

Chapter 2

The Design Principles and Practices of Interoperable Smart Spaces

Eila Ovaska

VTT Technical Research Centre of Finland, Finland

Tullio Salmon Cinotti

Università di Bologna, Italy

Alessandra Toninelli¹

INRIA, France

ABSTRACT

Smart spaces provide information about physical environments, shared with inherently dynamic applications. This chapter introduces a novel development approach with its focus on two key properties of smart space applications: the ability to interoperate and behave in a situation-sensitive manner. Sixteen principles are defined in order to guide the development of an interoperability platform for smart spaces and on how to create applications on top of it. The interoperability platform deals with information and is agnostic with respect to ontologies, programming languages, service frameworks, and communication technologies. The interoperability platform also supports extensibility, evolvability and context based adaptation, which allows new applications to be added and to behave in a situation based manner. Agile application development is based on scenario specifications, implemented by the means of the ontology and model driven development. The approach has been applied to the development of smart personal spaces, smart indoor spaces, and smart city applications.

INTRODUCTION

Everyday life can be enriched by services that exploit pervasive computing environments, which are embedded into our surroundings. These computing environments rely heavily on sensors and

sensor networks that produce a large amount of data to be analyzed and reacted to by users or/and devices (Hadim & Mohamed 2006). Since the number of these devices is increasing, the amount of information to be processed by people and devices is also rising. In order to handle the

DOI: 10.4018/978-1-60960-735-7.ch002

information overload, the environments are to be made to be smart.

The motivations for smart environments are quite well understood: to increase the visibility of opportunities, to support context understanding and ultimately to provide the correct information when and where it is required, even if not explicitly requested, with its content and format optimally adapted to the user situation and profile (Weiser 1993). Although smart spaces have been an interest of researchers and industrial professionals for years, there is hardly any easy working smart environment in practice. The main obstacle is the lack of the interoperability of devices and systems that provide the execution environment for ambient applications. Although the interoperability could be achieved at the device level by handling communication, connectivity and data with a set of standard protocols, selected for use in a situation based manner by using, e.g., the reflection pattern (Buschmann et al. 1996), there are still obstacles to handling the interoperability with higher system architecture levels, the service and information levels. Service interoperability concentrates on unambiguously describing service semantics, so that services can be searched and they can interact with each other (Kantarovitch & Niemelä 2008). Enhanced functionality is achieved through service discovery, matchmaking and the merged functionality of a selected set of services. Moreover, services enriched with semantic information on context and resources could provide the service that is the most suitable for the user's preferences and the situation at hand (Soylu et al. 2009). These enrichments however have costs; they make service centric systems more complex and error prone, and increase the need of computing resources.

Recently, two promising approaches for context-awareness have been proposed; a spatial application programming model (Meier, et al 2008) and an approach that is based on Model and Ontology Driven Development (Soylu et al. 2009). The spatial application programming model

uses a small set of predefined types for composing information and context. The approach has some similarity with ours, but it relies on a specific programming model, not a model driven development approach that embodies a generic ontology used for information sharing. Soyly et al. (2009) link the model driven development with ontology engineering and aim to assist context-awareness in all phases of the application development life cycle, i.e. at design time and run-time. Their special focus is on the context-awareness of the digital world, where the use of abstract models is easier and efficient. However, the approach introduced is at a conceptual level and still requires long and short term research. Our goal, on the contrary, is to adopt the semantic web technologies to physical spaces and to make the existing entities of our environments both smart and adaptive.

Our approach facilitates interoperability at the information level and lets devices and systems use existing solutions for describing, managing and executing services to facilitate the functionality which is required from a smart space (Sofia 2010). Thus, ambience is based on information which is provided by sensors embedded into environments and existing services, running on heterogeneous devices and systems, for free use in smart space applications, which also share the semantics of information. Thus, smart space interoperability is based on information sharing and the adaptation to existing (legacy) devices and systems. We exploit ontology orientation to represent the semantics of information. This information is shared and mapped onto a graph that uses a triple (two nodes – subject and object - connected by a predicate) as an information elementary element. Native information from the legacy devices and services is captured by agents that translate it into the specified information format of the smart spaces. Thus, a legacy device may be controlled by the interoperability platform through an enhanced legacy application, which is able to access and subscribe to the smart space.

28 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/design-principles-practices-interoperable-smart/55434

Related Content

Analog Learning Neural Network using Two-Stage Mode by Multiple and Sample Hold Circuits

Masashi Kawaguchi, Naohiro Ishii and Takashi Jimbo (2014). *International Journal of Software Innovation* (pp. 61-72).

www.irma-international.org/article/analog-learning-neural-network-using-two-stage-mode-by-multiple-and-sample-hold-circuits/111450

Development of a Master of Software Assurance Reference Curriculum

Nancy R. Mead, Julia H. Allen, Mark Ardis, Thomas B. Hilburn, Andrew J. Kornecki, Rick Linger and James McDonald (2010). *International Journal of Secure Software Engineering* (pp. 18-34).

www.irma-international.org/article/development-master-software-assurance-reference/48215

Project Contexts and the Possibilities for Mixing Software Development and Systems Approaches

D. Petkov, S. Alter, J. Wing, A. Singh, O. Petkova, T. Andrew and K. Sewchurran (2014). *Software Design and Development: Concepts, Methodologies, Tools, and Applications* (pp. 335-350).

www.irma-international.org/chapter/project-contexts-possibilities-mixing-software/77713

Evolution of Blockchain Technology: Principles, Research Trends and Challenges, Applications, and Future Directions

Oluwaleke Umar Yusuf and Maki K. Habib (2023). *Perspectives and Considerations on the Evolution of Smart Systems* (pp. 67-104).

www.irma-international.org/chapter/evolution-of-blockchain-technology/327527

Modeling and Re-Evaluating Security in an Incremental Development of RBAC-Based Systems Using B Method

Nasser Al-Mur Al-Hadhami (2019). *Exploring Security in Software Architecture and Design* (pp. 104-135).

www.irma-international.org/chapter/modeling-and-re-evaluating-security-in-an-incremental-development-of-rbac-based-systems-using-b-method/221714