On Using the Task Models for Refinement and Validation of Requirements Generated through Co-creation

Naqvi Bilal, Martinie Célia, Bakhaev Stepan, Smolander Kari

DOI: 10.1007/978-3-031-61688-4_5

On Using the Task Models for Refinement and Validation of Requirements Generated through Co-creation

Bilal Naqvi[1, ✉ [0000-0001-5271-5604]], Célia Martinie[2 [0000-0001-7907-3170]], Stepan Bakhaev[1 [0000-0002-1130-8743]], and Kari Smolander[1 [0000-0002-7043-0458]]

[1] Software Engineering, LENS, LUT University, Lappeenranta, 53850, Finland
[2] ICS-IRIT, Université Toulouse III Paul Sabatier, Toulouse, France
syed.naqvi@lut.fi

Abstract. Among the several other approaches for gathering software requirements, co-creation is often used. The co-creation approach based on elements of participatory research is aimed at involving the end users during the requirements elicitation process. While this approach has many merits, certain limitations need to be addressed. Two limitations include (i) limitations induced due to the subjective judgment of requirements analysts during the analysis and translation of user statements, goals, and desires (gathered during co-creation) into software requirements, and (ii) limitations induced due to varying abilities of different users to conceptualize the systems being developed during early phases of the system development lifecycle, specifically during the co-creation workshops conducted for requirements gathering. To address these limitations the paper proposes a three-step task model-based approach for validation and refinement of requirements generated through co-creation. To instantiate the approach, the paper also presents an illustrative case study featuring the development of a novel electronic identity (e-ID) scheme.

Keywords: co-creation, requirements, task models, usability, usable security, validation

1 Introduction

Amid the digital transformation which has arguably revolutionized every sector from finance to health and from education to governance, several challenges have also emerged that need to be addressed. Among these challenges are cyber security threats, several of these cyber security threats can be attributed to human mistakes and negligence. Consequently, there has been a focus on the development of more robust yet usable security technologies and mechanisms that can withstand threats from both internal and external attackers while ensuring user experience. One such technology that enables ease of use (in terms of the less cognitive burden on the user's mind) while ensuring security is a biometric-based electronic identity (e-ID) scheme for authentication.

Biometrics-based e-ID authentication is centered around authenticating an individual based on their anatomical, physiological, and behavioral characteristics [2]. It includes a variety of methods, such as fingerprint scans, finger, hand geometry, iris, or retina scans, etc. As said earlier, besides the well-documented security benefits, biometric-based e-ID schemes are based on the premise of reducing the cognitive burden on the user.

Furthermore, there has been a debate on involving the end-users during the system development lifecycle to ensure a human-centered focus in the development of novel technologies. Co-creation approach attempts to actively involve the people who are being served by a design, in the process of problem-solving [19]. While users may not be able to communicate precisely or technically some requirement or other, they can explain their goals and how they approach their tasks [12]. Therefore, users should not be passive informants as they have different values in relation to a system and its use [13]. Involving the end-users to participate in a design process by means of co-creation practices encourages the elicitation of requirements from a human-centered perspective. However, with the benefits of this approach, there might be additional challenges, such as during the translation of users' goals into software requirements, it is critical to accurately translate the users' goals and statements into software requirements. Moreover, an additional challenge is that the users at the early stages of the software development lifecycle might not be able to conceptualize the system and accurately contribute their desires for the system being built. It is therefore vital to validate these requirements before development.

This paper considers a primary research question i.e., *'how to validate and refine the user requirements gathered by co-creation before development?'*. In line with the research question and to overcome the challenges discussed earlier, the paper proposes a three-step task models-based approach to validate the requirements gathered by co-creation approaches. To instantiate the approach, we present a case study featuring the development of a novel biometric-based e-ID scheme across six public administrations in five European countries. The limitations in the requirements gathered by co-creation were identified using the proposed approach and the requirements were refined.

Structure of the paper: Section 2 presents the co-creative approach for requirements elicitation, Section 3 presents the task model-based approach for validation of the requirements generated from co-creation, Section 4 presents the validation case study, Section 5 presents the related work, Section 6 presents the discussion, and Section 7 concludes the paper.

2 Co-creative approach for requirements elicitation

This section presents the co-creation approach adopted for the identification of requirements for the development of an e-ID scheme. The same approach was executed across six public administrations in five European countries. During the co-creation process, workshops at each public administration were conducted to identify the requirements. A common (five-step) template was created for all workshops. The workshops were executed by each public administration locally in both physical and online settings. The

participants (prospective users of the technology) were asked to give their consent individually for participation in the workshop. The total number of attendees in each workshop was between 12 and 20 persons. During each workshop, data in the form of audio recordings, screen captures, and other visual recordings as well as the minutes taken during the workshop sessions were recorded and later analysed for identification of the software requirements. The data was garnered in the local languages of the respective workshop venue; however, it was translated into English before being analysed and incorporated into the findings. Each workshop was organized in 5 steps.

- **Step 1 – Introducing the workshop goals and objectives:** At the beginning of each workshop session, participants were introduced to the key objectives and goals. The workshop facilitators communicated these objectives framed in a "What? How? Why?" outline, using short and simplified statements.
- **Step 2 – The positives:** This step involved exploring the positive user expectations from the e-ID technology. This was mainly a warm-up exercise designed to engage the participants and acclimate them to the workshop format.
- **Step 3 – The negatives**: This step involved exploring the user's pain points. It was completed in three parts, i.e., activities aimed at defining and prioritizing the potential problems. Accordingly, in the first part, participants were tasked to write down their concerns and issues. These negative anticipations reflected the potential drawbacks in the technology design from a human-centred perspective. Next, the participants were asked to vote for the concerns they deemed most important using red voting dots. Each of the participants was given three of those dots to vote for any idea placing one, two, or all three dots on one or more sticky notes with a negative statement. Finally, the participants were asked to engage in discussion about the top-voted concerns and explain their priorities.
- **Step 4 – "How might we…?":** This step was done in two rounds where at least five top-voted problems from Step 3 served as inputs. The participant's task was framed to overcome the concerns (from the third step), devise solutions, and think of the benefits they can get from them. For this task, the workshop participants are randomly assigned to five groups. This activity was designed to represent the World Café method.
- **Step 5 – Categories**: The final step of the workshop was essentially an open discussion about the benefits and solutions that the participants had voted for in the previous step. All participants could overview the statements selected in the priorities to find similar ideas and create the categories. These categories were labelled based on their perceptions and feelings evoked with the solution in mind.

After the workshops, the user statements grouped by the context were assigned with unique identifiers that reflected the case study origin, the nature of the requirement (i.e., 01 – user expectation, 02 – user pain point, 03 – user need), and a serial number. Furthermore, a cross-case analysis was performed to identify the similarities and differences across the six different cases. Finally, the resembling features of the user needs and concerns identified in the cross-case analysis allowed for transforming the user statements into formal requirements. For exemplary purposes, a subset of the requirements gathered following the process just discussed are presented in Table 1.

Table 1. List of requirements gathered using the co-creation approach

Requirement statement	
1.	The system shall be interoperable with the legacy (national) e-ID schemes to ensure user uptake
2.	The system shall inform users about the processing of their data
3.	The system should provide simple and well-guided user actions when collecting image samples for face recognition
4.	The system shall provide users with informed consent in a legal language and accessible with dedicated icons
5.	The system shall allow users to control their data in a self-sovereign manner
6.	The system shall reduce the cognitive burden (remembering many user accounts and passwords) for users
7.	The system shall prevent unauthorized access and processing of user data
8.	The system shall issue a challenge to verify that the user who is trying to authenticate is the owner of the VC (e.g., OTP via SMS or email)

3 Task model-based approach for the refinement and validation of requirements

Before discussing the approach, it is vital to discuss the rationale behind it. The software requirements generated from co-creation approaches (such as the one just discussed) can be prone to (i) limitations induced due to the subjective judgment of requirements analysts during the analysis and translation of user statements, goals, and desires (gathered during co-creation) into software requirements, and (ii) limitations induced due to varying abilities of different users to conceptualize the systems being developed during early phases of the system development lifecycle (specifically during the co-creation workshops conducted for requirements gathering). Therefore, to overcome these limitations, we propose a three-step approach for the validation of software requirements gathered by co-creation (see Fig.1).

- The *first step* involves modelling the user tasks according to each requirement. The task models depicting the users' tasks are generated and analysed.
- The *second step* involves the assessment of the task models (generated in the first step) and identifying possible limitations. If the task model depicts that the elements of usability have been catered to, the requirements are finalized for development. However, if there are limitations identified, the proposal for refinements is created and the requirements are refined to overcome these limitations.
- The *third step* involves modelling user tasks according to the refined requirements. The task models thus specified are analysed. If the task model depicts that the elements of usability have been catered to, the refined requirements are subjected to validation by the participants of the co-creation workshop and finalized for development. However, if there are limitations identified, another refinement iteration (starting from the second step) is performed.

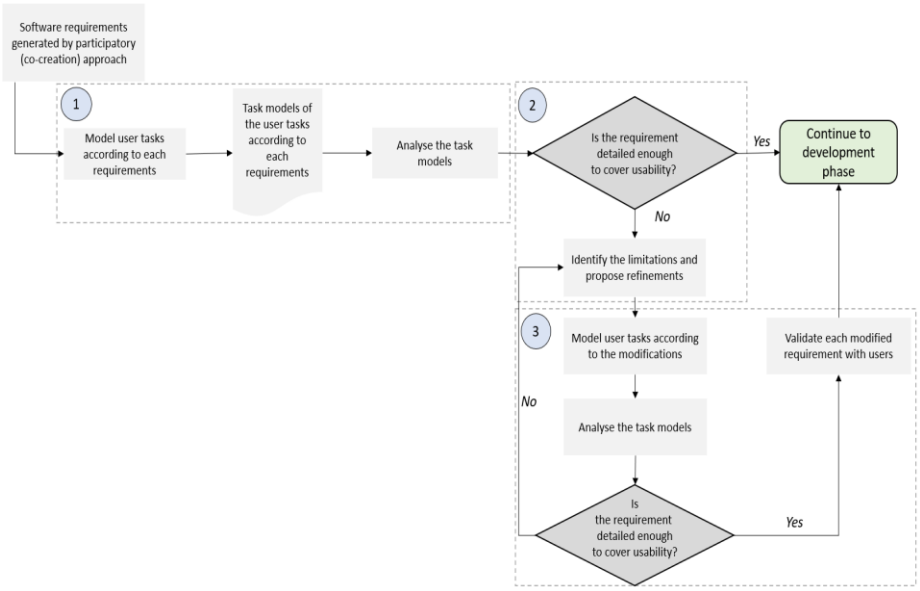

Fig. 1. Three-step task model-based approach for the validation and refinement of requirements gathered from co-creation

The three-step approach just discussed was instantiated by considering the requirements gathered during the case study considered in this paper, the details of which are presented in the following Section.

4 Instantiation of the approach: An illustrative case study

In this section, we instantiate the proposed approach by generating the task models for two randomly selected requirements from Table 1. The selected requirements are as follows:

- **Requirement 3:** The system should provide simple and well-guided user actions when collecting image samples for face recognition.
- **Requirement 8:** The system shall issue a challenge to verify that the user who is trying to authenticate is the owner of the VC (e.g., OTP via SMS or email).

Both these requirements were subjected to the three-step approach presented earlier and discussed in the relevant subsections.

4.1 Model user tasks according to requirements

We selected the HAMSTERS notation [15], it is tool-supported and embeds the common ground elements required to model user tasks [14]. It enables the modelling of refined user task types, as well as of objects required to perform the tasks. In line with

the approach, the first step involved modelling the user tasks according to each requirement. Fig.2. presents the task model produced using requirement 3. As shown in the Figure, it contains one task, *"Provide image sample"*, which is the main goal for the user.

Fig. 2. Task model describing the user actions to provide image samples (from requirement 3)

Furthermore, the task model for requirement 8 was also specified, which is presented in Fig.3. As shown in the figure, the main user goal involved in this requirement is to *"undergo the verification of the verifiable credential"* and is composed of a sequence (sequence temporal ordering operator under the main goal) of an interactive output task and a subtree below the disable temporal operator. The interactive input tasks describe that the system will inform the user that the user has to undergo a challenge to verify the user's verifiable credentials. The subtree describes that the user will take up a challenge, which will be stopped and disabled when the system either issues that the user is the owner of the verifiable credential or not. In both cases, the system informs the user about the issue.

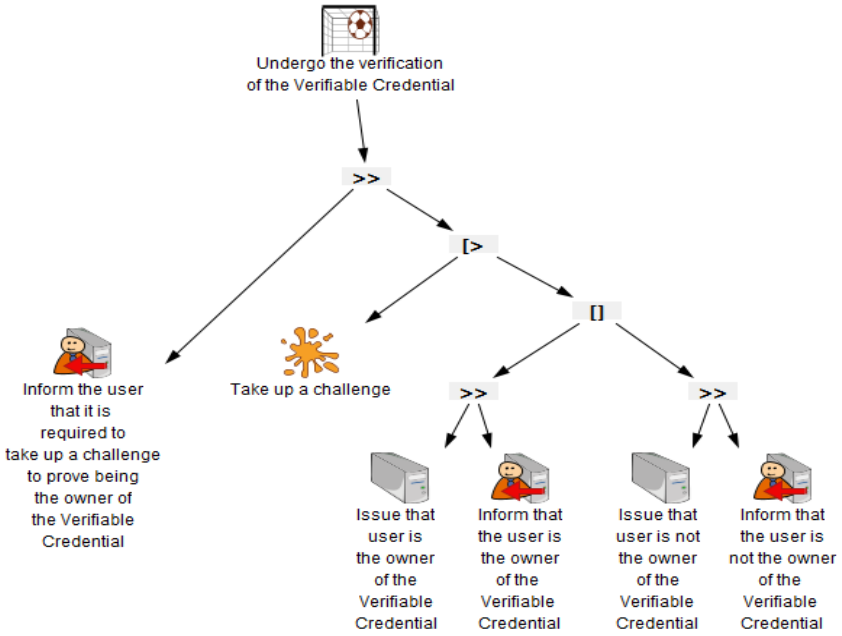

Fig. 3. Task model describing the user actions to undergo the verification of verifiable credentials (from requirement 8)

4.2 Identify the limitations and propose refinements

In line with the second step, the limitations of both requirements were identified. As depicted in the task model specified for requirement 3 (see Fig. 2.), the requirement description contains information only about the main user goal. However, it does not provide information about the possible relevant tasks for reaching the goal (i.e., to provide image samples), which may have a huge impact on usability. The requirement is thus not detailed enough to take into account elements of usability. Several alternatives can be proposed for supporting the users to provide image samples. They can use a local repository, a live camera, or a cloud repository. Considering these aspects, the requirement was modified. Table 2 presents a comparison of requirements before and after being subjected to the approach.

Table 2. Comparison of requirements before and after being subjected to the approach

Requirement 3 (before refinement)	Requirement 3 (after refinement)
The system should provide simple and well-guided user actions when collecting image samples for face recognition.	The system should provide several simple ways and well-guided user actions when collecting image samples for face recognition and in particular, should propose to upload image samples from a local repository, from a live camera, or from a cloud repository. The system should inform the user when enough valid samples have been provided.

Furthermore, the same procedure was performed for requirement 8. From the task model (see Fig. 3), it was identified that the requirement description contains partial information about the user tasks. The system shall issue a challenge to the user but there is no information about the possible relevant tasks to perform the challenge. The tasks may be different depending on the type of challenge and may thus impact usability. This requirement is also not detailed enough to take into account usability. Several alternatives can be proposed for supporting the users to take up a challenge for the verification of the verifiable credentials. The user can enter a one-time password (OTP) received by email or SMS, but the user can also answer a set of personal questions, or scan fingerprints. While considering these aspects, the requirement was modified and presented in Table 3.

Table 3. Comparison of requirements before and after being subjected to the approach

Requirement 8 (before refinement)	Requirement 8 (after refinement)
The system shall issue a challenge to verify that the user who is trying to authenticate is the owner of the VC (e.g., OTP via SMS or email)	The system shall issue a challenge to verify that the user who is trying to authenticate is the owner of the VC. The challenge can be of the following types: to enter an OTP received via SMS or email, to answer a set of personal questions, or to scan fingerprints

4.3 Model user tasks according to the refined requirements

Finally, in line with the third step, task models for the refined requirements were specified. The Fig. 4 presents the task model specified for the refined requirement 3. When compared to the task model of the initial version of the requirements (in Fig. 2), the main goal has been refined. The user can provide image samples iteratively (abstract iterative task "Provide an image sample"), using different sources of images, until (temporal ordering operator "DISABLE") the system issues that the samples are enough and valid.

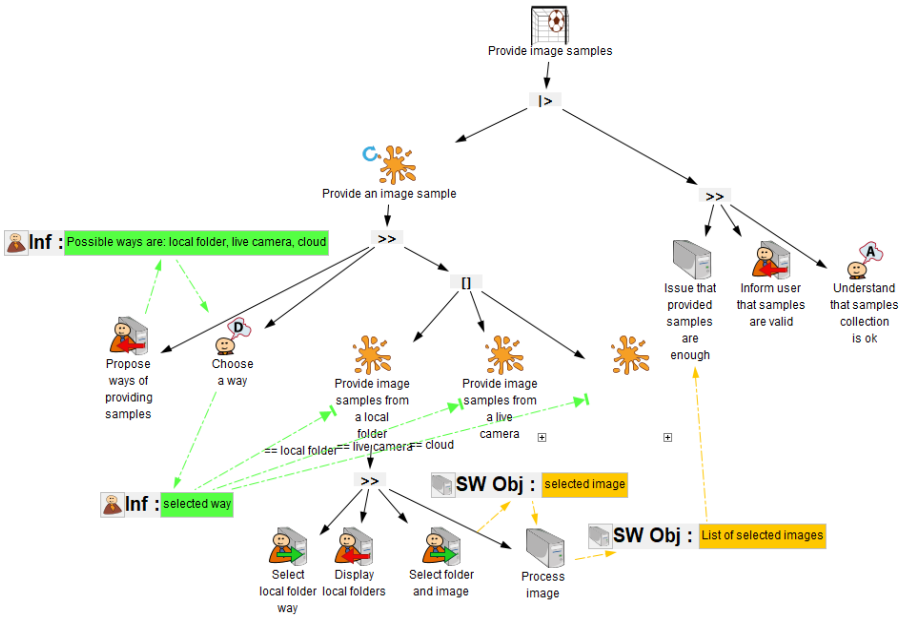

Fig. 4. Task model of the refined requirement 3

Similarly, Fig. 5 presents the task model produced using the modified requirement 8. Compared to the task model of the initial version of the requirements (in Fig. 3), the abstract goal "Take up a challenge" has been refined. The user can either receive an OTP, answer a set of personal questions, or scan fingerprints.

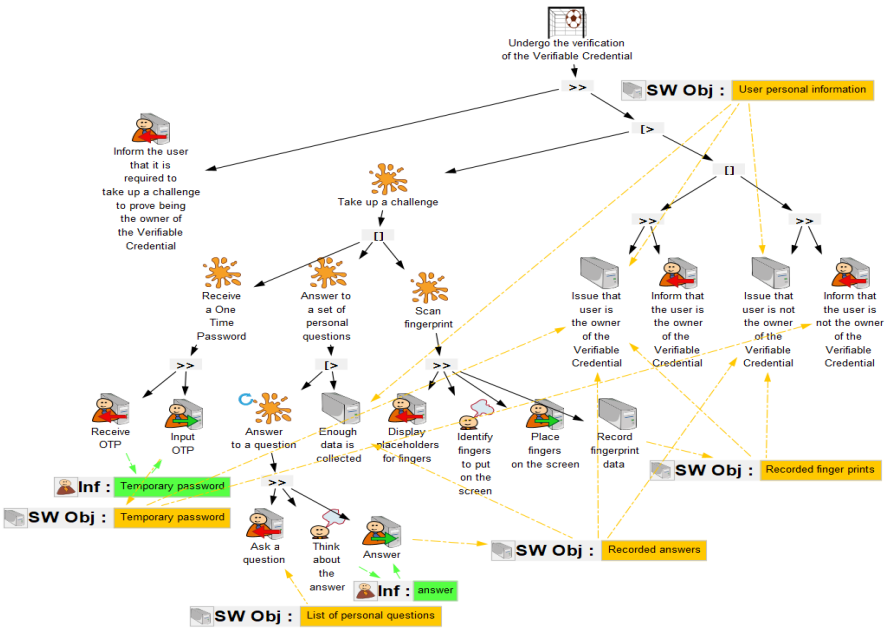

Fig. 5. Task model of the refined requirement 8

The refined requirements along with their associated task models now contain information about the tasks to be performed by the user and their temporal ordering. They can be used to analyse the 'effectiveness' dimension of usability, as well as to support the discussion with the users. As the tasks get more concrete, the users can indicate whether they match their needs or not. Such feedback may lead to refining the requirements again and going again through the loop composed of steps 2 & 3.

5 Related work

Task models enable the identification and description of the required functions for interactive systems [18]. Furthermore, task models enable [2]: to create an understanding of the user requirements, illustrate and validate them, as well as to communicate these requirements with the different stakeholders. The existing literature recommends the usage of task models but does not provide explicit methods or processes for requirements engineering. Garrido et al. [7] proposed task diagrams (in the COMO-UML format) to specify collaboration requirements. The produced models are then used in a model-driven approach to contribute to the automatic generation of a groupware application that matches collaborative user tasks. The modeling technique is closer to workflows and different from the hierarchical point of view available in task models. It thus only captures the computer-supported collaborative tasks. Task models have also been used to automatically generate prototypes [5,17] that users can interact with. The provided feedback then helps to elicit the requirements. This type of approach does not directly use task models to communicate with the stakeholders and is constrained by

the software platform used to generate prototypes. It frames the point of view of the user on the final user interface and may not apply when designing user interactions with new technologies.

Alternative solutions to task models are storyboards [9], business process models [20], probing techniques [8], and the use of semi-formal scenarios, including Behaviour Driven Development (BDD) [1]. These techniques are complementary to the use of task models. Storyboards [9] aim to support multidisciplinary teams in specifying functional and non-functional requirements. Their main advantage is that they are understandable by members having different backgrounds. This technique mainly focuses on building scenarios. Scenarios are complementary to task models as they contain specific sequences of user tasks, whereas task models provide a systematic and abstract view of all the possible user tasks [14]. Business process models [20] aim to describe the possible workflows between users and interactive systems (i.e. flow of events or scenarios). This technique removes the ambiguities of textual descriptions that are scenarios because it is based on a specified notation. It is also complementary to the use of task models because task models provide a systematic and abstract view of all the possible temporal ordering of user tasks [14] whereas workflows describe sets of sequences. Probing techniques [8] aim to understand how users may engage with a novel device, by observing them interact with a prototype, and by discussing with them their thoughts when interacting with the prototype. This technique is also complementary to the use of task models, which can be used to record the tasks the users engaged with the prototype. BDD [1] is a software development approach that uses executable user stories to capture requirements. This technique removes the ambiguities of textual descriptions that are scenarios because it is based on a specified language. It is also complementary to the use of task models because task models provide a systematic and abstract view of all the possible temporal ordering of user tasks [14] whereas executable user stories describe sets of sequences.

6 Discussion

Task models are an accurate mean to represent the outcomes of task analysis and they consist of a graphical representation of the work the users perform with an interactive application or system. Task models enable ensuring the effectiveness of an interactive system, i.e., to guarantee that users can perform their work and reach their goals. Complete and unambiguous description of the users' tasks is a cornerstone of user-centred design approaches as they provide a unique way of describing precisely and entirely users' actions that have to be performed for them to reach their goals [15].

Task identification and description are part of the UCD process. Standard ISO 9241-210 on "Human-centred design for interactive systems" [10] indicates that the specification of the context of use requires identification of user goals, user tasks, and subtasks, but also that the production of design solutions requires allocating tasks and subtasks to the user and to the system. Moreover, it indicates that the evaluation of the design requires verifying the accuracy and completeness with which the user achieved their tasks. This is why almost all UCD techniques require identifying and describing

user tasks. Thus, task models have been proven useful for making a predictive assessment of user performance when interacting with the system [11], and in particular for assessing task complexity and workload (motor, cognitive, perceptive) [16]. Task models have also been proven to increase the coverage of usability issues when performing the heuristic evaluation of the usability of interactive applications [3].

In this paper, we presented how task models can be used by designers and developers for the refinement and validation of requirements. The next step of the work is to investigate if the task models can be used as support for discussions with the end users during the co-creation phases. For that purpose, we need to investigate to what extent the task modelling notation is usable for end users, and if a short tutorial or training could help to use them during co-creation sessions. Then, we also need to investigate alternative representations based on task models or produced from task models.

7 Conclusion

Task models have been proven to be useful for supporting the user-centred design of interactive systems in many ways, from the identification of the required system functionalities to the preparation of usability evaluations. This paper argues for the need for a systematic approach to task models-based refinement and validation of user requirements. Moreover, the paper proposed a three-step approach that integrates with user-centred design practices and in particular with participatory design sessions. The results from the application of the approach to the case study discussed in the paper enabled refinements to the requirements generated using the participatory mechanism. The next steps are to apply several user feedback loops of the process to determine the user-perceived benefits of this approach.

Acknowledgment

This work is supported by the European Union's Horizon 2020 research and innovation program under grant agreement No 101004459, project IMPULSE (Identity Management in Public Services).

References

1. Alhaj, M., Arbez, G., Peyton, L. Using behaviour-driven development with hardware-software co-design for autonomous load management. In: 2017 8th International Conference on Information and Communication Systems. ICICS, IEEE, pp. 46–5 (2017).
2. Balbo, S., Ozkan, N., Paris, C.: Choosing the Right Task-modeling Notation: A Taxonomy. In: Diaper, D., Stanton, N.A. (eds.) The Handbook of Task Analysis for Human-Computer Interaction, pp. 445–466 (2004).
3. Cockton, G., and Woolrych, A. Understanding inspection methods: Lessons from an assessment of heuristic evaluation. People and Computers, Springer, pp. 171–192 (2001).
4. Coventry. L. "Usable Biometrics." In Security and Usability, Lorrie Cranor and Simson Garfinkel (Eds.). O'Reilly, Cambridge, MA. 43 (2005).
5. Fischer, H., Rose, M., Yigitbas, E.: Towards a task driven approach enabling continuous user requirements engineering. In: 2nd Workshop on Continuous Requirements Engineering

(CRE) (REFSQ-JP 2016), Gothenburg, Sweden, March 14, 2016.CEUR-WS.org, vol. 1564 (2016).
6. Garfinkel, S. and Lipford, H. R. “Usable Security: History, Themes, and Challenges.” San Rafael, California, Morgan and Clay Publishers (2014).
7. Garrido, J.L., Noguera, M., González, M., Hurtado, M. V., Rodríguez, M. L. Definition and use of Computation Independent Models in an MDA-based groupware development process, Science of Computer Programming, Vol. 66, Issue 1, 25-43 (2007).
8. Gough, P., Kocaballi, A. B., Naqshbandi, K. Z., Cochrane, K., Mah, K., Pillai, G., Yorulmaz, Y., Deny, A. K., Ahmadpour, N. Co-designing a Technology Probe with Experienced Designers. In Proceedings of the 33rd Australian Conference on Human-Computer Interaction (OzCHI '21). Association for Computing Machinery, New York, NY, USA, 1–13 (2022). https://doi.org/10.1145/3520495.3520513
9. Haesen, M., Luyten, K., Coninx, K. Get Your Requirements Straight: Storyboarding Revisited. In: Gross, T., et al. Human-Computer Interaction – INTERACT 2009. INTERACT 2009. Lecture Notes in Computer Science, vol 5727. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03658-3_59
10. ISO 9241-210:2019(en), Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems, International Standard Organization (2019).
11. John, B. Kieras, D. E. The GOMS family of user interface analysis techniques: comparison and contrast. ACM Trans. Comput.-Hum. Interact. 3, 4 (December 1996), 320-351 (1996).
12. Kujala, S. User involvement: A review of the benefits and challenges. *Behaviour & Information Technology*, *22*(1), 1–16 (2003). https://doi.org/10.1080/01449290301782
13. Kujala, S. Effective user involvement in product development by improving the analysis of user needs. *Behaviour & Information Technology*, *27*(6), 457–473 (2008). https://doi.org/10.1080/01449290601111051
14. Martinie, C., Palanque, P., Barboni, E. Principles of Task Analysis and Modeling: Understanding Activity, Modeling Tasks, and Analyzing Models. Handbook of Human Computer Interaction. Springer, Cham (2022).
15. Martinie, C., Palanque, P., Bouzekri, E., Cockburn, A., Canny, A., Barboni, E. Analysing and demonstrating tool-supported customizable task notations. PACM on Hum. Comput. Interact. 3(EICS), 26 (2019).
16. O’Donnell, R. D., Eggemeier, F. T. Workload Assessment Methodology; In Handbook of Perception and Human Performance (Vol. II Cognitive Processes and Performance, pp. 42-41 - 42-49). Wiley and Sons, 1986.
17. Reichart, D., Forbrig, P., Dittmar, A. Task models as basis for requirements engineering and software execution. In Proc. of the 3rd annual conference on Task models and diagrams (TAMODIA '04). ACM, New York, NY, USA, 51–58 (2004).
18. Paternò, F. Task models in interactive software systems, Handbook of Software Engineering and Knowledge Engineering, Vol 1, Publisher: World Scientific, pp. 1-19 (2002).
19. Sanders, L. An evolving map of design practice and design research. *Human Factors*, 7 (2008).
20. Sindhgatta, R., Thonse, S. Functional and Non-functional Requirements Specification for Enterprise Applications. In: Bomarius, F., Komi-Sirviö, S. (eds) Product Focused Software Process Improvement. PROFES 2005. Lecture Notes in Computer Science, vol 3547 (2005). Springer, Berlin, Heidelberg. https://doi.org/10.1007/11497455_16
21. Smart, J.F.: BDD in Action: Behavior-Driven Development for the Whole Software Lifecycle. Manning Publications, Shelter Island (2014).