An IoT Infrastructure for Ubiquitous Notifications in Intelligent Living Environments

Thomas Kubitza
University of Stuttgart
Stuttgart, Pfaffenwaldring 5a, Germany
thomas.kubitza@vis.uni-stuttgart.de

Dominik Weber
University of Stuttgart
Stuttgart, Pfaffenwaldring 5a, Germany
dominik.weber@vis.uni-stuttgart.de

Alexandra Voit
University of Stuttgart
Stuttgart, Pfaffenwaldring 5a, Germany
alexandra.voit@vis.uni-stuttgart.de

Albrecht Schmidt
University of Stuttgart
Stuttgart, Pfaffenwaldring 5a, Germany
albrecht.schmidt@vis.uni-stuttgart.de

Figure 1: The meSch**Hub** allows the rapid interweaving of arbitrary devices and communication technologies. Tapping a NFC enabled smartphone on the meSch**Hub** will instantly connect it to its embedded WiFi access point and give access to its web based GUI

Ubicomp/ISWC'16 Adjunct , September 12-16, 2016, Heidelberg, Germany

DOI: http://dx.doi.org/10.1145/2968219.2968545

Abstract

Notifications are an important feature to proactively inform smartphone users. However, reading notifications on phones often has a disruptive effect on tasks at hand or social situations. Today's smart home and office environments with their increasingly connected devices provide rich opportunities to deliver notifications in a context-sensitive and multi-modal way into the direct surroundings of a user eliminating the need to fish up the smartphone from the pocket. Due to the missing tool support these opportunities have however neither been sufficiently investigated nor picked up for use in daily lives. With this work we introduce an infrastructure for homes and offices that enables designers and web-developers to design and deploy context sensitive notification strategies using arbitrary connected things and smart home products such as TVs, tablets, projections, lamps, speakers and many more. We introduce the opportunities offered by this platform to go beyond simple Event-Condition-Rules and describe how the system can be used to carry out remotely controlled "in the wild" experiments.

Author Keywords

Notifications; Internet of Things; Smart Environments, Smart Home; Smart Office

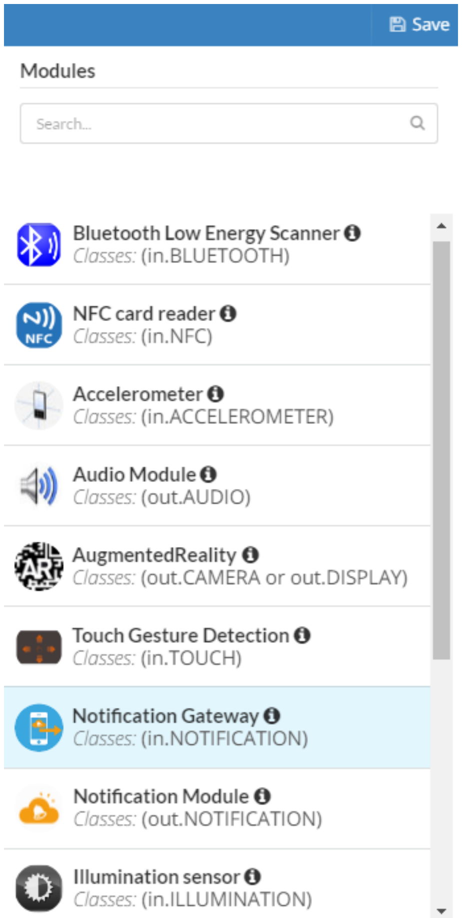

Figure 2: meSchup web GUI: Subset of sensor/actuator modules that can be remotely activated on an Android device

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous;

Introduction
Notifications on smartphones are an important feature to proactively inform their owners about new messages, calendar events, reminders and many other events. However, often notifications have a disruptive effect for instance when working on a task or in social situations. Further the effort of reading notifications on phones (pulling out the phone, activating the screen, unlocking it, reading a new notification, putting it away) rises with the frequency of received notifications. We believe that smart home and office environments offer the ability to detect a user's context and use the direct surroundings to display notifications (additionally) in appropriate and ultimately more efficient and less disruptive ways. Notifications could be for instance shown on the living-room TV when sitting on the couch, visualized on the digital photo-frame or indicated by a Philips Hue light when in the bed room and read by the Amazon Echo device when cooking in the kitchen. The interweaving of such heterogeneous devices usually requires a high implementation effort and low-level expertise in various platforms, programming languages and communication technologies and is thus out of reach for many researchers, designers and end-users.

We propose an infrastructure for ubiquitous notifications based on the meSchup IoT platform that is easy to setup, supports a rich set of DIY platforms and commercial smart home products out of the box and provides flexible means for implementing context-sensitive notification strategies using solely web technologies such as HTML, CSS and JavaScript. Overall it aims at lowering the barrier for the design, deployment and exploration of ubiquitous notifications in real-live home and office environments.

Previous work has explored the effect of notifications on mobile phones [5] as well as desktop PCs [2] in isolation. Recent work has as well analysed the usage of smartphone notifications in the large [6] and explored infrastructures to broadcast smartphone notifications among screen based devices (tablet, smartwatch, TV, desktop PC) of a user [8]. Some Popular cloud based service such as IFTTT[1] or Pushover[2] allow users to send notification to smartphones based on external triggers but not to use notifications on smartphones as triggers. Apps such as Tasker[3] allow to circumvent this but still only provide simple trigger-action rules. Weber et al. established design guidelines for displaying notifications on smart TVs [9]. However, what is missing is the exploration of context-sensitive intelligent display strategies of smartphone notifications in real world environments that go beyond screens as well as the necessary technical foundations for their realisation.

With "Designing Calm Technology" Mark Weiser has early described the concept of peripheral displays [10] that allow us to perceive information without explicitly shifting our attention to it. This has inspired the design of many ambient information displays and visualisations in art and research [1, 7, 4]. These insights can act as inspiration for designing ubiquitous

[1] http://ifttt.com

[2] http://pushover.net

[3] http://tasker.dinglisch.net

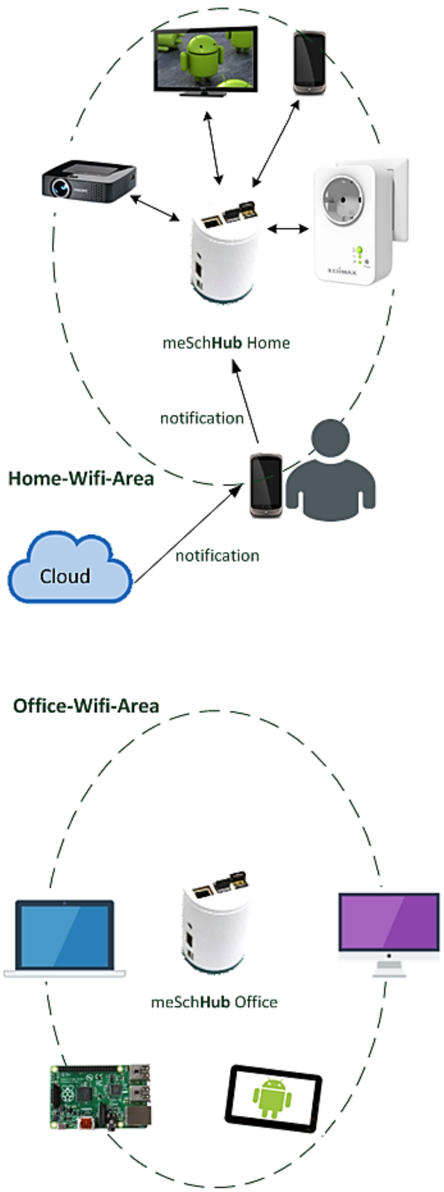

Figure 3: When a new notifications arrives at a users' phone it is forwarded to the local meSchHub in proximity which then controls local devices based on active interaction scripts.

notification displays that are less disruptive. However, it remains mostly unclear when smartphone notifications are forwarded to smart environments which strategies and multi-modal display forms users find useful, effective and appropriate for certain situations and notification types. Also further interaction methods need to be explored that allow users to remove or detail ubiquitous notifications from the environment similar to the interaction with the notification bar on smartphones. For instance a user might find it inappropriate that a certain WhatsApp notification is shown on a publicly visible near screen while being in a social situation. Here a tap gesture on the phone in the pocket could be for instance implemented that allows to remove the most recent message from any information display or to toggle its detail level.

Architecture and Components

The proposed architecture uses the meSchup IoT platform [3] for the realisation of ubiquitous notifications in smart local environments. meSchup is a generic modular IoT platform for the rapid creation of local interactive smart spaces. It consists of a meSchHub (Figure 1) that is equipped with communication adapters for various common communication technologies (e.g. WiFi, BLE, ZigBee, Z-Wave, etc.), the meSchup middleware, a script engine that runs user-defined *interaction scripts* and a web based development IDE that assists designers in creating this scripts. One hub can be set up per smart space (e.g. office, flat, house), see Figure 3. A large number of IoT and DIY platforms such as Android, Arduino, Raspberry Pi 1-3, Intel Edison and of-the-shelf smart home products (SmartPlugs, Ambilights, Chromecast, Amazon Echo, etc.) can be used with meSchup out of the box. The platform takes care of device discovery and fully abstracts from platform and communication technology heterogeneities. Interaction scripts allow to interweave arbitrary input with arbitrary output in real-time only requiring basic knowledge in web technologies (JavaScript).

Sensing Smartphone Notifications

meSchup provides an Android App that allows to expose any device sensors to the platform as well as to control any of its actuators in real-time from interaction scripts. This includes a "NotificationGateway"-module (Figure 2) that can sense any smartphone notification and forward it to the meSchup platform where interaction scripts define what to do with it. To respect the users' privacy only notifications from Apps are forwarded that the user has explicitly permitted (Figure 4). Besides setting up these permissions once, the App usually runs persistently and invisibly in the background. Notification events consists of the content, the originating App name, a timestamp and a rich set of additional meta-data (Figure 5) that can be used as additional context information for better notification display decisions. When a smartphone leaves the range of a certain smart space (e.g. out of home WiFi) and comes into the range of another known smart space (e.g. office WiFi) the App automatically discovers the local meSchup server and starts working with the interaction scripts that are defined for this space. This allows transparent roaming without any need for manual configuration.

Notification Output Opportunities

Based on the available devices in a smart home or office environments many output opportunities for incoming notifications may exist. Rich visual

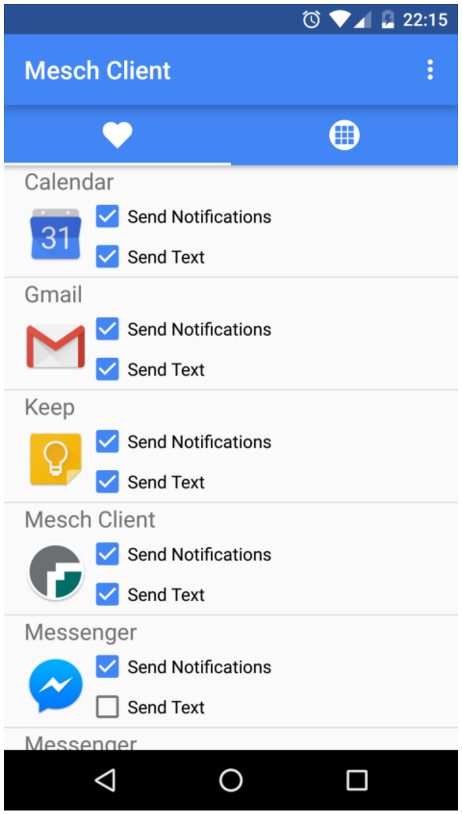

Figure 4: meSchup Android App: A user needs to explicitly give his permission that notification from the selected App should be forwarded

information can be displayed on TVs (using Chromecast), Projectors (using a Raspberry Pi), Desktop PCs (using Notifications in the right lower corner), tablets, smartphones and smartwatches. Further devices with lower visual information display bandwidth can be used to indicate notifications, for instance Lamps, Ambilights, LEDs (e.g. on/off, colour scale, blinking). Further notifications can be displayed or indicated in an auditory or tactile way. Devices that have integrated speakers and text-to-speak capabilities are for instance smartphones, tablets, TVs and the Amazon Echo device. Tactile notifications can be for instance triggered on tablets, smartphones or vibration motors connected to Arduinos or Raspberry Pis. When a notification is received the interaction scripts decide whether, when and what output events are triggered. An output event can be triggered on specific devices, specific output modules or even all available devices and modules. Obviously an intelligent notification display is selective and based on the current context.

Sensing Context

In the same way as meSchup allows to instantly trigger actuators in the environment it also allows to remotely activate available sensors and to collect and interpret its data as sensor events. Interaction scripts typically use the sensor data from stationary deployed and mobile devices in the environment to derive high-level contextual information from these sensor events. Two of many contextual information that can be derived with the system are for instance *person-tracking* and *phone-pose detection*.

Person-Tracking: We have implemented interaction scripts that use a few statically deployed Raspberry Pi and Android devices with BLE adapters that are able to robustly detect near BLE key-fobs or activity trackers (e.g. FitBit) on room level. By annotating these personal devices with person names and the stationary BLE scanners with room names it can easily be derived who is in a room at the moment when a new notification arrives. This obviously requires users to actually wear their key-fob, activity tracker or any other device with BLE advertising capabilities. This contextual information can be then used to deliver notifications to devices that are in the same room as the envisioned recipient.

Phone-pose detection: meSchup can activate a multitude of sensors embedded in smartphones. For instance we use the data from the proximity sensor, the accelerometer, and the display state to estimate whether the phone is located in the pocket, lying on a table or is in active use. When a notification arrives this is for instance used in an interaction script to decide whether it is actually forwarded to the environment or ignored (e.g. because a user is currently anyway interacting with his or her phone).

Much other contextual information can be derived using interaction scripts of the meSchup platform and the creativity of a script designer.

Designing Interaction Scripts

Interaction scripts are named pieces of JavaScript code that combine sensor input, contextual state and actuator output in real-time. When a script is saved it is instantly applied to the environment. A typical script could for instance forward notifications from the smartphone owned by "Tom" to any device with a display that is located in "Toms" proximity in the moment of the notifications arrival.

```
{ appName: 'WhatsApp',
  packageName: 'com.whatsapp',
  tickerText: 'Msg from John',
  title: 'John',
  titleBig: 'John',
  text: 'Hey Tom, how are you?',
  textBig: '',
  textInfo: '',
  textSub: '',
  textSummary: 'Hey Tom, how..',
  textLines: '',
  postTime: 1465907782096,
  number: 0,
  category: 'msg',
  priority: 1,
  visibility: 0,
  systemTime: 1465935004648,
  offset: 7200000,
  isOngoing: false,
  isClearable: true,
  when: 1465934997575,
  flags: 17,
  defaults: 4,
  ledARGB: 0,
  ledOn: 0,
  ledOff: 0,
  ringerMode: 2,
  isScreenOn: true,
  isGroupSummary: false,
  actionCount: 2,
  isLocalOnly: false,
}
```


Figure 5: A notification event in JSON format received from a smartphone which can be interpreted by a meSchup interaction script

Incoming notifications usually run through a processing pipeline that consists of the following steps:

1. **User filtering**: Through the permissions set on a users' smartphone only a subset of notifications from Apps that a user considers as useful are forwarded to the smart space.
2. **Interaction script filtering**: Interaction scripts may exist only for notifications from certain users or notification types.
3. **Contextual filtering**: Is the user currently in a room that contains a device with display capabilities? Is the user currently using his phone anyway?
4. **Modality selection**: Which of the available output modalities is best suited in the current situation? Should one or multiple be used in parallel? (e.g. visual and auditory)
5. **Display rendering**: Based on the chosen output-devices the content needs to be rendered compatible to the outputs' capabilities. (e.g. How to display a message with an Ambilight? How to display a received emoji through a text-to-speech interface? etc.)

Enabling "in the wild" Experiments

The exploration of appropriate ubiquitous notification strategies will usually not be representative when it is only based on pure lab experiments. In the wild experiments are in two ways supported by the proposed infrastructure. First, it is very easy to setup in any environment (power on and connect to router). Second, a cloud plugin can be activated that allows to remotely deploy, edit and remove local interaction scripts via a cloud based service website. This allows to remotely fix implementation issues of interaction scripts, for instance when reported by users during a long term experiment. Or it allows running remotely controlled A-B experiments that test various notification routing and displaying strategies against each other at different times and/or locations.

Conclusion and Future Work

In this paper we propose an infrastructure for smart spaces that allows forwarding and displaying smartphone notifications of inhabitants in intelligent context-sensitive ways directly in their surroundings. A large number of distributed sensors for context-acquisition and actuators for multimodal displaying of notifications are supported out of the box. Interaction scripts are written purely in JavaScript which enables the broad group of web developers to quickly start and realize custom ubiquitous notification strategies. Finally the integrated cloud connection supports the remote management of interaction scripts and facilitates a simpler realization of long term real world experimental deployments.

For our current and future research the proposed infrastructure acts as a valuable tool to answer a number of research questions:

1. Are specific notification types preferred to be forwarded to smart environments?
2. Can ubiquitous notifications reduce the disruption effect? How should these be designed?
3. Can ubiquitous notifications be more effective compared to the state of the art reading directly on smartphones?

4. Which notification output modalities are best suited for which situations?
5. How can content privacy be realized in multi-user and (semi-)public spaces?
6. How can users intuitively interact with ubiquitously displayed notification?
7. How can simultaneous notifications of different users be displayed by the same output resources?
8. How long (or often) should ubiquitous notifications be displayed and what are strategies for their removal?

We are looking forward that the proposed work will help us and other researchers in answering these and many other interesting research questions related to ubiquitous notifications.

Acknowledgements

This work is supported by the EU (MeSch, no. 600851), by the BMBF (13N13481) and by the DFG (EXC 310/1).

References

1. D. Hausen, S. Boring, C. Lueling, S. Rodestock, and A. Butz. 2012. StaTube: Facilitating State Management in Instant Messaging Systems. *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction - TEI '12*, ACM Press, 283. http://doi.org/10.1145/2148131.2148191
2. S.i T. Iqbal and E. Horvitz. 2010. Notifications and awareness: A Field Study of Alert Usage and Preferences. *Proceedings of the 2010 ACM conference on Computer supported cooperative work - CSCW '10*, ACM Press, 27. http://doi.org/10.1145/1718918.1718926
3. T. Kubitza and A.Schmidt. 2015. Towards a Toolkit for the Rapid Creation of Smart Environments. In *5th International Symposium on End-User Development (IS-EUD '15)*. Springer, 230–235. http://doi.org/10.1007/978-3-319-18425-8_21
4. H. Müller, M. Pielot, and R. de Oliveira. 2013. Towards ambient notifications. *Interact, Workshop Proceedings*.
5. M. Pielot, K. Church, and R. de Oliveira. 2014. An in-situ study of mobile phone notifications. *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services - MobileHCI '14*, ACM Press, 233–242. http://doi.org/10.1145/2628363.2628364
6. A. S. Shirazi, N. Henze, T. Dingler, et al. 2014. Large-scale assessment of mobile notifications. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, ACM Press, 3055–3064. http://doi.org/10.1145/2556288.2557189
7. T. Skog, S. Ljungblad, and L.E. Holmquist. 2003. Between aesthetics and utility: designing ambient information visualizations. *IEEE Symposium on Information Visualization 2003*, IEEE, 233–240. http://doi.org/10.1109/INFVIS.2003.1249031
8. D.Weber, A. S. Shirazi, and N. Henze. 2015. Towards Smart Notifications using Research in the Large. *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct - MobileHCI '15*, ACM Press, 1117–1122. http://doi.org/10.1145/2786567.2794334
9. D. Weber, S. Mayer, A. Voit, R. Ventura Fierro, and N. Henze. 2016. Design Guidelines for Notifications on Smart TVs. In *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video* - TVX '16. ACM Press, 13-24. http://dx.doi.org/10.1145/2932206.2932212
10. M. Weiser and J.S. Brown. 1996. Designing calm technology. *PowerGrid Journal*.