Demo Abstract MoodChat: Using Context-Awareness to Connect Likeminded Co-located Individuals

Tomas Díaz[†] and Christine Julien[‡]
[†]Universidad de Chile
[‡]The University of Texas at Austin
Email: tomas.diaz@ing.uchile.cl, c.julien@utexas.edu

***Abstract*—This paper describes a demonstration of an application called *MoodChat*. This application is built to showcase the Grapevine context sharing middleware. Grapevine provides a very lightweight but potentially lossy way to efficiently share context information among a group of co-located pervasive computing devices using just device-to-device connections to exchange small bits of context information. Further, Grapevine embodies abstractions that allow flexible definition of context-defined *groups*, i.e., clusters of co-located users that are similar in some way based on their context values. MoodChat uses Grapevine to share context about device users' personal states (i.e., mood), including an individual's own assessment of his mood but also values sensed about the device or environment that might also be indicative of the user's mood (e.g., the level of ambient sound, the number and length of sent or received text messages, etc.). Co-located users with the same assessed level of mood are connected into groups, and users within the same group can chat with one another. The demonstration will allow the visitors to interact in MoodChat as one of the users, showing how context-defined groups can enable a wide range of flexible pervasive computing applications.**

I. Introduction

Applications in pervasive computing environments commonly use context information about the devices and their users to provide flexible and expressive application-level behavior. MoodChat, described in this demonstration, is an example of such an application. MoodChat (1) assesses a device user's mood; (2) shares metrics of that mood with other nearby devices and their users; (3) forms groups of co-located users with the same or similar mood; and (4) allows nearby users a variety of views of these mood groups.

To assess a device user's mood, we use a suite of metrics that the application can use individually or in aggregate. Most directly, we allow the user to explicitly select his mood from a set of choices. For now, these choices are *happy*, *sad*, *angry*, and *disgusted*[1]. However, we also measure things about the user's device and surroundings that may either indicate or influence his mood. In this demonstration, we measure the level of ambient sound and the number of text messages the user has sent and received in the recent past. This is a trimmed down set of metrics that might indicate the user's mood; other ongoing research works in a similar vein to generate a broader and more accurate view of a mobile device user's view based on a wider variety of on-device measurements [6], [7], [5].

To share metrics of mood with nearby devices, we rely on and extend the Grapevine context sharing framework [2]. Grapevine allows sharing very small summaries of a pervasive computing device or user's situation. We describe Grapevine in more detail in Section II. We also rely on Grapevine's expressive group mechanisms [4] to automatically aggregate context received about multiple users into a shared view about a group of pervasive computing users in a similar situation (i.e., with the same mood and in the same general location). MoodChat automatically creates groups of users with the same or similar mood. Then other users that do not share that mood can see the general location and members of the group (e.g., a sad person could seek out a group of happier users). Members within the same group can also use MoodChat to chat directly with other like-minded nearby people. Section II also describes the details of the MoodChat implementation.

This demonstration showcases both what Grapevine can do in supporting expressive context-sharing pervasive computing applications and the potential uses of sharing emotional states of users through a fun and interactive application.

II. Implementation

Our implementation of MoodChat can be separated into two main components; (1) the core module, built on the Grapevine framework [2], [4], which enables an easy-to-use way for applications to share succinct context information between peer devices, and (2) the Android application itself, which provides the user interface and the networking capabilities.

A. The Grapevine Framework

The Grapevine middleware enables succinct context information sharing, the dynamic computation of groups based on shared situations and the assessing of the aggregate context of each group.

Like much work on context-awareness in pervasive computing, we take a wide perspective on the nature of context. Specifically, context is any information used to characterize the user's situation [1]. Such context may be obtained from many different sources. For example, the devices' onboard sensors may capture information about the ambient environment (such

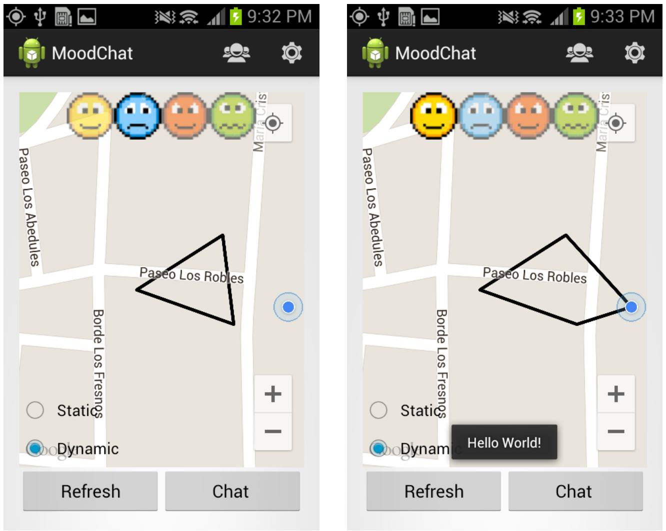
[1] This set could easily be extended in the future to include *fear* and *surprise*, thereby covering Ekman's six basic emotions [3].

as the surrounding sound level) or the device’s physical situation (such as GPS location or battery level). Other aspects of the device might also capture the user’s context. For example, the user’s device usage may provide important context information (such as the number of calls or messages sent or received in a given period, the number of applications running on the device, or the number of clicks or other interactions with the device). Alternatively, directly connected sensing devices may capture more specific context information (such as the user’s heart rate and or other biometrics). Finally, context could also be directly input by the user via buttons or text entry. All of these aspects are generically *context*, and they may all prove relevant in describing an entity’s situation.

The Grapevine context framework is generic and allows easy representation of a rich variety of context. Each piece of context information is stored as a key-value pair in a Grapevine context summary. We do not describe the inner details of Grapevine in detail, but the Grapevine approach provides a (potentially lossy) way to radically reduce the amount of data stored to represent an entity’s context. With this reduction in the size of the context representation, it becomes technically feasible for devices to share their context summaries with all directly connected devices using device-to-device communication capabilities. This allows a pervasive computing device (and its applications) to learn the context of other nearby users and their devices. These are situations to which the user (or his device) may not reasonably have direct access.

Within the Grapevine framework, a *Context Handler* (on every device) can then process the summaries received from nearby individuals to determine the existence of groups whose context summaries satisfy some aggregate constraint. More concretely, at the application level one defines Grapevine groups as subclasses of a *GroupDefinition* class. Within this definition, the application provides a group definition predicate that is evaluated over the context summaries received from other individual devices in the surrounding network. Some possible examples include groups defined over a set of devices whose locations are within a given radius, all those devices whose measured ambient sound levels are below a given threshold, or all devices whose battery levels are low and those who are currently charging.

Grapevine can then also process the context summaries belonging to all members of a group to obtain an aggregate context summary that describes the group members’ shared situation. The context attributes stored in this group context summary may or may not be the same attributes used to compute the group in the first place. For instance, in MoodChat, a group may be formed simply by combining all of the individuals who have selected the same “mood” value from a set of buttons on the user interface. However, an attribute of that group’s context may be the average sound level of all of the group members (e.g., if we wanted to attempt to uncover whether the ambient environment is influencing the users’ moods) or it may be a bounding box that encompasses the locations of all of the group members (e.g., if we wanted to be able to publicize where the “happy” people are).



Hello World!

(a) The client can see nearby groups formed based on their mood

(b) The client becomes part of the group and receives group messages

Fig. 1: MoodChat’s primary view

B. The MoodChat Application

The MoodChat application has been designed to showcase the sharing of context summaries in a hyper-localized network so that users can discover the aggregate mood of users around them and interact with other individuals with moods similar to their own. Upon launching the MoodChat application, Grapevine is automatically invoked and creates the device’s individual context summary. This also starts a location service running in the background, periodically updating the summary with its current location. In the primary application view, the user sees a map of the nearby area (marking the user’s location) and an interface to manually set mood values. This basic view is shown in Fig. 1. The user’s current location (and its uncertainty) is depicted as a blue dot enclosed in a larger circle. The four moods available for the user to select are shown across the top; in these screenshots, the user has chosen “happy” (the yellow face), which is shown darkened. Through the user interface of the application, the user can set options, defining what context is to be shared with peers and controlling the network settings that will handle connections with peer devices. For the former, the MoodChat application currently allows the user to select whether the following attributes are included in the individual context summary: mood, number of incoming and outgoing calls and text messages, ambient sound level, and current activity (as defined in the Google Play Services API). These are selected using the settings menu (accessed via the cog in the upper right corner of the view shown in Fig. 1). These are the initial context elements we chose to include in MoodChat; including additional elements also indicative of mood is straightforward.

For controlling the network resources, MoodChat does not assume that the client's device is persistently connected to all peers all the time; instead MoodChat uses technologies that allow peer discovery of devices in the neighboring pervasive computing environment, e.g., via Bluetooth or Wi-Fi Direct connections. Once a connection is made, Grapevine communications commence. If connections break and reform over time, the context exchanges can pick up and drop off transparently. When connections are available, the Grapevine framework, running underneath the MoodChat frontend, periodically sends a context summary containing the device's context attributes across any connection to a peer device. In our implementation, we send updated context summaries every 4 seconds. The Grapevine framework beneath the MoodChat application is likewise also receiving any context summaries sent by any of the neighboring devices.

As described in the overview of Grapevine, applications can provide a *GroupDefinition* class to create groups using the individual devices' context summaries. For the purposes of this demonstration, a group is defined as every connected user who has the same mood value (e.g., happy) as the device's user. Grapevine allows the application to define how far reaching the groups are; in our implementation, we allowed a device's group to include any devices within three network hops. Because WiFi-Direct networks are always in a hub-and-spoke form, the maximum distance between any two nodes in our demonstration is two hops. However, theoretically (and with a different network technology), multi-hop networks could conceivably result in *asymmetric* views of the group at different devices. We compute the aggregate context of this group to be the convex hull of the locations of all of the group members. This is also shown in Fig. 1; on the left is the group definition before the user of the depicted device is included in the group, while the figure on the right shows the updated context information (i.e., convex hull) once the depicted user has been included. Once a device has computed a new group (and its group summary), the device shares the group context summary with neighboring devices (in addition to continuing to share the individual context summary).

In the demonstration, we use groups in two ways. First, for the users in the group, MoodChat provides a way for users of the group to converse as a group. When the user selects his mood, MoodChat starts a dedicated chat thread, allowing the device to listen for or send messages to every co-located peer in the same group as the user. Second, anytime a device receives a group context summary, the convex hull of the group is depicted on the map. Because the group's mood is not part of its context (for privacy reasons), just viewing a convex hull of a group does not tell you what the users have in common. The demonstration also adds an additional group context, namely the number of chat messages sent within a particular mood group. This additional piece of context information can communicate to MoodChat users how active different groups are.

In our implementation mood is manually selected, which allows for a nice and interactive demonstration. However, the Grapevine middleware handles context sharing and group formations independently from the application thread, which provides the context input, so other approaches in mood recognition such as video and voice, smartphone usage or physiological signals, amongst others, would integrate easily with the MoodChat framework and the Grapevine middleware underneath. A more complete view of the demonstration is available as a video[2].

III. Demonstration

The demonstration will allow visitors to use MoodChat on devices we will supply as part of the demonstration. In theory, users with WiFi-Direct enabled Android devices will also be able to install MoodChat on their own devices and join the demonstration. We use WiFi-Direct communication to support the device-to-device exchange of Grapevine context summaries; the devices that we will use for the demonstration will already be connected to one another in a pre-defined WiFi-Direct group; others' devices will have to be added to this group.

When the demonstration starts, the user will be brought to the main interface, with a Google Map showing the area of the conference center in Saint Louis, Missouri. Using the symbolic selector for *happy*, *sad*, *angry*, or *disgusted* (as depicted in Fig. 1, the demonstration visitors will be able to select their current mood and that will be shared with the other devices. As this happens on multiple devices, and multiple devices have users with different moods, the participants will see their displays update to depict the convex hulls associated with the different likeminded groups. A text field will allow participants to write and send messages to the other participants with the same mood, which will be displayed as a toast on their screen.

IV. Technical Requirements

This demonstration has no particularly special technical requirements. We will provide our own devices (and therefore our own networking). Proximity to power would be nice to ensure that the devices stay powered for the entire demonstration.

References

[1] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 304–307, 1999.

[2] E. Grim C.-L., Fok, and C. Julien. Grapevine: Efficient situational awareness in pervasive computing environments. In *Procedings of Percom (WiP)*, pages 475–478, March 2012.

[3] P. Ekman. An argument for basic emotions. *Cognition and Emotion*, 6(3-4):169–200, 1992.

[4] C. Julien. The context of coordinating groups in dynamic mobile environments. In *Proceedings of Coordination*, pages 49–64, June 2011.

[5] Robert LiKamWa, Yunxin Liu, Nicholas D. Lane, and Lin Zhong. Moodscope: Building a mood sensor from smartphone usage patterns. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, pages 465–466, 2013.

[6] MoodTraces. http://www.cs.bham.ac.uk/research/projects/tod/MoodTraces.html, 2014.

[7] V. Pejovic and M. Musolesi. InterruptMe: Designing intelligent prompting mechanisms for pervasive applications. In *Procedings of Ubicomp*, 2014.

[2] http://goo.gl/WKzkLw