Decentralized Supervision for Home Automation

Renato Jorge Caleira Nunes
IST - Technical University of Lisbon / INESC-ID
Lisbon, Portugal
E-mail: Renato.Nunes@inesc-id.pt

***Abstract*—Today many home automation technologies exist and new ones are emerging, especially wireless. However, those technologies are incompatible with each other and address only the communication protocols and physical media used to interconnect the home automation devices. No normalized solutions exist specifically for the supervision of home automation systems and the definition of the required behavior for a home. In this paper we tackle this problem. We start by presenting an abstract model for home automation devices, independent of the technology used. Next we address system supervision and propose a model to specify the intended behavior for a home. This model is easily understood by common users and allows building generic interfaces to define and modify the behavior of homes. Finally, we propose a decentralized architecture to implement our system supervision approach, offering a flexible and scalable solution.**

I. INTRODUCTION

Home automation is slowly becoming more popular and this tendency will increase in the future. Currently there are many standard and open technologies available for home automation (for example, X10 [1], CEBus [2], LonWorks [3], EIB / KNX [4] [5]) and also many proprietary technologies. This latitude of choices and the fact that those technologies are incompatible with each other helps confusing the market and, in our opinion, contributes significantly to limit the expansion of home automation.

Technology is evolving very rapidly and new standards are being developed. For example, the recent standard on wireless communication ZigBee / IEEE 802.15.4 [6] targets consumer-oriented applications and home sensing and control. Although recent developments offer better and powerful solutions to home automation, no approaches have been proposed regarding interoperability and system supervision. This constitutes a shortcoming in current panorama as it restricts the capability to mix different technologies to benefit from each one's strong points or to allow expanding a system with new and enhanced products.

It would be very interesting to have the equivalent of BACnet [7] (developed in the context of Intelligent Buildings) as a lingua franca for home automation technologies. To be successful, such a standard should be well adapted to the application domain, be flexible, simple to implement and allow an easy interface with current technologies. Regarding interoperation, one should also mention approaches such as UPnP [8], Jini [9] and OSGi [10]. However, these were developed in a different context and are not specifically suited to home automation [11]. In particular, they impose some restrictions (e.g., the exclusive use of TCP/IP or Java programming environment) and their implementations are complex and, typically, require significant hardware resources.

Concerning system supervision, no normalized solutions exist for the definition of the required behavior for a home. Currently, the simpler systems follow an approach where functionality is hardwired into the system and cannot be changed without intervention of a knowledgeable person or technical personnel. More complex systems may include a controller and an interface that allows the user to issue commands and set some parameterizations such as schedules for predefined tasks or define lighting ambiences. However, the capabilities offered are usually quite limited and we strongly advocate that the users should have more control over their homes, being able to define and customize the system's behavior to their needs and preferences.

In this paper we start by presenting a generic and technology independent model for home automation devices. Next we address system supervision and present a proposal for specifying the intended behavior for a home. Finally, we detail a decentralized architecture that supports our system supervision approach, offering a flexible and scalable solution, which also allows the integration of different home automation technologies.

II. A MODEL FOR HOME AUTOMATION DEVICES

To allow a generic and flexible means of managing the behavior of a system, independently of the particularities of each device, it is fundamental to have an abstraction model to represent uniformly every device.

Our model is very simple but proved to have the required expressiveness. It assumes that each device is described by a set of properties and each property has a value. As an example, a light can be represented by a set of two properties: *OnOff* and *LightIntensity*. And a temperature sensor can be represented by a single property: *TemperatureValue*. It is also possible to represent complex apparatus such as an air-conditioner, a hi-fi or a television.

Only the relevant entities, over which one wants to interact, need to be expressed. In the case of a television, for example, that could be just the sound volume, mute and channel.

An important advantage of this model is its simplicity and uniformity of treatment of devices, which are just collections of properties that can be read or written to. Of course, sensors' properties can only be read and actuators' properties can be read or written. In this last case, writing to a property is the way to perform an action.

With our model it is easy to add new devices at any time as that just involves specifying the required properties for them. The different meanings of each property and other particularities are only apparent at a high level and regarding interaction with the user. At this high level, properties have a name, a description, a range of valid values, and strings associated with specific values to help its interpretation and simplify direct actions. For example, in the case of the property *Function* of a hi-fi, instead of using cryptic values one could just select "Radio", "CD" or "DVD". At the user interface level these strings will be translated into the correct property's value. All the information mentioned is specified in a generic way using XML. For further information refer to [12]. We have also developed a Web application that allows an interactive definition of home automation devices [13].

We assume that the interaction with actual devices is done through the exchange of messages, which specify a device's property, an operation to execute (read or write), and a value (in the case of a write). This model can be mapped easily into available home automation technologies, although in some cases that may require the usage of interface modules with some additional capabilities. That is the case of X10, for example, as typical devices do not support bi-directional communication. So, the interface module will have to monitor all actions and record each device's state, to be able to answer specific inquiries.

III. SUPERVISION MODEL

For the supervision model we propose also a simple approach based on the use of IF / THEN clauses. These can be mapped, at a user level, on the intuitive notion of *scenarios*. We may have scenarios such as "Waking up", "Back from work", "Watch TV", "Going to bed", "Holiday". A scenario corresponds to a set of actions that are executed when some condition occurs. The users have the ability to define whatever scenarios they wish or modify existing ones.

To define a scenario a user specifies an activation condition that can be as complex as needed. A condition may involve multiple tests of properties' values, combined using the logical operators AND, OR and NOT. A condition may also refer to time, which is also treated as a property's value (of the device *clock*) to maintain the uniformity of the model.

When a scenario's condition becomes true, a specified set of actions will be executed – *activation actions*. Optionally, the users may also specify a set of *deactivation actions* that will be executed when a scenario becomes inactive.

Using this approach we can design a simple and generic interface that allows a common user to browse the available devices and their properties (present in the system's description file in XML), and specify what tests to perform regarding a given scenario and the corresponding activation and deactivation actions. Browsing of devices can be done through lists or a graphical representation of a house's plan. A test compares a device's property with a given value. An action specifies a value to attribute to a property.

As illustrated by our description, our model of defining scenarios and specifying the behavior of a home is simple to understand and to use by common users, offering a flexible means for the home dwellers to customize and adapt their homes to their needs and preferences. In this way users can benefit from increased functionality from their systems, adding value to their investment.

Next we propose an approach to the implementation of our supervision model, which can also support interoperation between different home automation technologies.

IV. DECENTRALIZED SUPERVISION

Our objective is to be able to supervise home automation systems that may be complex and have a very big number of control points. Using the simple model we proposed, a key aspect is the evaluation of the activation conditions of scenarios. Each condition may involve the execution of multiple tests of devices' properties. This implies knowing the value of those properties, as they evolve over time, and evaluate the logical expressions to decide which scenarios become active and which must be deactivated.

A centralized solution is possible, using the same PC (or equivalent platform) where scenarios were defined. However, this centralized solution has several disadvantages. In systems with many devices, the communication channel with the PC may become a bottleneck, affecting the system's operation or even preventing it from working properly. A large flow of messages may also have a negative impact in the system's performance, especially if there are many scenarios defined. Finally, from a reliability point of view, the PC becomes a critical point as any fault will disrupt supervision actions and have a huge impact on system's operation. Only in small systems the centralized solution is suitable.

So, we propose a decentralized approach to system supervision, which is illustrated in figure 1. In this figure we represent a hierarchy of entities, where we identify the home automation devices, Test Processes (TP) and Scenario Processes (SP).

Test Processes are responsible for receiving information concerning changes in devices' properties and they execute the tests that exist in the scenarios' conditions. In practice, they receive messages whenever a value measured by some sensor changes, when someone presses a switch, when an actuator is activated or when any other event occurs associated with physical devices or the passing of time.

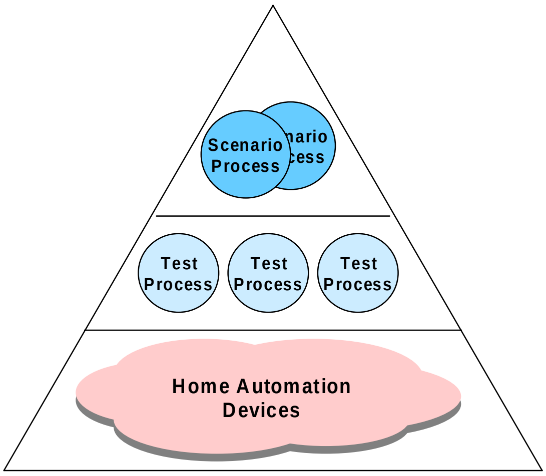

Figure 1. Decentralized supervision.

Regarding supervision, the task of continuously receiving status information and verifying which tests become true or false is the most demanding in terms of communication traffic and processing. For this reason, we may have as many Test Processes (TP) as required. Each TP will be associated with a specific set of devices, receiving only their status information and evaluating the tests that refer to those devices. In this way we offer a flexible and scalable approach that can be applied to systems of any dimension.

The Scenario Process (SP) is responsible for the effective execution of scenarios, i.e., evaluation of conditions and sending of messages specifying what actions to carry out. The role of a SP is simplified as it only receives information already filtered by TP. For example, while a TP may receive 1000 messages from a single sensor, the SP will only be notified when a given test becomes true or false, having a small flow of incoming messages.

The messages that inform of a test becoming true or false are called notifications. Each notification has a unique id and a binary value (true or false) that reflects the result of a test. Whenever a SP receives a notification it evaluates the activation conditions (logical expressions) of available scenarios and determines which scenarios become active or inactive. Accordingly with the results, it may emit the corresponding activation or deactivation actions.

It is possible to have just one SP, even for a big system, as its assignments are not very demanding. However, it is not difficult to implement a solution with several SP, which can offer advantages, mainly regarding reliability or in situations with a great number of complex scenarios. A scenario has, typically, a system wide nature, but we can decompose scenarios into parts (sub-scenarios) and attribute them to different SP. Each SP will supervise a part of the overall system. For a given scenario, each SP will have the same activation condition but, regarding the actions, each SP will have only those that refer to devices belonging to the part of the system it is associated with. For this to work, tests performed by TP must generate notifications to all the relevant SP.

V. SUPERVISION ARCHITECTURE AND INTEROPERATION SUPPORT

In this section we propose an implementation of the decentralized approach described earlier. Our proposal has also the ability to support interoperation between different home automation technologies.

Figure 2 illustrates the basis of our approach in which a system is decomposed into parts and, for each one, we suggest the usage of a Supervision Module (SM). Each SM runs a Test Process (TP), described in the previous section. The dimension of each system's part may vary and may map the physical structure of the home, such as the floors (basement, first floor, attic, etc) or individual divisions in the case of homes with a very big number of control points.

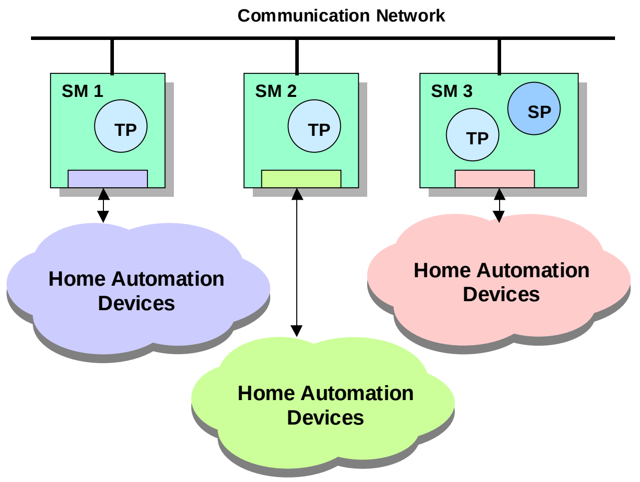

Figure 2. Decentralized supervision.

This division of a system into parts is natural for many technologies that support large systems. In EIB / KNX [4] [5], LonWorks [3] and CEBus [2], it is common to divide a system into network segments, for efficiency and other reasons, which are then interconnected with bridges or routers. That division is also typical when different communication media are used. In this way, our proposal can be easily fit into those technologies. However, it is important to note that the system division can be just logical. In fact, we may have more than one SM in a single network segment, each monitoring a different set of devices.

We propose also that one (or more) SM will run a Scenario Process (SP). For a small or medium size system a single SP will suffice. For bigger systems, two or more SP may be used. A SM does not need to be a complex or powerful device. It can be a simple and relatively inexpensive embedded controller, and only needs to interface with the home automation network in order to monitor devices and send commands. The interconnection between various SM can use the home automation network or a data network such as Ethernet, which offers very good bandwidth and simplifies interaction with other equipment and remote access through the Internet.

A big advantage of the proposed architecture is its ability to allow integration of different home automation technologies. Each system's part may use devices of a specific technology. The corresponding SM will interface and interact with those devices, mapping our generic model of a home automation device (described in section 2) into the particularities of the technology used.

With our approach the user can use a single paradigm to specify the intended behavior for the home and, seamlessly, monitor and control devices of different technologies.

VI. IMPLEMENTATION OVERVIEW

In this section we present an overview of some implementation aspects of Test Processes (TP) and Scenario Processes (SP).

Given the uniformity of our model it is possible to express an IF / THEN clause as a compact block of data. Each action consists simply of an id of a device's property and a value to attribute to it. A condition is more complex and is composed by a logical expression that may involve multiple tests of properties' values.

A test (for example, KitchenSensor.Temperature > 24) is expressed as an id of a property, a code specifying the type of test (equal, greater, less than, etc) and a constant value. There is an id associated with each test, which will be used in notifications (recall section 4). Information regarding tests is handled by the Test Processes, which send the corresponding notification to a Scenario Process when a test becomes true or false. Scenario Processes receive notifications and evaluate scenarios' conditions, which are expressions represented in data structures containing logical operators and notification ids (operands). If a scenario is activated, the Scenario Process will emit the associated actions, which are also stored in data structures.

The definition of scenarios is typically done at a PC and the corresponding data structures are downloaded to Test Processes and Scenario Processes located in Supervision Modules. After download, the PC can be turned-off as the supervision actions are effectively executed at the Supervision Modules. These modules are based on embedded controllers or single board computers, which can be much more reliable than common PCs (considering both hardware and software), have much less power consumption and do not have mechanical components such as disks and fans. The amount of memory needed to store scenarios' data is small and we can use solid state memory (FLASH or SRAM with battery backup).

In the context of the development of the DomoBus system [14] we were confronted with other needs that could be easily added to the Test Processes. So, we extended their functionality as described below.

As Test Processes receive all changes in status of the devices they monitor, they can easily register that information and be able to directly answer requests about the status of a specific device. For example, if we have a user interface that allows a user to monitor and command a home, when information is needed regarding a particular device, its status can be obtained questioning the corresponding Test Process and avoiding querying the device itself. This can offer performance advantages, especially if Supervision Modules are connected by a high throughput data network.

Furthermore, we implemented a publish-subscribe mechanism in which applications that require knowing the status of some device can subscribe that information. In this way, whenever a change in a device status occurs the subscriber applications will be informed. Additionally, as the Test Processes can perform tests over devices' status (values of properties), we introduced the notion of conditional subscriptions. In these, the subscribing applications specify a condition for notification (for example, the value of a device's property being greater than a certain value). The application will be informed only when that condition is satisfied. This capability has a positive impact as it can reduce very significantly the network traffic.

VII. CONCLUSIONS

In this paper we presented a decentralized approach to home automation supervision. It is based on an abstract model that offers a uniform and generic way to represent home automation devices, independently of the technology used. We proposed also a simple model to specify the behavior of a home and described how supervision actions can be implemented in a decentralized way, offering a flexible and scalable solution. Our approach allows targeting systems with a very big number of control points and allows integration of different home automation technologies.

REFERENCES

[1] X10, http://www.x10.com

[2] CEBus - Consumer Electronics Bus, http://www.cebus.org

[3] LonWorks, http://www.echelon.com

[4] EIB - European Installation Bus, EIB Association, http://www.eiba.com

[5] KNX, Konnex Association, http://www.konnex.org

[6] ZigBee wireless technology (IEEE 802.15.4), http://www.zigbee.org

[7] BACnet - Building Automation and Control network, http://www.bacnet.org/

[8] UPnP - Universal Plug and Play, http://www.upnp.org

[9] Jini Network Technology, http://java.sun.com

[10] OSGi - Open Services Gateway Initiative, http://www.osgi.org

[11] Sérgio Silva, "A Modular System for Monitoring and Controlling Home Automation Systems", MSc Thesis, Instituto Superior Técnico - Technical University of Lisbon, November 2003.

[12] Renato Nunes, "A Specification and Programming Model for a Home Automation System", IADIS Conference Ibero-American WWW/Internet 2004, Madrid, Spain, October 2004.

[13] Renato Nunes, "A Web-Based Approach to the Specification and Programming of Home Automation Systems", MELECON 2004, Croatia, May 2004.

[14] Renato Nunes, "DomoBus - A New Approach to Home Automation", 8th International Congress on Electrical Engineering, Portugal, July 2003.