From the Internet of Things to the Internet of People

The main objective in developing applications for the Internet of Things is to integrate technology into everyday lives. How this integration is implemented, however, leaves much room for improvement. Often the user must set parameters within the application, and when the person's context changes, they must manually reconfigure. What was meant to be a commodity in an unforeseen situation then becomes extra noise. Here, the authors describe a reference architecture that uses smartphones to improve integration with the IoT, thereby opening the way to new IoT scenarios that support evolution towards the Internet of People.

Javier Miranda
Gloin

Niko Mäkitalo
Tampere University of Technology

Jose Garcia-Alonso and Javier Berrocal
University of Extremadura

Tommi Mikkonen
Tampere University of Technology

Carlos Canal
University of Malaga

Juan M. Murillo
University of Extremadura

The Internet's availability almost everywhere has favored the emergence of myriad devices equipped with a connecting interface. Linking these and other everyday physical objects to the Net is getting ever easier, thus enhancing the popularity of the Internet of Things (IoT). One primary goal behind such smart integration of devices is to simplify people's lives by having technology work for them seamlessly.[1] For instance, you can remotely switch on your house's air conditioning using your smartphone to get a comfortable temperature when arriving home. Or you might schedule this task in accordance with your daily routine.

However, the way IoT technologies are currently integrated with humans leaves much room for improvement.[2] The technology has yet to develop suitable mechanisms to properly adapt to people's context or mood. Instead, far from making the technology work for people, people are forced to either change their context to fit technological requirements, or be slavishly aware of the system so as to send it commands or modify their schedule if their routine changes.

In a more desirable IoT scenario, technology would take people's context into account, learn from it, and take proactive steps according to their situation and expectations, avoiding user intervention as much as possible. Thus, if someone plans to arrive home late, they would like the air conditioning kept off until they're actually on their way back home.

Enabling such scenarios requires moving from the Internet of Things to the *Internet of People* (IoP). Here, we propose an infrastructure supporting this evolution, and making it possible to construct software for it. In our proposal, smartphones play a central role, reflecting their current use as the main interface connecting people to the Internet. Thus, we advocate endowing smartphones with a set of capabilities that improve the connection between people and the IoT. First, the smartphone must be able to learn about its owner and his or her context by constructing a digital profile (understood as the whole of the context information the device manages). Second, it needs to transparently negotiate and propose interactions with other devices on the Internet, reacting to stimuli and handling relationships. Finally, it needs to be able to manage digital profiles and act accordingly, providing its owner's context as a service for others, and scanning for services that might be of interest to its owner or to update his or her digital profile.

The infrastructure to support IoP is managed through a combination of the Social Devices[3] and People as a Service (PeaaS)[4] platforms (see the sidebar "IoT-Linked Smartphones" for more information). The Social Devices platform enhances smartphones' proactive capabilities to orchestrate their interactions with other devices connected to the IoT. PeaaS provides smartphones with serving capabilities that let people offer services from their devices, including providing their context and sociological profile. The combination of the two contributes to constructing an IoP by allowing people's context information to be included in the coordination and interaction management of IoT devices. The main benefit is the integration of people as first-class citizens in the IoT universe. This opens the way to the development of new kinds of human-friendly services and applications in the field.

People Aren't Things

The IoT increasingly pervades our daily lives. Currently, there are approximately 12.6 billion connected devices,[5] including people, processes, data, and things. This last "connected things" group includes not only mobile devices such as smartphones or tablets, but also desktop and laptop PCs, printers, smart TVs, cars, refrigerators, smart light bulbs, and so on. IoT numbers are increasing at a dizzying pace. Experts forecast 25 billion connected devices in 2020.[6] Even devices that don't come equipped with a connecting interface can easily be integrated into an IoT system using such hardware platforms as Arduino, .NET Gadgeteer, or even Lego Mindstorms for kids.

The complexity and heterogeneity of such a large variety of smart devices that can be connected to the Internet requires specific tools to manage them. Current approaches, at least within the industry, have merely been based on offering remote interfaces and endpoints to configure and manage devices, leading to the "basket of remotes" problem.[7]

The need to combine interactions between several devices led to the development of machine-to-machine (M2M) communications. There have been attempts at standardization, such as the Devices Profile for Web Services (DPWS)[8] and Constrained Application Protocol (CoAP).[9] Although user-friendlier approaches, such as Apple's HomeKit, help integrate home automation via smartphones, they're still in the development stage.

For practical applications, individual gadgets alone are insufficient. Instead, many real-life applications connect devices to cloud computing and storage facilities, enabling the creation of arbitrarily complex systems in which nodes, ranging from sensors to datacenters, perform computations. This architecture is a natural starting point for creating mashups that combine data from various sources into new, compelling ways to consume those data. It has, for instance, rapidly become a practical way to keep track of personal sports activities and interests.

However, although the problem of accessing, programming, and combining IoT objects can be settled with a cloud-based architecture, making them accessible to a broad spectrum of users (the way that IoT devices are related to people) requires more attention. Current tools and technologies supporting the interaction between humans and the IoT force the user to adapt to technology, instead of making technology adapt and assist the user. As the user's circumstances change, a conductor is required to manage the orchestra of IoT-linked devices and reduce the complexity of user interactions with them. In our opinion, smartphones can successfully perform this role. They're the natural interfaces connecting their owners to the outside world. Due to its intrinsically personal nature, a smartphone is almost always on its owner's person, and its capabilities, geolocating in particular, can be used to learn about the owner's context (detect how he or she behaves when at certain locations, surrounded by certain other people, and so on). This, combined with smartphones' growing computing capabilities, makes them ideal for learning about people and orchestrating their surrounding IoT. Still, every interaction involving people should be casual and hassle-free, giving the user full control of private data

IoT-Linked Smartphones

Two of our previous works conform to the basis of the Internet of People (IoP), using smartphones as the primary conductors to gather, link, and manage personal information. Here, we discuss these works.

Social Devices

Social Devices is an Internet of Things (IoT) model, first introduced in 2011 as a joint work by Nokia Research Center, Aalto University, and Tampere University of Technology.[1] The motivation behind the model was that smartphones have not only a lot of information about their owners, but also modalities that enable them to resemble humans. They can translate text into speech, for example. At present, the Social Devices concept is supported by a middleware platform named Orchestrator.js (www.orchestratorjs.org). This allows proactive triggering of interactions between devices of co-located people. Additionally, it offers a complete set of Web-based tools to define the interactions and their triggering contexts. Currently, the middleware allows custom IoT smart objects to be constructed using Arduino, Raspberry Pi, or .NET Gadgeteer. Work is ongoing to extend the support to new platforms, given that Social Devices aims to enable the creation of heterogeneous multi-user and multi-device applications for any type of device.

Various IoT-related scenarios have been implemented and introduced using Social Devices. One example is the Social Coffee Machine (http://vimeo.com/nikkis/gadgeteer). This application, based on user context, asks if the user wants to have coffee, and then, when the coffee is ready, invites other people to join the coffee break.

People as a Service

People as a Service (PeaaS)[2] is a mobile-centric computing model that allows a user's sociological profile to be generated, kept, and securely provided as a service to third parties directly from a smartphone.

PeaaS emphasizes smartphones' capabilities and relies on them for inferring and sharing sociological profiles. These profiles are kept on the device, making it easier for owners to keep their virtual identity under their own control while still allowing external systems to consult those identities.

Serving individuals' virtual sociological profiles through smartphones goes one step beyond other mobile-centric models that only serve data, such as GPS coordinates and temperature. PeaaS allows a variety of information to be collected, such as the moods, trends, social statuses, and health habits of a group of people, to define their digital projection. However, filtering and analyzing this information to infer user's characteristics or to extract useful data isn't a trivial task. Different techniques, including activity-recognition approaches[3] and affective computing,[4] are considered in PeaaS for building the richest sociological profile possible.

Current mechanisms used to specify the orchestration of "smart" things could take advantage of the final user's sociological profile to suggest (or even automatically provide) a customized IoT workflow in accordance with the user's likes and preferences. Similarly, in events in a smart-city context, the set of sociological profiles extracted from a crowd could be applied to automatically redirect people so as to avoid unnecessary agglomerations.

The PeaaS model is currently implemented as a component of a mobile notification platform named nimBees (www.nimbees.com), which provides smart push notifications with advanced segmentation capabilities based on a user's sociological profile.

References

1. N. Mäkitalo et al., "Social Devices: Collaborative Co-located Interactions in a Mobile Cloud," *Proc. 11th Int'l Conf. Mobile and Ubiquitous Multimedia*, 2012, article no. 10.
2. J. Guillen et al., "People as a Service: A Mobile-Centric Model for Providing Collective Sociological Profiles," *IEEE Software*, vol. 31, no. 2, 2014, pp. 48–53.
3. P. Gupta and T. Dallas, "Feature Selection and Activity Recognition System Using a Single Triaxial Accelerometer," *IEEE Trans. Biomedical Eng.*, vol. 61, no. 6, 2014, pp. 1780–1786.
4. Y. Ma et al., "Daily Mood Assessment Based on Mobile Phone Sensing," *Proc. 9th Int'l Conf. Wearable and Implantable Body Sensor Networks*, 2012, pp. 142–147.

so that interactions are perceived as taking place between people, and not between things, connected to the Internet.

IoP Manifesto

Because current IoT technology needs people-centric enhancements, we've made our proposal a step in this direction. It conforms to a set of features we believe are essential foundations for any approach to the IoP. In the following, we put forward four guidelines for an IoP manifesto, in terms of the desirable service composition goals for any pervasive computing context (context awareness, contingencies management, heterogeneity, and empowering users), as set out in the work of Jeppe Brønsted and his colleagues.[10]

Principle 1: Be Social

Interactions in which things, devices, and people participate must be social. In particular, the IoP should allow for heterogeneity by supporting the different types of devices that people use, and let them interact with each other and with people more socially than does the IoT. Devices will have to be context-aware and able to automatically adapt their own and other devices' social behavior. Users need to be empowered to adjust their preferences and policies about when and with whom their devices are socializing, and with which modalities.

Principle 2: Be Personalized

Interactions between devices must be personalized to users' sociological profiles and contexts, allowing for contingencies and providing a transparent mechanism for this customization. The IoP must consider the sociological profiles of all participating people. Again, users must be empowered to adjust their preferences to control how other IoP stakeholders use their profile.

Principle 3: Be Proactive

The triggering of interactions must be proactive, not manually commanded by the user. Most IoT scenarios today only consider remote interfaces for managing connected things. But ever-more things online mean more distractions and work in managing them. The IoP should allow for device heterogeneity so that they can all interact more proactively. Users must be empowered to adjust their preferences to control how proactive their surrounding things are. Note that letting devices act proactively could entail security risks that we must analyze and delimit, keeping users informed of any established proactivity policy.

Principle 4: Be Predictable

Interactions must be predictable — that is, they should be triggered according to a predictable context that the user has previously identified, and for which a specific behavior has been defined. Users must be empowered to identify and tag that context, specify the expected behavior of the things involved, and set the privacy policies for sharing their information by being advised of what information they're sharing and with whom. Given that complete predictability of interactions is hard to achieve, the user must always understand how the interaction can be stopped immediately (by, for instance, whacking your phone)[11] and also how to prevent this misbehavior in the future.

IoP Blueprint

We designed a reference architecture supporting our vision of the IoP, based on the Social Devices and PeaaS platforms. The result is a more human-like and less user-dependent IoT ecosystem. The architecture is a natural consequence of the objectives set out in the IoP manifesto. To illustrate our point, let's consider a particular case study scenario.

Scenario: Smart Transportation

David lives in a residential suburb of Paris. Today, like every day, he's driving his daughter to school before he goes to work. His smartphone has learned where he lives, where he works, and the route he usually takes and at what time. The phone routinely reports this information to the city's transportation control systems. David is happy to anonymously contribute such information, because it's used for simulations and previsions of potential traffic problems, and to plan improvements in the city.

David has no need to interact with his smartphone because, when he left home this morning, it already knew where he was going. This was confirmed when it detected that the route and speed were as usual. Unfortunately, a traffic accident just occurred, and David is stuck in a jam. His smartphone detected an abnormally low speed at a certain point in his route. It asks the smartphones of people nearby whether they're stuck, too. On confirmation, it alerts the transportation control systems of a possible incident. Because every smartphone is reporting the same issue, the control system raises a traffic alert and notifies the smartphones of people who usually take the same route, suggesting an alternative. David's smartphone is now receiving information about the new route to follow and the estimated time of arrival. It then passes the new route to the car navigator, which immediately informs David about the best way out of the jam. The smartphone knows that David is now late, and reports to his office, indicating his expected arrival time.

Most of the people behind David manage to avoid the jam, and David arrives only 10 minutes late. His unhappiness about the congestion is mild, because he recalls the days when smartphones were only used to talk, read emails, and surf the Web during traffic jams. He's also pleased that he contributed to reducing the traffic jam's impact on his neighbors.

IoP Middleware

Figure 1 shows a high-level architectural description of our proposed IoP middleware, conceived as a service-oriented system comprising various components: a device registry, application manager, application repository, and action repository.

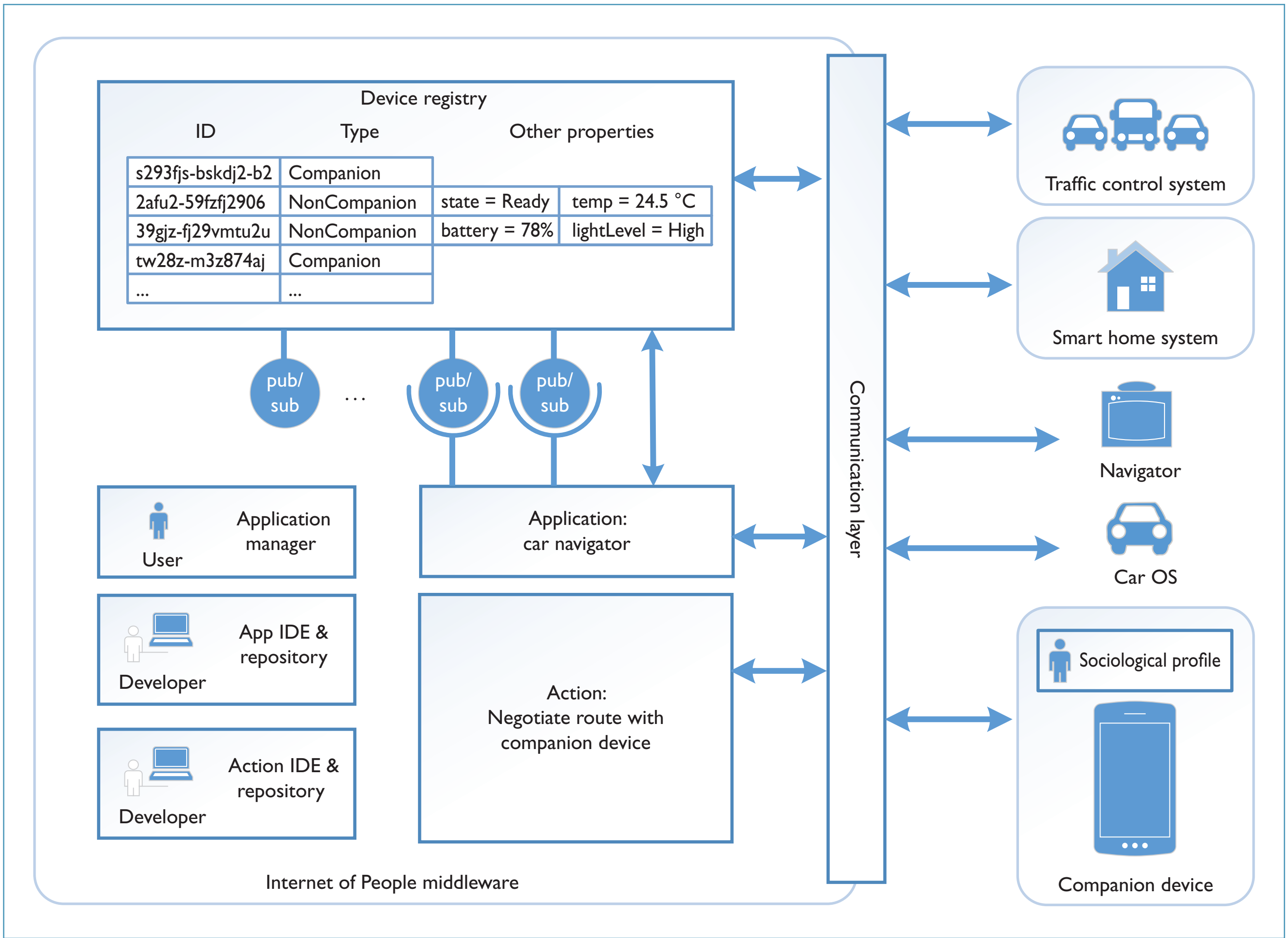

Figure 1. Our proposed Internet of People (IoP) middleware. The service-oriented system contains various components, including a device registry, application manager, application repository, and action repository. (IDE = integrated development environment.)

The central component is the device registry. This maintains the information about the different devices the system manages. These devices are classified into Companion and NonCompanion.

Companion devices are those that maintain contextual and sociological information about their owners, and serve that information by querying the sociological profile component inside them. In the illustrative scenario, David's smartphone is registered as a Companion device in the IoP context to allow interactions with other things in that context, such as surrounding people's smartphones or his car navigator. Identifying smartphones as Companion devices in the IoP registry allows them to be considered as representations of people in the IoP context, and, if they're socially capable devices, as conforming to the manifesto's "be social" principle.

NonCompanion devices are those without a sociological profile component. They're "things," which can register their capabilities and any additional information about their state, including any sensing properties and values. The registry incorporates a publish-subscribe API mechanism to allow monitoring the registered devices' state and property changes. In our scenario, things like car navigators and traffic control are registered as NonCompanion devices. They publish their properties' values in the registry to allow the IoP system to take the appropriate measures according to monitoring rules, such as alerting the smartphones of people close to a traffic jam. This behavior conforms to the manifesto's "be proactive" principle.

The action repository component manages the middleware's set of actions. Each action defines how the devices should interact with each other and with people. The application repository component stores the different applications defined in the IoT context and managed by the IoP middleware. An application consists of a set of instructions defining under what conditions to trigger a set of actions and which devices will be involved. An application's triggering conditions can be defined by monitoring the changes in the state of one or more devices using the registry's publish-subscribe API protocol. In the smart transportation scenario, that mechanism lets an application be written to monitor the state of traffic reported by the traffic control system, and to trigger appropriate actions to redirect people in the traffic jam.

The application manager component lets people enable or disable the different applications according to whether they would like to be present and contribute with their sociological profiles.

The application and action specifications might include requests to Companion devices' different sociological profile components, so that the specified behavior can adapt to the user's context, personality, or mood.

Sociological Profile

The sociological profile component is a fundamental part of every Companion device. This component maintains the information that has been gathered and inferred about people and their contexts. This information covers people's behavior and preferences, as well as raw contextual data such as current or historical locations, sensing information such as accelerometer status, and processed contextual information such as a proximity graph of surrounding people/devices/things. In the smart transportation scenario, David and other users' Companion devices detected a delay in their route on the basis of their location, movement information, and expected arrival time according to their sociological profiles' behavioral pattern data.

This component serves information in response to requests via a query-based service. When it receives a request for information about some property stored in the sociological profile, the Companion device answers it, providing that information as a service response. Companion devices can also notify the registry about significant changes in their sociological profile, so that subscribed applications will be notified when an on-change event occurs, letting them rerun the queries to get freshly updated information. In the smart transportation scenario, David's smartphone uses this mechanism to ask nearby people's smartphones for verification of the traffic state.

The component also lets users define and adjust the sharing schema for the profile. Users should define what information will be available for which applications, devices, or users, and in which context the information should be provided. In the smart transportation scenario, David's smartphone shares his daily route with his workplace, but makes it inaccessible for other systems. Such privacy rules align this approach with the manifesto's "be predictable" principle.

Finally, the sociological profile provides an interface for device owners to personalize how the device behaves under specific circumstances. In the smart transportation scenario, when the traffic control system notifies people's smartphones about the traffic jam on their route, their profiles can mold the resulting alarm action by translating the message into their native language and shaping the voice to the proper accent, timbre, or mood according to their profile information and context. This conforms with the manifesto's "be personalized" principle.

Most of the profile's customizable policies and preferences will be supported by a user-friendly wizard. The component will progressively learn the customization during the device's use from the interactions and contexts in which it's involved, and from the owner's decisions about how the device should behave in those interactions. In the smart transportation scenario, David's device recognized a previously learned pattern about his daily driving routine, which let it detect deviations and react accordingly. This behavior also conforms to the manifesto's "be predictable" principle.

Building Proactive Interactions

Many IoT and pervasive service systems have their own communication protocols and ways of interconnecting objects. But they typically overlook how the "things" interact with people. To construct more human-like, predictable, proactive, and social interactions between people and "things," the IoP middleware offers a Web-based integrated development environment (IDE) and tools. These tools let applications be constructed to monitor user context, and then, based on the context, proactively trigger interactions between the people and "things" involved.

Two examples in the smart transportation scenario might be that when the Companion device receives an alert, it can read it aloud for the owner while he drives, and that the smartphone and navigator can inform David while rerouting the way to work, so that he can be aware of what's going on. We describe these tools, and the development effort for building new devices and applications, in more detail elsewhere (see http://orchestratorjs.org).[12]

The success and growth of the IoT is unquestionable. However, people's interaction with this kind of system is still far from friendly. Here, we presented a reference architecture to smooth out how people relate with IoT systems. By basing people's interactions with the system through their smartphones, our proposal reduces the complexity inherent in the IoT, contributing to evolution towards a true IoP.

The term "Internet of People" has been used before, but usually to refer to traditional Web systems designed only for humans to use. Here, we use it in the sense of bringing the IoT closer to people, for them to easily integrate into it and fully exploit its benefits. Technologies such as Siri for iOS or Sherpa for Android have endowed smartphones with a kind of persona. The next step could be to adjust this persona to fit the user's personality and mood.

This potential use of smartphones as personal information gatherers and managers, and conductors of orchestras of other devices, has also motivated other research initiatives. In addition to the examples we noted in the sidebar, "IoT-Linked Smartphones" (Social Devices and PeaaS), in Mark Jansen's work,[13] smartphones are seen as service providers, with an implementation based on Web service standards interfacing the user with other smartphone services. This could provide a mechanism for serving user's contextual information, but the management of IoT devices, as provided by the IoP approach, is out of its scope. Paraimpu (www.paraimpu.com) provides a social tool for people to connect, compose, and share things, services, and devices to create personalized IoT applications. Node-RED (www.nodered.org) provides a Web-based visual interface for connecting and building compositions of hardware devices, APIs, and online technical and social services. Both work well with IoT management, but they lack the predictability and sociological profile features provided in our IoP approach. Apple HomeKit promises integration with a large set of compatible devices, and managing them remotely, but it's also a platform-specific solution.

Hopefully, our manifesto encourages other IoT system developers and researchers to pay attention to the relation between people and IoT systems. Currently, Orchestrator.js and nimBees are consolidated implementations that complement each other in our IoP middleware. Later, other systems (such as recommendation systems) can be applied to our approach.

The maturity of our previous work (Social Devices and PeaaS) allows supporting interactions between people and household devices based on pattern recognition applied to smartphone usage. For example, now we can turn on a coffee machine when a user starts her journey to work. Implementing scenarios, such as the smart transportation one, helps address possible scalability issues.

The major privacy and socially related questions that proposals such as ours raise will set a new horizon for research in the coming years. Our own future work plans focus on taking advantage of smartphones' ever-increasing capabilities to improve how they make inferences about their owners, and to use that information to enhance owners' roles in IoT systems.

Acknowledgments

This research has been partially funded by the Spanish government (projects TIN2012-34945, and TIN2012-35669), the Academy of Finland (264422), and by the Nokia Foundation. It has also been funded by the Government of Extremadura and the European Regional Development Fund (FEDER).

References

1. L. Atzori, A. Iera, and G. Morabito, "From 'Smart Objects' to 'Social Objects': The Next Evolutionary Step of the Internet of Things," *IEEE Comm.*, vol. 52, no. 1, 2014, pp. 97–105.
2. J. Gubbi et al., "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, vol. 29, no. 7, 2013, pp. 1645–1660.
3. N. Mäkitalo et al., "Social Devices: Collaborative Co-located Interactions in a Mobile Cloud," *Proc. 11th Int'l Conf. Mobile and Ubiquitous Multimedia*, 2012, article no. 10.
4. J. Guillen et al., "People as a Service: A Mobile-Centric Model for Providing Collective Sociological Profiles," *IEEE Software*, vol. 31, no. 2, 2014, pp. 48–53.
5. "Cisco IoT Connections Counter," *The Network*, 2014; http://newsroom.cisco.com/feature-content?type=webcontent&articleId=1208342.
6. "Internet of Things Installed Base Will Grow to 26 Billion Units By 2020," Gartner press release, 2013; www.gartner.com/newsroom/id/2636073.
7. J.-L. Gassée, "Internet of Things: The 'Basket of Remotes' Problem," *Monday Note*, 12 Jan. 2014; www.mondaynote.

com/2014/01/12/internet-of-things-the-basket-of-remotes-problem.
8. *Devices Profile for Web Services Version 1.1*, Oasis standard, 2009; http://docs.oasis-open.org/ws-dd/dpws/1.1/os/wsdd-dpws-1.1-spec-os.html.
9. M. Kovatsch, "CoAP for the Web of Things: From Tiny Resource-Constrained Devices to the Web Browser," *Proc. 2013 ACM Conf. Pervasive and Ubiquitous Computing Adjunct Publication*, 2013, pp. 1495–1504.
10. J. Bronsted, K. Hansen, and M. Ingstrup, "Service Composition Issues in Pervasive Computing," *IEEE Pervasive Computing*, vol. 9, no. 1, 2010, pp. 62–70.
11. J. Lyon et al., "Controlling Audio of a Device," US patent application 13/046,680, 13 Sept. 2012; www.google.com/patents/US20120231838.
12. N. Mäkitalo, "Building and Programming Ubiquitous Social Devices," *Proc. 12th ACM Int'l Symp. Mobility Management and Wireless Access*, 2014, pp. 99–108.
13. M. Jansen, "About Using Mobile Devices as Cloud Service Providers," *Proc. 2nd Int'l Conf. Cloud Computing and Services Science*, 2012, pp. 147–152.

Javier Miranda is a software engineer, researcher, and CEO of Gloin. His research interests include mobile and cloud computing. Miranda has an MSc in computer science from the University of Extremadura, where he's currently a PhD candidate. Contact him at jmiranda@gloin.es.

Niko Mäkitalo is a PhD candidate and researcher at Tampere University of Technology. His primary focus of research has been on mobile devices, cloud software, and pervasive systems, with a particular focus on coordinating IoT devices and enabling humans to interact with these devices in a more human way. Mäkitalo has an MSc in software engineering from Tampere University of Technology. Contact him at niko.makitalo@tut.fi.

Jose Garcia-Alonso is a cofounder of Gloin and an assistant professor at the University of Extremadura. His main research interests include product lines, software architectures, and model-driven development. Garcia-Alonso has a PhD in computer science from the University of Extremadura. Contact him at jgaralo@unex.es.

Javier Berrocal is a cofounder of Gloin. His main research interests are software architectures, model-driven development, and business processes. Berrocal has a PhD in computer science from the University of Extremadura, where he's currently an assistant professor. Contact him at jberolm@unex.es.

Tommi Mikkonen is a professor of software systems at the Tampere University of Technology. His research interests include software architectures, distributed systems, and mobile and Web software. Mikkonen has a PhD in computer science from the Tampere University of Technology. Contact him at tommi.mikkonen@tut.fi.

Carlos Canal is an associate professor in computer science from the University of Málaga, Spain. His research interests include component, service, and cloud development, in particular formal methods for the specification, composition, and adaptation of interacting components. Canal has a PhD in computer science from the University of Málaga, Spain. Contact him at canal@lcc.uma.es.

Juan M. Murillo is a cofounder of Gloin and is an associate professor of software engineering at the University of Extremadura. His research interests include software architectures, mobile computing, and cloud computing. Murillo has a PhD in computer science from the University of Extremadura. Contact him at juanmamu@unex.es.