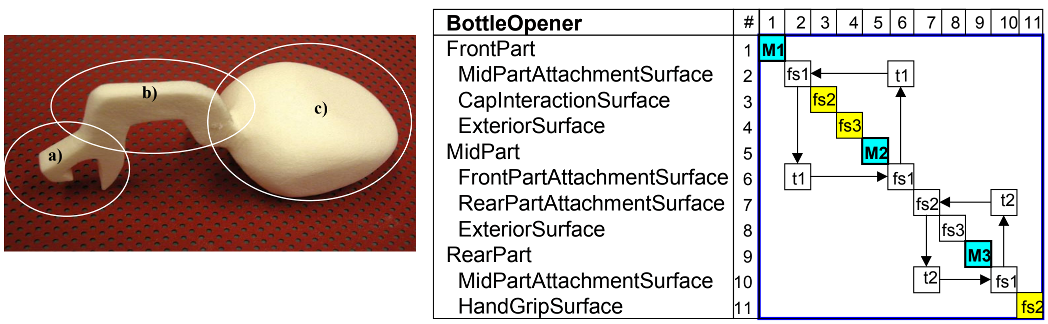

Figure 9. An ISM of the bottle opener technical system

If we add the environment (the bottle and its cap) and a human to the ISM matrix, as shown in figure 10, we have an extended ISM representation of a systems model that includes human, environmental, and technical systems, and the interaction between the three physical domains. The technical, ergonomic, and communicative interfaces are labeled *ti*, *ei*, and *ci*, respectively. The initial technical functional surface on the front part (i.e., the *CapInteractionSurface* in figure 9) has been decomposed into two functional surfaces, denoted *CapGripSurface* and *CapSupportSurface* in figure 10.

4.3 Knowledge integration matrices

We have already discussed the concept of a CFM matrix that relates the stated customer requirements to the defined functional requirements. If we add a matrix (FIM in figure 11)) that relates the objects represented by the extended implementation structure matrix to the functional requirements, we have a traceability mechanism that enables cause (customer requirement) and effect (implementation) studies and free navigation between the customer, functional, and implementation domains. The purpose of these two knowledge (or domain) integration matrices is to assist product development by reducing some of the complexity issues in design in general and in re-design in particular.

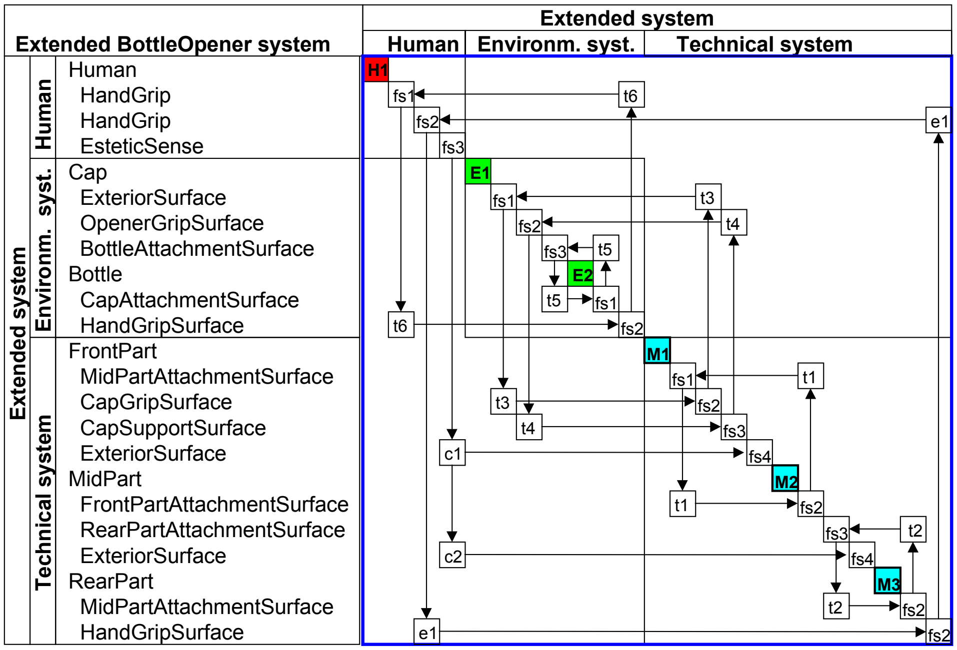

Figure 10. An extended ISM of the complete bottle opener system

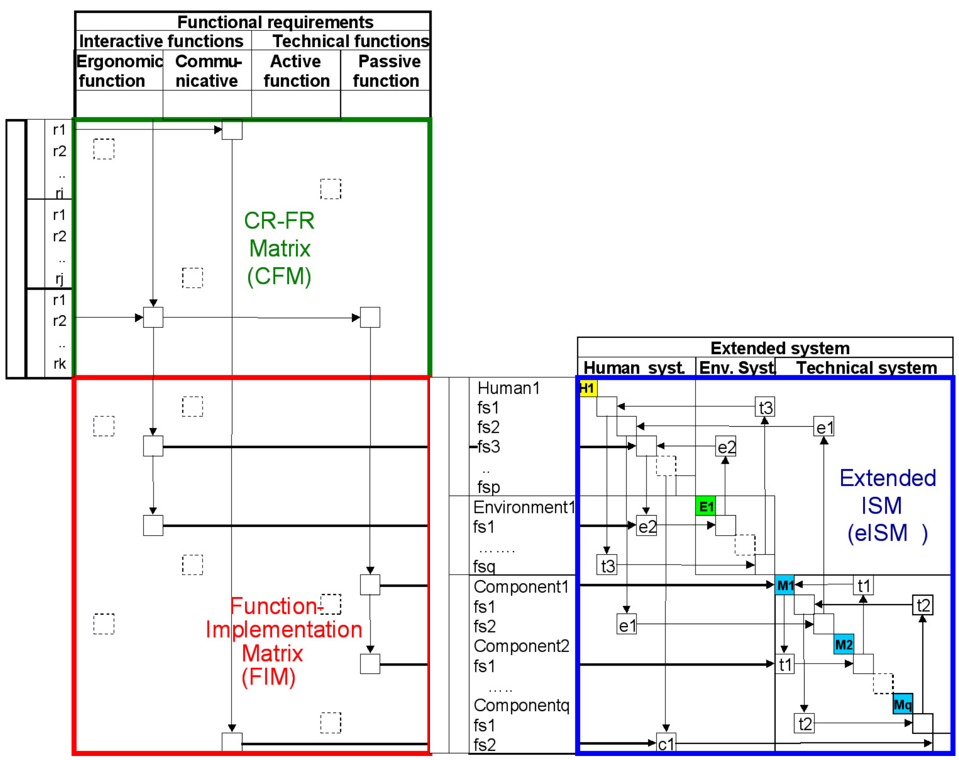

Figure 11. A structure of matrices linking CR and FR to an extnded MSM representation of the design

5 The representations and the process

To support efficient configuration of complex models, the ISM can be used as a navigation and architecting tool. ISM can be viewed as a model-based DSM. By extending the ISM to include the environment and human objects, both technical and interactive functions can easily be treated and modeled in a consistent way. Thus the extended ISM makes the model representation more complete and situated than a model that is limited to technical objects.

By analyzing the results of the simple bottle opener design case and a more complex case of designing a new truck seat [2], we arrive at a preferred development procedure with eight distinct steps:

1. Define the user context, e.g., define the properties of the targeted users and the market.
2. Collect the list of customer requirements and structure them (e.g., as end-user, corporate, and regulatory requirements). Create the CR vector.
3. Analyze the different requirements in terms of technical and interactive functional requirements. Create the FR vector and the CFM matrix.
4. Generate the base for the extended ISM with the basic human, environmental, and technical submatrices. Create the basic structure of the eISM matrix, which will be expanded as the process continues.
5. Define the control volume of the product (i.e., the spatial environment), the functional surfaces of the environment, and the functional surfaces (i.e., the relevant senses) of the human(s). Expand the eISM with these objects.
6. Relate the eISM matrix to the functional requirements. Create the FIM matrix.
7. Start to generate technical concepts. Expand the technical eISM submatrix with principal solutions and/or reuse components, and define interfaces.
8. Proceed by decomposing the functional requirements and objects of the technical system. Update FR, CFM, FIM, and eISM.

Following the conceptual design activities and some detail design, a design concept, such as the bottle opener, can be presented for go/no-go evaluation. Many of the requirements listed are directly related to the functional surfaces that were defined. Simply by representing the relations between the different subsystems of the product, the functional surfaces on the subsystems, and the functional surfaces of the environment and a human, it is easy to trace the relations between the different solutions and the customer requirements. It is then easy to illustrate how the different requirements are fulfilled in the actual design by using the representation presented in figure 11.

6 Conclusions and discussion

This paper examines interactive and technical functional surfaces and how they fit into a general modeling principle for technical systems, using a framework originally proposed in [19] and further developed in [24]. This novel modeling principle includes both previously presented technical interface models and interface models representing interactive functions. The use of functional surfaces and interactive functions were inspired, respectively, by Tjalve's theory of form design [21], Warell's definition of interactive functions [18], and the function base set presented in [16] and [17].

Using an approach with functional surfaces and interactive functions, we have found that system models that include technical systems, the environment, and human actor(s) can be represented with the same formalism previously used (e.g., [24]), to represent purely technical systems. Such an implementation structure matrix (ISM) may represent the structure of principal solutions as well as the more detailed implementations in the later stage of the development process. An ISM is thus a snapshot of the structure of the product and it is expanded as the development of the product proceeds. We have also shown that the matrix representation of the implementation domain can be linked to the functional and customer domains with a set of knowledge integration matrices.

Furthermore, we have presented an approach to product development that utilizes this representation. We argue that starting product development by analyzing the customer's and others demands from a technical and an interactive functional view tends to result in a better correlation between the customer demands and product properties. We believe that the success factor in this case is proper development of functional surfaces and their interaction with other functional surfaces within the system or in the environment, and with the human(s) who use or interact with the product. The relations between the customer requirements and the features of a developed product are not always direct and easy to track. The approach presented here suggests a new way of managing this type of complexity, which presents a significant industrial challenge.

The presented approach has been studied by retrospectively analyzing two recent projects. The results are tentative, but promising. We plan to scrutinize the approach and further develop it to be more situated in upcoming projects. The Little base set also has to be expanded and adapted to handle communicative and also ergonomic types of functions more efficiently. Another target research area is to close the loop from the eISM back to the customer requirements in order to be able to assess how customers judge a developed concept in relation to their stated requirements.

7 Acknowledgements

The work presented here was performed in the Interface project and financially supported by the Swedish Strategic Research Foundation.

References

[1] Lange, M.W., "Design semiosis – Synthesis of products in the design activity," Doc. Thesis, Dept. of Machine Design, KTH, TRITA MMK 2001:14, Stockholm, 2001.

[2] Andersson, S and Sellgren, U., "Representation and use of functional surfaces", Proc. 7th Workshop on Product Structuring, Chalmers, Sweden, 24-25 March, 2004

[3] Smets, G., "Industrial design engineering and the theory of direct perception and action", Ecol Psychol, 7(4), pp. 329-374, 1995.

[4] Horváth, I., "A treatise on order in engineering design research", Research in Engineering Design , 15, pp.155-181, 2004.

[5] Dix, A., finlay, J., Abowd, G., and Beale, R., "Human-computer interaction", 2nd edn. Prentice-Hall, Upper Saddle River, New Jersey, 1998.

[6] Kitamura, Y, Sano, T., Namba, K., and Mizoguchi, R., "A functional concept ontology and its application to automatic identification of functional structures", Advance Engineering Informatics, Vol. 16, pp. 145-163, Elsevier Science Ltd., 2002.

[7] Magee, C.L. and de Weck, O.L., 2004, "Complex system classification", Fourteenth Annual International Symposium of the International Council On Systems Engineering (INCOSE), June 20-24, 2004.

[8] Hubka, V. and Eder, W.E., "Theory of technical systems", Springer-Verlag, Berlin, 1988.

[9] Pahl, G. and Beitz, W., "Engineering design a systematic approach", 2nd edition, Springer, 1996.

[10] Van Wyk, R.J., "A standard framework for product protocols", in Khalil (ed.), Management of Technology, Geneva, 1988, pp.93-99, 1998.

[11] Ramaswamy, R. and Ulrich, K., 1992, "Augmenting the house of quality with engineering models", Proc. 1992 ASME Design Technical Conf. 18th Design Automation Conf., Scottsdale, AZ, USA, 1992.

[12] Hauser, J.R. and Clausing, D., "The house of quality", Harward Business Review, 1988.

[13] Gershenson, J.K. and Stauffer, L.A., "A Taxonomy for Design Requirements from Corporate Customers", Research in Engineering Design, vol. 11, pp. 103-115, Springer-Verlag London Limited, 1999.

[14] Suh, N.P., "The principles of design", Oxford University Press, Inc., New York, USA, 1990.

[15] Qian, L. and Gero, J.S., "Function-behavior-structure paths and their role in analogy-based design", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 10, pp. 289-312, 1996.

[16] Little, A., Wood, K., and McAdams, D., "Functional analysis: a fundamental empirical study for reverse engineering, benchmarking and redesign", Proc. of the 1997 Design Engineering Technical Conference 97-DETC/DTM-3879, Sacramento, CA, USA, 1997.

[17] Stone, R., "Towards a Theory of Modular Design", Doc. Thesis, The University of Texas at Austin, 1997

[18] Warell, A., "Design syntactics: a functional approach to visual product form", Doc. Thesis, the Chalmers University of Technology, Gothenburg, Sweden, 2001

[19] Sellgren, U., "Simulation driven design – motives, means and opportunities," Doc. Thesis, Dept. of Machine Design, KTH, TRITA MMK 1999:26, Stockholm, 1999.

[20] Sellgren, U., "Architecting models of technical systems for non-routine simulations," International Conference on Engineering Design ICED 03, Stockholm, 2003.

[21] Tjalve, E., "A short course in industrial design," Newnes-Butterworths, London, 1979.

[22] Pimmler, T.U. and Eppinger, S.D., "Integration analysis of product decompositions," ASME Design Theory and Methodology Conference, DE Vol. 68, 1994.

[23] Wood, K.L., Van Wie, M.J., Greer, J.l., Campbell, M.I., and Stone, R.B. "Interfaces and product architecture", ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC01/DTM-21689, 2001.

[24] Andersson, S. and Sellgren U., "Modular product development with a focus on modeling and simulation of interfaces", Design Society – Workshop on Product Structuring, Copenhagen, January, 2003.

Ulf Sellgren and Sören Andersson
Royal Institute of Technology (KTH)
KTH-Machine Design
SE – 100 44 Stockholm
Sweden
Phone: +46 – 8 790 73 87
E-mail: ulfs@md.kth.se