Revisiting Phone Call UIs for Multipurpose Mobile Phones

Matthias Böhmer  
DFKI GmbH  
Saarbrücken, Germany  
matthias.boehmer@dfki.de

Jonas Hempel  
Saarland University  
Saarbrücken, Germany  
jonashempel@gmail.com

Sven Gehring  
DFKI GmbH  
Saarbrücken, Germany  
sven.gehring@dfki.de

Antonio Krüger  
DFKI GmbH  
Saarbrücken, Germany  
antonio.krueger@dfki.de

Abstract
While mobile phones made a significant evolution in recent years from single-purpose communication devices to multi-purpose devices, the fundamental design of phone call applications did not evolve accordingly. While its implementation leveraged from new hardware and software capabilities, the fundamental decisions people are able to make when they receive a call did not change. Currently, when a call comes in, a modal dialog opens where the callee can either decline or accept the call. A recent study found that the current user interfaces of phone call applications (phone UIs) often lead to an increased overhead when application usage is being interrupted by phone calls [6]. In this paper, we revisit phone call UIs for multipurpose smartphones. We contribute a new design space for mobile phone call UIs, going beyond the simple accept-or-decline dilemma. We present a prototype implementation and discuss open challenges.

Author Keywords
Smartphones; phone calls; interruptions; design space.

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous.
Introduction and Related Work

Smartphones became multifunctional devices, which integrate a huge variety of services beyond pure communication capabilities. In addition to making phone calls, people can use their mobile phones for instance for navigation, mobile payment, making videos, taking photos or gaming and entertainment. The variety of activities that people use their smartphones for, and the number of installed applications is steadily increasing [4].

Recent research found that this dense integration of various functionalities comes at a certain cost [6]: people have to multitask between services. They have to switch between apps, what introduces an overhead on the app usage time. This overhead in particular is high when an application is being interrupted by an incoming phone call. The interruption might increase the completion time of the interrupted application by a factor of four. Albeit this happens only rarely [6], overall smartphone usage time is anticipated to increase further [4]. This will also result in an increased probability of smartphone applications being interrupted by incoming phone calls. Hence, rethinking user interfaces for phone call applications becomes indispensable.

Related work mainly investigates phone calls interrupting primary tasks outside of the phone, e.g. driving or office working. Brumby and Seyedu [3] study the auto-lock feature of modern mobile phones and find that people get under pressure to return their attention to the phone sooner than usual to prevent the lock. Teevan and Hehmeyer [7] find that a callee is interestingly more likely to accept a call when signaling “busy” or “do not disturb” to the caller. Wöckl et al. [8] investigate actions to follow-up on accept and decline for video telephony, e.g. like sending text messages or opting out video. However, although it is known that phone calls can be interruptive, new designs for call notifications with concurrent mobile application usage have not been investigated yet.

While computers that were enriched with phone call functionality (e.g. voice over IP and Skype) allow for a high degree of multitasking, mobile phones that were enriched with computing capabilities (i.e. smartphones) do not allow for any multitasking. Users of PCs can for instance continue writing a text while a call notification of Skype is pending (desktop scenario); but smartphone users cannot continue using a map app when a call notification is pending. There are only two options to continue with mobile app usage if a call comes in: (i) wait for the caller to hangup or for the answering machine to pickup; (ii) decline the call, what might have social implications.

When a phone call comes in on a mobile, not only the call application will open instantly, also the user can only either accept or decline the call. The current design of phone call applications is a remnant of times when the main purpose of mobile phones was making phone calls. A comparison of the two mobile phone call UIs shown in Figure 1 suggests that when mobile phones evolved to multi-functional devices, the design of phone call applications essentially remained the same. Apart from the fact that hardware buttons have been replaced by touch screen buttons and additional content can be shown, the user’s options for handling incoming phone calls essentially did not change.

In this poster, we revisit and discuss the design of phone call apps in the context of multipurpose smartphones. We aim on mitigating the negative effect of phone calls interrupting mobile app usage. We explore the design space and give rise to new designs of mobile phone call UIs.
Extending Phone Call Applications

After analyzing current implementations of phone call applications on some popular smartphone models, we found that they commonly have two shortcomings, which in particular may amplify the effect of interruptions:

1. Current phone call apps use full-screen modal dialogs to notify the user of incoming phone calls. This visually detaches the user from his previous application and thus might lead to higher impact of the interruption.
2. Current phone call apps only provide the user with two options to promptly either accept or decline the incoming phone call. This inevitably required decision (accept vs. decline) may amplify the interruption. Further, accepting the call pulls the user’s attention further away from the previous application to the phone call, and declining may have additional negative side effects.

As found by Leiva et al. [6], this particular design of call applications results in high overheads in usage times of the primary applications, which e.g. might be writing a mail or searching for something on the Internet. As described earlier, this design results from adopting phone call UIs of the previous generation of mobile phones with hard-buttons to the current generation of smartphones providing touch screens. In fact, current mobile operating systems allow users to multitask between different applications, including the phone call application. However, current phone call applications do not provide any options to the user for multitasking or handling incoming calls beyond accepting or declining the call.

We tackle the two described issues by revisiting and extending the design space of phone call applications as follows: In particular, we (1) increase the user’s freedom in deciding when to pickup a call by introducing the possibility of postponing an incoming phone call. Furthermore, we (2) re-iterate on the design of user interfaces for phone call applications to mitigate the interruptive effect of incoming calls while an application is being used. In particular, we are extending the conceptual design of current phone call applications to allow for a higher degree of multitasking and additional options to handle incoming calls as described hereinafter and shown in Figures 2b) to 2d).
**Time-multiplexing by postponing calls**

A first improvement to mitigate the effect of phone call interruptions is to give the callee an additional option when to pick up the call. We provide a third option for postponing the call besides accepting or declining it. Hence, the user can easily and quickly return to his previous task without a need to decide on how to react on the incoming call. This picks up the idea of time multiplexing the primary application with the phone call application [1]. The approach of postponing calls transfers a user’s ability to pick-up the call at will from landline phones to mobile phones: When a user is working on a stationary computer and the landline phone rings, he can either immediately answer the call or first finish his primary task (e.g. writing an email).

As Figure 2b) shows, this option can be implemented as an additional button besides accept and decline when the full-screen notification for a call pops up. When a call is postponed, the phone call app should go into background and the user can continue working in his previous application. The caller will wait in line. Upon the user’s decision or after a certain amount of time the call app will come to the foreground again, and again the user has the three options to accept, postpone or decline the call.

**Space-multiplexing notifications**

Space-multiplexing of primary and secondary tasks on the device’s limited screen real estate has been found to providing the user with more control while being interrupted [1]. Therefore a second approach to mitigate phone call interruptions is to alter the visual appearance of call notifications, as Figure 2c) suggests. Rather than having a full-screen notification as in nowadays phone apps we propose to divide the mobile screen’s limited space into two areas: In a smaller area the user is notified of the incoming call, as shown at the bottom in Figure 2c). An again, the user has the aforementioned options to react on the interruption. In the second larger area, the user can continue working on his primary task, which is for example navigating a map (Figure 2c).

Following on the idea of space-multiplexing, the user can keep his attention in the primary app and will not be visually decoupled from the previous app. For instance, he can continue reading or finish what he was just typing.

**Background alerts**

Another possibility for notifying users of incoming calls are background alerts, which will occupy only very little space of the screen. As can be seen in Figure 2d), notifications for incoming phone calls can be presented in the same way as other common events, e.g. by a notification in the device’s top bar. This results in incoming calls being put into background directly. Hence, the user can stay in the primary application and continue with his current task and reach an intermediate state to leave his application. This allows the user to intentionally leave his current application to pickup the call.

**Scheduling on app completion**

Messages have been found to be best delivered on the transition of activities [5]. Therefore, an additional approach to schedule a call interruption is to wait until the user closes his current application. Following on this idea, when a call comes in, the user will not be directly notified of the call. But when he closes the current application, a notification will be shown. Since the user is done with the recent application, a full-screen notification with a postpone option is reasonable. Since we cannot know for how long the user will stay in the current application, after a certain short amount of time a fallback to one of the aforementioned approaches is required.
Prototype Implementation

While in previous generations of mobile phones the phone call application was the device’s essential part, and everything else was build around it or integrated into it, phone call apps on the current generation of smartphones are just pieces of software like any other app running on the device. This enabled us to build a prototype that implements the aforementioned approaches to notify users of incoming calls and to test new ways of handling calls.

We implemented the presented design options for call notifications for Android-based smartphones. Figure 3 depicts the four different UI designs for a user navigating on a map as a primary task. Figure 3a) shows the option to push a postpone button, 3b) shows the call notification multiplexed with the user’s map application so that he is able to continue working in his primary task, and 3c) shows how the user is able to continue in the primary application when the incoming call is put into background. Note that in the latter case the user is notified of the incoming call only by the icon in screen’s top bar.

Discussion, Challenges and Future Work

Besides adding the opportunity to postpone a call and extending the UI aspects of the notification to mitigate the problem of call interruptions, we also need to be concerned about the caller, about other modalities like ring tone and vibration, and about which UI option should be chosen to display the call.

Reasoning on predicted overhead

The four extensions of current phone call apps differ in the degree in which they pull away the users attention from the application that he currently is using. One can measure the overhead that results of an phone call interruption [6]. To reason on how to handle an incoming phone call, i.e. which UI to use to notify the user of the incoming call, a model that predicts the overhead will be beneficial. While for instance the accept/decline/postpone dialog might be best suited when the user is making an entry in his calendar, a space-multiplexed notification might be best suited when he is watching a video, and a background notification might be best suited when the user is playing a highly engaging game.

To predict the resulting overhead various variables are available: Besides the interrupted app’s name and type also the user’s last reaction on an interruption, and the caller ID as well as contextual factors like time of day or location can be taken into account. Such a model will be beneficial when reasoning on how to react on an incoming call: If the overhead is estimated to be high, a less intrusive background notification will be chosen, if the overhead is estimated to be low, a more interruptive accept/decline/postpone dialog might be the best option.
Keeping the caller waiting in line
When the receiver postpones a call or the call is put to background automatically, the caller will be put in the waiting line. Since related work [2, 7] found that it has a positive effect when the caller is aware of the callee’s status, one idea might be to signal the caller what is happening. One possibility could be the design of special call-progress tones (e.g. for call was postponed or call was notified in background), or we could use speech-synthesis to tell the caller that "callee is watching a movie".

Other modalities
The notification scenarios we described are targeting visual attention. In addition, incoming calls are also being notified by auditive and haptic signals. A holistic design needs to concern these modalities to notify the user on incoming phone calls. One possibility could be changing the ringtone to unobtrusive sounds. Hence, the user could be notified about an incoming phone call in an ambient way. As another possibility, we could apply different vibration patterns to create haptic notifications in accordance with the visual notification. The integration of different modalities therefore needs to be addressed in the future, and be aligned with the visual notification.

Conclusion and Future Work
In this paper we revisited the design of mobile phone call UIs. Motivated by the high interruption cost of incoming phone calls on concurrent app usage, we explore new UIs for notifying users of incoming calls. We extended users’ options on how to handle calls by a postpone function in addition to accept and decline, further we presented the approaches of multiplexing, background alerts and scheduling on app completion for call notifications.

For future work, we are planning two complementary studies: In a controlled lab study we will investigate the proposed UI designs (esp. task performance and resumption lag). We will control for the phone call UI and the interruption. In a second quasi-experimental study will deploy our app to the app store and investigate more diverse and natural usage of our prototype. Especially we are interested in how often people will postpone calls, how they will configure certain parameters (e.g. duration of postponing), and if behavior depends on caller ID. A similar commercial app already is available.1

References

1Android App Andro+ Call Backgrounder/Call Manager