Building Software for the Internet of Things

Luciano Baresi
Politecnico di Milano, Italy

Luca Mottola
Politecnico di Milano, Italy and SICS Swedish ICT

Schahram Dustdar
Vienna University of Technology, Austria

The next computing revolution is about to happen. We will equip persons, places, and everyday commodities with networked embedded devices sensing from individuals and the surrounding environment, and acting on the physical world to accomplish high-level goals. As these devices coordinate among themselves and integrate with existing computing infrastructures, We will create an Internet of Things (IoT) that improves society and our quality of life.

Along the way, should visions such as 50 billion Internet-connected embedded devices become a reality (see http://goo.gl/tcMP), We will need to radically revisit a significant fraction of current software development techniques.[1] The sheer number of computing units, the systems' heterogeneity, the need for integrating into larger computing infrastructures, and the increasing demand for decentralized control loops will challenge the way we currently build these kinds of distributed systems.

Tackling Challenges in a Vast Landscape

Currently, researchers in diverse disciplines are attacking these problems from very different perspectives. For example, networking researchers are focusing on how to enable Internet Protocol (IP)-based communication on severely resource-constrained embedded devices;[2] the control community is devising decentralized control loops battling the woes of wireless communications;[3] and middleware research is tackling the previously unseen scalability problems that the IoT brings about.[4] These efforts will ultimately provide the necessary building blocks towards a global IoT.

How to combine such building blocks in the design, implementation, and validation of IoT software is, however, still largely unclear. This is a direct effect of the technology's interdisciplinary nature. Publications documenting the individual building blocks are scattered across different scientific communities. Achieving a meaningful view over this vast landscape is thus difficult.

Even if such a view could be obtained and novel solutions designed to address the related problems, such works tackling "the whole problem" in the design, implementation, and validation of IoT software would hardly find

a spot within individual research communities, which tend to favor specialized topics. The scientific spread of new ideas in this field — and industry adoption of the corresponding solutions — are thus ultimately hampered, because of the lack of proper communication with and towards interested parties.

Gathering Solutions Here

This special issue of *IEEE Internet Computing* provides an ideal forum to gather contributions from different research communities and addresses eminently interdisciplinary topics on building IoT software. It caters to a unique opportunity of fostering scientific discourse across disciplines, and offers a cornerstone for others to build upon in future research efforts.

Building the IoT is currently of topical interest, and because of its popularity, initially we received a record number of 37 submissions, which required a thorough two-phase evaluation process. The first three reviews for each paper helped us filter the submissions and identify a first set of 10 potential candidates. The authors revised their work and the second round of reviews allowed us to select the five articles that are part of this theme issue. The fierce competition led to some sad decisions — we had to reject some good papers — but the final result provides a good summary of the different activities the research community is carrying out to address the many challenges that come with crafting IoT software.

Thinking of the software behind the IoT, one may think of diverse functionality, ranging from low-level embedded components as well as communication and control platforms, up to user-oriented applications. This means that the five selected articles cannot provide an in-depth presentation of every single facet; instead, they can only focus on some particular aspects.

In "Thingsonomy: Tackling Variety in Internet of Things Events," Souleiman Hasan and Edward Curry propose an event-based middleware. They argue that "smart" event processing can contribute to filling the gap between application layers and IoT infrastructures that comprise billions of devices and create large-scale, dynamic, heterogeneous, and open environments. The semantic coupling among events could hamper the resulting system's scalability, but the authors propose an innovative solution based on the idea of approximate semantic matching.

Besides a connected network of interacting things, application developers need proper abstractions to ease their work and to deliver quality results to users. Two of the articles deal with this issue and propose different solutions. In "A Vision of Swarmlets," Elizabeth Latronico and her coauthors explain how to create *swarmlets* — that is, applications that integrate networked sensors and actuators, cloud services, and mobile devices. Swarmlets are compositions of accessors, which in turn are wrappers for sensors, actuators, and services. An actor semantics helps associate the composition with a disciplined and understandable concurrency model. Accessors hide the details of the many mechanisms defined for sensing data, controlling actuators, or accessing services, and provide a homogeneous abstraction layer.

Michael Mrissa and his coauthors, in contrast, stress the importance of the Web as a gluing solution and propose a Web of Things architecture in their article "An Avatar Architecture for the Web of Things." They propose avatars as new virtual abstractions to extend physical objects on the Web. In this work, avatars are extensible, distributed runtime entities with autonomous behavior; semantic annotations help provide their functionality as Web services, exploit their capabilities, and foster the cooperation among them to accomplish complex tasks.

These abstractions can help create quality applications, but the interactions with possible users must be taken into account explicitly. This is the motivation for Javier Miranda and his coauthors to present a reference architecture in "From the Internet of Things to the Internet of People." Their claim is that IoT solutions should be means to help people live their lives and carry out their tasks faster and with better results. Such solutions should be commodities and cannot become obstacles that impose further complexity. Thus, the interactions between humans and things must be mediated properly and the user should always be at the ecosystem's center.

Orthogonal to the articles just introduced, the fifth article is about the problem of creating software for working with and extending existing deployments and test beds. In "Integrating Smartphones into the SmartSantander Infrastructure," Georgios Mylonas and his coauthors introduce SmartSander — one of the largest

existing testbed infrastructures for the IoT — deployed in the city center of Santander, Spain. The authors discuss how to augment the existing infrastructure with the sensing capabilities of Android smartphones, and they present solutions to design and to deploy experiments that exploit the features of both the existing infrastructure and the additional capabilities provided by the volunteers' smartphones.

This issue primarily addresses the creation of complex IoT software. The aim is to provide the reader with knowledge on the state of the art for creating and operating these systems through sound engineering principles. To the best of our knowledge, we are unaware of other prime venues that have already considered reporting on these topics, and we hope this special issue could become an essential reference for the community.

References

1. F. Kawsar, G. Kortuem, and B. Altakrouri, "Supporting Interaction with the Internet of Things across Objects, Time, and Space," *Proc. Internet of Things Conf.*, 2010; doi:10.1109/IOT.2010.5678441.
2. J.W. Hui and D.E. Culler, "IP Is Dead, Long Live IP for Wireless Sensor Networks," *Proc. ACM Conf. Embedded Network Sensor Systems*, 2008, pp. 15–28.
3. J. Stankovic et al., "Opportunities and Obligations for Physical Computing Systems," *IEEE Computer*, vol. 38, no. 11, 2005, pp. 23–31.
4. A.B. Hamida et al., "An Integrated Development and Runtime Environment for the Future Internet," *The Future Internet*, LNCS 7281, 2012, pp. 81–92.
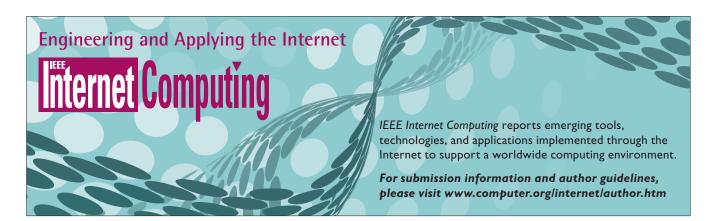
Luciano Baresi is an associate professor at the Politecnico di Milano, Italy. His research interests include dynamic software systems, service-oriented applications, and self-adaptive and autonomic software systems. He's currently on the editorial boards of *ACM Transactions on Autonomous and Adaptive Systems, IEEE Transactions on Services Computing*, and *IEEE Transactions on Software Engineering*. Baresi has a PhD in computer science from the Politecnico di Milano. Contact him at luciano.baresi@polimi.it.

Luca Mottola is an assistant professor at the Politecnico di Milano, Italy, and a senior researcher at the Swedish Institute of Computer Science, Sweden. His research focuses on designing, building, and validating networked embedded software to power the operation of systems such as wireless sensor networks, cyber-physical systems, and the Internet of Things. Mottola has a PhD in computer engineering from the Politecnico di Milano. Contact him at luca.mottola@polimi.it.

Schahram Dustdar is a full professor of computer science (informatics) and he heads the Distributed Systems Group at the Vienna University of Technology. His work focuses on Internet technologies. Dustdar is a member of the Academy Europeana, an ACM Distinguished Scientist, and recipient of the IBM Faculty Award 2012. Contact him at dustdar@dsg.tuwien.ac.at; dsg.tuwien.ac.at/.

cn *Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.*