Configuração de Sistemas Domóticos

João Pereira Nunes

Instituto Superior Técnico
Mestrado em Engenharia Informática e de Computadores
joao.p.nunes@tagus.ist.utl.pt

1. Introduction

Home automation is a technology that allows management of different house resources. This automation simplifies the day life of common people satisfying their needs for comfort, security and communication.

When home automation appeared, in the 80's, the principal areas of intervention were light control, ambient control and security.

Home automation systems offer a wide number of advantages and they are increasing in popularity. Technologies like internet and mobile devices have contributed for this popularity.

Home automation also provides energy saving capabilities. It is possible to add scheduling policies to some equipment to use them when the electricity is cheaper, turn lights and air conditioning off when divisions are empty and also to manage water consumption. It is also possible to take advantage of renewable energies produced at home like solar and wind energy.

However, house automation had a hard evolution. It is common to find proprietary products that cannot connect to products of different brands. Efforts were made to standardize technologies to allow communication between products of different manufactures achieving lower prices and increasing diversity to final customer. Several standards were developed like EIB/KNX, CEBus and LonWorks. Currently the most used standard in Europe is KNX, which is the result of joining three previous technologies. The configuration of EIB/KNX is made using the ETS tool.

In general home automation technologies don't provide configuration tools but when they do, they are closed, associated with a specific technology and made for skilled professionals. These tools are expensive, complex to us and only work with one specific technology.

Due to these problems, in this work we propose a home automation system configuration tool aimed to be generic and to work with different home automation technologies.

The proposed tool will allow users to create and modify the configuration of home automation systems. It will also allow changing home's behavior to adjust the house to new user's needs and preferences.

In this tool we will use a generic and flexible model called DomoBus, to create the system configuration and system parameterization. The tool will allow users to describe and modify any house and home automation system. All the information will be stored using XML.

2. Home automation technologies

To learn and understand home automation technologies features we studied various home automation technologies.

2.1. X10

The first was X10. This is a home automation technology that uses power line as its communication infrastructure. It was invented in 1974, it is still used nowadays and it is one of the most used home automation systems. Each device in X10 has its own address composed by a letter from "A" to "P" and a number from 1 to 16, allowing up to 256 devices. To issue a command one message is sent with an address that selects a device, followed by a second message that specifies the action to be executed.

The usage of the power line for communication brings some problems into this technology. Some equipment can introduce electrical noise that makes the communication unreliable. To minimize this problem it is necessary to install filters. Another problem is the fact that communication signals can reach other houses in the vicinity. To solve this, it is also necessary to install filters in the power's connection to the outside. There are also problems with three-phase system. The X10 technology offers a limited number of commands, which makes the technology very simple and there are computer programs to control a house.

An advantage of X10 technology is that there is no need to install a new communication bus as happens with other technologies, reducing installation costs. It is as simple to install an X10 infrastructure in a new house as in an old house. To configure an X10 system it is necessary to define the address of each device. Although simple it is a slow and error prone procedure in a house with many devices.

2.2. EIB/KNX

This technology was developed by a group of leader companies in electrical products in the European market with the objective to create an alternative system to systems produced in Japanese and USA markets. The first aim was to create a European standard that allows device communication. EIB has a decentralized architecture allowing distributed processing between sensors and actuators. After some years KNX appeared as a new European standard mixing EIB with Batibus and EHS to provide a better service to the end user with new functionalities a better quality. KNX uses twisted pair, power line, Ethernet and some more communication media but the most used is twisted pair. In this technology there are three ways of configuration. S-mode configuration is the most used and it is done using a configuration tool named ETS. In this mode everything can be parameterized in the tool. E-mode allows users to configure just small things in the device as they came preconfigured. A-mode is a Plug&Play mode allowing the user to install the device and it will be ready to work. This technology has many good characteristics but also some problems. The configuration of a system is not as simple as they try to show. Almost all devices only support S-mode configuration which means that a normal user need to hire a skilled professional to install the devices, since the ETS tool is expensive and not accessible to a common user.

2.3. LonWorks

The LonWorks was developed by Echelon. LonWorks is based in a "LON" network - Local Operating Network. In this technology every device, also know as node, has its own "neuron" chip, a transceiver, power source and input and output hardware. Devices can communicate using the "LonTalk" protocol. Each device's function is kept inside the neuron chip and executed there. The running code is written in a language called "Neuron C", which is an adaptation of the C language. LonWorks uses the concept of "network variable" for communication. A network variable can be an output in one device and input in another one. A device will export values as output network variables and the other will import them as input network variables up to a

maximum of 62 network variables. If one of these network variables is standard it is called SNVT - standard network variable type. This technology includes some more concepts such as Standard Configuration Property Type, Functional Profiles and User defined Configuration Property Types. This makes the technology complex and difficult to use by a common user. This technology supports optical fiber, power line and twisted pair as communication media. The configuration of devices is made using a specific tool. In general LonWorks is a complex technology and hard to find documentation. If a user wants to install a LonWorks system it is required to hire certified professionals to install it.

2.4. DomoBus

This technology was developed in an academic environment. It has two levels: the command and monitoring level and the management and supervision level.

The command and monitoring level consists of physical devices (sensors and actuators) and their control modules. DomoBus has a distributed architecture where control modules are interconnect through a network. The network may include various segments interconnect through router modules. Each control module may control one or more devices (sensors and actuators). Control modules are managed by one or more supervisor modules (which may be common PCs).

The management and supervision level is the top level. Here it will be described the house and the system configuration in a generic way. DomoBus configuration is specified in XML and can be changed by common users using the configuration tool DomoConfig.

In this level there is an abstract model that is used to represent any device. A device is defined by a group of properties, each one with a value. Each property has its own data type that can be an enumerated, scalar or array. Device manufactures are responsible for the creation of all this information about devices and aggregate it in a XML description. With this device description a user or installer can add newly acquired devices to his home. System behavior is also described in this level where a user has to configure which sensors and actuators will communicate. A house in DomoBus is a simple entity with floors, where each floor has divisions.

The house supervision will be done using supervision modules which can be PCs or SBCs (single board computers). They will receive information from the control modules and will process it according to some preprogrammed rules that will generate actions over the home's actuators. A system can have more than one supervision module and they can communicate with each other to share information and coordinate actions.

DomoBus allows connecting to other home automation technologies using specific gateway applications running in the supervision modules. The gateways are responsible to translate commands from the DomoBus logic to the other technology logic.

DomoBus is a simple and flexible solution because of its generic model allowing working with almost every technology. It also allows the user to change interactively the system behavior and adapt it to their new needs and preferences. DomoBus is also economic because control modules can control more than one device saving energy and resources.

3. Configuration Tools

3.1. ETS

ETS is the KNX configuration tool developed to design and configure home automation systems based on EIB/KNX technology. This tool is aimed to configure devices in S-mode. The ETS tool needs a product database to know all the information about devices. This database is provided by device manufacturers and normally it comes with devices. To create these databases, manufactures need another tool - the manufacturer tool. These databases are also private and can't be read by other tools.

-There are three versions of the ETS tool: ETS Tester aimed for testing and learning before buying the tool, which is free but very limited; ETS Starter aimed at common users allowing the creation of small projects, using a specific device database; and ETS Professional without any limits.

ETS users need to configure the entire house, group addresses, communication "lines" and then they send this configuration to the system. The configuration is a mix of program code with parameters configured by the user, which is stored into each device. If a user wants to change just one parameter it is necessary to send all the program code with all the parameters to the device. Products database installation and device insertion is a very slow process for an effective utilization. ETS was made to work only with KNX technology and it is an expensive tool for common users.

3.2. LonMaker

Another studied tool was LonMaker. This tool was made by Echelon to design and configure LonWorks home automation networks. LonMaker uses Microsoft Visio as its main design component. There are two LonMaker versions: Standard Turbo Edition and Professional Turbo Edition. The biggest difference between the two versions is the version of Visio included, as the standard version has less functionality. LonMaker is a paid tool with a credit system. Every device to be configured needs a credit and every credit has a cost. To reconfigure a device it is not needed to spend more credits.

LonMaker tool is based in LNS – Local Network operating System - a service based software with a database made by Echelon to be used by tools manufacturers to help them with PC applications development. LNS allow the creation of other applications like HMI (human machine interface) or GUI (graphical user interface) as LonMaker.

There is a LNS server that offers installation mechanisms, network creation, maintenance, control and monitoring services and also a recover database. It is also possible to have collaborative work using this server.

LonMaker was designed to manage all LonWorks network life cycle since the design up to the operational phase. LonMaker offers these features: network design, where users will draw all the network connections with devices and device behaviors: network installation, where configurations are sent to devices; network documentation and network maintenance. To configure a LonWorks network it is needed a LNS server which will include a database where all device configuration is kept. It is always needed a certified professional, if a user wants to add a new device and it will involve costs with credits and hiring professionals. There are alternate tools to configure LonWorks networks but also with the same credit system. LonMaker was made to configure only LonWorks networks and it is an expensive tool for common users.

4. Solution architecture

After having studied some home automation technologies and their configuration tools, we proposed a generic solution to design and configure home automation systems based on the DomoBus model, due to its flexibility and generic features.

In the DomoBus model a device is represented by a collection of properties with standard operations (get, set, notify) to access them. With this model a generic and flexible tool can be developed to allow the configuration and definition of behavior for home automation systems, independent of any specific technology.

In the DomoBus approach a home automation system is made of home automation modules, each one with one or more devices. Every module comes with a XML description that will be imported by the configuration tool. After that it will be possible to configure devices and save the configuration allowing future editing.

The proposed tool has the architecture shown in the following picture.

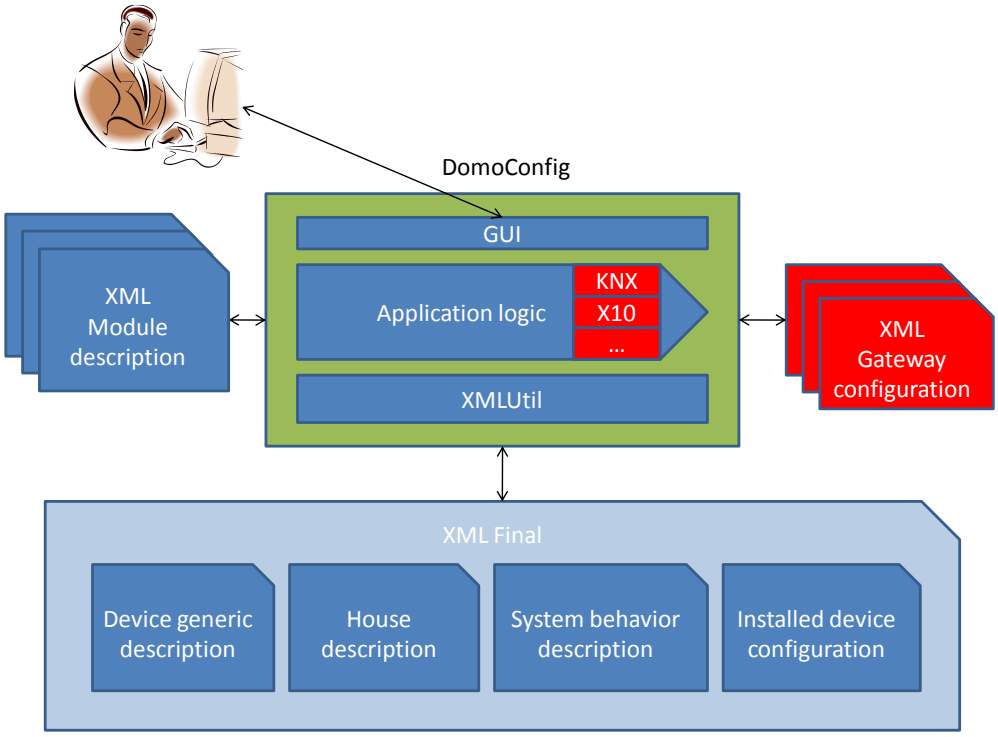

Fig 1. DomoConfig application architecture

The DomoConfig application is represented in the middle and it includes a GUI, the application logic and the XMLUtil components.

The GUI component implements the graphical user interface allowing the user to interact with the application. It includes also some features related to house definition, behavior configuration available and some more functions for device manufacturers.

The application logic component has all data structures and application rules. The function of this component is to manage everything related to actions accessed by the GUI. This component also supports the installation of plug-ins for gateway configuration.

The XMLUtil component is responsible to save and load all the information that is used in the application in XML files. It will save all the data structures that the application logic component has to manage.

Around the application there are several input and output XML files. The application will work on these files that we describe next.

4.1. Module description

The module description file includes all the information about devices present in a module and it is created in the application by the device manufacturer. This file normally comes with newly acquired modules and includes a device list. Each device in the list implements a generic device type. Each device type has a list of properties and a class. This class is used for device categorization. Properties have a data type, permissions information (read/write) and a configuration type associated. This configuration type includes a set of items and each one indicates the maximum and minimum values possible for certain notifications supported by the property. A property data type can be enumerated, scalar or an array. Sometimes the value of a property has to be converted to be read by applications and it can include a data converter for each data type. Data converters can be simple formulas or a class DLL with complex code.

4.2. Final XML

The second file is the final XML that will keep the entire house configuration and it has four subcategories. In the device description it is included all the generic data types and, normally, they came with the modules description. They are: device types with their properties, data types and their converters, configuration types and device classes. With this generic data types it is ensured data reutilization. The house description has information about house's floors and divisions. Each house has a floor list and each floor has a division list. There is also the possibility to include logical divisions, like a garden. It is also included a user list, access levels and a list of house's services. In the system behavior description it is included the interactions between devices in the form of scenarios. Each scenario has a condition, an action list and a deactivation list. In the condition it is possible to put recursive conditions using logic operators AND and OR. Each test in the condition is a comparison between a device property value and a user predefined value. The action list is a list with new values to assign to device's properties. The action list is executed when the condition becomes true and the deactivation list is executed when the condition is false. There is also a macro feature that only has an action list and it is used to quickly trigger a set of actions in a house. The installed device configuration keeps all relevant data about device configuration, parameters and localization. To install a device it is first needed to import a module description; then a user must choose a division to put one of the available devices. Installed devices can then be categorized into services. Devices have an access level and can be blocked by users. An installed device has to make reference to its device type to know their base properties and configuration options. These configurations allow a device to make notifications when some of their properties values had reached a certain value. The user has to choose which values will trigger notifications and the type of notification. These values have to comply with minimum and maximum values that are defined in properties of the configuration type. There are six different notification types: the first one is a maximum limit value. The second is a minimum limit value. The third is periodical notification sending. The fourth is a variation notification, when a property varies a certain value the device will make a notification. The fifth is an actuator state change notification, when an actuator changes its state it can notify instantly or it can wait Δt time to notify. The sixth type consists of specific conditions, up to a certain number. These conditions are defined by the user and use logic operators and comparison values.

4.3. Gateway configuration

The third file is the gateway configuration. This file contains all the gateway parameters. A gateway will allow the DomoBus system to interact with other technologies. The gateway will convert all addresses and commands from one technology to another. This gateway can be configured in the application using a specific technology plug-in. This plug-in also include specific data types for device properties to install in the system configuration. To configure a gateway a plug-in is always needed. Gateways have two fundamental configurations, one for address mapping and another for command mapping. In the address mapping there is a one to one map and a group map. In the group mapping there is a virtual address with a list of addresses that makes the group. In the command mapping it is needed to use plug-in data types previously installed. To make the mapping plug-in will look at this data types and will make the correct command conversion. The gateway with this configuration will manage the conversion from one technology to another.

4.4. Development environment

The application was developed in C# language using Visual Studio 2008 SP1 with some useful technologies like linq and WPF. This development ambient was chosen to simplify interface creation and ease the processing of XML files. This technology does not preclude the use of other operating systems. It is possible to use .Net in Unix systems using Mono technology (however it is not yet available for the versions we used).

4.5. Application interface

The application interface has two strands: one for the normal user or the installer and another one for module's manufacturers. In the first one there are options to create houses, install and configure devices and create house behavior. If the user needs a gateway there is also an option to configure the gateway. In the second one there are options to create data types and their conversions, create device types and configuration types and module management.

4.6. Middle layer

The application was build using a three layer architecture. In the middle layer there is all the application logic. This component is divided in two parts: a DLL with all data structures that is shared with plug-ins and another with application logic. This data structures hold all the information present in the XML files or they are empty in the case of a new configuration. There is an aggregator class to keep all data structures together. In this class there are lists to all data structures. Every list keeps different entities of information. Each entity can make reference to another entity like house and their floors. To keep the address configuration under control there is an address manager that will ensure that there are no repeated addresses. The global class and all data structures are included in the "TiposComuns" project and they are compiled as a DLL. To create a plug-in for gateway configuration it is needed to implement an interface available in this DLL. In this interface it is needed to implement four methods and to fill three strings. The methods are used to generate addresses, properties and to install new data types.

The application will manage all of this information and user interaction to make a correct configuration and save it as a XML file.

4.7. Data layer

To save and load XML files there is a software module called XMLUtil that was made to interpret all XML data and to fill the application data structures. Module and gateway files are also read with this module. This module is also responsible to convert all data structures into XML files using the inverted process. In this software module linq technology is widely used making it easy and fast to work with XML files

4.8. Code organization

The code is organized in two big projects inside the solution: “DomoConfig” and “TiposComuns”. These two projects are the application core. To create a new plug-in it is necessary to create a new project. There is also a secondary project to create the application installation program.

The DomoConfig project includes all the windows, images and icons and the “XMLUtil“ module.

In “TiposComuns” project, a DLL library, there are all data classes in the folder “Dados”, exceptions in “Excepções” folder, a global class and a plug-in interface.

In a plug-in project there is only one class that implements “IPlugin” interface from “TiposComuns” project. Plug-ins projects and DomoConfig project both has to make a reference to “TiposComuns” project since they need their classes.

5. Work evaluation

The work was evaluated using usability tests to check its correct execution. It was used use cases and tasks for users. We calculated the number of errors, the number of steps and the total time that a user needed to finish every task. We also made a comparison with other tools in terms of price.

In these tests users had to configure a house and it was verified how easy/difficult that was. Also it was generated a home automation module and verified the same. The analysis of the results allowed to conclude that the tool was easy to use but some users had difficulties defining system behavior.

Regarding costs, the tool is much more economic than the other tools available because it allows the user to work with more than one technology and it is completely free.

6. Conclusions

This work presented a proposal of a generic and flexible tool for the definition and configuration of home automation systems, independent of its technology, using the DomoBus model.

Home automation systems have increased in popularity but there are many technologies available with specific tools to configure the systems. Another problem is that users cannot, or it is hard, to change the configuration or behavior of a system. Just a small change can imply contacting skilled professionals and incur in significant costs.

To overcome these problems several home automation technologies were studied and their configuration tools where analyzed. We concluded that these tools had some problems and limitations and we developed a new tool, based on the DomoBus model, which allows interacting with different technologies and simplifies making small changes in the configuration and behavior of a system. The tool developed is easy to use by a common user and avoids the need to contract technical personnel.

Another objective was to reduce drastically the associated costs with this kind of tools and to allow intercommunication with other technologies. The DomoBus model allows the creation of a generic configuration

for every technology saving the information in XML format. This XML specification had to be extended to comply with the new requirements and the new features.

The developed tool allows overcoming almost all identified problems in other technologies. The interaction with other technologies uses a plug-in and a gateway. We developed an X10 plug-in for testing purposes. In the future this plug-in can help implement plug-ins for other technologies.

Tests were made to ensure that the application was correctly made and easy to use. With these tests we verified that the tool was easy to use and users succeed in performing common tasks.

With the development of this tool we expect to improve the definition and configuration of home automation systems offering better services and making it easier to use by common users.

6.1. Future work

The tool was easy to use because we followed user's suggestions. But in one test we verified that defining behavior for a system was not so easy to do. As future work it is suggested to improve the definition of system behavior introducing a simpler and more advanced mode to perform that task.

In other configuration tools like ETS and LonMaker it is possible to upload the system configuration to the devices on the bus. In a next version DomoConfig should also offer this possibility although this was not a requirement for this work.