Pervasive and Mobile Computing 8 (2012) 614–628

Contents lists available at SciVerse ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

DoMAIns: Domain-based modeling for Ambient Intelligence

Dario Bonino *, Fulvio Corno

Politecnico di Torino, Dipartimento di Automatica ed Informatica, Corso Duca degli Abruzzi 24, 10129 - Torino, Italy

ARTICLE INFO

Article history:
Available online 4 November 2011

Keywords:
Home Automation
Intelligent Domotic Environment
Ambient Intelligence
State diagrams
Domotic plant
Modeling

ABSTRACT

Ambient Intelligence and Smart Home Automation systems are currently emerging as feasible and ready to exploit solutions to support more intelligent features inside future and current homes. Thanks to increased availability of off-the-shelf components and to relatively easy to implement solutions we are experiencing a steady evolution of households, causing an ever-increasing users' awareness of the capabilities of such innovative environments. To foster effective adoption of Smart Home Automation technologies in our home environments, traditional architectural and plant design must be complemented by sound design methodologies and tools, supporting the whole environment design cycle, including for example modeling, simulation and emulation, as well as, when feasible, formal model-checking and verification. Several research efforts have already addressed the design of expressive modeling tools, mostly based on Semantic Web technologies, as well as of suitable platforms for adding interoperation and rule-based intelligence to home environments. This paper proposes a new modeling methodology designed to fit the different phases of Intelligent Environments design, with a particular focus on validation and verification of the whole system. Carefully designed separation of modeled entities permits to exploit the DoMAIns framework during all phases of the environment design, from early abstract conception to the final in-field deployment. The DoMAIns design methodology is applied to a sample use case that involves comprehensive modeling and simulation of a Bank Security Booth, including the environment, the control algorithms, the automation devices and the user. Results show that the approach is feasible and that can easily handle different types of environment modeling, required in the different design phases, and for each of them it may support simulation, emulation, or other verification techniques.

1. Introduction

Ambient Intelligence (AmI) and Smart Home Automation (SHA) systems are currently gaining momentum by being recognized as affordable and easy to extend solutions for bringing intelligence to new and existing households. Although there exists a clear evidence of incompatibilities between domotic systems available on the market, researchers are converging on a common vision of SHA where traditional Home Automation (HA) plants built with off-the-shelf components are integrated by a single "computationally strong" element (device) supporting inter-plant interaction and bringing intelligence to the home [1]. Initial efforts [2–5] involving these new Intelligent Domotic Environments (IDE) are nowadays maturing in a more structured research field including contributions from Ambient Intelligence [6], Pervasive Computing [7,8] and Smart Environments [9–11]. Moving from first, sparse approaches, the research community is now tackling the design of next generation buildings and homes by applying well known, sound methodologies developed in the context

* Corresponding author. Tel.: +39 011 564 7170; fax: +39 011 564 7099.
E-mail addresses: dario.bonino@polito.it (D. Bonino), fulvio.corno@polito.it (F. Corno).

1574-1192/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.pmcj.2011.10.009

of Software Engineering [12–14]. At the basis of this new wave of research lies the need to organically design models for complex Intelligent Environments [15,16] and for the associated Context information [17,18]. These models must permit, on one side, to address interoperability issues by exploiting a shared environment abstraction that enables the development of technology-independent home intelligence. On the other hand, thanks to formal representation of environments, devices, and behaviors, design models can be validated, verified and simulated prior to deployment, checking the compliance of envisioned solutions to security, functionality, reliability and other requirements typical of such complex systems.

While context modeling originally attracted more attention, nowadays increased focus on home and device modeling has lead to several interesting approaches, mainly based on ontologies [19,15] and web services. Only in the last years, modeling efforts started to focus on formal and dynamic models with the goal of supporting simulation and formal verification [20–23]. However, additional contributions are still needed since most of available approaches limit the application of modeling and validation techniques to the early design stages and/or to specific, homogeneous subsystems. Conversely, the possibility of applying these tools throughout the entire design and development chain is seldom addressed, preventing the evolution of more comprehensive and integrated solutions, that could benefit form early verification as well as from development-time emulation, with some simulated devices and some real devices.

To overcome this issue and to support a more integrated and engineered design of Intelligent Domotic Environments, we propose a general and modular modeling approach that seamlessly supports different design phases and their corresponding configurations, by adopting modeling techniques that enable validation and verification at different stages. This design framework, called DoMAIns (recursive acronym for ''Domain-based Modeling for Ambient Intelligence''), supports the whole IDE design chain, from early specifications to final on-the-field deployment. To support this ambitious goal, we strictly apply well known partitioning and modularity principles, striving to achieve a clear separation between different modeling aspects and concerns. Such a differentiation is crucial to support verification, simulation and emulation in the same model. By keeping modeling concerns (i.e., representation domains) separated, we expect the DoMAIns framework to successfully tackle complex, Intelligent Environments, supporting, in each case, clear identification of solutions under test and concurrent design and development of multiple, alternative ideas.

This paper contributes a first high-level view over DoMAIns modeling, basing on 5 pre-defined, yet extendable, *Representation Domains* and shows a concrete application example with the aim of offering a better understanding of the proposed methodology as well as to foster further investigations. Representation Domains are different, and possibly overlapping, sub-models of an Intelligent Domotic Environment connected through explicit ''boundary'' definitions or *interfaces*. Within each Domain, a given model is easily replaceable by alternative ones (based on the same or on different formalisms, e.g. a statechart model can be replaced by a finite elements model, etc.), or by its ''real'' counterpart, i.e., by actual software modules, hardware devices or users.

Whenever real and simulated items co-exist in the same DoMAIns model, we refer to it as to an *Emulation Model*, i.e., a model in which elements under test (verification/validation) are connected to real items in an hardware-in-the-loop chain. Emulation models clearly require suitable adapters to convert information across the ''virtual'' and ''real'' domain elements.

Separating dynamic IDE models into Domains brings several advantages:

1. It permits to detail domain models with *different granularities*, depending on the representation goals, e.g., pure simulation, preliminary design verification, fine-tuning of control algorithms, etc.
2. It supports different *representation techniques*, enabling designers to tackle every domain with the most suited solutions, e.g., finite elements simulations for heat transfer, state machines for heating actuators, robust control theory for fine temperature regulation.
3. It provides clean *interface points* for cutting ''virtual'' IDE representations and for integrating real devices in the end-to-end simulation chain (emulation).

Clearly, many efforts are still needed to fully realize the vision underlying the DoMAIns approach, however this paper tries to lay down the general structure of Representation Domains on top of which more effective and comprehensive IDE modeling can be devised.

The remainder of the paper is organized as follows: Section 2 reports an overview of relevant related and complementary works. Section 3 introduces the representation domains defined in DoMAIns, describing their peculiar features and the interfaces between them. Section 4 shows use cases and applications in which DoMAIns can be successfully exploited, highlighting the flexibility of the proposed modeling methodology. Section 5 provides a modeling example where the DoMAIns framework is applied to the representation and simulation of a Bank Security Booth control, whereas Section 6 provides the relative implementation details. Finally Section 7 concludes the paper and proposes future works.

2. Related work

Research in modeling, simulation and verification of Intelligent Environments is rather new and still consists of sparse and mainly un-coordinated efforts. Currently, few works try to tackle the whole IDE design life cycle, from early design stages to final in-field deployment. On the converse, several interesting approaches can be found dealing with specific issues relevant to single design/development stages.

Habitation [14] is currently, to our knowledge, the most relevant modeling effort tackling the whole IDE design process. It is a Domain Specific Language (DSL) explicitly designed for Home Automation and one of the earliest attempts to apply