Proceedings of the 5th International Conference on Applied Human Factors and Ergonomics AHFE 2014, Kraków, Poland 19-23 July 2014
Edited by T. Ahram, W. Karwowski and T. Marek

Levels of Abstraction for User Modeling in the Usability Engineering Repository UsER

Amelie Roenspieß, Marc Paul, Tilo Mentler and Michael Herczeg

Institute for Multimedia and Interactive Systems (IMIS)
University of Lübeck
Ratzeburger Allee 160, 23562 Lübeck, Germany

ABSTRACT

User analysis and user modeling provide important information for the development of usable interactive computer systems. The Usability Engineering Repository (UsER), a web-based collaboration platform, supports analysis, design, and evaluation in the corresponding processes. In order to take specific users into account, UsER integrates modules covering various methods for user modeling within a development process. Users can be modeled on several levels of abstraction as personas, stereotypes or user classes. How they fulfill their tasks within their work environment may be outlined in organizational charts, task models, or scenarios. Mapped onto a linear documentation structure, model entities and descriptions can be interlinked to point out semantic relationships, ease navigation and allow tracing of user goals and requirements throughout the user-centered development process.

Keywords: Usability Engineering, User Analysis, Computer Supported Cooperative Work, Human Centered Design

INTRODUCTION

Understanding the users is one of the most fundamental challenges of Usability Engineering (Nielsen, 1993). Misunderstanding them is one of the main reasons for failing software development projects (ISO 9241-210, 2011). This implies that end users have to be taken into account early during the development of an interactive system. Who will use the system, which tasks will they have to perform, and how shall they be supported? We will outline basic approaches, explaining the necessity as well as the general concept of user analysis and user modeling. When user models are available, how can they be managed in a way that they can be considered and reflected upon throughout the development process in a way that the resulting system will meet the requirements? This is where tool support is required that goes beyond simple documentation and enables the integration of user models into the whole development process.

We introduce the Usability Engineering Repository UsER (Herczeg et al., 2013), which has been designed and implemented for collecting, structuring, and organizing analyses, concepts, as well as other artifacts in the development process, using established methods as well as innovative ones. After a general summary of the system's features and the rationale behind it, the module for user analysis and modeling will be described in detail.

USER ANALYSIS

User analysis aims to developing software that helps users attain their goals effectively, efficient, and in a satisfying way – in a word, usable software. As a piece of software is usually used by lots of different people it would not be sensible to optimize it for just one specific person. Instead, appropriate representatives of the intended users have to be taken into account right from the start; they have to be analyzed and modeled considering their specific characteristics, including their tasks, goals, and intentions as well as their skills and experiences, needs and expectations, also reflecting their mental models (Herczeg, 2001; Herczeg, 2009). Figure 1 shows an overview of different approaches to this issue which vary especially in their level of detail.

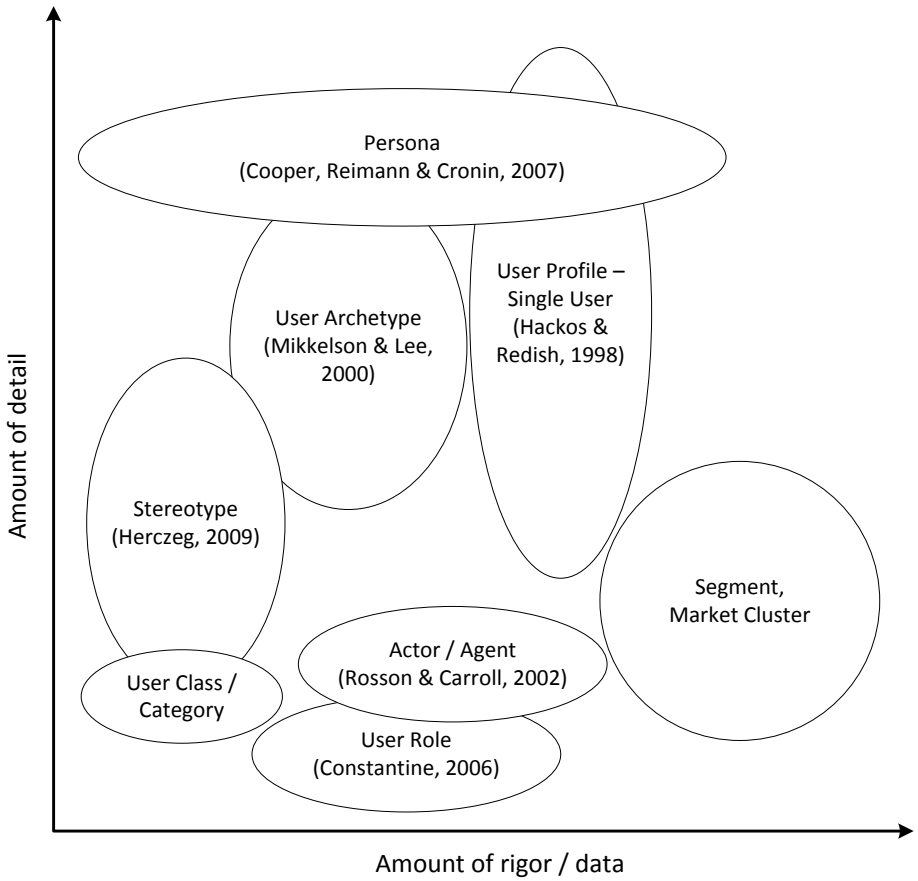

Figure 1: User representations with different level of detail (Pruitt & Adlin, 2006, p. 99; Hüttig, 2012, p. 15)

User classes can be considered a first basic step of dealing with the variety of users, gathering those with similar characteristics concerning their goals, experiences, tasks or organizational roles. Classification criteria may vary depending on the context and the system to be developed. *Stereotypes* can be used to give more precise impressions of user classes and usually only have a name that triggers certain associations instead of precise characteristics (Herczeg, 2009).

The *actors* described by Rosson & Carroll (2002) in their scenarios are pretty much replaceable, only communicating that there are users a little more specific than their role. *User archetypes* (Mikkelson & Lee, 2000) and *user profiles* (Hackos & Redish, 1998) however, are more precise descriptions of potential users.

An even more detailed approach is the use of *personas*, focusing on the user goals of concrete but fictitious users (Cooper, 1999; Pruitt & Adlin, 2006; Mulder & Yaar, 2007; Cooper, Reimann & Cronin, 2007).

Personas

Personas can be described with name, age, nationality, family status, language skills, profession, expertise, interests, attitudes, and especially their goals (Cooper, 1999; Cooper, Reimann & Cronin, 2007). The most common types of personas are the following:

- *primary personas*, representing the most important user classes;
- *secondary personas* show special aspects of the most important user classes;
- *supplemental personas* that might help solve specific issues;
- *customer personas* as the ones who buy the system, but do not use it;
- *served personas* who will not use the system themselves, but will be affected by it as they are served by its users; and
- *negative personas* who will not be users of the system at all, defining the scope of whom the system is actually developed and optimized for.

There are different opinions on how personas should be developed: one is the *Persona Lifecycle* as presented by Pruitt and Adlin (2006), another one is the idea of *Goal-Directed Design* (Cooper, 1999; Cooper, Reimann & Cronin, 2007). The latter one focuses on the user goals which are not to be mistaken for the user tasks: Tasks are considered as single process elements that need to be dealt with in order to achieve a certain goal. With the choice of means, the tasks and the process may vary, while the goal remains the same.

Developing personas according to Goal-Directed Design starts with research, case studies, interviews or *Contextual Inquiries* (Beyer & Holtzblatt, 1998; Holtzblatt, Burns Wendell & Wood, 2005) analyzing potential users in their context of work. In order to map these observations of real persons onto fictitious personas, Cooper, Reimann, and Cronin (2007) suggest using goals and behavioral variables which help identify clusters of user attitudes toward certain aspects (Figure 2).

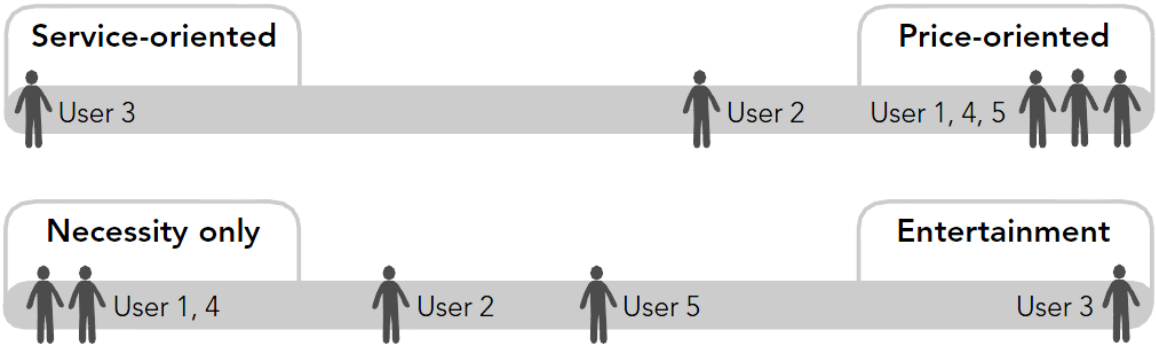

Figure 2: Behavioral variables (Cooper, Reimann & Cronin, 2007, p. 99)

After the personas have been created and validated, *Scenario-based Design* (Rosson & Carroll, 2002) can be applied, resulting in persona-based scenarios which then can be used for developing the system and testing it against them. Still, personas are only one possible aspect in creating usable software successfully; they must be considered carefully and with an awareness of quality, like every other development artifact or method. Their flexible applicability, however, allows for direct combination with methods like task analysis, requirements management or participatory design (Pruitt & Grudin, 2003).

THE USABILITY ENGINEERING REPOSITORY USER

The Usability Engineering Repository UsER is a modular, web-based tool for collaborative analysis, design, documentation, and evaluation of interactive systems (Roenspieß, 2011; Kammler, Roenspieß & Herczeg, 2012; Paul, Roenspieß & Herczeg, 2013). It provides several modules supporting different methods of Software Systems Engineering, Requirements Engineering and Usability Engineering (Nielsen, 1993; Mayhew, 1999) that can be used in established software development processes as well as innovative ones. UsER provides one central access point for all development documents and artifacts – no more searching for the latest revision of scenarios, requirements or specifications. The integrated tools can be combined flexibly according to project specific needs and all the information gathered and documented in the process can be retraced and reused – no more wondering where certain requirements originate from and whether they are qualified.

While integrating many different connectable modules, UsER still provides a linear document structure for the project overview, as seen on the left in Figure 3. Modules can be combined and rearranged via drag&drop from the central column like chapters as needed in this linear structure, concealing the potentially complex hypermedia topology lying beneath. The latter one becomes visible where needed, typically in the modules or chapters themselves, providing access to associated information. The collections of modules on the right side of Figure 3 can be reused as a structure without content for other projects, e.g. for recurring technical specifications or contracts.

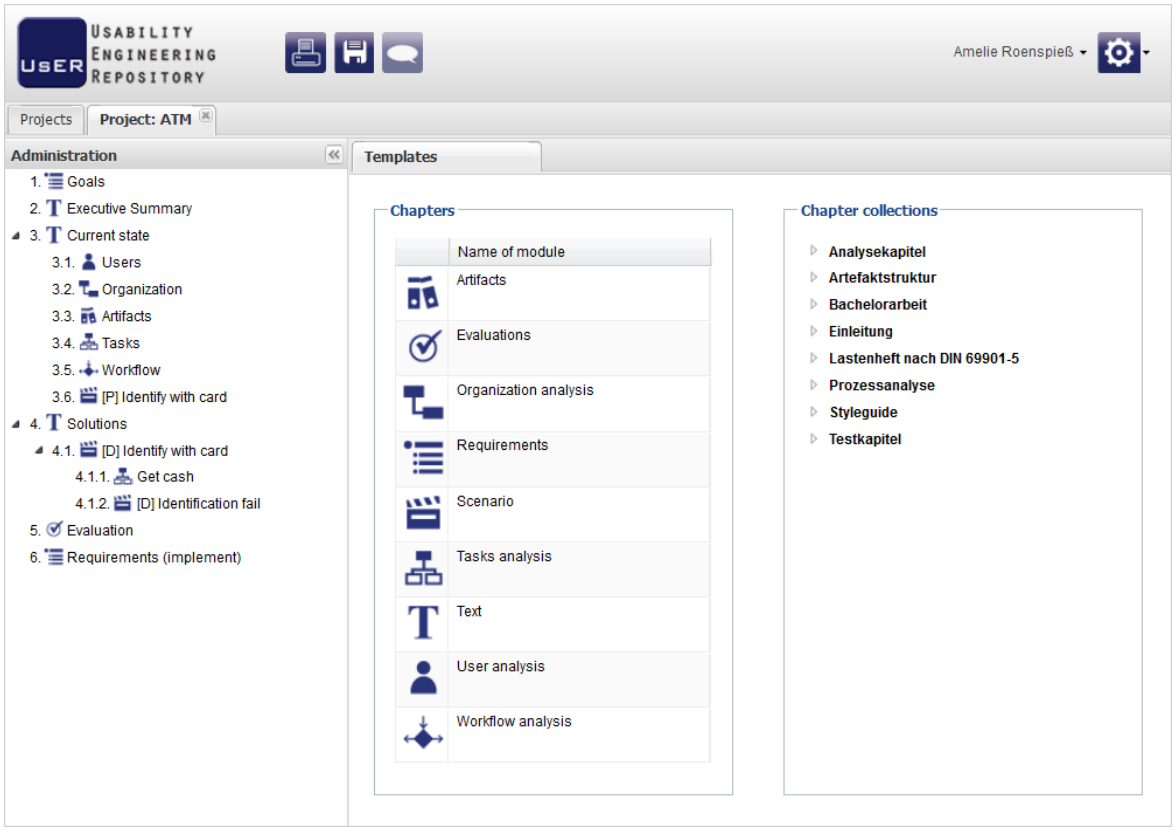

Figure 3: Project view in UsER with the different module templates

Every piece of information gathered during the process of analyzing, designing, implementing, and evaluating the system can be managed in UsER and linked to all other information. A subtask of the hierarchical task analysis (HTA) could be described in a scenario which mentions users and their work items, while each of those is further depicted in their own module: The attributes, features or states of a work item are specified in the artifact module; the user is linked to its description in the module for user analysis, where – amongst others – his goals and requirements are described, which themselves are specified in detail in the requirements module. The following modules are currently integrated into the systems; others are yet to come.

Artifact modeling: Work items of every kind can be described in this module with attributes and values – documents like receipts and contracts as well as technical instruments or any other artifact.

Evaluation: This module is designed for constructing and managing questionnaires, distributing them to test persons for evaluations and visualizing the results for interpretation.

Organization analysis: Structures and functions for organization charts with different organizational units are provided by this module.

Requirements management: All requirements accumulated over the development process can be managed in this module, including detailed description, refinement, and links to other elements.

Scenarios: Verbal description of tasks or contexts, enhanced with pictures, eases the communication between the involved parties, as they do not require any additional formal or syntactical knowledge. Scenarios can be specified with different properties like pre- or postconditions in this module.

Task analysis: In this module, tasks can be depicted in hierarchical task trees, so called HTAs (Annett & Duncan, 1967; Kirwan & Ainsworth, 1992).

Rich text: Text passages, e.g. for introducing other chapters, can be written in this module without requiring any specific attributes or structure.

User analysis: The module for user analysis provides different methods for representing users and will be described in detail in the next section.

Process modeling: Another representation of tasks and processes can be chosen in this module in the form of workflow diagrams according to a subset of the BPMN 2.0 standard.

In order to integrate customer and co-worker feedback, UsER provides a feedback function that allows annotating elements. Not sure if that section of the problem scenario is detailed enough? Noticed a wrong turn in the workflow model or a conceptual flaw in the user description? The task description is lacking an important aspect? Just open the annotation area, write down your comment and address it to the responsible person who should reconsider it.

Figure 4 shows how the different UsER modules could be used in a sample process on the way from first ideas to the release of an interactive application.

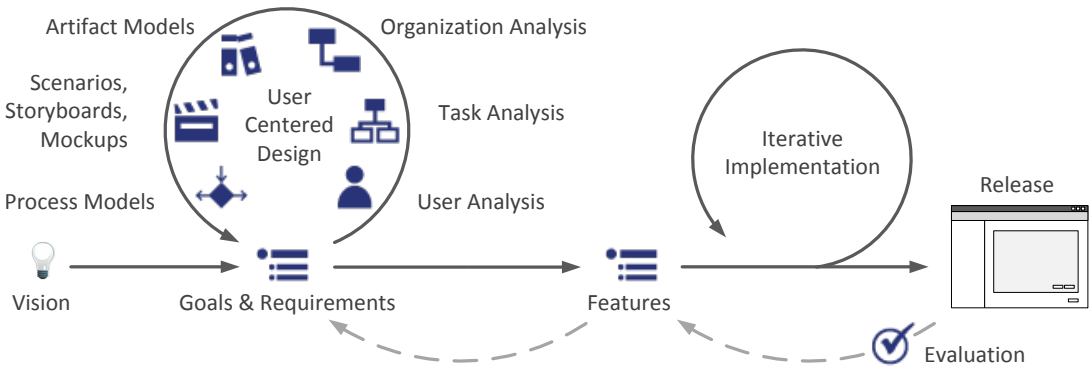

Figure 4: Sample development process with the UsER modules

UsER MODULE FOR USER ANALYSIS AND MODELING

One major issue that makes it so difficult to integrate knowledge and findings about users into conventional software engineering processes is the lack of methods for recording user goals, user analyses, and user models, map them onto requirements and, on the other hand, trace them back to the users in order to evaluate how well their claims have been met. The Usability Engineering Repository UsER has been constructed for that very purpose: enabling the applicability of *User Centered Design* (Norman & Draper, 1986), *Contextual Design* (Beyer & Holtzblatt, 1998; Holtzblatt, Burns Wendell & Wood), *Scenario-based Design* (Rosson & Carroll, 2002), *Goal-Directed Design* (Cooper, 1999; Cooper, Reimann & Cronin, 2007) as well as other usability methods in the process of software engineering.

The first version of the module for user analysis was just a persona editor, where personas (Cooper, Reimann & Cronin, 2007) could be developed and referred to from other modules. But considering the effort required for producing a viable persona, the module was unlikely to be used at all. Therefore, a more flexible mechanism for efficiently characterizing users has been added, inspired by the idea of classifying user descriptions on different levels of detail (Pruitt & Grudin, 2003; Pruitt & Adlin, 2006; Herczeg, 2009). Now it is possible to specify user classes either as roles or as target groups and personas can be assigned to them. Figure 5 shows an overview of the user classes in one specific UsER project and the users that are related to them. In standard setting those are tied to one project, but users as well as user classes can also be specified as global content in order to be reused in other projects that are managed by UsER.

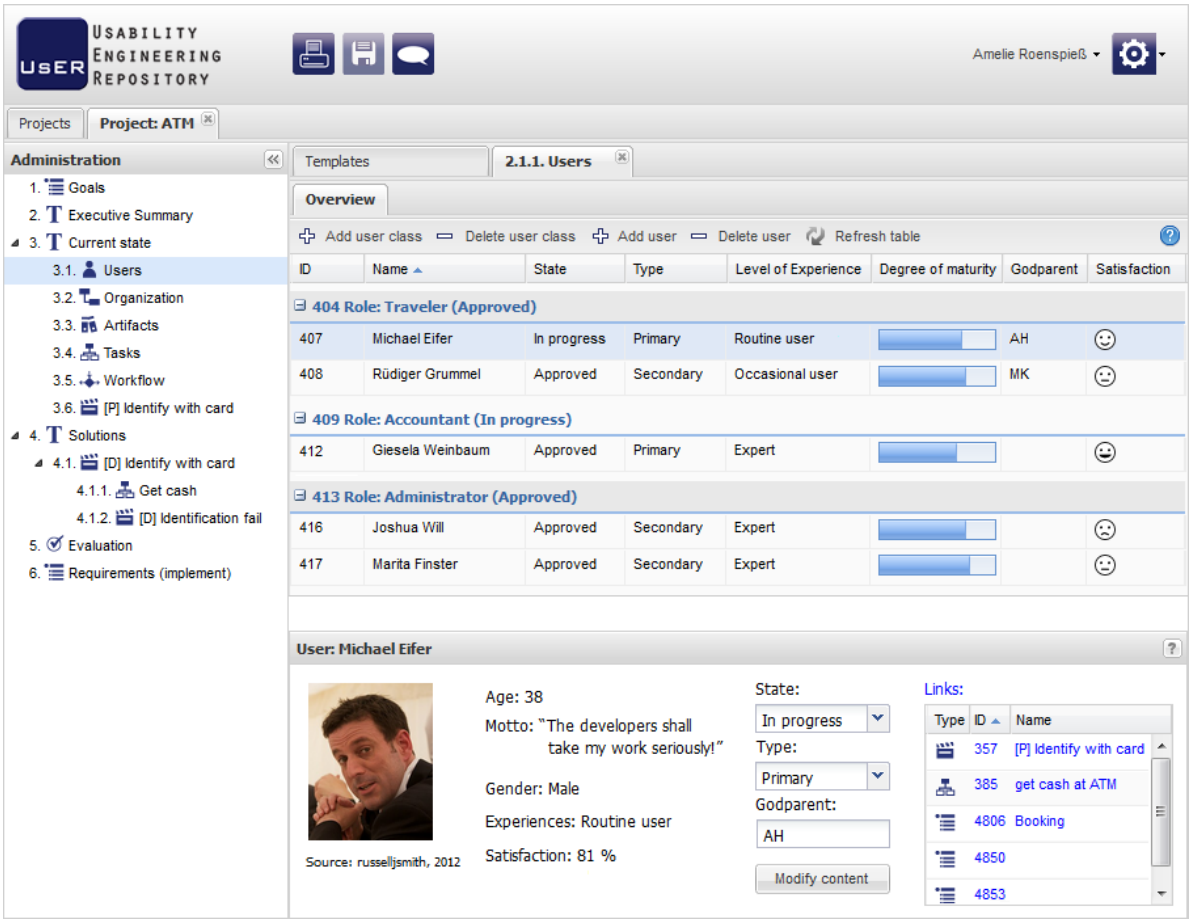

Figure 5: Overview: user classes, users, and details

Users

Users in UsER are personas in that they are *"detailed descriptions of imaginary people constructed of well-understood, highly specific data about real people."* according to the definition of Pruitt and Adlin (2006). The automatic integration of real user data is an aspect that will be further extended.

Apart from their user class and their name there is some other basic information about the users available at first sight in the list overview:

- *ID:* unique ID within UsER
- *status:* e.g. user description finalized or still under construction
- *type:* e.g. primary, secondary or negative persona
- *level of experience:* e.g. novice, routine user or expert
- *degree of maturity:* Visualized as a progress bar, this value indicates how much detail information has already been provided for the user.
- *godparent:* the member of the development team who feels responsible for the user and supervises his or her development as well as the fulfillment of the goals and needs
- *satisfaction:* A smiley icon indicates how happy the user is expected to be with the current state of the system. The more goals fulfilled, the happier the smiley – and vice versa: If a primary persona has an unhappy smiley, everyone knows that their goals have to be taken into account more seriously. A detailed insight into the calculation of the "happiness" will be given later; the resulting percentage of satisfaction is mapped onto the smileys that are shown in the user analysis overview according to the scheme in Figure 6.

Figure 6: Mapping degrees of user satisfaction onto smiley happiness

When a user is selected in the list, more details (especially a picture) are shown in the detail panel (lower area in Figure 5), some of which can be edited directly. The other content can be modified in the detailed user description which is opened in an own tab (see Figure 7). It includes some basic information already shown in the overview like name, age, experience, user class, and a short description. A picture can (and should) be added to make the persona more tangible: *"Photographs make them feel more real as you create the narrative and engage others on the team when you are finished."* (Cooper, Reimann & Cronin, 2007).

Personal and work information as well as special skills and abilities can be provided considering different aspects. Further input fields e.g. for life goals, a typical workday or the user's relationship to the product under development are available via checkbox. The degree of maturity is calculated considering how many of the input fields have been filled in so far and might be a motivation to fill in the missing ones. Other elements that are somehow associated with this user can be connected via links, e.g. goals are specified as requirements that have been identified as relevant for this user. All of them are listed in the requirements module of UsER so they can be included in the development process like any other requirement. These are used to calculate the user satisfaction visualized by the smileys. Simply speaking, if a user goal is fulfilled, the user satisfaction rises.

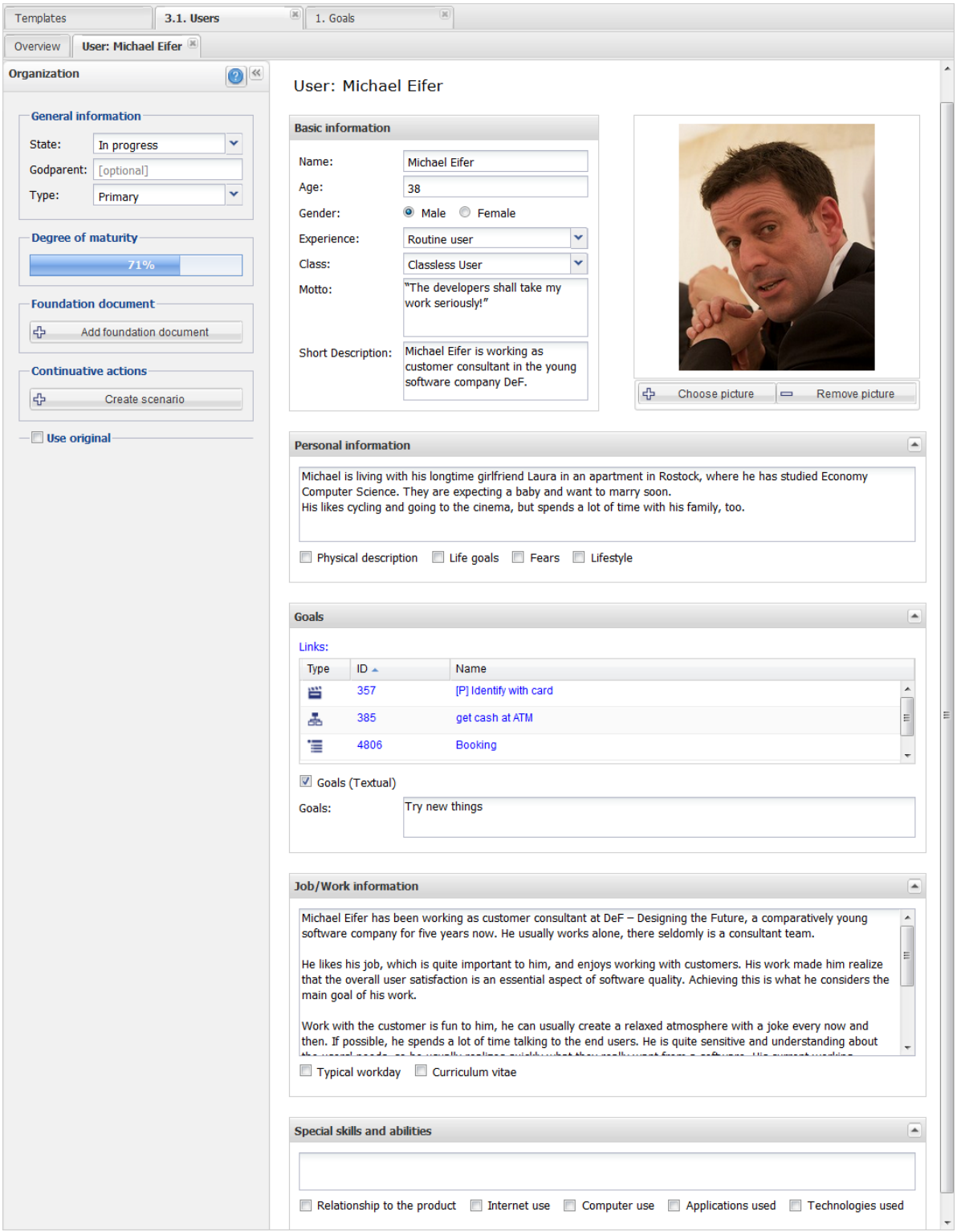

Figure 7: Detailed user description

Precisely, the level of satisfaction is calculated as a value between *0* and *1* in the following way (Hüttig, 2012):

$$satisfaction = \frac{1}{n} * \sum_{i=1}^{n} cat_i$$

$n = number\ of\ categories$ $cat_i = value\ of\ category\ i$

Currently there are just two categories of linked elements included in the calculation – requirements and scenarios – but the formula can handle arbitrary numbers of them. Categories are only considered if they are not empty.

$$satisfaction = \frac{(cat_1 + cat_2)}{2}$$

Category 1: associated requirements – cat_1

$$cat_1 = \frac{1}{r} * \sum_{i=1}^{r} stat_r$$

$r = number\ of\ requirements$ $stat_i = weight\ of\ requirement\ i$

The weight depends on the status of the requirement, it lies between *0* (requirement deferred or rejected) and *1* (requirement realized or closed).

Category 2: associated scenarios – cat_2

The scenarios are weighted differently for different types of personas.

- primary personas:

$$cat_2 = \begin{cases} 1 & 3\ or\ more\ scenarios \\ 0.6 & 2\ scenarios \\ 0.3 & 1\ scenario \\ 0 & no\ scenario \end{cases}$$

- secondary personas:

$$cat_2 = \begin{cases} 1 & 2\ or\ more\ scenarios \\ 0.5 & 1\ scenario \\ 0 & no\ scenario \end{cases}$$

- all other personas:

$$cat_2 = \begin{cases} 1 & 1\ or\ more\ scenarios \\ 0 & no\ scenario \end{cases}$$

The calculation of the user satisfaction is mainly experimental and will be validated and adapted continuously with new experiences and insights from projects.

User classes

User class properties for *roles* include name, a short description, qualifications, expectations, conditions, and tasks. *Target groups* also have a name and a short description and can further be defined by gender, nationality, denomination, family status, social status, generation, income, education, knowledge, and other aspects. Every user derived from a user class automatically inherits their properties, while they still can be edited and refined.

Additionally, user classes can be provided with behavioral variables in order to represent the characteristics of real users in an abstracted and aggregated manner (Cooper, Reimann & Cronin, 2007). This could be, for example, information about their preferred style of interaction or their usage of smartphones and the internet (see Figure 8). The users' answers can then be arranged and clustered on a bidirectional scale in order to identify trends, preferences, and tendencies based on real user data. The function for automating this process is currently under development: Information given by users in the evaluation module of user could automatically be transferred to the user module, for every person answering the corresponding questionnaire another icon would appear on the scale.

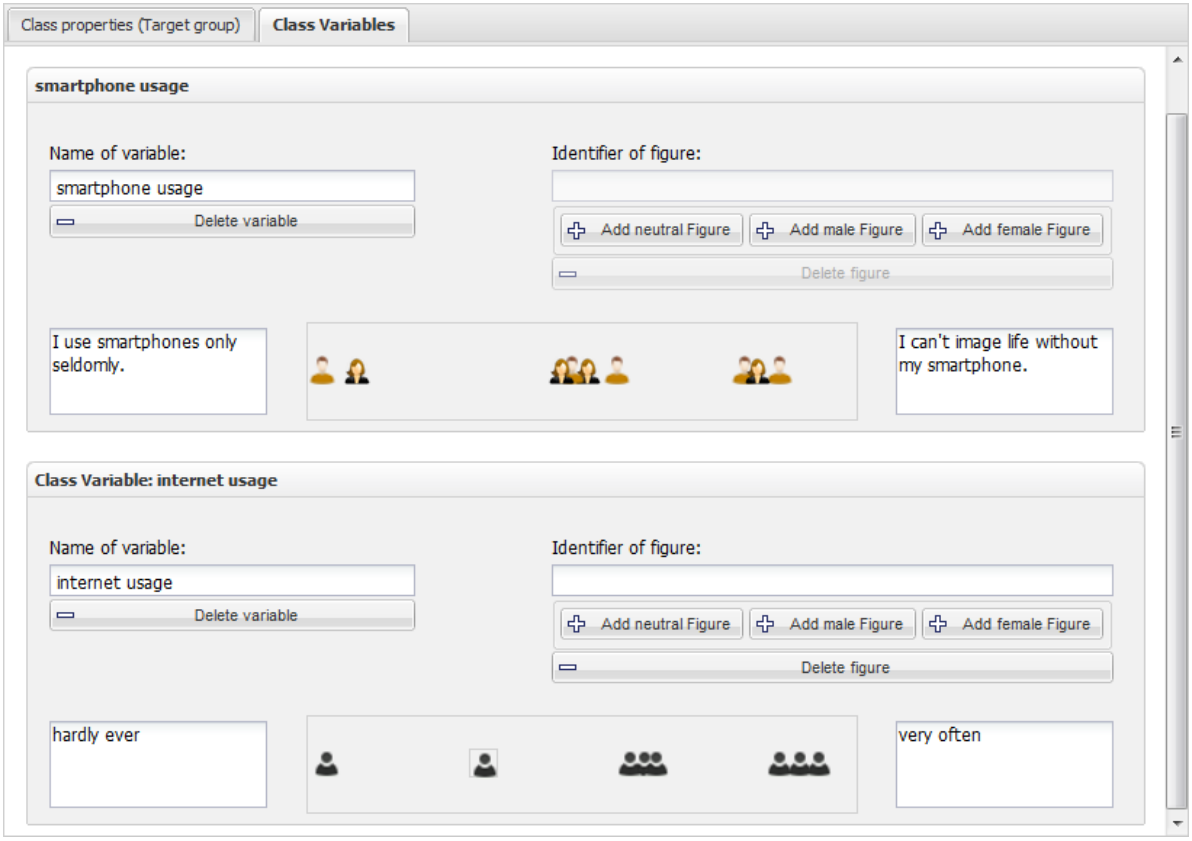

Figure 8: Behavioral variables for a user class

CONCLUSIONS AND FURTHER DEVELOPMENT

The Usability Engineering Repository UsER provides a variety of methods for integrating user and user class descriptions into the software engineering process. These can be used and combined flexibly in the context of all other information gathered throughout the development. Tasks or requirements derived from user goals can be retraced to them at every point of time in the process, allowing a deeper understanding of the users' needs and saving time and effort in the long run, because the origin and reason of requirements remains comprehensible.

Along with other modules, the UsER module for analyzing and modeling users and user classes will be developed further. Especially the calculation of the user satisfaction has potential for further refinement, as well as improving the automatic integration of behavioral variables by directly importing users' answers from questionnaires filled out during evaluations.

In order to develop software for safety- and time-critical contexts, a module for performing human failure mode and effect analysis (H-FMEA, cf. Frieling, Schäfer & Fölsch, 2006) will be implemented. Its entities can be linked to the described solutions for user modeling and analysis as well as descriptions of hazardous scenarios.

A mobile version of UsER will ease user analysis by offering functions for recording information in the field with tablet computers and integrating user feedback seamlessly, e.g. parts of interviews or pictures of artifacts in the working environment.

REFERENCES

Annett, J., Duncan, K. (1967), *"Task Analysis and Training Design*". University of Hull: Department of Psychology.

Beyer, H., Holtzblatt, K. (1998), *"Contextual Design: Defining Customer-Centered Systems"*. San Francisco: Morgan Kaufmann.

Constantine, L. (2006), *"Users, roles, and personas"*, in: The persona lifecycle: keeping people in mind throughout product design, Pruitt, J., Adlin, T. (Eds.), p. 499 – 519. Amsterdam: Morgan Kaufmann/Elsevier.

Cooper, A. (1999), *"The Inmates are Running the Asylum: Why High-Tech Products Drive us Crazy and How to Restore the Sanity"*. Indianapolis: SAMS/Macmillan.

Cooper, A., Reimann, R., Cronin, D. (2007), *"About Face 3: The Essentials of Interaction Design"*, Indianapolis: Wiley Publishing Inc.

Frieling, E., Schäfer, E., Fölsch, T. (2006), *"Human Failure Mode and Effects Analysis (H-FMEA). Operation Error Recognition and Avoidance"*, in: International Encyclopedia of Ergonomics and Human Factors, Second Edition – 3 Volume Set, CRC Press.

Grudin, J. (2006), *"Why Personas work: the psychological evidence*", in: The persona lifecycle: keeping people in mind throughout product design, J. S. Pruitt, & A. T. (Eds.), pp. 643-663. Amsterdam: Morgan Kaufmann/Elsevier.

Hackos, J. T., Redish, J. C. (1998), *"User and Task Analysis Techniques: User and Task Analysis for Interface Design"*. New York: John Wiley and Sons.

Herczeg M. (2001), *"A Task Analysis and Design Framework for Management Systems and Decision Support Systems."* ACIS International Journal of Computer & Information Science, Vol. 2, 3, pp. 127-138.

Herczeg, M. (2009), *"Software-Ergonomie: Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme"*, München: Oldenbourg Wissenschaftsverlag.

Herczeg M., Kammler M., Mentler T., Roenspieß A. (2013), *"The Usability Engineering Repository UsER for the Development of Task- and Event-based Human-Machine-Interfaces"*, in: 12th IFAC, IFIP, IFORS, IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems, Narayanan, S. (Ed.), pp.483-490. Las Vegas: International Federation of Automatic Control.

Holtzblatt, K., Burns Wendell, J., Wood, S. (2005), *"Rapid Contextual Design: A How-To Guide to Key Techniques for User-Centered Design"*. San Francisco: Morgan Kaufmann/Elsevier.

Hüttig, A. (2012), *"Development of a module for user analysis within the Usability-Engineering Repository UsER"*. Unpublished Bachelor thesis at the University of Lübeck.

Kammler M., Roenspieß A., Herczeg M. (2012), *"UsER: Ein modulares Usability-Engineering-Repository"*, in: Mensch und Computer 2012: interaktiv informiert und allgegenwärtig allumfassend!?, Reiterer, H., Deussen, O. (Eds.), pp. 333-336. Oldenbourg Verlag.

Kirwan, B., Ainsworth, L. K. (1992), *"A Guide to task analysis*". London, Washington, DC: Taylor & Francis.

Mayhew, D. J. (1999), *"The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design"*. San Francisco, California: Morgan Kaufman Publishers.

Mikkelson, N., Lee, W. O. (2000), *"Incorporating User Archetypes into Scenario-based Design"*. 9th Annual Usability Professionals' Association (UPA) Conference. Asheville, North Carolina.

Mulder, S., Yaar, Z. (2007), *"The User is Always Right. A practical guide to creating and using personas for the web"*, Berkeley: Riders.

Nielsen, J. (1993), *"Usability Engineering"*. Cambridge, Mass: AP Professional.

Norman, D. A., Draper, S. W. (1986), *"User centered system design: New perspectives on human-computer interaction"*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Paul, M., Roenspieß, A., Herczeg, M. (2013), *"UsER - Ein prozessorientiertes Entwicklungssystem für Usability-Engineering"*, in: Mensch & Computer 2013, Boll, S., Maaß, S., Malaka, R. (Eds.), pp. 181-190. München: Oldenbourg..

Pruitt, J., Grudin, J. (2003), *"Personas: practice and theory"*, in: Proceedings of the 2003 conference on Designing for user experiences (DUX '03, S. 1–15). New York, NY, USA: ACM.

Pruitt, J., Adlin, T. (Eds.) (2006), *"The persona lifecycle: Keeping people in mind throughout product design"*. The Morgan Kaufmann series in interactive technologies. San Francisco: Morgan Kaufmann Publishers.

Roenspieß, A. (2011), *"UsER: Konzeption und Design einer Kollaborationsplattform für betriebliches Usability Engineering*". Unpublished Master thesis at the University of Lübeck.

Rosson, M. B., Carroll, J. (2002), *"Usability Engineering: Scenario-based Development of Human-Computer Interaction"*. San Francisco: Morgan-Kaufman.