A Review on Operating System Management to Get User Attention

Muhammad Tayyab
Dept. of computer science
University of Lahore Gujrat campus
kapoorsb10@gmail.com

Sohail Ahmed
Dept. of computer science
University of Lahore Gujrat campus
sohail.ahmed@cs.uol.edu.pk

***Abstract*—This Mobile computer are highly engaged with our daily routine activities, from wearable displays to the smart watches than to in vehicle infotainment systems. The interactions are mostly driven by applications that are running in the background and attract users when their attention is required. Current paper will dispute that the current existing operating systems should manage the user attention as a resource. the OS should instead predict the importance and complexity of new interactions and compare the demand for attention to the attention available after accounting for the user's current activities. In contrast to permission-based models that either allow applications to interrupt the user continuously or deny all access, it will premise the operating system to initiate proper interactions at the absolute time along with the right modality we explain a design for such system along with we found key challenges.**

I. Introduction

All Mobile computing devices are increasingly engaged with our day to day activities. The ongoing world has a large effect of mobile computing devices. This trend is further growing with the rise of wearable computer devices and platforms as known Google glass and smart watches. Also, along with the huge platforms like in vehicle infotainment systems.

A large number of mobile devices helps the users while they are performing primary tasks as like walking, driving and interacting with other people along with their environment. A user is engaging in such tasks that he has very limited number of attention to spare for the application also mobile applications. Consequently, these applications may running in the background and interact with user only when the interaction is being called meaningful. It is oppositely different model of interaction as used by the traditional desktop systems. Instead that the user interacts with the application at an efficient moment. As example while on opening the application, it's now initiates the interaction e-g via an audio tone or by a smartphone notification by an in-vehicle system.

We argue that the OS should ultimately responsible for making the decision. User attention is no doubt a precious resource and the OS should manage how mobile applications allowed to consume that resource. Recently, most O systems used permission based approach for managing the user attention and requiring the attention of the user. For a moment when the user installed an application, the user has the permission to allow or deny haptic and audio notification on a smart phone. but this approach is so coarse-grained. A user may consider some notification may important than the others. Availability of attention differs. the operating system is uniquely suited with the user attention.

we require access to raw sensor data and sensitive information such as calendars by Determining the right time to interrupt the user requires understanding the user's current activities,

Mobile application should be responsible for notifying the operating system when it wishes to interact with the user; they should have to express the possible series/modes of interaction and it assigns the local importance. The operating system should convert the local importance to globally scaled and quantify the expected attention that a user need to devote to0 the interaction. The supply and demand for the attention may be divided into distinct elements as such video, audio and also cognitive attention, this division enables us for the possibility. E-g a person is listening a music while driving on a vehicle. the operating system guesses the importance and attention demanded by the user's activities, it includes both activities external to the comp system (talking\Walking) as well as the internal activities as the user's interaction with the other applications. If the user has situation to the spare, or the priority of the notification is high enough to interrupt the user current activity, the operating system initiates the interaction. Otherwise the interaction is referred for a higher priority moment. if the mobile application present the multiple modalities of possible interactions than in this situation the operating system chooses the best one based upon its assessment of available visual, audio and haptic attention.

II. Overview

The major responsibility of the operating system is the Management of user attention as a resource, but the operating system needs input from both applications and the user in order to do a good job.

The user can provide valuable feedback about what interactions were and were not worthwhile so that the operating system can learn models that adapt the systems interruption behavior to that user's preferences. In this section, we outline interfaces that separate these concerns among the operating system, the user, and applications.

On using with the upgrade mobile system, we see that the application that are running in the background must go through the operating system to initiate the user interaction (mobile phone apps we are using today must go through system software to pop a push notification. In contrast with simply allow or deny such interactions we propose a more nuanced approach. The application provides a numeric measure of how important it believes the interaction will be to the user. This is an application specific measure it ranks the importance only relative to other interactions initiated by that application.

This application also provides the list of possible modalities of interactions (E mail application may be able to read a new mail aloud or display on a touchscreen).

For each modality, the application may optionally specify a quantitative prediction of the user attention that will be consumed during the resulting interaction. This interface requires that the system have some model for quantifying attention. While any such model will necessarily be a gross simplification, there are some properties we wish to expose. First, attention may take many forms. For instance, a driver may have the attention to listen to directions from a GPS application but not to look at the screen. This suggests that attention could be expressed as a vector over those various forms (e.g., audio, visual, haptic, cognitive, and other forms of attention). Second, paying attention can be thought of imposing a load over a given time period on each of those forms. This leads to a very useful analogy: attention management can be modeled as a scheduling problem in which each form of attention is a separate core on a heterogeneous multiprocessor. A user interaction can be modeled as imposing load on some or all of these cores.

III. CHALLENGES

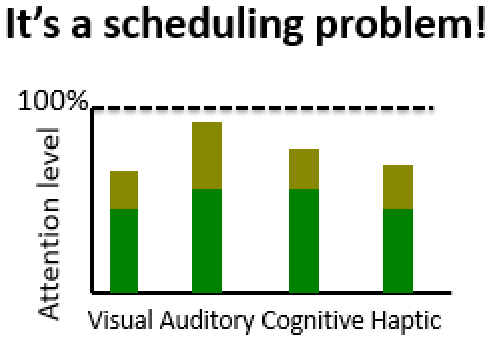
In current section, we declare three challenges and discuss possible solutions to each.

A. Predicting interaction importance

In ideally a mobile system only interrupts the user when the interrupt is more important than the current using application Further, the difference in importance should be sufficient to make up for the context-switch overhead of pausing and resuming the current activity. For example, if a user is in meeting for an advertising email from a retailer, a mobile device should not bother such user, but it should likely notify a user if there is an urgent email from his boss. The application is in the best position to assess the importance of interactions that it initiates relative to all other interactions it initiates.

B. Predicting attention demand

In addition to knowing the importance of future interaction, the operating system must know the complexities as how much amount of user interaction consumed by the interaction. As described previous the user attention importance in this paper. The approach we are following in this paper is inspired by our previous work. In this view, we observed load by observing interactions with touchscreen in vehicle system. For example, the amount of text in the screen, the presence of animation features, the placement of text, the size a, the button presses required to complete a task and so on. For all low-level analysis, we put a threshold to find whether or not an application demanded too much attention to be used while interacting or driving a vehicle. This start was appropriate for the work when considered one and only one foreground activity means Driving.



On broaden this approach by using quantities functions to figure up low level interactions to the specific form of attention like visual and audio. For example, a notification containing text demands visual interaction, and as the text size going greater it demands a high attention paid by the user. On using AMC-like tools, we can figure how user interact with the application in the respond to a notification through button presses, touchscreen events, Voice recognition, and also, we measure the kind and quantity of the output that is produced. For low level interaction measurements e-g button pressed, voice command issued, text displayed etc., the operating system can figure out the demand of interaction needs for each interaction.

An application might support multiple modes of interaction as it might read a text message aloud or the text message displayed on the screen. This made it possible for the operating system to realize that the incoming text read aloud either the person is walking or driving even though the message couldn't be displayed on the screen.

C. MEASURING AVAILABLE ATTENTION

The final major challenge we perceive is assessing the user's available attention. And this move is more challenging than measuring the demand because it will involve many external factors and the factors related to the environment such as walking, driving, eating and conversing etc. .as these factors will involve usually detecting and evaluating activities external to the computer system.

Further, a light analysis and blanket classification of activities is totally insufficient for such challenge. For instance, if we take an example of a person driving a car on a totally empty straight highway will typically have some attention to spare or have some free space i-e to select music, to look at back whereas the same person driving a car on a busy rush highway or rush hour on a snowy day may have no available attention to risk.

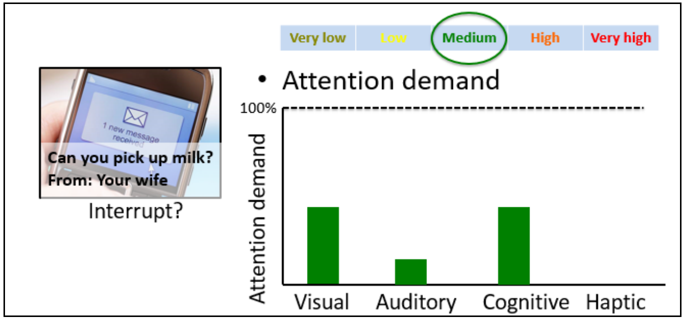


Figure 1: Proposed approach for priority set

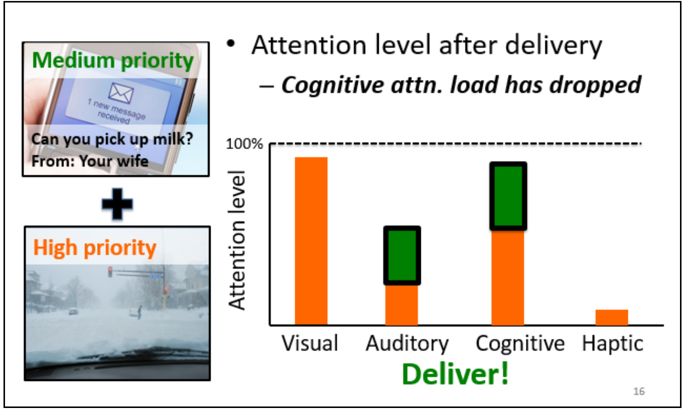


Figure 2: Attention level after delivery

Therefore, the determining of measuring available attention is such a difficult task. Therefore, in order to perfectly measure the available attention level, a mobile system may need to consider not only the user's current activity also the user's level of engagement with each and every activity.

Luckily, there is enough work on activity recognition that helps and we can use to meet this challenge. the mobile devices acquire multitude of sensors (e-g gyroscope, microphone, camera, accelerometer, GPS etc.). Usage of these [1] sensors for the activity recognition has been well studied [2][3][4][5].

For example, kern et al uses audio sensor data to check and dictate whether the user is in lecture. or in a restaurant, meeting with boss in a conversation or on the street. They also use body-worn accelerometers to determine the current state of the user whether the user is in the state of standing, walking, sitting or running. Thus, on relying these results to enable the operating system to determine the current activities that the user is currently engaged with which performances.

TABLE I ATTENTION LEVEL OF ACTIVITIES

Activity	**Possible attention level**
Sitting around	Very low-very high
Playing with a phone	Mid-very high
Walking	Mid-High
Having a conversation	Low-High
Writing an email	Low-Mid
In a meeting	Very low-Mid
Driving high speed	Very Low-Mid

IV. CONCLUSION

In this paper, we contend that the operating system of mobile's devices should be responsible for managing the attention of the user as a resource. With this advance responsibility, a mobile operating system can create an attention aware-notification for a user that initiates new interactions with right modality and at the right time without disturbing the high priority tasks. For creating such purpose system, we have lay out a design and methodology and we recognized key challenges in perceiving our vision.

REFERENCES

[1] N.D. Lane, E. Miluzzo, Hong Lu, D. Peebles, T. Choudhury, and AT. Campbell. A survey of mobile phone sensing. Communications Magazine, IEEE, 48(9):140{150, Sept 2010.

[2] Nickey Kern and Bernt Schiele. Context-aware noti_cation for wearable computing. In Proceedings of the 7th IEEE International Symposium on Wearable Computers, pages 223{230, Washington, DC, October 2003.

[3] Joyce Ho and Stephen S. Intille. Using context- aware computing to reduce the perceived burden of interruptions from mobile devices. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '05, pages 909{918, Portland, Oregon, April 2005.

[4] Tadashi Okoshi, Hideyuki Tokuda, and Jin Nakazawa. Attelia: Sensing user's attention status on smart phones. In 16th International Conference on Ubiquitous Computing, pages 139{142, Seattle, Washington, September 2014.

[5] Yunus Emre Ustev, Ozlem Durmaz Incel, and Cem Ersoy. User, device and orientation independent human activity recognition on mobile phones: Challenges and a proposal. In Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication, UbiComp '13 Adjunct, pages 1427{1436, Zurich, Switzerland, September 2013.