Developing Intuitive User Interfaces by Integrating Users' Mental Models into Requirements Engineering

Diana Loeffler[1], Anne Hess[2], Andreas Maier[2], Joern Hurtienne[1], Hartmut Schmitt[3]

[1] Wuerzburg University
97074 Wuerzburg, Germany
diana.loeffler@uni-wuerzburg.de
joern.hurtienne@uni-wuerzburg.de

[2] Fraunhofer IESE
67663 Kaiserslautern, Germany
anne.hess@iese.fraunhofer.de
andreas.maier@iese.fraunhofer.de

[3] a3 systems GmbH
66130 Saarbruecken, Germany
hartmut.schmitt@a3systems.com

Today, the demand for software that is 'intuitive to use' is very high. In fact, this has become a determining factor for the success of a system. However, building software that is intuitive to use is challenging. This is particularly true for small and medium-sized enterprises (SMEs). They have to face a variety of problems to remain competitive: Usually no or just small staff is available that is specialized in user requirements engineering research, design, and testing. Furthermore, time schedules and budget are tight. All these factors require a method that delivers creative and intuitive-to-use software even with little design experience and expertise. In this paper, we address this problem by introducing a method for capturing and specifying the user's mental models with image schemas and image-schematic metaphors during the requirements engineering phase of a software engineering project. This method also enables SMEs to systematically transfer these elicited requirements into design solutions, which then result in software that is intuitive to use.

Keywords: Requirements Engineering, Intuitive Use, User Interface Design, Usability Engineering

1. MOTIVATION

The demand for intuitively usable software is very high [19]: Interactive software applications are becoming more and more complex; the time users are willing to spend on learning how to use a system is quite limited; and software is often used by very heterogeneous user groups - just to highlight some of the most important reasons. Although intuitive use has become one of the unique selling propositions of software systems, research has only recently started to formally define this concept [11].

According to the German IUUI (Intuitive Use of User Interfaces) research group, a technical system is intuitively usable if the users' unconscious application of prior knowledge leads to effective interaction [19]. Methods for measuring and designing intuitive use can be partially derived from the related research field on usability, but its focus is narrower. Whereas usability deals with effectiveness, efficiency, and satisfaction while using a system or product, intuitive use especially emphasizes the low mental effort required while interacting with a system or product [11]. For example, if an interaction is designed to be more click-intense in order to guide first-time users through their task (e.g., a wizard which is a time-consuming hindrance for frequent users), intuitive use can violate usability criteria of (physical) efficiency. However, more clicks to fulfil a task are

accepted when this leads to a decrease in mental effort for the user. Therefore, the concept of "intuitive use" is especially interesting when designed for beginners, rare users, diverse user groups which all need to work with the same system, or users who are unwilling to learn how to operate a product. Accordingly, intuitive use is to be seen as a sub-concept of the general concept of usability [8].

However, no consensus exists about how to develop software that is intuitive to use. Basically, designing intuitive-to-use interfaces means achieving a match between the user interface and the mental model of the user.

Design guidelines are available today, such as gestalt principles, affordances or consistency, that support designers in achieving intuitively usable user interfaces (see [3] for an overview). But these guidelines have a variety of disadvantages in their application, like being only suitable for specific user groups or offering only vague and general advice for the design of user interfaces. In particular, the step of transferring user requirements into design is underspecified and discussed under the term "Design Gap" [22]. Thus, the design outcome is highly dependent on the experience and expertise of the designer.

This is especially challenging for small and medium-sized enterprises (SMEs): Due to the lack of research capabilities and resources, low budget, and a small number of employees, SMEs are

usually unable to access state-of-the-art expert knowledge. Today's practice among SMEs shows that software engineers are often responsible not only for the technical development of a software system, but also for its design. This may result in suboptimal solutions because a professional design background cannot be expected [21]. One solution to tackle this challenge is to provide SMEs with methods that can be tailored to their specific engineering processes and that allow bridging the above mentioned "Design Gap" between requirements engineering and design. This can be achieved by providing support in translating the mental models of the user into design solutions - even if only little design experience is available. As a result, SMEs will be able to develop software that is more intuitive to use and thus increase their competitiveness. In this paper, the IBIS method (German for *design of intuitive use with Image Schemas*) is presented which provides a solution for bridging the aforementioned gap and adressing user requirements in terms of mental models of human-computer interaction (HCI). The IBIS method describes a user centred design (UCD) approach [14] with emphasis on requirements engineering [6] and elements borrowed from Rapid Contextual Design [9]. However, the innovation of this method compared to existing approaches lies in using image schemas [10] to capture and specify user's mental models during requirements engineering activities. These elicited requirements will then function as design guidance resulting in intuitive-to-use interfaces. Image schemas as the core concept of the IBIS method will be introduced in the following background section. The IBIS method is introduced in detail in section 3. Section 4 gives a summary of the evaluation of the IBIS method from the developers' perspectives. A more detailed discussion about the relation of the IBIS method to other UCD methods is provided in Section 5. The paper concludes with a summary and outlook on future work in Section 6.

2. BACKGROUND

Image schemas are core cognitive structures arising from physical and cultural experience (see the left column of Figure 1) and structure human perception [16].

Daily interaction with the environment leads to the formation of abstract, multimodal sensorimotor patterns of these bodily interactions, so-called *image schemas*. This includes simple concepts like CONTAINER, spatial dimensions like NEAR-FAR or UP-DOWN, and basic force dimensions like BLOCKAGE, MOMENTUM and COUNTERFORCE. Table 1 provides an overview of the most important image schemas according to Hurtienne et al. [11], grouped in seven categories.
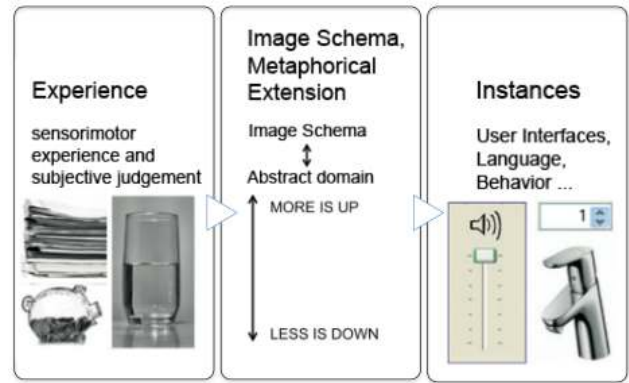

Figure 1: *Image schema method: Acquisition and instantiation of metaphorical mappings of image schemas, with examples*

Table 1: *Overview of the most important image schemas*

category	image schemas
Basic schemas	OBJECT, SUBSTANCE
Space	CENTER-PERIPHERY, CONTACT, FRONT-BACK, LEFT-RIGHT, LOCATION, NEAR-FAR, PATH, ROTATION, SCALE, SURFACE, UP-DOWN
Containment	CONTAINER, CONTENT, FULL-EMPTY, IN-OUT
Force	ATTRACTION, BALANCE, BLOCKAGE, COMPULSION, COUNTERFORCE, DIVERSION, ENABLEMENT, MOMENTUM, RESISTANCE, RESTRAINT REMOVAL, SELF-MOTION
Multiplicity	COLLECTION, COUNT-MASS, LINKAGE, MATCHING, MERGING, PART-WHOLE, SPLITTING
Process	CYCLE, ITERATION
Attribute	BIG-SMALL, DARK-BRIGHT, HEAVY-LIGHT, STRAIGHT, SMOOTH-ROUGH, STRONG-WEAK, WARM-COLD

By using the structure of image schemas, more abstract concepts can be described and understood. People perceive for example "quantity" on a vertical continuum: MORE IS UP – LESS IS DOWN. The concept of "importance" can be described along a continuum of IMPORTANT IS CENTRAL – UNIMPORTANT IS PERIPHERAL and the concept of "time" is perceived on a spatial continuum from FRONT (future) to BACK (past) [17, 15]. This mental process of coupling image schemas with abstract concepts is called the *metaphorical extension* of an image schema [4] (see also the middle column of Figure 1). Image schemas are also instantiated in language, for example in sentences like "Inflation is *rising* again", "That's just a *peripheral* issue" or "That's all *behind* us now". Since image schemas and image-schematic metaphors can be gathered by analysing natural language on a certain topic, access to the unconscious elements of users' mental models is gained, which can be used to design user interfaces accordingly. Figure 1 illustrates the

acquisition of the image-schematic metaphor MORE IS UP – LESS IS DOWN through the recurring shared experience of the image schema UP – DOWN with the abstract concept of "quantity" and the instantiation of the image-schematic metaphor in user interfaces, e.g. a vertical slider for volume control, a water tap or a spin box. The acquisition of the above mentioned image-schematic metaphors IMPORTANT IS CENTRAL – UNIMPORTANT IS PERIPHERAL and THE FUTURE IS FRONT – THE PAST IS IN THE BACK with the involved image schemas CENTER – PERIPHERY and FRONT – BACK is similar. More examples of derived image schemas, image-schematic metaphors and instantiations in user interfaces are introduced in section 3.2.

Originally the theory of image schemas has been developed in the field of cognitive linguistics [7, 15, 17]. Image schemas are encoded and retrieved from memory very frequently throughout lifetime. Thus, according to theory, they are processed unconciously, which makes them in turn interesting as patterns for designing user interfaces [11]. Since image-schematic metaphors are instantiated in language, they are accessible to analysis. If design decisions are guided by image-schematic metaphors, the users' mental model is translated into an intuitive-to-use user interface (see also the right column of Figure 1).

A variety of studies have already demonstrated the usefulness of image schemas for the design of interfaces, including full body interaction (e.g., [2]), graphical (e.g., [13]) as well as tangible user interfaces (e.g., [12]). Empirical data shows that users are faster, make fewer errors and prefer user interfaces that are consistent with image schemas – even though they look less familiar [13]. The design of a user interface with vertical buttons for example was tested using the UP – DOWN image schema. If button labels were compatible with the image-schematic metaphor (e.g., "good" is UP, "bad" is DOWN), reaction times were faster than with metaphor-incompatible labels (e.g. "good" is DOWN, "bad" is UP). The compatible labels were also preferred to enter data [10].

So far the image schema method has not been systematically integrated into a software engineering process and has not yet been tested outside the laboratory. However, literature supports the assumption that incorporating the promising image schema method into a software engineering process enables software engineers to systematically make decisions about the design that are less dependent on design expertise than on users' mental models of abstract domains. Image schemas can be used to transform requirements into design solutions and still leave enough space for creativity since they can be translated into several concrete designs.

The next section shows our effort in merging the image schema approach with a user-centred software engineering process with special emphasis on requirements engineering. The result is called the IBIS method, which can be completely or partly integrated into any software development process of any enterprise in order to use image schemas as design guidance for intuitive-to-use interfaces and to follow UCD principles.

3. IBIS METHOD

3.1 Development approach

The overall goal when developing the IBIS method was to integrate the image schema method into a requirements engineering and user-centred user interface (UI) design process according to ISO 9241-210 [14]. In order to achieve this, we also adopted elements of Contextual Design [9] and a customizable engineering approach to be applied in early development phases for interactive information systems [6], called "Satisfy".

From Contextual Design we adopted the Contextual Inquiry as a basis for eliciting language material about the user, user tasks and the users' work environment which can be used for image schema analysis. We also adopted the activity of low-fidelity prototyping and testing in order to evaluate the designs in early product stages. "Satisfy" enabled us to integrate detailed guidance to systematically elicit and refine system functionalities by first analysing supported stakeholders, their goals and tasks. This refinement is done over three different levels of abstraction according to the levels and decisions of the well-established Task-Oriented Requirements Engineering (TORE) approach [1]. Table 2 provides an overview of the IBIS method with the underlying UCD process according to ISO 9241-210, elements of Contextual Design, "Satisfy" and the integrated image schema method.

The resulting IBIS method basically comprises four successive phases as illustrated in Figure 2.

- *Preparation phase*, comprising all activities that have to be performed in order to prepare a contextual inquiry which is used for deriving image schemas and image-schematic metaphors.
- *Elicitation phase*, dedicated to conducting and transcribing a contextual inquiry.
- *Analysis phase*, comprising activities related to the analysis of the results of the contextual inquiry (i.e., interview transcriptions) in terms of image

Table 2: Overview of the IBIS method with elements adopted from UCD, Contextual Design, Satisfy and image schema method

IBIS Phase	Preparation	Elicitation	Analysis	Design
UCD phase	*Planning the human centred design process*	*Analysing the context of use*	*Specifying requirements*	*Producing design solutions, evaluating design decisions*
Contextual Design	Planning *Contextual Inquiry*	*Contextual Inquiry*	Work modelling Affinity Diagramming	*Low-fidelity prototyping and test*
Satisfy / TORE	*Project Initialisation*	*Stakeholder and Goal Analysis, As-Is Analysis*	*As-Is Analysis*	*To-Be and Interaction Analysis*
Image schema method			*Extraction of image schemas from users' language and formation of image-schematic metaphors, linking image-schematic metaphors to requirements*	*Designing user interfaces according to the image-schematic metaphors, testing if the interface is intuitive to use*

schemas, image-schematic metaphors, as-is activities / as-is processes, to-be activities and to-be processes, respectively.

- *Design phase*, aimed at designing, prototypically implementing and evaluating image schema-based UI designs.

The IBIS method has been defined in a flexible and lightweight manner so that it can be easily integrated into existing software engineering methods of SMEs. That is, depending on a particular software development process, the phases of the IBIS method can either be completely or partly integrated into the standard software development process of an enterprise.

In order to develop the IBIS method towards this goal, existing software development processes of two SME project partners have been analysed within the IBIS research project. Both project partners are established enterprises in the area of information and communication technology and create, implement and evaluate software systems for their customers.

Based on these analysis results, we then transformed the IBIS approach into a method description that enabled the SMEs to easily incorporate and apply the method in their daily business.

It should be noted that in case an enterprise wants to apply the IBIS method, it does not have to discard or completely change its standard software development process. The standard development process is rather enhanced with detailed information about the users' views on their tasks and the image schemas and image-schematic metaphors they apply while using a particular system. According to our experiences gained in case study projects, the IBIS method has proven to be easily integrable into existing development processes. This flexibility allows the IBIS method to become an optimal enhancement for software developing enterprises.

Finally, we defined detailed activity descriptions (introduced in section 3.2) as well as relevant roles involved in the IBIS method (introduced in section 3.3). In addition to this description, the participating SMEs were also provided with a 3.5-day tutorial where they received training in applying the IBIS method as a preparation for the two case study projects (see also Section 4). After that, the SMEs were able to conduct the IBIS method with all subsequent steps themselves.

In the following, we will describe the IBIS method in more detail by providing information about the activities within each of the four phases of the method and the relevant user roles which are involved in the execution of these activities.

3.2 Phases and Activities

In the following, goals and activities of each of the four phases of the IBIS method will be illustrated (see also Figure 2). Some of the activities are illustrated with examples from two case study projects with a real customer from industry (re-design of image manager / database (case 1) and customer / order administration software (case 2)) in which the usefulness and applicability of the IBIS method have been evaluated by each of the two SME partners within the scope of the IBIS research project (see also Section 4).

In the **preparation phase,** the requirements engineer *elicits initial requirements* in talks with the customer. These requirements are mainly related to (1) the goals to be achieved within a project, (2) information about the end-users, and (3) current processes to be supported by the system. Information about future end-users is important in order to recruit suitable participants for the end-user interviews (contextual inquiries) that will be conducted in the elicitation phase.
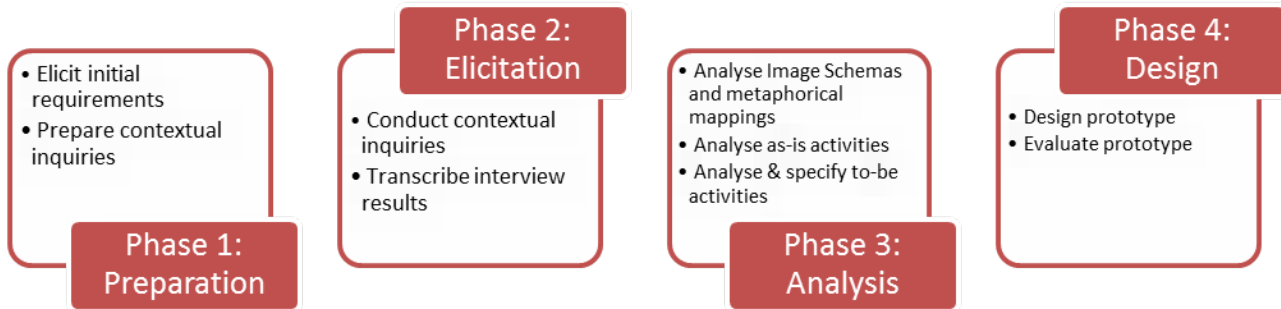
Phase 1: Preparation
- Elicit initial requirements
- Prepare contextual inquiries

Phase 2: Elicitation
- Conduct contextual inquiries
- Transcribe interview results

Phase 3: Analysis
- Analyse Image Schemas and metaphorical mappings
- Analyse as-is activities
- Analyse & specify to-be activities

Phase 4: Design
- Design prototype
- Evaluate prototype

Figure 2: IBIS Process Phases and Activities (Overview)

During the talks with the customer it should be discussed and agreed on whether the customer has access to real end-users or whether representatives have to be recruited. Initial information about current processes is also very helpful to prepare the contextual inquiries.

Besides the elicitation of initial requirements, the *preparation of the contextual inquiry in form of end-user interviews* belongs to the activities of the preparation phase. This activity includes (1) planning and preparing the interview setting and defining the guidelines based on the initial requirements, (2) selecting and recruiting suitable end-user representatives, and (3) preparing technical setup (e.g., audio and / or video capturing) and organizational issues (e.g., scheduling dates) for the interviews.

To support an easy integration of the IBIS method into existing development processes, this initial requirements elicitation activity has been defined in a lightweight manner. That is, this activity is not intended to represent a detailed requirements analysis or completely substitute existing requirements engineering activities. This activity is rather intended to raise awareness and highlight important requirements that are relevant for the subsequent analysis and that should explicitly be considered during existing requirements elicitation activities and possibly supplement typically elicited requirements.

The main activity during the subsequent **elicitation phase** is *conducting the contextual inquiry* with end-users. During these interviews, the interviewer observes the end-users in their work environment while they execute typical work tasks that are to be supported by the system. While executing these tasks, the end-users are asked to think aloud [18]. In parallel to the observation, the interviewer asks questions in order to identify any issues that arise while the end-user is accomplishing the tasks. Besides a detailed understanding of how processes and tasks are currently performed, the language and speech material collected from the end-users during these interviews based on statements, "thinking aloud", and answers to the questions constitutes very important and valuable output of

this activity. Based on this material, image schemas and image-schematic metaphors are derived. After the interviews have been performed, *the speech records are transcribed into digital text form* to allow further processing during the subsequent analysis phase.

During the **analysis phase**, image schemas and image-schematic metaphors are derived from the transcribed interview records.

This analysis is in fact the most crucial activity during the method. It requires the image schema expert (see role description in section 3.3) to analyse all sentences spoken by the end-users regarding image schemas and image-schematic metaphors. In order to support the process of identifying image schemas in a transcribed interview, the enterprises were provided with a list of keywords which point to certain image schemas. In parallel to this activity, the requirements engineer analyses the interview records to *identify and derive as-is processes and as-is activities* performed by the end-users while executing the tasks observed during the interview. This is in fact a very helpful activity, as it helps to identify any drawbacks and problems within the current situation that could be handled by the system to be developed. Activities that are currently performed manually could for example be supported by the system in the future. In addition, based on these as-is activities, *future to-be processes and to-be activities are derived and specified* as use case descriptions as well as system function descriptions.

Finally, image-schematic metaphors that are suitable for the implementation are prioritized and afterwards mapped to those use cases where they fit according to the interview. Together with the use case descriptions they become part of the requirements specification document.

The following example will illustrate this analysis activity. Within the scope of the first case study project aiming to re-design an existing image manager / database (case 1), the interviewees used the MATCHING image schema to describe the relation between photos in the database and the related meta data. This image schema has been

identified in the following statement of an interviewee: "I <u>compared</u> images with corresponding meta data and uploaded them into the database", leading to the formation of the image-schematic metaphor PHOTOS AND RELATED META-INFORMATION ARE MATCHING. Furthermore, since the image database and also the process of editing a picture were always described with the CONTAINER image schema ("I open the picture <u>in</u> the database"), the image-schematic metaphor IMAGE DATABASE AND PHOTO EDITING ARE CONTAINERS was documented.

In the second customer project (case 2) aiming to develop software supporting customer / order management, the interviewees raised statements like "Now I need to <u>compile</u> the desired products into an order." The analysis of this statement revealed the image-schematic metaphor SELECTING PRODUCTS FOR AN ORDER IS MERGING. Some users also talked about the order process of using the PATH image schema: "Everything is done manually - <u>from</u> receiving a call of a client <u>to</u> finally forwarding the order <u>to</u> the haulier."

In the final **design phase**, *prototypes of the product are implemented* based on the use cases, the system functions, and the image-schematic metaphors identified during the analysis phase.

Since image-schematic metaphors are a common vocabulary for describing users' mental models as well as user interfaces, they offer concrete guidance for interface design, yet leave enough space for the creativity of the designer.

The initial prototypes may be low-fidelity prototypes (like pen-and-paper prototypes) or high-fidelity prototypes and are used for evaluating basic concepts.

Afterwards, *the prototypes are evaluated* by the evaluator. The SMEs were provided with an Excel-based toolbox to evaluate how intuitive-to-use their concepts are. The toolbox consisted of questionnaire-templates and instructions how to measure effectiveness, mental efficiency and user satisfaction. By filling out the Excel-templates of the toolbox after the evaluation of the prototype was finished, scores were automatically calculated and reported. It should be noticed that the activities of designing and evaluating can be done iteratively.

The image-schematic metaphor (case 1) PHOTOS AND RELATED META-INFORMATION ARE MATCHING led to the design decision of highlighting a selected image in the same color as the corresponding meta data which was displayed in another frame (Figure 3). This emphasized that both belong together, although they could not be presented directly next to each other since this would have cluttered the central frame. Based on the image-schematic metaphor IMAGE DATABASE AND PHOTO EDITING ARE CONTAINERS, the whole database screen was framed, implying that the user was looking inside the database. Images located on the hard drive

were presented to be outside of the database container (i.e., outside of the frame). The function of photo editing was designed as a box which opened during mouseover, indicating that images can be placed inside via drag and drop. This result differed from the conventional design in which photo editing options open when images are selected and a button labeled "photo editing" is pressed.
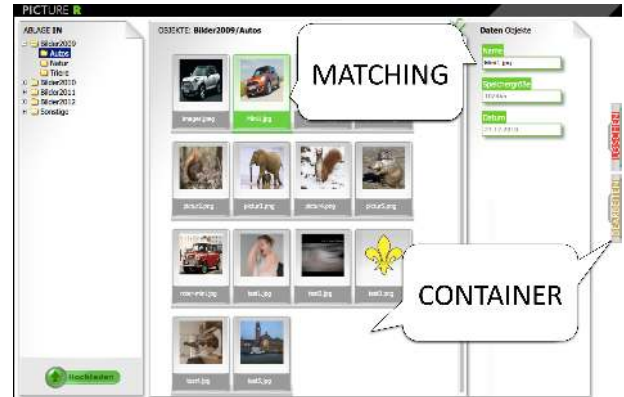

Figure 3: *Example Design Case 1 (*PHOTOS AND RELATED META-INFORMATION ARE MATCHING, IMAGE DATABASE AND PHOTO EDITING ARE CONTAINERS*)*

The image-schematic metaphor (case 2) SELECTING PRODUCTS FOR AN ORDER IS MERGING led to the design of a table with three columns. The left and right column contained different product types and a drag and drop operation allowed to pull products into the middle column which then contained the selected products for the order. This design differed from the conventional solution which showed all products on one screen. Selection was done by clicking on radio buttons next to the products. The PATH image schema which described the ordering process was implemented as a wizard, guiding the user through all steps required to enter the order into the system. The conventional solution was a mask containing all input fields within a single screen (Figure 4).
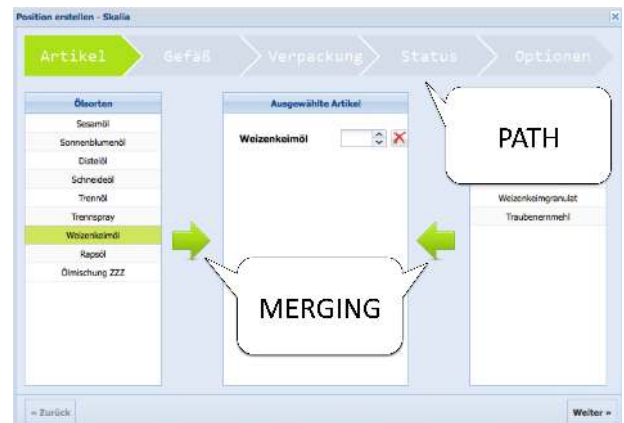

Figure 4: *Example Design Case 2 (*SELECTING PRODUCTS FOR AN ORDER IS MERGING, ORDERING PROCESS IS A PATH)

The results of the IBIS method are processed in the enterprise's standard development process. That is, the IBIS method ends with the implementation of the functional prototypes and their evaluation. The market-ready product implementation takes place in the standard software development process of an SME. This allows a better flexibility of the overall software development process.

3.3 Involved Roles

In order to provide a quick overview of required skills and expertise allowing to perform certain activities within the method appropriately, we identified nine different roles that are involved in the IBIS method. This makes it easy for software engineers to allocate responsibilities since all the activities are linked to the role descriptions. According to the qualifications, software engineers might pick roles and will know the corresponding activities in which their roles are involved. It is important to know that this does not necessarily imply that nine different software engineers have to be involved in order to apply the IBIS method. It might even be beneficial if a single person takes over several roles. To assume a certain role, additional training might be required (such as image schema expert). However, this additional effort does not significantly influence the overall efficiency of the IBIS method as it has been evaluated in the project [23]. Some roles are represented by external stakeholders (customer, end-user) or can be taken over by third parties (e.g. transcriber). Each of the roles is briefly introduced in the following:

The **customer** states functional and qualitative requirements within a project as well as constraints related to time and budget. Furthermore, the customer provides information about the end-users of the product to be developed, has to accept the final product, and is the main contact person for both developers and designers in terms of product acceptance and evaluation.

The **requirements engineer** is responsible for the elicitation of initial information and requirements (e.g., in talks with the customer) and for the analysis and specification of further and detailed requirements like as-is processes, to-be processes, and use cases identified during the application of the method.

The **interviewer** is responsible for the preparation and performance of the contextual inquiries. In order to conduct these inquiries, it is recommended that the interviewer has experience in preparing interview guidelines and in conducting interviews.

The **end-user** plays a crucial role within the method. He is involved both in the contextual inquiries and in the product evaluation. It is very important for the success of the method application that the prospective end-users participating in the

contextual inquiries represent the future group of end-users in the best possible way. We emphasize that this role cannot be combined with any other role in order to engage the involvement of real end-users.

The **transcriber** is responsible for the transfer of an audio record of the contextual inquiries into a digital form. As a qualification, the transcriber should be skilled in transcription techniques. Optionally, the role of the transcriber might also be replaced/supported by transcribing software or a transcription service.

The **image schema expert** analyses the transcriptions of the contextual inquiries and identifies image schemas and image-schematic metaphors. Furthermore, he / she maps the image-schematic metaphors to the use cases identified and specified by the requirements engineer. In terms of qualifications, the image schema expert has to know the concepts of image schemas and the process of metaphorical mapping, the majority of existing image schemas and needs to speak and understand the end-users' language.

The **designer** creates ideas for the interaction design and the visual design of the product based on the image schemas and image-schematic metaphors that the image schema expert has identified.
Afterwards, the designer implements his / her ideas in the form of low-fidelity prototypes. In terms of qualifications, the designer has to be able to derive design solutions for the functionality to be supported by the system based on the image schemas and image-schematic metaphors.

The **developer** implements functional prototypes based on the ideas and the low-fidelity prototypes of the designer. In terms of qualifications, the developer has to be able to implement the ideas of the designer accurately.

The **evaluator** is responsible for the preparation, conduction, and analysis of the end-user evaluation of the prototypes. Furthermore, the evaluator has to check if the image schemas and image-schematic metaphors were adequately implemented and applied. It is recommended that the evaluator is skilled in evaluation techniques (e.g., expert walkthroughs, usability testing).

4. QUALITATIVE ANALYSIS OF THE IBIS METHOD

The usefulness and applicability of the IBIS method was intensively evaluated throughout the IBIS research project. For this purpose, a team of developers at each of the two SME partners' sites applied the IBIS method within the scope of a case study project with a real customer from industry. In parallel, another development team at each partner's site followed their established development process to address the customer's

goals. Any interaction between the two teams at each SME was strictly forbidden throughout the complete customer projects. This procedure finally allowed a comparison of the resulting prototypes at each SME in end user studies.

Besides these comparisons at the end of the project, the IBIS method was also continuously evaluated throughout the case study projects by regularly filling out feedback questionnaires at the end of each of the four IBIS phases (see Figure 2). These questionnaires captured data about the experiences that the development team made while applying the IBIS method, such as occuring problems, required efforts, ideas for improvements, etc. Moreover, at the end of each case study project, the developers were asked to retrospectively analyse the IBIS method from their perspectives. For this purpose, a conjoint workshop with researchers and practitioners was organized and conducted [24]. In this workshop, the participants summarised their experiences and lessons learned regarding the general applicability and usefulness of the IBIS method, as well as their ideas for further improvements. The following statements reveal some impressions of the development teams' attitudes towards the IBIS method:

- "The development team that applied the IBIS method was more efficient in various project phases and was overall more structured on their procedures than the development team that followed the established process."
- "The time required to plan, conduct and transcribe the contextual inquiries was strongly underestimated. In the future, we will be more efficient based on our lessons learned."
- "One of the interesting observations was that the IBIS method has positively influenced the way of thinking and the way how to cope with changes in the requirements. While the IBIS team analysed change impacts on the design of the user interface (in particular on the metaphor realization), the other development team analysed change impacts on the underlying data models, performance issues, response times, etc."
- "Incorporating changes identified in the final end user studies required more effort in case of the IBIS prototypes than in case of the "traditional" prototypes. This observation might be attributed to the fact that the IBIS prototype is based on a more complex implementation that often combines unusual combinations of UI elements."

5. COMPARING IBIS TO OTHER UCD APPROACHES

According to Norman's UCD considerations [20], a User Centred Design should "follow natural mappings between intentions and the required actions; between actions and the resulting effect; and between the information that is visible and the interpretation of the system state." The IBIS method helps find out which mental mappings between the system and the real world are made by users. This is a prerequisite to follow a UCD approach and a technique that supplements Norman's approach. Norman implicitly postulates an intuitively usable user interface in order to achieve a UCD: "Design should make use of the natural properties of people and of the world: it should exploit natural relationships and natural constraints. As much as possible, it should operate without instructions or labels. Any necessary instruction or training should be needed only once". Again, a concrete technique on leveraging this in the design is lacking. We claim that this gap is bridged by the IBIS method. The need for the use of knowledge and ideas of the users and their mental conceptions of the system's functionality is emphasized in principle 1 "Use both knowledge in the world and knowledge in the head" and principle 4 "Get the mappings right" of Norman's "Seven Principles for Transforming Difficult Tasks into Simple Ones". But how to get mappings right?

In order to get mappings right, knowledge in the head must be known sufficiently. With the IBIS method, this knowledge can be derived from the utterances made by users within interviews. The image schemas and image-schematic metaphors identified in these utterances show sufficient evidence for the user's view on the system and its functionality and is therefore suitable for becoming aware of the user's knowledge and mental conceptions. With this awareness, we ensure a basis for the matching of a design model and the user's model as well as for a proper system image. Thus, the IBIS method covers the aspects of mental models as distinguished by Norman. However, the system designer is responsible for the development of a proper system image that is consistent with the user's model according to Norman. The user's involvement in the system's design or a requirements elicitation with interviews, workshops, etc. is not required by his UCD approach. Here, the IBIS method shows the largest difference to the UCD approach. According to Norman, the user's conceptions and wishes are not incorporated into the system, but the system is designed in a way that only allows actions which succeed, nonetheless in due consideration of the user requirements. All in all, Norman's UCD approach requires a rethinking by the system designer: he designs for the user, not for himself. The IBIS method not only supports the system designer and this kind of rethinking, but exceeds Norman's UCD approach by providing methods for the appropriate involvement of user conceptions and wishes, as demanded by [14]. The IBIS method supports the creation of intuitively usable

UIs instead of restrictive UIs as suggested by Norman's UCD approach.

Other UCD approaches like Cooperative Design [5], Participatory Design [8], and Contextual Design [9] follow the Human-Centred Design standard by ISO 9241-210. According to this standard, a design is user-centred if: "The design is based upon an explicit understanding of users, tasks and environments. Users are involved throughout design and development. The design is driven and refined by user-centred evaluation. The process is iterative. The design addresses the whole user experience. The design team includes multidisciplinary skills and perspectives." [14]. Most of these principles are satisfied by the IBIS method. Thus, the IBIS method is an instantiation of the HCD proposed in [14]. The activities that have to be performed during a human-centred design of a system are: "Plan the human-centred design process, understanding and specifying the context of use, specifying the user requirements, producing design solutions, evaluating the design" [14].

The IBIS method mainly differs in the activities 1 and 2. Interviews with users help to identify their views on their tasks and the context of use. Derived image schemas and image-schematic metaphors specify the users' conceptions and are used to specify the user requirements. The innovative design solutions produced by applying the IBIS method cover the requirement of taking into account the whole UX given by [14], but additionally follows the psychological factor "curiosity", emphasized by Norman. Cooperative Design and Participatory Design follow the principle of actively involving all stakeholders of the product under development. In the IBIS method, this is guaranteed by the incorporation of the "Satisfy" process [6], where one of the first activities is to identify supported stakeholders (end-users) and their tasks.

The main principle which Contextual Design follows is that detailed information about how users interact with a product has to be captured in the field where they work and where they will use the product under development by observations and conversations. The IBIS method adopts the contextual inquiry from Contextual Design, but afterwards focuses on image schemas instead of affinities and the creation of various work models. The creative part of visioning in Contextual Design is substituted by the straight forward identification of image schemas for particular activities in the IBIS method.

6. CONCLUSIONS AND FUTURE WORK

"Intuitive use" has become a buzzword when talking about interactive technology, especially software. It has become a determining factor for the success of a system [11]. Yet there is a lack of guidance on how to design intuitive-to-use products. We believe that such guidance may enable enterprises to develop innovative and intuitive-to-use software products that will allow them to compete with the highly competitive market of software-based products. This paper makes a contribution in terms of incorporating an approach for designing intuitive-to-use products into the software engineering process.

The IBIS approach makes use of so-called image schemas, which are core cognitive structures of human perception [16]. As image schemas function as a common vocabulary for describing users' mental models and user interfaces [11], they can bridge the design gap [22]. By integrating image schemas into the requirements engineering phase, they can later be used for deriving design solutions that fit the mental models of the users and thus lead to software that is intuitive to use. Especially when design expertise is scarce as it may be the case in many SMEs, software developers can profit from the design-facilitating effect image schemas provide. The user-centred IBIS method is flexible enough to be completely or partly integrated into the standard software development process of any enterprise. As a result, by applying the IBIS method a UCD approach will be used to develop software-based products with image schemas as design guidance.

As mentioned in this paper, the resulting software prototypes that were developed in the two case study projects following the IBIS method have been compared to the prototypes according to the UCD processes that have been established at each partner SME. The results of this comparison revealed that the IBIS prototypes were more intuitive to use and thus justify to be determined extra effort and expenses.

In the near future it is planned to apply the IBIS method in further projects and continuously investigate and improve the method (e.g., regarding the optimisation of the cost / benefit ratio). Furthermore, limitations of the approach (e.g., factors like framework restrictions on development that limit effects of image schema instances in user interfaces) need to be discussed and investigated in more detail.

ACKNOWLEDGMENT

REFERENCES

[1] Adam, S., Doerr, J., Eisenbarth, M. & Gross, A. (2009). Using Task-oriented Requirements Engineering in Different Domains - Experiences

with Application in Research and Industry, *RE '09. 17th IEEE International*, pp. 267-272.

[2] Bakker, S., Antle, A. N., & van der Hoven, E. (2009). Identifying embodied metaphors in children's sound-action mappings. *In P. Paolini & F. Garzotto (Eds.)*, Proceedings of the 8th International Conference on Interaction Design and Children, pp. 140-149.

[3] Blackler, A. & Hurtienne, J. (2007). Towards a unified view of intuitive interaction: definitions, models and tools across the world. *MMI-Interaktiv*, 13, pp. 36-54.

[4] Grady, J. (1997). *Foundations of meaning: Primary metaphors and primary scenes.* PhD dissertation. University of California, Berkeley.

[5] Greenbaum, J., Kyng, M. (Eds.) (1992). *Design at work - cooperative design of computer systems*. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, USA.

[6] Gross, A., Adam, S., Doerr, J., Eisenbarth, M., Kerkow, D., Riegel, N., Trapp, M., Uenalan, O. (2011). Satisfy – An Integrated RE/UE Approach for Interactive Systems. Fraunhofer IESE. IESE Report 2011.11/E.

[7] Hampe, B. (2005). *From perception to meaning: Image schemas in cognitive linguistics*. Mouton de Gruyter, Berlin.

[8] Helander, M. G., Landauer, T. K., Prabhum P. V. (Eds.) (1997). *Handbook of Human-Computer Interaction* (2nd ed.). Elsevier, Amsterdam.

[9] Holtzblatt, K., Wendell, J. B., Wood, S. (2005). *Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design*. Morgan Kaufmann, San Francisco.

[10] Hurtienne, J. (2009). *Image schemas and design for intuitive use. Exploring new guidance for user interface design*. Doctoral dissertation, Technische Universität Berlin.

[11] Hurtienne, J. & Blessing, L. (2007). *Design for Intuitive Use – Testing image schema theory for user interface design*. In: ICED 07 Paris, 16th International Conference on Engineering Design, Proceedings of the conference P_386, pp. 1-12. Ecole Centrale, Paris.

[12] Hurtienne, J., Israel, J. H., Weber, K. (2008). *Cooking up real world business applications combining physicality, digitality, and image schemas.* In: Schmidt, A., Gellersen, H., van den Hoven, E., Mazalek, A., Holleis, P., Villar, N. (Eds.) TEI'08, pp. 239--246. ACM Press, New York.

[13] Hurtienne, J., & Langdon, P. (2010). Keeping warm in winter: Image-schematic metaphors and their role in the design of central heating controls. *In Fourth International Conference of the German Cognitive Linguistics Association*, Bremen, 2010, pp. 53-54. University of Bremen, Bremen.

[14] International Organization for Standardization (2010): ISO 9241-210: Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems.

[15] Johnson, M. (1987). *The body in the mind: The bodily basis of meaning, imagination, and reason.* University of Chicago Press, Chicago.

[16] Lakoff, G., & Johnson, M. (1980). *Metaphors we live by*. University of Chicago Press, Chicago.

[17] Lakoff, G., & Johnson, M. (1999). *Philosophy in the flesh: The embodied mind and its challenge to Western thought*. Basic Books, New York.

[18] Lewis, C. H. (1982). *Using the "Thinking Aloud" Method In Cognitive Interface Design* (Technical report). IBM. RC-9265.

[19] Mohs, C., Hurtienne, J. , Kindsmüller, M. C. , Israel, J. H. , Meyer, H. A. & die IUUI Research Group (2006). IUUI – Intuitive Use of User Interfaces: Auf dem Weg zu einer wissenschaftlichen Basis für das Schlagwort "Intuitivität". *MMI-Interaktiv*, 11, pp. 75-84.

[20] Norman, D. (1988). *The Psychology of Everyday Things*. New York, Doubleday.

[21] Oberquelle, H. (2005). Gestaltung von Benutzungsschnittstellen: Geht es ohne Designkompetenz? *In: Cremers, A.B., Manthey, R., Martini, P., Steinhage, V. (Eds.). Informatik 2005. Informatik LIVE!*, Vol. 1, pp. 232-236. Gesellschaft für Informatik, Bonn.

[22] Wood, L. E. (1998). *User Interface Design: Bridging the Gap from User Requirements to Design*. CRC Press, Boca Raton.

[23] Fetzer, K., Hess, A., Lange, K., Loeffler, D., Maier, A., Schmitt, H., Weber, S. (2013). *Schlussbericht Gestaltung intuitiver Benutzung mit Image Schemata* (Final report), (www.ibis-projekt.de)

[24] Bjarnason, E., Hess, A., Doerr, J., Regnell, B. (2013). *Variations on the Evidence-Based Timeline Retrospective Method.* EUROMICRO Conference on Software Engineering. and Advanced Applications. (SEAA), 2013