

Software Engineering Aspects of Ubiquitous Computing and Ambient Intelligence

This is the preface for the special section entitled “Software Engineering Aspects of Ubiquitous Computing and Ambient Intelligence” which gathers a selection of papers from the 5th International Symposium on Ubiquitous Computing and Ambient Intelligence (UCAmI 2011), held in Riviera Maya, Mexico, 5–9 December 2011.

This special section of Science of Computer Programming is devoted to papers selected from the 5th International Symposium on Ubiquitous Computing and Ambient Intelligence (UCAmI 2011), held in Riviera Maya, Mexico, 5–9 December 2011. UCAmI focuses on the Ambient Intelligence (AmI) research discipline, which represents a new generation of user-centred computing environments aiming to find new ways to obtain a better integration of information technology in everyday life devices and activities.

AmI environments are integrated by several autonomous computational devices of modern life ranging from consumer electronics to smartphones. Ideally, people in an AmI environment will not notice these devices, but they will benefit from the services these gadgets provide. Such devices are aware of the people present in those environments by reacting to their gestures, actions and context. The interest in AmI Environments remains strong due to new challenges posed by society, giving place to new interesting associated research disciplines such as Vehicular Ad hoc Networks (VANET), Ambient Assisted Living (AAL), e-Health, Internet of Things and Home Automation, among others.

This special action focuses on software paradigms, techniques, design patterns and models that are suitable for the development of the complex distributed, often embedded, software systems associated to AmI. So far, the focus has been placed on enabling functionality, and also on making new services and systems available to address the AmI vision, without paying too much attention on how well those AmI systems are produced from the software engineering perspective. This special action deepens in all those aspects, bringing together works addressing the need to take more seriously the software engineering methods and techniques to produce successful, high quality and responsive AmI systems.

The best papers of the conference addressing software engineering aspects of Ubiquitous Computing and Ambient Intelligence were selected by the programme committee after the conference. Extended versions of these papers were subjected to additional rounds of anonymous reviews. The revised versions of the selected papers are included in this special section, covering topics such as requirement analysis in ubiquitous systems, quality assessment in context modelling, simplifying and enhancing rule-based logic for system reactivity through more user-friendly graphical editors or declarative languages, middleware to enable the desired features of ubiquity and unobtrusiveness in ambient systems, and the implementation of presence awareness in ubiquitous applications supporting loosely-coupled mobile activities.

Recent technological advances are increasing the spread of Ambient Intelligence, leading to the appearance of numerous software systems, which benefit from the features of this new paradigm. Nevertheless, there is a lack of methodologies to properly support the development process of these systems. An important part of the Software Engineering lifecycle is the Requirements Engineering stage, as it grounds the success of system design. The paper *“REUBI: A Requirements Engineering Method for Ubiquitous Systems”* by Tomás Ruiz-López et al. describes a Requirements Engineering Method that supports the decision making process conducted by developers during the conception and design of Ubiquitous Systems. Such a method, called REUBI, provides an evaluation procedure to determine the satisfaction of non-functional requirements, e.g. services dynamicity and adaptation.

Context modelling is also another key part of ubiquitous applications and services. The design of such systems remains challenging as there are very few models and tools to guide application designers and developers in mastering the complexity of context information. This becomes even more crucial as context is by nature imperfect. The paper *“Building Ubiquitous QoC-Aware Applications through Model-Driven Software Engineering”* by Sophie Chabridon et al. addresses this issue by associating to context information, meta-data representing its quality. This work proposes a generic and extensible design process for context-aware applications, taking into account the quality of context (QoC). The cost of adding QoC management is measured using every-day mobile devices, equipped with multiple sensing capabilities.

A considerable part of the behaviour in smart environments relies on event-driven and rule specification. Rules are the mechanism most often used to enable user customization of the environment. However, the expressiveness of the rules available to users in editing tools is usually either limited or the available rule editing interfaces are not designed for end-users with low skills in programming. The paper *“A Meta-Model for DataFlow-Based Rules in Smart Environments: Evaluating User Comprehension and Performance”* by Alejandro Catala et al. describes a generic and flexible meta-model to support expressive rules enhanced with data flow expressions. This meta-model is used through a graphical user interface that allows the definition of rules without writing code. An empirical study is conducted on the ease-of-understanding of the visual data flow expressions for expressing user-customized reactive behaviour in Aml environments.

On the other hand, advances in wireless sensing and actuation technology allow embedding significant amounts of smart environments’ logic inside wireless sensor networks. Such active Wireless Sensor Networks (WSN) applications are more autonomous, but significantly more complex to implement. Event-based middleware usually supports the implementation of these applications, offering developers fine-grained control over how an individual node interacts with other nodes of the network. However, this control comes at the cost of event handlers, which lack composability and violate software engineering principles such as separation of concerns. The paper *“CrimeSPOT: a Language and Runtime for Developing Active Wireless Sensor Network Applications”* by Coen De Roover et al. describes a domain-specific language for programming WSN applications on top of event-based middleware. The node-centric features of this language enable programming node interactions through declarative rules

rather than event handlers. A key contribution of CrimeSPOT is its support for associating application-specific semantics with events that carry sensor readings.

Middleware in Aml typically isolates the end-user from the underlying hardware and software that enables smarter and adaptive environments. One of the features enhanced by Aml-enabled middleware is ubiquity, i.e. the continuous provision of services anywhere at any time. The paper "*Towards Ubiquity in Ambient Intelligence: User-guided Component Mobility in the HI³ Architecture*" by Alejandro Paz et al. focuses on the development of two elements related to ubiquity: (1) the physical migration of components between different platforms, together with their associated runtime state, and (2) the adaptation of those components to the target platform and physical environment. The article also presents a use case showing the mobility capabilities of the HI³ platform, and it also performs a comparison of the proposal with six other software mobility solutions.

The automation of user routine tasks is another important challenge in the development of Ambient Intelligence systems. However, this automation may be annoying since some tasks may grab users' attention in inappropriate situations. Since user attention is a valuable resource, task automation must behave in a considerate manner demanding user attention only when it is required. To address this issue, the paper "*Designing for User Attention: a Method for Supporting Unobtrusive Routine Tasks*" by Miriam Gil et al. presents a systematic method for supporting the design and automation of unobtrusive routine tasks, which can adjust their obtrusiveness level at runtime according to the user attention, considering the current available resources and context. This method allows designing the routine tasks that the system must carry out, and also how they must interact with users in terms of obtrusiveness. The method also provides a software infrastructure that determines the appropriate obtrusiveness degree for the execution of every routine task.

Most ubiquitous systems support mobile activities, particularly loosely-coupled work, in which mobile users conduct on-demand interactions among them and also with devices embedded in a smart environment. In this scenario, the capture and use of context information to provide presence awareness becomes a key design aspect. The current proposals to provide user presence awareness consider the use of centralized components, have reusability limitations or provide part of the required service. The article "*A Lightweight and Distributed Middleware to Provide Presence Awareness in Mobile Ubiquitous Systems*" by Rodríguez-Covili and Ochoa describes a lightweight and fully distributed middleware named Moware, that eases the task of embedding presence awareness services in ubiquitous applications. The design strategies to provide presence awareness are presented as a way to reuse such knowledge. Finally the article reports the results of an evaluation process that measures the usefulness and performance of the presence awareness services provided by middleware.

The proceedings of UCAmI 2011, where the original version of these papers appear, were published as a book with ISBN: 978-84-694-9677-0 by AIAM (Asociación para la Inteligencia Ambiental). We gratefully acknowledge the anonymous reviewers as well as the program committee members of UCAmI 2011 for their help in selecting the papers for this special section. Finally we would like to thank the referees for this special section for the high quality of their reviews.

Diego López-de-Ipiña*

MORElab, DeustoTech-INTERNET, University of Deusto, Bilbao, Spain

E-mail address: dipina@deusto.es

Sergio F. Ochoa

Computer Science Department at the University of Chile (Chile)

E-mail: address: sochoa@dcc.uchile.cl

José Bravo

Mami Research Lab – Castilla-La Mancha University (Spain)

E-mail: address: jose.bravo@uclm.es

* Corresponding editor.