Motive-Oriented Design

Helping Automobile Engineers to Take the User's Perspective!

Constantin von Saucken, Ioanna Michailidou, Simon Kremer, and Udo Lindemann

Technische Universität München, Institute of Product Development, Germany
{saucken,michailidou,kremer,lindemann}@pe.mw.tum.de

Abstract. Modern car interiors are often overloaded and not self-explanatory. We supported running development projects within car industry and observed the following reasons: Similar functions are developed in different departments without a sufficient coordination and integration into the car. Functions are arranged according to technologies. Engineers have trouble with putting themselves in a user's position. Therefore, we present a motive-oriented approach: It supports engineers in taking the user's perspective by tools for investigating users' motives, clustering them in use scenarios, matching them with functions, illustrating them in an understandable way and running real-user tests.

Keywords: Emotional design and persuasion design, management of DUXU processes, mental model design, metaphor design, usability methods and tools.

1 Introduction

1.1 Motivation

User-centered design aims at improving the user's experience with a product – making it useful, usable and joyful. The real-world user should always be considered within the development process: by user insights, field observations and several evaluation loops. His needs and problems should be taken seriously [1].

In reality of automotive industry, engineers often design their products from a technological perspective. Products are developed on the basis of new technologies unaware of the impact on and benefit for the customer.

As a consequence, interfaces are mostly designed too complicated for most inexperienced users. Many different interface elements are all available at any time – even if they cannot be used in a specific situation, e.g. parking assistant while driving on the motorway or Cruise Control while driving in the city. Furthermore, the labeling of these elements is often poor – showing the abbreviated function or technology name, e.g. AC for air condition and not self-explanatory symbols like a snowflake.

When it comes to evaluating, engineers tend to see themselves as the potential user and design functions and interfaces accordingly. This perspective can be so extreme that real users who do not get on with the product get labeled as incapable, as if it was their own failure [1].

A. Marcus (Ed.): DUXU 2014, Part IV, LNCS 8520, pp. 370–377, 2014.

Fig. 1. Overloaded, not self-explanatory car interior, © by automobilemag.com

1.2 Goal

To meet these challenges, we want to support developers taking the customer's perspective by three integrated approaches:

Determine User Motives. Developers should always be aware of real users, what they want and what they are (not) able to handle. Therefore, the developer needs methods and tools to determine user motives based on real customers, trend reports and further sources.

Reduce Perceived Complexity. We see a high potential in reducing the perceived complexity of interfaces. Instead of mapping single functions separately with single buttons, the goal is to rearrange functions according to the investigated user motives. This way, users could easily understand what the fewer buttons are good for.

Provide Consistent Mental Model. We want to support the design of intuitive interfaces providing a consistent mental model. Users should understand them without a specific previous knowledge. We propose the use of metaphors and analogies instead of abbreviations of technology or function descriptions.

1.3 Overview

In this paper we present the motive-oriented design approach to help engineers broaden their perspective to a more user-centered one. In an automotive use case we describe our experiences applying it within a design team. The idea of our approach is to cluster functions and controls according to these user motives. A motive describes the cause of a user to perform an action [2]. For example the motive »delegate driving tasks« motivates the action »activate cruise control«. A motive is more abstract than action goals (»press cruise control button«) and more concrete than psychological needs (»feel secure«).

Furthermore, metaphors and consistent system descriptions allow the easy generation of a user's mental model. With this approach designers get encouraged to handle with the real users by taking their perspective on the product and designing it accordingly. This leads to understandable products that fulfill customer needs [3].

2 Method

We present a five step proceeding for developers to fulfill the motive-oriented approach. Figure 2 shows the first three of them. Starting point is the investigation of user motives and needs. We propose different methods and tools for deriving motives in form of »I want to…« from real users in section 2.1. In the next step these motives are clustered to consistent use scenarios resulting in a story which gives the impression of a solution and its experience (see section 2.2). Based on this story existing and new functions and technologies are selected to be integrated in one interface element. This step is described in section 2.3.

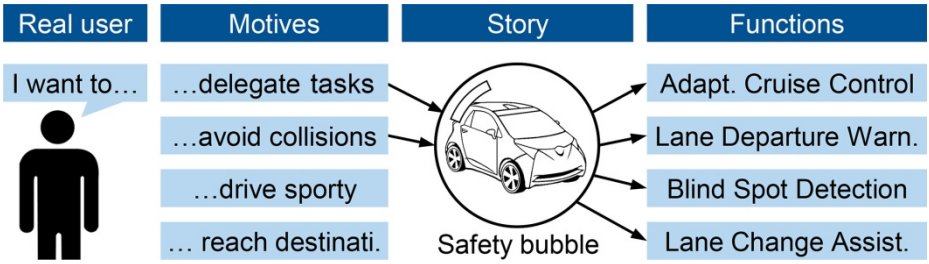

Fig. 2. Overview on motive-oriented approach for clustering functions

After having selected functions that can be integrated based on the story, an interface element needs to be designed accordingly that is easy to understand in order to create a consistent mental model (see section 2.4). Finally, this interface concept, the function cluster and the story as such need to be evaluated with real users. Therefore experience prototypes need to be tested. Section 2.5 explains this final step.

A positive example of a motive-oriented design in existing cars is the driving mode button (see figure 3). Many manufacturers provide an interface element that allows users to select »sport« or »eco« setup. This setup changes numerous settings of the engine and chassis fitting to the motive of »driving sporty« or »driving eco-friendly« but with only one interaction. At the same time, the description is intuitively understandable, even to new customers. Besides this example, we see a high potential for many more motive-oriented interaction concepts to improve the user's experience.

Fig. 3. Driving mode interface as a successful motive-oriented cluster, © by Volkswagen

2.1 Step 1: Get to Know the User Motives

In the first step we suggest methods to derive user motives from interviews and observations but also from literature-based trend research. The challenge is to interpret and translate the collected data and to formulate motives without fixation on existing solutions. The designer needs to uncover the real user needs and motives [3]. He must repeatedly ask for the reasons of a user's statement or behavior until the real desire becomes clear. We suggest the following steps for deriving user motives:

Define User Group. Before observing and asking the user the design team needs to know who the potential user is. Therefore, the market situation needs to be analyzed enriched by future trends: What will future regions, products, competitors and customers look like? We suggest existing marketing methods for this step.

Gather Current State of User Behavior. In this step the goal is to really understand the user defined in the previous step. We could observe several times that engineers tend to see themselves as potential users and therefore think they know whom to design for. But it is highly important to observe and ask real non-expert users from different cultural and social backgrounds. We suggest semi-structured interviews, the »think aloud« method and field operation tests to investigate the user behavior, his needs, problems and wishes.

Structure Data to Derive User Motives. The current state should be sufficiently extensive and thereby too complex to easily overview. For this reason, the insights get structured by paraphrasing and clustering in user motives in form of »I want to…« and corresponding needs, e.g. according to Sheldon et al. [3].

Formulate Chances. Mostly engineers think problem-driven – they are trained to be problem-solvers. In order to create a positive experience it is essential to formulate positively in terms of chances instead of problems. Finally, the user motives get enriched by future trends (e.g. Delphi analysis) based on literature or expert interviews.

As an example we derived the motives »I want to delegate tasks« and »I want to avoid collisions« in our supported industrial project.

2.2 Step 2: Cluster Motives in a Story

In the next step consistent use scenarios are developed which cluster these motives and chances and get illustrated using storytelling. Similar motives fulfilling the same psychological user need [3] are connected by developing a story that describes a positive experience. This story must be kept simple and easy to understand in order to convince other designers and decision makers. Furthermore, these stories must be evaluated with real customers. Finally, the most promising stories are chosen to be implemented.

Cluster Chances and Motives in Use Scenarios. As there should be many motives and chances derived in the first step (we gathered 34 in our project), they need to be further clustered and compared. We suggest to run a comparison in pairs and judge each combination with a value in the range from 0 (low similarity) to 3 (high similarity). Groups of motives with a high inter-relatedness should finally be connected by describing a consistent use scenario.

Create Stories. The most promising use scenarios now get illustrated by creating stories. We suggest applying the proceeding described by Michailidou et al. [4]: After creating the story setting (persona, their motives, environment) the plot is determined (interactions, products, events) and formulated as a text enriched with a sketch.

Evaluate Stories. These stories can afterwards be evaluated with potential users. These users should be selected carefully to ensure to have a representative result. After having read the story, the users are asked qualitatively with an unstructured interview: Did they understand the story? Is it realistic? What are first impressions? Additionally, we suggest running a quantitative questionnaire focusing the upcoming anticipated experience and emotions, e.g. AttrakDiff [5] or PrEmo [6].

Assess and Select Stories. Finally, the decision is made how many and which of the stories are assessed to be implemented in the next stage. We suggest applying a weighted score assessment [7] based on the evaluation results.

As an exemplary outcome we developed the story of a »repelling safety bubble around car« which combines several motives like e.g. »I want to avoid collisions«. This image shall give the impression of a security experience by avoiding any collisions in any direction around the car.

2.3 Step 3: Match Story with Functions and Technologies

The chosen motives now get matched with given technologies and functions. In this step, developers decide which functions and technologies fit to the selected stories. We suggest applying the Design Structure Matrix (DSM) method [8] for this step: in a table (named DMM – Domain Mapping Matrix) functions get listed in the first column, motives in the first row. The design team now decides for each motive which function fulfills it. Extended by the use scenarios, the DMM makes consistent function clusters obvious. In our example figure 4 shows a selection of functions that fulfill the two motives integrated in the »safety bubble« story.

	Functions	Adapt. Cruise Control	Lane Departure Warn.	Blind Spot Detection	Lane Change Assist.	Hill Descent Control	Traffic Sign Recogn.	Night Vision	Navigation System	Drowsiness Detection	Automatic Parking	Adjust chassis	Adjust engine setting	Speed Information
Motives	…delegate driving	X				X					X			
	…avoid collisions	X	X	X	X			X		X	X			
	…drive sporty									X		X	X	X
	…reach destinati.								X	X				

Fig. 4. Domain Mapping Matrix linking motives with functions

When the functions and components for integration are determined, the rather abstract story needs to be elaborated to a storyboard [4]. The design team must define every system behavior in every possible use scenario – always with the story in mind. The goal is to develop a system that reacts consistently according to the story. It is highly important also to design every misuse and to consider possible combinations of impacts on the system that could arouse a system failure. The resulting storyboard still needs to be illustrated in a simple way to be tested with real users.

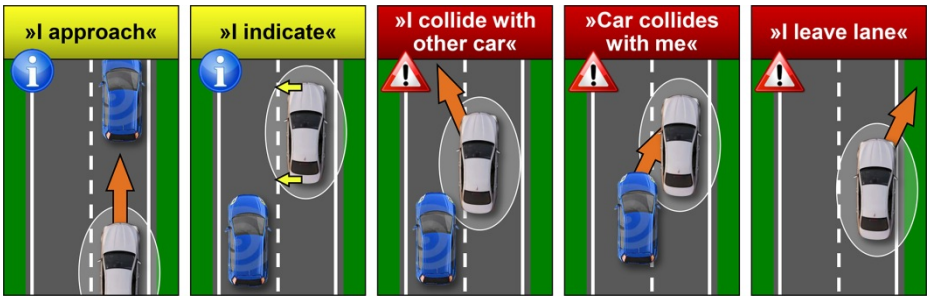

Fig. 5. Use scenarios for the »repelling safety bubble« story

In our industrial application we decided to integrate the existing advanced driver assistance systems (ADAS) »adaptive cruise control«, »lane departure warning«, »blind spot detection« and »lane change assistant« into a storyboard based on the »safety bubble« story showing different use scenarios (see figure 5).

2.4 Step 4: Design Interface According to Clusters

Finally, the user interface must be designed according to the developed story set of technologies, functions and motives. We suggest the use of metaphors to create an adequate mental model and a corresponding, self-explanatory system description.

In our example we suggested to show a protective circle around the car. This circle could have different colors: green for no potential dangers, yellow for potential threads and red for an immediate danger (cp. figure 5). Thereby, the interface could integrate four ADAS buttons. Figure 6 shows a similar implementation of BMW.

Fig. 6. Interface showing integrated safety systems, © by BMW Group

At this stage it is important to start prototyping the concept and interface taking other systems into consideration as all subsystems need to be integrated in one product in the end. This way, all system prototypes can be tested in a real environment by real users simulating the system behavior, e.g. by applying the »Wizard of Oz« method.

2.5 Step 5: Evaluate Concept with Real Users

The evaluation of developed motives, scenarios and stories with real users is highly important to meet real needs. These users are not to be found within the company. We observe that oftentimes engineers take themselves or colleagues as potential user. But as they deal with their product every day they are not representative and do not see obvious weaknesses. They do hard envisioning real users' problems and reactions on the product and tend to make them responsible, if the interaction fails. We suggest running tests during the whole development process – not only in the end:

Evaluate Stories and Storyboards. The evaluation of stories and storyboards is already described in section 2.2. We propose running qualitative unstructured interviews enriched with quantitative questionnaires (e.g. [5] [6]).

Evaluate Prototypes. We propose building different prototypes for agile testing: rough prototypes in early stages which show the basic function, detailed prototypes in later stages giving an impression of surface, material, design and feedback. These prototypes shall be tested by interviewing experts in the fields of engineering, usability and user experience. Furthermore, we highly recommend testing with inexperienced users by observation and the »think aloud« method enriched with tools used for the story evaluation. The focus of the evaluation lies on qualitative feedback in order to iteratively refine the prototypes.

Evaluate Integrated Product. Different integrated systems in a product finally need to be evaluated with real users. We propose running basic usability tests (visibility, packaging, accessibility), evaluating the integration concept (dissociation of interface elements and systems, no redundancy of functions) and check if the initial story is still preserved in the product. We propose using semi-structured interviews and semantic differentials for evaluating the users' perception and experience.

3 Insights from Industrial Use Case

The exemplary application shown in the sections of this paper is based on a real development project within German car industry, but we needed to adjust the contents due to nondisclosure agreements. We applied this approach in industrial projects and got positive feedback from engineers. The proceeding helped them to overcome fixations on existing solutions and to anticipate the customer's view on and potential problems with their product. During the project we gained several insights:

Challenge of Expressing Added Value. We met engineers that were not able to express the customers' benefit of their system – they »just« fulfilled technical requirements without consideration of potential users. In those cases it appears likely that the final user experience is rather random instead of systematically developed.

Too Late Integration. Oftentimes, interface designers and usability experts were integrated too late into the development process. For that reason, many solutions on the market provide a bad user experience due to an insufficient integration although the isolated system experience and its functionality are good.

Resistance Towards Cooperation. Our approach requires an intense cooperation and integration of systems developed by different departments. This coordination presents a challenge in large companies like car manufacturers and leads to high efforts. We could also observe political resistances towards a collaboration of different departments as there are (like in any company) conflicts of interests and relationships.

4 Conclusion

From the application of our motive-oriented approach within industrial projects we got the feedback that on the one hand the approach has a high potential for improving the company-internal communication and cooperation resulting in a better user experience of singular systems and the overall experience within a car. It helps dealing with customers' motives and helps integrating systems consistently on the story basis. On the other hand, it appears to be very challenging to implement this approach in a large company for the reasons described in section 3.

For this reasons, we need to apply the approach in further branches and companies with different conditions. Furthermore, we want to test other integration methods based on the DSM approach: Instead of investigating motives and clustering functions accordingly, the approach could be used to cluster according to situations or needs.

References

1. Norman, D.: The Design of Everyday Things. Basic Books, New York (2002)
2. Kaptelinin, V., Nardi, B.: Acting with Technology: Activity Theory and Interaction Design. MIT Press, Boston (2006)
3. Sheldon, K., Elliot, A., Kim, Y., Kasser, K.: What is satisfying about satisfying events? Journal of Personality and Social Psychology 80(3), 325–339 (2001)
4. Michailidou, I., von Saucken, C., Lindemann, U.: How to create a user experience story. In: Marcus, A. (ed.) DUXU/HCII 2013, Part I. LNCS, vol. 8012, pp. 554–563. Springer, Heidelberg (2013)
5. Hassenzahl, M., Burmester, M., Koller, F.: AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. In: Ziegler, J., Szwillus, G. (eds.) Mensch & Computer 2003: Interaktion in Bewegung, pp. 187–196. B.G. Teubner, Stuttgart (2003)
6. Desmet, P.: Measuring emotion: development and application of an instrument to measure emotional responses to products. In: Blythe, M.A., Monk, A.F., Overbeeke, K., Wright, P.C. (eds.) Funology: From Usability to Enjoyment, pp. 111–123. Kluwer Academic Publishers, Dordrecht (2003)
7. Lindemann, U.: Methodische Entwicklung technischer Produkte. Springer, Berlin (2007)
8. Lindemann, U., Maurer, M., Braun, T.: Structural Complexity Management: An Approach for the Field of Product Design. Springer, Berlin (2009)