

Optimizing User Interaction for Mobile Web Browsing

Dong Zhou, Ajay Chander, Hiroshi Inamura
DOCOMO USA Labs
3240 Hillview Ave., Palo Alto, CA 94304, USA
{zhou,chander,Inamura}@docomolabs-usa.com

ABSTRACT

The small form-factor of mobile handsets and the longer, variable latency of cellular networks negatively affect user experience in mobile web related activities. In this paper we describe the design and prototype implementation of a framework for improving mobile web interaction based on monitored interaction history and runtime interaction context. Our framework predicts future interaction sequences and optimizes predicted user interactions with navigation shortcuts and automatic text copying and form-filling

Categories and Subject Descriptors

D.2.2 [Design Tools and Techniques]: User interfaces

General Terms

Design, Experimentation, Human Factors

1. INTRODUCTION

The small form factor of mobile devices makes it much harder to interact with: the small screen makes it difficult to read what's displayed and to click/tap at the right places; the lack of full-sized keyboard and stable arm-support makes it difficult to input text; and the lack of multi-window and mechanisms for easy copy/paste makes it difficult to copy data from one application to another.

Furthermore, despite recent progress in technology, cellular network connections are still weaker and less stable than their fixed-line counterparts. The result of this is longer and less predictable network latency and user wait time, and consequently worse user experience, especially for web sessions involving sequences of user actions and Internet requests/responses.

Various approaches have been proposed and implemented to improve mobile user experience for web-related activities, including slimming down web sites to customize for mobile devices intending to reduce page sizes and downloading time [6], as well as reorganizing the information hierarchy of web sites to narrow navigation breadth at the cost of deeper navigation depth. [2][3] There are also research work and commercial applications on automatic form filling for reducing user typing. [1] Such approaches are usually more effective when they are used in predefined tasks, and they typically use dictionary-based

approach in filling input fields.[6]

While above mentioned approaches help improving mobile user experience, they are still limited to varying extents in: 1) considering a sequence of interaction instead of a single step in optimization; 2) minimizing user involvement and maximizing context sensitivity; 3) adapting to minor changes in user tasks and changes in user interaction context; 4) having a comprehensive optimization target that reduces both user input and user wait time.

We propose the design and describe the prototype implementation of an initial framework towards a mobile user interaction optimization system that requires minimal explicit user involvement; is sensitive to users' interaction contexts; looks beyond users' immediate next step; targets costs in both user input and user wait time; and can, to some extent, adapt to changing user tasks.

2. FRAMEWORK

Our framework is composed of three general steps or conceptual modules: monitoring, rule generation, and rule matching and application. The monitoring module captures and logs users' interaction history with their browsers. The rule generation module creates interaction prediction rules from such interaction history, and generates optimized forms for predicted user interaction sequences. The rule matching and application module matches a user's current interaction context against generated rules and, if there are any matching rules, non-intrusively communicates the optimized predicted interaction sequences to the user.

Interaction history is defined as set of *interaction state* transition sequences, where the interaction state is defined by the following variables: 1) the page that is currently open in the browser; 2) the highlighted text on the current page; 3) content of the clipboard, and 4) form inputs made to the current page. An *Interaction event* changes the value of one of the above interaction state variables. Such event can be one of Page-Load event, URL-Input event, Text-Highlight event, Text-Copy event, Text-Paste event, or Form-Input event.

Captured interaction event sequences are broken into Trails. A *Trail* is a segment of user interaction sequence that reflects a burst of user interaction activities. A long sequence of user interaction with a device can be divided into multiple Trails at abnormally long time gaps along the sequence. Such long time gap is usually resulted from the user reading a page in detail, or from the user shifting attention away from the browser.

A server in the network collects and maintains a database of trails. The server also hosts the prediction rule generation module that

runs periodically and/or after a Trail database has been updated. A prediction rule created by the module takes the following form:

`<Trail_Seg> :- <Optimized_Trail_Tail>`

where `<Trail_Seg>` is one of the segments of one of the Trails in the database, and `<Optimized_Trail_Tail>` is the tail of a candidate Trail after optimization.

Candidate trails for optimization are selected through a recursive algorithm which takes a Trail segment (TS), and either goes to the next level of recursion or returns a pair (TS, MS), where MS is a set of Trails matching TS whose size could be zero. The purpose of the algorithm is to find likely future interaction sequences (i.e., MS) given an interaction context (TS). A weight function that takes into account of information such as visit count and the cost of each interaction event in the tail is used to prioritize matching trails.

The tail of each matching Trail in the MS of the (TS, MS) pair is then optimized to reduce user input, as well as to reduce user wait time for page loading. Specifically, for a Form-Input event we determine if such input is resulted from copy/paste operations by tracking back along the Trail. If it is, we associate the value(s) of the text string(s) copy/pasted with the Form-Input event. For each Page-Load event, we check whether the URL or other request parameters of the page to be loaded is associated with one or more of preceding Form-Input events (for example, the text a user typed in a search box is associated with the search results page). We then link the Page-Load event with its associated Form-Input event(s). If a Page-Load event has no Form-Input event associated with it, we create a new Page-Prefetch event for it and add the new event to the start of the tail.

The Rule Matching and Applying module deployed on a mobile device monitors the user's interaction with the browser, and similarly breaks interaction sequences into Trails. The most recent Trail (which haven't terminated) is used as the current interaction context (CIC). The CIC is used in real-time to match against prediction rules, and the optimized Trail tail of each matching prediction rule is non-intrusively presented to the user.

3. IMPLEMENTATION

We have implemented a proof-of-concept prototype of our framework as a Mozilla extension for the Firefox web browser. In this prototype, we only retrieve the rule with the highest interaction cost. We then optimize the events in the body of the rule, and use XUL elements to represent such optimized interaction. We then display such XUL elements on the status bar of the browser (see Figure 1).

Specifically, for each Page-Load event, we use a clickable label, with its name set to the abbreviated title of the page to be loaded. Clicking the panel will result in loading the page associate with the panel in the browser, without needing to input the URL of the page in browser URL bar. For each Form-Input event, we use an editable textbox, with its value initialized to the value of the event. If the Form-Input event is associated with a later Page-Load event, we modify the hyperlink of the clickable panel for the Page-Load event whenever the value of the textbox changes. In addition, when the Page-Load event is immediately after the associated Form-Input event, we simply use "go" as the label of the clickable label for the Page-Load event (see last widget in Figure 1).

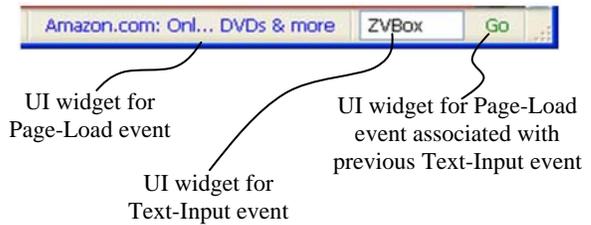


Figure 1. Widgets for optimized rule body displayed on browser status bar

Such optimization widgets serve as reminder to the user that the system predicted the next likely steps of the user, and that the user can take advantage of such widgets to make such interactions easier to perform. Note that the user can choose to activate any of these widgets: activating a widget later in the line essentially lets the user to skip previous interaction steps, or he can opt to ignore such optimization widgets. Since such widgets are displayed on the status bar of the browser and do not obtrude with user interaction, we believe that it generally does not pose undue negative effects on user experience.

4. CONCLUSION AND FUTURE WORK

We described our framework for enhancing user experience through reducing user input and wait time for page loading. Our framework depends on collecting users' interaction history, from which, based on current interaction context, we predict sequences of likely next interaction steps. We then find out likely data input for each step, pre-fetch when necessary pages that can be pre-determined and are expected to be used in one of the successive steps, and provide short-cuts for skipping unnecessary interactions. Our work can thus potentially both reduce the amount of user input and shorten user wait time.

REFERENCES

- [1] E. Rukzio, A. Schmidt and H. Hußmann. Privacy-enhanced intelligent automatic form filling for context-aware services on mobile devices. *Artificial Intelligence in Mobile Systems 2004 (AIMS 2004)*.
- [2] A. Geven, R. Sefelin and M. Tscheligi: Depth and breadth away from the desktop: the optimal information hierarchy for mobile use. *Proceedings of MobileHCI 2006*.
- [3] G. Hattori, K. Hoashi, K. Matsumoto, and F. Sugaya. Robust web page segmentation for mobile terminal using content-distances and page layout information. *Proceedings of WWW 2006*.
- [4] C. C. Yang and F. L. Wang. Fractal summarization for mobile devices to access large documents on the web. *Proceedings of WWW 2006*.
- [5] G. Leshed, E. Haber, T. Matthews and Tessa Lau. CoScripter: Automating & Sharing How-To Knowledge in the Enterprise. *Proceedings of CHI 2008*.
- [6] J. Nichols and T. Lau. Mobilization by Demonstration: Using Traces to Re-author Existing Web Sites. *Proceedings of IUI'2008*