Works in Progress

Editor: Anthony D. Joseph ■ UC Berkeley ■ adj@eecs.berkeley.edu

Location- and Context-Aware Computing

EDITOR'S INTRO

This issue's Works in Progress department looks at different issues and applications in location-aware computing: letting users set and control privacy policies, cold-starting recommender systems, aggregating contextual information, and applying location-based services to public transportation environments. The department also includes a report on middleware to support transient virtual networks over low-power wireless personal-area-network nodes. —*Anthony D. Joseph*

SIMPLIFYING USER-CONTROLLED PRIVACY POLICIES

Mark S. Ackerman, Tao Dong, Scott Gifford, Jungwoo Kim, Mark W. Newman, Atul Prakash, and Sarah Qidwai, *University of Michigan, Ann Arbor*

Location-aware computing infrastructures are becoming widely available. However, a key problem remains: letting users manage their privacy while also giving them interesting applications that take advantage of location information.

Several systems have attempted to provide interfaces for expressing policies to give users substantial control over their privacy. Examples include restricting the times and places of access, editing the location resolution (for instance, room level versus building level), and excluding other people from accessing your location.

Unfortunately, preliminary experience with such systems indicates that users have trouble creating detailed policies and predicting the effects of their privacy preferences in advance (for example, see the work of Scott Lederer and his colleagues, "Personal Privacy through Understanding and Action: Five Pitfalls for Designers," in *Designing Secure Systems That People Can Use*, L. Cranor and S.L. Garfinkel, eds, O'Reilly, 2005. pp 421-445). To address the problem of poor prediction, Lederer suggested the notion of *privacy dials*, which give users a simple interface for controlling their location privacy at any time from a mobile device. A privacy dial can control the granularity at which location information is available to others (both the location resolution and whether the user is identified). Interfaces such as privacy dials can be useful, but they push the burden completely back to the user to maintain the settings accurately and actively at all times. Because privacy dials must be easy to use, they are also coarse-grained tools; for example, it's difficult to use different settings for different users.

In our work, we're exploring whether there's a middle ground between these two ends of the spectrum. In particular, we're examining ways to greatly simplify privacy-policy creation for users. Our work uses the contextual information from applications that help users coordinate or communicate with others, such as their calendars, messaging contacts, and address books. Our assumption is that this contextual information, produced through everyday applications, can help create privacy policies for location-aware systems. Users can then more easily create high-level policies

An example of where this would be valuable for users can be seen in the "Where Is Mark?" application (see Figure 1). In this application, users can determine whether they'd like their location to be shared shortly before a meeting, which lets other meeting participants know whether everyone will be on time. (The application was named for an often-tardy faculty participant.) Users need to set policies for making their location available: amount of time prior to the meeting, whether to include their exact location, and so on. Asking them to set the policies at a low level would be frustrating and lead to low compliance. On the other hand, it's quite easy to ask them whether they want such a policy set for the participants in a meeting scheduled on Google Calendar. Customizing the policy is only a matter of setting the amount of time prior to the meeting and whether they want their exact location provided.

This work has resulted in a new kind of infrastructure, one that is privacy

Published by the IEEE CS ■ 1536-1268/09/$26.00 © 2009 IEEE

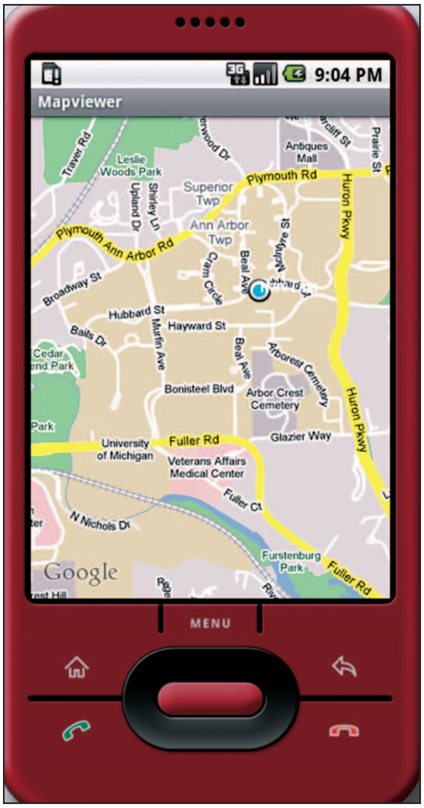

Figure 1. The GUI for the "Where is Mark?" application. At 9:04pm, it's obvious that Mark is not only late for the 9pm study group, but is at least 10 minutes away from the Computer Science building.

sensitive. The Whereabouts system is the base infrastructure that provides for high-speed policy invocation and a secure publish-subscribe mechanism for data sharing given a set of user-specified policies (see K. Borders et al., "CPOL: High-Performance Policy Evaluation," *Proc. ACM Conf. Computers and Communication Security*, ACM Press, 2005, pp. 147–157). The architecture is currently centralized but can be distributed for more privacy protection.

In addition, the project is creating two privacy-management utilities. The first, Policy Mirror, lets users see what the effects of any given policy, given their previous location traces and those of other users. In other words, Policy Mirror lets users to see what will happen on the basis of what they've done in the past.

The second utility, Privacy Circles, lets users share policies. A general rule of thumb in HCI is that only 1 percent of users create new customizations, but a much larger number will use other people's customizations (see W.E. Mackay, "Patterns of Sharing Customizable Software," *Proc. ACM Conf. Computer-Supported Cooperative Work*," ACM Press, 1990, pp. 209–221). Thus, we want to make it easy for users to use and share privacy policies.

The Privacy Circles and Policy Mirror utilities require an additional level of system support. We've also constructed Designers' Ubiquitous Computing Testbed (DUCT) Replay, as part of the infrastructure. DUCT Replay lets users replay past event streams, such as from location-aware sensors or identification services.

For more information, contact Mark Ackerman, Atul Prakash, or Mark Newman at {ackerm, aprakash, mwnewman}@umich.edu.

WHAT DO YOU LIKE HERE? COLD STARTING LOCATION SERVICES

David García, Paulo Villegas, and Alejandro Cadenas, *Telefónica I+D*

Recommender systems have achieved a satisfactory level of performance in many cases. Technologies such as collaborative filtering are currently well-tested and mostly reliable, but some rough edges remain. One active R&D area is the augmentation of recommender systems with contextual information to better match a user's instant interests. Location-based services include the time and location contexts of a request. Recommender systems will use this information to constrain their suggested best matches for a user to what's available here and now.

However, most recommender systems suffer from the cold-start problem: when users first enter the system, no information about them exists to help guide the recommendation algorithm—neither a user profile (for content-based recommendation) nor recorded past user activity (for collaborative filtering). The system must somehow acquire the initial user data. In location-based services, the problem can be more acute because the user data is context-dependent and might not be directly reusable across contexts.

We're prototyping a mobile service for personalized, context-aware leisure recommendations (see Figure 2). The service will suggest appropriate nearby activities (restaurants, bars, cinemas, and so on), adapted to the location, the time, and the user's tastes. When a new visitor enters a service area, we must characterize user preferences on the basis of the available local services. Some of the information thus acquired about user tastes might be reusable across different geographical areas, if we map profiles to new local offerings.

To create the initial profile, we're implementing an automatic procedure that builds a questionnaire and submits it to users to help define their tastes. Given users' natural reluctance to answer lengthy surveys, exacerbated by the complexities of answering them through the limited usability a mobile system interface, the procedure keeps the questions to a minimum. Our prototype uses a decision tree that, at each branching level, employs a defined utility function and the user's previous answers to discriminate among the options available in the current spatial-temporal context and select the next question. We designed the process to be incremental, with each step adding more discriminative capacity to the profile. The user can stop answering at any moment, and the answers collected to that point will still provide a profile

WORKS IN PROGRESS

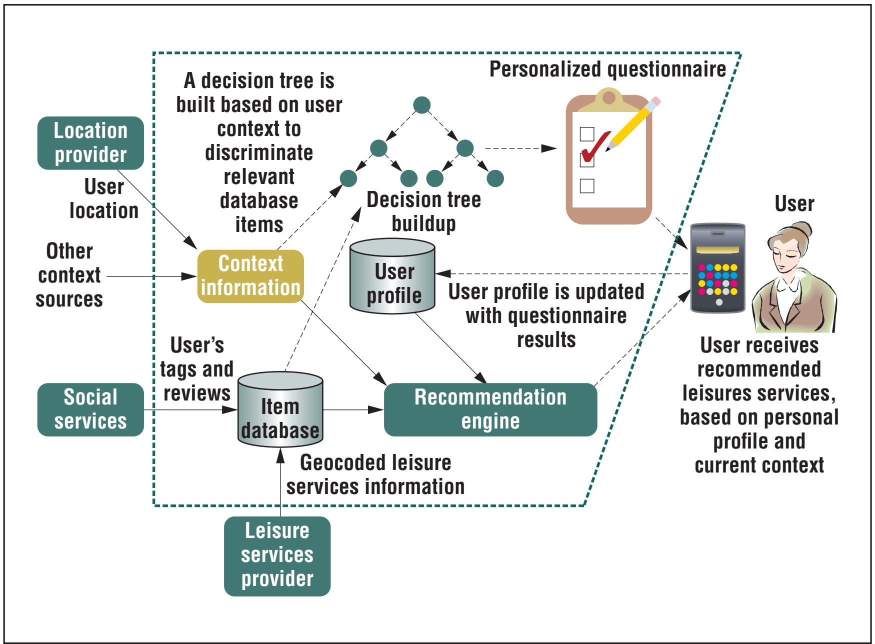

Figure 2. Mobile service for context-aware recommender system. A questionnaire automatically generated from a decision tree helps the system overcome the recommender cold-start problem.

that the recommendation algorithm can use.

To build the decision tree, our database needs fine-grained characterizations of spatially tagged items. Accordingly, the system fetches social data (folksonomy-based user tags, reviews, and categories gathered from online social services) for each local feature and uses statistical processing to structure that information into data that our defined utility functions can use. A pending design issue is how to evolve the acquired user tastes over time, so we can differentiate between short-term desires and longer-term stable preferences.

For more information, contact Paulo Villegas at paulo@tid.es.

LOCATION-BASED CONTEXT-MANAGEMENT PLATFORM

Alejandro Cadenas and Antonio Sánchez-Esguevillas, *Telefónica I+D*
Javier Aguiar and Belén Carro, *University of Valladolid*

Over the past few months, a research group composed of University of Valladolid professors and Telefónica I+D engineers has been developing a global convergent architecture for managing user contexts. The architecture bases a convergent-control layer on the IP Multimedia Subsystem (IMS) framework (see Figure 3). The control layer captures a user's context from different context providers, such as sensors or applications, over different access networks. Any service or application can subscribe to the centralized context management element—namely, the Context Management Enabler in Figure 3—to receive context notifications for specific subscribers via a contextual protocol we defined based on the Session Initiation Protocol (SIP) for transport.

The design and simulation phase is nearly finished, and we're preparing for a field deployment. The Context Management Enabler platform is based on the SIP registrar and proxy services of Mobicents, a Java open source SIP application server. The platform deploys the Fraunhofer FOKUS IMS control layer. The access network is the data network of the University of Valladolid's Higher Technical School of Telecommunications Engineering. We've designed the context providers to be the Bluetooth modules of the professors' mobile phones, detected at a Bluetooth dongle installed in each room, office, and lab of the school's building. Through an intelligent aggregation of location information and class timetables for each professor, the Context Management Enabler composes a contextual status that students can check to verify the professors' availability for tutoring.

This ongoing work will identify implementation-specific issues of the proposed reference architecture. It will also provide valuable performance benchmark data for system and network modeling in large-scale deployments.

For more information contact Alejandro Cadenas at cadenas@tid.es.

BUSTRACKER: DIGITALLY AUGMENTED PUBLIC TRANSPORTATION

Sean Mailander, Ronald Schroeter, and Marcus Foth, *Queensland University of Technology*

Public transportation is an environment with great potential for applying location-based services through mobile devices. The BusTracker study is looking at how real-time passenger information systems can provide a core platform to improve commuters' experiences. These systems rely on mobile computing and GPS technology to provide accurate information on transport vehicle locations. BusTracker builds on this mobile computing platform and geospatial information. The pilot

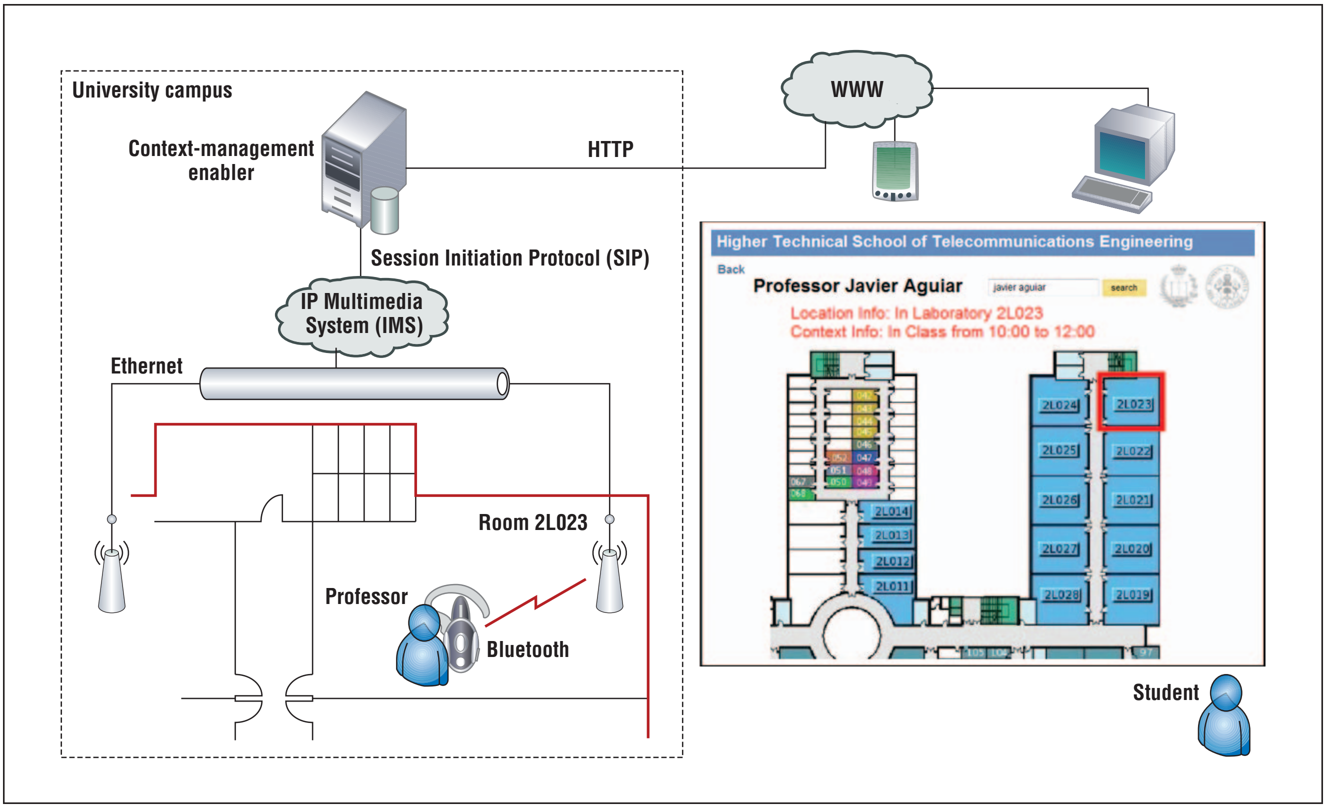

Figure 3. Global architecture for user's context management. The deployed infrastructure captures the professors' context, which the Context-Management Enabler processes. Any application can request the context information by implementing the appropriate contextual protocol, such as the Web-based application shown on the right.

study is running on the open source BugLabs computing platform, using a GPS module for accurate location information.

Previous research to enhance the user experience in urban environments has led to applications such as CityWare (www.cityware.org.uk), which uses Bluetooth nodes at public locations and a link from a user's Bluetooth device to his or her Facebook profile. CityWare presents information about the people an individual encounters most frequently. However, this system doesn't fully exploit the public transportation environment where familiar strangers, as Stanley Milgram described them (*The Individual in a Social World: Essays and Experiments*, McGraw-Hill, 1977), are together for extended periods at regular frequencies with little environmental stimulation. The characteristics of such spaces offer opportunities to test digital-augmentation scenarios that foster social connections between individuals or use ambient visualizations of historic presence data that don't require commuters to directly interact.

The BusTracker study is initially investigating the provisioning of real-time scheduling information to users through innovative design solutions on Web systems, mobile applications, and urban information displays. Once these interfaces are in place, the study will look at how to use the interfaces to engage commuters—either by embedding portals to social networking sites or by creating novel social networking experiences. Both approaches will exploit real-time location information to add new value to existing social networking.

In the first case, adding real-time location information can enhance existing social networking sites by supporting a collective presence online. For example, all passengers on a particular bus can join a collaborative group to chat, share podcasts, signal intended destinations, or ask for advice on tourist attractions. In the case of new applications, real-time location information can display accurate scheduling information. It can also assist in capacity management and on-demand public transport by letting people signal their intended trips in advance. Other applications of this kind might inform individuals of friends who are on closely aligned trips and suggest impromptu rendezvous through minor trip modifications, such as catching an earlier train. Or an application might suggest waiting an extra half hour at work to miss peak-hour crowds.

For more information, contact Sean Mailander at s.mailander@qut.edu.au or see www.urbaninformatics.net.

WORKS IN PROGRESS

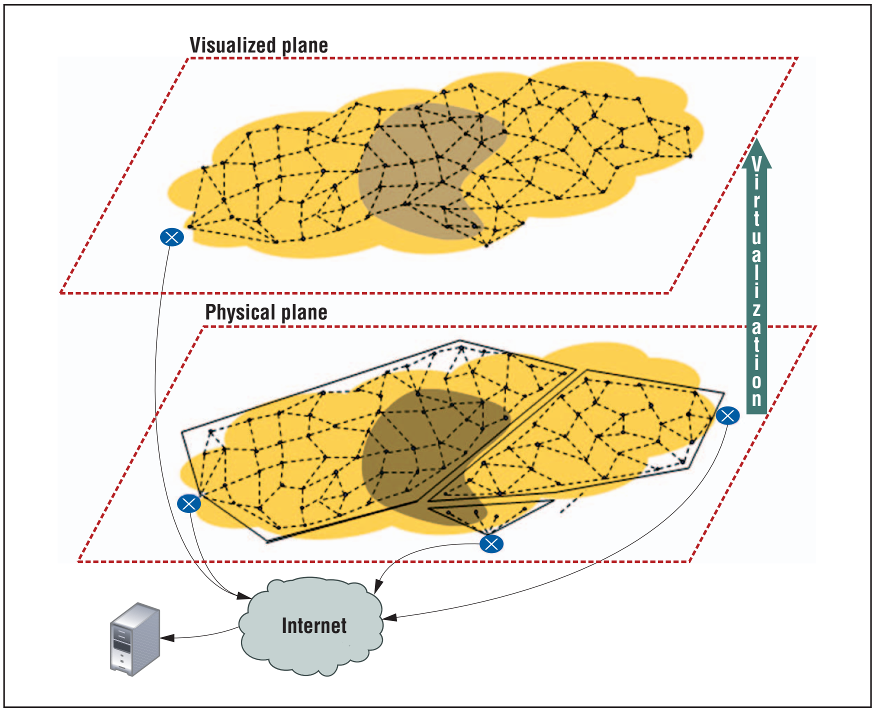

Figure 4. A virtual WPAN covering multiple physical WPANs. The shaded area shows a contour running through three physical sensor WPANs.

LIGHTWEIGHT VIRTUALIZATION OF LOW-POWER WPAN SENSOR NODES

Amiya Bhattacharya and Partha Dasgupta, Arizona State University

Wireless personal area networks (PANs) of embedded sensors are traditionally conceived to be privately owned and deployed with a single specific application in mind. An interesting possibility of participatory sensing emerges if we consider forming a transient, virtual, and fully programmable sensor network by stitching together a closely spaced cluster of real wireless PANs (WPANs), especially if minimal disturbance to the native applications can be ensured on the constituent sensor networks.

Work is underway at Arizona State University and New Mexico State University to develop middleware for supporting lightweight virtualization on resource-constrained WPAN nodes (popularly known as motes) along with MAC-layer bridging on their wireless interfaces. Power-efficient virtual WPANs require both technologies.

Lightweight virtualization of WPAN nodes turns out to be quite useful for incremental deployment of a wireless sensor-network infrastructure that accommodates heterogeneous mote hardware and operating system platforms, provided all motes support the identical MAC standard. Users can deploy each batch of identical motes to build the host WPANs, all of which can then support a virtual WPAN to perform untethered networked sensing applications. Spanning multiple domains is the most noteworthy feature that users can leverage to operate a virtual sensor network over host WPANs, even if they're across multiple ownership domains. Figure 4 shows a shaded area with a contour running through three physical sensor WPANs. A simple, power-efficient contour-detection algorithm can be used if the zone is covered wholly within a single virtual WPAN.

This new kind of participatory sensing and sensor data processing infrastructure is termed a *community sensor grid*, based on its similarity with the participation model used in computational grids.

For more information, contact {amiya, partha}@asu.edu.