Guest Editors’ Introduction: Software Engineering for the Wireless Internet

Maurizio Morisio, Member, IEEE Computer Society, and Markku Oivo

In the recent years, the wireless Internet has become reality. Access is wireless via personal (such as Bluetooth), local (Wi-Fi), or global (cellular) networks. A variety of mobile terminals can be used, PC-like (PDAs, laptops, tablet PCs), or phone-like (smart phones, mobile phones). Several applications are being and will be developed, either to adapt existing ones to the new infrastructure, or to support mobile services, such as travel information, roadside assistance, etc.

This scenario raises new engineering challenges, both at the connectivity and at the application level. This special issue deals with the latter, and stems from work done by the guest editors in the domain.

The key question is: What is the impact of the wireless Internet on building applications—or what is specific to software engineering for the wireless Internet, as pointed out in the title.

From an application developer point of view the challenges include:

- **Terminal characteristics**
  - Battery life, processing power, and storage capabilities are usually limited compared to traditional applications.
  - Interaction with the user is also limited by small screen sizes, reduced input devices (small or reduced keyboard, limited, or missing pointing device).
  - Bandwidth is reduced and quality of service is lower.

Despite the diversity and limitations, the ideal application should run on all terminals without specific adaptation.

- **Use of several different terminals by the same end user.** The same application should run on any terminal, with a reasonably similar look and feel.

- **Mobility of the end user.** Here, the application should adapt relevant functions to the geographical position. Further, mobility of the terminals should be transparent to the developer if she chooses so.

The answer to many of these issues is the concept of context awareness and adaptation, or the capability of an application to adapt to different terminals and/or to a changing geographical position.

Most of the papers we received deal with this theme, underlining its importance for the research community. We can recognize two research threads in the papers:

- **Middleware for mobility.** Traditional middleware supports the communication among applications or components resident on a node. With mobility of terminals and components from node to node new problems arise, for instance for communication, data replication, or synchronization. The trend is to extend middleware with new functions, or modify existing functions so that applications can abstract from mobility.

- **Adaptation of the application to the terminal.** This is a hard problem because it is strictly linked with the semantic of the application. A reduction in screen size may involve reducing the functionality offered, the content, and/or the sequence of interaction; effective adaptation requires working at the semantic level.

After the peer review process, we selected three papers dealing with middleware for mobility, one dealing with adaptation, and two dealing with process issues. In the case of middleware extensions, the status of research aims at demonstrating the feasibility of the approach and at analyzing the computational overhead involved, with related performance issues.

M. Caporuscio, A. Carzaniga, and A.L. Wolf in the paper “Design and Evaluation of a Support Service for Mobile, Wireless Publish/Subscribe Applications” consider the basic publish and subscribe communication mechanism. They extend it so that the communicating parties can migrate on a network, while abstracting from issues related to mobility, such as migration of a party or intermittent connectivity. They have implemented and evaluated the mechanism on top of three distributed publish and subscribe systems.

The next two papers both propose extensions to the middleware to support the adaptation of an application due to dynamic changes in context.

A.T.S. Chan and S.-N. Chuang in the paper “MobiPADS: A Reflective Middleware for Context-Aware Mobile Computing” use an event notification model to make applications aware of context changes. An application can subscribe to a number of events (such as changes in battery
power, bandwidth, and quality of service). The middleware monitors the events and notifies the application. The application has the responsibility to react. The assumption is that the middleware alone does not have enough information to perform the adaptation; the application is in the best place to do it. Besides, MobiPADS offers the possibility to define events as composition of other events and applications as composition of lower-level services (or mobilets). Dynamic adaptation then takes the form of changing the composition of mobilets, changing the behavior of a single mobilet, or changing the composition of events.

P. Bellavista, A. Corradi, R. Montanari, and C. Stefanelli in the paper “Context-Aware Middleware for Resource Management in the Wireless Internet” propose an approach to handle data resources needed by a terminal that moves on a network. They use metadata to describe the context (description of the terminal, description of the user, and location of the terminal) and the policies to adapt the service to the context. Adaptation is made in terms of access control to the service and access to data resources (e.g., move a resource, create and move a copy, or access remotely). At provisioning time resources may migrate and the binding terminal—resources recomputed.

W.Y. Lum and F.C.M. Lau in the paper “User-Centric Content Negotiation for Effective Adaptation Service in Mobile Computing” present an approach to adapt the content of an application in function of the context (quality of service, device characteristics) and user preferences. The premise is that adaptation should match the user preferences and not be the best possible. A complex negotiation approach is used to take into account several conflicting decision factors. The approach is demonstrated on adaptation of PDF documents.

Finally, two papers deal with process issues, notably test of wireless applications and process selection.

Mobility makes testing applications an even harder task. Ideally, a mobile client should be tested in a variety of networking contexts, and this is complex and time consuming. In the paper “A Testing Framework for Mobile Computing Software,” I. Satoh proposes an approach based on substituting physical mobility with code mobility. A software agent emulates the mobile client and is capable of migrating on a network to encounter varied situations. The approach is demonstrated on two cases.

Project managers face a difficult situation in selecting a suitable process (in term of procedures, techniques, and tools) to develop wireless applications and services. Time to market pressure and lack of guidance in a new field are the key problems. A. Ocampo, D. Boggio, J. Münch, and G. Palladino in the paper “Toward a Reference Process for Developing Wireless Internet Services” propose an initial reference process issued from observing two processes and an extensive literature survey.

Before wishing to the readers to enjoy the papers in this special issue, the guest editors want to thank the IEEE Transactions on Software Engineering Editor-in-Chief John Knight and the IEEE Computer Society staff for their patience, support, and cooperation during the review process.

They would also like to thank the anonymous reviewers for their timely and accurate reports; their efforts certainly improved the quality of the special issue.

Finally, they thank colleagues and friends in the projects Torino Wireless, Monica, MOOSE, and IST WISE for their support and suggestions.

Maurizio Morisio
Markku Oivo
Guest Editors

Maurizio Morisio received the PhD degree in software engineering and the MSc degree in electronic engineering from Politecnico di Torino. He is an associate professor in the Dipartimento di Automatica e Informatica at the Politecnico di Torino, Turin, Italy. From 1989 to 1991, he worked as senior consultant at IGL Technology, Paris, France. From 1998 to 2000, he worked with the Experimental Software Engineering Group at the University of Maryland, College Park. During that time, he was the codirector of the Software Engineering Laboratory (SEL), a consortium of NASA Goddard Space Flight Center, the University of Maryland, and Computer Science Corporation, which has the mission of improving software practices at NASA and CSC. He is currently principal investigator of the IST-30028 project WISE, Wireless Internet Software Engineering, funded by the European Commission, DG-INFOSO. His research is aimed at understanding how software is produced and maintained, in order to improve software processes and products in industrial settings. His research interests include software reuse, software architectures, COTS-based development, open source development, and agile methodologies. He is member of the IEEE Computer Society. He has been a member of the international program committees of several conferences and symposia, including METRICS, ICSR, ICCBSS, and PROFES. He has (co)authored more than 40 papers in international journals and conferences and one book.

Markku Oivo is a full professor at the University of Oulu, Oulu, Finland. He has over 20 years of management and R&D experience in industry and academy in Finland, the US, France, and Germany. He has managed and initiated both national and international projects including EU consortia. He has also consulted EU in R&D planning, proposal evaluation and project reviewing. Until 2001, he was vice president of R&D and director of Solid Applied Research Center, with responsibilities of data management and synchronization in ubiquitous computing, network management including 3G networks, and Internet-based applications. Before joining SOLID in 2000, he was a research professor and the head of embedded software at VTT Electronics. He has worked at the University of Oulu 1981-1982, at Kone Co. in 1982-1986, and he has held visiting positions at the University of Maryland in 1990-1991, Schlumberger Ltd. in Paris in 1994-1995, and Fraunhofer IESE in 1999-2000. He has published more than 60 papers in international conferences and journals. He has served as chair and committee member in organizing numerous international conferences, and he is a founding member of the ISERN network. He has served as a reviewer for several international journals (including IEEE and ACM journals and magazines).