

Home Automation - A Step Towards Better Energy Management

Renato Jorge Caleira Nunes

IST – Technical University of Lisbon / INESC-ID

R. Alves Redol, 9, 1000-029 Lisboa, Portugal

Phone: +351 21 3100341, fax: +351 21 3145843, e-mail: Renato.Nunes@inesc-id.pt

Abstract. This paper argues that home automation can make a difference regarding better energy management and usage of renewable energy sources. People are sensible to the need of using energy and other resources more rationally but do very little to that end on their daily lives at home. However, home automation can come to their rescue and we give examples of the functions and the benefits that can be achieved.

The home automation market is very fragmented and many different technologies exist, which are incompatible. To overcome this problem, in this paper we advocate that interoperation should be achieved at a high level and we describe an abstraction layer that allows access to different home automation devices in an uniform and generic way, independently of their technology.

Next we address aspects of system supervision, which are essential to provide the required functions and offer an adequate control and management of the available resources. We propose an approach based on software agents, whose behavior is defined by rules. This offers an expressive means to state the users requirements and preferences.

Key words

Home Automation, Energy Management, Interoperability, Software Agents, Rule-Based Control, XML.

1. Introduction

Almost everyone has heard of renewable energies and is sensible to the need of using energy in a more rational way. However, in practice, people tend to require better comfort in their homes, increasing the use of energy for heating and cooling, for example, and invest little in a more efficient construction, in new energy sources or in means to use energy more rationally.

A dramatic shift in this panorama will require a significant change in mentalities, a change in social and cultural factors, and, in some cases, financial incentives. However, there are several simple measures in terms of home automation that can be applied immediately and offer many benefits: better comfort, increased security,

better management of water, electricity and gas, with significant savings, and improved use of renewable energy equipment that may already exist in the home.

Home automation can offer many benefits and free the users from the burden of having to care if energy is being used rationally, if savings are taking place or if waste is being avoided.

In this paper we start by illustrating some functions that can contribute to better energy and water usage in the home. Next we present three representative home automation technologies and describe their main characteristics. Many other technologies exist, some standard and some proprietary. This diversity constitutes a problem, as these technologies are incompatible with each other. This poses a dilemma to constructors and homeowners regarding which technology to choose, and this market fragmentation is contributing to a slow widespread and evolution of home automation.

Rather than proposing yet another technology (although we also have done that - see [1] and [2]), we advocate that integration and interoperation can be achieved at a high level. We present an approach to define and access home automation devices in a generic and abstract form, which is independent of their underlying technology. This approach favors the integration of different types of devices and allows achievement of interoperability.

Furthermore, we tackle the problem of system supervision, which is an area where current solutions are quite limited. We propose an innovative approach that uses software agents whose behavior is defined by a rule-based system. This approach offers many potentialities and a rich set of features, allowing complex behavior to be expressed and changed with ease, and permitting to adapt the home functions to the user needs and preferences.

Our agent based approach uses the proposed abstraction layer for the home automation devices, allowing it to be applied to different technologies.

2. Benefits of Home Automation

As already referred, home automation can offer many benefits. In this section we present some examples that illustrate how home automation can be used to better manage consumption of energy and other resources. Several measures can be taken in order to reduce consumption, reduce the cost of energy and achieve a better balance between expenditure and comfort.

As a first example, consider home lighting. This can account for a significant part of the electric bill, particularly if common lamps are used instead of high efficiency ones. With home automation it is easy to monitor which lights are on and turn them off in places that became vacant. This can be especially handy in the case of children (which tend to never turn off any light).

But other behaviors, more elaborated, are also possible. For example, light intensity can be adjusted accordingly to the tasks being performed (one needs much less light to see TV than to read), and lighting conditions can take into account intensity levels coming from the outside through windows.

Regarding outside lighting (important, for example, in a detached house), time programming can be used in conjunction with intensity monitoring and presence detection, to provide a good functionality and also increased security.

The measures described can contribute to significant savings while keeping adequate levels of comfort.

Heating and cooling is, typically, one of the areas with more impact in energy consumption. In this domain, home automation can also offer many benefits as it can perform an intelligent control taking into account the required comfort levels, which rooms are occupied and vacant, outside weather conditions and hour of the day.

As an example, let's consider a winter situation and a home that needs to be heated. A common heating system allows the users to define set points for different rooms and little else. With home automation, and an adequate control algorithm, many more possibilities exist. The system can behave differently according to the type of room and hour of the day. For example, in the bedroom the system may assure the required temperature some time before wake time, to allow people to get out of bed in an agreeable environment. After people leave the bedroom, the temperature can be allowed to drop significantly. This situation can be maintained until time to go to bed approaches. During the night, the temperature can be reduced a bit while people are sleeping.

In a child's bedroom the behavior would be different and take into consideration arrival time from school and periods dedicated to playing and studying.

In the case of the living room, the temperature could drop significantly when people are out working and during the night when people are sleeping.

The examples given can be extrapolated to other rooms in the house and enhanced with other factors, allowing adequate levels of comfort and significant savings.

Home automation can also manage other sources of energy, such as solar water heaters, that can be used to pre-heat the water before it enters a gas or oil-fired heater. In terms of cooling, home automation can be used to control air-conditioners in a smart way and also, for example, the orientation of exterior shades or window blinds, avoiding sun heat to enter the house.

Other area where a home automation system can help reduce the energy bill relates to the ability to program the working times of appliances such as washing machines and dishwashers, allowing to take advantage of electricity lower rates during "off-peak" hours.

Another interesting application regards the possibility of direct interaction with the utilities, allowing these companies to shed non-priority loads in the client's homes. The utilities can dialogue with the home automation systems to know what loads are on and what consumption is expected in the near future. Based on that information utilities can manage demand in real time and, if the need arises, request to turn off some loads to reduce consumption at peak hours. Users that adhere to these programs pay energy at lower rates.

Home automation can also help reduce consumption of water. This is very important because residential consumption is the second biggest, after agriculture, and has a low efficiency (58% in Portugal [3]). Consumption reduction can be achieved exerting some control over toilet flushes and showers. Water pressure can be reduced and, ideally, water from showers that is lightly dirt can be stored and re-used in toilets. A simpler area where home automation can give good results is garden irrigation. The system can consider atmospheric conditions such as temperature, amount of rain and soil humidity, to better control watering periods.

Besides helping reducing energy and other resources, home automation can play an important role in auditing the energy efficiency of a home. It can be used, for example, to monitor temperatures inside and outside the house, and monitor water and energy consumption. This can be very important to assess the performance of passive and active mechanisms put in place and quantify the actual savings.

In this section our main aim was to briefly point out several areas of intervention where home automation can offer benefits regarding a more rational use of energy and other resources. It was also our intention to show that, for most examples, best results require an elaborate behavior in terms of system control and supervision. Current solutions do not offer the needed flexibility to easily program these types of functions and that was our main

motivation for developing the work presented in this paper.

3. Home Automation Technologies

The evolution of home automation has been difficult. After an initial period dominated by proprietary products, some standardization movements were initiated. The objective was to allow compatibility between products of different companies and promote cost reductions, increase product diversity and foster a more rapid market expansion. At the end of the nineties several international standards were available, namely CEBus - *Consumer Electronics Bus* (ANSI/EIA 600) [4] and LonWorks (ANSI/EIA 709.1-A) [5]. Some companies also took the initiative of forming associations, freely accessible, and disclosed the full specifications of their products to its members (for example, EIB [6]). However, the anticipated results did not materialize. The existence of various standards divided and confused the market, and new products and technologies kept appearing.

While the standardization movements did not succeed, up to now, in conveying global uniformity and compatibility at the low level, work can be done to achieve interoperability at a high level, enabling different products and systems to interact and cooperate, attaining the benefits of integrated systems.

Next we present a brief overview of three representative home automation technologies - X10 [7], CEBus [4] and EIB [6] - and compare their main characteristics. A more complete description can be found in [8].

A. X10

This technology appeared in the seventies and is one of the most successful commercially, particularly in the USA. It has a small cost and its popularity has been increasing also in Europe and in the rest of the world. Although it is a proprietary technology, its patent has expired and has become a *de facto* standard.

X10 uses the electric power lines as a communication medium. This is a key point of this technology and a major advantage when compared with other solutions because it can be easily installed in already built homes without the need to lay additional wires. X10 modules can be connected directly to power sockets to turn on/off appliances or increase/decrease the intensity of lights.

The main advantages of X10 are its simplicity, ease of installation and usage, and a reduced cost. However, it has several disadvantages, namely a very small communication bandwidth, low reliability, and a limited number of functions that are restricted, basically, to lighting and on/off control of appliances.

It is possible to have very simple X10 systems where a small number of devices are controlled directly by the user using, for example, a remote control. More complex systems can be built using specific X10 controllers, which can communicate with a PC and accept time

programs to command devices at certain moments and activate sets of actions when certain conditions are met.

B. CEBus

The CEBus - *Consumer Electronics Bus* - offers a complex and powerful communication protocol. Its roots date from 1984, having evolved up to its standardization in 1995 (ANSI/EIA-600). The CEBus protocol follows the OSI model and implements the physical, data link, network and application layers.

The CEBus specification is very ambitious and supports many physical communication media: power line, twisted pair, coaxial cable, optical fiber, infrared and radio frequency. In some of these media, data communication can coexist with audio and video signals. A great effort was made to assure a noise immune and reliable exchange of data, independently of the communication media used. Interconnection of different communication media is achieved through specific devices called *routers*, *brouters* e *data-bridges*.

CEBus offers a peer-to-peer communication model where a device can interact directly with any other device. This allows distributed control mechanisms to take place and supports *plug-and-play*, which permits to add new devices to a system at any time.

To support high level interaction between CEBus devices, a specific language was proposed: CAL - *Common Application Language*. This language uses the notion of *context* and assumes that a device can be described as a collection of contexts. Currently, more than 50 contexts are defined that contemplate areas such as lighting, security, heating and appliance control. A given context behaves always in the same way independently of the type of product it is inserted into. For example, the "audio amplifier" context has exactly the same behavior whether it is associated with a TV or a CD player.

A context possesses several control objects. There are 26 different objects that represent, for example, digital sensors, analog sensors, switches, motors, keypads, etc. Each object is described by a fixed set of variables. The CAL language allows inquiring and modifying the values of those variables, which affects the behavior of the corresponding devices.

From this brief description one can see that CEBus is far superior to X10. However, its products are not easily available and have a higher cost. This is making difficult for this technology to become widespread and, currently, it is expressive only in the USA. CEBus is a sound technology that offers many services, like plug-and-play, authentication and secure communication, but we will have to wait and see what will be its market acceptance.

C. EIB

EIB - *European Installation Bus* - was born as a proprietary technology but their full specifications are

accessible to any company that becomes a member of EIBA - European Installation Bus Association [6]. This association has a great number of affiliates that produce and sell a vast range of devices compatible with each other.

EIB supports several communication media, such as twisted pair, power line and radio frequency. The most common media is twisted pair, which can be used both for communication and device powering.

Each EIB device has a physical address and may have multiple group addresses. When a message is sent to a group address it is received by all devices belonging to that group. This is an important feature of EIB and is used to manage interaction between different devices in a system without a central controller. For example, if we want a switch to control two lights, we assign both lights a group address and configure the switch to send a message to that group whenever it is activated.

EIB uses a specific tool - ETS (EIB Tool Software) - for system design, test and commissioning. With this tool it is possible to describe the physical structure of a house or building, identify the devices to use, and execute their configuration (assignment of addresses - physical and group addresses - and define how devices interact with each other). Although effective, this interaction mechanism between different devices is less flexible and less powerful than CEBus's CAL language.

The main disadvantages of EIB are its high cost and the need to use the ETS tool to design and implement a system. Its installation and commissioning is also complex and requires intervention of expert personnel. This technology is quite common in Europe, but is applied mainly to building automation.

D. Comparative synthesis

To conclude this section we present a brief comparison of important characteristics of the technologies just described - see table I.

TABLE I. - Comparison of technologies

Feature	X10	CEBus	EIB
Transmission rate (bps)	~ 50	~ 10 000	9 600
Typical frame duration	940 ms	~ 25 ms	20 a 40 ms
Maximum number of devices	256 (2 ⁸)	~ 4 G (2 ³²)	~ 64 K (2 ¹⁶)
Cost	Low / Medium	High	Medium / High
Difficulty of installation	Low	Medium	High
Functionality	Low	High	High

EIB technology is complete and powerful. At the data communication level it is very robust and comparable to CEBus. Both technologies are far superior to X10. The same happens regarding data communication throughput and services offered.

Considering the number of devices supported, EIB is very superior to X10, although inferior to CEBus. However, EIB supports very big systems and cannot be considered limited at this level.

Regarding cost and ease of installation, X10 is the technology that offers more advantages and these aspects seem to be responsible for its popularity, despite being technically quite limited.

In this section we described very briefly three technologies. However many more exist, such as, LonWorks, EHS, BatiBUS, Konnex, HomeRF, HomePlug, HomePNA. As the market is very fragmented we advocate that people should be able to use whatever technology they like best (as long as technically adequate to the purpose) and that interoperation should be made possible at a high level. In the next section we present a proposal to tackle this problem.

4. Device Abstraction Layer

In order to achieve interoperability between different technologies we propose a software-based solution that offers an abstract model for home automation devices.

A. Overview

Our approach allows encapsulation of each device's particularities under a common API (Application Programming Interface) and uses a XML based language to describe each device's capabilities in a generic and abstract form.

In figure 1 we illustrate the architecture proposed for our approach.

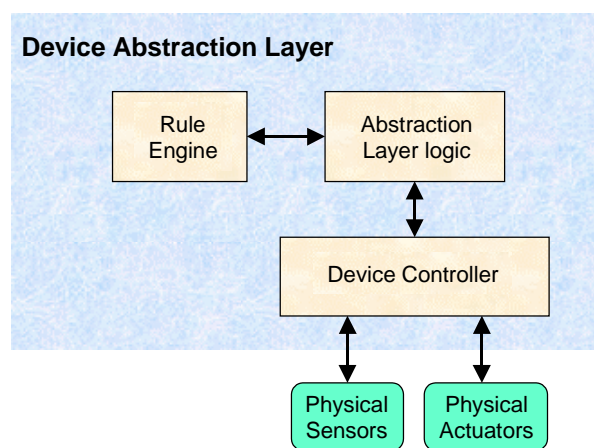


Fig. 1. Architecture of the Device Abstraction Layer

We consider two basic types of devices: *sensors* and *actuators*. Sensors are responsible for measuring physical entities (e.g., temperature of a room). The actuators can directly or indirectly affect or control the value of a physical entity. For example, a heater can affect the temperature of a room.

To seamlessly integrate a device on the Device Abstract Layer (DAL), each device must have associated a

representation of its capabilities, measured entities, pre-requisites for each functionality provided and events or triggers supported. We propose a representation based on XML which provides a flexible and architecture independent solution.

Once we have a representation of the constituents of a home automation system, we may specify how these components interact with each other and with their surrounding environment. To that end, we propose the usage of an inference engine.

B. Abstraction Layer logic

The Abstraction Layer logic defines an interface that exposes methods that allow an external application to monitor and control home automation devices (sensors and actuators). The exposed methods are:

- *getDevices* - gets a description of the available devices in the system.
- *addRules* - allows the definition of triggers or events for each device.
- *rmRules* - removes a defined trigger or event.
- *executeCommand* - executes a command on a device.
- *getDeviceStatus* - gets the status of a device.
- *initializeDAL* - loads and initializes all device's instances and driver code.

With these methods, it is possible for an external application to obtain a description of the home automation system deployed and to request the execution of a command on a device. It is also possible to specify which events the application is interested in and to construct triggers based on the capabilities of a device.

C. Device Controller

The Device Controller module is responsible for managing each device's instances and life cycle, and is able to load and unload each device's specific driver code from a dynamic link library. It is up to this driver code to deal with the specificity of each device - technology, communication protocol and functions offered.

The common representation of a home automation device is specified by an abstract class that defines the interface that all device's must implement on the respective dynamic link library in order to be transparent to the Abstraction Layer logic.

The Device Controller handles the interactions with each device, exchanging events and messages between the home automation devices and the Abstraction Layer logic.

D. Rule Engine

The Rule Engine is an inference engine used to define the events and triggers that an external application is

interested in receiving from the home automation devices (sensors and actuators).

Each time there is a change in the state or in a property of a device, that information is fed to the Rule Engine which may in turn trigger an event to an external application or initiate an action upon another device.

The Rule Engine evaluates the *facts* produced by the devices accordingly to its knowledge base and executes whatever actions result from that evaluation.

E. Device description language

We defined a language to express the characteristics and capabilities of a device. This language is based on XML [9]. It is out of the scope of this paper to detail our specification. So, we limit ourselves to describing its main features:

- Device description - contains information such as the device name, type and dynamic link library location.
- Monitored entities - defines the physical entities that a device can act upon; those entities are described by a name and a collection of values (each characterized by a type - integer, float, etc - and a value range - from *min.* to *max.*).
- Supported functions - indicates the functions that a device's driver code implements; for each function is specified a name, an argument list, a return value, physical entities affected (if any), pre-conditions and post-conditions expressed in terms of facts and rules that the Ruler Engine understands.
- Provided events - lists the events that a device raises when certain conditions are met or when there is a change in the device's state.

The proposed language is very expressive and generic, and accomplishes our goal allowing an abstract representation of any home automation device, from a simple electric switch up to, for example, a surveillance camera with motion detection.

5. Agent-Based Supervision

In the previous section we described a Device Abstraction Layer that offers a model to generically access any home automation device. Our approach allows a uniform access to different devices, independently of their technology, and promotes interoperation at a high level. Although this is already very valuable, we gave a step further and addressed a new approach to system supervision. Our aim was the testing of our abstraction model and offering of new forms of control in order to tackle the type of functions and behavior we described earlier in this paper.

The proposed approach is conceptually characterized by a set of sensors, a set of actuators and a processing module. The system acquires information about the world

through sensors. It processes the information collected and reasons about the course of action to follow, activating the appropriate devices (actuators) to achieve the desired goals. The plan can take the form of an immediate action over some objects or devices, or it can take a more elaborate form requiring intervention of several system components.

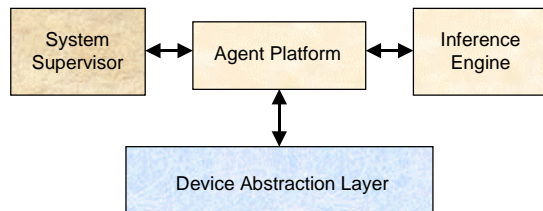


Fig. 2. System architecture

Figure 2 illustrates the system architecture. The Device Abstraction Layer was described in the previous section and virtualizes the home automation physical devices. The Agent Platform supports the execution of agents. It interacts with the Inference Engine that offers support to system reasoning. The System Supervisor allows the user to interact and operate the system.

A. Agents

We identified four types of agents: device agents, generic agents, coordination agents and directory facilitator agents. Device agents are associated with devices (sensors or actuators) and mediate any action regarding them. They access the physical devices through the Device Abstraction Layer. They can also simulate a physical device, which can be very valuable for testing or evaluation purposes.

Generic agents do not need to be associated with a device. They can receive events and can react (accordingly to existing rules) sending information to other agents.

Coordination agents receive and process relevant information coming from other agents. They are responsible for system coordination and inform other agents of obligations and restrictions arisen from the detected situations.

The directory facilitator agent helps agents to contact another agents in order to get assistance for carrying some tasks.

B. Events

Events are triggering mechanisms that represent changes in objects. Whenever something changes in the world there is an event reporting that change. There are three types of events: basic events, rule events and device events. A basic event is originated by a change in the value of an entity. A rule event is created when a rule condition is evaluated to true. A device event can be any event that a device wishes to create.

A situation, representing a partial state of the world, can be constructed by using a set of events. The system detects a situation when all the events describing that situation are activated.

C. Rules

The system's behavior is described by rules. Rules specify what to pay attention to and how the system should react.

A rule has conditions and actions. If the conditions are evaluated to true the actions are performed. A rule has the following form:

IF <conditions> THEN <actions>

Please note that despite the reactive appearance of the rule's form, it is far from being reactive when we analyze the semantics of conditions and actions.

D. Conditions

A condition must be able to reference objects, entities and entity's values. To that end we have predicates such as:

(object (name ?objectName))
(entity (name ?entityName))
(entityValue ?entity ?object ?value)

The last predicate indicates that the entity referenced by *?entity* in the object referenced by *?object* has the value *?value*.

As an example of usage consider the two facts and the predicate just below, which specify that the object *Kitchen* has a temperature of 18 degrees.

?temp ← (entity (name Temperature))
?room ← (object (name Kitchen))
(entityValue ?temp ?room 18)

E. Actions

Regarding actions, we consider three types: pre-defined, device specific and high level. Pre-defined actions are actions like the following:

(action ?direction ?entityName ?objectName ?location)

This indicates an action over an entity *?entityName* of the object *?objectName* in location *?location*. *?direction* can be *lower* or *augment*.

Device specific actions correspond to inherent capabilities of a device. They are expressed as:

(<actionName> <arg1> <arg2> ... <argN>)

Each device is described as a set of functions. A function has effects (actions) over entities and objects. So, *<actionName>* is the name of the effect (action) of the function describing that specific action.

High level actions can be considered internal system actions. They do not necessarily correlate directly to any observable external action.

Using rules it is possible to specify how the system will react to events and situations, and allow the establishment of priorities for each situation.

6. Conclusion

In this paper we proposed the use of home automation as an effective means to reduce consumption of energy and other resources. To that end we suggested several functions and procedures, and illustrated how home automation can help balance comfort and energy usage, allowing for significant savings.

Next we described and compared the main characteristics of three representative home automation technologies. Many more technologies exist and we emphasized how this market fragmentation is affecting home automation spreading.

To overcome this problem we proposed and described a Device Abstraction Layer that allows access to different home automation devices in a uniform and generic way, independently of their underlying technology. This allows a high level integration and interoperation of different technologies, which can offer many advantages.

Finally we tackled the problem of system supervision and proposed an agent-based solution. This approach offers many potentialities and a rich set of features, allowing the usage of a rule-based system to define the system's behavior.

A prototype of the proposed approach was successfully implemented and future work should be carried on to evaluate the expressiveness of the solution and its suitability to address big and complex situations. Future work should also be done to assess maintainability and how easy the system's behavior can be modified.

References

- [1] Renato Nunes, José Delgado, "An Architecture for a Home Automation System", ICECS '98 - International Conference on Electronics, Circuits and Systems, September 1998, pp.259-262.
- [2] Renato Nunes, "An Easily Configurable and Programmable Home Automation System", SCI 2002 - 6th World Multiconference on Systemics, Cybernetics and Informatics, July 2002, pp.266-271.
- [3] Instituto da Água, "Plano Nacional para o Uso Eficiente da Água", Lisboa, Setembro de 2001, <http://www.ambiente.gov.pt/docs/pnuea.pdf>
- [4] CEBus - Consumer Electronics Bus, <http://www.cebus.org>
- [5] LonWorks (ANSI/EIA 709.1-A), <http://www.echelon.com>
- [6] EIB - European Installation Bus, EIB Association, <http://www.eiba.com>
- [7] X10, <http://www.x10.com>
- [8] Renato Nunes, "Análise Comparativa de Tecnologias para Domótica", JEACI 2002- III Jornadas de Engenharia de Automação, Controlo e Instrumentação, EST Setúbal, Maio 2002.
- [9] XML - Extended Markup Language, <http://www.w3.org>
- [10] Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence, MIT Press.