# Context-aware Communication in the Car

**Tobias Martin**
University of Munich (LMU)
Munich, Germany
t.martin@campus.lmu.de

**Stefan Lang**
University of Munich (LMU)
Munich, Germany
stefan.lang@campus.lmu.de

**Simon Weiser**
University of Munich (LMU)
Munich, Germany
simon.weiser@campus.lmu.de

## ABSTRACT
TODO: put text here

## ACM Classification Keywords
H.5.2 Information Interfaces and Presentation (e.g., HCI): User Interfaces — *Prototyping*; H.4.3 Information Systems Applications: Communications Applications

## Author Keywords
Automotive user interfaces, calling while driving, context sharing, driving safety, phone call.

## INTRODUCTION
TODO: put text here

## RELATED WORK
TODO: put text here

## CONCEPT
TODO: put text here

## IMPLEMENTATION
We designed a simple and ordinary contacts app which is enhanced by displaying the context of the person you want to call. For our goal the term context means to consider mostly all of these information:

| Context Information |
| --- |
| Activity |
| Road Type |
| Destination |
| Remaining Travel Time |
| Position |
| Hands-free Speaking |
| Speed |
| Weather |

Table 1: Context Information.

First of all you want to know what the person you want to call is doing at the moment. We distinguished between the activities *driving*, *cycling* and *still* where the latter means the person to call is doing nothing right now and his/her device is motionless on a table for example. Depending on if someone is driving you want to have the listed additional information. One would be the road type which lets you know if someone is driving e.g. in a city or on a highway etc. Another one is the destination the person is heading to and the remaining travel time. Therefore you need the person's position. This could either be a pinpoint GPS-coordinate, or if the user does not want to share his exact location, just a radius or a place name of his current position. Another very important aspect of the person's context is if hands-free speaking is enabled or disabled while in the car.

Besides information about the speed and the current weather situation are helpful to know. At the current state of our app all of these information can be accessed automatically except the destination and the remaining travel time, which have to be typed in manually.

### App Design
The Android-App we build basically consists of two different views, an overview and a detail view (see figure 1).
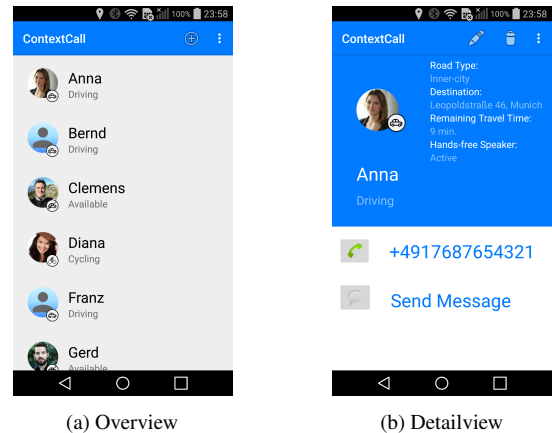


(a) Overview      (b) Detailview

Figure 1: Android Application ContextCall

On the overview on the left you see all your contacts as you know it from other contacts apps, but with the additional information about a person's activity, which is shown by the icons next to the image and the string below the name. By

taping on a contact, the detail view appears and shows the additional context information mentioned in table 1. The particular case in figure 1b for example gives information about the road type, the destination and the remaining travel time. Furthermore you get informed that hands-free speaking is enabled. If you then decide to call 'Anna' although she is driving, a small alert pops up and gives you three options (see figure 2). You can either call her or not, or you make use of the '*remind me*'-option which notifies you when her status is 'still'.

⚠ Context Info

'Anna' is currently driving.
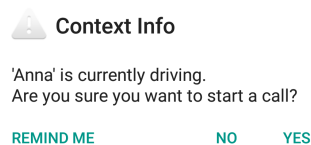Are you sure you want to start a call?

REMIND ME          NO        YES

Figure 2: Alert Pop-Up

At the beginning we implemented most of the context recognition by our own. We used for example GPS to find out if someone is driving. If someone's speed was over 10 kilometers per hour we set the status to 'driving'. But in mid-May of 2016 Google released its new so-called *Awareness API*[1] which made things much easier for us as programmers. From this point on we were able to get the activity of a user by only a few lines of code and this API became the core of our application.

The Awareness API is part of the *Google Play Services* and is a unified sensing platform, enabling apps to be aware of all aspects of a user's context, while managing system health for you [?]. With this API your app is able to recognize the following 7 different context types [?]:

**Location** The user's current location as a latitude and longitude value.

**Place** A semantic version of a location that is called place (including the place type, e.g. a coffee shop).

**Beacons** What is around a user? Are there nearby beacons that can be detected and identified?

**Time** The local time of a user that can be combined with other context information to form a more complex condition.

**Headphones' State** Are headphones plugged in the device or not?

**Weather** Ambient conditions like the weather, which have an effect on the user's behavior.

**Activity** The detected user activity (e.g. walking, running, biking and driving)

The latter of these is the important one for the goal of activity and context recognition. All of these information can be combined using *AND*, *OR*, and *NOT* boolean operators to build

complicated conditions that have to be met to trigger a notification. E.g. you can construct a condition that says that the user is driving in the car AND he is near a pharmacy AND it is during the opening hours of that shop. If these requirements are fulfilled, then you tell the user that he can pick up the wanted medication.

In particular, we used the *Fence API*[2], which is part of the Awareness API. The concept of *fences* is taken from Geo-Fencing, in which a geographic region, or "Geo- Fence", is defined, and an app receives callbacks when a user enters or leaves a region. Only that in this case it is not a region that is entered but an activity. So whenever the activity state transitions, it lets our app react to the user's current activity. For example, "Tell me whenever the user is driving". Once the conditions are met the app receives a callback and we can update the status of a user.

## USER STUDY
To evaluate our concept and the actual implementation we invited a total of 25 participants to our user study. As the overall goal was to evaluate if users perceive the context information provided by our application correctly, we decided to generate 10 different use-cases (scenarios). We vocally discussed each scenario with the current participant, but all the context information had to be pulled out from the running app on a provided smartphone. Differing by the call trigger (meaning the reason why a participant had to call a person) and the context the person being called is in, the participants had to decide whether they would like to:

1. Make the call or

2. Not make the call and

3. Not make the call but send a text message and/or

4. Not make the call but get notified when the driver status changes

Following the scenarios we asked additional questions with the participants acting the role of the driver (the person being called). Goal of these questions was to gather inside in the willingness of potential users to share more or less specific information.

The following figures visualise the outcome of our user study. Figure 3 to 5 focus on the 10 different scenarios which can be distinguished along the x-axis. As mentioned they differ in the status the person being called is in (driving, cycling, still). Distribution of the answers given by our participants can be seen along the y-axis, while colours visualise the four possible choices.

In figure 3

In figure 4

In figure 5

In figure 6

_____

[1]https://developers.google.com/awareness/

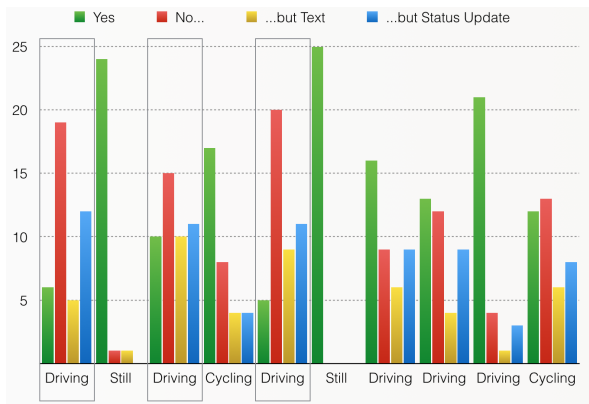[2]https://developers.google.com/awareness/android-api/fence-api-overview

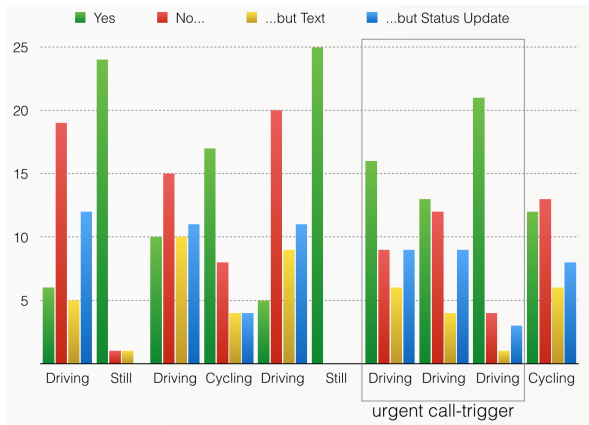Figure 3: User Study: Would you make the call (different scenarios)



Figure 4: User Study: Would you make the call (different scenarios)
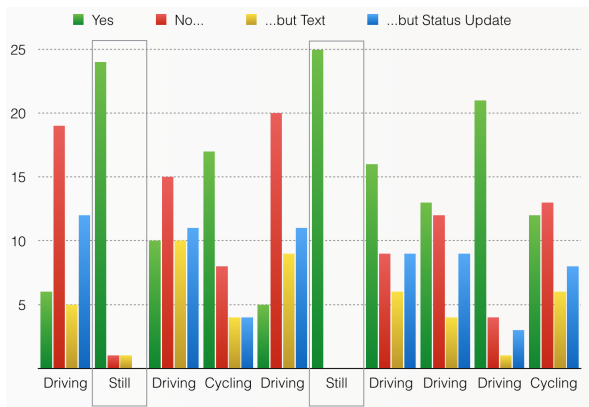


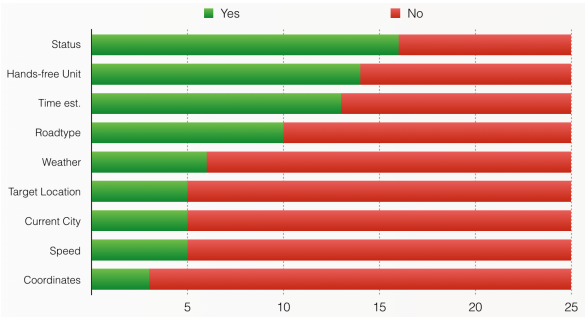Figure 5: User Study: Would you make the call (different scenarios)



Figure 6: User Study: What information are you willing to share

## CONCLUSION AND FUTURE WORK
TODO: put text here

## ACKNOWLEDGMENTS

Balancing columns in a ref list is a bit of a pain because you either use a hack like flushend or balance, or manually insert a column break. http://www.tex.ac.uk/cgi-bin/texfaq2html?label=balance multicols doesn't work because we're already in two-column mode, and flushend isn't awesome, so I choose balance. See this for more info: http://cs.brown.edu/system/software/latex/doc/balance.pdf

Note that in a perfect world balance wants to be in the first column of the last page.

If balance doesn't work for you, you can remove that and hard-code a column break into the bbl file right before you submit:

http://stackoverflow.com/questions/2149854/how-to-manually-equalize-columns- in-an-ieee-paper-if-using-bibtex

Or, just remove and give up on balancing the last page.

## REFERENCES FORMAT
Your references should be published materials accessible to the public. Internal technical reports may be cited only if they are easily accessible and may be obtained by any reader for a nominal fee. Proprietary information may not be cited. Private communications should be acknowledged in the main text, not referenced (e.g., [Golovchinsky, personal communication]). References must be the same font size as other body text. References should be in alphabetical order by last name of first author. Use a numbered list of references at the end of the article, ordered alphabetically by last name of first author, and referenced by numbers in brackets. For papers from conference proceedings, include the title of the paper and the name of the conference. Do not include the location of the conference or the exact date; do include the page numbers if available.

References should be in ACM citation format: http://www.acm.org/publications/submissions/latex_style. This includes citations to Internet resources [?, ?, ?] according to ACM format, although it is often appropriate to include URLs directly in the text, as above. Example reference formatting for individual journal articles [?], articles in conference proceedings [?], books [?], theses [?], book chapters [?], an entire journal issue [?], websites [?, ?], tweets [?], patents [?], games [?], and online videos [?] is given here. See the examples of citations at the end of this document and in the accompanying BibTeX document. This formatting is a edited version of the format automatically generated by the ACM Digital Library (http://dl.acm.org) as "ACM Ref." DOI and/or URL links are optional but encouraged as are full first names. Note that the Hyperlink style used throughout this document uses blue links; however, URLs in the references section may optionally appear in black.

BALANCE COLUMNS

REFERENCES FORMAT References must be the same font size as other body text.