

Design of a Software Tool for Mobile Application User Mental Models Collection and Visualization

Radek Vala, Roman Jasek, David Malanik

Tomas Bata University in Zlin, Faculty of Applied Informatics, nám. T.G.Masaryka 5555,
CZECH REPUBLIC
{vala, jasek, dmalanik}@fai.utb.cz

Abstract. Mental model is a fundamental term in human-computer interaction (HCI). The process of HCI creates different qualitative levels of user experience (UX) which can be determined by the quality of communication between a technological product (the system) and a user of this product. The level of user experience is indirectly derivable from measurement of different qualitative and quantitative aspects of the interaction (achievement of objectives, satisfaction, comfort, time required to perform the task). The user experience is the key factor which determines the popularity of web or mobile applications and software applications in general. During the HCI, the user is unconsciously comparing his mental model (own idea of functionality) with the system image (the real graphical user interface – GUI). The distance between the user's mental model and the system image should be minimal. Creation of successful GUI requires a real user-application interaction testing and statistical evaluation of the testing data. This paper describes a design of a software tool for user mental model collection and visualization in the area of mobile applications with emphasis on context of the HCI.

Keywords: conceptual model, mental model, mobile application design, system image, user experience, user interface.

1 Introduction

Recently, more and more mobile applications developers are facing the question of proper graphical user interface design. Area of mobile application development is rapidly growing along with increasing number of mobile applications users. According to Flurry Analytics [1], overall mobile applications use in 2013 growth by 115% year-over-year and it is expected that this trend will continue.

In segment of native application development, there is possible to follow guidelines for GUI design related with some specific platform. But these guidelines are describing only fundamental patterns or use cases. In contrast with web application UI design, there is a noticeable lack of studies and research in the area of mobile application design. Moreover, if we consider the hybrid mobile application development, a large amount of user interface design frameworks is not strictly following particular platform UI guidelines and achieving a successful mobile application design is not therefore a simple task.

The interaction between a user of mobile application and a graphical user interface could lead to some misunderstandings, errors and frustration from inability to achieve a goal. Designer is able to prevent this situation with a good UI design with respect to the user mental model (UMM).

By the user-application interaction, a user is comparing his UMM (complex idea of how the system works) with the system image (application GUI). If the distance between this two models is too high, it means that the design is confusing and users may not be able to accomplish their goals.

Good UI of a software application (system image) should help a user to create a productive UMM of the system. [2] This paper discuss best practices for consideration by a designer creating a conceptual model of an application and a software tool for UMM collection and evaluation.

2 Mental Models in Software Design

In the area of human-computer interaction (HCI) there are defined fundamental terms describing and simplifying the human computer recognition process.

2.1 Mental Models

The term Mental Model was firstly introduced by Craik [3], but in the 80th it became the fundamental part of terminology of the newly established field of cognitive science. According to Norman [4] and Krug [5], UMM is the key factor in user's perception of an object functionality and behavior. Users of an object are creating in their minds a simplified model that describes their ideas how the object works, or how to interact with it.

2.2 Conceptual Models

By the process of developing a software application, a conceptual model is created by the designer. This conceptual model reflects designer's understandings of the task and tools and abstractly describes the functionality of the system and its relations. [6] This model should be good understandable for end user and should focus on key functionality of the application. [7]

2.3 System Image

User of a software application compares unconsciously by working with GUI objects, own idea about virtual environment functionality, with real GUI objects which are creating the system image. In simplicity, by the process of HCI, user compares own mental model with the system image. [8]

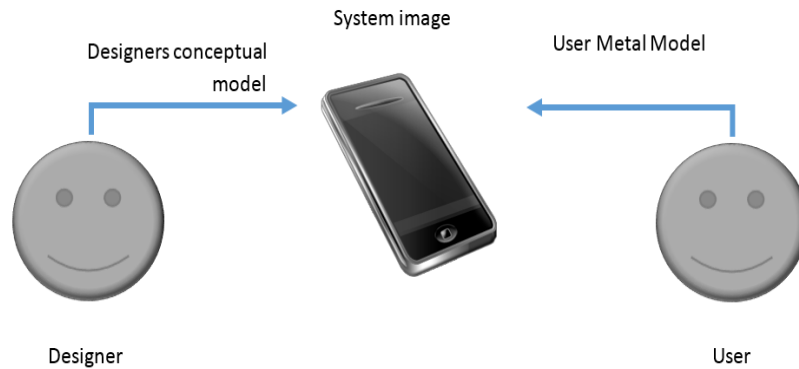


Fig. 1.Relation between conceptual model, user mental model and system image.

3 Conceptual Design Best Practices

Good conceptual model is an essential part in development process of successful software application. [9] Therefore application designer should consider following best practices published by IBM Corporation.

3.1 Simplicity

Mental models are the simplified image of reality, therefore GUI should simplified the key functions of the system. This key functions should be highlighted, lesser-used functions should be in background.

3.2 Familiarity

Users in general have some prior knowledge and using this knowledge, they are creating own mental model. The GUI should allow them to build on this knowledge.[10] The process of creation an adequate mental model, can be strengthen if the user is able to apply prior experience gained from the real world.

3.3 Availability

Because human beings are better at recognition then recall, GUI should contain visual stimulus, to fast identification of an object functionality.[10]

3.4 Feedback

GUI should provide continuous feedback about the results of actions.[4] Using appropriate feedback is possible to support user's mental model creation.[10] Positive feedback is good for strengthening current user's mental model, whilst using the negative feedback is possible to adapt the model.

4 Proposed Software Tool – MeMo2Ap

In order to assess the accuracy of the conceptual models, it is necessary to obtain a general UMM of specific application and to evaluate the extent to which these two models fits. Collecting the user models can be very costly activity and requires special commercial software or a test observer.

This paper describes a design of a context focused software solution for mobile application user mental models collection and visualization (named MeMo2Ap). Before the design of this software tool, these goals were formulated:

- 1) Simple implementation
- 2) Easy preparation of test scenario
- 3) Context focus (target application, target device)
- 4) Distributed test deployment
- 5) Results visualization and simple evaluation

To meet this objectives, hybrid mobile application development approach was chosen. Within this approach it is possible to use web technologies which are easy to implement and moreover it is very easy to distribute the test application to end users using URL address. In other way it is also possible to wrap the test application by wrapper technology, such as PhoneGap [11] and publish it on official distribution channels.

4.1 Principles of Testing and Data Collection

Hybrid mobile application MeMo2Ap is a client-server application which is able to perform a testing scenario and observe users touch gestures. Testing scenario contains 1 to N test cases, while the test case is determined by a test screen, description of user task to perform, and success area, where user should touch to complete the task. If the user touches the success area, the test was successful in other case, the test failed.

Testing result is immediately sent from client mobile device to database table on the server. Timestamp, vertical and horizontal position of the touch gesture are stored. This communication between client and server is established using JSON web service. The main advantage is the possibility of distributed testing and simple delivering to end testers.

4.2 Technologies Used for MeMo2Ap

As for server side, PHP programming language and MySQL database was chosen. The server was developed using QCubed [12] – open-source rapid application development framework, which uses ORM technology [13] and code generation [14] to accelerate the development process.

MySQL database was designed using open-source visual database designer MySQL Workbench 6.0 by Oracle (www.mysql.com/products/workbench/). Database table names and relations are shown in Figure 3.

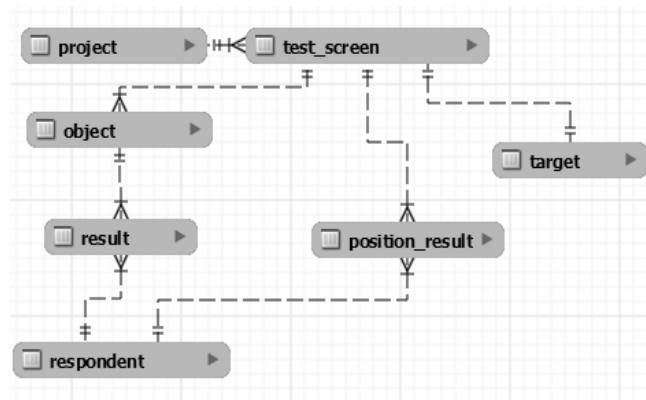


Fig. 2.MeMo2Ap database table relations.

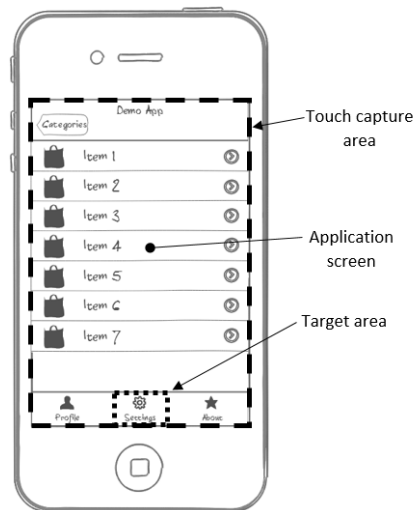


Fig. 3. Visualisation of target area within test screen.

The server side consists of the tests administration area (automatically generated QCubed Form Drafts [15]) and web services for the mobile client. The client side is

created using web technologies – HTML, CSS, JavaScript, jQuery and jQuery Mobile and it is possible to process the application in a mobile browser or a Web view within a native mobile application using Phonegap wrapper [11].

4.3 Proposed methodology of collecting user's mental models using MeMo2Ap

Using MeMo2Ap testing software is possible to create test scenarios in following way:

- 1) New testing project with specific name and identifier is created using server administration tool.
- 2) Within this project, new testing scenario can be created.
- 3) Testing scenario consists of 1 to N testing screens. Each testing screen is represented by PNG graphic file and is possible to prepend user task text which is shown as a modal dialog window, before the test screen.
- 4) When the project testing scenario consists of at least 1 test screen, it is possible to run the script for setting target area (Fig. 4). Setting the target area is done using drag & drop technology, by the administrator.

Once the test scenario is set, the user testing can be performed remotely, if needed (Figure 5) and the testing results are obtained in real time.

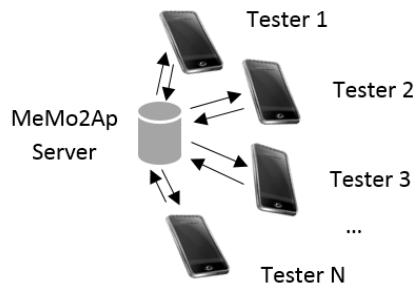


Fig. 4. Distributed testing using client-server architecture.

Using MeMo2Ap testing software from tester point of view:

- 1) Completion of initial form (information about gender, preferred platform, experience...)
- 2) Reading a user task.
- 3) Longer tapping the area to complete the task.

Points 2) and 3) are repeating for each test screen.

4.4 Testing Results Visualization and Evaluation

Once testing is complete (or during the testing), visualization can be displayed using MeMo2Ap software tool (Figure 5). Each test screen is overlaid with testing results

with successful touches in green and unsuccessful touches in red color. There is also obvious the frequency of tapping into particular area, from the color density.

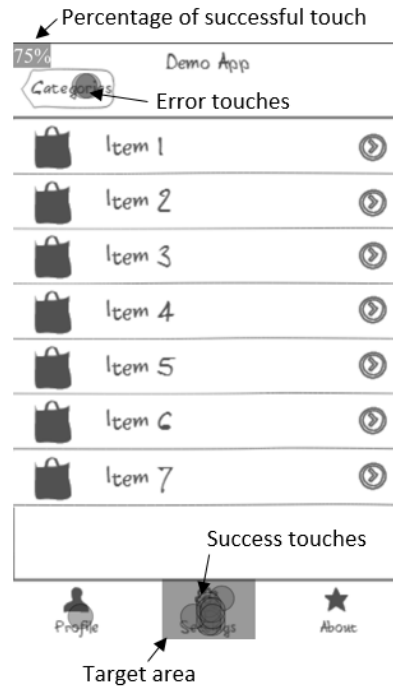


Fig. 5. Visualization of testing results.

In the left corner of each test screen, there is available the numerical information about successful tests in percent. Percentages are given by a success percent sP (1) of each test screen, where tT is number of total test count and sT is number of successful taps in target area.

$$sP = \left(\frac{sT}{tT} \right) \cdot 100 \quad (1)$$

5 Method of Incremental GUI Design Using Mental Models

Creating a mobile application with usable, attractive and especially understandable design can provide higher sales and growing numbers of users. On the other hand, bad user experience makes an application unsuccessful.

During the application design phase, conceptual model of the app is created. This conceptual model produces the system image with which the real user interacts. Inaccurate system image do not support creation of correct mental model of the user.

With respect to user mental model linked with specific application screen, it is possible to rearrange and redesign problematic parts and redo the tests. Comparing the

results between the first system image and the redesigned system image is possible to highly improve the user experience.

After the testing, a successful threshold for each test screen should be set up. If the success percentage of the test screen is lower than the threshold, design improvements are necessary. Another test should contain the improved test screen and again the success percentage is evaluated.

Better mobile design with high user experience can be achieved using this incremental testing (Figure 6).

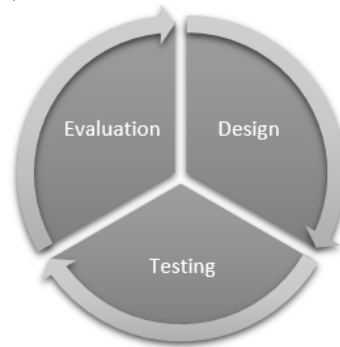


Fig. 6. UI design with incremental testing.

6 Future Research

Software tool MeMo2Ap for collecting mobile application user's mental model was developed as an open-source and allows easy creation of test scenarios, which are focused on evaluation the accuracy of conceptual models. Using this tool and incremental method of GUI design testing, an improvement in UX could be achieved.

In future research commonly used GUI patterns could be tested and the extent to fit user's mental model could be evaluated. It is also possible to search for the error dependence on user preference for specific mobile platform by using patterns linked to another platform.

Another interesting area is the creation of proper user mental models supported by appropriate UI techniques, such as overlap help. Research can be conducted with two testing groups – users who have not seen the overlap help and those who have seen the overlap help in prior.

7 Conclusion

Method of collecting mobile application user's mental models using the proposed software tool MeMo2Ap was introduced in this paper and best practices for mobile application GUI design were discussed. The software tool was designed as an open-source client-server application, to meet these goals: Simple implementation, easy preparation of test scenario, context focus (target application, target device), distributed test deployment, results visualization and simple evaluation.

Using the MeMo2Ap tool, the success rate of each test screen is reflecting the correctness of the conceptual model and if the correctness is insufficient it is possible to improve it with incremental design enhancing and testing. Also opportunities for further research, such as established GUI pattern success rate testing and evaluating or supporting methods of creation the proper user mental model are mentioned in chapter 6.

Acknowledgment

The research work was performed to financial support by the European Regional Development Fund under the Project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

References

1. S. Khalaf. (2014). *Mobile Use Grows 115% in 2013, Propelled by Messaging Apps*. Available: <http://blog.flurry.com/bid/103601/Mobile-Use-Grows-115-in-2013-Propelled-by-Messaging-Apps>. Last accessed 2014-03-04.
2. J. Preece. *Human-Computer Interaction*. Reading MA: Addison-Wesley, 1994.
3. K. Craik. *The nature of explanation*, Cambridge University Press, 1943.
4. D. A. Norman. *Design of everyday things*. Rev. and exp. ed. New York: Basic Books, 2013, xviii, ISBN 978-046-5050-659.
5. S. Krug *Don't make me think!: a common sense approach to Web usability*. Indianapolis, Ind.: Que, 2000, ix, ISBN 07-897-2310-7.
6. J. Johnson, A. Henderson. *Conceptual models: core to good design*. San Rafael, Calif., 2012. ISBN 978-160-8457-496.
7. D. A. Norman. *Design of everyday things*. Rev. and exp. ed. New York: Basic Books, 2013, xviii, ISBN 978-046-5050-659.
8. D. A. Norman, S. W. Draper. *User centered system design: new perspectives on human-computer interaction*. Pbk. ed. Boca Raton, FL: CRC Press, 1986. ISBN 978-089-8598-728.
9. J. Johnson, A. Henderson. *Conceptual models: core to good design*. San Rafael, Calif., 2012. ISBN 978-160-8457-496.
10. IBM Corporation (1992). *Object-Oriented Interface Design: IBM Common User Access Guidelines*. Indianapolis IN: QUE. R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547–588, Apr. 1965.
11. Adobe Systems Inc. (2014). *About the Project*. Available: <http://phonegap.com/about/>. Last accessed 2014-03-04.
12. QCubed. (2014). *Welcome to QCubed!*. Available: <http://qcu.be/content/welcome>. Last accessed 2014-03-04.
13. K. Roebuck. *Object-relational mapping: high-impact strategies - what you need to know*. S.l.: Emereo Pty Limited, 2011. ISBN 978-174-3044-759.
14. G. Rossini. *Rapid Application Development with Qcubed*. Gianni Rossini.
15. Form Drafts [online]. [cit. 2014-03-25]. Available from: <https://github.com/qcubed/framework/wiki/Form-Drafts>.