Extending the Information of Activity Diagrams with a User Input Classification

Cindy Mayas, Stephan Hörold, and Heidi Krömker

Ilmenau University of Technology, Ilmenau, Germany
{cindy.mayas,stephan.hoerold,heidi.kroemker}@tu-ilmenau.de

Abstract. This paper presents an extended notation of actions in activity diagrams. The suggested method combines activity diagrams with a user input classification in order to support interdisciplinary teams, particularly in the early phases of development. In this way, the user input classification serves as a communication basis for user requirements, which is adapted to the needs of software engineers. The method is evaluated within a case study in a nationwide research project for public transport.

Keywords: activity diagram, actions, user input classification, public transport.

1 Introduction

These days usability is an important quality for software engineering [1], [2] and a key factor for successful and profitable products [3]. Integration processes of human-computer interaction (HCI) in software engineering (SE) are widely discussed. But the communication between usability engineers and software developers often occurs too late in the development process, though key decisions are made in the early development phases. Consequently, changes in the later development process are difficult and cause a higher development effort [4].

In order to provide usability information for key decisions of the development as early as possible, an adequate communication and integration of the results of user requirements analysis into the software engineering workflow is essential. Therefore, the existing and detailed knowledge about the user requirements have to be adapted to the requirements documentation of SE. Along these lines, the development processes require a more user-oriented documentation regarding software engineers.

Existing approaches to bridge this gap between HCI and SE are presented and analyzed in chapter 2 and advantages of UML are covered. Consequently, an approach to enrich UML activity diagrams with usability information and the according case study is presented in chapter 3 and 4.

The key contribution of this paper to bridge this gap is the suggested approach, which supports the interdisciplinary communication about usability requirements in early development phases. The included user input classification (UIC) can serve as a guide for early architectural decisions and in this way increases the usability quality of software products.

M. Kurosu (Ed.): Human-Computer Interaction, Part I, HCII 2013, LNCS 8004, pp. 391–400, 2013.

2 Background

2.1 Bringing HCI and SE Together

Addressing the integration of HCI in SE processes, many authors focus on combining the methods of HCI and SE. Exemplary here, only some approaches can be mentioned: the usage-centered design of Constantine[5], the user-centered software development process of Ferre [6] or Nebe's approach to integrate software engineering and usability engineering [7]. Other solutions prefer separate procedures to analyze the usability during a software engineering process, such as Folmer's SALUTA [1].

But in contrast to an adaptation of methods and procedures, the presented approach of integrating a user input classification to UML diagrams suggests an adaptation of communication tools. Actually, the communication of user requirements is focused on the workflow of usability engineers, despite of the workflow of software engineers.

According to Bruegge [8] an appropriate communication notation has to meet the following three criteria: well-defined semantics, well suited for representing, and well understood. Established kinds of analysis documentation in HCI [9], which depend on the solution and can be used through the development process, have strengths and weaknesses according to these criteria as shown in table 1.

Table 1. Overview of established kinds of analysis documentation

Representation form (examples)	*Semantics*	*Representativity*	*Understandability*
Mainly narrative (e.g. Personas [10], Scenarios [11])	loosely defined	moderate	high
Mainly pictoral or artifactual (e.g. Mood boards, Culture Cards [12])	undefined	high	high
Mainly tabular (e.g. user needs, usability goals [13])	partly defined	low	moderate
Mainly diagrammatic (e.g. contextual workflow [14], UML [15])	well-defined	high	moderate

Mainly narrative, pictoral, and artifactual forms of representation benefit from a general understandability, but they lack of well-defined semantics. In addition, the mainly tabular forms of representation have some partly defined semantics, such as the use of signal verbs to prioritize user requirement and usability goals.

The Unified Modeling Language (UML) is one kind of diagrammatic representation. The advantages of well-defined semantics of diagrams cause a moderate understandability, particularly for users without the basic knowledge of the semantic. But diagrams are well understood tools for users with prior knowledge of the semantic. Regarding UML, which is widely common in several disciplines, such as SE and HCI, we agree with Bruegge [8], that UML meets the three criteria very well.

2.2 UML in HCI and SE

Due to the flexibility of UML, the use of UML in HCI and SE is very multifaceted. UML is widespread in the software industry, especially for object-oriented software engineering [8]. The use of UML for HCI was intensively discussed and extended in TUPIS2000 workshop, see [17-19]. Nowadays, especially the UML behavioral diagrams became established tools for usability engineers.

Several kinds of diagrams are available to describe the structure, behavior, and interaction of concepts [15]. But the use of diagrams depends on the elaboration of the solution. While an abstract idea of the solution is sufficient for some diagrams, such as use case diagrams, e.g. sequence diagrams require a more detailed idea of the solution. Thus, not every artifact is suitable to serve as a basis for user input classification. For instance, tasks are hierarchic and abstract descriptions of user goals, which are independent from the solution [9]. In contrast, the description of activities and actions bases on the tasks and requires a detailed idea of the solution up to a certain extend.

For this reasons, activity diagrams are used after the requirements analysis in HCI and SE processes as well. We choose the action nodes of activity diagrams to integrate the additional information about the user input.

3 Approach

3.1 User Input Classification (UIC)

Based on the results of user requirements analysis, an interdisciplinary team develops first ideas for technical solutions for possible background systems and interaction devices. For each solution schematic activity diagrams display the required interactions for the user. Thus, each action can be analyzed according to user input. Some examples of possible user interaction should be collected per action and build the bases for the classification.

The user input classification (UIC) bases on a pre-tested 5-point-scale, which displays the interaction task and the semantic design of a user input according to Foley [20]. Foley defines an interaction task as “the entry of a unit of information by the user” and distinguishes “position”, “text”, “select”, and “quantify” as the four basic interaction tasks [20]. In addition to the four interaction tasks, we enrich this set with the “confirmation” interaction, which is defined as a special kind of selection with only one or two alternatives and is often used to accomplish other input interactions or an information output.

3.2 Stereotypes in UML

In order to transfer the suggested UIC into UML, according UML stereotypes were defined. UML stereotypes are a kind of profile to classify UML elements with a “virtual meta-model concept” [20]. For instance, the Wisdom Approach of Nunes [19] provides stereotypes for interaction modeling with UML and Lieberman [21] presents stereotypes for activity diagrams detailing the user interface navigation.

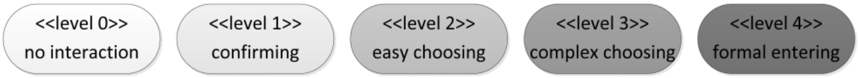

Fig. 1. Proposal for a UIC coloring concept for actions

We predefined a set of coloring information in grey shades for the background of the actions, as shown in figure 1, in order to display the different meaning between the actions with identical structure [15]. Lower stages are colored with a lighter grey than higher stages, according to defined shades. By these means, the density of user input is visualized by the lightness of the activities and provides further information about the users' interaction, in addition to the number of actions. The suggested set of stereotypes consists of five stereotypes, one per each level of UIC:

- <<Level 0>> The basic level requires no input interaction from the user. Information, which is already documented within the software or can be derived from, does not require an additional user input.
- <<Level 1>> The first level includes easy interactions, which do not require an input information, but rather confirmation of the user to proceed. Confirming actions are used at the end of a dialog box to return to the main page or in more complex interaction sequences to display the next dialog.
- <<Level 2>> The second level includes selecting tasks for small-sized choices, for example yes/no decisions or small menus with less than seven items, which are familiar and structured according to the knowledge of the user.
- <<Level 3>> The third level of UIC includes both, more complex selections with semantically well elaborated content, e.g. the choice of the home country from a list, as well as easy data entering, e.g. numeric quantities within a defined range or well-known text input, e.g. my name.
- <<Level 4>> The highest level 4 displays very complex interactions, particularly with semi-formalized or formalized input. For example entering the long and unfamiliar passport number in a visa tool is a very complex entering task, according to its formalized structure.

3.3 Procedure Model to Identify the Classification Level

Figure 2 shows the suggested procedure model to identify several classification levels of user interfaces. A conducted task analysis is the precondition to start the classification. We also recommend collecting some examples for user input for each action as a common classification basis. The following stages have to be passed through for each action:

- identifying the user interaction task,
- refining the user input for selection and entering tasks,
- identifying the dimension of semantic, which is used for the interactions,
- attributing the results to the UML stereotypes.

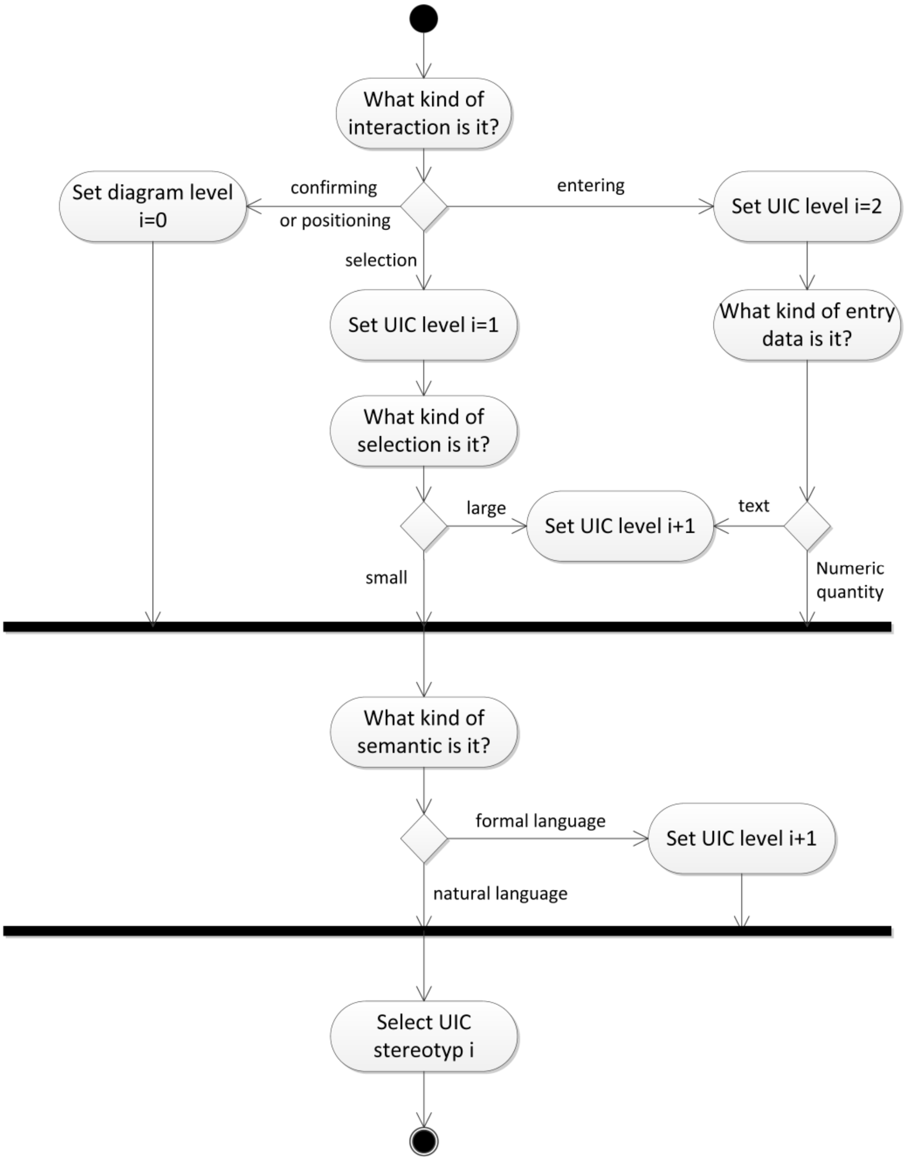

Fig. 2. Suggested procedure model to define UIC classes

4 Case Study

The application of the UIC was evaluated in a case study within the German research project IP-KOM-ÖV. The effort and understandability were evaluated with a standardized questionnaire with members of the interdisciplinary team, as described in section 4.1, and revealed first positive effects for user-centered development in interdisciplinary teams. Furthermore, the UIC was applied to evaluate activity diagrams for the passenger-to-vehicle-communication. The effects of this application to the decision making process are discussed in section 4.2.

4.1 Pre-evaluation

We conducted a pre-evaluation of the UIC approach in one working group of IP-KOM-ÖV with 10 participants. 90 per cent of the participants from the fields of human-computer interaction, software developments and management were familiar or even very familiar with UML. The participants were most used to state machine diagrams, activity diagrams, use case diagrams and sequence diagrams for designing the behavioral structure.

In the first step, the congruence of classifying the action according to the task interaction is measured with eight test actions. The test actions are consciously formulated neutral from the interaction task, in order to avoid an influence on the decisions. The congruence of six of eight actions was greater than 70 per cent, two tasks reached only 60 per cent. Deviating appraisals were caused especially by two multi-faceted tasks, which required more than one user interaction, for example choosing a point on a map. Another factor for slight deviations is the ambiguity between entering and selecting for larger groups of possible items. In these cases, few participants chose an entering interaction despite of an selection interaction.

In the second step, the participants classified the selection and entering interactions more detailed regarding the number of items and the entering string or number. While the congruence for entering a text or numeric quantities is 90 per cent the congruence of most kinds of selection is about 60 per cent. The appraisal between small and large numbers of items is not obvious and has to be improved by a hard indicator.

In the third step, the participants appraised the semantic of the user input. The congruence of all semantic appraisals were again only 60 per cent.

Finally, the result chart with the UIC classification was pre-evaluated by the assessment of clarity, intelligibility, and applicability. All evaluated items were assessed as positive from 90 respectively 100 per cent of the participants.

These results indicate, that the classification of user input is comprehensible to interdisciplinary teams. In contrast, the process of classification should be passed by usability experts in order to obtain more reliable data.

4.2 Application in the Decision Process

We evaluated the practical applicability of the UIC in another working group of IP-KOM-ÖV by applying the UIC to a decision process. Subsequent to the requirements analysis [22],[23], alternative proposals for the passenger-to-vehicle-communication were developed and analyzed by activity diagrams with UIC.

In one meeting we compared activity diagrams for the build-up of a WLAN connection by manual selection, ad hoc network selection, QR-code scan, or NFC support, see exemplary diagrams in figure 3. In addition, we considered different initial situations in order to vary user input.

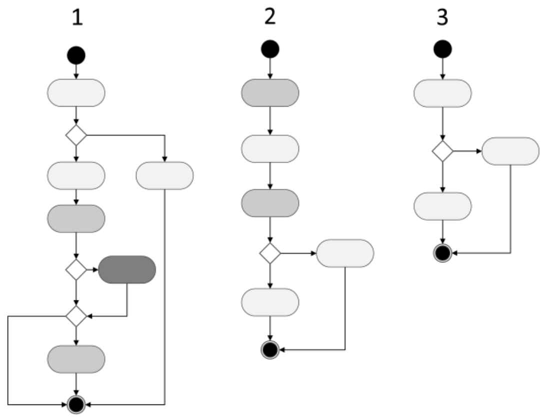

Fig. 3. Exemplary activity diagrams with UIC stereotypes within IP-KOM-ÖV

Subsequent to the discussion of several actions and exemplary user input, we used the criteria number of actions, maximum level of UIC, and mean level of UIC to compare the solutions. While the number of actions showed only slight differences for some solution, for instance between example (1) and (2) in figure 3, the mean level of UIC, and especially the required maximum revealed considerable differences between the suggested solution. While example (3) consists of level-1-actions only, example (2) requires level-1- and level-2-actions. At least, example (1) also contains level-3-actions.

As a consequence, possible improvements for the usability for technical solutions, which contained level-3- or level-4-interactions were discussed intensively. The impact of the UIC was intuitively understood with less explanations by all participants of the discussion. Next to the technical requirements, the information about the user requirements was directly taken into consideration for the decision process, according to the UIC.

5 Discussion

The UIC presupposes a task structuring process such as HTA [24] or provided by Martinie [25]. It is not possible to classify ambiguous tasks which are broadly defined and might include several different user inputs. Hence, the results of the classification depend on the results of the task analysis.

Target products with more than one user group, might evoke ambiguities of the UIC, due to the fact that the semantic model of the content depends on the users' mental models and daily routines. In relation to the user descriptions, the UIC value might differ for several user groups and require more than one value for one activity and additional information, at the same time. Furthermore, the requirements of handicapped users have to be considered separately, particularly in regard to user input technologies. This adaptation to differing user groups should be the subject in further research.

6 Conclusions

The extension of activity diagrams with UIC combines aspects of usability engineering with software engineering methods. Thus, the UIC extension provides an interdisciplinary approach, particularly for the analysis and comparison of proposed solutions in early phases of development. Decisions concerning the user interaction can be discussed in early phases of the development and can be taken into account for technical key decisions.

Furthermore, the UIC stereotypes support conclusions to improve a required user input. For instance, a level-3-selection out of many unstructured items could be improved by a hierarchical level-2-selection or a level-4-entering task with formal text input could be supported by proposals of the most frequent terms. Thus, according to the UIC stereotypes, recommendations for improvements could be given to the actions, in the future.

According to the positive first evaluation results in theory and in the field, the application of the classification should be evaluated more detailed in an entire workflow process with interdisciplinary teams. In this paper the UIC approach is applied solely to activity diagrams. In a next step we have to proof, if the UIC can be transferred to other behavioral diagrams.

Acknowledgements. Part of this work was funded by the German Federal Ministry of Economy and Technology (BMWi) grant number 19P10003L within the IP-KOM-ÖV project. The project develops an interface standard for passenger information in German public transport with focus on the connection between personal mobile devices, vehicle systems and public transport background computer systems.

References

1. Folmer, E., van Gurp, J., Dannenberg, R.B.: Software architecture analysis of usability. In: Bastide, R., Palanque, P., Roth, J. (eds.) EHCI-DSVIS 2004. LNCS, vol. 3425, pp. 38–58. Springer, Heidelberg (2005)
2. Jerome, B., Kazman, R.: Surveying the Solitudes. An Investigation into the relationships between Human Computer Interaction and Software Engineering in Practice. In: Human-Centered Software Engineering — Integrating Usability in the Software Development Lifecycle, pp. 59–70. Springer, Heidelberg (2005)

3. Williams, J.P., Bias, R.G., Mayhew, D.J.: Cost Justification. In: Seras, A., Jacko, J.A. (eds.) The Human-computer Interaction Handbook. Fundamental, Evolving Technologies and Emerging Applications, pp. 927–947. Taylor and Francis Group, New York (2008)
4. Hackos, J.T., Redish, J.C.: User and task analysis for interface design. Wiley, New York (1998)
5. Constantine, L., Biddle, R., Noble, J.: Usage-Centered Design and Software Engineering: Models for Integration. In: ICSE 2003 Workshop on Bridging the Gap Between Software Engineering and Human-Computer Interaction, Portland, Oregon (2003)
6. Ferre, X.: Integration of Usability Techniques into the Software Development Process. In: ICSE 2003 Workshop on Bridging the Gap Between Software Engineering and Human-Computer Interaction, Portland, Oregon (2003)
7. Nebe, K., Zimmermann, D., Paelke, V.: Integrating Software Engineering and Usability Engineering. In: Pinder, S. (ed.) Advances in Human Computer Interaction. InTech (2008)
8. Bruegge, B., Dutoit, A.H.: Object-Oriented Software Engineering – Using UML, patterns and Java. Pearson, Boston (2010)
9. Courage, C., Redish, J.G., Wixon, D.: Task Analysis. In: The Human-Computer Interaction Handbook. Fundamental, Evolving Technologies and Emerging Applications, pp. 927–947. Taylor and Francis Group, New York (2008)
10. Cooper, A., Reimann, R., Cronin, D.: About face 3, the essentials of interaction design. Wiley, Indianapolis (2007)
11. Rosson, M.B., Carroll, J.M.: Usability engineering –Scenario –based development of humen-computer interaction. Morgan Kaufmann (2002)
12. Foucault, B.E.: Contextualizing cultures for the commercial world: Techniques for presenting field research in business envirnoments. In: Salvendy, G. (ed.) Proceedings of the HCII Conference, Las Vegas (2005)
13. Mayhew, D.: The Usability Engineering Lifecycle. Morgan Kaufmann, San Francisco (1999)
14. Beyer, H., Holtzblatt, K.: Contextual design: defining customer-centered systems. Morgan Kaufmann, San Francisco (1998)
15. OMG: Unified Modeling Language (OMG UML), Infrastructure Version 2.4, http://www.omg.org/spec/UML/2.4.1/Infrastructure
16. Nunes, J.N.: What drives software development: Bridging the gap between software and usability engineering. In: Seffah, A., Vanderdonckt, J., Desmarais, M.C. (eds.) Human-Centerd Software Engineering, Software Engineering Models, Patterns and Architectures for HCI. HCIS, pp. 9–25. Springer, London (2009)
17. Markopoulos, P.: Modelling user tasks with the unified modeling language. In: Proc. of Workshop TUPIS 2000 in < <UML2000> > International Conference
18. Cook, S.: The UML Family: Profiles, Prefaces and Packages. In: Evans, A., Caskurlu, B., Selic, B. (eds.) UML 2000. LNCS, vol. 1939, pp. 255–264. Springer, Heidelberg (2000)
19. Jardim Nunes, N., Falcão e Cunha, J.: Towards a UML Profile for Interaction Design: The Wisdom Approach. In: Evans, A., Kent, S., Selic, B. (eds.) UML 2000. LNCS, vol. 1939, pp. 101–116. Springer, Heidelberg (2000)
20. Foley, J.D., van Dam, A., Feiner, S.K., Hughes, J.F.: Computer Graphics – Principles and Practice. Addison-Wesley, Boston (1997)
21. Lieberman, B.: UML Activity Diagrams: Detailing User Interface Navigation, http://www.ibm.com/developerworks/rational/library/4697.html (February 22, 2013)

22. Mayas, C., Hörold, S., Krömker, H.: Meeting the Challenges of Individual Passenger Information with Personas. In: Stanton, N. (ed.) Advances in Human Aspects of Road and Rail Transportation, pp. 822–831. CRC Press, Boca Raton (2012)
23. Hörold, S., Mayas, C., Krömker, H.: Identifying the information needs of users in public transport. In: Stanton, N. (ed.) Advances in Human Aspects of Road and Rail Transportation, pp. 331–340. CRC Press, Boca Raton (2012)
24. Annet, J.: Hierarchical Task Analysis (HTA). In: Handbook of Human Factors and Ergonomics Method, pp. 33–37. CRC Press, Boca Raton (2005)
25. Martinie, C., Palanque, P., Winckler, M.: Structuring and Composition Mechanisms to Address Scalability Issues in Task Models. In: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (eds.) INTERACT 2011, Part III. LNCS, vol. 6948, pp. 589–609. Springer, Heidelberg (2011)