The Autonomic Computing Paradigm in Adaptive Building / Ambient Intelligence Systems

Aliaksei Andrushevich, Stephan Tomek, and Alexander Klapproth

CEESAR-iHomeLab, Lucerne University of Applied Sciences and Arts, Technikumstr. 21,
6048 Horw, Switzerland
{aliaksei.andrushevich,stephan.tomek,
alexander.klapproth}@hslu.ch

Abstract. This work is devoted to the classification and adaptation of current ambient intelligence (AmI) research activities from the viewpoint of the autonomic computing paradigm. Special attention is given to the implementation of AmI's user-centric focus in autonomic computing.

Keywords: Ambient Intelligence, Autonomic Computing, self-adaptive system, user-centric requirements, human-centered design.

1 Introduction and Motivation

AmI systems are typically based on the adaptation of artificial intelligence (AI) techniques on low-power, low-cost, heterogeneous and physically distributed pervasive / ubiquitous computing infrastructure. The main aim of AmI systems is however in supporting the end-user's daily activities in an unobtrusive and easy way through user-centric architecture and human-centered design.

Building Intelligence (BI) is a type of AmI implemented on underlying technical architecture and infrastructure of buildings, homes and other construction objects. Considering constantly changing user-needs different building automation applications may have partially contradictory control strategies for commonly shared resources. Moreover, many building automation applications pursuing energy efficiency, comfort, user safety and security or ambient assistance are only widely accepted as a part of an AmI system if the configuration and maintenance efforts will be kept to a minimum. Considering constantly changing user requirements during the whole lifecycle of AmI applications, self-adaptive system properties quickly become inevitable for AmI adoption.

In order to make the system self-adaptive IBM introduced in 2001 the paradigm of autonomic computing (AC) [1] that systemizes and formalizes the necessary properties and functional components of *self-managing* system architecture. An extensive amount of focused research [2] has resulted in usage of AC paradigm in different applications areas including power management in wireless sensor networks (WSN), dynamic resource management and administration in GRID computing systems, and

D. Keyson et al. (Eds.): AmI 2011, LNCS 7040, pp. 98–104, 2011.

pervasive/ubiquitous computing vision aiming at building intelligent environments by usage of heterogeneous sensing-computing-actuating devices [3].

2 Autonomic Computing

The high-level core self-management properties of the AC paradigm include self-configuration, self-optimization, self-healing and self-protection. The intuitive sense of the autonomic system is in reflexing of the current environmental context or in reflecting dynamism in the system [2].

Generally, the implementation of AC paradigm is defined in the IBM's reference model for autonomic control loops and called MAPE-K loop - Monitoring, Analysis, Planning, Execution, and gathered Knowledge.

Five AC Adoption Model Levels were also introduced by IBM in 2003 to be able to measure the systems on their way to autonomicity. These levels include 1 - Basic, 2 - Managed, 3 - Predictive, 4 - Adaptive and 5 - Autonomic. A recent AC-focused survey [2] on self-managing systems (those of AC Levels 4 and 5) suggests to classify the ongoing AC research through the following four key autonomicity elements:

- Support – when improving the complete system performance by focusing on one aspect or component;
- Core – when end-to-end self-management solution drives the core application without heading higher-level human based goals;
- Autonomous – when full end-to-end self-management typically agent-based solution self-adapts to the environment but not measuring own performance;
- Autonomic – when full architecture is reflecting its own performance and adapting itself considering higher-level human based goals.

3 AmI Research in Components of AC Architecture

Figure 1 shows our understanding about the way different works of current AmI research can contribute to the implementation of intelligent self-managing and adaptive to the user building management system (BMS).

The state-of-the-art in context-aware ubiquitous computing middleware [4, 5] looks promising in terms of performing the data acquisition from different sensors and implementing data fusion mechanisms in the *monitoring* functional block of the AC control loop architecture.

The introduction of context, the usage of context-modeling as well as context deriving procedures based on consistent previously prepared sensor data can be used for implementations of the *analyzing* component of the MAPE-K loop.

Following the AC paradigm and mainly focusing on monitoring and analyzing, the AutoHome project [6] is a successful example on creation of the context-aware BMS.

The algorithmic know-how, gathered by AmI researchers offers a number of AI approaches meant for generation of correction action schedules or *plans* for adaptive systems. Useful for AmI techniques mainly address pattern recognition, unsupervised machine learning and scheduling. Some of these AI methods are included in [7]: