

# Task Analysis-Based User Event Logging for Mobile Applications

Qin Liu  
School of Software  
Engineering  
Tongji University  
Shanghai, China  
qin.liu@tongji.edu.cn

Rafael Durán-Sáez  
DLSIIS  
Universidad Politécnica de  
Madrid  
Madrid, Spain  
rafaduransaez@gmail.com

Xavier Ferre  
DLSIIS  
Univ. Politécnica de Madrid  
Madrid, Spain  
High-end Expert at Tongji  
University, Shanghai, China  
xavier.ferre@upm.es

Hongming Zhu  
School of Software  
Engineering  
Tongji University  
Shanghai, China  
zhu\_hongming@tongji.edu.cn

**Abstract**—Task analysis defines the user-centered tasks that users will carry out to use a system, and it serves as the basis for the definition of usability requirements related to efficiency and effectiveness. Usability and UX (User eXperience) are especially important for mobile application development. We aim to provide support to task modelling for Android application development, and to specify user events to be logged for further analysis. Selection of the task modelling technique has been carried out in a two steps approach, allowing, for the next step of developing, a tool to support task modelling and annotation.

**Keywords**—task analysis, task annotation, automated usability evaluation, usability, Android

## I. INTRODUCTION

Usability is the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use [1]. Usability requirements in the form of performance objectives are linked to a specific context of use that is to specific users, their goals and tasks [2]. Task analysis deals with the identification and specification of the structure, the flow, and the attributes of tasks.

Evaluation of the designs against requirements is one of the main activities of User-Centered Design (UCD), a development approach that aims to build usable products and systems [3]. Usability testing is a basic evaluation UCD method, based in having representative users use the system to measure usability criteria (possibly specified in the form of usability requirements). Alternatively, long-term monitoring studies, also known as follow-up studies, collect usage data under real use, and, according to Rubin & Chisnell, they are probably the truest and most accurate appraisals of usability [3].

Mobile apps differ from other systems in that they can be operated in a variety of contexts of use, thus making the task of usability testing under real-use conditions especially difficult. Mobile app development has a big pressure to achieve a shorter time-to-market due to increased competition, so reducing usability testing and carrying out long-term monitoring after release are especially appropriate for app development projects. Logged data can be analysed for both assessing the fulfilment of usability requirements and to uncover usability problems to need fixing.

One of the challenges of user event logging is the storage and retrieval of a huge amount of data in a dedicated server. Google Analytics for Mobile Apps (GAMA) offers a cloud service for keeping track of user actions in mobile apps. We have carried out a preliminary case study [4] where the feasibility of storing usage data in GAMA has been proved.

In this paper we formalize the task analysis activity necessary to carry out user event logging with a UCD focus, linked to usability requirements. The formalization aims to be manageable by average developers with an interest in usability issues, by means of a supporting tool built in the form of an Android Studio plugin.

## II. RELATED WORK

There exist many automated usability evaluation methods based on user events logging, especially in the web domain [5] [6], but the mobile domain has relevant differences with the web one. Mobile environments are constantly changing and concepts like location, connectivity, data communication, etc. are relevant for mobile development and dependent of the wide variety of users. Differences between mobile devices and limited screen space are also very relevant in the mobile domain.

Lettner and Holzmann [7] propose a toolkit aimed to automated and unsupervised evaluation for mobile apps, tracking users' behaviour through the interaction with the application, but they do not take into account task definition and modelling, relevant aspects for our project.

The EvaHelper framework [8] allows the insertion of events-calls code into the evaluated application code in order to indicate when the gathering of data will take place, but, again, it does not considerate task description and modelling.

Others, like [9] and [10], focus their work on video recording of user behaviour and live emotions logging through high quality mobile devices front cameras. This approach requires the existence of electronic devices for video recording.

The WUP (Web Usability Probe) tool [11] is based on the analysis of web applications by focusing on the client-side logs as data source. Its authors suggest a redefined tool for mobile applications based on web services.

Matsuzawa, Iwata and Shirogane [12] focus on usability evaluation by setting up user tests to acquire user operation histories, identify problem areas and later analyse the logged data, through a tool called Logcat, so they do not aim to evaluate the real usage of the app. They do not consider task modelling on their proposal either.

Porat, Schclar and Saphira [13] highlight the fact the analyser should spend time on deciding what type of metrics will be used for logging the data. Their tool, called MATE (Mobile Analysis Tool for Usability Experts), uses different effectiveness, efficiency and use-frequency metrics to define task scenarios to use in usability tests. This tool does not specify any task modelling technique to deploy the tasks.

RemUSINE extends a tool for evaluating desktop applications [14], and it is based on the comparison of the ideal path and the one travelled by the user, by logging different kind of timelines. To compare both routes, they predefine the tasks using the CTT modelling technique, later analysed in this document.

As a summary, most of these solutions are either not generically applicable to a variety of app development projects, or they are complemented by external recording devices. Additionally, none of them takes into account task analysis for basing the choice of relevant user events to log.

### III. TASK ANNOTATION APPROACH

Task analysis is used for identifying and understanding the structure, the flow, and the attributes of tasks. It works as a basis for, once the tasks are determined, facilitating the definition and formalization of usability requirements. It implies identifying the actions and cognitive processes required for a user to complete a task or achieve a particular goal. Some of their outputs are detailed descriptions of all kind of activities involved with each task, duration and variability, frequency, allocation, complexity, environmental conditions, etc.

UCD's main goal is to reach a high level of usability by considering the user the centre of focus. Task analysis provides a way of identifying and designing user activities, with or without a computer system involved. It combines the context and objectives the user is focused on achieving with the tasks actions to carry out and the correct workflow to stick to. It allows identifying and determining what concepts are relevant for the analysis and what aspects of tasks we should focus on in order to assure the task execution is the proper one. In other words, it permits to deeply dig in user actions and purposes to a better performance of tasks. Therefore, an early focus during the development of a project on user and task analysis is paramount for the success of requirements activities.

Task analysis activities make use of task models to represent the steps a user carries out when executing a task. Task models describe how to perform activities to reach user's goals. They represent the ideal interaction between the user and the application's user interface. Task models play an important role in the HCI field, because they represent the logical activities that should support users in reaching their goals.

Usability requirements are typically based on task analysis, since they specify efficiency and effectiveness objectives for the enactment of specific tasks. As shown in Fig. 1, task analysis is the base to provide information to define usability requirements in detail and also the possibility to redefine them after a real usage evaluation by getting use of the user's behaviour data when interacting with the app.

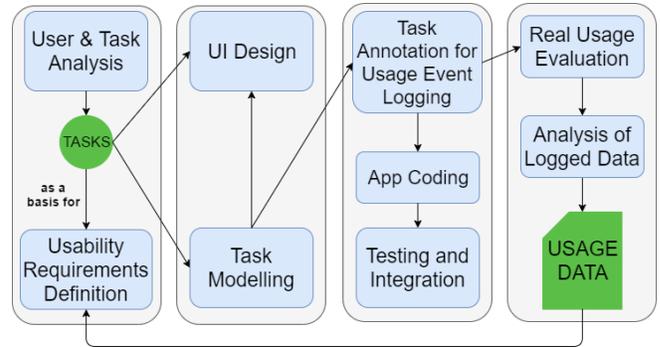


Figure 1. Development Activities

### IV. SELECTION OF TASK MODELLING METHOD

There are many representations of task models, but not all of them successfully fit with the requirements and necessities of all projects. Thus, we carried out a bibliographic research [15] of the most popular task models by the professionals in the field: HTA [16], GOMS [17], GTA [18], CTT [19], MAD\* [20], TKS [21], Diane+ [22], MUSE [23], TODD [24] and Use Cases [25].

We carried out a comparative analysis to select the most appropriate task modelling method for supporting user event logging for usability purposes (see Table 1). Our assumptions are the following ones: (a) it is easier to decompose a task by using graphical methods than only narrative text, although the existence of this last one eases the understanding of the task execution; (b) the tool needs a low complexity to facilitate tool learning; and (c) to achieve clarity and low complexity we just focus on individual tasks. Graphical Notation (GN) represents if a specific graphical nomenclature is provided for drawing a diagram representing the task decomposition; Narrative (N) whether the task decomposition is accompanied by narrative description of each step and the way to move from one to another; Task Orientation (TO): if the tasks modelled are oriented towards individual operation, cooperative or both; and Complexity (C), representing the level of complexity when applying the technique taking into account learning time, resources and procedures. The last column corresponds to Suitability (S), representing the level of affinity with our project needs. Green fields are graded with 2 points, yellow ones with 1 and

TABLE I. COMPARATIVE ANALYSIS AMONGST TASK MODELLING TECHNIQUES

	GN	N	TO	C	S
HTA	Specified	Yes	Individual	Low	8
Use Cases	Not Specified	Yes	Individual	Low	6
CTT	Specified	Limited	Both	Medium	6
GOMS	Not Specified	Limited	Individual	Low	5
MAD*	Not Specified	No	Both	Low	4
Diane+	Specified	No	Individual	High	4
TODD	Specified	No	Both	High	4
TKS	Not Specified	No	Individual	Medium	3
MUSE	Not Specified	No	Individual	Medium	3
GTA	Not Specified	No	Cooperative	Medium	1

By using Use Cases, we are able to explain to the user everything we want him/her to do in detail, because it is based on a narrative structure and it is actually the best way to define and describe any kind of problem. However, graphical support is an important factor to take into account, and it is missing in this technique. Use Cases provide easier understanding about the steps to do and how to do them, they clarify more the idea of performing tasks step by step, subtask by subtask. However, it lacks the expressiveness of graphic models to synthesize visually the main task features.

CTT is able to explain and narrate everything the user should do by just using a graphical diagram, a task tree. Its nomenclature identifies what kind of relation there is among all the linked elements. There exist different kind of linking constructors in order to indicate to the user what to do and how to do it. The relatively complex CTT syntax hinders its adoption by mobile app developers, with an important learning curve.

HTA is a kind of a mix of both previous techniques: Use Cases and CTT. It provides a task tree structure and plans to narratively describe the workflow and the different subtasks to perform. The plans substitute the nomenclature of CTT and facilitates the understanding and learnability of the method behaviour. This combination offers a better work fluency. Thus, we consider HTA the best modelling technique for our purpose. It provides graphical expressiveness with the added flexibility of natural language, and a low complexity. So we expect the learning of this technique to be less time-consuming for developers.

## V. TOOL SUPPORT

We are currently designing a plugin to support developers and analysts to perform task analysis annotation and facilitate the understanding and definition of usability requirements and further usability evaluation activities by modelling tasks through HTA. The plugin provides a diagram editor to allow users to graphically decompose the tasks into smaller units, named subtasks, for easier

examination. Each of those subtasks can be individually treated in order to determine what events to log. Moreover, different combinations of tasks can also be controlled and analysed depending on the purposes of the evaluation, e.g. for checking the correct execution of a sequence of tasks.

Not all the elements of an app interface are relevant for the analysis. As it happens with the elements, not all the occurring events are significant for the evaluation. The breakdown of the tasks is the fact that makes possible the identification of what elements have a significant value for the usability examination. The plugin allows the user to determine what objects s/he is interested in assessing and link them to the correspondent events to control by indicating in what part of the app code those objects are located and advising where to add the code in charge of logging the data. The user, through the diagram editor, as shown in Fig. 2, is always able to take control of which events are being evaluated. Depending on the elements or actions to collect data from, the support of the plugin varies with reference to the amount of code the user has to manually enter. For instance, if the user wants to log the number of clicks on any element or the time spent on the execution of any specific task, the code in charge of gathering information will be automatically injected by the plugin. However, complex events to control like getting specific object names or attributes, require the collaboration of the user by inserting some code in the areas suggested and facilitated by the plugin.

The data is gathered at the same time the application is used. As well as the data is collected, it is automatically sent to the GAMA servers in order to proceed to the analysis of the collected data.

The data collected by the plugin could be also used to further the comparison to the initial established usability requirements. By analysing the behaviour and interaction of the user with the application we can check whether the requirements have been respected. Then, depending on the results, we can easily detect what UI elements to modify or what requirements we should update, as it happens in Fig. 3,

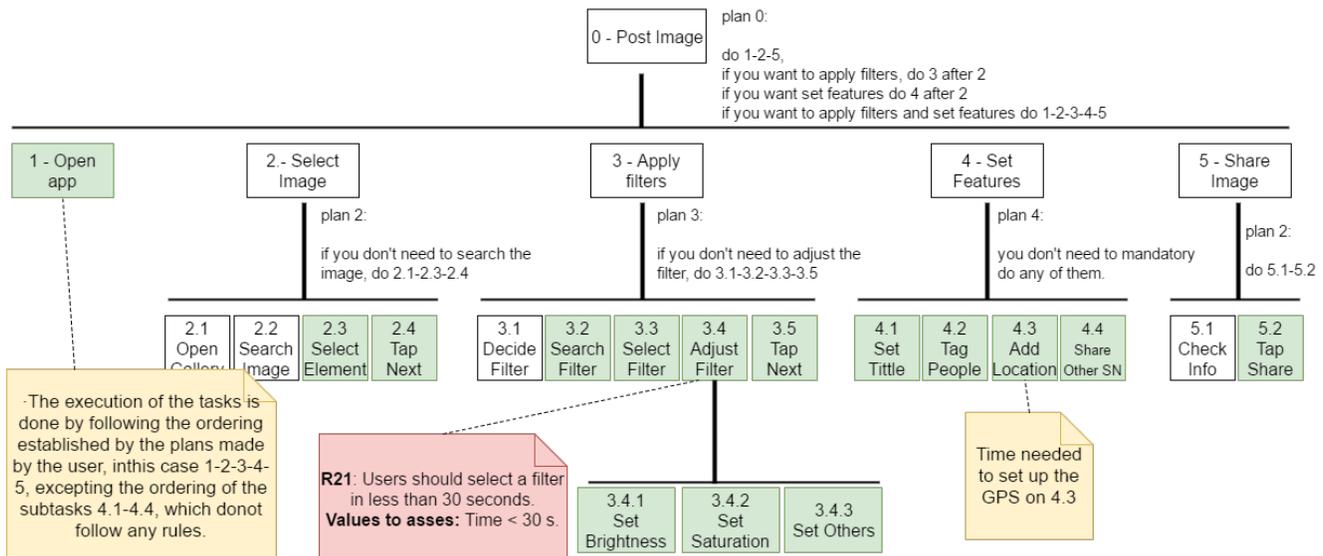


Figure 2. Plugin task analysis decomposition aspect of the example case of posting a photo on Instagram application

**R21:** User should select a filter in less than 30 seconds.  
**Results:** User spent 40 seconds to complete the task 3.4.

Figure 3. Requirements-Results comparison example

where the results of the logged data analysis show that users took too long to complete the assessed task.

## VI. CONCLUSIONS

We have introduced an approach for providing support to task analysis annotation in Android apps for long-term monitoring of usage. We have carried out a research on the most popular task modelling methods to select the best one for our objectives. HTA has been chosen, being the clearest and one of the easiest for learning its usage, providing a low level of complexity.

We are recently developing the plugin for Android Studio in charge of offering the mentioned support to developers to task analysis annotation by allowing the decomposition of any task into diverse sub-elements through a simple diagram editor by following the extended HTA task modelling method presented in this document. The plugin advises the code areas where to ideally inject the code in charge of logging the data of those elements and procedures represented in the HTA decomposition drawn by the user. Furthermore, at the same time the plugin is gathering the data, it is sending the information to GAMA servers for a future analysis and critique.

Next step will consist in testing the developed plugin with different type of users and real applications. We are currently carrying out task modelling, usability requirements definition and code instrumentation for two Android mobile applications to be used as case study for our proposal and prove its feasibility.

Future work will be related to the increase of the automation of the plugin. Automated facilitation of the elements and events for logging user-interaction data requires a deep study on all the domains that Android development encompasses. Further work could be dedicated to extend the tool to other integrated development environments, and also to iOS mobile operating system.

## ACKNOWLEDGMENT

We thank the anonymous reviewers whose feedback and suggestions helped to improve and clarify this manuscript.

## REFERENCES

- [1] ISO 9241-210, Ergonomics of human-system interaction - Part 210: Human-centered design for interactive systems, 2010.
- [2] NIST, Common Industry Specification for Usability - Requirements, NISTIR 7432, 2007.
- [3] J. Rubin and D. Chisnell, Handbook of Usability Testing, Wiley Publishing, 2008.
- [4] X. Ferre, E. Villalba, H. Julio and H. Zhu, «Extending Mobile App Analytics for Usability Test Logging,» de *INTERACT 2017*, 2017.
- [5] E. Arroyo, T. Selker and W. Wei, «Usability tool for analysis of web designs using mouse tracks,» In: *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, 2006.
- [6] L. Paganelli and F. Paternò, «Intelligent analysis of user interactions with web applications,» In: *Proc. of the 7th Intl. Conf. on Intelligent user*, 2002.
- [7] F. Lettner and C. Holzmann, «Automated and unsupervised user interaction logging as basis for usability evaluation of mobile applications,» In: *Proc. of the 10th Intl. Conf. on Advances in Mobile Computing & Multimedia*, 2012.
- [8] F. Balagtas-Fernandez and H. Hussmann, «A methodology and framework to simplify usability analysis of mobile applications,» In: *2009 IEEE/ACM Intl. Conf. on Automated Software Engineering*, 2009.
- [9] J. Feijó Filho, T. Valle and W. Prata, «Automated Usability Tests for Mobile Devices through Live Emotions Logging,» In: *17th Intl. Conf. on Human-Computer Interaction with Mobile Devices and Services Adjunct*, 2015.
- [10] K. Leichtenstern, D. Erdmann and E. André, «An Evaluation Component for Mobile Interfaces,» In: *10th Intl. Conf. on Human Computer Interaction with Mobile Devices and Services - MobileHCI 08*, 2008.
- [11] P. Burzacca and F. Paternò, «Remote usability evaluation of mobile web applications,» *Intl. Conf. on Human-Computer Interaction*, pp. 241-248, 2013.
- [12] M. Matsuzawa, H. Iwata and J. F. Y. Shirogane, «Support method of usability evaluations for Android applications based on operation histories,» In: *Fifth Intl. Conf. on Digital Information Processing and Communications (ICDIPC)*, 2015.
- [13] T. Porat, A. Schlar and B. Shapira, «MATE – A Mobile Analysis Tool for Usability Experts,» In: *Extended Abstracts on Human Factors in Computing Systems - CHI EA'13*, 2013.
- [14] F. Paternò, A. Russino and C. Santoro, «Remote Evaluation of Mobile Applications,» In: *Proc. of the 6th Intl. Conf. on Task models and diagrams for user interface design 2007*, 2007.
- [15] D. Diaper and N. A. Stanton, The Handbook of Task Analysis for Human-Computer Interaction, London: Lawrence Erlbaum Associates, 2004.
- [16] J. Annet and K. Duncan, Task analysis and training design, Occupational Psychology, 1967.
- [17] S. K. Card, T. P. Moran and A. Newell, The psychology of human-computer interaction, Hillsdale, NJ: Lawrence Erlbaum Associates, 1983.
- [18] M. van Welie, G. C. van der Veer and A. Eliens, «An ontology for task world models,» In: *Proc. of the Fifth Intl. Workshop on Design, Specification, and Verification of Interactive Systems (DSV-IS '98)*, 1998.
- [19] F. Paternò, Model based design and evaluation of interactive applications, 1999.
- [20] D. Scapin and C. Pierret-Golbreich, «Towards a method for task description,» In: *Proc. of Work with Display Units (WWU '89)*, 1989.
- [21] P. Johnson, Human-computer Interaction: Psychology, task analysis and software engineering, London: McGraw-Hill, 1992.
- [22] M. F. Barthelet and J. C. Tarby, The Diane+ method., Namur, Belgium: Presses Universitaires de Namur, 1996.
- [23] K. Y. Lim and J. Long, The MUSE method for usability engineering., Cambridge: Cambridge University Press, 1994.
- [24] A. Mahfoudhi, M. Abed and D. Tabary, From the formal specifications of user tasks to the automatic generation of the HCI specifications, London: Springer, 2001.
- [25] I. Jacobson, M. Christensen and P. Ö. G. Jonsson, Object-Oriented Software Engineering - A Use Case Driven Approach, Addison-Wesley, 1992.