Editorial: End-User Development for the Internet of Things

The Internet of Things promises to put within arm's reach a vast network of connected computerized devices, from sensors and actuators to more traditional devices, such as phones and tablets. Research in this field has been primarily focused on technical considerations, such as how devices will connect with other devices and how those connections will be kept secure. Relatively, less attention has gone into improving the usability of these constellations of devices for people using them who live and act within the dynamic technological habitat they provide. We do not yet know how best to help users to harness the potential power of these large collections of devices to accomplish their tasks. The Internet of Things is likely to transform these tasks by enhancing our physical environment and the objects that surround us with sensing and computational capabilities, which, in turn, is likely to result in new activities and needs for people. Already we are witnessing a dramatic increase in the data that can be collected from the physical environment with individuals and businesses seeking new opportunities and inventing new ways to use computation in close relation to people's physical environment and activities. These tasks, needs, and opportunities and even potential threats for people, are very much contextualized and likely to evolve over time to address requirements that are often unknown to technology developers *a priori* and which may be very specific to individual users.

The need to let users control and shape their environment is not endemic only to the Internet of Things. It has been a long-standing ambition for research in intelligent Environments, Pervasive Computing, and Ambient Intelligence. However, the Internet of Things brings about several new challenges and scales up old ones. Not only will most related systems be highly complex while remaining opaque, but each system is also likely to differ from every other in terms of device combinations and the setting in which those devices operate. Furthermore, users are also different and will have different use cases that they wish to enable with their systems of Internet of Things devices. Traditional design and development models will be difficult to employ in such an environment, where there is neither a common computational substrate on which to build nor a common set of use cases for which to design. Going a step further than user-centered and participatory design approaches, end-user development addresses this challenge by empowering non-technical users to assemble systems of devices to do their bidding without requiring any knowledge of typical computer programming. End-user development offers a potential solution in that end-users themselves could be enabled to define and tailor application functions, in order to satisfy their needs in the context of their devices and environment.

End-user development is a long-standing research area within the field of human-computer interaction. As a concept and as an area of inquiry it emerged in the era of the personal computer searching for ways to empower end-users to create their own applications. Solutions like spreadsheets, visual languages, macros, and programming by example have been studied intensively and many of these are now so wide spread that they can be considered general information technology skills that most users are expected to develop. However, the range of application of end-user development research is also shifting and considering challenges increasing in complexity and diversity. End-user developers can be hobbyists creating and programming robots, home

1073-0516/2017/04-ART9
DOI: http://dx.doi.org/10.1145/3054765

owners configuring smart objects to control their environment, or knowledge workers composing available web applications into mashups. They can also be professionals without training in computer programming who create tools that they can use in their practice.

In light of the trends and challenges discussed above for the Internet of Things, we turn to the field of end-user development to look for ways to help people control and benefit from the Internet of Things, and to become active contributors and co-creators of the technological landscape in which they live.

This special issue gathers together contributions related to end-user development for Internet of Things applications. They present not only innovative directions for the programmatic control of these environments, but also attempt to deepen our understanding into who end-user developers may be and their changing relationship to the artefacts they create.

Perhaps the most common programmatic control offered to end-users for Internet of Things applications is through specifying rules. However, it is by no means self-evident that people's most natural and effective means of reasoning is through specifying rules. Visual languages where users can describe compositions of services and objects, involving control or data flows, are often held as a more natural and accessible way of programming for the non-trained programmer. This special issue presents contributions in both these directions of inquiry.

Focusing on the data flow between different Internet of Things objects and services, Akiki et al. address the need to support end-users in mapping the output from object to the inputs of another. Rules describing data flows and contingent behavior of the system are composed visually with a drag-and-drop interface supporting a jigsaw metaphor.

The article by Brich et al. compares a rule based to a diagrammatic-process-oriented expressions of computational behavior for end-user programming of home-automation. They present an experiment that compared people's attitudes towards the two approaches and the complexity of the device configurations they create using paper mockups. Their results suggest that the diagrammatic process models are more expressive.

Desolda et al. examine closely event-condition-action rules aiming to extend their expressive power for non-programmers. Following a user-centered design approach they prototype three alternative user interfaces for specifying such rules that were proposed by their study participants during an elicitation study. A comparative evaluation regarding performance and satisfaction suggests that higher performance and satisfaction may be achieved by increasing complexity compared to currently popular systems such as IFTTT that use rules involving simple events. They argue for the need to support simulations and programming support tools. In contrast to *Birch et al.* their evaluation suggests that visual representation of component composition in graphs is not helpful for non-programmers.

Metaxas and Markopoulos delve into rule-based reasoning about context, aiming to increase the complexity of the rules that end-users can create and comprehend. They introduce an editor that allows specification of context sensitive behaviors using constrained natural language and present design heuristics for enabling the intuitive expression of rules that are founded upon users' mental models.

Ghiani et al. present a method and a set of tools that allow end-users without programming experience to personalize the context-dependent behavior of their Web applications through the specification of trigger-action rules. The authoring environment is able to support end-user specification of flexible behaviors, and is integrated with a middleware able to detect the events generated by the various sensors and devices in such a way to indicate the rules that can be immediately executed.

An alternative to the rules or visual notation approach is the idea of programming by demonstration. Chen and Li explore this approach to support end-users in composing on-the-fly cross-device interfaces to an existing application. Their cross-device input framework called Improv was tested with end-users who were able to construct arbitrary compositions as efficiently as programmers and who formed positive attitudes as to the potential usefulness of this approach.

Looking further than software composition, Sas and Neustadter raise the question of how people engage with Internet of Things objects. Their qualitative inquiry examines the creation, adaptation, use, and adoption of a smart energy monitor by two communities of hobbyists thus identifying key qualities that enable a transformation of Internet of Things objects from unremarkable stuff belonging to the background of human activity into meaningful things, to which meaning and value is attached. They describe these properties as transparent modularity, open-endedness, heirloom, and disruptiveness.

With the Internet of Things, the material characteristics of programming environments become more relevant and influence the way groups of users can engage in mutual support for their end-user development intentions. Ludwig et al. illustrate the challenges of configuring a production line for 3D printing to describe how additional sensors and actuators allow end-users to explain and interact on material aspects of their printing practices and problems. Their design approach of "Sociable Technologies" aims to integrate user-friendly collaboration functionality for expressing and communicating functional states as well as the relevant socio-material context into IoT technologies by the time we build them.

In summary, this special issue reveals the picture of a fast-evolving field. The availability of Internet of Things technologies is increasing dramatically, and end-user development practices are spreading to meet the real needs of Internet of Things users. Internet of Things technologies are not an abstract idea anymore, they are now technologies that *we use,* technologies *we configure*, and even technologies that *we develop* to match their final use cases. As users start living *in the interface* the technologies provide, we need to allow them to learn how to live in the combined programming interfaces of their technologies. The articles presented in this special issue demonstrate that end-user development is dedicated to not only addressing the idiosyncratic needs of end-users and to ease appropriation, but to also be a vehicle for creating value and meaning for users and redefining the relation of people with their physical environment.

Panos Markopoulos
Eindhoven University of Technology

Jeffrey Nichols
Google

Fabio Paternò
CNR-ISTI

Volkmar Pipek
University of Siegen