Self-configuration of "Home Abstraction Layer" via Sensor-Actuator Network

Zheng Hu[1,2], Gilles Privat[1],
Stéphane Frenot[2], and Bernard Tourancheau[2]

[1] Orange Labs, 28 Chemin du Vieux Chêne 38240 Meylan, France
{firstname.lastname}@orange-ftgroup.com
[2] CITI/INRIA INSA de Lyon, 6 Av. des Arts 69621 Villeurbanne, France
{firstname.lastname}@inria.fr

Abstract. We propose a mechanism and system for the identification, self-configuration, monitoring and control of non-networked home devices through a shared backplane of networked sensors and actuators. The resulting generic home abstraction layer interfaces to all kinds of physical entities of the home through a software proxy, as if they were state-of-the-art networked devices. The matching of the entities being discovered in the home/building environment to known semi-generic models is performed by iterative approximation. The architecture and OSGi-based implementation of this system is described. Examples are provided for typical home appliances and other subsystems of the home/building that may be dealt with in a similar way.

Keywords: Home as Smart Environment, Home device management, OSGi, Sensor, Actuator.

1 Introduction

Home automation systems have a decades-long track record, but they have yet to move beyond specialized applications tied to their own dedicated infrastructure. Sharing a multipurpose backplane of sensors, actuators, networks and local server/gateway devices to support a broad portfolio of home/building applications should make it possible to amortize the cost of this infrastructure across the board and to jumpstart the take-up of a home automation service portfolio that has, so far, proved vexingly elusive.

Among these applications, energy management has appeared as a new pacesetter, spurred on by the rising cost of energy and the requirement to shrink the carbon footprint of buildings, potentially warranting the investment in such a shareable open ICT infrastructure for the building. Home energy management services, offerings have so far been mostly limited to energy monitoring or simple load shedding/shifting. The next stage in home energy management systems will be towards the integration of all energy-relevant devices, appliances and components of the building in a comprehensive monitoring and control system relying on a shared infrastructure for this.

D. Keyson et al. (Eds.): AmI 2011, LNCS 7040, pp. 146–150, 2011.

The ReActivHome[1] project aims at designing and prototyping such a comprehensive home energy management system. This system is intended to manage and optimize at the home level the balance between energy consumption, generation and storage according to both local and global criteria. This system relies on the monitoring and actuation of the energy-consuming, -storing or -generating components, devices and legacy appliances of the home through a complete shared infrastructure of sensors and actuators. A key issue is then how to integrate all these entities in the perimeter of such a system, knowing that most of them do not have any kind of digital interface to a data network, neither for monitoring nor for control. Crucially, the economic viability of such a system mandates that setting it up should not be made conditional upon an all-out upgrade of the building and its appliances, and should not require a complex and costly manual configuration.

The solution we propose draws inspiration from networked-device configuration mechanisms such as UPnP or DPWS, applying similar concepts to non-networked legacy physical entities for which available sensors and actuators play the role of network interfaces. Addressing as target entities both devices and subsets of the home such as rooms, our approach conjoins two separate strands of research that had been pursued under the ambient intelligence research agenda[3], smart devices and smart environments.

2 Principle of the Home Abstraction Layer

The Home Abstraction Layer (HAL) we propose is an analogue of Hardware Abstraction Layers used by operating systems : it hides the specifics of the home hardware beneath a set of generic models and interfaces, acting as a generic informational interface to the home as a physical system. Within the HAL, Subsystem Identification Monitoring and Control (SIMC) modules act as proxies for each individual physical entity/subsystem of the home, using associated sensors and actuators, which can be shared among them (e.g. an infrared camera and a microphone could be used to monitor all appliances of a room).

2.1 SIMC Life-Cycle

The SIMC relies on a simple hybrid model of each target subsystem as a finite state machine, with associated continuous attributes. These models are not meant to be exact, they are supposed to be sufficiently generic to represent categories of subsystems. Figure 1 presents the finite-state machine model used by a room SIMC. Similar models can be used for usual home appliances.

The SIMC can be in either of two main modes: configuration or runtime. The configuration phase is used to identify the Runtime State Machine(RSM), the state machine model that best matches the entity, and associating sensors and actuators. There are three predefined states. The initialization state is active when the existence of a physical entity (such as an appliance) has just been

[1] ANR ReActivhome : https://reactivhome.rd.francetelecom.com

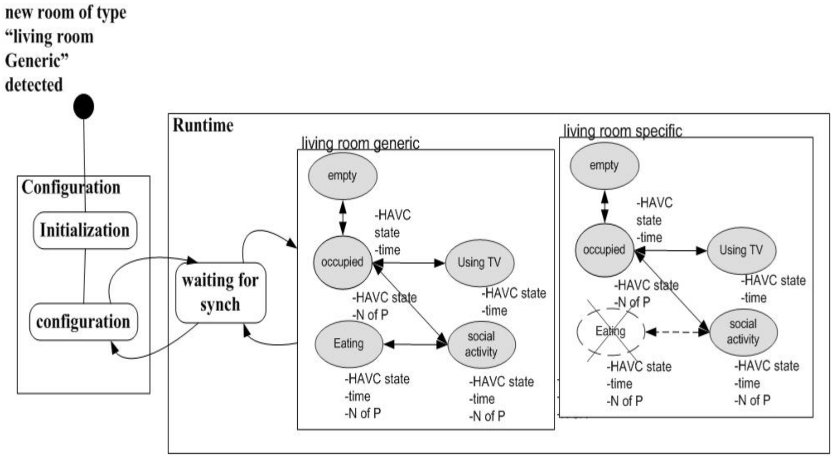

Fig. 1. Example SIMC life-cycle

discovered from a set of sensors. The configuring state is active when a dedicated SIMC instance is associated with real sensors/actuators and its RSM.The runtime mode maintains as persistent variables the current states of the SIMC while the physical entity works. In runtime mode the SIMC component may always go to the waiting for synch state when an error occurs, or when the state is unknown.Configuration is dynamic and may occur at anytime, following changes in the home environment.

2.2 SIMC Identification

The SIMC classification engine chooses one among several predefined prototype outputs from aggregate sensor inputs. and is used for the identification of the target physical entity model, the runtime identification of current states, and the state sequence conformance checking.

A ”root” SIMC contains a classifcation engine and a RSM. Its role is to monitor the global context. When new entities are detected, the ”root” SIMC tries to identify them and spawns a new generic SIMC. Those ”generic” SIMCs follow their own life-cycle and each can finally reach a specific SIMC by iterative approximation. Note that the ”root” SIMC, ”Generic” SIMCs and ”specific” SIMCs all have the same architecture.

The SIMC management system and architecture are independent of the particular classification algorithm used inside the classification engine. In our first prototype we use an Artificial Neural Network engine.

3 HAL Implementation with OSGi

3.1 SIMC Architecture in OSGi Framework

Our purpose in choosing OSGi[2] is to enable application reconfiguration at runtime. So, each SIMC functional module is modelled as a bundle[1], and its application functions are packaged in the bundle as services.

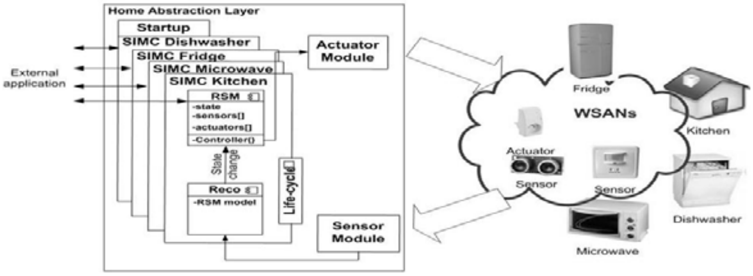

Fig. 2. SIMC architecture based upon the OSGi framework

To achieve the full functionality described in section 2, we model the architecture of the SIMC management system as shown in figure 2. Three bundles are initially started:

The Sensor Module bundle communicates with all the sensors in the HAL environment. Each piece of sensor data is transferred by the sensor module which achieves the following tasks: 1) Physical interface for heterogeneous sensor management[2]; 2)Identification and authentication of data source; 3)Cartography of the observed environment; 4)Dynamic integration of sensors according to their own protocol, for which we mostly (not exclusively)use Zigbee[4].

The Actuator Module, counterpart of the Sensor Module, takes in high-level commands expressed as state changes of target physical entities and fans them out as specific controls to the associated actuators.

The StartUp bundle waits for the activation of the Sensor Module and Actuator Module, in order to start the first ”root” SIMC. SIMC is made up of these operating services:1)The Life-cycle service is the core of SIMC, it is capable of starting or stopping SIMCs. It is invoked by the ”root” SIMC, and requests the creation of a defined type of device corresponding to a set of sensed data;2)Classification is based on a model which produces a unique output from various sensor inputs of sensed data in a single classifier. This output can be transferred to RSM;3)RSM service maintains the current state by receiving notification from the recognition engine. It can communicate with the application layer to provide the physical entity information as well as to receive commands for the physical entity.

[2] OSGi Alliance:http://www.osgi.org

3.2 Prototyping

As a proof of concept, we developed a prototype whose goal is to show how to handle sensor data, build an SIMC and notify client applications when the corresponding state changes. The prototype performs state identification on simple home appliances.

The physical characteristics that can be sensed are sound, vibration and temperature, together with electrical current. The first three were captured with a wireless sensor mote comprising of a microphone, a 2-axis accelerometer, a temperature sensor and a light sensor. For our test system, a single PC hosts the HAL running a SUN JDK 1.6 with the Apache Felix OSGi implementation. And the specific sensors setup that we use is a MEMSIC[3] MICAz wireless sensor mote and an MTS310 sensor board.

4 Conclusion and Future Work

The conceptual model described here has wide-ranging potential applications, which we have barely touched on. Limiting ourselves to the home environment for the time being, the aim is to validate this concept as a common foundational software layer that interfaces all kinds of legacy home entities for all kinds of ambient intelligence applications such as energy management, home automation, ambient assisted living, jointly broadening the scope of AmI and IoT research to non-networked entities that become interfaced through this common layer.

We have yet to validate the system with a broad set of target entities, appliances and rooms and corresponding models, monitored under varied operating conditions with a wide spectrum of shared sensors.

We will investigate more closely the potential improvement of the system from operating as a closed loop control system, where actuators come into play not only for the runtime control of the target home entities, but also for the self-configuration of the HAL system itself.

References

1. Bottaro, A., Simon, E., Seyvoz, S., Gérodolle, A.: Dynamic Web Services on a Home Service Platform. In: 22nd International Conference on Advanced Information Networking and Applications, pp. 378–385. IEEE (2008)
2. Gurgen, L., Roncancio, C., Olive, V., Labbé, C., Bottaro, A.: SStreaMWare: a Service Oriented Middleware for Heterogeneous Sensor Data Management. In: Proceedings of the 5th International Conference on Pervasive Services, pp. 121–130 (2008)
3. Streitz, N., Privat, G.: Ambient Intelligence. In: Stephanidis, C. (ed.) The Usniversal Access Handbook, pp. 1–60. CRC Press, Taylor & Francis Group (2009)
4. Ha, Y.-G.: Dynamic Integration of Zigbee Home Networks into Home Gateways Using OSGi Service Registry. IEEE Transactions on Consumer Electronics 55(2), 470–476 (2009)

[3] Crossbow/Berkeley sensor motes: http://www.memsic.com