Iterative Model-based Identification of Building Components and Appliances by Means of Sensor-Actuator Networks

Zheng Hu, Stéphane Frénot, Bernard Tourancheau, G. Privat

To cite this version:

Zheng Hu, Stéphane Frénot, Bernard Tourancheau, G. Privat. Iterative Model-based Identification of Building Components and Appliances by Means of Sensor-Actuator Networks. 2011. inria-00636055

HAL Id: inria-00636055

https://hal.inria.fr/inria-00636055

Submitted on 26 Oct 2011

6.4. Paper: Iterative Model-based Identification of Building Components and Appliances by Means of Sensor-Actuator Networks.

Zheng Hu, zheng.hu@orange-ftgroup.com
Gilles Privat, Senior Scientist, gilles.privat@orange-ftgroup.com
Orange Labs R&D, Grenoble, France
Stéphane Frenot, stephane.frenot@inria.fr
Bernard Tourancheau, bernard.tourancheau@inria.fr
INRIA/CITI INSA de Lyon, Villeurbanne, France

Abstract

A key challenge for home/building automation applications, especially energy management, is to enable the indirect integration of legacy home/building appliances and building components that do not have a network connection to the home/building network, to make it possible to monitor and partially control them. We propose a mechanism and system for the iterative identification and self-configuration of these devices through a shared backplane of networked sensors and actuators available in the building. This makes it possible to integrate these devices and interface them through a software proxy as if they were state of the art networked devices, thus extending the range o the network and the associated middleware towards all kinds of physical entities of that make up the home/building. These entities are supposed to be described in a model repository and a domain-specific ontology. The matching of the entities being discovered in the home/building environment to these known models is done by analog pattern matching, instead of requiring an exact match as would be the case with a standard digital networks protocol, so that it lends itself to iterative approximation. The architecture and OSGi-based implementation of this system are 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/Building Automation, Internet of Things, Ontology, Hybrid discrete-state model

6.4.1. Introduction

Home and building 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 applications 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 have been taking off slowly, with offerings that are so far mostly limited to energy monitoring or simple load shedding/shifting. Building energy management is more advanced when taken up by energy service companies, but still relies mostly on dedicated infrastructures. The next stage in home/building 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.

The ReActivHome project (https://reactivhome.rd.francetelecom.com) 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 sources and loads according to both local and global criteria. This system relies on the monitoring and actuation of these sources and loads through a complete shared infrastructure of sensors and actuators.

The project addresses the home environment, but most of what is put forward here can be directly transposed to the building environment, so that whenever we mention home in the following, it can be usually generalized to building.

A key issue in such a comprehensive home energy management system is how to integrate in the perimeter of the system the energy-consuming, -storing or generating components, devices and legacy appliances of the home, knowing that these do not have any kind of digital interface to a data network, neither for monitoring nor for control, and that setting up such a system should not be made conditional upon an all-out state-of-the-art upgrade of the building and its appliances where such interfaces could be taken for granted, Moreover, this integration should be as automated as possible and should not require a complex and costly manual configuration of the system by a skilled technician (or an extremely savvy and motivated end-user). The inspiration for this spontaneous integration can be in state of the art zero-configuration or network discovery protocols, especially at the level addressed by service-oriented architecture that make it possible to “recognize” devices according to a high-level model, supposed to be known and shared beforehand. How to emulate this spontaneous model-based identification of state of the art digital devices for legacy physical entities is the problem addressed by this paper.

The paper is organized as follows. We describe related work in this area in section 2. Section 3 gives an overview on the principle of the proposed approach and of the models it relies upon. Section 4 presents the overall system architecture. The more detailed architecture of the monitoring and control core component of the system is described in section 5, together with a prototype based on this architecture and experimentation. Finally, we will open a perspective on future work.

6.4.2. Related work

A broad concerted effort has been set up in the past few years to investigate the use of ICT for energy efficiency, under a specific chapter of the seventh Framework Programme. Among these projects we can mention AIM (www.ict-aim.eu/), Beywatch (www.beywatch.eu), and Beaware (www.energyawareness.eu/beaware/).

HOMES (www.homesprogramme.com) is a more short term industrial project whose purpose is to develop a framework for the integration of sensors and power-controllers to optimize energy management in buildings, targeting exclusively the building envelope and its fixed equipment. Like the ReActivHome project already mentioned that targets exclusively the home but broadens the scope to all kinds of energy-relevant equipment, it is a national collaborative project in France.

New possibilities offered by the home as a "smart space", equipped with a multipurpose shared backplane of sensors and actuators, had been widely explored in European FP6 projects dealing with ambient intelligence. Among these, the Amigo project (www.amigo-project.org) has been exploring early on the semantic-level interoperability of a middleware that would support both the individual home appliances and the sensor-based infrastructure providing context-awareness to the home as a smart space. It has been advocated (Privat, 2008) that this vast body of research can now be exploited for a new application, energy management, that could lead to a renewed, more pragmatic interest in smart spaces.

With a narrower focus on devices themselves rather than on the smart space, SOA-like and semantic-level distributed software infrastructures for the home with spontaneous configuration capabilities have been widely addressed and are an obvious inspiration for the solution presented here is, as already mentioned, what is done with classical zero-conf mechanisms for networked devices, at all levels. Besides the efforts from standardization bodies to elaborate complete solutions such as UPnP and DPWS, we can mention the following. Kushiro (Kushiro, Suzuki, Nakata, Takahara, & Inoue, 2003) provides a Residential Gateway Controller with Plug & Play mechanism. In their architecture, they integrate home appliances using the Echonet (www.echonet.gr.jp/english/8_kikaku/index.htm) communication protocol. Joo I. et al. (Joo, Park, & Paik, 2007) developed an ontology for Intelligent Home Service Framework (iHSF). In the iHSF architecture, a device handler has been designed. The system manages all existing devices in the home by means of the device handler. They assume that a preexisting mechanism brings and configures together the handler and the corresponding physical device. DomoNet (Miori, Tarrini, Manca, & Tolomei, 2006) is an approach that helps to integrate conventional home automation systems following the service oriented computing paradigm. DomoNet architecture describes a SOA model which essentially consists of a network connecting application gateways called TechManagers(TMs). Each TM handles one home automation subnet such as UPnP and X10.

All of these solutions require not only that the target devices should be natively endowed with the proper network interfaces, but also that these interfaces comply with corresponding standards at all appropriate levels, up to the semantic level. They also require that the most specific device types are known beforehand for their physical instances to be discovered. The proposed approach differs radically from this previous work in broadening the perimeter to integrate all non-networked legacy home appliances. It requires neither a prior equipment of these devices with standard interfaces, nor that these devices are known on the basis of the specific type or make that they belong to, as devices are identified only by approximation to a generic model. To our knowledge it is the only architecture whose goal to enable a comprehensive and spontaneous integration of such devices in the home/building environment.

Another very important strand of research and development has pursued a similar goal of network integration of physical entities, with a focus on the low-end of the device spectrum. This is what the Internet of Things is all about, at least in its most widely received acceptation. RFID is, under all its variants and guises, the most widespread technology for this, but the Internet of Things should not be limited to any such individual technology. The broader conceptual underpinnings of the present work have been laid out in (Privat, 2011) under the definition of phenotropic interfaces that make it possible to extend the reach of networks beyond sensors, so as to include in the network analog physical objects that have neither an interface in a digital networking protocol, nor a universal digital identification scheme such as RFID. In the approach presented here, the "things" to be identified and attached to the home/building network are the appliances or building components that are identified by iterative approximation to a model instance. Phenotropic interfaces, i.e. interfaces that operate through pattern recognition-from multidimensional and multimodal sensor data, are the intermediary that supports this identification process, so that the present work goes clearly much beyond RFID and the concept of phototropic interfaces.

Finally, in addressing as "subsystems" of the home/building both individual devices and subsets of the building such as rooms, our approach reunites the two separate strands of research that had been pursued under the ambient intelligence agenda (Streitz & Privat, 2009) : smart devices/ things on the one hand, smart spaces on the other hand.

6.4.3. Principle of the proposed approach

Target home subsystems

The problem addressed by our proposed approach can be stated in the most general possible way as follows (Figure 33): a generic extension of a home ICT system is set up to monitor and control individual physical entities that are self-contained sub-systems of the home itself and fully-fledged physical systems in their own right : they can be, at their most relevant level of granularity, individual rooms of the home or regular home appliances/devices (possibly including non-electrical appliances and energy generation or storage devices). Sensors and actuators are distributed as monitoring and control intermediaries through the home. Absent any native ICT interface on these subsystems sensors and actuators are supposed to be the only means to monitor and control the target physical subsystems, providing an indirect distributed coupling between the home ICT system and the home as a physical system.

The target subsystems of the home are defined as the components of the home that are relevant for being controlled and monitored by an application such as energy management, home security or home automation, using the ICT system that is being setup in this way as a generic supporting layer ,

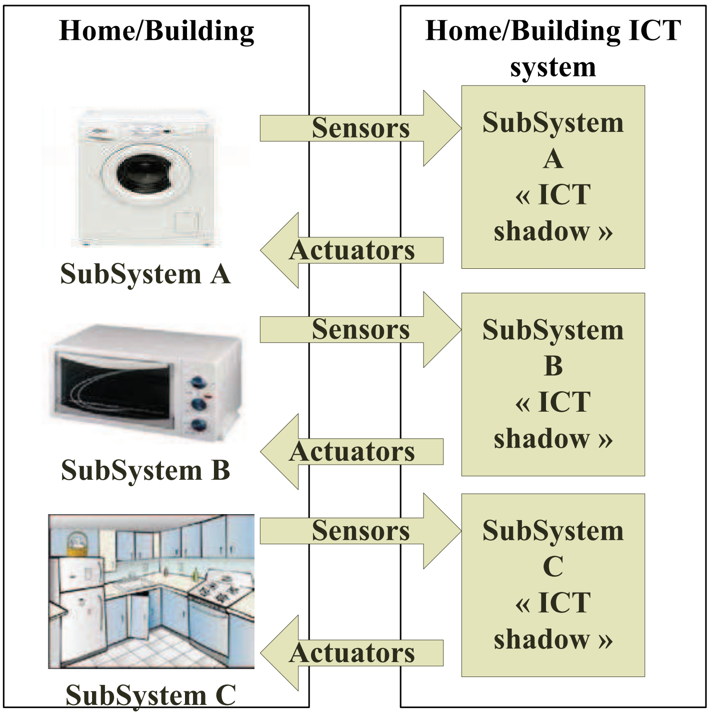

Figure 33**:** Principle of the proposed approach

The ICT system will "shadow" or mirror these physical entities/subsystems individually through matching self-contained ICT subsystems that will in turn be the primary building blocks of its own architecture. The ICT system should have the capability to create and configure these entities automatically, both for the initial configuration stage and when reconfiguration would be needed because of a change in the environment. This implies the need to associate automatically the subset of sensors and actuators that are used as intermediaries for the monitoring and control of a given subsystem, and to update the model of the subsystem accordingly.

The proposed approach is to dynamically create a "shadow" ICT component for each individual target subsystem of the home. We call this component the Subsystem Identification, Monitoring & Control component (SIMC) in the following. It connects to the physical subsystem through a dynamically configured set of potentially shared sensor and actuators, chosen among those available in the relevant environment. The sensors detect events and changes of state from the physical subsystem as their inputs and the SIMC sends the controls to the actuators to effect required actions on the physical subsystem. The system comprising the SIMC together with the physical subsystem and the sensors and actuators makes up a closed-loop control system.

Home subsystems ontology

Our approach is based on a pre-defined ontology that captures generic knowledge about the devices, appliances and rooms of the home domain through a multi-criteria hierarchical categorization and the definition of generic hybrid finite state models for each of the target subsystems. Initiating and configuring the SIMC for a given entity actually amounts to identifying, loading and iteratively adapting the most appropriate subsystem model from this model repository.

This ontology subsumes several relevant categorizations of the appliances or room. As illustrated in Figure 34 it can be structured as a directed acyclic graph that makes it possible to follow a path from the most generic parent models (closer to the root of the graph) to the more specific models (closer to the leaves). At the root is the main class *subsystem*, as understood above, which in the scope of this paper specializes into two descendant classes: appliance and room. Appliances and rooms can in turn be specialized and classified according to several different criteria that are either intrinsic to their main usage, or relative to the application (in our case energy management and energy efficiency). Examples of these criteria are illustrated below, each of them corresponding to intermediate nodes in the graph

Models are associated not only with the terminal nodes of the graph corresponding to the most specific categories, but also to the intermediate nodes. This full hierarchy of models provides a mechanism to identify a subsystem in an incremental way on the basis of observation data, starting from the most generic model if observed sensor data is inconclusive, and refining this first match to more specific models down the graph when further observation data becomes available.

We chose to model both devices and rooms with hybrid finite-state discrete-time models, where state information is possibly complemented with continuous-valued attributes. These states and the relevant attributes are then stored as the state of the shadow ICT subsystem. These hybrid models represent a tradeoff between expressivity and ease of identification. The full description of a physical system such as the target rooms and appliances would normally require a continuous state & continuous-time model, but automatic identification of the parameters of such models would be nearly impossible. Examples of such models are given below for a category of appliance (a generic washing machine) and a category of room (a living room modeled by its different states).

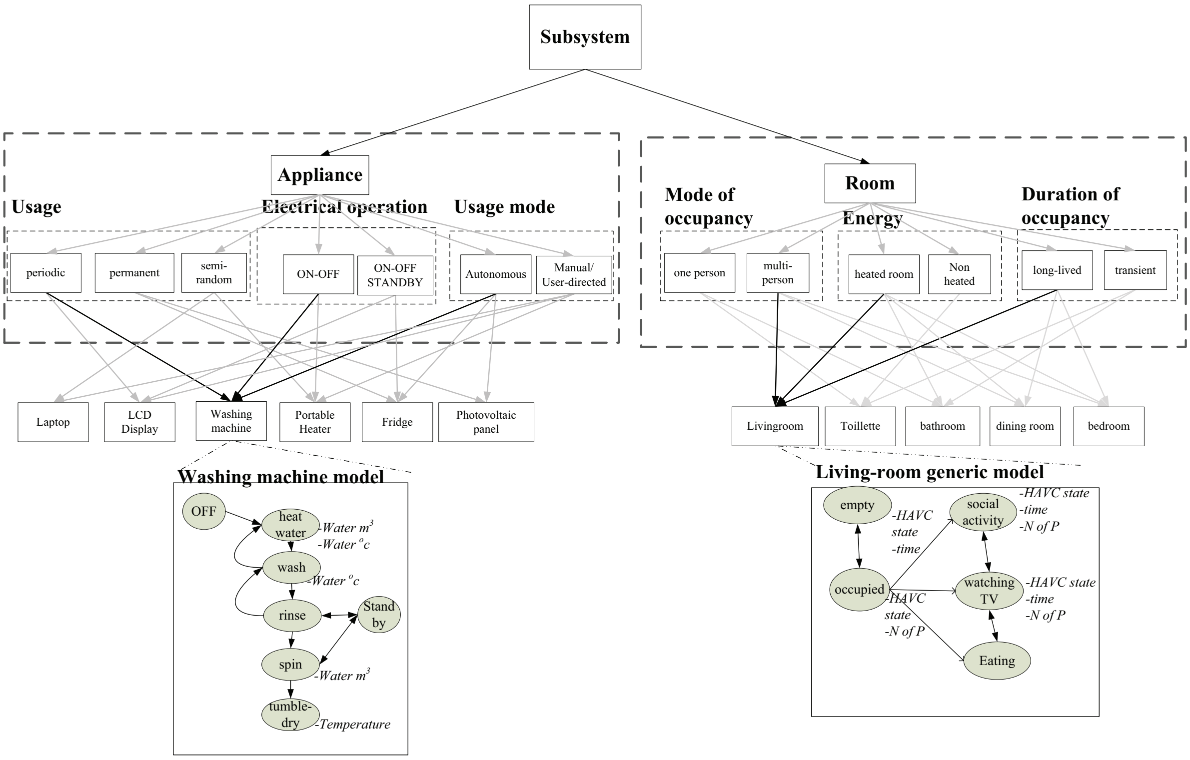

Figure 34 : Home subsystem ontology and corresponding hybrid discrete state model examples

Model identification and parameterization

Model identification is the process of matching a target subsystem to a category (a node of the graph) in the domain-specific subsystem ontology. This category may be very broad or very specific, depending on how much sensor data the system has available to perform the identification with. This category is associated with a default model of the target subsystem.

Once a given model has been identified, parameters for the corresponding model may be updated incrementally from new sensor data from the runtime SIMC, or the model may be modified so as to better fit observations: this may correspond to either changing continuous-valued state attributes, invalidating state transitions, or removing states from the state diagram. An example of this updating process is shown in Figure 35 for a washing machine, where the final (most specific) model reached in the model identification stage is refined by removing an unused state or an unused transition (in this case the washing machine is updated from the generic one as having no tumble-drying function) Adding new states is not considered possible here, and if the model does not fit the observations it is always possible to revert to the previous stage, starting with a new model one rung higher (i.e. less specific) in the hierarchy.

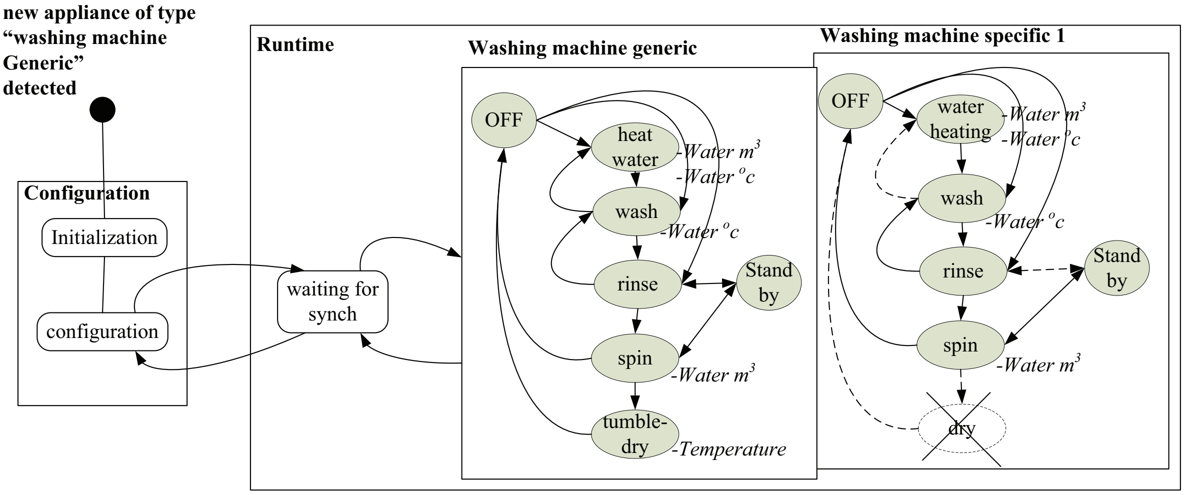

Figure 35 : Model update example

SIMC life-cycle

We now detail the SIMC life-cycle and the configuration mechanism involved in SIMC creation. As mentioned above, Figure 35 presents the finite-state machine model used by a washing machine SIMC. The SIMC can be in either of two main modes: configuration or runtime. The configuration phase is used to associate the kind of specific state automata that best represents the appliance, in our concept it is called Runtime State Machine (RSM). During the runtime phase, it represents not only the states where the SIMC may be in but also but also checks the conformance via the state sequence matching. The configuration phase consists in associating sensors, actuators and a Runtime State Machine (RSM). There are three predefined states. The newborn state is active when the existence of a physical washing machine has just been discovered from a set of sensors. The configuring state is active when a dedicated washing machine SIMC instance is associated with real sensors/actuators and its RSM. The destroying state is active when the physical washing machine is removed from the home or disconnected from the system; it corresponds to the deletion of the corresponding SIMC instance from the system.

The washing machine Generic Runtime maintains as persistent variables the current states of the SIMC while the physical entity works. For example ON/OFF states are specific to the generic kind of detected appliance. 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.

A reconfiguration stage may occur at anytime. For instance if we found there is no dry state for the washing machine, we go back in configuring mode, change values and set the SIMC in the waiting for synch state.

SIMC classification engine

The SIMC classification engine is a fundamental element of the system. As a classification or pattern recognition engine, it selects the best approximating prototype from a multidimensional dataset of sensor reading. Various algorithms from the machine learning repertoire may be used for this. Based on this principle, it is used in three cases: the identification of the best approximating model of the target physical subsystem; the identification of the instantaneous state of the subsystem according to this model, and the state sequence conformance checking. For instance, at the beginning, the recognition engine should identify the type of detected washing machine and spawn a generic washing machine model. Then the new captured data should be used to find which state the washing machine is in, matching it to one of the defined states of the automaton.

6.4.4. Architecture and implementation

Overall Architecture

The SIMC management system is hosted on a dedicated home server called Home Automation Box (HAB), itself connected to a wireless network of sensors and actuators (WSN). As shown in Figure 36, the system hosted on the HAB maintains a virtual representation of each physical entity as a distinct SIMC instance. Those representations interface the application layer with the managed physical subsystems. At the bottom of the HAB architecture a sensor/actuator layer provides an abstraction of the various types of connected sensors (Gurgen, Roncancio, Olive, Labbé, & Bottaro, 2008).

Each SIMC is associated with a set of sensors and actuators, which are shared between different entities. For instance, a microphone can be used to detect noise originating from different appliances; an infrared camera can also detect heat radiation patterns from several appliances. In the home context, SIMCs associated with rooms may “contain” those associated with appliances in this room in the sense that a room SIMC can use a heating SIMC as one of its associated actuators to effect changes in temperature, or use another appliance as a presence sensor. Our architecture assumes that the sensor-actuator network is available and has been deployed beforehand. Contrary to these “generic” sensors and actuators, the SIMC of an appliance used as a sensor or actuator is configured during the initialization phase of the system.

We focus on the legacy appliances which should be integrated in the management system in a similar way to what is done with regular networked devices. This means that they have to be identified and matched to an existing type of equipment

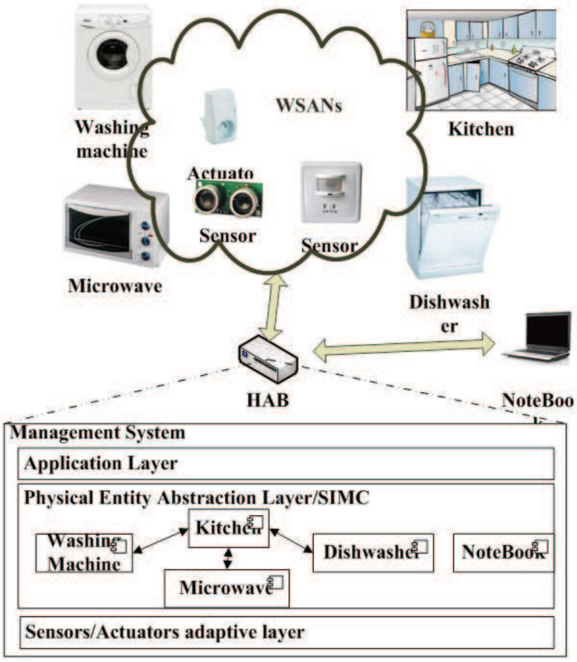

Figure 36: The SIMC management architecture

Figure 37 shows how the SIMC management system handles subsystems. A "root" SIMC contains a recognition engine and a runtime state machine (RSM). Its duty is to monitor the global context. When new appliances are installed, the "root" SIMC tries to identify them and spawns a new SIMC with "Generic" type. Those "Generic" SIMCs work in their own life-cycle and each can finally reach a "specific" SIMC. Note that the "root" SIMC, the "Generic" SIMC and the "specific" SIMC has the same architecture, the purpose is to keep the same recognition mechanism.

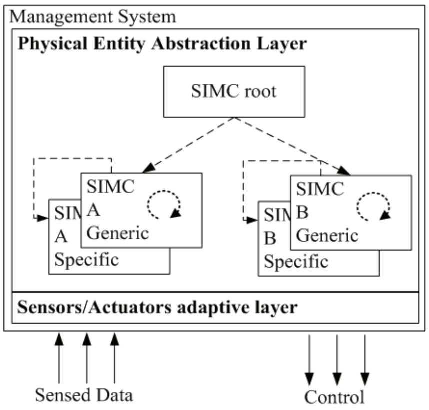

Figure 37: SIMC mechanism illustration

OSGi-based implementation

The OSGi specifications (OSGi, 2009) define a standardized, component-oriented computing environment for networked services that is the foundation of enhanced service oriented architectures. Using the OSGi framework for HAB adds the capability to manage the lifecycle of software components. They can be installed, updated, or removed on the fly without ever having to disrupt the operation of the HAB. Core features of OSGi are based on an original Java class loader architecture that allows code sharing and isolation between modules called bundles (Bottaro, Simon, Seyvoz, & Gérodolle, 2008). A bundle contains Java classes that implement zero or more services. Bundles are deployed in an OSGi service platform to provide application functions to other bundles or users (Lee, Nordstedt, & Helal, 2003).

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

Conclusion and future work

New home applications such as energy management require the extension of present-day Home Area Networks to legacy non-networked devices and building components, so as to acquire fine-grained real-time information about the operation of these entities and control them in return. We have described a system that identifies and monitors these non-networked entities using a shared sensor-actuator backplane. The identification is performed incrementally on the basis of a hierarchy of predefined models that approximate the behavior of the target entities according to relevant criteria. We prove the feasibility of our solution by validating the prototype on an OSGi execution environment.

In our future work we plan to test the system with a larger set of devices and corresponding models under varied operating conditions. 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 plant, but also for the self-configuration of the control system.

Taking in a classification of home appliances such as the one from EHS (European Home System, www.ehsa.com) we will draw upon related work (Sommaruga, Perri, & Furfari, 2005) (Joo, Park, & Paik, 2007) to further enrich and refine the home devices ontology.

We plan to investigate fuzzy-state models (Reyneri, 1997) as a potential enrichment of the hybrid discrete state models we presently use.

We will also port the system to an energy-efficient device such as a plug computer (www.plugcomputer.org) that is well adapted to be used as an always-on server inside the home, in order to evaluate the performance and scalability issues of the SIMC system on such a resource-constrained platform, notably which parts would need to be offloaded on a remote service platform across the wide area network.

References

Bottaro, A., Simon, E., Seyvoz, S., & Gérodolle, A. (2008). Dynamic Web Services on a Home Service Plateform. *22nd International Conference on Advanced Information Networking and Applications*, (pp. 378-385). Okinawa.

Gurgen, L., Roncancio, C., Olive, V., Labbé, C., & Bottaro, A. (2008). StreaMWare : a Service Oriented Middleware for Heterogeneous Sensor Data Management. *Proceedings of the 5th international conference on Pervasive services*, (pp. 121-130).

Joo, I., Park, J., & Paik, E. (2007). Developing Ontology for Intelligent Home Service Framework. *IEEE International Symposium on Consumer Electronics.*

Kushiro, N., Suzuki, S., Nakata, M., Takahara, H., & Inoue, M. (2003). Integrated Residential Gateway Controller for Home Energy Management System. *IEEE Transactions on Consumer Electronics, , 49(3)*, 629-636.

Lee, C., Nordstedt, D., & Helal, S. (2003). Enabling Smart Spaces with OSGi. *IEEE Pervasive Computing , 2(3)*, 89-94.

Miori, V., Tarrini, L., Manca, M., & Tolomei, G. (2006). An Open Standard Solution for Domotic Interoperability. *IEEE Transactions on Consumer Electronics , 52(1)*, 97-103.

OSGi. (2009). *OSGi Service Plateform Core Specification Release 4 Version 4.2.*

Privat, G. (2011). "Phenotropic and Stigmergic webs, the new reach of networks. *Journal of Universal Access in the Information Society* (3).

Privat, G. (2008). Ambient intelligence for indoor energy management. *ICT Event.* LYON.

Reyneri, L. (1997). An Introduction to Fuzzy State Automata. *Biological and Artificial Computation: From Neuroscience to Technology , 1240*, 273-283.

Sommaruga, L., Perri, A., & Furfari, F. (2005). DomoLM-env: an ontology for human home interaction. *Procedings of SWAP, the 2nd Italian Semantic Web Workshop.* In Proceedings of SWAP, the 2nd Italian Semantic Web Workshop .

Streitz, N., & Privat, G. (2009). Ambient Intelligence, Chapter 60. In C. Stephanidis, *The Universal Access Handbook.* CRC Press.