



Integrating Personas and Use Case Models

Anke Dittmar^(✉) and Peter Forbrig

University of Rostock, Rostock, Germany
{[anke.dittmar](mailto:anke.dittmar@uni-rostock.de),[peter.forbrig](mailto:peter.forbrig@uni-rostock.de)}@uni-rostock.de

Abstract. Multidisciplinary design is characterized by phases of distributed work and co-design activities. An effective sharing and integration of design representations that are created by sub-teams from different disciplines is still often challenging and typically requires the reconciliation of diverging design perspectives. This paper investigates an integrated use of personas and use cases - two popular types of design representations among interaction designers and software engineers respectively. The proposed integration is particularly suitable for role-based interactive systems and differs from existing integration approaches in that it is based on a critical examination of the prevalent understandings of the goal concept in persona and use case approaches. In the paper we suggest distinguishing between organizational and user goals (while at the same time acknowledging their interplay). Corresponding adaptations to use case notations and personas are introduced and discussed. These remove the tight coupling between goals and tasks and allow integration of organizational and different persona-specific design perspectives within one use case specification and at the interaction level. As a result, interactive systems can be specified by a more compact sets of use cases. This is illustrated by an example in the context of course management systems in higher education.

Keywords: User-centred design · Personas · Use cases · Design representations

1 Introduction

External design representations such as scenarios, prototypes, and formal models are ubiquitous in interaction design [17]. According to Visser [42], the interactive system under design emerges from the creation, transformation and evaluation of design representations about the what, how and why of this system. This paper will particularly look at personas and use cases: two types of design representations which have been developed (together with corresponding methods) in relative isolation from each other in the context of human-computer interaction and software engineering respectively. Personas mainly contribute to the description of the why and use cases more to the description of the what of

an interactive system. Multidisciplinary design is established as a basic principle of user-centred design [24,25]. It is generally assumed that multidisciplinary design teams are more likely capture the multiple viewpoints which need to be considered to achieve product quality. For instance, Mackay [32] points out that science, design and engineering disciplines offer valuable skills and perspectives, but each discipline also “has the potential to miss important aspects of the design problem”. Multidisciplinary teams need to acknowledge both distributed work by specialized sub-teams and discussions and sharing in heterogeneous teams [2]. Design representations created and used by specialists are shaped by the notations, techniques and methods they are used to in their professional practices to accomplish specific design tasks. External representations in heterogeneous teams must facilitate a flexible interpretation by collaborators from different disciplines. Team members couple and transform initially provided design representations in order to reveal and employ differences in existing design perspectives. However, even if an integration of these perspectives and a shared design understanding is achieved (and this may require the rethinking of assumptions and concepts of the collaborators’ respective design approaches), it only becomes effective if it is documented in ways that fit again into subsequent specialized design activities.

This paper investigates how interaction designers, requirements engineers, and other stakeholders can be supported in sharing and integrating their viewpoints by using personas and use cases. Personas are user representations which are widely employed in user-centred design approaches to establish the interaction designers’ empathy and understanding of the users of the system under consideration [11,22,37]. Use cases capture high-level functional requirements of systems from a usage perspective. They were adopted as a part of the Unified Modeling Language (UML)¹, the de-facto standard language for object-oriented software engineering. Use case diagrams are among the most widely used parts of the UML [27]. Both personas and use cases are ‘simple’ representations in the way that they can be easily understood and modified by stakeholders with different backgrounds. They thus support collaboration in heterogeneous teams. They also both embody the perspective of users, but do so with different understandings and objectives which therefore requires reconciliation. Use cases focus on functionality and are based on rather abstract models of users. Personas bring more emphasis to the diversity of users and contexts of use but are less detailed in their description of the interaction between user and system.

Use cases and personas are partly based on similar concepts and vocabulary, which is deceptive as there is not always the same underlying understanding. Existing integration approaches of personas and use cases such as [8,39] do not pay sufficient attention to the different uses of the goal concept. They map one or more personas to an actor of a use case at the goal level, but do not integrate the personas’ perspectives at the use case specification level (the interaction level). As a consequence, the set of use cases describing the system under consideration becomes difficult to manage and easily leads to incoherent designs.

¹ <https://www.omg.org/>.

This paper suggests an approach to reconcile the different understandings of goals. Based on a review of personas, use cases, and existing integration approaches (Sect. 2), more precise understandings of the concepts of user, goal and task are developed by distinguishing between organizational and user goals (Sect. 3.1). Basically, organizational goals and related tasks are understood in the context of work systems and are assigned to roles (described by actors in use cases). The concept of user refers to individuals acting in those roles. It is assumed that they generally share the goals established by the work system they are involved in but that they also have their specific backgrounds and personal goals. A single use case model has to specify both the similarities and the differences of the users' diverse ways to achieve the (organizational) goal of that use case.

A main contribution of the paper is the introduction and discussion of different adaptations to use case notations and personas to support an integration according to the above ideas (Sect. 3.3). Adaptations include persona-specific goals and actions in use cases, the <<automate>>-stereotype in use case diagrams, and task-related user goals for personas. They preserve the essential character of persona and use case descriptions, but also facilitate the enhancement of use cases by ideas derived from personas, and thus, the described functionality not only covers the perspective of the considered work system but also the perspectives of diverse users. The suggested approach is particularly suitable for analyzing and designing role-based systems. The proposed notations are applied to an illustrative example which is introduced in Sect. 3.2. The paper closes with a discussion, some conclusions and future work (Sect. 4).

2 Background and Related Work

This section gives a short background to use cases and personas. In the context of this paper, readers are assumed to be less familiar with use cases. Therefore, Sect. 2.1 provides an introduction to these. Subsection 2.2 provides an overview about persona approaches. We then review existing approaches for integrating personas into requirements engineering approaches.

2.1 Use Cases in Software Engineering

Use cases were introduced into object-oriented software development by Jacobsen et al. [26] in the early 1990ies, but it is not an inherently object-oriented modeling technique. The approach aims at supporting 'user-centric solutions' [30]. Use cases describe systems as they appear to outside users. They "represent the things of value that the system performs for its actors" [4], and according to Kulak and Guiney, "[a]ll requirements that drive the development of use cases come from actual business needs of the users" [30]. In the use case approach, *users* are represented by *actors* but the terms should not be used interchangeably. Cockburn [10] defines an actor as a role outside the system that can be played by people, organizations, or technical systems and further distinguishes

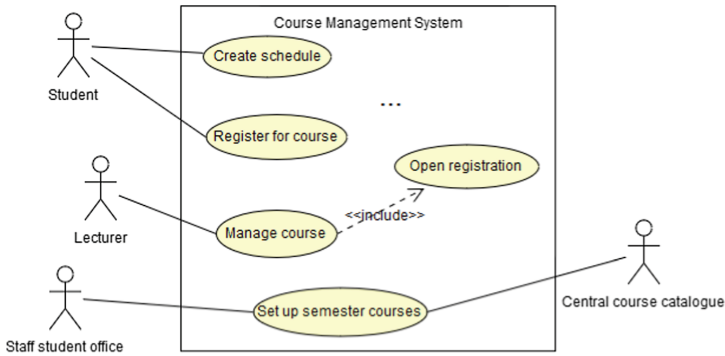


Fig. 1. Part of a use case diagram for a course management system with three primary actors and one secondary actor.

between primary actors and secondary actors. While the former interact with the system to achieve certain goals the latter are used by the system to provide services to primary actors [10]. Hence, Kulak and Guiney refer to primary actors in their definition of a use case as “a collection of possible sequences of interactions between the system under discussion and its Users (Actors), relating to a particular goal” [30].

Generally, a use case model consists of a use case diagram and a set of use case specifications. Use case diagrams are part of the UML² and can be understood as contextual descriptions. Systems and their environment are depicted by the names of use cases (indicating the actors’ goals), the actors, and their relationships [31]. The diagram in Fig. 1 shows, for example, that students want to use the system under consideration to achieve two goals (create schedule and register for course). The central course catalogue is a secondary actor that is used by the system to support staff members in setting up semester courses. The real work, though, is in the creation of the use case specifications with details about the interaction between primary actors and the system. There is no standardized format but the most common are semi-formal textual specifications. Cockburn’s [10] template (or variations) is widely used and it is also the basis for the adaptations to use cases which are suggested in this paper. The template recommends including the following points in a use case description.

- Name of the use case and context of use,
- Scope (enterprise/system/subsystem scope) and corresponding level of description (complex/task/function level),
- Primary actor and (other) stakeholders with their interests,
- Preconditions and trigger to run the use case,
- Success end condition and minimal guarantees,
- Main success scenario (basic course),

² The graphical notation for use case diagrams is defined in the UML specification (see <https://www.omg.org/spec/UML/>).

<p>Use Case: <i>Create schedule</i></p> <p>Description level: Task level</p> <p>Primary actor: Student</p> <p>Stakeholders: Lecturer - students should attend classes,...</p> <p>Precondition: Student is logged in</p> <p>Success end condition: Recommended schedule without conflicts</p> <p>...</p> <p>Main success scenario:</p> <ol style="list-style-type: none"> 1. System displays list of courses of study 2. Student selects course of study and semester 3. System displays all obligatory courses and available optional courses 4. UC: <u>Select optional courses</u> 5. System generates schedule <p>...</p> <p>Extensions:</p> <ol style="list-style-type: none"> 5a. Time conflict between two courses: <ol style="list-style-type: none"> 5a.1. System displays conflicting courses 5a.2 Student removes one course Rejoin at 4 <p>...</p>

Fig. 2. Part of the use case description *Create schedule* according to the template in [10].

- Extensions (alternative courses and their conditions),
- Variations of interactions,
- Other relevant information.

As mentioned previously, a use case specifies a set of sequences of actions or scenarios. The set contains at least the main success scenario which is the basic course of actions starting with the trigger and stopping with the success end condition and goal achievement. Cockburn [10] points out that steps or actions in the scenarios can be of three types: (1) an interaction between actor and system, (2) a validation by the system, and (3) an internal state change of the system. Scenarios should not only describe how primary actors achieve their goals but they should also make visible additional actions that are needed to consider other stakeholders' interests (e.g., validation steps). Figure 2 partly depicts the detailed description of one of the use cases in Fig. 1. The main success scenario is complemented by scenarios with additional actions (alternative courses) to deal with time conflicts (extension part). The example also illustrates that use cases can be included in or extend other use cases to obtain more succinct descriptions. For instance, sub-use case *Select optional courses* is 'folded' into a single action (step 4) in the main success scenario. The use case in Fig. 2 will be revisited later in this paper to illustrate the proposed integration approach.

Use cases can be used for both analyzing and designing systems but currently they are mainly employed for the latter purpose to capture functional requirements from the perspective of utility. Ideally, use cases should be used throughout the whole development process, but Jacobsen et al. [27] show at the example of use case slices for agile development that adaptations are needed to meet the requirements of specific development practices. It is generally not easy

to keep the set of use case specifications of a system small and to create and maintain manageable use case models of high quality. Cockburn [10] states in this context: “I often refer to the set of use cases as an ever-unfolding story. It is our job to write this story in such a way that the reader can move around it comfortably”. Recommendation and methodological guidance are given, e.g., in [10,30].

2.2 Personas in Interaction Design

The persona concept was introduced into interaction design by Cooper [11] in the late 1990ies. “A persona is an archetype of a user that is given a name and a face, and it is carefully described in terms of needs, goals and tasks” [5]. It represents a whole user group, but according to authors such as Cooper [11] and Grudin [22], interaction designers engage better with personas than with abstract information about user groups. Personas work as design representations due to the human ability to make predictions about a person’s behavior based on a precise description of their backgrounds and goals [23]. Typically, a small set of personas is identified for a system under consideration³. Designers then use the set as a communication tool to identify the functionality of the system, to support design decisions, to develop plausible usage scenarios or to evaluate whether a user interface covers the users’ needs. The persona set helps them to discuss the system from the perspective of different user groups, and at the same time they are prevented from designing systems “that supposedly fit everyone but in the end fit no one” [18].

However, personas are not a panacea to ensure a user-centred design perspective and Floyd et al. [20] argue that there is no single persona approach which is applicable in every context. The authors identify different persona types (e.g., initial and final personas [11], ad-hoc personas [38]) and discuss their benefits and limitations. In [16], six dimensions concerning the user representation and the practical implementation of the persona method are identified along which persona usage can vary. For example, there are differences in the identification of user groups and the empirical basis upon which personas are constructed [9]. Another dimension concerns the presentation of a persona that can range from simple bullet-point lists to narrative descriptions (enriched with visual material). A persona becomes useless if it describes almost everyone (‘elastic persona’ [29]) or if it excludes users from the represented user group by giving too many details of the fictive person [9]. Nielsen [35] distinguishes between ‘flat’ characters as used, e.g., in goal-directed design [11], and ‘rounded’ descriptions that include “both personal (inner) and inter-personal (social, public, and professional) elements” [36]. She points out that it is a rounded depiction that evokes the designers’ empathy.

There are other aspects that influence the effectiveness of personas. Although personas are considered to be ‘simple’ design representations, designers must

³ There can be a further distinction between primary, secondary and anti-personas to express priorities between the needs of different user groups [12].

be skilled to identify with personas that are different from themselves [37] or to recognize bad quality descriptions such as elastic personas or ‘my mother’ personas [29]. Blomquist and Arvola [5] suggest that personas are more likely be used when they are created in the design team itself. Persona approaches must be related to other design approaches but an appropriate combination is mostly discussed within the user-centred design community and less across disciplinary boundaries with software engineers (see next subsection). For instance, Grudin states that “[p]ersonas come first and drive the construction of scenarios around them” [22] and all participants in the empirical study in [37] understood personas and scenarios as a combined method. Bødker et al. [6] point out that personas should not replace the active involvement of users and other stakeholders and studies such as [43] illustrate how personas can support participatory design processes.

2.3 Existing Integration Approaches

Various authors recognize the potential of persona methods to particularly enrich requirements engineering practices. Acuña et al. [1] state that “[p]ersonas provide an understanding of the user, often overlooked in SE [software engineering] developments”. Schneidewind et al. [40] characterize challenges in requirements engineering as follows: (1) requirements of the users are often neglected, (2) insufficient communication about future users, and (3) users, tasks and context of use are not thoroughly connected to the requirements of a system. The authors suggest employing personas to better prioritize and illustrate use cases, and to identify and specify nonfunctional requirements. However, their approach is not further elaborated. Similarly, Francescomarino et al. [21] and Faily [19] call for integrating of concepts and analysis techniques of different approaches such as user-centred design and goal-oriented requirements engineering to better represent user needs in “user-intensive” systems [21] but do not discuss in detail the indicated relationships between personas, scenarios and use cases.

Acuña and colleagues [1, 8] assume that persona methods can only be successfully built into the requirements stage of regular software engineering developments if they come with a detailed definition of activities and products. The authors developed PersonaSE, a persona method that consists of eleven activities which are mapped to four common requirements engineering activities (elicitation, analysis, specification, and validation of requirements). Each activity is defined by a name, objectives, techniques and expected outcomes or products. In the context of this paper, activity 10 (“build use cases”) is of particular interest. Here, Acuña et al. [1] suggest to map a set of primary and secondary personas to each use case and to create separate use case descriptions for each persona (see Fig. 3(b)). However, this would lead to a large number of use case specifications for a system. Inconsistencies between single specifications are more likely. In our approach, we are less interested in normative process models for integrating persona and use case methods. We rather focus on adapting existing use case notations to enrich single specifications but keep the overall set of use case specifications minimal and manageable.

The case study in [39] investigates an integrated use of personas and use cases in the context of a small information system for tracking training of employees in a community hospital. Randolph [39] points out that personas better aid the designing of user interaction that meets a user's needs and goals, and use cases give more details about the specifics of users' task requirements. In the case study, personas are mapped in a one-to-one way to actors in use case models suggesting that a persona is a representation of role (see Fig. 3(a)). A more differentiated view is proposed in [34]. Miller and Williams argue that use cases only support a role-based requirements engineering and that the integration of the persona perspective allows "to examine the different types of people who could play a role" [34]. Similarly to [1], the authors suggest to develop separate use case descriptions for each persona. A slightly different approach is taken in [41]. Sim and Brouse also start with the identification of roles and corresponding use cases. Personas are used through a process of role refinement to construct viewpoints which form the basis for later conceptual modeling. One or more viewpoints can be defined for each persona. A viewpoint is described in terms of goals, concerns, scenarios, tasks, functional and non-functional requirements. However, the relationship between these concepts is not clarified.

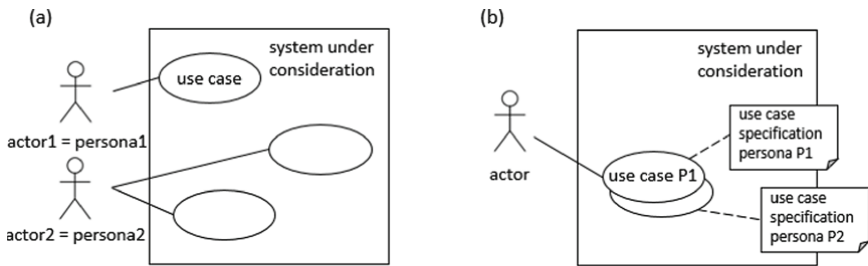


Fig. 3. (a) 1:1-mapping between primary actors and personas in [39], (b) separate use case specifications for personas in the role of the primary actor in [1, 34].

3 The Integration Approach

In the previous section, we have seen that personas as well as use cases aim at understanding an interactive system from the perspective of its users. Existing integration approaches stress that use cases reduce the description of users to abstract roles and refer to the potential of personas to provide a richer picture. However, they stop at the level of use case diagrams and separate specifications are developed for each persona assigned to a use case. The underlying assumption, although not stated explicitly, seems to be that there are no similarities at all in the personas' ways to achieve the goal of the use case. As a consequence, the number of use case specifications for a system 'explodes' and use case models become less useful for subsequent design activities. In our approach,

we aim at providing means to produce smaller, more manageable sets of specifications leading to more coherent designs. This requires a better clarification of concepts shared by persona and use case approaches. Based on the revision of goals and tasks in the next subsection, adaptations of the use case notations introduced above are developed which allow the unification of perspectives of different user groups within single use case specifications. The suggested adaptations are applicable to role-based interactive systems as will be illustrated by an example introduced in Subsect. 3.2.

3.1 Conceptual Understandings

Personas were originally intended to support a goal-directed design [11]. In this context, goals are mainly considered to be individuals' goals ('user goals') with some of them originating from a group an individual is involved in (e.g., a corporation, an institution). In [12], three types of user goals are distinguished: experience goals ("how someone wants to feel while using a product"), end goals ("motivation for performing the tasks associated with using a specific product"), and life goals ("long-term desires, motivations, and self-image attributes, which cause a persona to connect with the product"). User goals are complemented in [12] by customer goals, corporate goals, and technical goals. However, Cooper and Reimann [12] emphasize the role of end goals in driving the design of the interactive artifact. Cooper's personas were later criticized for their strong focus on goals easily leading to the description of 'flat' characters [35] which fail to evoke the designer's empathy. Still, goals in most persona approaches are mainly described as goals 'owned' by individuals. In contrast, goals in use cases have rather to be understood in an organizational context. Actors specify roles with "certain operational responsibilities imposed by the business processes and business rules of the business domain" [33]. Hence, their goals are 'organizational' goals that are established in an organization in order to consider (at least to a certain degree) the interests of involved stakeholders.

Goals in use case approaches and personas' end goals are seen as end conditions which can be achieved by accomplishing tasks. Use case specifications at the task level are restricted to goals that actors achieve through interactions with the system under consideration and show some similarities to task models commonly used in human-computer interaction [13]. In particular, nested goal structures are assumed which are tightly related to corresponding task hierarchies. This is graphically depicted in Fig. 4(a) where some of the actions of use case UC are 'folded' use cases. The goal of UC can be accomplished either by performing the actions or steps of the basic course (forming the main success scenario) or by 'departures' to those alternative courses which still guarantee the success end condition. On the one hand, these sequences of actions specify a task structure. On the other hand, actions can be use cases themselves and thus represent goals. Goal unfolding results in interaction or task refinement [33]. Figure 4 shows four sub-use cases or sub-goals, two of them are included in the basic course of UC (UC1, UC2) and two are steps in alternative courses (UC3, UC4) which is also indicated by the include- and extend-relationships in the corresponding use case

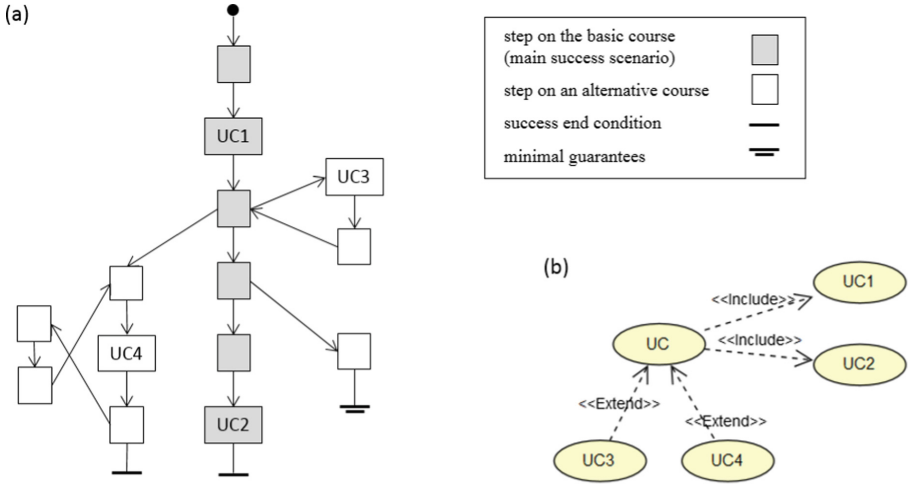


Fig. 4. A schematic use case UC: (a) Visualization of the specified action sequences (adapted from [33]). Sub-use cases UC1..UC4 are ‘folded’ to actions. (b) The corresponding part of the use case diagram depicts the nested goal structure more explicitly.

diagram. In the context of task analysis for human-computer interaction, Diaper and Stanton [13] criticize the broad, often vague use of the term goal and the extreme linking of goals and tasks described above. The authors point out that people rarely do anything for just one reason and that their multiple goals interact in complex ways [13, 15]. Diaper [14] also mentions problems of generalization and notes that “...almost all task analysis methods claim to be able to combine descriptions of a task performed by different people in different ways. Quite a few methods are able to combine different tasks into a single task representation”. This critique equally applies to use case specifications where even a main success scenario is assumed suggesting that there is ‘one best way’ to achieve a goal.

In our approach, we restrict ourselves to role-based interactive systems; that is, to software systems which are applied in work systems where roles guide the distribution of work. We distinguish between *organizational goals* as intended states or conditions of the work system and individual or *user goals* which express intentions and preferences of the people who are part of the work system and actually use the interactive system under consideration to fulfill their responsibilities (referred to simply as the users). We assume an interaction between organizational goals and individual goals. Users internalize or appropriate organizational goals by relating them to their own values, attitudes, and goals. Such appropriation processes are unique to every person, and at the same time shaped by procedures (task structures) that exist in the work system. Users externalize their individual goals in their particular ways to perform tasks in the work system (and this can, in turn, shape organizational goals). Based on these assumptions, we suggest complementing goal descriptions of use cases by task-related goals

of personas. Before we consider the implications of such different goal sets in more detail and develop corresponding adaptations to use case specifications, we briefly introduce the example that will be used to illustrate the proposed adaptations.

3.2 Introduction to the Illustrative Example

The example considers a use case of a course management system called *StudOrg*. Course management systems have become standard components in higher education and their features can be leveraged for a variety of academic purposes (e.g., management and retrieval of course material) [28]. Some typical roles of such systems are already shown in Fig. 1. We focus here on use case *Create schedule* with role *Student* as primary actor. Two personas are developed to give a richer and more differentiated picture of students (following recommendations in [35,36]). Figures 5 and 6 show those parts of the narrative descriptions of *Adrian* and *Stephanie* which are relevant for the use case, together with photos depicting them in action. The next subsection presents two alternative adaptations to use case specifications. The general ideas are illustrated and explained in more detail by extended versions of *Create schedule* in Fig. 2. Both versions integrate the perspectives of Stephanie and Adrian within a single specification.

Adrian grew up in a small village near Rotown and completed a vocational diploma in crop farming. His parents wanted him to go to university but he decided to work in a local farming company. Although he liked the work as such he soon realized that he really has few development options without further qualification. Now, Adrian is in his third semester. His girlfriend Iris already studied in Rotown and so it was no big decision to study agricultural science there.

For Adrian, much of what they learn is too theoretical. He focuses on what he likes to do such as practical projects and he is happy about his voluntary work in the “Intercultural Garden”. In their last soil measurement project he got a really good grade and could even help other students with the analysis tool. But he let other things slide and he sometimes forgets to go to classes. In one case, he now has to repeat the whole course. He had a fight with Iris about it. She said that he should have a printout of his schedule in their kitchen and that it is easy to create with StudOrg...



<https://www.pexels.com>

Fig. 5. Fragment of persona *Adrian* (25 years old).

3.3 Adaptations to Use Case Models

Similarly to [1,34], we consider different types of users who could play a role in a work system (see Subsect. 2.3). But due to the interplay between organizational and user goals described above, we assume both differences and similarities in the ways different user groups achieve a goal. In our integration approach, use

Stephanie had applied to Bellcity University. She knew that they always have many applicants and was not too disappointed about the rejection. So, she started studying communication and media studies in her hometown, but definitely wants to get a Master degree at Bellcity University. Last summer, she used the time between high school and university to go to Bellcity for an internship in a PR agency.

Stephanie manages the Facebook page of her volleyball club. She likes to post pictures from her iPhone, a Christmas present of her grandmother who is Stephanie's most faithful 'follower'. They try to meet regularly though they both need their calendars to find a date. Stephanie thought the first semesters would be more difficult but she could manage the workload well. Now she would like to attend additional courses, especially the one on creative writing. She wonders whether she can arrange it with her regular semester schedule...




Fig. 6. Fragment of persona *Stephanie* (20 years old).

cases represent organizational goals, primary actors represent roles, and personas represent user groups acting in the different roles of an organization. The persona descriptions especially provide insights into the users' specific appropriation of organizational goals. They allow implications to be drawn about the way tasks are performed in order to satisfy both the organizational goals and the users' goals and preferences. Personas also enable general implications for the user interface design which can be summarized in a table similar to the one recommended in [39]. Table 1 shows persona-specific implications in the running example. The explicit consideration of organizational and user goals leads to modifications in the use case specifications. Goal descriptions and end conditions need to be extended to include persona-specific aspects; implications for the user interface can be inserted. This is indicated below with Cockburn's template [10].

Use Case: ⟨name of the use case⟩

Description level: task level

Primary actor: ⟨name of the role⟩

with personas *Persona 1*, *Persona 2*,...

Goal: ⟨description of organizational goal⟩

Persona 1: ⟨description of task-related user goals⟩

Persona 2: ...

Success end condition: ⟨the state of the world upon successful task completion⟩

Persona 1: ⟨persona-specific state⟩

Persona 2: ...

Implications for user interface:

Persona 1: ⟨persona-specific implications⟩

Persona 2: ...

Use cases describe now a set of *persona-specific goal sets* which have the overall organizational goal in common. The resulting question is: how should

Table 1. Persona-specific implications in the example.

	<i>Adrian</i>	<i>Stephanie</i>
Description	Infrequent user of StudOrg via laptop at home, familiar with Word, Excel, domain-specific analysis programs; digital technology has low priority	Frequent user of StudOrg; enjoys using technology and social media (iPhone, iPad and Mac)
Interface implications	StudOrg should be easy to use; show only most necessary features	Flexible interface with additional features
<i>Create schedule</i> - task-related user goals	Printout of the schedule with a clear layout; registered for all courses in the schedule	Efficient but flexible composition of the schedule; transfer to iCloud Calendar

action sequences (scenarios) be specified to make visible the differences and similarities in the personas’ task structures. Two alternative ideas are sketched in Fig. 7.

- *Adaptation 1:* The constraint of having only one basic course (the main success scenario) is relaxed. There are overlapping persona-specific basic courses as

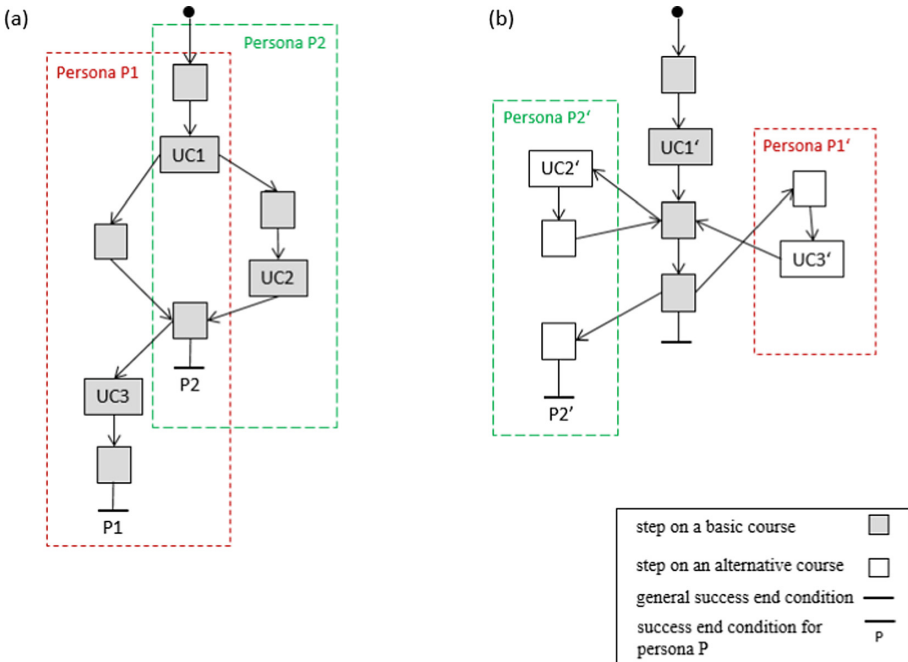


Fig. 7. (a) Adaptation 1: overlapping persona-specific main success scenarios, (b) Adaptation 2: one common main-success scenario with persona-specific extensions.

indicated for two personas in Fig. 7(a). The figure abstracts from alternative courses and variations which can additionally be defined for personas.

- *Adaptation 2*: As usual, one basic course is defined (possibly with persona-specific variations). Persona-specific interactions with the system are modeled exclusively by alternative courses and extension points. Figure 7(b) depicts this in a schematic way for two personas.

Figure 8 applies adaptation 1 to the use case *Create schedule* in Fig. 2 to better consider the specific perspectives of the two personas in the example. Two basic courses are specified: steps 1, 2, 3, 4, 5, A-6, A-7, A-8, 11, 12 form the main success scenario of *Adrian* and steps 1, 2, 3, 4, 5, S-6, S-7, S-8, S-9, S-10, 11, 12 form that of *Stephanie*. The achievement of the main goal - a schedule that is consistent with recommendations and without conflicts - is ensured by the shared parts in the scenarios (steps 1..5, 11, 12). It is an organizational goal that serves the interests of various stakeholders (e.g. lecturers, see Fig. 2). Additionally, *Adrian* is supported by an automatic course registration (steps A-6..A-8) and *Stephanie* can use an extended editing mode to add other than recommended courses to her schedule (steps S-6..S-10). The variations in step 12 handle the different preferences of *Adrian* and *Stephanie* concerning the form of the generated calendar. Extensions to the main success scenarios are only indicated in Fig. 8. The textual representation of the two basic courses in the example is similar to the graphical visualization in Fig. 7(a).

An example for adaptation 2 can be seen in Fig. 9. There is one main success scenario which guarantees the success end condition for the organizational goal. Persona-specific interactions are expressed by alternative courses in the extension part. According to [33], alternative courses are a guarded variation of a part of another course (in particular, of the basic course) and can describe optional parts of behavior, alternative interaction parts, business error recovery or fault handling. Figure 9 indicates two persona-specific alternative courses (one for *Adrian* and one for *Stephanie*, each starting at step 6 of the basic course). The corresponding guards 6a-A and 6b-S are persona-specific goals and preferences.

Annotated Use Case Diagrams and the <<automate>>-relationship: Figure 10 gives an overview of the similarities and differences in the personas' interaction with the considered system. The annotated use case diagram shows mappings between personas and use cases which result from the example specification in Fig. 8 (adaptation 1). Annotated <<include>>-relationships exist between use case *Create schedule* and those sub-use cases which are steps in the personas' basic courses. In contrast, the description of persona-specific behavior in adaptation 2 is exclusively based on <<extend>>-relationships (annotated by the personas' names).

<<Include>>- and <<extend>>-relationships are predefined stereotypes in the UML and describe required and optional subgoals respectively. We introduce another type - the <<automate>>-relationship - to describe different degrees of automation for different personas. In the example, *Adrian* prefers automatic course registration within the use case *Create Schedule* (see Figs. 8 and 9) while

Use Case: *Create schedule*

Description level: Task level

Primary actor: Student, with personas *Adrian* (A) and *Stephanie* (S)

Goal: Student wants to compose the schedule for the next semester.

Adrian: Wants to follow recommendations and automatic registration for all selected courses.

Stephanie: Composed schedule contains, in addition to recommended courses, other courses she is interested in.

Implications for user interface:

Adrian: Simple and consistent interface, only basic features visible.

Stephanie: Interface with additional features, can be customized.

Precondition: Student is logged in

Success end condition: Schedule which is consistent with recommendations and without conflicts.

Adrian: Printout of schedule, confirmation of course registrations.

Stephanie: Schedule is integrated in personal digital calendar.
...

Main success scenario:

1. System displays list of courses of study	
2. Student selects course of study and semester	
3. System displays all obligatory courses and available optional courses	
4. UC: <u>Select optional courses</u>	
5. System generates and displays schedule	
A-6. Adrian confirms and activates automatic registration	S-6. Stephanie activates the extended editing mode
A-7. System saves schedule and registers Adrian for each course in the schedule	S-7. System changes to extended editing mode
A-8. System sends confirmation message for registrations	S-8. UC: <u>Add additional course</u>
	S-9. UC: <u>Remove additional course</u>
	S-10. UC: <u>Select optional courses</u> (repeat steps S-8..S-10 in any order)
11. Student chooses format of schedule	
12. System provides schedule in chosen format	

Extensions:

5a. Time conflict between two courses: ...

A-7a. A course is fully booked: ...	S-8a. Time conflict: ...
-------------------------------------	--------------------------

...

Variations:

12. Schedule as printout (*Adrian*) or calendar export (*Stephanie*)

Fig. 8. Adaptation 1 applied to use case *Create schedule* in the example.

Use Case: *Create schedule*
Description level: Task level
Primary actor: Student, with personas *Adrian (A)* and *Stephanie (S)*
 ...
Main success scenario:
 1. System displays list of courses of study
 2. Student selects course of study and semester
 3. System displays all obligatory courses and available optional courses
 4. UC: *Select optional courses*
 5. System generates and displays schedule
 6. Student confirms schedule
 7. System saves schedule
 8. Student chooses format of schedule
 9. System provides schedule in chosen format
Extensions:
 5a. Time conflict between two courses: ...
 6a-A. Adrian additionally wants to be registered automatically:
 1. Adrian activates automatic registration
 2. ...
 6b-S. Stephanie wants to add other courses:
 1. Stephanie activates the extended editing mode
 2. ...
 ...

Fig. 9. Adaptation 2 applied to use case *Create schedule* in the example. (Descriptions of goals, end conditions etc. are the same as in Fig. 8.)

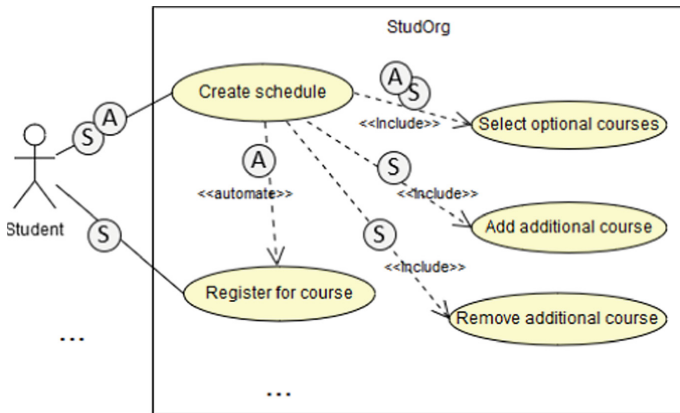


Fig. 10. Adaptation 1: use case diagram in the example, annotated by personas *Adrian (A)* and *Stephanie (S)* acting in role *Student*.

Stephanie interacts with the system to register for single courses. Hence, the diagram in Fig. 10 depicts *Stephanie* (but not *Adrian*) as primary actor of use case *Register for course* and there is a corresponding <(automate)>-relationship annotated by *Adrian* (A).

4 Discussion

“Interaction design requires input from science, engineering and design disciplines” [32]. Authors such as Mackay discard the idea that interaction designers should develop expertise in all of the component disciplines but emphasize that multidisciplinary teamwork requires from participants an increased understanding and appreciation for other disciplines. Design activities have to be created “in which all members of the design team, including users, can participate equally” [32]. The suggested coupling of personas and use cases, in the context of role-based interactive systems, supports a shared discussion and refinement of design ideas across sub-disciplines.

Bellotti et al. [3] point out that for an effective collaboration in multidisciplinary teams, a revision of each others’ assumptions can be necessary. Existing integration approaches of personas and use cases do not reflect sufficiently on the different understanding of the goal concept. This paper shows that the tight coupling of tasks and goals as is common in use case specifications should be considered critical. The suggested distinction and interplay between organizational and user goals acknowledges differences and similarities in the ways tasks are performed by the different people in a work system. It is a valuable conceptual contribution that can also be applied to hierarchical task modeling approaches in human-computer interaction [13].

The introduced adaptations to use case notations allow the consideration of persona-specific sets of goals without resulting in a large set of use case specifications with overlapping descriptions of interactions between user and system, which is less manageable and prone to inconsistencies. Adaptation 1 rejects the idea of one basic course or main success scenario. The persona-specific basic courses are easy to read and compare (Figs. 7 and 8). This notation is to be preferred if the personas have the same priority as Adrian and Stephanie in the example. Adaptation 2 (Figs. 7 and 9) rather considers persona-specific goals and behaviors as non-frequent alternatives or interruptions from the basic course and is more applicable to personas with lower priority.

In contrast to integration approaches such as [1], we do not aim at adapting a prescriptive software development method. Instead, we take a design-oriented perspective and consider design activities as domain-specific construction of design representations [42]. In [7], we argue that designers need to be supported in generating ideas, but also in comparing different design representations and in understanding how they are related and whether or not they satisfy some initial or evolving design specifications and constraints. Generally, it is still challenging to

effectively couple different design representations in interaction design [7]. The proposed stronger interweaving of personas and use case models supports their co-development and the integration of different design perspectives. Use case models and personas change their character with the suggested adaptations. For instance, implications from single personas can be traced in the models and the proposed <<automate>>-relationship facilitates thinking about automation and interaction. However, the changes are not radical ones and it can be assumed that they help to 'bridge' practices of interaction designers and software engineers.

Personas help to view interactive systems from an individual's perspective, although one should keep in mind that they are abstractions from user groups themselves which can complement or support, but not replace, an active involvement of users and other stakeholders into the design process. The success of the suggested integration depends to a great extent on the description of the personas' appropriation of organizational goals. 'Flat' descriptions (see [35]) where personas do not reflect on their roles and responsibilities do not encourage more discussion among the design team members nor do they make abstract roles and corresponding actors in use case models 'more alive'.

The suggested concepts and notations for an integrated use of personas and use cases were successfully used in teaching. Empirical evidence of the benefits to software development practice still needs to be shown. In [24], an action research approach was applied to investigate how changes in the development process aimed at improving user involvement and usability influenced the outcome of the process. A similar approach could be taken here.

5 Conclusions and Future Work

With personas and use cases, two types of design representations have been investigated that are popular and commonly used in the disciplines of interaction design and software engineering respectively. An integration of these representations has been proposed which preserves their essential character, but at the same time supports creative collaboration of interaction designers, software engineers and other stakeholders in capturing requirements for role-based interactive systems from an organizational perspective as well as from the users' perspectives. The adaptations are based on a revised conceptual understanding of goals and task-goal relationships and include task-related user goals for personas, persona-specific goals and actions in use cases, and the <<automate>>-stereotype in use case diagrams. A carefully worked out example serves as an initial validation of the integration approach, but future empirical studies are needed to examine its effectiveness. At the conceptual level, we want to investigate in future work how to better relate the adapted use case models to other design representations from interaction design such as 'rich' scenarios and user interface models.

Acknowledgements. We thank Louisa Jochum for the discussions and her help.

References

1. Acuña, S.T., Castro, J.W., Juzgado, N.J.: A HCI technique for improving requirements elicitation. *Inf. Soft. Technol.* **54**(12), 1357–1375 (2012)
2. Baker, M., Détienne, F., Burkhardt, J.M.: Quality of collaboration in design: articulating multiple dimensions and viewpoints. In: 1st Interdisciplinary Innovation Conference, Paris, France (2013)
3. Bellotti, V., Shum, S., MacLean, A., Hammond, N.: Multidisciplinary modelling in HCI design in theory and in practice. In: SIGCHI Conference on Human Factors in Computing Systems, CHI 1995, pp. 146–153. ACM (1995)
4. Bittner, K., Spence, I.: Use Case Modeling. Addison-Wesley, Boston (2002)
5. Blomquist, A., Arvola, M.: Personas in action: ethnography in an interaction design team. In: Proceedings of NordiCHI 2002, pp. 197–200. ACM (2002)
6. Bødker, S., Christiansen, E., Nyvang, T., Zander, P.O.: Personas, people and participation: challenges from the trenches of local government. In: Proceedings of PDC 2012, pp. 91–100. ACM (2012)
7. Bowen, J., Dittmar, A.: Coping with design complexity: a conceptual framework for design alternatives and variants. In: Bernhaupt, R., Dalvi, G., Joshi, A., Balkrishnan, D.K., O’Neill, J., Winckler, M. (eds.) INTERACT 2017. LNCS, vol. 10513, pp. 483–502. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67744-6_30
8. Castro, J.W., Acuña, S.T., Juzgado, N.J.: Enriching requirements analysis with the personas technique. In: I-USED. CEUR Workshop Proceedings, vol. 407. CEUR-WS.org (2008)
9. Chapman, C.N., Milham, R.P.: The personas’ new clothes: methodological and practical arguments against a popular method. In: The Human Factors and Ergonomics Society Annual Meeting, pp. 634–636. SAGE Publications (2006)
10. Cockburn, A.: Writing Effective Use Cases. Addison-Wesley, Boston (2000)
11. Cooper, A.: The Inmates are Running the Asylum. Macmillan Publishing Co., Inc., Indianapolis (1999)
12. Cooper, A., Reimann, R.: About Face 3.0: The Essentials of Interaction Design, 3rd revised edn. Wiley, Hoboken (2007)
13. Diaper, D., Stanton, N.: The Handbook of Task Analysis for Human-Computer Interaction. L. Erlbaum Associates Inc., Hillsdale (2004)
14. Diaper, D.: Understanding task analysis for human-computer interaction. In: Diaper, D., Stanton, N. (eds.) The Handbook of Task Analysis for Human-Computer Interaction. L. Erlbaum Associates Inc., Hillsdale (2004)
15. Diaper, D., Sanger, C.: Tasks for and tasks in human-computer interaction. *Interact. Comput.* **18**(1), 117–138 (2006)
16. Dittmar, A., Hensch, M.: Two-level personas for nested design spaces. In: Proceedings of CHI 2015, pp. 3265–3274 (2015)
17. Dix, A., Gongora, L.: Externalisation and design. In: Second Conference on Creativity and Innovation in Design, pp. 31–42. ACM (2011)
18. Eriksson, E., Artman, H., Swartling, A.: The secret life of a persona: when the personal becomes private. In: Proceedings of CHI 2013, pp. 2677–2686. ACM (2013)
19. Faily, S.: Bridging user-centered design and requirements engineering with GRL and persona cases. In: de Castro, J.B., Franch, X., Mylopoulos, J., Yu, E.S.K. (eds.) Proceedings of the 5th International *i** Workshop 2011. CEUR Workshop Proceedings, vol. 766, pp. 114–119. CEUR-WS.org (2011)
20. Floyd, I.R., Jones, M.C., Twidale, M.B.: Resolving incommensurable debates: a preliminary identification of persona kinds, attributes, and properties. *Artifact J. Virtual Des.* **2**(1), 12–26 (2008)

21. Francescomarino, C.D., et al.: A bit of “persona”, a bit of “goal”, a bit of “process” ... a recipe for analyzing user intensive software systems. In: iStar. CEUR Workshop Proceedings, vol. 586, pp. 36–40. CEUR-WS.org (2010)
22. Grudin, J.: Why personas work: the psychological evidence. In: Pruitt, J., Adlin, T. (eds.) *The Persona Lifecycle: Keeping People in Mind Throughout Product Design*. Morgan Kaufmann Publishers Inc., San Francisco (2005)
23. Grudin, J., Pruitt, J.: Personas, participatory design and product development: an infrastructure for engagement. In: *Proceedings of PDC 2002*, pp. 144–161 (2002)
24. Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Cajander, Å.: Key principles for user centred systems design. *Behav. Inf. Technol.* **22**(6), 397–409 (2003)
25. ISO: ISO 9241-210:2010 - Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems (2010). http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=52075
26. Jacobson, I., Christerson, M., Jonsson, P., Övergaard, G.: *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, Boston (1992)
27. Jacobson, I., Spence, I., Kerr, B.: Use-case 2.0. *Commun. ACM* **59**(5), 61–69 (2016)
28. Jarrahi, M.H.: A structural analysis of how course management systems are used in practice. *Behav. Inf. Technol.* **29**(3), 257–275 (2010)
29. Jones, M.C., Floyd, I.R., Twidale, M.B.: Teaching design with personas. In: *Proceedings of HCIEd (2008)*
30. Kulak, D., Guiney, E.: *Use Cases: Requirements in Context*. Addison-Wesley, Boston (2000)
31. Larman, C.: *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, 3rd edn. Prentice Hall, Upper Saddle River (2004)
32. Mackay, W.E.: Educating multi-disciplinary design teams. In: *Proceedings of Tales of the Disappearing Computer*, Santorini. ACM Press (2003)
33. Metz, P., O’Brien, J., Weber, W.: Specifying use case interaction: types of alternative courses. *J. Object Technol.* **2**(2), 111–131 (2003)
34. Miller, G., Williams, L.: *Personas: moving beyond role-based requirements engineering*. Technical report, North Carolina State University. Department of Computer Science (2006)
35. Nielsen, L.: From user to character: an investigation into user-descriptions in scenarios. In: *Proceedings of DIS 2002*, pp. 99–104. ACM (2002)
36. Nielsen, L.: Personas In “*The Encyclopedia of Human-Computer Interaction*, 2nd edn.” (2013). <http://www.interaction-design.org/encyclopedia/personas.html>
37. Nielsen, L., Hansen, K.S.: Personas is applicable: a study on the use of personas in Denmark. In: *CHI*, pp. 1665–1674 (2014)
38. Norman, D.: Ad-hoc personas & empathetic focus. In: Pruitt, J., Adlin, T. (eds.) *The Persona Lifecycle: Keeping People in Mind Throughout Product Design*. Morgan Kaufmann Publishers Inc., San Francisco (2005)
39. Randolph, G.: Use-cases and personas: a case study in light-weight user interaction design for small development projects. *Informing Sci. Int. J. Emerg. Transdiscipline* **7**, 105–116 (2004)
40. Schneidewind, L., Hörold, S., Mayas, C., Krömker, H., Falke, S., Pucklitsch, T.: How personas support requirements engineering. In: *Proceedings of the First International Workshop on Usability and Accessibility Focused Requirements Engineering, UsARE 2012*, pp. 1–5. IEEE Press (2012)
41. Sim, W.W., Brouse, P.S.: Empowering requirements engineering activities with personas. *Procedia Comput. Sci.* **28**, 237–246 (2014)

42. Visser, W.: Designing as construction of representations: a dynamic viewpoint in cognitive design research. *Hum.-Comput. Interact.* **21**(1), 103–152 (2006)
43. Wärnestål, P., Svedberg, P., Nygren, J.: Co-constructing child personas for health-promoting services with vulnerable children. In: *Proceedings of CHI 2014*, pp. 3767–3776 (2014)