7th Workshop on Product Structuring – Product Platform Development
Chalmers University of Technology, Göteborg, March 24-25, 2004

REPRESENTATION AND USE OF FUNCTIONAL SURFACES

Sören Andersson and Ulf Sellgren

Abstract

Manufacturing companies that operate in a global business environment face challenges and opportunities to develop their business by providing a variety of attractive products to the market in an environment where speed and agility are essential. 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. This normally means that the technical functional requirements and the cost constraints must be fulfilled. But, sometimes even more important, the product must also have “something more” properties of a semiotic, ergonomic and/or aesthetic nature. Consequently, a holistic view is necessary in the product realization process and both the technical and the “something more” aspects must be considered. We will name the “something more” properties *interactive properties*.

The perceived use of a product by a human being relies on subjective understanding to a great extent. We therefore have to find ways to model the "hard" and the "soft" requirements, the technical and the interactive properties implemented in a product, and their relations. 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 previously presented technical interface models and interface models of interactive functions. This paper, furthermore, presents an integrated matrix-based representation that models the technical and interactive properties of a technical product and relates these properties to the customer requirements. The presented approach is exemplified with a recent design project.

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

1 Background

The general industrial demands on the product development process are to reduce the development time, improve the product quality, and reduce the corporate cost. As a direct consequence, industry is changing from a manufacturing economy to a digital economy, in which the processing of product knowledge plays an important role [1] and there is increased interest in platform-based product development. Product platform development includes the tricky task of answering questions on how to handle variety, how to structure products, and how to structure the associated digital models of the products. In addition, methods and techniques for handling different system structures and different models of these structures must be developed and thoroughly tested.

The research of Sellgren [2] and Blackenfelt [3] at KTH Machine Design provides important background to this work. Sellgren developed a general modeling principle for systems. This principle is based on a modular approach with strict definitions of how to deal with components, modules, and interfaces. Modularization has a close connection to platforms and product structures. Blackenfelt scrutinized modularization from different perspectives and studied both technical and business aspects of product structuring. Modular function development (MFD), a modularization principle developed at KTH Manufacturing Systems [4], supplied the foundation for Blackenfelt’s work. An important concept in MFD is that of module drivers, i.e. reasons for modularization. Blackenfelt concluded that modularization is done primarily in the embodiment phase and that “the detailing of the modular structure is done by optimizing the degree of variety and by freezing interfaces after the variety has been considered.” Building on this work, research at KTH Machine Design has focused on the interface problem, and the modeling principle proposed by Sellgren [2] has been further developed with a particular focus on interface modeling (e.g., [5]).

According to Lange [1], products are designed by someone to be perceived by someone. Lange’s research concentrated on design synthesis rather than on design analysis, which is the more common focus. He came to the conclusion that design is an act of semiosis and adopted “a designer-centered perspective of the design activity where it is the individual who has a perception of a product that shall be created.” It also takes a human to perceive value in the attractiveness of a product. We thus have to consider human aspects both as designer and as user

The attractiveness of a product is normally related to its aesthetics. However, this statement is an oversimplification, since the attractiveness of the product is related to how we perceive the product with all our senses in relation to what we expect to get from the product. This means that the technical functional requirements and the cost constraints must be fulfilled, but it also often means that the product must have “something more,” some syntactic, semantic, ergonomic, or aesthetic properties. Consequently a holistic view is necessary in the product realization process and both the technical and the “something more” aspects must be considered.

All product realization activities benefit from a holistic approach. Both the technical and the “something more” aspects must be considered simultaneously. We will name the “something more” properties “interactive properties.” Furthermore, modeling and simulation of products for their whole life is becoming increasingly important. We therefore have to find ways to model both the technical and the interactive properties of products. Our proposed solution to this problem is an integrated method that combines the general modeling principle previously proposed by Sellgren [2], the general theory of form design and functional surfaces developed by Tjalve [6], and the interactive functions discussed by Warell [7].

2 Functions and surfaces

2.1 Technical and interactive functions

The word "function" appears in many different situations, with different meanings. Here we focus on the functions of physical products. Product function is normally something that the product does or is intended to do. According to the working definition used in the Endrea program, *product function* is “*what a product or an element of a product actively or passively does in order to contribute to a purpose, by delivering an effect”* [7].

Warell [7] divided functions into technical functions and interactive functions. Technical functions are internal product functions while interactive functions are human-product functions.

Technical functions are associated with the flow, transformation, and storage of energy, materials, and information in the product. Typical basic technical functions include storing, transporting, transforming, supporting, and preventing something. These basic functions can also be expressed using synonyms or more natural formulations, such as enabling something to open or allowing something. 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 [7]. 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. 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.

2.2 Functional surfaces

Functional surfaces on technical products are *carriers of technical and interactive functions*. What we mean by functional surfaces on a product can be exemplified by the bottle opener concept in figure 1. 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. The function of that part is ergonomic, and thus the related functional surface is an *ergonomic functional surface*. The demand on the rest of the tool is 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 function of supporting the load is fulfilled. The surface of the middle part is thus a *communicating functional surface*.

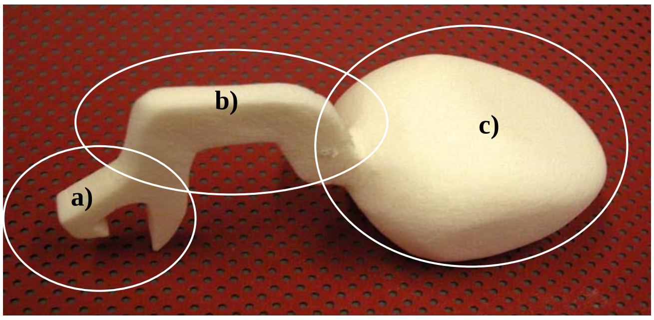


Figure 1. . Illustration of three functional surfaces on a bottle opener: a technical functional surface, a; an interactive (communicative) functional surface, b; and an interactive (ergonomic) functional surface, c. Modeled by a student in the Design and Product Realization Program at KTH. Photo: C.-M. Johannesson.

The two interactive surfaces can be formed in many different ways. Figures 1 and 2 show some of the concepts produced by freshman students in the Design and Product Realization Program at KTH.

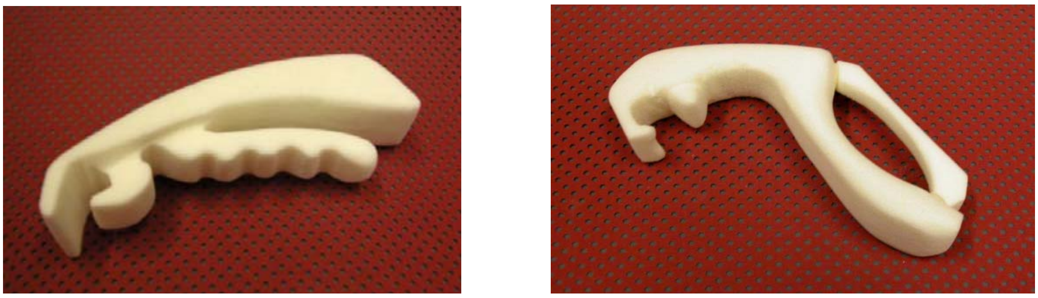


Figure 2. Two examples of bottle opener concepts developed by freshman students in the Design and Product Realization program at KTH. Photo: C-M Johannesson.

3 Technical systems and functional interfaces

A technical product can be viewed as a system, which can be defined as a set of subsystems or elements that are interrelated to each other and to the whole so as to achieve a common goal, or function. Even a simple product such as the bottle opener presented above can be looked upon as a system, for it comprises a front part, a middle part, and a rear part. Many subsystems are assemblies of machine elements. The technical function of a component often relies on mechanical contact relations within the component and between the component and surrounding components of the system, or on interaction relations between the component and the environment

3.1 Functional interfaces

Sellgren [2] defines an interface as an attachment relation between two mating faces. A mating face is typically a surface with an intended contact function on a component. In this paper, we refer to surfaces whose main purpose to interact with other surfaces or with a human as functional surfaces. *Functional interfaces of a product are the interfaces by which the different technical and interactive functions are actively performed when the product is in use.* Consequently, we can modify the original definition of a (functional) interface as follows:

An interface is an interaction relation between two functional surfaces.

This definition of an interface easily embraces all technical functional surfaces within the system and those in the environment that interact with the system. Furthermore, if we represent the human side of the human–product (or man-machine) interaction as an interactive functional surface, we can include all the interactive functions in the definition as well.

In summary, the following types of interfaces can be identified:

- *Technical interface* – a technical functional surface in or on a technical system that interacts with another technical functional surface within the technical system or in the environment.
- *Interactive interface* – an ergonomic or communicative functional surface on a technical system that interacts with a human through one of his or her senses.

3.2 Modeling and simulation of technical systems

Behavior and performance simulations of the entire product life cycle have become increasingly important. We therefore have to find useful ways to represent both the technical and the interactive product properties in our simulation models.

Simulation is here used to mean imitating how we perceive a real system by experimenting with computer models of the system and its interaction with the environment and human. We expect simulations to produce information that may be transformed to knowledge through analysis. Knowledge is required for making rational decisions. In order to simulate how a system behaves and how it is perceived in different situations and from different perspectives, an adequate and reliable model must be built for each situation studied. A flexible and consistent representation of the system and its models is therefore necessary. Model flexibility is generally not crucial for routine simulations, which normally involve properly defined events and a behavior that can often be simulated with rather simple standardized models. But a simulation that is addressing a non-routine design question usually involves many interacting components and several physical phenomena. Such a non-routine simulation normally relies on an iterative modeling and simulation activity that is significantly facilitated by a flexible model. The flexibility of a systems model is mainly determined by its architecture.

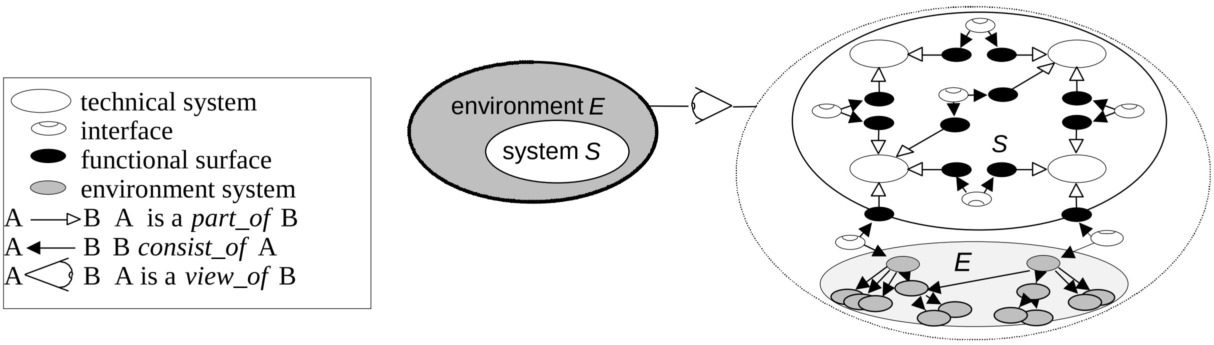
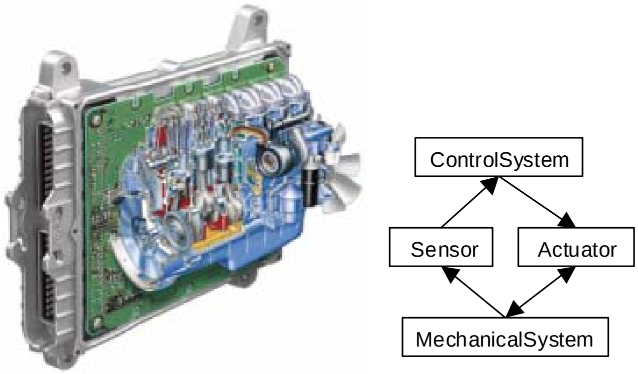


Figure 3. A system *S* as an aggregation of subsystems and interfaces in an environment *E*

Our principle is based on the modular approach previously proposed by Sellgren [2]. We look at a system as an aggregation of subsystems and interfaces (see figure 3). The subsystems have defined functional surfaces, which interface with other functional surfaces. Using a modular architecture allows a model to be changed easily.

The architecture of a systems model can 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) [8] provides a compact, complete, and clear representation of a complex system, but can be difficult to communicate to non-experts. Thus a combination of a graphical/symbolic representation and a matrix-based representation of subsystems and interfaces is 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 model.

Figure 4 shows an artistic VR representation, a graph, and a DSM representation of a mechatronic system with the four main components *MechanicalSystem*, *Actuator*, *Sensor*, and *ControlSystem*. The system can be a one-DOF servo system of an industrial robot. Both the graph and the DSM show a causal relation (i.e. a directed relation) between the sensor and the control system and a non-causal relation between the actuator and the mechanical system. A cross in the DSM matrix indicates an interface between two subsystems. Since the interface between the mechanical system and the actuator is a mechanical technical interface, the representation in the DSM matrix is symmetrical (action and reaction). However, for the interface between the mechanical system and the sensor, the representation is nonsymmetrical, since the sensor will record the state of the mechanical system but the mechanical system will not do the same in relation to the sensor.

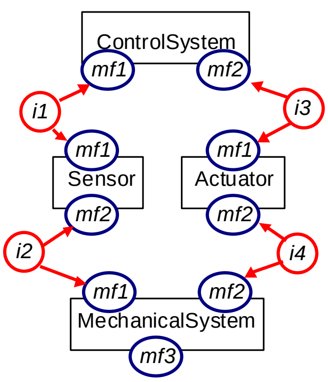


	ControlSystem	Sensor	Actuator	MechanicalSystem
ControlSystem	•	X		
Sensor		•		X
Actuator	X		•	X
MechanicalSystem			X	•

Figure 4. VR, graph-based, and DSM representations of a mechatronic system

4 The extended model structure matrix representation

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. [9], have argued that the DSM is a convenient and 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, an architecting tool known as the model structure matrix (MSM) has been developed [10]. The MSM, which is a model-based DSM, provides a compact representation of a complex model and its building blocks (i.e., subsystem models and interface models). Figure 5 shows an expanded graph and an MSM representation of the *MechatronicSystem* in figure 4. Each subsystem has defined functional surface models (*mf1*. etc. …) that are referenced by four interface models labeled *i1* to *i4*. The causality is an internal property of each interface. In the MSM representation, the four submodels, *MechanicalSystem*, *Actuator*, *Sensor*, and *ControlSystem*, and the functional surface models are labeled *M*, *A*, *S*, *C*, and *mf*, respectively, in the diagonal of the MSM. The four interface models are off-diagonal terms in the matrix. Non-causality is easily observed as a symmetric relationship, as in the two instances of *i4* in the matrix.



MechatronicSystem	#	1	2	3	4	5	6	7	8	9	10	11	12	13
ControlSystem	1	C												
ControlSystem.mf1	2		mf			i1								
ControlSystem.mf2	3			mf										
Sensor	4				S									
Sensor.mf1	5					mf								
Sensor.mf2	6						mf					i2		
Actuator	7							A						
Actuator.mf1	8			i3					mf					
Actuator.mf2	9									mf			i4	
MechanicalSystem	10										M			
MechanicalSystem.mf1	11											mf		
MechanicalSystem.mf2	12									i4			mf	
MechanicalSystem.mf3	13													mf

Figure 5. A graphic and an MSM representation of the architecture of a mechatronic system

We will now represent the model of the bottle opener in figure 1 with an MSM. The three subsystems in this case are the *FrontPart*, the *MidPart*, and the *RearPart*. We assume that the functional surfaces are included in 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 6, 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.

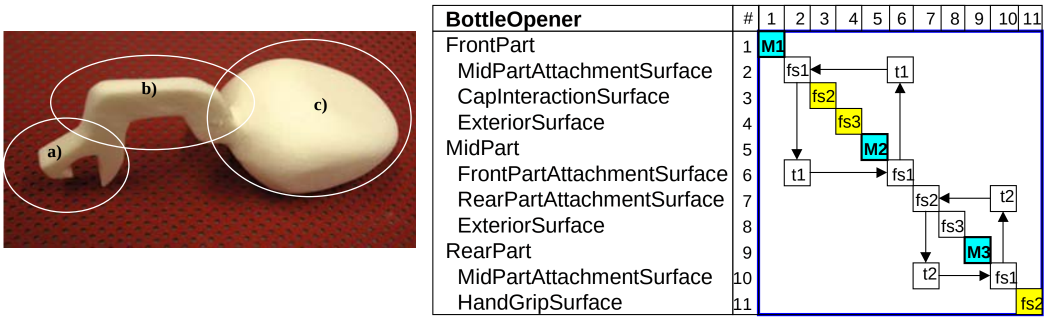


Figure 6. An MSM of the bottle opener technical system

If we add the environment (the bottle and its cap) and a human to the MSM matrix, as shown in figure 7, we have an extended MSM representation of a systems model that includes human, environmental, and technical systems, and the interaction between the three 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 6) has been decomposed into two functional surfaces, denoted *CapGripSurface* and *CapSupportSurface* in figure 7.

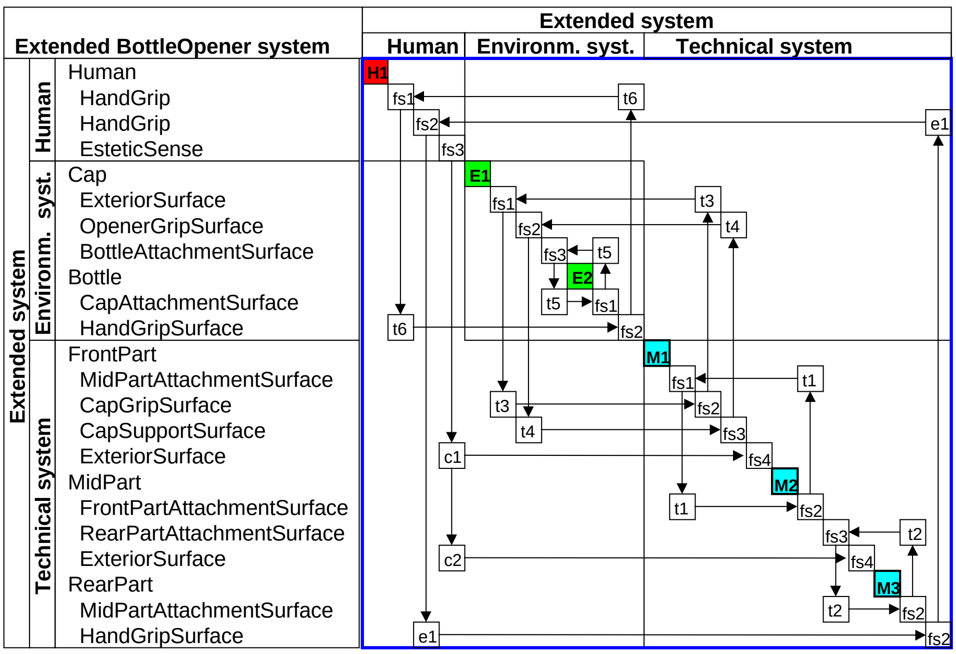


Figure 7. An extended MSM of the complete bottle opener system

4.1 An extended MSM representation of a new driver seat for trucks

We will exemplify the use of the extended MSM representation with a more realistic system, a new driver's seat for trucks developed by students in the Advanced Course in Machine Elements at KTH [11]. The final concept of the seat, which is shown in figure 8, consists of the main modules *Suspension*, *Frame*, and *Seat*. The design is unique in that the suspension module is placed behind the backrest and not under the seat, as is standard at present. The suspension module contains a spring-damper system for transverse vertical motion and for rotational motion. It has a cover for aesthetic and safety reasons. The frame is connected to the suspension and the seat is connected to the frame. The technical and interactive functional surfaces of the frame are properly designed. Besides the aesthetic aspect of the seat, a lot of attention was paid to its ergonomic functions. It is proposed that the main part of the seat be covered in a mesh fabric and have some ergonomic bags.

Figure 8. The driver's seat concept and its main modules (Henriksson et al. [11])

The principal structure of an extended MSM is shown in figure 9. The technical MSM is extended with submatrices representing the environment, the human (i.e., the driver and others who may have an opinion on the driver's seat), and the interactions between the human, the environment, and the technical system. An extended MSM representation of the driver's seat model is given in figure 10.

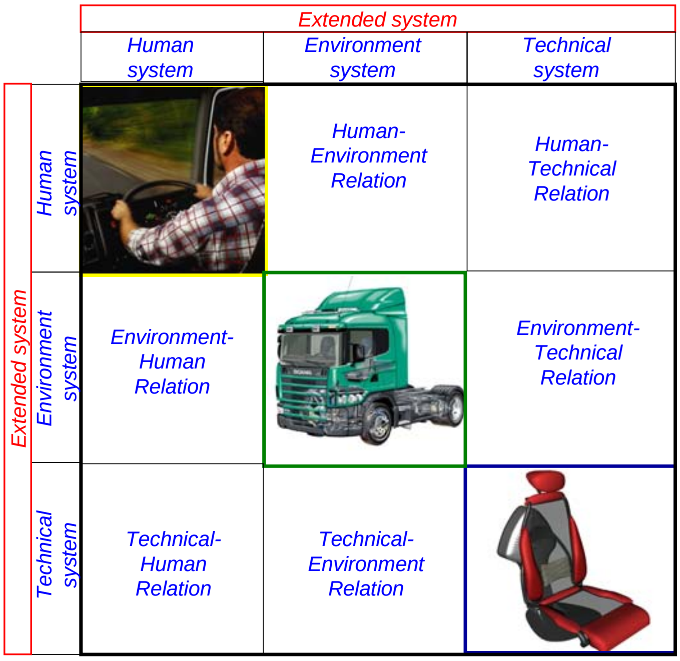


Figure 9. Principal structure of the extended MSM representation of the driver's seat

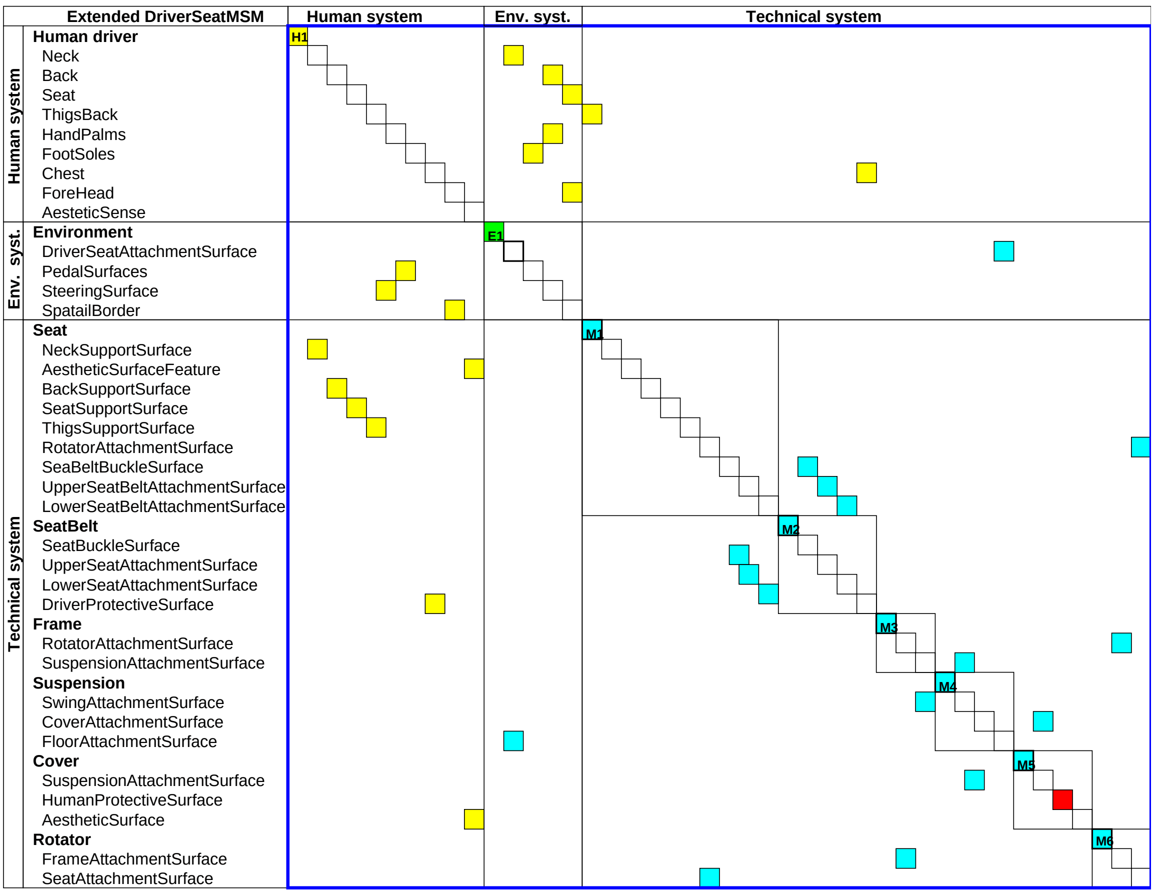


Figure 10. Extended MSM representation of the driver's seat shown in figure 7

5 The relation between extended MSM and customer requirements

On the previous pages, we have shown that a technical system can be modeled efficiently with its technical and interactive functions and represented as matrices. A design structure matrix (DSM) is a compact, complete, and clear representation of a complex system. To support efficient configuration of complex models, the MSM can be used as a navigation and architecting tool. MSM can be viewed as a model-based DSM. By extending the MSM 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 MSM makes the model representation more complete and situated than a model that is limited to technical objects.

The question is now:

> Can this modeling principle (i.e., representing models of both interactive and technical functions, interfaces, and functional surfaces by an extended MSM) be a useful support tool in product development?

Since we have only applied this principle retrospectively and not directly used it in a realistic development project, we are currently not in a position to give a reliable answer to that question. But we believe that the principle can be usable if there is a traceability mechanism or a representation that is capable of capturing the relations between the customer requirements and the extended MSM. In figure 11, we propose such an integrated representation, where the customer requirements (CR) are related to the objects of the extended MSM (eMSM) via functional requirements (FR) and a knowledge integration matrix, which we refer to as a function–means matrix (FMM).

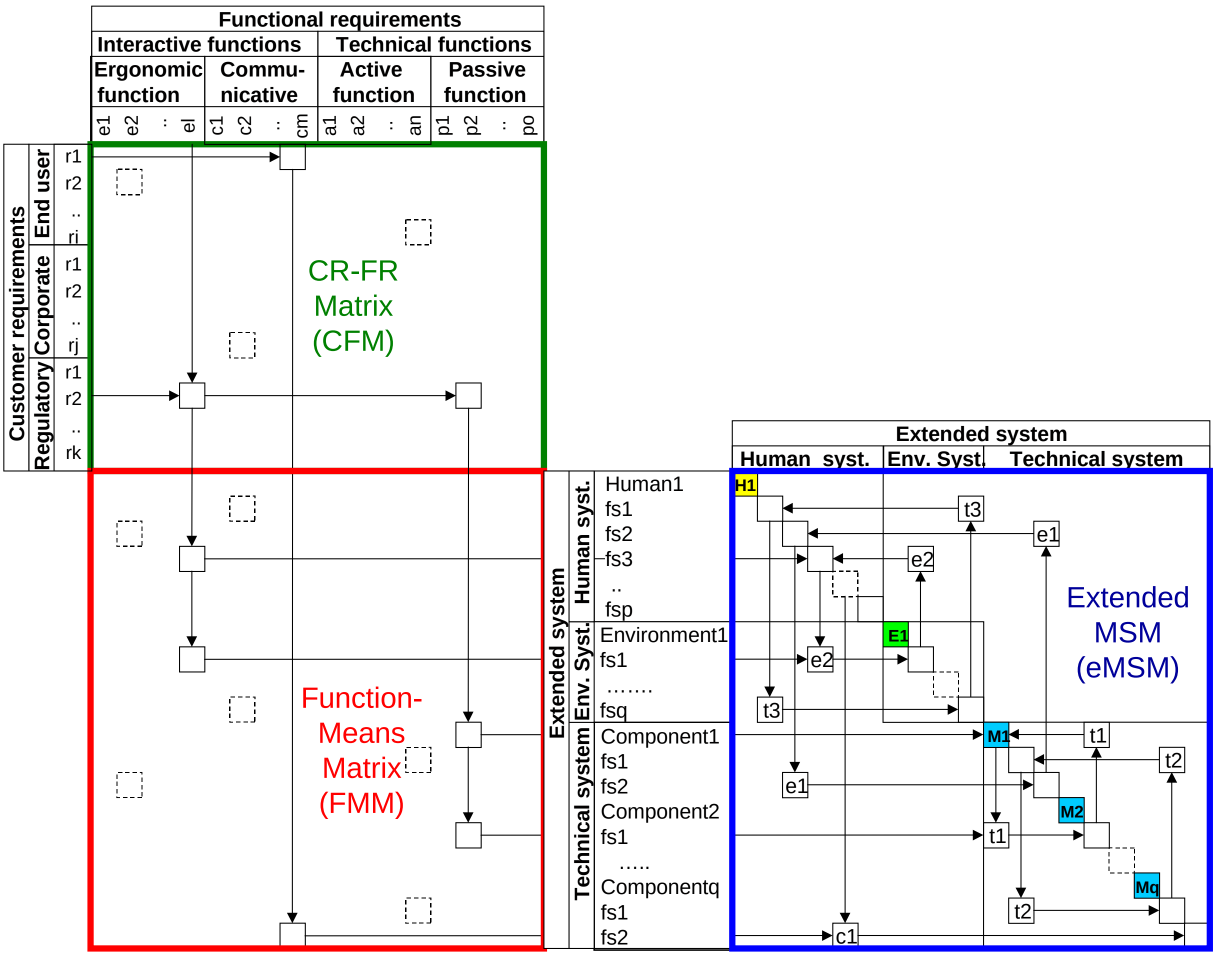


Figure 11. A structure of matrices linking CR and FR to an extended MSM representation of the design

Analysis of the development work for the driver's seat presented above reveals some interesting points. The seat was developed by a group of students for a truck company. The design was to be based on the demands and wishes of both the truck company and some drivers. The students thus undertook a minor market study and formulated a list of requirements that was confirmed by the company. Prior to the final formulation of the requirements, a summary of the demands and wishes was prepared and checklists and specifications, such as MIL standards and company-specific specifications, were studied. A structured and condensed version of the list of requirements is shown on the left in figure 12. The customer requirements were 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 driver seat is shown on the right-hand side of figure 12. The interactive functions were divided into ergonomic functions and communicative functions and the technical functions were divided into active and passive functions. The focus in the presented work has been on the interactive functions. The set of technical functions used in figure 11 can easily be refined and adapted to definitions published in the research literature, e.g. [12].

Customer requirements (CR)	Functional requirements (FR)
End-user requirements	**Interactive requirements**
Good sitting comfort	**Ergonomic requirements**
"No-swetting" fabric	Provide good sitting comfort
Seat squab width 490-510mm	Fit human back
Enough backrest support (for XX% drivers)	Enable alternate sitting position
Softer support from backrest upper part	Prevent sore back side
Good driving comfort	Provide adjustable seat width
Adjustable height (XX - YY mm)	Provide good driving comfort
Horizontal adjustment (XX - YYmm)	Protect against rough road
Good resting comfort	Enable good driving position
Easy seat entry/exit	Provide good resting comfort
Adjustable backrest angle (XX - YY degrees)	Provide adjustable backrest angle
Aesthetically attractive	Provide neck-rest
Innovative look	Provide good driving safety
Traditional truck-look	Facilitate seat-belt buckling
Safe look	**Communicative requirements**
Corporate requirements	Balance visual composition
Modular	Express ease of buckling
Standard attachment interface	Identify company brand
Lumbar support adjustable XXmm all markets	Express innovative design
Life-cycle issues	**Technical requirements**
Easy to disassemble for maintenance	**Active requirements**
Brand recognition	Provide vertical vibration damping
Recognizable and consistent company-look	Provide lateral vibration damping
Safety requirements	Provide sufficient suspension movement
At least 20 mm suspended travel	**Passive requirements**
Regulatory requirements	Provide standard interface with truck
Crash safety (Norm)	Provide durable structure
Vibration and vertical shock safety (Norm)	Facilitate assembly
Roll-over safety (Norm)	Provide clearance to surrounding components

Figure 11. A condensed list of customer (left) and derived functional requirements (right) for the driver's seat

By combining analysis of the results of the driver's seat design using the list of requirements and insights into the progress of the student project, we arrived at a preferred development procedure with seven distinct steps:

1. Collect the list of customer requirements and structure them (e.g., as end-user, corporate, and legal requirements). Create the CR vector.
2. Analyze the different requirements in terms of technical and interactive functional requirements. Create the FR vector and the CFM matrix.
3. Generate the base for the extended MSM with the basic human, environmental, and technical submatrices. Create the basic structure of the eMSM matrix, which will be expanded as the process continues.
4. 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. Expand the eMSM with these objects.
5. Relate the eMSM matrix to the functional requirements. Create the FMM matrix.
6. Start to generate technical concepts. Expand the technical eMSM submatrix with components and interfaces.
7. Proceed by decomposing the functional requirements and objects of the technical system. Update FR, CFM, FMM, and eMSM.

Following the conceptual design activities and some detail design, a design concept, such as the driver's seat, can be presented for go/no-go evaluation. Most 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 integrated 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 by Sellgren [2] and further developed in a previous paper by the authors [5]. 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 [6] theory of form design and Warell's [7] definition of interactive functions.

The work presented in [2] and [5] focused on technical interface models and technical functional surface models. The authors have previously modeled interactions between technical systems and the environment but not between these two domains and humans who use or perceive the product. 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., [5]), to represent purely technical systems.

In this paper, we have presented both an integrated matrix-based representation that models the technical and interactive properties of a technical product and relates these properties to the customer requirements, and 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 factor that makes for success 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 humans 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.

This approach has been studied by retrospectively analyzing a recent project. The results are tentative, but promising. We plan to scrutinize the approach and further develop it be more situated in upcoming projects. We also plan to close the loop from the eMSM 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," Doctoral Thesis, Dept. of Machine Design, KTH, TRITA MMK 2001:14, Stockholm, 2001.

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

[3] Blackenfelt, M., "Managing complexity by product modularisation – Balansing the aspects of technology and business during the design process," Doctoral Thesis, Dept. of Machine Design, KTH, TRITA MMK 2000:1, Stockholm, 2000.

[4] Erixon, G., "Modular function deployment – A method for product modularisation," Doctoral Thesis, Dept. of Manufacturing Systems, KTH, TRITA MSM R-98-1, Stockholm, 1998.

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

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

[7] Warell, A., "Design syntactics: A functional approach to visual product form," Doctoral Thesis, Chalmers University of Technology, Göteborg, 2001.

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

[9] 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.

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

[11] Henriksson, R., Ivarsson, J. and Wallentin, L., "INSIDEO," Dept. of Machine Design, KTH, TRITA MMK 2002:11, Stockholm, 2002.

[12] Stone, R., "Towards a theory of modular design", Doctoral Thesis, The University of Texas at Austin, 1997.

Sören Andersson
Royal Institute of Technology
Department of Machine Design
S – 100 44 Stockholm, Sweden
Phone: +46 (0)8 790 60 00
E-mail: soren@md.kth.se

Ulf Sellgren
Royal Institute of Technology
Department of Machine Design
S – 100 44 Stockholm, Sweden
Phone: +46 (0)8 790 60 00
E-mail: ulfs@md.kth.se