Exploring the Servicelization of Mobile User Interface Evaluation

Shih-Chun Chou, Tse-Ming Tsai and Yuting Lin

The Innovative DigiTech-Enabled Applications & Services Institute (IDEAS)
Institute for Information Industry (III)
Taipei, Taiwan

ABSTRACT

Publication apps, as applications of magazines, books, journals or catalogs published for mobile devices, promise to revolutionize the way we read. A feature central to the success of publication apps is the design of a satisfied interface to link interactive components to the content, which plays a pivotal role in providing readers with good memories and increasing reader satisfaction. However, today's production platform cannot directly feedback suitable mobile user interfaces for designers or editors, as well as the review processes are so lengthy that companies spend a lot of time in waiting for the results or fixing apps based on the rejection reasons. In order to improve the quality of publication apps, we propose a systematic pre-evaluation process that assists businesses to start a pleasant design and to offer consistent mobile experiences to their specific reader communities. The current study integrates association rule mining algorithm with rejection letters in order to identify possible unsatisfied user interface. We find that publishers who submit numerous publication apps save time in both development and submission, resulting from reducing the communication gap with programmers. Therefore, this approach not only enhances efficiency for mobile application development through technology-mediated service, but also effectively generates quality assured user interface meeting mobile reader needs while remaining economically competitive.

Keywords: Association Rule, DevOps, Digital Publishing, Mobile Application, User Interface

INTRODUCTION

The awareness and ownership of new media devices (also well known as tablet and smart phone) rapidly expand, spurred by marketing campaigns from Apple, Sony, or Samsung recently as well as press coverage and word-of-mouth buzz. The behavior of reading, watching, listening and surfing on the mobile devices becomes immersive and personalized. (Bringhurst, 2013) Publication apps, as applications of magazines, books, journals or catalogs published for mobile devices, thus promise to revolutionize the way we read. A feature central to the success of publication apps is the design of a satisfied interface to link interactive components to the content, which plays a pivotal role in providing readers with good memories and increasing reader satisfaction. (Benbasat and Dexter, 1985) However, today's production platform cannot directly feedback suitable mobile user interface for designers or editors (Chen et al., 2009; Schur, et al., 2013), as well as the review processes are so lengthy that companies spend a lot of time in waiting for the results or fixing apps based on the rejection reasons. (Apple Inc., 2014) Furthermore, review groups' user experiences may vary due to their different roles and purposes. (Umanath et al., 1990) Sometimes companies find it hard to stick with numerous guidelines. It's better for apps developers or designers to understand the review principle in advance so as to prevent from consuming time and labor.

In this research, we aims at improving user interface testing processes by providing automatic pre-evaluation

services within apps production platform. The current study integrates association rule mining algorithm with rejection letters in order to identify possible unsatisfied user interface. We find that publishers who submit numerous publication apps save time in both development and submission, resulting from reducing the communication gap with programmers. As a result, designers and editors can focus on their creativities to layout or to set interactive effect with abundant content in the publication apps attempting to enhance readers' satisfaction. Moreover, this approach not only enhances efficiency for mobile application development through technology-mediated service, but also effectively generates quality assured user interface meeting mobile reader needs while remaining economically competitive.

The remainder of the paper is organized as follows. State-of-the-art defines various concepts associated with servicelization, including development and operations method for digital publishing along with technology-facilitated service. Followed by the literature review, we present a methodology for designing servicelization of mobile user interface evaluation. The last section concludes the paper and illustrates the future work.

EXPLORING STATE-OF-THE-ART TO SUPPORT SYSTEMS DESIGN

DevOps

DevOps, an abbreviation for development and operations, is a software development method that stresses communication, collaboration and integration between software developers and information technology professionals. It aims to help an organization rapidly produce software products and services. (Rajiv, 2009) Companies with very frequent releases may require a DevOps awareness or orientation. DevOps is an enterprise capability for continuous software delivery that can enable organizations to seize market opportunities, respond more rapidly to customer feedback, and balance speed, cost, quality and risk. By applying lean and agile principles across the software delivery lifecycle, DevOps helps organizations deliver a differentiated and engaging customer experience, achieve quicker time to value, and gain increased capacity to innovate. (Roche, 2013)

Nowadays, more and more mobile apps companies tend to apply DevOps because they not only need to update their apps very often due to the problems of bug fixed or operating system updated, but also develop various apps as much as they can in order to get more public attention in this app economy world. This mechanism aids in release management for a company by standardizing development environments. In particular, the responsibilities for app quality assurance teams consist largely of mimicking customers' usage patterns and minimizing the discovery of unpleasant surprises. With adopting DevOps practices in quality assurance, apps review processes can be more streamlined and efficient. In this study, we regard user interface evaluation as events that can be more easily tracked as well as more flexibly managed, driving automation in detecting unsatisfied interface - without needing to enter everything manually at the command-line. The designers act as apps developers to edit their own publication apps and also act as apps operators who can test the apps from end users' point of view. Our goal is to maximize the predictability, efficiency and maintainability of operational processes in designing publication apps. This objective is very often supported by automation.

Technology-Facilitated Service

Due to the lack of professional programming skill, current publishing industry relies on expensive software packages to produce e-publications for mobile devices. (Bringhurst, 2013) This customized approach discourages small and medium publishers from publishing electronically. Therefore, an automation system to allow small publishers access to e-publishing is developed. This content production platform integrates with visual design editing and render engine automatically previews interactive applications on various mobile devices. Anyone with no programmer skill is able to produce interactive apps cost-effectively. With tools, editors and designers in publishers of any size can focus on content instead of on programming. As shown in Figure 1, the platform process for editors or designers to produce apps starts in collecting material such as text, images or videos for editing the content layout as well as previewing the designed effects to ensure the packaging results passing apps official review. We consider this platform to be a technology-facilitated service that assisting non-programmers, especially who are from small and medium publishers, to rent this automatic publishing service to spread their own publication apps. Based on this content production platform, mobile user interface evaluation mechanism can be added to enhance the service value. Users can further utilize pre-evaluation module in attempt to test whether their design are

followed the review guidelines.

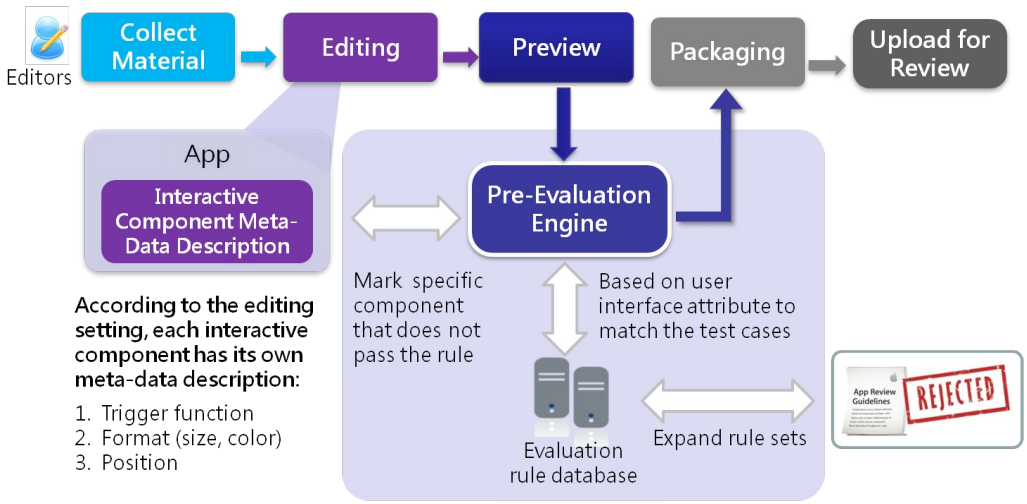

Figure 1. Service overview of mobile user interface evaluation

In this study, we propose a rule-based pre-evaluation engine within the preview process so that this evaluation can facilitate the procedure of user interface testing before wrapping the finalized content up. The engine collects rule sets from review guidelines and rejection letters to establish rule database. When users start the evaluating service by either uploading a designed folio or outputting from previous editing step, the pre-evaluation engine will match the user interface attribute to the test cases in rule database attempting to mark specific component that does not pass the rule. Each component that set from the users has its own metadata description which can be match with our predefined evaluation rules.

METHODOLOGY FOR DESIGNING THE SERVICELIZATION OF MOBILE USER INTERFACE EVALUATION

Interface design is a significant quality attribute of mobile phones and thus user interface evaluation is an essential activity for securing a highly usable mobile phone, which should be conducted during all the phases of design life cycle. Various evaluation methods have been developed and can be classified into three types: testing, inquiry, and inspection (Heo et al., 2009). Testing employs representative users on typical tasks using a system or a prototype and then evaluates how user interface supports the users to do their tasks. Typical methods include co-discovery learning, question-asking protocol and shadowing method. Inquiry method talks to users, observes how they use a system in real work settings, and let them answer questions in order to understand users' feeling about the system and their information needs. Field observation, focus groups, and questionnaire survey are typical inquiry methods. In inspection, experts examine usability-related aspects analytically. Typical methods are cognitive walkthrough and heuristic evaluation which are formed by complicated interaction among a mobile phone user interface, user and task characteristics, and other environmental factors.

In order to improve the quality of publication apps, we propose a systematic pre-evaluation process that assists businesses to start a pleasant design and to offer consistent mobile experiences to their specific user communities. The current study integrates association rule mining into *user interface rule extraction module* to alleviate problems of frequent update and numerous guidelines. Then, the key terms will be extracted from the guideline set, and the initial representation of all reviews is further enriched by using hypernyms in order to exploit the semantic relations between terms. (Chen et al., 2010) An association rule mining algorithm for texts in *rule transformation module* is employed to discover a set of highly-related frequent rules, which contain possible rejection principles for external knowledge experts to implement test codes. Finally, each page in mobile application is executed by test codes within *app page evaluation module* to find out which user interface elements are failed to pass the test cases. Every rejection letter and user feedback can be also input into *user interface rule extraction module* to expand our rule sets.

See below figure.

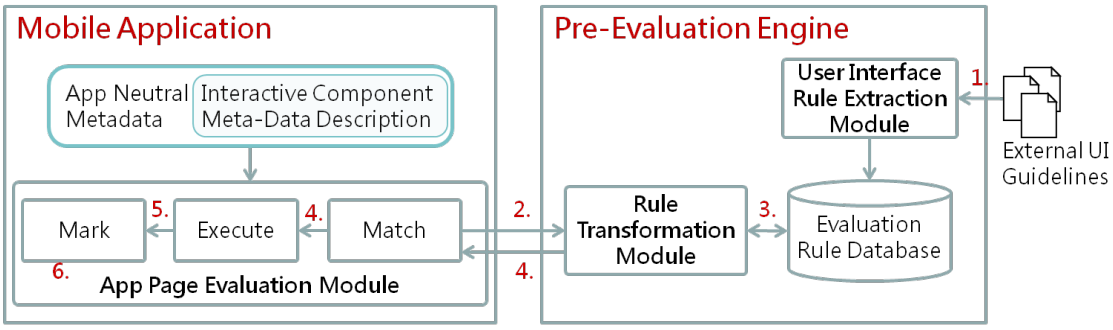

Figure 2. System modules diagram

User Interface Rule Extraction Module

In this module, when receiving a set of external documents including review guideline, human interface guideline, design guides, this module will extract and select the key term set which is equivalent to user interface attribute. There are two stages in the first module, namely Key Term Extraction and Key Term Selection, for reducing the dimensionality of the source document set. The first process for Key Term Extraction is that each guideline is broken into sentences. Then, terms in each sentence are extracted as features. In this paper, a term is regarded as the stem of a single word. Then, the terms appeared in a predefined stop word list are removed. Thirdly, remained terms are converted to their base forms by stemming. The terms with the same stem are combined for frequency counting. Finally, the frequency of each term in each document is recorded.

The second stage is Key Term Selection. We understand that terms of low frequencies are supposed as noise and useless for identifying the appropriate cluster. Thus, we apply the frequency method to choose the key terms for the document set. A term will be discarded if its weight is less than a threshold. Subsequently, this stage will base on the usage of WordNet for generating a richer document representation of the given document set. As the relationships of relevant terms have been predefined in WordNet, in this module, we intend to use the hypernyms provided by WordNet as useful features for document clustering. After key terms are extracted from the document set, they can be organized based on the hierarchical (IS–A) relationship of WordNet to construct term trees. A term tree is constructed by matching a key term in WordNet and then navigating upwards for five levels of hypernyms. Eventually, all term trees can be regarded as a term forest for the document set.

Rule Transformation Module

After the above processes, documents or review rules are converted into structured term vectors. Then, the fuzzy data mining algorithm is executed to generate fuzzy frequent item sets and output a candidate clusters set. In the following, we define the membership functions and present our fuzzy association rule mining algorithm for texts.

Firstly, each pair (term and frequency) of a document can be transformed into a fuzzy set with its frequency being represented by three fuzzy regions, namely low, middle, and high, to depict its grade of membership within corresponding documents. Each fuzzy value has a corresponding membership function to convert the key term frequency into a range value. Secondly, to generate the target cluster set for a document set, a candidate cluster set will be generated after the mining process. A candidate cluster is a two-tuple including those documents which contain all the key terms and a fuzzy frequent item set. This module generates fuzzy frequent item sets based on predefined membership functions and the minimum support value, from a large textual document set, and obtains a candidate cluster set according to the minimum confidence value. Since each discovered fuzzy frequent item set has an associated fuzzy count value, it can be regarded as the degree of importance that the item set contributes to the document set. Two confidence values of a rule pair is used to measure the strength of association among the key terms of the fuzzy frequent item sets.

App Page Evaluation Module

The objective of the last module is to assign each document to multiple clusters. For assigning documents to the target clusters, each candidate cluster with fuzzy frequent item set is considered in the clustering process. Item set

will be regarded as a reference point for generating a target cluster. In order to represent the degree of importance of a document in a candidate cluster, a document-cluster matrix will be constructed to calculate the similarity of terms. Finally, to avoid low clustering accuracy, the inter-cluster similarity between two target clusters is calculated to merge the small target cluster into the similar target cluster. In our current study, we classify the user interface elements into 13 types: activity, activity view controller, collection view, container view controller, image view, map view, page view controller, popover, scroll view, split view, table view, text view, and web view. Every type has predefined test codes written by knowledge engineers. Besides, these test rules separate the behavior in plain text and the step definition in order to facilitate the evaluation. Once the page function in the apps exists, then we only need to modify the test cases. Conversely, when the page function in the apps changes, then we only need to modify the scenarios.

DEMONSTRATION SCENARIO

In this section, we are going to introduce how the proposed service system processes in order to further demonstrate the servicelization of mobile user interface evaluation. When designers already produce an app folios from our predefined content production platform, every app page will have interactive component metadata description file which includes interaction correlation, source metadata, and position metadata. The interaction correlation specifies the relationship among components. This correlation can be touch event for image components to pop-out or scroll events for text components to display more words. As for the source metadata, it includes each component's specific information which is recorded according to user interface element attributes in the guidelines. Position metadata describes the arrangement of interactive components. These metadata files were all wrapped up in an app folio in order to be subsequently parsed for the evaluation.

As shown in below figure, the first step is transforming the guidelines and rejection letters into evaluation rules. This transformation will extract user interface related words and sentences for knowledge engineers to establish corresponding detecting codes. Followed by the content views retrieval, interactive components metadata description will match each rule to see if certain component has specific rules in the database. The aim of this action is to ensure every component is well-checked in attempt to find out possible problematic user interface components. Thirdly, when the components match certain rules, the components will be examined based on test cases to execute corresponding detecting codes. For example, in this flowchart scenario, the app page includes six button components that will be checked for size, color and position. Lastly, components that are problematic according to the rules will be marked for reminding the designers. Take the button size for instance; previous rejection letter has mentioned that the button size should be at least 44pt so that the buttons will be highlighted for being too small. To sum up, this pre-evaluation process assists the designers to preliminarily filter problematic user interface from rejection rules.

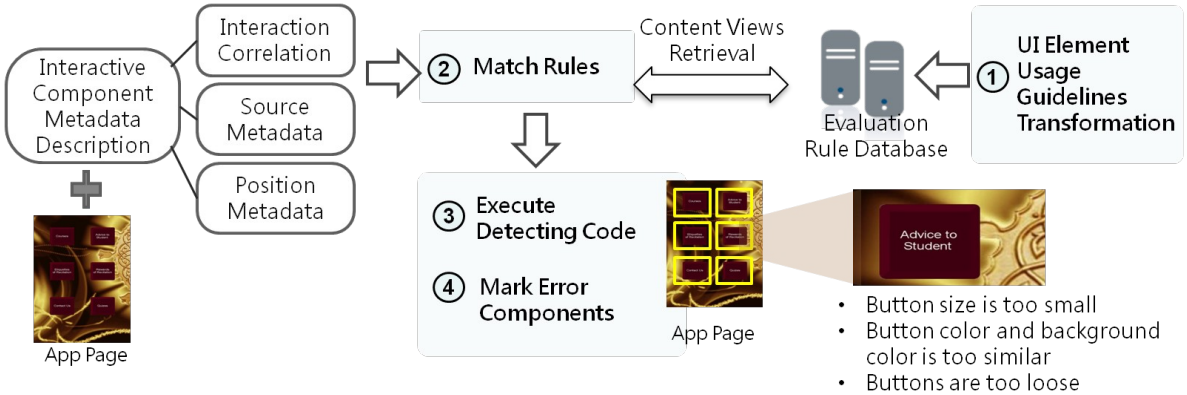

Figure 3. Flowchart of mobile application pre-evaluation

In this current study, we firstly conducted a convenience subject in order to offers a unique lens through our pre-evaluation methodology, and then in the next phase of our study we will collect more samples to evaluate the model.

The samples here refer to our app folios that are uploaded to the app store for reviewing.

CONCLUSIONS

Grounded on our research, we firstly conduct a convenience subject in order to identify attitude towards using the publication apps will be more positive when the information format matches the pre-evaluation results. We find that publishers who submit numerous publication apps save time in both development and submission that resulted from reducing the communication gap with programmers. The current study provides a mechanism for editors to sketch a blueprint or a solid configuration with definite expression toward user experience knowledge. The ratings and comments from pre-evaluated publications will be further collected to guarantee the satisfied user interface. Therefore, this approach not only enhances efficiency for mobile application development through technology-mediated service, but also effectively generates quality assured user interface meeting mobile reader needs while remaining economically competitive.

ACKNOWLEDGMENT

This study is conducted under the "Digital Convergence Key Technology and System Develop Project" of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China.

REFERENCES

Apple Inc. (2014), "*App Review*", Developer Website: https://developer.apple.com/app-store/review/

Benbasat, I., Dexter, A. S. (1985), "*An Experimental Evaluation of Graphical and Color-Enhanced Information Presentation*", Management Science, Vol. 31, pp. 1348-1364.

Bringhurst, B. (2013), "*Design Decisions for Digital Publishing APPs*", InDesign Docs Website: http://blogs.adobe.com/indesigndocs/

Chen, C. L. Tseng, Frank S.C. & Liang, T. (2010), "*An Integration of WordNet and Fuzzy Association Rule Mining for Multi-label Document Clustering*", Data & Knowledge Engineering, Vol. 69, No. 11, pp. 1208–1226.

Chen, R., Rose, A., & Bederson, B. B. (2009), "*How People Read Books Online: Mining and Visualizing Web Logs for Use Information*", European Conference on Research and Advanced Technology for Digital Libraries, pp. 364-369.

Heo, J., Hamb, D. H., Parkc, S., Songc, C., & Yoon, W. C. (2009). "*A Framework for Evaluating the Usability of Mobile Phones based on Multi-level, Hierarchical Model of Usability Factors*", Interacting with Computers, Vol. 21, No. 4, pp. 263–275.

Rajiv, P. (2009), "*DevOps*", Organizing a Technology Department Website: http://www.rajiv.com/blog/2009/03/17/technology-department/

Roche, J. (2013), "*Adopting DevOps Practices in Quality Assurance*", Communications of the ACM, Vol. 56, No. 11, pp. 38-43.

Schur, M., Roth, A. & Zeller, A. (2013), "*Mining Behavior Models from Enterprise Web Application*", Proceedings of Joint Meeting on Foundations of Software Engineering, Saint Petersburg, Russia, pp. 422-432.

Umanath, N.S., Scamell, R.W. & Das, S.R. (1990), "*An Examination of Two Screen/Report Design Variables in An Information Recall Context*", Decision Sciences, Vol. 21, pp. 216-240.