

# Context-Based Mobile User Interface

Mao Zheng, Sihan Cheng, Qian Xu

Department of Computer Science, University of Wisconsin-La Crosse, La Crosse, WI, USA  
Email: mzheng@uwlax.edu

Received 13 May 2016; accepted 16 July 2016; published 19 July 2016

Copyright © 2016 by authors and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY).  
<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

Context awareness is increasingly gaining applicability in interactive ubiquitous mobile computing systems. Each context-aware application has its own set of behaviors to react to context modifications. Hence, every software engineer needs to clearly understand the goal of the development and to categorize the context in the application. We incorporate context-based modifications into the appearance or the behavior of the interface, either at the design time or at the run time. In this paper, we present application behavior adaption to the context modification via a context-based user interface in a mobile application. We are interested in a context-based user interface in a mobile device that is automatically adapted based on the context information. We use the adaption tree, named in our methodology, to represent the adaption of mobile device user interface to various context information. The context includes the user's domain information and dynamic environment changes. Each path in the adaption tree, from the root to the leaf, presents an adaption rule. An e-commerce application is chosen to illustrate our approach. This mobile application was developed based on the adaption tree in the Android platform. The automatic adaption to the context information has enhanced human-computer interactions.

## Keywords

Pervasive Computing, Context-Aware Systems, Mobile User Interface, Adaption Tree

---

## 1. Introduction

Our world gets more connected every day. These connections are driven in part by the changing market of smartphones and tablets. Pervasive computing environments are fast becoming a reality. The term “pervasive”, introduced first by Weiser [1], refers to the seamless integration of devices into the user's everyday life. One field in the wide range of pervasive computing is the so-called context-aware system. Context-aware systems are able to adapt their operations to the current context without an explicit user intervention and thus aim at increasing usability and effectiveness by taking environmental context into account.

We are interested in a context-based user interface in a mobile device: the mobile user interface will be automatically adapted based on the context information. The user interface can include many features such as font, sound level and data entry. Every feature has some variables. For example, for the data entry, it has typing, voice and tapping. From the designer's perspective, the adaptability of these features is planned either at the design time or during the runtime. We use the adaption tree, named in our methodology, to represent the adaption of mobile device user interface to various context information. The context includes the user's domain information and dynamic environment changes. Each path in the adaption tree, from the root to the leaf, presents an adaption rule. An e-commerce application is chosen to illustrate our approach. This mobile application was developed based on the adaption tree in the Android platform. The automatic adaption to the context information has enhanced human-computer interactions.

With traditional e-commerce applications, the user can browse the products, select a product and view the details. In the purchase process, the user will add the product to the shopping cart, enter or select payment options and enter a shipping address. From the application interface perspective, the inputs to the application are mainly through the user's tapping, typing and clicking. The outputs of the application are in the forms of text, picture and video. In the context-based mobile e-commerce application, the mobile application's input and output have additional forms: voice input and sound output. We will discuss the categorization of context information in detail later on.

There are two major platforms in the mobile device community: iOS and Android. This project chose Android development mainly for the reason of its openness. In addition, all the tools in the Android development are free and no special hardware is required.

This paper is organized as follows: in Section 2, we describe our context-based mobile application, E-commerce system. Section 3 presents the rule-based approach and the adaption tree used in our research described in detail. Section 4 discusses the design of the context-based E-commerce application. Section 5 shows the implementation details of the application. In Section 6, we discuss the testing conducted in our application. In Section 7, we compare how our views are similar to those of others and how they are different. Section 8 concludes this research work and outlines the contributions.

## 2. Context-Based E-Commerce Application

With traditional e-commerce applications, the user can browse the products, select a product and view the details. In the purchase process, the user will add the product to the shopping cart, enter or select payment options and enter a shipping address. From the application interface perspective, the inputs to the application are mainly through the user's tapping, typing and clicking. The outputs of the application are in the forms of text, picture and video.

In our context-based mobile e-commerce application, the user interface will automatically adapt to the context information to improve the usability. We categorized the context information into two categories as shown in **Table 1**. We utilize the mobile device's sensors to collect physical context information. The logical information is gathered through the user's registration process. In addition, the mobile application's input and output have additional forms: voice input and sound output.

Note: VIP users are those who have made more than 50 orders within three months, or users which purchased the membership to the system.

Some example behaviors of our context-based mobile e-commerce application are listed below:

- 1) The user can search a product by simply talking to the device, or saying "check out" to enter the purchase stage.
- 2) If the user is using the app outdoors on a bright day, either the device will automatically adjust the screen brightness or sound out the product description.

**Table 1.** Context information categorization.

Physical context	Battery level, light, noise level, Wi-Fi, network speed
Logical context	User profile (age, gender, preferred input/output for the application, first time using the app or not) user's category (VIP or non-VIP)

3) If the device is currently running low on battery power or the device is not connected to a Wi-Fi signal, the app will display the product description in text format instead of picture or video forms.

### 3. Adaption Tree

Our work depends on the internal sensors of a mobile device, the user profile and the adaption of the mobile user interface features for both entering and accessing data. The key point of the approach is to capture and represent the knowledge required for the mobile user interface to self-adapt at run time or to implement the adaption at design time. The rule-based approach representation is what we are proposing. **Figure 1** shows our proposed approach.

Specifically in our approach, we are using the adaption tree to describe and represent the adaption rules in the system. The adaption tree is a graph that consists of nodes and edges. Each node represents a single or compound condition, and each edge represents the control flow. A path in the adaption tree is the sequence of edges starting from the root node to a leaf node. Each path represents an adaption rule. In the adaption tree, the priority of the conditions is shown as the position of the nodes in the tree. The higher of the node's position means the more important the condition. The system will check this condition first before moving to nodes found lower in the tree.

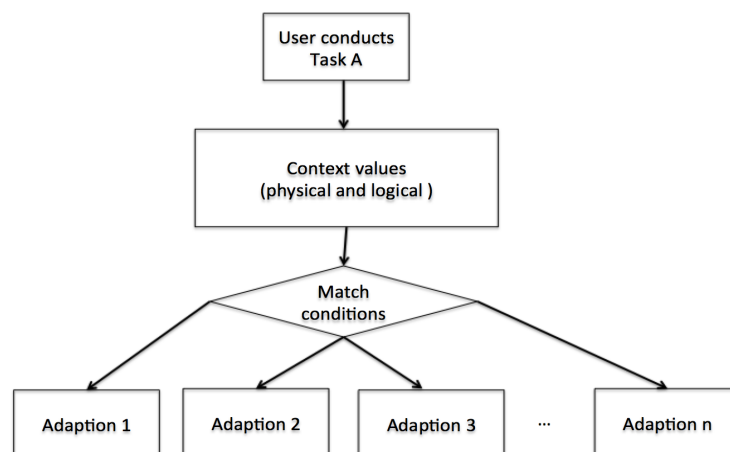
**Figure 2** is the adaption tree for our context-aware e-commerce application. The symbols used in the adaption tree are explained in detail in **Table 2**. For example, if a user is a VIP user, he/she will have the option to change his/her user interface theme (f1). His/her screen will show a VIP account interface (f2) with many pictures of the products that are available for purchase (g1) for the user to browse. Sample screen shots are shown in **Figure 3** and **Figure 4**. In another scenario, if the device's battery level is high, but the device is not connected to a Wi-Fi signal, the product's video will not be presented (a2) regardless the speed of the network. If the network speed is low, the picture will not be presented (j2), otherwise the picture will be presented (j1).

Each path in the adaption tree, from the root to the leaf represents how the app will automatically adapt to context information. However, the user is able to manually override the adaption: set the video, picture, sound or brightness according to his/her preference or for his/her special request/need at the particular time.

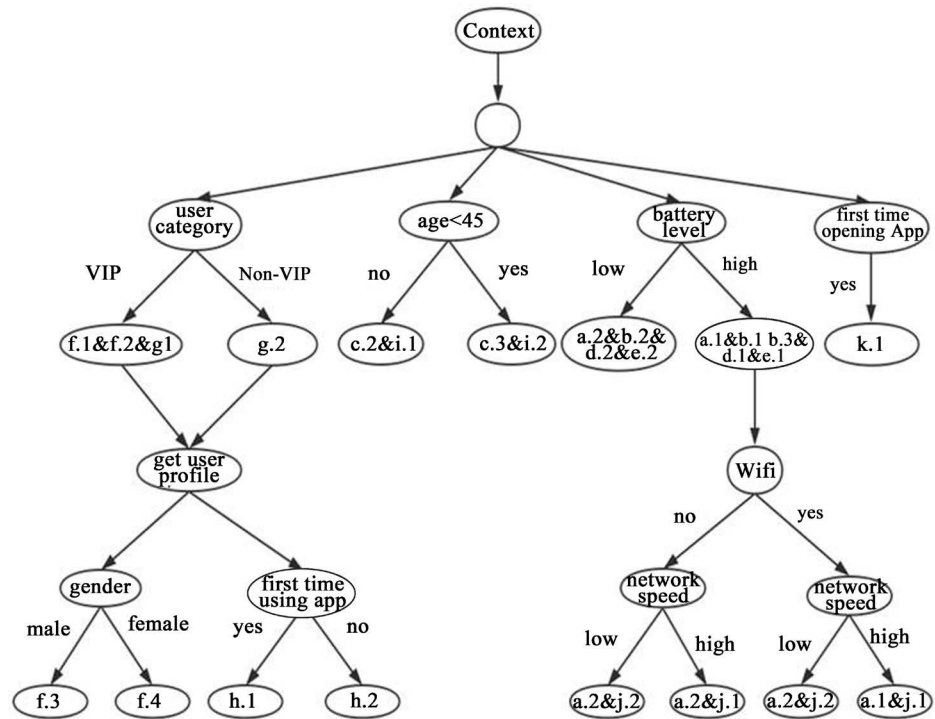
### 4. Architecture Design

In our E-commerce application, we used Mobile Backend as a Service model (MBaaS), also known as "backend as a service" (BaaS). It is a model for providing web and mobile app developers with a way to link their applications to backendcloud storage. These services are provided via the use of custom software development kits (SDKs) and application programming interfaces (APIs) [12]. APIs provided by backend applications include features such as user management, push notifications, and integration with social networking services. **Figure 5** is the architecture model of BaaS.

In our application, we chose the cloud database provided by Bmob company. This company is the first cloud



**Figure 1.** Rule-based approach.



**Figure 2.** The adaption tree for context-aware e-commerce application.

**Table 2.** The legend of the symbols used in the adaption tree.

GUI features	Action
a. Video	1. Videos will be present 2. Video will not be present
b. Media sound	1. Adjust the sound level to “sound on” 2. Adjust the sound to “sound off” 3. Adjust the sound to auto adjusted
c. Font	1. Adjust the font to “medium” 2. Adjust the font to “big” 3. Adjust the font to “small”
d. Brightness	1. Adjust the brightness level to user preference 2. Adjust the Brightness level to auto adjusted
e. Voice input	1. Enabled 2. Not enabled
f. Background theme	1. Optional theme(Blue, Red) 2. VIP account interface 3. Unchangeable color—grey 4. Unchangeable color—pink
g. Homepage	1. Picture style 2. Plain text style (detailed classification)
h. Tutorial	1. Display 2. Not display
i. Item description	1. Sound 2. No sound
j. Picture	1. Present picture 2. Not present picture
k. Welcome page	1. Present welcome page when opening the app
l. Voice command	1. Enabled 2. Not enabled

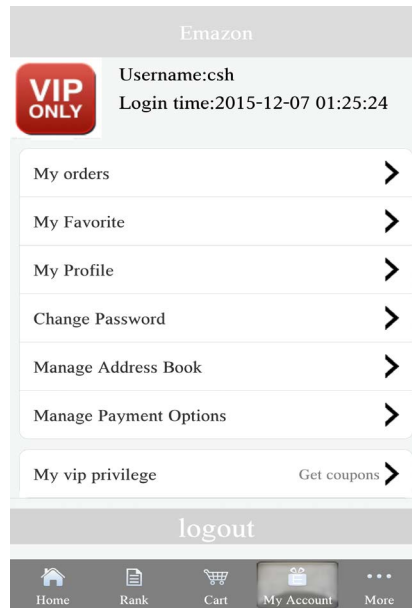


Figure 3. VIP user.

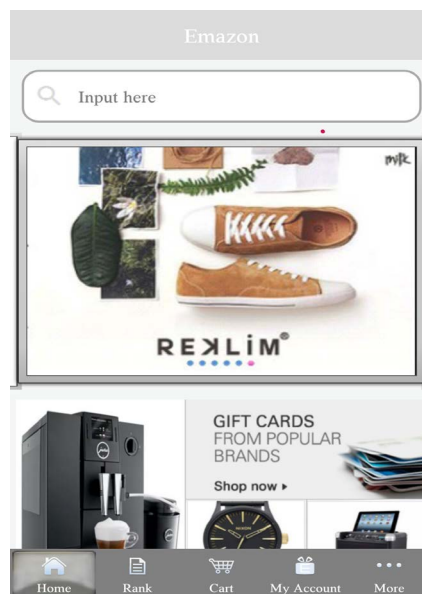


Figure 4. Picture style for VIP user.

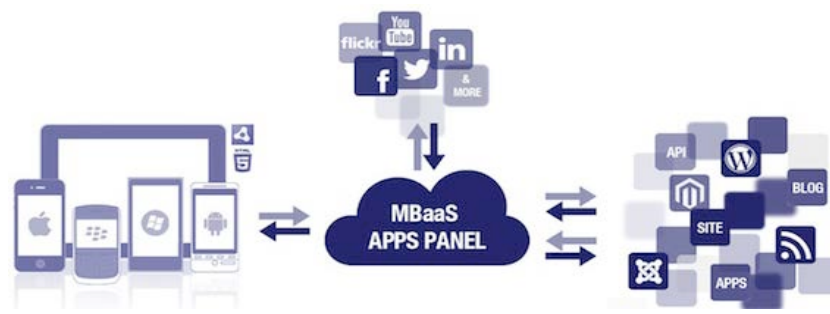


Figure 5. Architecture of BaaS.

storage company which focuses on mobile applications cloud in China. Bomb cloud offers free SDK for users to download and plug in the application. Like most of the BaaS, Bomb cloud provides features such as user-management, data-storage, queries, search, etc. Essentially, it saves developers copious hours of time. We do not need to build our own server and virtually no backend infrastructure is needed to run our app. Instead, we used BaaS as our backend server.

## 5. Implementation

The implementation strictly followed the adaption tree. Below is the sample code segment to check if the video will be present or not. The app will first check to determine if the user is able to watch a video in a given situation. The user's preference will override the context adaption. If not, the context adaption will start to check to determine if the mobile device is getting a Wi-Fi signal, the network speed, etc.

```

public boolean getVideoSetting() {
    if (!video) // if video is closed
        return false;
    if (getBatteryLevel() <20) { // if battery level <20 %
        // video is open, not present video when low battery
        if (video && !lowb_video) {
            return false;
        }

        // video is open, present video when low battery
        if (video && lowb_video) {

            // get the wifi status
            String wifiStatus = getNetworkStatus();

            // if video is not present under 3G/4G mode , return false
            if (!data_video) {
                if (!wifiStatus.equals("WIFI"))
                    return false;
            }

            // present video when no wifi environment, then consider the
            // network speed
            // else , present video when no wifi environment, then consider the
            // network speed, read the testing case first. If it is true, then
            // speed is 19
            int networkSpeed = 0;
            boolean testingNetwork = readBoolean(Constant.TESTING_LOWN);
            if (!testingNetwork) {
                String speed = readData("speed");
                // if no avg speed tested data found, set it to 21 kb/s
                if (speed.length() >0 && !speed.equals(""))
                    networkSpeed = Integer.parseInt(speed);
            }
            else
                networkSpeed = 21;
            } else {
                networkSpeed = 19;
            }
        }

        if (lowSpeed_video)
            return true;
        else {
            if (networkSpeed <20) {
                return false;
            } else
                return true;
        }
    }
}

```

```

    }
    return false;
  }

```

## 6. Testing

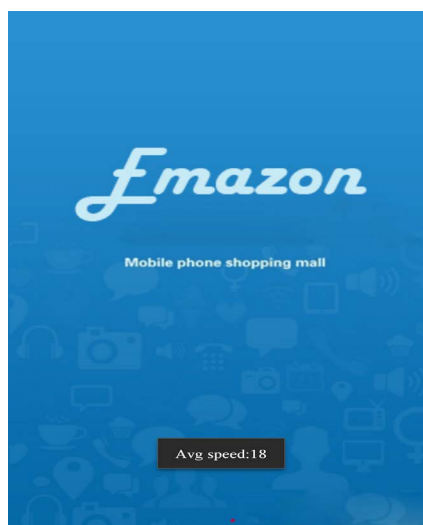
It is difficult to test a mobile device with different battery levels, network speeds, and environment noise levels in a real environment. We designed and implemented a testing simulation to check each path of the decision tree. It simulates low battery, low network speed or high environment noise cases.

In order to detect the network speed of the mobile device, the mobile device has to conduct certain network activities. Otherwise a value of zero will be shown as the network speed. However, continuous network activities will consume the user's data plan and will affect the device's performance. For these reasons, we tested network speed only when the app started. The specific test is to let the app download a small size picture and then we calculated the average network speed by dividing the picture size by the downloading time. If the network is not available, the testing activity for network speed will not be conducted. **Figure 6** is a screen shot of the average network speed testing.

## 7. Related Work

Some researchers define context as the user's physical, social, emotional or informational state, or as the subset of physical and conceptual states of interest to a particular entity [2]. The authors in [2] have presented the definition or interpretation of the term by various researchers, including Schilit and Theimer [3], Brown *et al.* [4], Ryan *et al.* [5], Dey [6], Franklin & Flaschbart [7], Ward *et al.* [8], Rodden *et al.* [9], Hull *et al.* [10], and Pascoe [11]. In Dey and Abowd [2], the authors are interested in context-aware systems, and so they focused on characterizing the term itself. In Pascoe [11], the author is interested in wearable computers, so his view of context is based on environmental parameters as perceived by the senses. Our work depends on the internal sensors of a mobile device, and the adaption of the mobile user interface features for both entering and accessing data. Our model is based on separating how context is acquired from how it is used, by adapting the mobile user interface features to the user's context.

Most of the research in this area has been based on analyzing context-aware computing that uses sensing and situational information to automate services, such as location, time, identity and action. More detailed adaption has been generally ignored, for example, input data based on context. In our research, we attempted to build the user's characteristics from both domain experience and mobile technology experience, and to collect all the context values corresponding to the user's task and then to automatically adapt the mobile user interfaces to the



**Figure 6.** Testing average network speed when the app starts.

context information.

The process of developing a context-based user interface has been explored in a number of other projects. Clercks *et al.* [12], for example, discussed various tools to support the model-based approach. Many studies have been conducted on adaption using a decision table. In [13], an approach is proposed for modeling adaptive 3D navigation in a virtual environment. In order to adapt to different types of users, they designed a system of four templates corresponding to four different types of users. Our work differs in that our adaption technique is based on composite context information that extracts values from sensors in smartphones and relates with the user's domain and mobile technology experiences. Then we developed a set of rules for the mobile user interface adaption.

## 8. Conclusions

With ubiquitous computing, users access their applications in a wide variety of environments. To cope with various and dynamic execution environments, the adaptive mobile user interface is desired to enhance human-computer interactions. This project is our attempt to address this issue. We used the rule-based approach and represented as adaption tree to describe the adaption rule for the mobile user interface based on the various context information. Our implementations strictly followed our proposed approach.

It is important to point out we are separating how context is acquired from how it is used, by adapting mobile user interface features to various context information. The user, as a composite entity, is a part of the context. Each context-aware application has its own set of behaviors to react to context modifications. Hence, every software engineer needs to clearly understand the goal of the development and categorize the context in the application.

The contributions of this research work lie in 1) considering both the user's domain and mobile technology experience in context, 2) detailed modeling inclusion on both input and output data, and 3) using the rule to present acquired knowledge in the application. The adaption built into a mobile user interface can enhance the accessibility in the e-commerce domain. The additional benefits are a) increasing usability; (for example, if the mobile user interface only supports one interaction model, such as typing or voice input/sound output, the usability of the service would be drastically decreased.); b) increasing awareness of social ethics, e.g. in a quiet room after midnight, the sound could be turned off automatically; c) improving workflow productivity because the mobile user interface is automatically adapted to the dynamic environments.

## References

- [1] Weiser, M. (1991) The Computer for the 21st Century. Scientific American. <http://dx.doi.org/10.1038/scientificamerican0991-94>
- [2] Dey, A.K. and Abowd, G.D. (2000) Towards a Better Understanding of Context and Context Awareness. ACM Press, New York.
- [3] Schilit, B. and Theimer, M. (1994) Disseminating Active Map Information to Mobile Hosts. *IEEE Network*, **8**, 22-32. <http://dx.doi.org/10.1109/65.313011>
- [4] Brown, P.J., Bovey, J.D. and Chen, X. (1997) Context-Aware Applications: From Laboratory to the Marketplace. *IEEE Personal Communications*, **4**, 58-64. <http://dx.doi.org/10.1109/98.626984>
- [5] Ryan, N., Pascoe, J. and Morse, D. (1997) Enhanced Reality Fieldwork: The Context-Aware Archaeological Assistant. In: Dingwall, L., Exon, S., Gaffney, V., Laflin, S., & van Leusen, M., Eds., *Archaeology in the Age of the Internet. CAA97. Computer Applications and Quantitative Methods in Archaeology. Proceedings of the 25th Anniversary Conference, University of Birmingham, April 1997 (BAR International Series 750)*. Archaeopress, Oxford, 269-274. Leusen, Exxon (Eds.), *Computer Applications in Archaeology*.
- [6] Dey, A.K. (1998) Context Aware Computing: The Cyber Desk Project. Technical Report SS-98-02, 51-54.
- [7] Franklin, D. and Flaschbart, J. (1998) All Gadget and No Representation Makes Jack a Dull Environment. Technical Report SS-98-02, 155-160.
- [8] Ward, A., Jones, A. and Hopper, A. (1997) A New Location Technique for the Active Office. *IEEE Personal Communication*, **4**, 42-47.
- [9] Rodden, T., Cheverst, K., Davies, K. and Dix, A. (1998) Exploiting Context in HCI Design for Mobile Systems. *Workshop on Human Computer Interaction with Mobile Devices*. Glasgow, 21-22 May 1998, 12 p.
- [10] Hull, R., Neaves, P. and Bedford-Roberts, J. (1997) Towards Situated Computing. *1st International Symposium on*



- 
- Wearable Computers*, Cambridge, 13-14 October 1997, 146-153. <http://dx.doi.org/10.1109/iswc.1997.629931>
- [11] Pascoe, J. (1998) Adding Generic Contextual Capabilities to Wearable Computers. *2nd International Symposium on Wearable Computers*, Pittsburgh, 19-20 October 1998, 92-99. <http://dx.doi.org/10.1109/iswc.1998.729534>
- [12] Clerckx, T., Winters, F. and Coninx, K. (2005) Tool Support for Designing Context-Sensitive User Interface Using a Model-Based Approach. *TAMODIA'05 Proceedings of the 4th International Workshop on Task Models and Diagrams*, Gdansk, 26-27 September 2005, 11-18. <http://dx.doi.org/10.1145/1122935.1122939>
- [13] Cheng, S.W. and Sun, S.Q. (2009) Adaptive 3D Navigation User Interface Design Based on Rough Sets. *IEEE 10th International Conference on Computer-Aided Industrial Design & Conceptual Design*, Wenzhou, 26-29 November 2009, 1935-1940.



Scientific Research Publishing

**Submit or recommend next manuscript to SCIRP and we will provide best service for you:**

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>