INFORMATION SYSTEMS, WEB AND PERVASIVE COMPUTING SERIES

Enablers for Smart Cities

Edited by Amal El Fallah Seghrouchni
Fuyuki Ishikawa, Laurent Hérault
and Hideyuki Tokuda

iSTE

WILEY

Enablers for Smart Cities

Series Editor
Jean-Charles Pomerol

Enablers for Smart Cities

Edited by

Amal El Fallah Seghrouchni
Fuyuki Ishikawa
Laurent Hérault
Hideyuki Tokuda

iSTE

WILEY

First published 2016 in Great Britain and the United States by ISTE Ltd and John Wiley & Sons, Inc.

ISTE Ltd
27-37 St George's Road
London SW19 4EU
UK

www.iste.co.uk

John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030
USA

www.wiley.com

Library of Congress Control Number: 2016939641

British Library Cataloguing-in-Publication Data
A CIP record for this book is available from the British Library
ISBN 978-1-84821-958-8

Contents

Preface

Introduction

The concept of *smart cities* emerged few years ago as a new vision for urban development that aims to integrate multiple information and communication technology (ICT) solutions in a secure fashion to manage a city's assets. Modern ICT infrastructure and e-services should fuel sustainable growth and quality of life, enabled by a wise and participative management of natural resources to be ensured by citizens and government. The need to build *smart cities* became a requirement that relies on urban development that should take charge of the new infrastructures for smart cities (broadband infrastructures, wireless sensor networks, Internet-based networked applications, open data and open platforms) and provide various smart services and enablers in various domains including healthcare, energy, education, environmental management, transportation, mobility and public safety.

The smart enablers raise new research challenges that emerge across areas such as urban development and spatial planning, network infrastructure, technology platforms, services and applications, user behavior, cognitive modeling, service engineering, innovation theory and urban economics.

This book aims to present the most significant emerging, or already mature, research results in the domains mentioned below. This may help academics and practitioners to explore new directions and generate knowledge and solutions toward smarter cities.

This book has been written by experts and outstanding researchers in the main domains involved in smart cities' development.

Preface written by Amal EL FALLAH SEGHROUCHNI, Fuyuki ISHIKAWA and Kenji TEI.

The co-authors cover complementary domains of computer science needed such as cloud and distributed computing, artificial intelligence and sensors and belong to various and prestigious institutions in France, Japan, Italy and Spain, which bring rich experience and a broad overview of the topic.

This book consists of 11 chapters, covering a wide range of topics for smart cities. The chapters are ordered to go from technical foundations to enable certain advanced capabilities, to platform/middleware for supporting construction and execution of various applications to field experiences in cities. Each of these three parts and the involved chapters are self-contained and address issues in smart cities from various points of view. Thus, the reader can take any of the chapters independently according to his/her interest.

Chapters 1–4 present technical foundations to enable certain functions in smart cities. Specifically, the two capabilities of sensing and reasoning are focused on as the keys for smart cities. Sensing various kinds of context information is the starting point for smart support of human and societal activities. Then reasoning mechanisms make proper decisions about execution or deployment plans. Both of them need to consider the existence of heterogeneity regarding devices, information sources and applications. Chapters 1–4 address difficulties with this point as follows.

– Chapter 1 by Tei K. addresses how to manage multimodal sensor devices in wireless sensor networks collectively to support multiple applications with runtime task allocation and self-adaptation;

– Chapter 2 by Nakazawa J. addresses how various information sources on the Web, currently under limited use with Web browsers, can be transformed to sensor application programming interfaces (APIs) or data streams ready for easy integration into various application programs;

– Chapter 3 by Chaouche A. *et al.* addresses how smart behavior can be realized through spatiotemporal reasoning, not only by contextual planning but also by learning from past experiences;

– Chapter 4 by Piette F. *et al.* addresses how to dynamically use distributed smart applications over devices with heterogeneous resources with a special respect about privacy concerns.

Chapters 5–9 present platforms/middleware to support construction and execution of various applications. On one hand, this concerns how to support design, programing and verification of smart city systems. On the other hand, this concerns the architectural design in order to be holistic or to support specific features for smart city systems. Chapters 5–9 provide a wide range of discussions over these points.

– Chapter 5 by Tei K. addresses how to define the holistic architecture that supports a variety of functions necessary in smart city systems, such as virtualization, resource management and service composition;

– Chapter 6 by Gürgen L. *et al.* addresses how to support construction of event-driven behavior of smart city systems, which also requires proper configuration with devices;

– Chapter 7 by Ishikawa F. *et al.* addresses how to support verification of event-driven behavior of smart city systems, which can easily lead to conflicts between multiple devices, users and applications;

– Chapter 8 by Galache J. *et al.* discusses how to realize integration of heterogeneous devices for various smart city applications, with their experience in the Santander city, Spain;

– Chapter 9 by Olaru A. *et al.* discusses how to design the middleware tailored for smartness or intelligence provided on user devices at the right time.

Chapters 10–11 present field trials in cities. It focuses on making use of technology to realize values in cities and citizens' lives.

– Chapter 10 by Sotero Muñiz S. *et al.* reports their experiences in Santander, Spain, including city management by means of participatory sensing and traffic mobility management;

– Chapter 11 by Yonezawa T. reports the author's experiences in Fujisawa, Japan, including air-pollution monitoring using garbage trucks, delivery of live city information and tourist guidance by "smile coupon".

Although each chapter can be read independently, it is worth being aware of the connections between chapters. There is a specific emphasis on two aspects: practical integration and intelligence, both over heterogeneous devices, services and applications. Regarding the first aspect, some of the chapters come from the ClouT project, which aims at providing the infrastructure/platform of Cloud and Internet of Things for smart cities, as well as field trials. Chapters 1, 2, 5, 6, 7, 8, 10 and 11 are from the project (the whole picture of the project is found in Chapter 5). These chapters focus on how to integrate various devices or services to compose smart-city applications or experiences in cities on the basis of the integration. Regarding the second aspect, Chapters 3, 4 and 9 complement the other chapters by providing discussions on advanced techniques for smartness by multi-agent systems. This aspect is essential for emerging smart-city systems that embed interesting "human-like" characteristics of autonomy, sociality, adaptability and so on.

Biographies of the editors

Amal El Fallah Seghrouchni

Amal El Fallah Seghrouchni is a Full Professor at the University of Pierre and Marie Curie (Paris 6 – France) where she heads the MAS team at LIP6 (Laboratory of

Informatics of Paris 6). Her research is focused on autonomous agents and MAS. She is interested in the design, analysis and verification of cognitive MAS.

Her research topics find a large number of applications in the fields of design of complex systems (e.g. coordination of UAVs, multi-agent planning of aircraft missions etc.), ambient intelligence (smart-room, smart campus projects) and smart cities. These topics and applications are supported by a large number of academic collaborations and industrial projects. She has been involved in the major conferences related to autonomous agents and multi-agent systems and has published more than 150 referred papers and co-edited 16 books or post-proceedings.

Fuyuki Ishikawa

Fuyuki Ishikawa is an Associate Professor at National Institute of Informatics, Japan. He has worked on smart and trustworthy software engineering with emphasis on service-oriented computing and formal methods. His research experience includes leadership in several funded projects and publication of more than 100 referred papers about service composition, formal refinement, test generation, legal compliance and industry education.

Laurent Hérault

Dr. Laurent Hérault received his BS degree in electrical engineering and his MS degree in control engineering from the Grenoble Institute of Technology in 1987, followed by a PhD in computer science in 1991. He won the Best Junior Researcher Award from the University of Grenoble, France, in 1990. He is a CEA fellow since 2014, and led the Wireless Communications and Security labs from 2009 to 2011. Since 2011 he is VP, Director of the Europe division at CEA-LETI.

Hideyuki Tokuda

Hideyuki Tokuda obtained his PhD in computer science in 1983 from the University of Waterloo, Canada. He is a Professor of the Graduate School of Media and Governance and a Professor at the Faculty of Environment and Information Studies, Keio University, Japan. His current field of research is Information Appliance and Smart Spaces for the Ubiquitous Computing Environment. His research lab is working on applications, middleware, networks, and hardware to realize “Smart Spaces”.

Acknowledgments

The idea of this book emerged from a collaboration of more than 10 years between the Honiden-Lab at the National Institute of Informatics (NII), Japan, and the SMA team (Systèmes Multi-Agents) at LIP6 at Université Pierre et Marie Curie (Paris 6), France. The editors would like to thank all the participants of the collaboration, especially those who contributed as the authors and the reviewers of the chapters.

We thank the consortium of the ClouT (Cloud of Things for empowering the citizen clout in smart cities) project that provided interesting insights and experiences in this book. The ClouT project is supported by the seventh framework programme for research (FP7) in the European Union (EU) and National Institute of Information and Communications Technology (NICT) in Japan.

Last but not least, we are very thankful for ISTE Ltd. and Professor Jean-Charles Pomerol who gave us the opportunity and support for publication.

Introduction

Chapter 1: *Shared wireless sensor networks as an enabler for a context management system in smart cities*. This chapter by Tei K. proposes middleware for Wireless Sensor Networks (WSNs), which provides the essential infrastructure for smart cities to have sensing capabilities, flexibly used anywhere. The focus of the middleware is shared by the use of WSNs: each sensor has several capabilities of sensing, and the sensors can be collectively used to allow for multiple applications to run. The middleware addresses difficulties because of the distributed and dynamic nature by means of a multi-level task description language as well as mechanisms for runtime task management and self-adaptation.

Chapter 2: *Sensorizer: an architecture for regenerating cyber physical data streams from the Web*. This chapter by Nakazawa J. proposes architectures for enhancing the sensing capability of smart city systems, by excavating information sources on the Web. Currently, there are various kinds of information on the Web only consumed by Web browsing of human users. The proposed approach enables what is called the "Sensorizer", which means making the information on the Web into sensor data streams easily accessible by application programs. The chapter shows how it is easy to sensorize existing Web information by a browser extension and also reports their experience with parking lot occupancy.

Chapter 3: *Smart agent foundations: from planning to spatio-temporal guidance*. The chapter by Chaouche A. *et al*. proposes an approach based on intelligent agents to build applications for smart cities. It highlights how software agents can be designed to assist users in their tasks and objectives in the context of the smart-city. These agents can be viewed as an interface between the user/citizen and the computational world of the smart city. The proposed approach is original and relevant to smart application design, as the agents can react on the fly, to the changes

Introduction written by Amal EL FALLAH SEGHROUCHNI, Fuyuki ISHIKAWA and Kenji TEI.

of contexts, sometimes unexpected, of their environment. A scenario of an ambient intelligent system dedicated to a smart campus is presented as well as the whole approach including BDI model of the agent, the planning process and the learning mechanisms that help to improve the agent performances and hence the quality of the smart applications.

Chapter 4: *A multi-agent middleware for deployment of ambient applications*. This chapter by Piette F. *et al*. proposes a middleware to ease the development, deployment, configuration and monitoring of applications for ambient systems. The main advantage of this middleware is that it decorrelates applications from hardware infrastructures by separately describing with different levels of the various entities of the system. The specifications and properties of the available hardware entities of the infrastructure, and the requirements of ambient applications are modeled with graphs. These descriptions allow us to reason about the deployment of applications on a heterogeneous hardware infrastructure by using a graph matching algorithm that finds a graph homomorphism between the application graph and the hardware infrastructure graph. The approach is based on multi-agent systems paradigm (MAS). The agent organization ensures data and resources privacy. Infrastructure agents are geographically located and manage a part of the hardware infrastructure. Application agents manage the applications used on it and the data generated by these applications.

Chapter 5: *ClouT: cloud of things for empowering citizen's clout in smart cities*. This chapter by Tei K. overviews the ClouT project, which investigates the power of combining Cloud and IoT (Internet of Things) for smart cities, through Europe–Japan collaboration. The project involves a conceptual definition to handle various kinds of things in a unified way following the Cloud model. The project also proposes a holistic architecture that involves various functionalities for the infrastructure and platform for smart cities. Notable techniques and field trials with cities are found in other chapters of this book (Chapters 1, 2, 6, 7, 8, 10 and 11).

Chapter 6: *sensiNact, IoT Platform as a Service*. This chapter by Gürgen L. *et al*. proposes the sensiNact framework to support the construction of IoT applications. The framework includes a platform to enable easy access to various kinds of devices. The key here is to handle the heterogeneity: various protocols with the application side and with the device side. The framework also includes a studio or integrated development environment (ID) to support description of event-driven behavior to sense and act using the devices. Thus, the chapter gives a good overview of how to support application construction with various IoT devices.

Chapter 7: *Verification and configuration of smart space applications*. This chapter by Ishikawa F. *et al*. proposes a framework for verifying and configuring smart-space applications. As depicted in Chapter 6, the behavior of smart applications is often event driven, often specified in terms of event–condition–action rules. These rules can easily cause conflict between one another but they are difficult to detect as they are hidden in a great many possible scenarios. The proposed framework facilitates using the model checking technique, which exhaustively checks possible transitions, by providing translation between engineer-familiar descriptions and mathematical descriptions for model checking.

Chapter 8: *SmartSantander, a massive self-managed, scalable and interconnected IoT deployment*. This chapter by Galache J. *et al*. reports their experience in Santander, Spain, regarding the architecture that supports deployment of various applications and services using IoT devices. The experience, supported by two projects of SmartSantander and ClouT, includes a unique massive deployment of IoT devices and field applications involving transportation monitoring, participatory sensing, augmented reality and so on.

Chapter 9: *Using context-aware multi-agent systems for robust smart city infrastructure*. This chapter by Olaru A. *et al*. focuses on how to achieve the goal of providing smart city users with fresh, relevant information, promptly, without the users needing to offer personal information in exchange, except when the information is absolutely necessarily and the user is fully aware of the transfer. The chapter proposes MAS-based architecture for context-aware AmI applications in which context is a first-class entity in the design of the system. The MAS uses a fully distributed context management architecture that requires no centralized components and relies as much as possible on the computational resources that belong to the user (rather than to other organizations). The DESIA architecture integrates the user's social context and the tATAmI framework/S-CLAIM language, which are the foundation for a middleware for the management of the user's data in a context-aware manner.

Chapter 10: *Santander City*. This chapter by Sotero S. *et al*. reports their experience in Santander, Spain, the same city as Chapter 10, but from the city government's side. This chapter, therefore, gives interesting insights about requirements and applications that should be supported by enabler techniques such as those described in the previous chapters. Specifically, the chapter reports smart-city applications for city management and traffic mobility management by using not only physical sensors, but also participatory sensing.

Chapter 11: *Fujisawa, towards a sustainable smart city*. This chapter by Yonezawa T. reports their experience in the Fujisawa city, Japan. The city investigates smart city applications, but in a sustainable way, that is, not accepting "introducing a lot of devices!" Their interesting idea is to attach sensors to garbage trucks to realize the sensing capability that covers the whole city. This chapter also reports other ideas to encourage, involve and interact with people by means of dashboards in stations and a mechanism with a "smile" game.

1

Shared Wireless Sensor Networks as Enablers for a Context Management System in Smart Cities

Wireless sensor networks (WSNs) are commonly used as a sensing infrastructure for smart city applications. A WSN is easy to use and can cover a wide area at low costs because of its wireless communication capability. The sensor nodes constituting a WSN are usually equipped with one or more sensor devices and can be used for different measurement purposes by reprogramming them. If WSNs could be shared by different smart city applications, they could be even more valuable enablers for smart cities. However, it is not easy to share WSNs. A shared WSN needs to support different kinds of measurement tasks at the same time and be able to accept new tasks at runtime. Even in a traditional closed WSN, its software should be carefully developed to satisfy certain quality requirements despite the severe resource constraints affecting the individual programmable sensor nodes (the sensor nodes of WSNs usually have quite limited resources, e.g. small batteries, low-spec CPU and narrow bandwidth). This issue is much harder to resolve in the case of a shared WSN. To satisfy the quality requirements of different applications, a WSN should be configured carefully according to specifications of the tasks, their quality requirements, and the environment, and should adapt its configuration in response to changes in the environment and the applications. A shared WSN should support various measurements, manage tasks at runtime and adapt to changes in the environment to reduce unnecessary consumption of resources. To develop such a shared WSN, we propose a middleware support for the network. In this chapter, we describe the architecture of our XAC middleware and the issues relevant to the shared WSN from the viewpoints of the task-description language, runtime task management and self-adaptation.

1.1. Introduction

In the smart cities of the future, many context-aware applications will support the citizen's activities by proactively controlling the various devices used therein.

Chapter written by Kenji TEI.

Enablers for Smart Cities, First Edition. Edited by Amal El Fallah Seghrouchni, Fuyuki Ishikawa, Laurent Hérault and Hideyuki Tokuda.

Context-aware applications will recognize the current context of the city they are monitoring and actuate devices to amend their status. A key service in smart cities will be context management systems, which estimate context of cities and provide it to applications.

A context management system should be able to collect and update various types of content required by context-aware applications and should be able to be used easily in various environments. Here, a wireless sensor network (WSN) will be a key infrastructure in context management systems. A WSN is a wireless *ad hoc* network consisting of tiny computers equipped with sensors and wireless communication devices. It continuously records and produces data by measuring the environment via sensor nodes. It can produce one or more kinds of data, because its nodes are equipped with one or more kinds of sensors, and it can be programed to alter or switch tasks between the different sensor devices used for monitoring. Moreover, it can be easily used because it does not require any communication cables. Its nodes communicate with each other via wireless links and transmit the measured sensor data via multi-hop communications.

These features of the WSN are quite important for context management systems in smart cities. First, its "easy-to-deploy" feature is suitable for the smart cities. Sensor nodes are usually used in outdoor spaces, but it is not easy to connect sensor nodes using cables because of monetary and legal constraints. Second, the "reprogrammable" feature is suitable because a smart city usually hosts many applications that require different kinds of sensor data. A context management system should carefully balance the demands of these applications and the resource consumption of the sensor nodes. This can be realized by reprogramming a WSN. Therefore, the WSN is a key enabler for a context management system in a smart city.

A shared WSN for smart cities should:

1) support various kinds of measurements;

2) manage tasks at runtime;

3) adapt to changes in the environment to reduce unnecessary resource consumption.

A shared WSN is used by many context-aware applications, which require different kinds of sensor data and different levels of accuracy. Therefore, it should be able to handle various measurements to produce one or more kinds of sensor data required by these applications with a level of accuracy. Moreover, applications using a WSN appear and disappear at runtime. Therefore, the WSN should be able to add or remove tasks at runtime without having to stop and start. Finally, a WSN should be able to adapt its behavior in response to changes in the environment. A WSN has severe resource limitations because each node in a WSN has CPU, memory, bandwidth and battery restrictions, and resources must be saved to increase the number of tasks that it can handle and to prolong its lifetime. Therefore, the WSN needs to automatically adapt to changes in the environment to reduce unnecessary resource consumption, that is to say without human intervention.

To develop such a shared WSN, middleware supports are needed. This chapter describes an example of middleware for a shared WSN, called XAC middleware. In addition, we discuss the research issues related to shared WSNs and the techniques used in XAC middleware.

1.2. Background

WSN software development is not easy because it requires programmers to have an in-depth knowledge of various fields, such as analysis of sensor data, distributed programing in wireless *ad hoc* networks and optimization of embedded systems. This section presents examples of types of WSN software to identify the issues concerning shared WSNs.

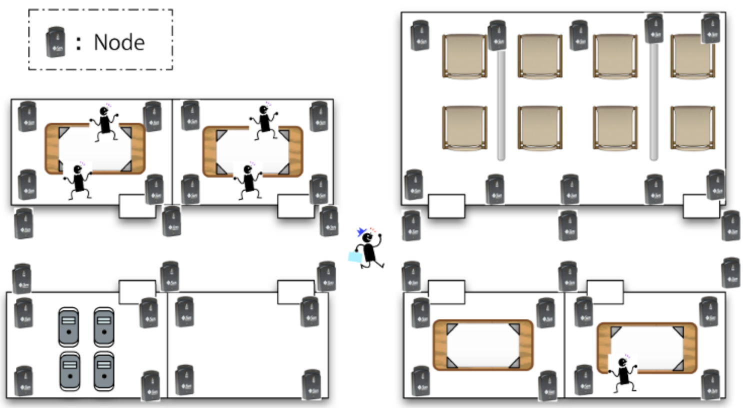

Figure 1.1. *A smart environment*

Let us first define our example– a smart environment is set up in an office building, where context-aware applications are introduced to optimize everyday business tasks. Consider the environment illustrated in Figure 1.1. Sensor nodes are used throughout rooms, corridors and stairwells to enable monitoring and to establish the current context of the building. Data sensed by the nodes are transferred via multi-hop communication to a central server (called the *base station* from here on).

Table 1.1 shows four scenarios, S1, S2, S3 and S4, envisioning context-aware applications in this example environment. In scenario S1, an application maintains the temperature levels in the conference rooms according to the preferences of the people in the room. The application in S2 determines the occupancy of conference rooms on the basis of the presence of people in the room and the reservation data of the room. Scenario S3 involves tracking applications that continuously monitor the current locations of staff inside the building, whereas the application in S4 detects suspicious intruders.

Scenarios	Application name	Operational tasks	Environmental information	Accuracy requirement
S1	Temperature management	Adjust room temperature according to preferences of people in the room	Temperature in the room	Within 2°C of actual value
S2	Meeting-room management	Maintain occupancy of conference rooms based on their current occupancy and reservations	Presence of people in conference rooms	Determine correct room occupancy with 99% accuracy
S3	Staff-tracking management	Determines the current locations of staffs	Location of staff	Within 1 m range in a public space or room
S4	Intruder detection	Determines suspicious intruders for instance by raising an alarm if people remain near an access lock for long periods without authenticating	Presence and position of people in certain locations	Within 2 m range

Table 1.1. *Examples of context-aware application scenarios*

Each application requires environmental context information related to its own operational tasks. S1 requires temperature data, S2 requires data on the presence of staff in each room, S3 requires data on the location of each member of the staff and S4 requires the location information of people in designated areas.

As we can see, each scenario possesses different non-functional requirements in terms of accuracy. Generally speaking, sensor data include a certain level of sensor error. A well-known way to improve accuracy is to aggregate sensor data coming from neighboring sensor nodes. For instance, knowing only the rough locations of staff (like the room in which a person is currently located) is enough for S3. In this case, the low accuracy requirements can probably be satisfied with sensor data from just one or two nodes. On the other hand, S4 requires the specific positions of staff to accurately track them within 2 m. This in turn entails gathering more sensor data than in S3.

Although many sensors are required from the viewpoint of accuracy, resource usage in a WSN should be kept as low as possible to prolong the lifetime of the network. Software developers should take into account the severe resource limitations of nodes in terms of CPU power, memory, communication bandwidth and so on when creating a WSN. In particular, the battery is a precious resource. For example, on average the battery of the commonly used Crossbow Mica2 node will deplete in just 7 days by reading its temperature sensor value and sending transmissions every second [SHN 04]. Even though the lifetimes of sensor nodes are gradually increasing as a result of hardware improvements, energy consumption is still an important issue in WSNs. Load concentrations on specific nodes will drain batteries quickly, which are then hard to recharge during runtime.

As such, to extend the network lifetime, it becomes necessary to use various optimization methods to extend each node's operational time, for instance, by aggregating sensor readings before transmitting them, by adjusting the sensing frequency to meet certain accuracy requirements or by duty-cycling node operation. To yield optimal results, these methods also need to comply with the requirements of multiple applications.

1.3. XAC middleware

XAC middleware is a middleware for a shared WSN. Its main features are as follows:

– WSN as a multi-modal sensor: XAC middleware uses the fact that a WSN is a multi-modal sensor. It hides the low-level details of the WSN from context-aware applications. A WSN can therefore be seen as a single sensor covering a large area by these applications. XAC middleware also provides a way to use the WSN for different measurement purposes. One or more applications can use the WSN at the same time.

– Runtime management: XAC middleware allows context-aware applications to register or unregister their measurement tasks during runtime.

– Self-adaptation: XAC middleware monitors changes in the WSN and adapts configurations in response to these to reduce unnecessary resource consumption and to maintain the required level of accuracy.