Sabine Prezenski and Nele Russwinkel. 2014. Cognitive Modeling offers Explanations for Effects found in Usability Studies. In *Proceedings of the 2014 European Conference on Cognitive Ergonomics* (ECCE '14). ACM, New York, NY, USA, Article 36 , 4 pages. **This is a post-print authors version. The original publication is available at the ACM Digital Library via DOI=http://dx.doi.org/10.1145/2637248.2743001**

Cognitive Modeling offers Explanations for Effects found in Usability Studies.

Sabine Prezenski **Nele Russwinkel**
Dep. of cognitive Modeling in dynamic HMS,
TU Berlin Marchstr. 23,
10587 Berlin, Germany
sabine.prezenski@tu-berlin.de

ABSTRACT
Two studies evaluate the usability of two versions of an android shopping list application. ACT-R modeling approaches and empirical findings are presented. It is shown that semantic networks have a strong influence on performance and learning. Effects of version updates are discussed.

Author Keywords
Usability; cognitive modeling; ACT-R; application; mobile; semantic network

ACM Classification Keywords
User centered design; ergonomics; theory and methods design tools and techniques; human factors; human Information processing; experimental design

INTRODUCTION

These days, life without mobile applications and smart phones is hard to imagine. The market for applications is growing rapidly [1]. For an application to be successful, high usability is compulsory. Conventional usability testing is time and money consuming. We therefore ask, how can usability of applications be guaranteed without testing costs exploding? The following paper argues that cognitive modeling with ACT-R can serve as substitute for extensive usability testing. We will present results of two studies of an application to show how learning in applications proceeds, why semantic knowledge is important and also focus on version updates effects.

COGNITIVE MODELING & ACT-R

Cognitive architectures such as ACT-R [2] offer a computable platform that represents well established theories about human information processing. With cognitive architectures it is possible to simulate cognitive mechanisms and structures such as visual perception or memory retrieval. ACT-R is a hybrid architecture, which means that is has symbolic (knowledge representations such as chunks and rules called productions) and sub symbolic components (activation of chunks and utility of productions). The structure of chunks is characterized by different slots (or attributes), that can be filled with information. Category membership is represented in slots; this allows building semantic networks. Furthermore, new chunks can be learned during a task. The production system persists of rules defined by an "if" and "then" part. If the cognitive system with its modules and chunks in the buffers meet the conditions of the rule it can be selected. Then the action part is executed.

ECCE '14, September 01 - 03, 2014, Vienna, Austria
 ISBN 978-1-4503-2874-6/14/09…$15.00
DOI: http://dx.doi.org/10.1145/2637248.2743001

USABILITY

Standard ISO 9241-11 defines usability as effectiveness, efficiency and satisfaction. General ergonomic principles for the design of dialogues between humans and information system are specified in Standard ISO-9241-110, seven criteria are outlined (suitability for the task, suitability for learning, suitability for individualization, conformity with user expectations, self descriptiveness, controllability, and error tolerance).

Most usability criteria however can be assessed with quantitative user tests. Suitability for learning can be measured via comparison of several runs [3].

It is commonly agreed, that human knowledge is represented in form of a semantic network [4]. Within these net categories, associations with subcategories and retrieval of subcategories succeed best and faster when the category representations are addressed. To answer the question, what is the best design for a menu structure, we have designed two versions of an application. One version has two subcategories (memory-like, Version A) with the disadvantages of more required clicks; the other has only one level of sub-categories (Version B) and therefore requires fewer clicks.

Cognitive Modeling and Usability

Rather than user tests, cognitive models can be used to evaluate usability. User models build with the cognitive architecture ACT-R can simulate the interaction of a user with a particular task. There are two advantages of cognitive modeling over user tests; not only the effort of testing is omitted, but most importantly information about underlying cognitive processes can be uncovered. Implications from these findings can then be applied to follow up projects.

We developed a tool called "ACT-DROID"[5]. This tool enables a direct connection of the cognitive architecture with an adroid smartphone application.

PROCEDURE

The following two studies compare two slightly different versions of a shopping list application for Android [6]. The first study is designed in order to investigate, if the two versions differ on a statistical significant level. The second study is conducted in order to evaluate hypothesis derived from the modeling approach of the first study.

Application

Both versions of the application allow users to choose products out of either an alphabetically ordered list or via categorical search. The chosen products are then added to a list. Menu depth differs between the two versions: Version A has one menu level more than Version B. The first page of the application is the same for both versions: Three buttons are visible: "overview", "shops" and "my list". For both versions the user gets a list of the alphabet when selecting "overview". If you select "shops then for both versions a list of seven shops appears. For Version B, selecting one of the shops results in an alphabetical ordered list of the products available in that particular shop. For Version A, the shops each have seven subcategories. When selecting a subcategory, a list of products that can be selected, appears. For both versions, selecting "My List" from page one results in a shopping list which comprises the selected products plus information about the store in which the products are available.

Procedure

In both studies participants were asked to find certain products. In the first study participants were free to choose the pathway. In the second study participants were only allowed to use the store path. 26 student participants (twelve male and fourteen female, age_{mean}=23) took part in the first study and 17 student participants (six male and eleven female, age_{mean}= 26) took part in the second study. After receiving instructions participants were asked to select products. After selecting a product, participants were asked to return to the first page and then the next trial started. After selecting eight or nine products, participants were asked to read the shopping list (in order to assure learning of the store categories) and then the next block started, this time the same items were presented in a different sequence. After completing the second block, the version changed and the two blocks of trials were repeated. For the first study half of the participants first worked with Version B and half began with Version A. For the second study all participants first worked with Version and then switched to Version A.

RESULTS

The main dependent variable is the mean trial time for the different blocks, which is defined as the time difference starting from when the participant leaves the start page until the product is selected.

Study 1

Figure 1, shows the mean trial time and standard deviations for the different conditions.

To look at the differences between the conditions a 2x2x2 ANOVA is conducted. The following factors are considered: Factor order of the version with the two steps "first A than B" and "first B than A"; factor version (repeated measurement) with the two steps "Version A" and "Version B" and factor novelty (repeated measurement) with the two steps "new" and "expert". The ANOVA revealed a significant main effect of factor version with $F(1,24)=12.527$, $p<0.005$ and a medium effect size (partial $\eta 2=0.343$). Descriptive results indicate that Version B is overall faster than Version A. Another significant main effect was found for the factor novelty $F(1,24)=29.625$, $p<0.001$ and a medium to large effect size (partial $\eta 2=0.552$). Descriptive results show, that performance in the new conditions is slower than in the expert conditions. The interaction between version and order is also significant $F(1,24)=7.076$, $p<0.05$, with a medium effect size (partial $\eta 2=0.228$). The interaction between version, novelty and order is further significant, with $F(1,24)=13.661$, $p<0.001$ and a medium effect size (partial $\eta 2=0.363$). Our data show an overall learning effect (main effect of novelty), a version effect (Version B is overall better than Version A) and an interaction between all three factors, which we label "version update" effect.

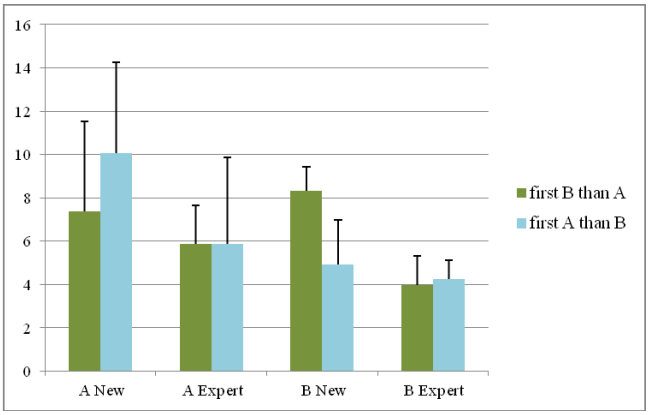

Figure 1: Mean trial time of study 1

Modeling approach of using a shopping app

The model selects products only via the shops button. It is expected that people know most of the products and categories, therefore some chunks are prepared that hold the information like "a bananas is a fruit" and "fruit can be found in a greengrocery". The app is started and the model searches for the button labeled "shop". After the correct button is found and selected, the model finds one shop after the other, reads them and tries to retrieve from declarative memory in what shop the product might be found. The model finds the correct shop and selects it. For Version B the products are presented and the model is searching through the list to find the right item. If it is found two things happen. First the product is selected, second the model builds up a chunk that the product banana in found in the greengrocery store for later use. This we will later refer to as *expectancy chunks*. If a product is not found the model has to go back and try something else. During different trials of searching for products the user model learns the menu structure and which shop holds what product. For Version A the model has to learn an extra level of menu structure. Therefore Version A requires intensified learning but benefits from less visual search after the structure and categories are learned.

Which is the better version?

Empirical: Version B is overall faster than Version A, especially for novice users, the benefit of Version B over A decreases, as block 4 for Version A and Version B (expert) show. Therefore more required clicks in Version A are probably not the reason for the benefit of Version B over A for novice users.

Modeling: The building of expectations-chunks takes longer for Version A than for Version B, because there are more interaction steps in A and therefore more encoding is required. For Version A more semantic knowledge (which shop holds which subcategory and which subcategory holds which product) is needed, the knowledge of subcategory is unnecessary for Version B.

Are there version update effects?

Empirical: A version update effect occurs when participants familiar with Version B change to Version A. We call this version update effect because we believe it occurs outside the laboratory when new versions of application are used. This can be seen in the increase of time from B first expert to A second. Nevertheless participants, who use Version A second still profit from Version B, since A second is faster than A first.

Modeling: Switching from Version B to A irritates the users because they end up with a menu they did not expect. The model has to go back and search for the back button and then learn the items that belong to the new categories, this takes time because this causes a number of additional productions to fire. But after a few trials new category-product pairs (e.g. expectancy chunks) are learned and the version update effect disappears. In the opposite case, users end up earlier with the final (more familiar) list that is already encoded in the expectancy chunks. They do not have to learn new category members and do not need to encode representations to declarative memory; therefore fewer productions have to fire and mean trial duration is low.

Does Learning occur?

Empirical: Our data shows a clear learning effect as participants become more familiar with the application, the mean trial duration decreases- there also seems to be a learning transfer from Version A to Version B.

Modeling: Production compilation is a useful ACT-R mechanism to model learning. In the beginning for every interaction step, a memory retrieval of the next processing step is required. After a few trials often used information is integrated in the productions. Trial duration decreases, since retrieval time is redundant and proceeding productions are integrated. Furthermore, retrieved expectancies can give detailed information were the next relevant button will be located. Therefore eye- and finger-movements can be prepared early and initiated more quickly with practice. Because no additional information needs to be learned when switching from Version A to B (note that Version A includes all menu-structures of Version B but has more menu depth) the above mentioned learning processes are not disturbed and learning continues.

Our model clearly indicates, that differences between Version B and Version A can be explained, through the extra encoding of category-pairs in Version A. Our model also explains performance improvement from the first run of a version (new) to the second run (expert) of the same version through category learning. In order to test if unknown category pairs are responsible for the difference between the two versions and the learning effect we conducted a second study.

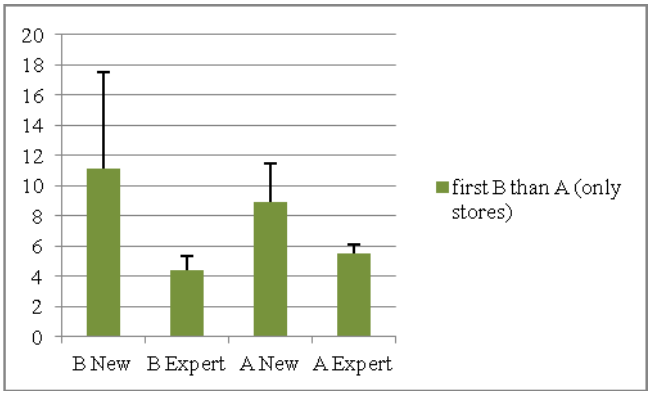

Figure 2: Mean trial time of study 2

Study 2

For the second study we predict that some word-category pairs are more unfamiliar than others, and therefore produce longer product-search times. We also predict that as category affiliation become more familiar, differences in search time between products disappear. Since we are

interested in how menu depth affects performance, participants in the second study were only allowed to find products via categories. Since the interesting version update effect takes place when participants switch from Version B (less menu depth) to A (more menu depth) all participants first worked with Version B (no subcategories) and then switched to Version A.

Results:

As figure 2 shows, there is a clear version update effect (e.g. an increase in mean trial time, when participants switch from Version B to Version A). In the new conditions strong time variations can be observed, in the expert conditions there are very little variations in time over the different product. The expert conditions indicate that some products are easier to find than others. Especially the second "clabbered milk", the third "canned pineapple" and the eighth product "gilthead" produce large search time in the novice condition. Products that didn´t produce long product search time are product number four "body wash" and product number seven "top-fermented dark beer". In post-hoc questioning the participants revealed, that they expected "clabbered milk" in the "beverage store" and not in the "deli" as it was presented in the app. They also reported, that they didn´t expect "canned pineapple" in the "corner store" and some participants weren´t aware that "gilthead" is a fish. A plausible explanation for variations between products is the fact, that some product-store pairs are more familiar for the participants than others. Higher standard deviations for the more "difficult" products in the new conditions also provide evidence for this explanation.

CONCLUSION.

Conclusion over the usability of the two versions

Both versions are suitable for users, but Version B is slightly faster than Version A. The benefit of Version B decreases as user experience increases. Shallow menu structures are more convenient for novice users. Both versions of the application are easy to learn. Version update from Version A to B has additional time cost in the beginning, whereas switching from B to A has not.

Overall, product search time is less with Version B, but if one focuses on the second study (product search only via stores), it revealed that after some practice both versions produce an almost equal products-search-time. Further note that disadvantages of search via categories arise primarily from the fact that specific category assignments are unknown to participants. In this context we would like to stress the importance of specifying user profiles when designing and evaluating an application. It is crucial to know who potential users of the application are and to figure out what category assignment is reasonable for this specific user group).

One way to redesign this app, so that it considers user profiles (e.g. the individual semantic network of the user) is to design the app customizable-so that users can move products according to their own notions of category membership.

We showed that user models can provide informed interpretations about the causes of usability e.g. differences between versions can be explained through specific learning processes; a finding that is not possible with conventional usability tests.

Outlook

The goal of our research is to develop guidelines for ACT-R modelers describing the most relevant modeling concepts for usability of applications. These guidelines will make it possible to quickly develop user models and improve and evaluate the usability of applications. As the number of new applications on the market further increase cognitive modeling provides the solution for affordable and capacious usability

ACKNOWLEDGMENTS

We thank all members of our team for support. Special thanks to Lisa Doerr and André Brandewiede.

REFERENCES

1. Koetsier, J. (2013). Google Play will hit a million apps in JuneTitle. Retrieved from http://venturebeat.com/2013/01/04/google-play-will-hit-a-million-apps-in-2013-probably-sooner-than-the-ios-app-store/
2. Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* (p. 304). New York: oxford University Press.
3. Zhang, D., & Adipat, B. (2005). Challenges, Methodologies, and Issues in the Usability Testing of Mobile Applications. *International Journal of Human-Computer Interaction*, 18(3), 293–308. doi:10.1207/s15327590ijhc1803_3
4. Collins, A. M., & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of verbal learning and verbal behavior,* 8(2), 240–248. doi:10.1016/S0022-5371(69)80069-1
5. Lindner S., Büttner P., Taenzer G., Vaupel S. & Russwinkel N. (accepted). Towards An Efficient Evaluation of the Usability of Android Apps by Cognitive Models. .In: M. Jipp, D. Soeffker, A. Kluge &, A. Wendemuth (eds.). *Kognitive Systeme III*, DuEPublico, (2014).
6. Prezenski, S. and Russwinkel, N. (2014). Combining cognitive ACT-R models with usability testing reveals users mental model while shopping with a smartphone application. *International Journal On Advances in Intelligent Systems, 7*(3-4), 700-71