A Generic Visualization Approach Supporting Task-Based Evaluation of Usability and User Experience

Regina Bernhaupt, Célia Martinie, Philippe Palanque, Günter Wallner

A Generic Visualization Approach Supporting Task-Based Evaluation of Usability and User Experience

Regina Bernhaupt[1], Célia Martinie[2(✉)], Philippe Palanque[2(✉)], and Günter Wallner[1]

[1] Department of Industrial Design, Eindhoven University of Technology, Eindhoven, The Netherlands
{r.benrhaupt,g.wallner}@tue.nl
[2] ICS-IRIT, Université Paul Sabatier Toulouse III, Toulouse, France
{martinie,palanque}@irit.fr

Abstract. Analyzing evaluation results of usability and user experience studies has its limitations when it comes to personalized user interfaces, highly complex and connected systems, or internationally used services involving millions of users. To support the analysis of the evaluation results of usability and user experience, a task-based evaluation approach is proposed. This approach uses multiple visualization views to support the analysis of evaluation results. The visualization offers different temporal views ranging from individual to more cumulative data views in order to combine results from evaluations and task models. The applicability of this approach is presented on a simple but demonstrative case study from television and entertainment.

Keywords: Visualization · User experience · Usability · Task modelling · Evaluation · User studies

1 Introduction

Traditional approaches, processes, and methods of user-centered design and development have recently been challenged by a variety of technological shifts and changes in users' behaviors and expectations. The introduction and usage of artificial intelligence (AI) technologies, the challenges of the internet of things with its plethora of connected devices, and the phenomena of internet-based services and companies with millions of users push these methods, processes, and approaches to their boundaries. Especially when it comes to usability and user experience (UX) evaluation.

While evaluation methods have partly evolved and changed, some main challenges remain. It is unclear how to evaluate and deal with systems where every individual user will get a personalized recommendation that evolves each time

https://doi.org/10.1007/978-3-030-64266-2_2

another user is interacting with the system (cf. [27]). Due to such personalized recommendations every experience for one user will be different, leading to incomparable different user experiences for a user and between users in terms of evaluation findings.

The challenge when performing an evaluation related to connected devices, like in a household today, is the multitude and plethora of available devices and services that people use (alongside the product that might be available) making it difficult to understand the technological context of usage that can affect the outcomes of the evaluation. When looking at web services and companies like Facebook, the number of evaluation participants must be extremely high to represent characteristics of the world-wide user base. This brings the challenge of comparing millions of data points and the key question is how to make such data points comparable, when the same or similar user activities are performed.

Finally, research in human-computer interaction (and other domains such as software engineering) faces the same challenges when it comes to research evaluation methods. It becomes more and more difficult to make sense of gathered data, as the contextual information when it was gathered is merely not recorded/represented. And it is challenging to compare evaluation data that was gathered over time or in different studies. When it comes to empirical and artefact research contributions, evaluation methods [15] have to be adapted.

A common problem of these challenges is the non-comparability of the results for different user experiences when performing the same activity (within user) and for different user experiences (between users). Key challenge is the missing ability to understand what people have been doing and which task or tasks people were performing, and how this relates to the evaluation data that was gathered. The proposed approach to address these challenges is to design visualization techniques for evaluation data in order to support evaluators' activities. More precisely, these techniques enable to (a) compare unique personal user experiences for a user or between users, (b) understand the impact of contextual influences on the evaluation data, (c) compare evaluation data over longer periods of time or time spans – or over different user studies, and (d) compare multiple software qualities like usability and UX and their effects on one another.

More precisely, we developed a novel visualization approach that supports evaluators (when conducting evaluations) that use as a basis user activities that can at least be loosely described. This includes usability and UX studies or tests and experiments with given user activities the participants have to perform. The visualization approach is combined with a task-based evaluation approach using task models as key component for extracting scenarios for experiments and for presenting evaluation results. As task models are able to cope with complex work, the approach is intended for large-scale and complex interactive systems such as command and control systems.

2 Related Work

Evaluation methods and research evaluation methods in human computer interaction focusing on usability and user experience can be classified in (a) methods

involving users such as observations, field studies, surveys, usability studies, interviews or focus groups [15], (b) methods performed by experts, and (c) automated methods [2]. Today evaluations, like user studies, are performed either in person (in the laboratory or in the field) or remotely (asynchronous or synchronous; independently by the users including users self-observations or guided by evaluators;. . .) [14].

When it comes to research focusing on UX and trying to understand UX, a broad range of evaluation methods has been developed, adapted, and adopted [29] to address the temporal aspect of UX, typically classified as before, during, and after interaction with a product, as well as the overall episodic UX.

Performing a usability or UX evaluation involving users requires activities beyond conducting the study per se [15]. Evaluation starts with the identification of evaluation goals and the selection (or even adaptation or development) of a set of evaluation methods. This is followed by the preparation of the necessary material, like usability study protocols or guidelines that will be used to perform the study, recruiting and selection of participants, and pre-testing of the evaluation methods. After conducting the study, various other tasks such as data collection and data pre-processing and the analysis of data are performed. Key to every evaluation is the interpretation of the data and what the evaluation results mean for the iterative (re-) design of the system and the required changes to the software or product. As Bernhaupt et al. [2] indicated, this step typically fails in many projects.

Limitations of current methods in usability and UX evaluation and research are [14,15]:

- missing validation of results (typically evaluation studies are not re-done in research nor confirmed when applied in industry)
- sharing of evaluation results between researchers or beyond
- analysis phase of results, not done in relation with activities or tasks, influences of context (temporal, physical, social, technological, and societal), ability to analyze cross-over studies
- support for long-term studies or comparison of studies.

2.1 Task Driven User Interface Design and Evaluation

User tasks are analyzed since the early phases of the design of interactive systems and their analysis provides support to various stages of User-Centered Design and of interactive system development, from first prototype assessment to interactive system deployment. Beyond the identification and description of required functions for an interactive system [10,23], this includes the identification and description of knowledge required to perform a task [6,13,18,26], as well as different roles users can have in groupware systems [24,31], and how users collaborate [24,31]. Central for understanding evaluation results is the understanding of the application domain [22], the ability to produce scenarios for user evaluation [33], and the identification and generation of models describing user errors [8].

PRENTAM is a process model describing how to integrate usability and UX evaluation results into task models [4]. Key for such an approach is the integration of functionalities like data import and export alongside the ability to report evaluation data alongside the task models [3].

2.2 Visualization as a Tool for Evaluation

Especially for the evaluation of complex environments such as games [32], visualizations have become a vital aspect for the interpretation of user behaviour and subsequent reporting thereof. For instance, heat maps and gaze plots have also become popular in gaze-based usability evaluation [9] of not only games but also websites and applications. These visualizations are, however, usually tailored towards specific analysis goals and the virtual environment itself can be used as reference for the visualization. Many evaluations, however, take a more abstract, task-based, perspective where usability and UX measures are evaluated over time. In this regard, several existing tools aim to support the design and implementation of experiments.

The *Touchstone2* tool aims to facilitate the exploration of alternative sample sizes and counterbalancing strategies for experiment design. It provides support to weigh the cost of additional participants against the benefit of detecting smaller effects [7]. It is hence focused on the statistical settings of experiments. It provides support for visualizing the possible combinations of devices and interaction techniques but does not provide support for visualizing the results of the experiment itself. *NexP* [20] aims to facilitate the preparation and implementation of experimentation for persons with limited knowledge in experimental design. The results of the experiment are collected by a running platform and are saved in csv files. However, it does not provide support for visualizing the results. These tools were demonstrated with examples of measuring temporal performance and motor accuracy, they thus focus on usability measures and do not provide explicit support for dealing with UX measures.

What is currently missing in evaluation as well as task modeling tools is the ability to combine both worlds. Such a combination would allow to present evaluation data in the task, to provide the possibility to visualize data, and to support the analysis of evaluation results with values, depictions, or visualizations beyond basic statistical values (such as average time spent on a task [3]).

3 Task-Based Evaluation of Interactive Systems

3.1 Problem Description and Background

Task-based evaluation of usability and UX as key software qualities involves several components. First, the user activity has to be analyzed, described in terms of goals to reach, and tasks to be performed in order to reach those goals. Often, these tasks are described, in its simplest form as free text or using Excel

files ending up in long lists of activities. As pointed out by Peter Johnson in his keynote at TAMODIA 2004 [12] starting with: *"What is the point of modelling anything. . . "*, for complex systems, it is advisable to use dedicated software tools for task modelling in order to deal with large and complex task models. The detail of the task model typically is heavily influenced by the expressive power of the tool.

Usability and UX as key software qualities benefit differently from a task-based evaluation approach. Usability evaluation is still in the majority of cases performed as a laboratory study, often only once within the overall user-centered development process. One of the reasons might be that usability, as a quality, is seen as rather static (after a training/learning period) and not easily influenced by small changes in the work environment. On the contrary, user experience is diverse along several dimensions:

1. temporal aspect: UX is changing over time, the user experience before, during, and after the interaction with a product and the cumulative user experience need different measures over time. This results in the need for a tool that allows to show these different measures in one place and to enable the understanding of influencing factors like contextual changes during the analysis.
2. multi-dimensional: UX per se is multi-faceted and typically a construct of different dimensions [25] that are evaluated, thus the (sub-) dimensions of UX have to be visualized and made accessible during the analysis phase in 'one image'.
3. inter-dependability: The interplay between usability and UX (and other software qualities) is dynamic and complex. To enable improved evaluation results, the comparison of usability and UX must be supported by such a system, as well as the investigation how different software qualities impact each other.

In the following we use UX as a more general term for all possible software qualities that can be evaluated, including usability.

To enable task-based evaluation of usability and UX a system must support the above mentioned aspects. To accomplish this it is necessary to have task representations and task models that support user studies, providing the ability not only to describe tasks, but also to define scenarios. Scenarios represent activities that users perform when conducting standard usability evaluation studies. Key for the success of such a system is the degree to which the task model notation and the support tool are able to represent and store evaluation results from user activities.

The central problem is how to help evaluators make the most out of evaluation results. Our proposed solution is the combination of task models that incorporate scenarios to represent user activities performed with a system and to visualize the usability and UX evaluation results accordingly. The goal is to solve and support evaluators when they face the challenge of comparability of personalized user interfaces, when the technological usage context is difficult to describe, as well as when large amounts of data points have to be compared.

3.2 Tasks Representation and Task Models to Support User Studies

Task modelling and task models support task analysis activities and are a means to represent the outcomes of the task analysis. Task models consist of an abstract description of user activities structured in terms of goals, sub-goals, and actions [22]. The models that result from task analysis will differ according to the features of the selected modelling language or notation. These modelling differences are likely to illuminate (or suppress) different aspects of the interaction. It is, therefore, important to choose the most suitable task modelling technique, i.e. the notation with the most suitable expressiveness, which highlights the aspects that are relevant to the goals of the analysis.

As highlighted in Bernhaupt et al. [4], task analysis and modelling provides support to check for completeness of the sets of tasks used for user studies. Moreover, if the task modelling notation is exhaustive enough, it provides support to identify the types of data that has to be collected (e.g., motor/reaction time to be monitored, input data to be recorded in the system,. . .). Bernhaupt et al. [4] also show that task models can be used to record the results of user evaluation to inform re-design.

3.3 Scenarios

As defined by Rosson and Carroll [28], scenarios *"consist of a setting, or situation state, one or more actors with personal motivations, knowledge, and capabilities, and various tools and objects that the actors encounter and manipulate. The scenario describes a sequence of actions and events that lead to an outcome. These actions and events are related in a usage context that includes the goals, plans, and reactions of the people taking part in the episode"*. The main differences between scenarios and task models are that scenarios contain concrete data (e.g., user name and characteristics) whereas task models are abstract. A scenario is a sequence of actions (like a story line) whereas task models are hierarchical (from more abstract to more concrete) and can describe several possible actions for the user in different temporal orders. At last, scenarios may be borderline (e.g., represent cases at the limit) whereas task models are mainline as they represent the standard, usual, and prescribed activities.

Although they differ, task models and scenarios are complementary. Scenarios can be produced by the execution of task models [21] and they are also a means to verify task models. Most of the task modelling tools provide a simulator which allows to analyze the possible sequences of actions for a task model. These possible sequences can be recorded as scenarios. Scenarios produced from task models can be used as input for driving the execution of the system [21] and the execution of training sessions [19]. They can also be used as input to prepare empirical studies [33], as it is the case in this paper. Furthermore, when tool-supported, task models can also provide support to produce scenarios that will be used for conducting the study and in which the various types of measures will be recorded (according to the type of task) [3].

3.4 Importance of the Expressiveness of Modeling Notation and Tool

The systematic identification of possible user actions and the analysis of effects of the study setup on user tasks requires to exhaustively and precisely describe:

1. user actions and their type (e.g., perceptive, motor, cognitive) in order to identify where a usability or UX issue can come from,
2. their temporal ordering and/or temporal constraints to compare the impact of the possible temporal ordering of user actions on usability and UX,
3. the information, knowledge and devices being manipulated during these actions in order to identify the information, knowledge, and devices required to perform the evaluation, as well as to compare the impact of the possible alternatives on usability and on UX.

4 The Visualization Approach

To enable the usage of task models and support a systemic view on the design and development of interactive systems, we used the previously defined PRENTAM model to integrate usability study results into task models [4]. The HAMSTERS-XL notation [17] embeds all these elements that are important to support user evaluation. We complement the tool HAMSTERS-XLE with a novel visualization approach.

The visualization was built with the overarching goal to support evaluation by visually highlighting potential issues and to inform redesign of the evaluated system. To support this, the visualization addresses the following three sub-goals:

1. provide an overview of temporal aspects of UX and usability for scenario-based evaluations
2. allow to compare different conditions (e.g., design variants) and scenarios against each other
3. provide a summary of UX/usability measures while also allowing to view details with regard to individual users and the underlying task model

To help assist with evaluating UX and usability based on task models of user activity and the above mentioned goals we devised an interactive visualization consisting of three different views. The main view provides an aggregated overview of all the collected UX data across multiple scenarios and conditions over time (Goal 1 and 2). This view also acts as a hub to add views of more detailed data for selected scenarios and conditions (thus following the established visual information seeking mantra *Overview first, zoom and filter, then details-on-demand* postulated by Shneiderman [30]). These views – supporting Goal 3 – provide a) an aggregated summary of user-selected UX measures on a per-activity basis and b) an overview of individual activity-execution times and UX measures for each participant, and c) also allow to visually compare results across conditions and scenarios. All views offer tooltips showing detailed data of the color-coded measures. In the following, these views will be explained in detail. Views can be combined freely to match analysis goals and preferences of the user.

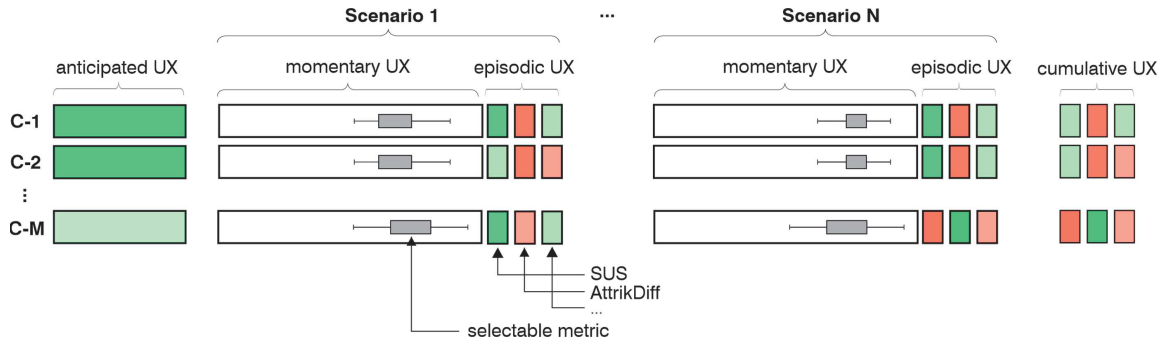

Fig. 1. Overview: Columns represent different scenarios while rows indicate different conditions for the individual scenarios. The visualization follows the *UX over time* concept, covering anticipated, momentary, episodic, and cumulative UX. Colors reflect the values of the UX values (in this case a red-to-green gradient was chosen). (Color figure online)

4.1 Overview

The overview, as illustrated in Fig. 1, follows the *UX over Time* [16,29] scheme, showing from left to right, the anticipated UX, momentary and episodic UX for different scenarios, and finally cumulative (or sometimes also referred to as remembered) UX across all scenarios.

Each UX measure is represented using a block. For instance, anticipated UX could be measured using five seconds UX impressions in interviews, while standard questionnaires like the *SUS* [5] or *AttrakDiff* [11] can be used to assess episodic or cumulative UX. Momentary UX could be assessed through metrics such as time taken, ease of use, or aesthetic appeal. Except in case of momentary UX, these blocks are also color-coded to ease comparison of the measures. The color scheme can be defined for each UX measure separately, for example, a unipolar color gradient for a unipolar scale such as the *SUS* can be used or a bipolar color scheme for diverging measures such as the *AttrakDiff*. The momentary UX block contains a box plot to offer more details about the distribution of a selected metric at a glance.

In terms of overall layout, the overview follows a grid-based approach with rows representing different conditions and columns representing the scenarios. This can either be different scenarios or the same scenario at different points in time. In the latter case, these are sorted based on recency with the most recent one depicted in the right-most column to reflect the progression over time.

4.2 Aggregated User Activity View

This view (see Fig. 2) depicts the activities in a simplified tree structure, with the different activities represented as color-coded blocks, where the color indicates the mean of a momentary UX measure (e.g., average time or average difficulty rating). Multiple trees can be viewed side-by-side to allow for comparisons between different conditions or between scenarios (useful, for instance, when the

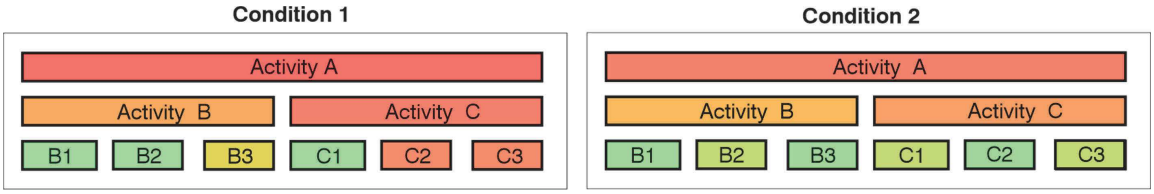

Fig. 2. Aggregated User Activity View: This view provides on overview of UX measures (here time needed: min max) for the different activities performed within a scenario. Multiple trees can be visualized side-by-side to ease comparisons across different conditions. (Color figure online)

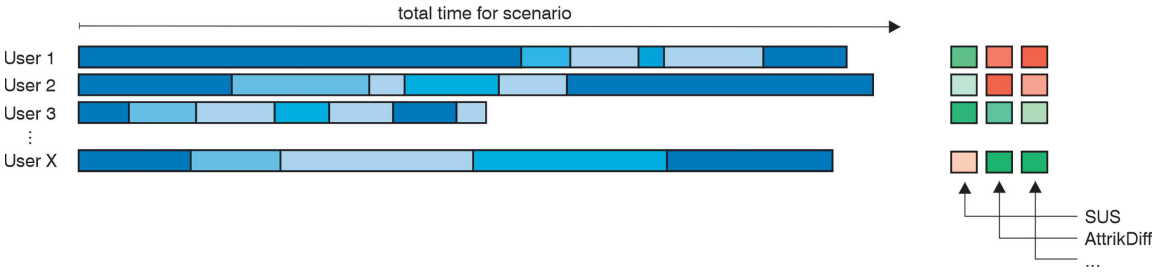

Fig. 3. Individual User View: Visualization of activity times together with UX measures such as *SUS* or *AttrakDiff* on a per-user basis for a selected scenario and condition. Colors indicate the number of tasks within an activity (in this particular case: 1 4). (Color figure online)

same scenario has been assessed at multiple points in time). Figure 2 illustrates this concept by comparing two different conditions. Assuming the colors encode time, with more reddish values indicating higher required times, one can see that Activity A on overall required more time in Condition 1 than in Condition 2, which is mainly caused by the sub-activities C_2 and C_3 within Activity C.

4.3 Individual User View

In order to allow the evaluator to inspect individual participant data in more detail, this view summarizes activity execution order on a per-user basis (cf. Fig. 3). Each row represents a single user and each block represents a single activity with the color indicating the number of tasks within it. The width of each block indicates the time the user has needed to perform the respective activity. The color-coded blocks to the right indicate the different episodic UX values as provided by the user. For example, in the example in Fig. 3 the first user took a comparatively very long time for the first activity (dark blue) compared to the others which may explain the lower UX ratings (red blocks). The last user – User X, to give another example, required a comparatively long time for the third activity which only includes one task, indicating potential struggles. As is the case with the *Aggregated User Activity View*, the data of multiple conditions and/or scenarios can be represented side-by-side to help with comparisons.

5 Case Study

As case study, we opted to use *Netflix* and compare different scenarios on two different conditions (PC and Smartphone). In the following we will first present the task models of the case study, explain the data collection procedure, and finally present how the collected data can be analyzed with the visualization outlined above.

5.1 Tasks Models and Relation to Task-Based Evaluation

As explained earlier, we used the HAMSTERS-XLE environment [17] to model the main tasks that users have to perform in order to consume content with *Netflix*. Figure 4 shows the main user tasks to perform in order to reach the goal "Consume content on Netflix" (located at the top of the task model). As the name indicates, it corresponds to a broad and loosely defined activity the user wants to perform. That task model must be read as follows: The user first opens Netflix and log-in into the service. Opening Netflix and log-in are interactive tasks, as indicated by the symbol for an interactive input/output tasks on the left-hand side of Fig. 4. Then, the main sub-goal of the user is to consume movies or content, this can be repetitive (indicated by a blue repetition arrow on the task symbol), until the user decides to quit Netflix (represented with the temporal ordering operator DISABLE, labelled "[>" upon the interactive input task "Leave Netflix").

The "Consume content" sub-goal is decomposed into several tasks described at the bottom of the task model (read from left to right). First, the Netflix application filters content and then processes the profile of the user. This is represented as tasks from the system labelled "Filter content" and "Process profile". Then it displays the welcome page (interactive output task named "Display a welcome (landing) page"). Then, the user can choose (represented by the temporal ordering operator OR labelled "[]") to select either content from the current Top 10 (abstract task named "Select a content from the Top 10") or to search for content (abstract task named "Search for content"). The detailed actions for the sub-goals "Select a content from the Top 10" and "Search for content" are presented in Fig. 5.

What is important is to understand the expressiveness of the task model notation, as this lays the basis to understand and interpret related evaluation results. Activities such as the sub-goal "Select a content from the Top 10" are represented as an abstract task. Such an abstract task allows, during the interpretation of the evaluation tasks, to look at summative values like task times or overall ratings on user experience like aesthetics values, to understand how users in general perform selection.

These abstract tasks can then be refined (if needed) into concrete tasks. Indeed, Fig. 5 (left-hand side) shows that "Select a content from Top 10" is decomposed into three actions that happen sequentially (represented by the temporal ordering operator SEQUENCE, labelled ">>"): "go to the Top 10" (interactive input/output task), "select a number" (interactive input task) and "go

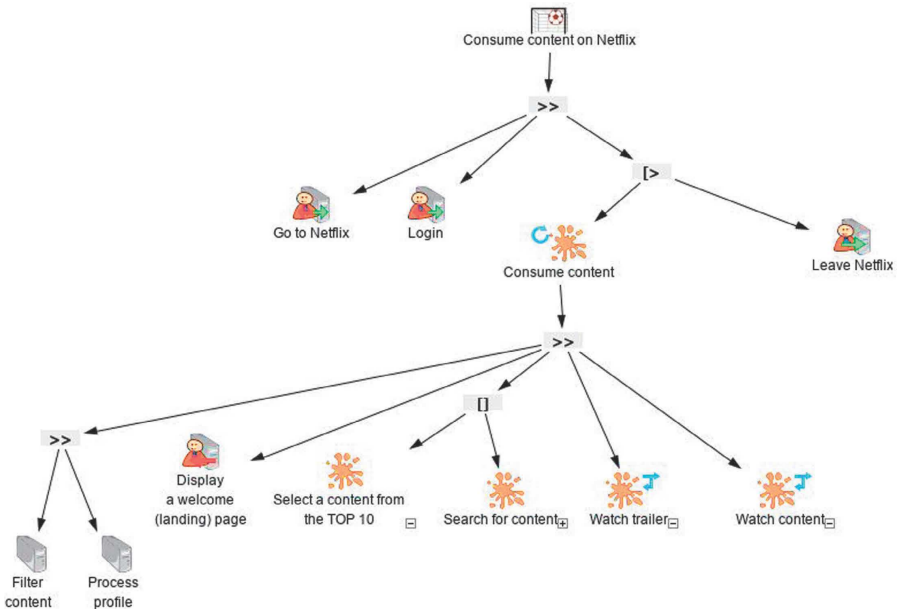

Fig. 4. Overview of the task model describing user actions to consume content. (Color figure online)

to the detail page" (interactive input/output task). The interactive input task "Select a number" modifies the value of the software object labelled "selected content" (represented by an orange rectangle with an incoming arrow from the "Select a number" interactive input task). This representation allows, for example, to investigate if any latency of the system in value modification or representation has an impact on the evaluation results.

The sub-goal "Search for content" (represented with an abstract task) is also decomposed into a sequence (">>") of two abstract tasks: "Use search function" and "Select target content". The "Use search function" abstract task is refined into the following sequence of tasks: "Locate search function" perceptive task, "Enter in search field" interactive input/output task, "Type the first characters" interactive input/output task (which modifies the content of the software object "characters"), and "Search content which name starts with input characters" system task (which uses the value of the software object "characters" and modifies the value of the software object "List of movies starting with characters"). This task is performed by Netflix and returns a list of movies matching the filter defined by the user while typing characters.

Computing the number of interactive tasks supports the evaluator to understand the quantity of interaction that the users will have to perform with the system (Netflix) before being able to watch the desired content.

In the task models of Fig. 4 and Fig. 5 we have not represented (due to space constraints) the cognitive activities of users. However, in this case, looking at cognitive activities can help the evaluator understand how long people on average need to decide, but more importantly how quick (minimum time) or how long (maximum time) users take to make up their mind. Such details can be important when redesigning the system to enhance user experience and, in this case, the

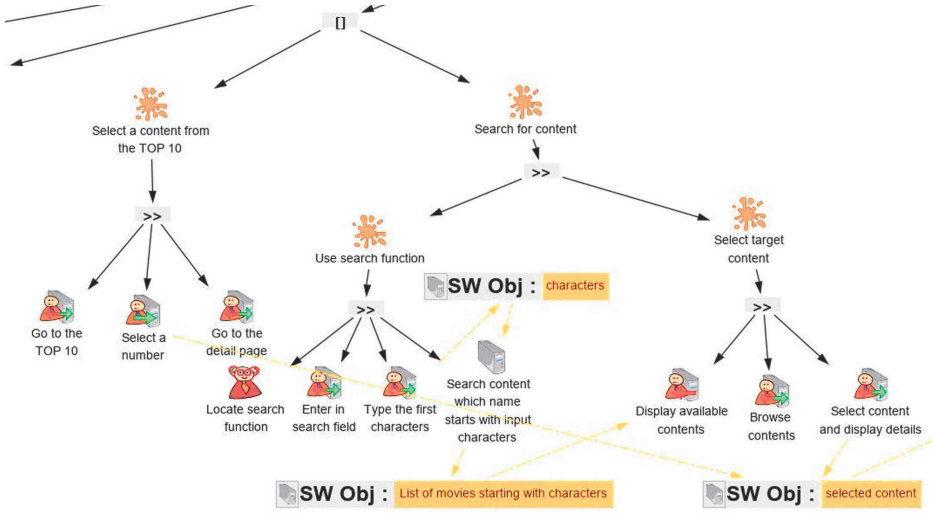

Fig. 5. Focus on the sub-goals "Select a content from the Top 10" and "Search for content" of the task model "Consume content on Netflix". (Color figure online)

long term perception of the user to, for example, *"never find the right content for me"*.

The "Select target content" abstract task is refined into the following sequence of tasks: "Display available contents" interactive output task (which uses the value of the software object "List of movies starting with characters"), "Browse contents" interactive input/output task, and "Select content and display details" interactive input/output task (which modifies the value of the software object "selected content").

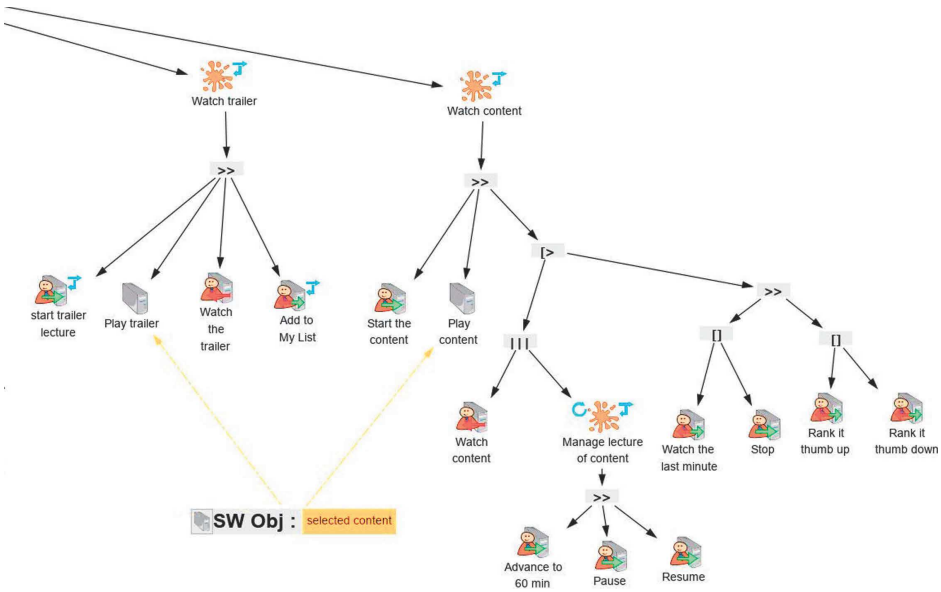

Fig. 6. Focus on the sub-goals "Watch trailer" and "Watch content" of the task model "Consume content on Netflix".

The detailed actions for the sub-goals "Watch trailer" and "Watch content" are presented in Fig. 6. The abstract tasks "Watch trailer" and "Watch content" are optional (represented by the blue arrows symbol in the upper right corner of the task), meaning that the user may or may not perform these tasks.

For the evaluation, basic statistics such as which percentage of users performed which activity can be insightful to decide if further data analysis on usage statistics would be beneficial for understanding the evaluation results. An example could be that a very low number of people are not watching the content once they engage with the trailer, or vice versa.

The abstract task "Watch trailer" is refined into the following sequence of tasks: "Start trailer lecture" interactive input task (which is optional as the system may directly play the trailer), "Play trailer" system task, "Watch the trailer" interactive output task, and "Add to my list" interactive input/output task (which is optional). Here a comparison if asking the user to engage in an activity is beneficial compared to the automation (trailer starts directly) can be insightful for the evaluation. Especially when user interfaces on different devices show different behaviors such comparisons can be helpful to understand the effect of automation on user experience aspects, for instance, users' perception of the ability to control the user interface.

The abstract task "Watch content" is refined into the following sequence of tasks: "Start the content" interactive input task, "Play content" system task and then, the execution of tasks that occur concurrently (represented by the temporal ordering operator CONCURRENCY, labelled "|||"), "Watch content" interactive output task and "Manage lecture of content" abstract task. The "Manage lecture of content" abstract task is optional. It is further broken down into the following sequence of tasks: "Advance to 60 min", "Pause", and "Resume", with all of them being interactive input tasks. The concurrent execution of the "Watch content" and "Manage lecture of content" tasks stops (represented with the temporal ordering operator DISABLE, labelled "[>") when the user performs the "Watch the last minute" interactive input/output task or (represented with the temporal ordering operator OR, labelled "[]") the "Stop" interactive input task. Then (represented by the temporal ordering operator SEQUENCE, labelled ">>"), the user can "Rank it thumb up" (interactive input/output task) or (represented with the temporal ordering operator OR, labelled "[]") "Rank it thumb down" (interactive input/output task). In the evaluation activity of such media browsing activities, the representation of data beyond the visualization might be beneficial, as it can show the relationship between user behaviors, like overshooting the video, repetitive presses when users are impatient with a slowly reacting system, etc.

For evaluation comparison between two different systems, the ability to instantiate different task models is important. The "Consume content on Netflix" task model, for example, is once instantiated for the personal computer and once for the smartphone. This instantiation aims at exhaustively representing which user action can be performed on each specific device. Figure 7 and Fig. 8 present excerpts of the "Consume content on Netflix" task model instantiated

for the task to be performed with a Personal Computer (PC) and with a Smartphone. In this task model, we see that interactive tasks are connected to devices. For example, the interactive input/output task "Go to the Top 10" requires the use of the output device display (represented with a stroke between the task and the blue rectangle labelled "out D: Display") and of the input device mouse (represented with a stroke between the task and the blue rectangle labelled "in D: Mouse"). For the evaluation such indicated connections can allow the investigation of special cases, for example, by taking into account logging of system data like mouse movements and which device is used for which task.

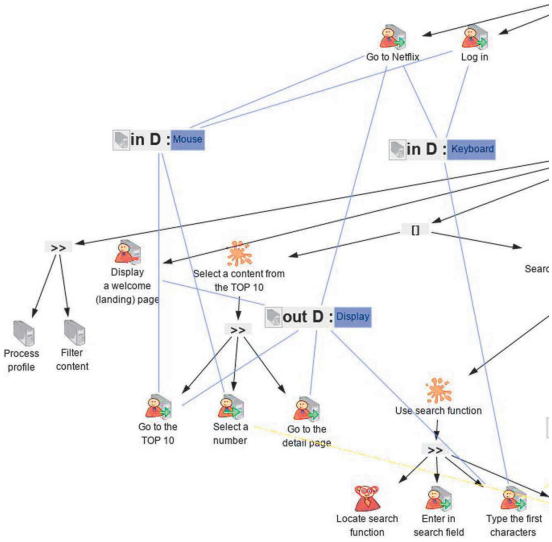

Fig. 7. Extract from the task model "Consume content on Netflix" instantiated for PC devices.

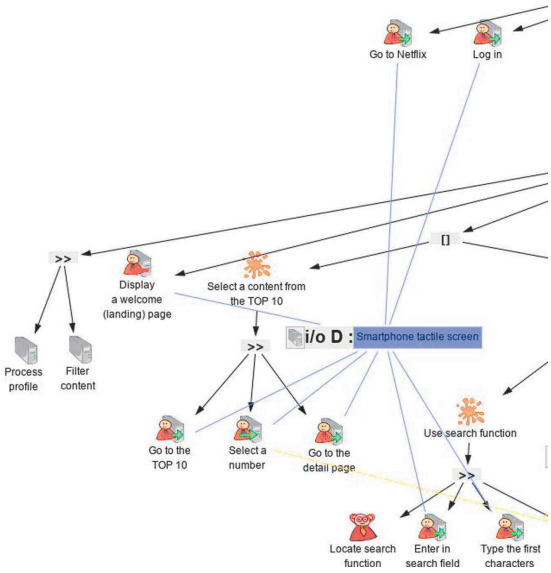

Fig. 8. Extract from the task model "Consume content on Netflix" instantiated for a Smartphone device.

5.2 Scenarios of the Case Study

Once the user tasks are described, we produce the scenarios that are required for the user study. For that purpose, we use the HAMSTERS-XLE simulator to execute the task model and generate several possible scenarios for the study. An example of output of such a simulation is the scenario presented in Fig. 9. This scenario contains a sequence of actions that the user has to perform to watch the 7th item of the Top 10 list of content on Netflix.

For each generated scenario, we identify the actions for which the time should be measured during the study. In Fig. 9, they are represented with the black heptagons labelled "T1", "T2", "T3", and "T4". A scenario that is selected to be performed during the study is modified to remove the description of the system tasks (e.g., system task labelled "3-Process profile" in Fig. 9) from the

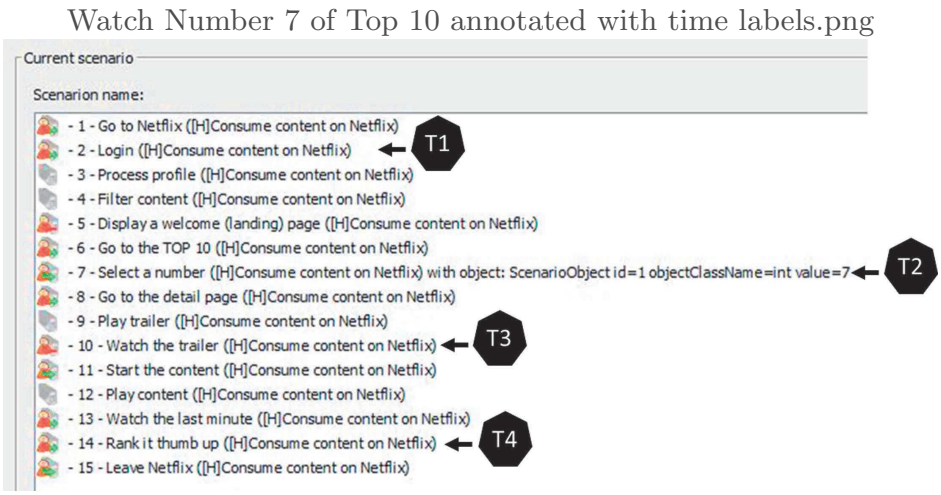

Fig. 9. Scenario "Watch number 7 of Top 10" produced using the task model simulator.

sequence of tasks. The final sequence of task we refer to as user activity as it contains only the actions that will be concretely performed by the user.

5.3 Data Collection

A remote user study was conducted to collect data for our case study. For that purpose, we prepared a guide for users how to perform such a remote study themselves, including descriptions of the activities to be performed, i.e. it included step-by-step instruction to be followed by the participants. Each step was linked with usability and user experience measures participants had to report as detailed in the following. Participants were also instructed to self-record themselves while performing the scenarios and extract the timings for different activities from it. In addition to the time taken for the different activities, participants had to report the following for each scenario: perceived difficulty of the scenario on a 5-point scale (1 = *very easy* to 5 = *very difficult*); aesthetic appearance of the interface, also on a 5-point scale (1 = *very beautiful* to 5 = *very displeasing*). Before commencing the scenarios, participants had to rate their anticipated UX on a scale from 0 (worst experience in your life) to 100 (heavenly experience). Once they were finished with all tasks, a *SUS* and *AttrakDiff* were used to measure cumulative UX. Participants were provided with a spreadsheet which they had to fill-in with the above mentioned requested data. The returned spreadsheets were checked for mistakes before processing them further.

Participants where free to choose one of two conditions (PC or Smartphone). Of the ten complete responses we received, seven performed the scenarios on a PC. These were imported into the visualization with exemplary results shown in Fig. 10.

5.4 The Visualization Approach Applied to the Use Case

Let us assume we are interested in getting an overview of how the different scenarios compare across the PC and Smartphone *Netflix* application. We thus take a look at the overview representation (Fig. 10, top) and can see that the aesthetics (A, ❶) on the smartphone were rated lower for each of the three scenarios it was assessed (more reddish values). In case of difficulty (D, ❶) there are no pronounced differences, except for the login which was rated to be easier on the PC, and which also shows in the much lower time needed for completing the activity. Generally, the scenarios were faster to solve on the PC than on the smartphone, where the times needed also exhibit more variation (box plots, ❷). We can also immediately see that users expect the same, rather high, user experience on both devices (dark green boxes, ❸). Cumulative UX measures were similar in general, suggesting that overall users had the same experience on both devices. However, the hedonic quality stimulation subscale of the *AttrakDiff* scored lower on the smartphone ❹.

Next, we are interested in taking a closer look at individual user experience and thus add the individual user views for each scenario (except the first, which only consisted of a single activity) and condition and arrange them like shown in Fig. 10 (middle). For instance, in case of the "Watch Number 7" scenario we can witness that users on the smartphone needed almost the same time for the "go to the details page and watch the trailer" activity (2nd to last box, ❺) while on the PC the timings are more varied. Its also apparent that one PC user was extremely fast (fourth user from the top) which could have been an expert user (or due to the user reporting wrong values). Lower timings in the other two scenarios confirm this impression to some extent. We can also notice that the time needed to complete the scenario was not necessarily a decisive factor for the perceived difficulty, with some people rating it more difficult while taking less time than others ❻. As another example, consider the individual timings for the "Watch Trailer" scenario where PC users usually required less time than smartphone users, except two. Of these two the first ❼ required relatively long for logging into Netflix (first box) and selecting the series (fourth box), pointing to some potential struggles of the user with navigating within Netflix. The other ❽ mainly did watch the trailer for a longer period of time (second to last box).

Suppose we would like to investigate overall differences in time needed for the different conditions in more detail. We thus add the *Aggregated User Activity View* for the "Watch Trailer" scenario for both the PC and smartphone and arrange them side-by-side (Fig. 10, bottom). Please note, that the color-coding is not based on individual trees but based on all displayed activity trees and thus can also be compared across them. For instance, we can observe that on average the whole scenario took longer to complete on the PC. However, it is also noticeable that users on the smartphone required longer to log into Netflix than PC users ❾. Searching for content (including the two sub-activities "locate search function" and "Type HYM") took about the some time in both conditions. Other noticeable differences arise for the "Watch Trailer" part of the scenario. Users on the smartphone spent more time on this activity, but this was not

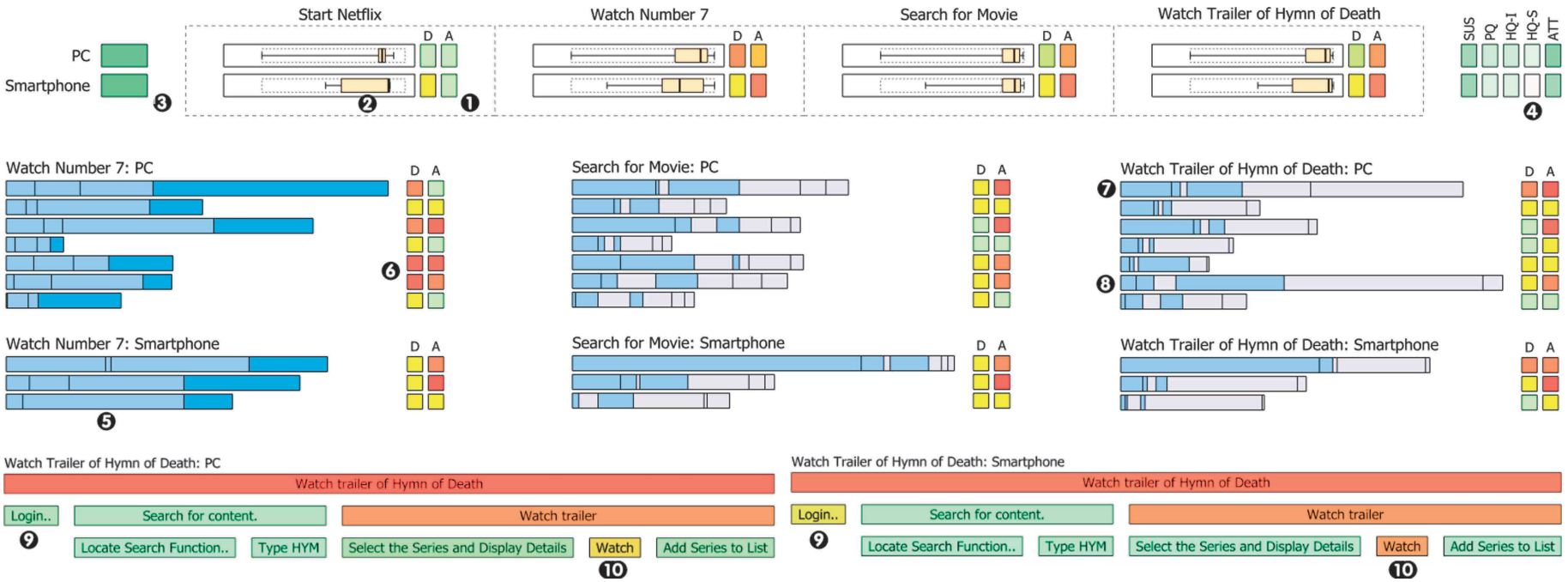

Fig. 10. Results of the evaluation visualized using the proposed approach. Overview (top) with added detail views of individual user data (middle) for all scenarios except the first and for both conditions. In addition, activity trees (bottom) for the "Watch Trailer" scenario for both conditions were added. Anticipated UX and cumulative UX (*SUS* and *AttrakDiff*) values are encoded using a white-green gradient ( ), episodic UX values in terms of difficulty (D) and attractiveness (A) with a green-red gradient ( ), with green corresponding to 'better' values. Time, as shown in the activity trees, also uses a red-green gradient with 2.3 94.7 s. Activities in the individual user views are color-coded based on the number of contained tasks, with $= 1$, $= 2$, and $= 3$. (Color figure online)

actually caused – as one might assume – by users requiring more time to select the series and display the details but rather by users spending more time on average to watch the trailer ❿.

6 Conclusion and Perspectives

A generic visualization approach for task-based evaluation is a first building block for a more systemic view on interactive system development. Today it is not enough any more to only look at the user(s)/system(s). Instead, we need a view on how organizations and regulations change the requirements of such systems and how the environment overall affects usability and UX (POISE model [1]).

The majority of systems developed today, has some form of task descriptions or basic models describing user activities. The usage of detailed task models is more common for large-scale systems that are either safety-critical or economically-critical. Nevertheless, the task model approach combined with this type of evaluation result visualization does not need detailed or elaborate task models, but can also be performed with rather abstract models. The ability to show evaluation data not only from generic to individual user activity and their relation to the task model, but to also enable the presentation of different software qualities in one visualization is a key advantage of this approach. Especially with recent worldwide developments such an approach will also help to support evaluation of societal impact (e.g., an evaluation of a tracking app for diseases vs privacy). In terms of research the key advantage of this approach is the ability to counteract the lack of longitudinal studies [15], by supporting the comparison and visualization of data also over different studies (and years), for example, by showing averaged usability scores for key activities.

Our proposed approach clearly has limitations and limits its functionality on purpose. It is not intended to replace and support all activities during an evaluation and excludes activities like recruiting processes for participants or extensive data analysis (that can be ideally performed using standard statistics programs). Nor is it intended for the ideation phase or early (sketch) design phases, or artefact research where the task performed by users with the artefact can not be foreseen.

We demonstrated the feasibility of the application of the visualization approach through a use case of comparing UX of consuming content on *Netflix* for two different conditions (PC and Smartphone). In the future, this visualization approach needs to be assessed with evaluators, who are the target users of this approach, to ascertain its usefulness and efficiency for them. There are also opportunities for future extensions to the visualization itself such as including more ways to interact with the data (e.g., filtering of the data based on different criteria) and representation of human errors made. Here, we also focused on the visualization itself and less on the integration into an existing toolchain which should deserve further attention as well.

References

1. Bernhaupt, R., Palanque, P.: POISE - a framework for systemic change analysis (2020, in preparation)
2. Bernhaupt, R., Navarre, D., Palanque, P., Winckler, M.: Model-based evaluation: a new way to support usability evaluation of multimodal interactive applications. In: Law, E.L.C., Hvannberg, E.T., Cockton, G. (eds.) Maturing Usability: Quality in Software, Interaction and Value. HCIS, pp. 96–119. Springer, London (2008). https://doi.org/10.1007/978-1-84628-941-5_5
3. Bernhaupt, R., Palanque, P., Drouet, D., Martinie, C.: Enriching task models with usability and user experience evaluation data. In: Bogdan, C., Kuusinen, K., Lárusdóttir, M.K., Palanque, P., Winckler, M. (eds.) HCSE 2018. LNCS, vol. 11262, pp. 146–163. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05909-5_9
4. Bernhaupt, R., Palanque, P., Manciet, F., Martinie, C.: User-test results injection into task-based design process for the assessment and improvement of both usability and user experience. In: Bogdan, C., et al. (eds.) HESSD/HCSE -2016. LNCS, vol. 9856, pp. 56–72. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44902-9_5
5. Brooke, J., et al.: SUS-a quick and dirty usability scale. In: Jordan, P.W., Thomas, B., McClelland, I.L., Weerdmeester, B. (eds.) Usability Evaluation in Industry, pp. 189–194. Taylor & Francis, Abingdon (1996)
6. Diaper, D.: Task analysis for knowledge descriptions (TAKD): the method and an example. In: Task Analysis for Human-Computer Interaction, pp. 108–159. Lawrence Erlbaum Associates (1990)
7. Eiselmayer, A., Wacharamanotham, C., Beaudouin-Lafon, M., Mackay, W.E.: Touchstone2: an interactive environment for exploring trade-offs in HCI experiment design. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. ACM, New York (2019)
8. Fahssi, R., Martinie, C., Palanque, P.: Enhanced task modelling for systematic identification and explicit representation of human errors. In: Abascal, J., Barbosa, S., Fetter, M., Gross, T., Palanque, P., Winckler, M. (eds.) INTERACT 2015. LNCS, vol. 9299, pp. 192–212. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22723-8_16
9. Goldberg, J.H., Wichansky, A.M.: Eye tracking in usability evaluation: a practitioner's guide. In: The Mind's Eye, pp. 493–516. Elsevier (2003)
10. Greenberg, S.: Working through task-centered system design. In: Diaper, D., Stanton, N. (eds.) The Handbook of Task Analysis for Human-Computer Interaction, pp. 49–66. Lawrence Erlbaum Associates, Hillsdale (2004)
11. Hassenzahl, M., Burmester, M., Koller, F.: AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. In: Szwillus, G., Ziegler, J. (eds) Mensch & Computer 2003. Berichte des German Chapter of the ACM, vol. 57, pp. 187–196. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-322-80058-9_19
12. Johnson, P.: Interactions, collaborations and breakdowns. In: Proceedings of the 3rd Annual Conference on Task Models and Diagrams, pp. 1–3. ACM, New York (2004)
13. Johnson, P., Johnson, H., Hamilton, F.: Getting the knowledge into HCI: theoretical and practical aspects of task knowledge structures. In: Cognitive Task Analysis. Lawrence Erlbaum Associates (2000)

14. Lallemand, C., Gronier, G.: Méthodes de design UX: 30 méthodes fondamentales pour concevoir et évaluer les systèmes interactifs. Editions Eyrolles (2015)
15. Lazar, J., Feng, J.H., Hochheiser, H.: Research Methods in Human-Computer Interaction. Morgan Kaufmann, Burlington (2017)
16. Marti, P., Iacono, I.: Anticipated, momentary, episodic, remembered: the many facets of user experience. In: 2016 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 1647–1655 (2016)
17. Martinie, C., Palanque, P., Bouzekri, E., Cockburn, A., Canny, A., Barboni, E.: Analysing and demonstrating tool-supported customizable task notations, vol. 3. Association for Computing Machinery, New York, June 2019. https://doi.org/10.1145/3331154
18. Martinie, C., Palanque, P., Ragosta, M., Barboni, E.: Extending procedural task models by systematic explicit integration of objects, knowledge and information. In: Proceedings of the of the 31st European Conference on Cognitive Ergonomics, ECCE 2013. ACM, New York (2013)
19. Martinie, C., Palanque, P., Navarre, D., Winckler, M., Poupart, E.: Model-based training: an approach supporting operability of critical interactive systems. In: Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS 2011, pp. 53–62. ACM, New York (2011)
20. Meng, X., Foong, P.S., Perrault, S., Zhao, S.: NexP: a beginner friendly toolkit for designing and conducting controlled experiments. In: Bernhaupt, R., Dalvi, G., Joshi, A., K. Balkrishan, D., O'Neill, J., Winckler, M. (eds.) INTERACT 2017. LNCS, vol. 10515, pp. 132–141. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67687-6_10
21. Navarre, D., Palanque, P., Paternò, F., Santoro, C., Bastide, R.: A tool suite for integrating task and system models through scenarios. In: Johnson, C. (ed.) DSV-IS 2001. LNCS, vol. 2220, pp. 88–113. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45522-1_6
22. Paterno, F.: Task models in interactive software systems. In: Handbook of Software Engineering and Knowledge Engineering. World Scientific (2000)
23. Paternó, F., Mancini, C., Meniconi, S.: ConcurTaskTrees: a diagrammatic notation for specifying task models. In: Proceedings of IFIP INTERACT 1997, pp. 362–369. Lawrence Erlbaum Associates (1997)
24. Pinelle, D., Gutwin, C., Greenberg, S.: Task analysis for groupware usability evaluation: modelling shared-workspace tasks with the mechanics of collaboration. In: ACM Transactions on Computer-Human Interaction, vol. 10, no. 4. pp. 281–311. ACM, New York (2003)
25. Pirker, M.M., Bernhaupt, R.: Measuring user experience in the living room: results from an ethnographically oriented field study indicating major evaluation factors. In: Proceedings of the 9th European Conference on Interactive TV and Video, pp. 79–82. ACM, New York (2011)
26. Ragosta, M., Martinie, C., Palanque, P., Navarre, D., Sujan, M.A.: Concept maps for integrating modelling techniques for the analysis and re-design of partly-autonomous interactive systems. In: Proceedings of the 5th International Conference on Application and Theory of Automation in Command and Control Systems, ATACCS 2015, pp. 41–52. ACM, New York (2015)
27. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: an open architecture for collaborative filtering of netnews. In: Smith, J.B., Smith, F.D., Malone, T.W. (eds.) Proceedings of the Conference on Computer Supported Cooperative Work, pp. 175–186. ACM (1994)

28. Rosson, M.B., Carroll, J.M.: Usability Engineering: Scenario-Based Development of Human-computer Interaction. Elsevier, Amsterdam (2001)
29. Roto, V., Law, E., Vermeeren, A., Hoonhout, J.: User experience white paper: bringing clarity to the concept of user experience. In: Dagstuhl Seminar on Demarcating User Experience, pp. 1–12 (2011)
30. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: Proceedings of the 1996 IEEE Symposium on Visual Languages, pp. 336–343 (1996)
31. Van der Veer, G., Lenting, B., Bergevoet, A.: GTA: groupware task analysis - modelling complexity. In: Acta Psychologica, New York, NY, vol. 91, no. 3, pp. 297–322 (1996)
32. Wallner, G., Kriglstein, S.: Visualization-based analysis of gameplay data - a review of literature. Entertain. Comput. **4**(3), 143–155 (2013)
33. Winckler, M., Palanque, P., Freitas, C.: Tasks and scenario-based evaluation of information visualization techniques. In: Proceedings of the 3rd Annual Conference on Task Models and Diagrams, TAMODIA 2004, pp. 165–172. ACM, New York (2004)