J Reliable Intell Environ (2015) 1:1–10
DOI 10.1007/s40860-015-0005-3

EDITORIAL

Introduction to the inaugural issue of the *Journal of Reliable Intelligent Environments*

Juan Carlos Augusto[1] · **Antonio Coronato**[2]

Abstract As envisioned by Weiser, computing is in the process of being everywhere and becoming invisible. As Milner noticed, the question now is whether we shall understand this ubiquitous computer we are building. This is especially true as designers are more and more using complex techniques for every component of the system and building systems which are made of increasingly heterogenous parts. With this extended editorial, we embark on an exploration journey into the exciting new area of “reliable intelligent environments” (RIEs). Taking the perspective of an RIE engineer, we present a selection of approaches that have been put forward to design, verify, and operate IEs in a manner so that users can rely on Intelligent Environment systems. We outline crucial challenges: the situatedness which exposes IE to challenges similar to those known from robotics and control systems, the embedding of human users and the safety, privacy, and usability requirements thus entailed, and the amounts of data produced by sensors and actuators, which require advanced reasoning and learning mechanisms to handle them in a reliable way in real-time. We also sketch the opportunities reliable IEs provide to developing new markets and products.

Keywords Reliability · Intelligent environments · Sensing · Actuation software engineering

✉ Juan Carlos Augusto
j.augusto@mdx.ac.uk

Antonio Coronato
antonio.coronato@na.icar.cnr.it

[1] Department of Computer Science, Middlesex University, England, UK

[2] Institute for High Performance Computing and Networking, National Research Council, Naples, Italy

1 Introduction

In 1991 [56], Mark Weiser described a vision of the future in which computing would be on of the “profound technologies [which] weave themselves into the fabric of everyday life until they are indistinguishable from it”. Today, this vision is gradually becoming a reality not only with smart phones and tablets everywhere, but also with computing technology everywhere which is increasingly interconnected and connected to the internet. People are now relying on this invisible fabric like they rely on bridges and cars, but it was not until 2006 that Robin Milner asked “Ubiquitous computing: shall we understand it?” pointing at a wealth of questions ubiquitous computing raises as a field of computing in which systems cannot be isolated but form a single ubiquitous computer.

Within the last 25 years, ubiquitous or pervasive computing systems have become more and more complex with artificial intelligence techniques, with reasoning and learning now forming the core of their operation. The result is what we call *Intelligent Environments* (IEs) [4]. Intelligent Environments not only comprise smart homes and smart offices, but also smart cities, geosensor networks, or wearables. The hardware employed ranges from next generation smart materials to wireless sensor networks with distributed intelligence to smart phone networks.

Intelligent Environments [4] is growing fast as a multidisciplinary field allowing many areas of research to have a real beneficial influence in our society. Reliable Intelligent Environments are IEs that have the degree of maturity that is necessary to actually deploy them outside of laboratories.

The basic idea behind IE systems is that by enriching an environment with technology (sensors, processors, actuators, information terminals, and other devices interconnected through a network), a system can be built such that based on the real-time information gathered and the historical data accumulated, decisions can be taken to benefit the users of that environment.

Technology available today is rich. Several artifacts and items in a house can be enriched with sensors to gather information about their use and in some cases even to act independently without human intervention. Some examples of such devices are electro-domestics appliances (e.g. cooker and fridge), household items (e.g. taps, bed and sofa) and temperature handling devices (e.g. air conditioning and radiators). Expected benefits of this technology can be: (a) increasing safety (e.g. by monitoring lifestyle patterns or the latest activities and providing assistance when a possibly harmful situation is developing); (b) comfort (e.g. by adjusting temperature automatically); and (c) economy (e.g. controlling the use of lights).

This abundance of technology has given place to the new notion of *Smart Environments* (SmE). The notion of SmE extends to other environments and applications such as offices, hospitals, shopping malls, factories, roadways, and cities. "Ambient Intelligence" is the intelligent software which orchestrates these distributed devices in such a way that they provide appropriate services to users in a sensible way. Although Ambient Intelligence and Smart Environments are strongly related, we can distinguish them by going back to the old "mind/brain" metaphor used in AI. The first one is more concerned with the specific techniques to make an environment behave intelligently whilst the second one is more related with the intelligent interconnection of resources and their collective behaviour. Both overlap hugely and share many common objectives and it is difficult to tell apart one from the other. This is evident in the composition of topics in related conferences taking place around the world as these events compete to attract the attention of interesting research in both areas. Such interaction is a reason for us to propose a journal addressing both areas. We blend those two areas under the single term: *Intelligent Environments* (IE). Smart homes are currently the dominating force driving the area ahead; however, many other research projects are based on different applications of the concept to create smart offices, smart classrooms, smart cities, and to increase safety for drivers in a car or employees in a production line. For a more extensive coverage of the fundamental concepts and applications in this area see [8,17,44].

There is no doubt that these topics are new, attractive increasing attention and will reshape the world as we know it today. Due to the importance of the subject, there is a significant amount of ongoing research in the area. The underpinning conceptual frameworks, the technology, and thus the research market is ready for a global, multidisciplinary, and broad community to shape up and develop new technologies for bringing about reliable applications based on Intelligent Environments. Hence, we believe the current project of a journal in the area is timely and offers a needed forum for this growing community.

Systems in this area are made of a collection of preexisting technologies: sensors, actuators, networks, mobile technologies and other interfaces, and intelligent software, and this technology has to autonomously interact with humans successfully. Hence, it is understandable that despite the significant effort invested by companies and research centres around the world these systems still remain an engineering challenge, from the simple deployment and maintenance issues to the most core algorithmic challenges and system organization decision-making. Intelligent Environments are still predominantly built in labs instead of as commercial products massively consumed around the world.

This article is intended to highlight the challenges associated with building IEs and to describe some of the areas which can potentially help creating a more standardized methodology which can increase the level of confidence that all stakeholders have in the system being created. So far there is a lack of agreement on the methods and tools which are most effective to develop IEs and a debate is needed so that members from our scientific community can learn from each other and collaboratively distil a collection of good practices to improve industry in this sector.

Our article is organized around three main stages of development: the creative process of conceiving a system, the challenge of assessing correctness of the strategic vision for the system and all the issues associated with its operational viability. All of these have associated a number of more focused tasks which have been developed by previous researchers as a way to cope with the complexity of making a system effective. So for example, in the first section we look at the activities which help designing and defining the system. In the second section we look at methods developed more directly to achieve correctness of the main functionality and services the system is supposed to deliver. The last section considers a number of areas which are equally important in achieving an acceptable system as well as coping with the changes that the evolution of the system will bring.

2 System planning

Intelligent Environments are more and more frequently designed for critical domains like houses for frail and elderly inhabitants, hospitals, emergency scenarios, etc. These applications must be considered safety critical and, consequently, such criticality should be taken into account during the design phase. However, IEs present peculiarities that reliability engineers typically are not used to handle, i.e. system characteristics that cannot be effectively handled by means of traditional and well established reliability engineering methods: location, context and situation-dependent system behaviour. These characteristics are related to pre-existing concepts like situatedness [16]. An IE is both embedded in the physical world and embedding a portion of the physical world.

Among the dimensions of the physical domain, location has received the most attention so far, as location-awareness was among the first parameters of context being added to mobile applications to facilitate usage through "implicit interactions" [38]. Location, however, is a quite new concept to handle for reliability engineers. Campbell and Ranganathan [48] were among the first to argue for the need to formally specify and verify location-dependent software services and mobile entities in smart environments. They adopted *Ambient Calculus* (AC), a theoretical framework developed by Cardelli and Gordon [14] to model and analyse multi-agent and mobile agent-based systems.

Ambient Calculus relies on the concept of an Ambient, which is a bounded place that can (1) execute processes, (2) host other ambients, and (3) move a sibling ambient in/out. Indeed, AC provides some native operations to describe movements, replication, creation, and dissolution of ambients. Therefore, one could easily describe a sequence of movements of users and devices (both classes are modelled as ambients) within a Smart Environment and its locations (ambients) in terms of in and out operations. AC comes with *Ambient Logic*, a first-order logic that offers a specific construct (@) to allow the specification of properties and constraints related to specific locations.

Another formal approach useful for coping with the mobility of elements in IEs is the *Bigraphical reactive systems* (BRS). The BRS theory, due to Robin Milner [43], is based on a graphical metaphor for mobile computation that emphasizes both locality and connectivity. A bigraph comprises a place graph, representing locations of computational nodes, and a link graph. Dynamics is expressed by reaction rules that specify how bigraphs changes their structure whenever a specific condition is reached.

BRS has been adopted by Birkedal et al. [12] to define formally models of context. In particular, authors preliminarily attempted to model location-aware services naively in bigraphs. Successively, they proposed a model of context-aware computing called *Plato-graphical*, which comprises three bigraphical reactive systems: the context C; its representation or proxy P; and the computational agents A.

Roman et al. [50] proposed *Context UNITY*, derived from *Mobile UNITY*, which provides constructs that allow the reasoning about the manipulation of context, as well as the interaction of systems with the context. One of the most relevant feature is defining individual contexts; that is, context is defined from the perspective of each component and consequently different components may perceive different contexts.

Cafezeiro et al. [13] present the algebra of contextualized ontologies and an approach to specify context-aware systems. The algebra is designed to support context modelling and aims at the specification of modular and scalable description of arbitrarily complex systems. Contextualization is a basic notion and a small set of simple and powerful operations defined to compose and decompose contextualized entities. The specification approach considers the gap between the formal specification and the real application and splits the specification process in three levels varying from the system design to the complete formalization using the algebra.

Moving towards situation-awareness, it is useful to refer to the work of Dobson et al. [58] who give an overview of techniques available for situation identification. Situation identification as characterized there deals with (i) tracking the sequence of actions performed in the environment; (ii) checking them against temporal and spatial constraints; and (iii) predicting next situation.

With IEs being situated intelligent systems, also approaches from the areas of autonomous intelligent robots and agents are applicable. *Situation Calculus* [40], for instance, is a first-order logic that offers formal constructs to express and verify situations in an IE. *Situation Calculus* specifications can be implemented in *Golog* [49], a Prolog dialect that allows an engineer to build intelligent reasoning agents for the identification and verification of the current situation, as well as for the prediction of future situations and for goal-directed reasoning.

Situation Calculus and *Golog* have also been adopted to perform runtime verification of formal correctness properties of human actions and behaviours in IEs [21]. This may be applied both for the safety of cognitively impaired inhabitants in Ambient Assisted Living scenarios and for the prevention of system failures due to intentional or unintentional incorrect user actions in IEs.

From a methodological point of view, Coronato and De Pietro [22] proposed a methodology for the design of Ambient Intelligence and Location-Aware applications which is composed of (i) meta-model of an ambient intelligence application; (ii) a detailed development process and (iii) a set of suggested tools to use at any stage of the process, along with guidelines and principles for designers. The meta-model formally defines the entities in the application domain, their unambiguous meaning, and the relations between them. The development process defines four different stages: (1) informal modelling, (2) structural modelling, (3) behavioural modelling, and (4) simulation and verification. Stage-by-stage tools to accomplish the tasks and guidelines are suggested.

An extension of such a methodology has been proposed [7] to encourage development teams to adopt more rigorous and formal approaches in the engineering of IE systems, the *MIRIE* methodology. There are other methodologies available in the area, however, the authors emphasize that tools for using formal approaches need to be developed, which are easy to use. Hence the authors chose a tool which is well known, robust and free, *SPIN*, which is based in a modelling language that resembles very much the programming languages most programmers are familiar with. To facilitate the adoption of the tool and method a guide to install the tool and to create models a supplementary document, publicly available, was created to support the published article.

More recently, a remarkable interest is moving around *Requirements Engineering* [47]. In [27], authors present a framework for engineering requirements of IE such as context-awareness. The framework defines a requirements engineering process model, called *Requirements for Intelligent Environments* (R4IE). R4IE proposes a series of concerns that are of particular relevance while engineering requirements of an IE. The process has also been specialized for Ambient Assisted Living applications (*R4IE-AAL*).

A similar theme has been faced by Sutcliffe et al. [52]. They present a layered model, called *PC-RE*, relating to spatio-temporal and individual concerns. The model is not a new formal process. Instead, it consists of a road-map designed to complement existing requirements engineering methodologies.

3 Correctness analysis

The problem of understanding the ramifications of the logic embedded in a computer program is as old as the programming discipline itself. Initially programmers developed *testing* and then in the 1980s and 1990s the complementary strategy of *(formal) verification* matured and reached a stage where it was possible to apply it to industrial problems. *Validation* is a complementary area which looks at the alignment between the product developed and the product which is expected.

All these areas are as relevant as ever for the development of Intelligent Environments given the complex mix of sensing, networking, interfaces, humans and specialized software. Although Testing and Validation have been historically more prolific areas, most of the explorations so far focused on correctness of IEs is Verification. The following sections provide some selected references to developments in these areas. We start with related works in the area of testing as it was the first area for which approaches were reported and developments in verification are more recent.

3.1 Simulation, testing and validation

Testing and validation have not received as much attention yet as we will expect given that systems in this area are a heterogeneous collection of system components.

Guo and Heckel [29] address modelling and analysis of context-aware scenarios for mobile applications by using UMLlike meta-models and graph transformation techniques. Based on conceptual and concrete models of mobile systems, they simulate mobile platforms at different abstraction levels. The explicitly modelled mobility and context aspects, like locations and network connections, can be used as a dynamic test environment for the context-aware applications.

Kim et al. [34] introduce the *Context-Awareness Simulation Toolkit (CAST)*, able to simulate users and devices in a virtual home domain, designating their relation and creating virtual context information. For the simulation of a Smart Home the system offers three main sections: CAST_home, CAST_middleware and CAST_admin. The first one allows the specification of the main components in the system, including expected human participants; the second one focuses on the communication amongst components and the last one allows defining relationships between devices and humans and run the system.

Park et al. [45] present the *Context-Aware Simulation System (CASS)* that detects rule conflict and simulates rule behaviour for a smart home. The system is rule-based, rules are fed with users possible behaviours and contexts, as well as sensors and actuators values, to detect possible rule conflicts.

Wang et al. [55] presented an approach to enhance the test suites of context-aware applications. The focus of the contribution is on identifying when context changes may be relevant, and a control mechanism to guide the execution of given tests in a way which is meaningful to the contexts considered.

Bertran et al. [11] explain the *DiaSuite* tool suite for the development of sense–compute–control applications. The system is based on the DiaSim parameterized simulator which facilitates acquisition, testing and interfacing of software and hardware. DiaSpec is a specification language used to parameterize the system. Systems can be specified with DataSpec and the results of simulations visualized in DiaSim. The platform has been tried in telecommunications, software monitoring, and robotics and has potential for Intelligent Environments as well.

Yu et al. [59] proposal is based on BRS (*Biographical Reactive System*) to generate test cases for the changing environment of context-aware applications. The method uses *bigraphical labelled transition system* (B-LTS) to model the environment of the system and EFSM (Extended Finite State Machine) to model the middleware. The proposal uses Pattern Flow-based Test Case Strategies to produce tests from the sets of context information combinations in the real-world as represented in their B-LTS.

Most of the work is concentrated on simulations based on models and some cases of test generation. When the word validation is used it is usually meant: a way of reassuring the developers of their work through simulation pre-deployment. There have been few cases reported of true validation involving the end user after implementation.

3.2 (Formal) verification

Early calls to the importance of adopting formal verification as part of the development of ubiquitous computing systems and IEs include [5,9,42]. The latter two illustrate the potential of the use of formal verification techniques to increase the reliability of Intelligent Environments in relation to Smart Home systems. In [9] it is also highlighted the different benefits obtained with tools like *SPIN* and *UPPAAL*, with the first one offering a better modelling starting point and the second one being able to include time constraints.

Preuveneers and Berbers [46] applied *SPIN* to the modelling of a Smart Home and report best practices on how using SPIN relates to the focus on contexts which is valuable for us in this area.

D'Errico and Loreti [24] proposed *Klaim*, a process algebra designed to provide programmers with primitives for handling physical distribution, scoping and mobility of processes, as a useful tool to approach engineering of systems which are relying on the notion of 'context'. Properties in Klaim can be specified by means of *MoMo*, a modal logic equipped with primitives for assessing properties concerning distributions of resources within localities initially proposed to be used with Mobile Systems. Klaim has an associated model-checking algorithm to verify whether considered specifications satisfy or not the expected properties. If the specification satisfies a property then whenever the context is instantiated with components satisfying the assumptions the property satisfaction can be guaranteed.

Corno and Sanaullah [18] reported on tools which facilitated the representation of domotic systems through *DogOnt* to represent the ontology and *Dog* for the communication layer (based on OSGI). State Charts and UML were used to represent the dynamic part of the system and *DogSim* is used to generate state charts in SCXML format. Conditions can be model checked in UCTL, a UML-oriented branching time temporal logic.

Coronato and De Pietro [19] use Ambient Calculus and Ambient Logic to specify dynamic aspects of users and resources in pervasive systems. Their tool *Ambient Designer* provides support for modelling and testing of the specification. A pre-existing methodology and notation, real-time temporal logic (RTTL), is used to specify temporal constraints separately.

Benghazi et al. [10] uses a pre-existing framework *MEDISTAM-RT* for the design and analysis of real-time systems, which combines an extension of the user-friendly notation *UML-RT* with the formal language *CSP + T*, and shows how these methods and tools can be applied to the verification of non-functional requirements in Ambient Assisted Living scenarios.

Hussein et al. [33] proposed a scenario-based technique to specify properties of services graphically, an associated technique to generate variant specifications for services from its functional and adaptation scenarios, which then facilitates checking the consistency of the scenarios for service adaptation. The author's work build on previous work which allows the transformation of a scenario of a service variant into a Petri net as well as a pre-existing method to transform the service properties to CTL formulas, and the *Romeo* model checker for final conformance check.

Liu et al. [39] focuses on improving the relation between the models and the deployment environment. Their concern is to reduce the number of false positives and also on the prioritization of the cases left to consider by the developers. The authors use *AFChecker* as the model checker. The model checker derives a state transition model from a set of user-configured adaptation rules and verifies the model to detect five common types of adaptation faults: (1) non-deterministic adaptations, (2) dead rule predicates, (3) dead states (meaning that no rules can be satisfied in these states), (4) adaptation races, and (5) unreachable states (meaning that the states cannot be transitively reached from other states). The constraint inference engine of AFChecker infers both deterministic and probabilistic constraints. AFChecker relies on the previously existing *Choco* constraint solver to derive deterministic constraints by analysing the propositional atoms in the user-configured adaptation rules.

A relatively new direction of verification is *Runtime Verification*, which is defined in [37] as the discipline of computer science that deals with the study, development, and application of those verification techniques that allow checking whether a run of a system under scrutiny satisfies or violates a given correctness property. Runtime verification of running systems deals with the detection of violations (or satisfactions) of correctness properties. Thus, whenever a violation is observed, it typically does not influence or change the systems execution. However, runtime verification of a running system may be adopted to react on faults, before they turn into failures. From this point of view, runtime verification can be distinguished from other verification techniques due to the fact that the verification, at least in online monitoring, is performed while running the system. This offers the possibility of preventing/reacting to violations of correctness properties by executing recovery strategies as proposed in [20] for Ambient Intelligence applications.

4 Deployment and operability

Intelligent Environments found their ability to proactively serve users on the possibility of collecting, analysing and sometimes storing huge amounts of data. While the information is collected with the aim to improve the assistance to individuals, it may be considered an invasion of privacy, as well as the massive use of sensors and wireless technologies poses undebatable threats for the security of our environments.

To forecast how people would behave in an Intelligent Environment and perceive security and privacy threats is a key problem to establish the acceptance or rejection of a technology. Hayes et al. [31] report their evaluation of a pervasive computing system for recording everyday user experiences. Drawing on these experiences, they present a model that relates how users use physical, social, and experiential knowledge to what level of utility and privacy a new technology offers.

Several approaches may be adopted to face privacy concerns. One aims at letting the user have the possibility to control some functions of the system. Kriplean et al. [35] presented the deployment of a building-wide RFID infrastructure that can track people and objects. For this system they defined a model for physical access control, letting the user to restrict what historical information a person can see. However, even if personal information is not directly obtained by an unwanted party, much of the information can be inferred even from aggregated data. For this reason, some techniques have been defined to contrast such a possibility [2].

Other approaches require that privacy be a primary design requirement. *Privacy by Design* (PbD) [36] advocates full privacy provisions during design; that is, once privacy requirements have been determined, then the design of the sensor system itself can be completed.

A different approach aims at making users confident of the technology by involving them already in the development stage and taking into account their actual concerns and suggestions. An interesting report on the feeling of users of a monitored environment is [23]. The paper presents a participatory evaluation of an actual smart home project implemented in an independent retirement facility. The participatory evaluation allowed residents to get actively involved in the realization of the project and to express their perceptions of the sensor technologies. Finally, they did not express privacy concerns but, this result was mainly due to the adoption of a process for the acceptance of the technologies that included three phases, familiarization, adjustment and curiosity, and full integration.

Security issues related to singular classes of components of an IE are well known, e.g. from Wireless Sensor Networks [15]. The integration of such components, however, as well as the peculiarities of IEs may raise new kinds of threats. Wright et al. [57] present several critical scenarios for AAL applications ranging from the case of denial of service causing severe injuries or the death of the monitored patient, to the case of burglars able to obtain details on the lifestyle of an elder living alone. Security threats may also come from the ability of Intelligent Environments to collect large amounts of data about each individual [28].

Other relevant aspects of the operability of an IE are adaptiveness and autonomicity [25]. We report here only few examples of solutions that focus on such characteristics at different levels of an IE. Ros et al. [51] defined a method to identify abnormal human behaviour in a controlled environment. The approach is able to adapt online to environmental variations, changes in human habits, and temporal information, defined as an interval of time when the behaviour should be performed. Acampora et al. [1] presented a long-life learning strategy able to generate context-aware-based fuzzy services that maximize the users' comfort and hardware interoperability level by anticipating user's requirements. An example of adaptive architecture is provided in [54].

Deployment of systems is a demanding part of the process, and it has as much influence as the areas described in the previous sections on the overall reliability of the system and the level of satisfaction obtained from the users. Development teams had typically developed their systems largely in their labs before deploying them in a real environment. These lab-focused systems have been sometimes criticized for lacking elements which make them viable for real use by real users. For example, some of these systems may have unfriendly interfaces, or maybe the ratio of success identifying critical situations is too low (75 % may sound very effective but it means one in four situations are not correctly handled). Based on this problems, several development groups started developing the concept of what it was then called "Living Labs". [1] These living labs entail that an environment is built which resembles the real deployment environment as much as possible and humans experience the system as they will do once they acquire it as consumers. This concept can apply to any intelligent environment, but it is by far more popular in connection with Ambient Assisted Living. In that area the living lab can be either a purpose-built environment by a company or a research centre where people (e.g. a family or elderly people living independently) will be invited to live as they usually live at their homes, except this environment will have technology added which is being validated. Another option may be that the technology is deployed in the home of the users for validation. Their main feature is that it is not the scientists or the developers who are trying the technology, a normal citizen is

---

[1] http://www.openlivinglabs.eu.

experiencing whatever the development team has produced. Typically living labs will center on a number of concepts like user-centred design. Sometimes these living labs may be interconnected in a network, this can be a pilot run by a municipality or a company to assess the pros and cons of specific service innovation on a mass scale. These large-scale deployments are useful to the concept of "Smart Cities" for example.

Some of the challenges faced by teams responsible for maintenance of Intelligent Environments are: (a) *keeping the equipment powered* sensors may run out of battery if they are not serviced regularly enough, power cuts should be prevented; (b) *data transmission should be guaranteed* networks should be regularly monitored and mechanisms introduced to increase their resilience. There is an increasing reliance on wireless systems, which in some environments can prove too unreliable; and (c) *protecting the system setup* this will include making sure the equipment stays in the place it is intended for and that it is not occluded or changed by humans, pets or robots. See for example [32] for a specific experience on the challenges of deploying technology for smart homes.

User-centred design [30] is still an issue in our field and although Living Labs brought the attention on this issue and good progress has been made, there are still too many developments which are made "because we can do it" and not "because we should do it". In an attempt to highlight the importance of the user for the acceptance and long-term possibility of adoption of these technologies, some research initiatives have started to propose more fundamental ways of changing the strategy to develop Intelligent Environments, for example challenging the traditional software development processes which is focused on productivity and deliverables for other methodologies which are more specifically put the users at the centre of the innovation process [6].

5 On this inaugural issue

This inaugural issue of our journal has four selected articles which both provide a first analysis of our area on a more holistic way and at the same time help to draw an initial agenda for the community.

Our first article *Designing for User Confidence in Intelligent Environments* by Fulvio Corno, Elena Guercio, Luigi De Russis, and Eleonora Gargiulo, highlights the importance that user-centred design has for this area. One of the main contributions of this article is the 12 principles to increase the involvement of users in the development of the systems to be sold to them. This set of principles hopefully will form the backbone of a future agenda which takes user satisfaction more seriously than in the past.

Our next article *The Application of Statistical Reliability Theory in the Context of Intelligent Environments: A Tutorial Review* by Gordon Hunter, discusses the use of statistical analysis as a tool to guide development teams in their focus during the engineering process. This is an interesting new angle of vision to the problem which we hope stimulate similar creative cross-fertilization with other areas to increase the understanding of IEs as complex systems.

In *On Resilient Behaviors in Computational Systems and Environments* Vincenzo De Florio considers the notion of resiliency at an abstract level and explores ways to examine the relationship between system and environment under this light. Again, a contribution which hopefully will spark discussion and stimulate interactions between this community and others concerned about strongly related problems.

Finally, in *Building a reliable Internet of Things using Information-Centric Networking*, George C. Polyzos and Nikos Fotiou tackle an important aspect of the infrastructure sustaining any development of an Intelligent Environment: security of the information which travels through the networked system. The article explains an Information-Centric Networking system where the information itself is secured, rather than the communication channels, or the storage and processing nodes. The framework is exemplified through 'Internet of Things' scenarios; however, the authors argue one of the main advantages of the proposal is that it can be generalized and provide support for the 'Internet of Everything'.

We are hopeful these articles will encourage members in our community to interact with the authors and to propose solutions inspired by these ideas and the challenges they have highlighted.

6 The challenges and opportunities ahead

The sections above have shown that an IE engineer currently faces a significant number of interesting challenges:

- *Situatedness* An IE can be conceived of as a robotic system or a control system, it interacts with an environment by processing input from sensors and generating output to actuators which in turn change invoke changes to the environment that will be perceived. Situatedness invokes challenges well known from robotics and control systems.
- *Embedding of human users* IEs encompass human inhabitants and often perform acts of care or support, such as providing a habitable environment, more safety, or supplement basic cognitive functions for a user. All provisions with regard to safety, privacy, and sociability we would expect from a provider, care taker, or nurse

become requirements for the environment. For technical systems this means, in particular that the system needs to be enabled to react to the user in a way that would "feel natural" and does not have high cognitive load.

- *Large amounts of data* Sensors produce large amounts of data, which have to be processed and communicated in real-time, so as to reach actuators and trigger the correct reaction. This requires novel reasoning and learning techniques that draw on AI techniques, ontologies, and Big Data analytics.
- *Heterogeneity of system parts* As IEs are made of a collection of different interacting components like sensors and actuators, networks inter-connecting those, software, interfaces, and humans. Each of these are also varied, different gadgets from different vendors, different software modules possibly created by different developers, and different humans with different preferences and needs. The orchestrations of these elements is needed to allow interaction with each other, to resolve conflicts and misunderstandings, and to deliver the services expected from them.

Tackling these will open a range of opportunities for developing new markets and technologies, in particular in the following areas:

- *Urban and household infrastructure* Smart cities, with optimized transportation and living conditions belong to this group of technologies as well as the smart grid [3]. Reliability is key for these technologies to actually be deployed.
- *Remote area sensing* Large-scale geosensor networks need in-network processing capabilities. Successful disaster detection and recovery depends on the reliability of such systems over extended durations and with minimal opportunity for maintenance [26].
- *Ambient Assisted Living* This is a well established area judging for the attention and funding attracted in the last decade [8]; however, there are no standards methods and tools to develop them. Given systems in this area are given the responsibility of looking after the well-being of (often vulnerable) humans, the processes applied to develop them should be more thoroughly investigated.
- *Clothes and materials* Smart materials will become the next computing platform for IEs with novel opportunities for sensing and actuation [41]
- *Interconnected everyday objects* The internet of things (IOT) and RFID technology allow every object to become annotated data for an IE [53]. Applications range from retail to monitoring of production chains, and prevention of forgery of medications.

7 Conclusions

There are a number of convincing reasons for investing further time and effort into the reliability of Intelligent Environments.

It is unquestionable that Intelligent Environments is becoming a popular topic. It provides a natural scientific niche for developing solutions to challenges in the health sector, industry, teaching, transportation, and other areas where technology can enrich a place to assist humans. There has been considerable funding in the EU throughout the last decade under the terms of Ambient Intelligence and Ambient Assisted Living. In general, IE technologies are increasingly finding use also in critical systems in locations such as hospitals and in scenarios such as home monitoring, disaster response and crisis management, where conditions are highly dynamic.

However, there has been little emphasis on safety, standards and methodologies which can lead to a stable and successful industry based on this topic. Reliable IEs are a multi-disciplinary field with unique characteristics, requiring novel methodologies, tools, theoretical frameworks and ways to do experimentation and validation of results. As it stands now, without an outstanding research devoted to the theory and engineering of reliable IEs, the possibility of realizing high-confidence IEs even in a critical scenario is not at the level expected by the end consumer.

In summary, we strongly feel that there are not only enormous challenges to be faced in order to realize high-quality reliable IEs, but also great opportunities due to growing interests and markets and we invite the communities interested in these challenges to use JoRIE as the forum for their discussions.

References

1. Acampora G, Gaeta M, Loia V, Vasilakos AV (2010) Interoperable and adaptive fuzzy services for ambient intelligence applications. ACM Trans Auton Adapt Syst 5(2):8:1–8:26. doi:10.1145/1740600.1740604
2. Aggarwal C, Yu P (2008) A general survey of privacy-preserving data mining models and algorithms. In: Aggarwal C, Yu P (eds) Privacy-preserving data mining, advances in database systems, vol 34, pp 11–52. Springer, US. doi:10.1007/978-0-387-70992-5_2
3. Amin SM, Wollenberg BF (2005) Toward a smart grid: power delivery for the 21st century. Power Energy Mag IEEE 3(5):34–41
4. Augusto J, Callaghan V, Cook D, Kameas A, Satoh I (2013) intelligent environments: a manifesto. Hum-Cent Comput Inf Sci 3(1):12. doi:10.1186/2192-1962-3-12
5. Augusto JC (2009) Increasing reliability in the development of intelligent environments. In: Intelligent environments 2009, Proceedings of the 5th international conference on intelligent environments, Barcelona, Spain 2009, pp 134–141. doi:10.3233/978-1-60750-034-6-134

6. Augusto JC (2014) User-centric software development process. In: Callaghan SEV, Shen L (ed) Intelligent environments, IEEE Press, pp 252–255
7. Augusto JC, Hornos MJ (2013) Software simulation and verification to increase the reliability of intelligent environments. Adv Eng Softw 58:18–34. doi:10.1016/j.advengsoft.2012.12.004
8. Augusto JC, Huch M, Kameas A, Maitland J, McCullagh PJ, Roberts J, Sixsmith A, Wichert R (eds) Handbook of ambient assisted living, technology for healthcare, rehabilitation and well-being, Ambient intelligence and smart environments, vol 11, IOS Press. http://www.booksonline.iospress.nl/Content/View.aspx?piid=26497
9. Augusto JC, McCullagh P (2007) Ambient intelligence: concepts and applications. Int J Comput Sci Inf Syst 4(1):1–28. http://www.comsis.org/pdf.php?id=nst-4604
10. Benghazi K, Hurtado MV, Hornos MJ, Rodríguez ML, Rodríguez-Domínguez C, Pelegrina AB, Rodríguez-Fórtiz MJ (2012) Enabling correct design and formal analysis of ambient assisted living systems. J Syst Softw 85(3):498–510. doi:10.1016/j.jss.2011.05.022
11. Bertran B, Bruneau J, Cassou D, Loriant N, Balland E, Consel C (2014) Diasuite: a tool suite to develop sense/compute/control applications, Experimental Software and Toolkits (EST 4): a special issue of the workshop on academic software development tools and techniques (WASDeTT-3. Sci Comput Program 79(0):39–51. doi:10.1016/j.scico.2012.04.001
12. Birkedal L, Debois S, Elsborg E, Hildebrandt T, Niss H (2006) Bigraphical models of context-aware systems. In: Proceedings of the 9th European joint conference on foundations of software science and computation structures, FOSSACS'06. Springer, Berlin, pp 187–201. doi:10.1007/11690634_13
13. Cafezeiro I, Viterbo J, Rademaker A, Haeusler EH, Endler M (2014) Specifying ubiquitous systems through the algebra of contextualized ontologies. Knowl Eng Rev 29:171–185. doi:10.1017/S0269888914000046
14. Cardelli L, Gordon AD (2000) Anyti anywhere: modal logics for mobile ambients. In: Proceedings of the 27th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, POPL '00, ACM, New York, pp 365–377. doi:10.1145/325694.325742
15. Chen X, Makki K, Yen K, Pissinou N (2009) Sensor network security: a survey. Commun Surv Tutor. IEEE 11(2):52–73. doi:10.1109/SURV.2009.090205
16. Clark A (1997) Being there: putting brain, body, and world together again. MIT press, London
17. Cook DJ, Augusto JC, Jakkula VR (2009) Ambient intelligence: technologies, applications, and opportunities. Pervasive Mobile Comput 5(4):277–298. doi:10.1016/j.pmcj.2009.04.001
18. Corno F, Sanaullah M (2011) Design time methodology for the formal verification of intelligent domotic environments. In: Novais P, Preuveneers D, Corchado J (eds) Ambient intelligence, software and applications, Advances in intelligent and soft computing, vol 92. Springer, Berlin, pp 9–16. doi:10.1007/978-3-642-19937-0_2
19. Coronato A, De Pietro G (2011) Formal specification and verification of ubiquitous and pervasive systems. ACM Trans Auton Adapt Syst 6(1):9:1–9:6. doi:10.1145/1921641.1921650
20. Coronato A, De Pietro G (2011) Tools for the rapid prototyping of provably correct ambient intelligence applications. Softw Eng IEEE Trans 99:1. doi:10.1109/TSE.2011.67
21. Coronato A, De Pietro G (2013) Situation awareness in applications of ambient assisted living for cognitive impaired people. Mobile Netw Appl 18(3):444–453. doi:10.1007/s11036-012-0409-8
22. Coronato A, Pietro GD (2010) Formal design of ambient intelligence applications. Computer 43(12):60–68. doi:10.1109/MC.2010.335
23. Demiris G, Oliver DP, Dickey G, Skubic M, Rantz M (2008) Findings from a participatory evaluation of a smart home application for older adults. Technol Health Care 16(2):111–118. http://iospress.metapress.com/content/
24. D'Errico L, Loreti M (2012) Context aware specification and verification of distributed systems. In: Bruni R, Sassone V (eds) Trustworthy global computing, Lecture notes in computer science, vol 7173. Springer, Berlin, pp 142–159. doi:10.1007/978-3-642-30065-3_9
25. Dobson S, Denazis S, Fernández A, Gaïti D, Gelenbe E, Massacci F, Nixon P, Saffre F, Schmidt N, Zambonelli F (2006) A survey of autonomic communications. ACM Trans Auton Adapt Syst 1(2):223–259. doi:10.1145/1186778.1186782
26. Duckham M (2012) Decentralized spatial computing: foundations of geosensor networks. Springer Science & Business Media, New York
27. Evans C, Brodie L, Augusto JC (2014) Requirements engineering for intelligent environments. In: Proceedings of the 2014 international conference on intelligent environments, IE '14, IEEE Computer Society, Washington, DC, USA, pp 154–161. doi:10.1109/IE.2014.30
28. Friedewald M, Costa OD, Punie Y, Alahuhta P, Heinonen S (2005) Perspectives of ambient intelligence in the home environment. Telemat Inform 22(3):221–238. doi:10.1016/j.tele.2004.11.001
29. Guo P, Heckel R (2004) Modeling and simulation of context-aware mobile systems. In: Proceedings of the 19th IEEE international conference on automated software engineering, ASE '04, IEEE Computer Society, Washington, DC, USA, pp 430–433. doi:10.1109/ASE.2004.49
30. Haines V, Maguire M, Cooper C, Mitchell V, Lenton F, Keval H, Nicolle C (2005) User centred design in smart homes: research to support the equipment and services aggregation trials, technical report. Loughborough University, Loughborough
31. Hayes GR, Poole ES, Iachello G, Patel SN, Grimes A, Abowd GD, Truong KN (2007) Physical, social, and experiential knowledge in pervasive computing environments. IEEE Pervasive Comput 6(4):56–63. doi:10.1109/MPRV.2007.82
32. Hnat TW, Srinivasan V, Lu J, Sookoor TI, Dawson R, Stankovic J, Whitehouse K (2011) The Hitchhiker's guide to successful residential sensing deployments. In: Proceedings of the 9th ACM conference on embedded networked sensor systems, SenSys '11, ACM, New York, NY, USA, pp 232–245. doi:10.1145/2070942.2070966
33. Hussein M, Han J, Yu J, Colman A (2013) Scenario-based validation of requirements for context-aware adaptive services. In: Proceedings of the 2013 IEEE 20th international conference on web services, ICWS '13, IEEE Computer Society, Washington, DC, USA, pp 348–355. doi:10.1109/ICWS.2013.54
34. Kim I, Park H, Lee Y, Lee S, Lee H, Noh B (2006) Design and implementation of context-awareness simulation toolkit for context learning. In: Sensor networks, ubiquitous, and trustworthy computing, IEEE international conference, vol 2, pp 96–103. doi:10.1109/SUTC.2006.51
35. Kriplean T, Welbourne E, Khoussainova N, Rastogi V, Balazinska M, Borriello G, Kohno T, Suciu D (2007) Physical access control for captured rfid data. Pervasive Comput IEEE 6(4):48–55. doi:10.1109/MPRV.2007.81
36. Langheinrich M (2001) Privacy by design, principles of privacy-aware ubiquitous systems. In: Proceedings of the 3rd international conference on ubiquitous computing, UbiComp '01, Springer, London, pp 273–291. http://dl.acm.org/citation.cfm?id=647987.741336
37. Leucker M, Schallhart C (2009) A brief account of runtime verification, The 1st workshop on formal languages and analysis of contract-oriented software (FLACOS07). J Logic Algebr Program 78(5):293–303. doi:10.1016/j.jlap.2008.08.004

38. Lieberman H, Selker T (2000) Out of context: computer systems that adapt to, and learn from, context. IBM Syst J 39(3.4):617–632
39. Liu Y, Xu C, Cheung SC (2013) Afchecker: effective model checking for context-aware adaptive applications. J Syst Softw 86(3):854–867. doi:10.1016/j.jss.2012.11.055
40. McCarthy J (1963) Situations, actions, and causal laws, technical report memo 2, Stanford Artificial Intelligence Project, Stanford University, Stanford
41. McEvoy M, Correll N (2015) Materials that couple sensing, actuation, computation, and communication. Science 347(6228):1261–1689
42. Milner R (2006) Ubiquitous computing: shall we understand it? Comput J 49(4):383–389
43. Milner R (2008) Bigraphs and their algebra. Electr Notes Theor Comput Sci 209:5–19
44. Nakashima H, Aghajan H, Augusto JC (2009) Handbook of ambient intelligence and smart environments, 1st edn. Springer, New York
45. Park J, Moon M, Hwang S, Yeom K (2007) Cass: a context-aware simulation system for smart home. In: Software engineering research, management applications, SERA 2007, 5th ACIS international conference, pp 461–467. doi:10.1109/SERA.2007.60
46. Preuveneers D, Berbers Y (2012) Consistency in context-aware behavior: a model checking approach. In: Bota JA, Schmidtke HR, Nakashima T, Al-Mulla MR, Augusto JC, Aztiria A, Ball M, Callaghan V, Cook DJ, Dooley J, O'Donoghue J, Egerton S, Haya PA, Hornos MJ, Morales E, Orozco JC, Portillo-Rodrguez O, Gonzlez AR, Sandoval O, Tripicchio P, Wang M, Zamudio V (eds) Intelligent environments (workshops), ambient intelligence and smart environments, vol 13, IOS Press, pp 401–412
47. Preuveneers D, Novais P (2012) A survey of software engineering best practices for the development of smart applications in ambient intelligence. J Ambient Intell Smart Environ 4(3):149–162. http://dl.acm.org/citation.cfm?id=2350776.2350779
48. Ranganathan A, Campbell RH (2008) Provably correct pervasive computing environments. In: Proceedings of the 2008 sixth annual IEEE international conference on pervasive computing and communications, PERCOM '08, IEEE Computer Society, Washington, DC, USA, pp 160–169. doi:10.1109/PERCOM.2008.116
49. Reiter R (2001) Knowledge in action: logical foundations for specifying and implementing dynamical systems, illustrated, edition. The MIT Press, Massachusetts
50. Roman G, Julien C, Payton J (2004) A formal treatment of context-awareness. In: Proceedings of fundamental approaches to software engineering, 7th international conference, FASE 2004, held as part of the joint European conferences on theory and practice of software, ETAPS 2004 Barcelona, Spain, March 29–April 2, 2004, pp 12–36. doi:10.1007/978-3-540-24721-0_2
51. Ros M, Cullar M, Delgado M, Vila A (2013) Online recognition of human activities and adaptation to habit changes by means of learning automata and fuzzy temporal windows. Inf Sci 220(0):86–101 (online fuzzy machine learning and data mining). doi:10.1016/j.ins.2011.10.005
52. Sutcliffe A, Fickas S, Sohlberg M (2006) Pc-re: a method for personal and contextual requirements engineering with some experience. Requir Eng 11(3):157–173. doi:10.1007/s00766-006-0030-0
53. Uckelmann D, Harrison M, Michahelles F (2011) Architecting the internet of things. Springer Science & Business Media, New York
54. Viroli M, Zambonelli F (2010) A biochemical approach to adaptive service ecosystems. Inf Sci 180(10):1876–1892 (special issue on intelligent distributed information systems). doi:10.1016/j.ins.2009.11.021
55. Wang Z, Elbaum S, Rosenblum D (2007) Automated generation of context-aware tests. In: Software engineering, ICSE 2007, 29th international conference, pp 406–415. doi:10.1109/ICSE.2007.18
56. Weiser M (1991) The computer for the 21st century. Sci Am 265(3):94–104
57. Wright D, Gutwirth S, Friedewald M, Vildjiounaite E, Punie Y (2008) Safeguards in a world of ambient intelligence (The international library of ethics, law and technology), 1 edn
58. Ye J, Dobson S, McKeever S (2012) Review: situation identification techniques in pervasive computing: a review. Pervasive Mob Comput 8(1):36–66. doi:10.1016/j.pmcj.2011.01.004
59. Yu L, Tsai, WT, Jiang Y, Gao J (2014) Generating test cases for context-aware applications using bigraphs. In: Software security and reliability (SERE), 2014 eighth international conference, pp 137–146. doi:10.1109/SERE.2014.27