A Domotic System with Remote Access based on Web Services

Mauricio Esteban Pardo
mepardo10@hotmail.com
and
Guillermo Enrique Strack
guillotest@gmail.com
and
Diego C. Martínez
dcm@cs.uns.edu.ar
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur,
Avenida Alem 1253 Bahia Blanca, Buenos Aires, Argentina

ABSTRACT

Domotics systems are intelligent systems for houses and apartments to control several issues as security and light or climate devices. In this work we present the development of an economic domotic system to control different electrical devices in a private house. This is achieved either from inside the building or by remote control using a regular Internet connection. In order to provide this functionality, the system includes a server that provides web services to the controlling applications in the house. This server also offers an interface that uses AJAX to optimize device handling.

Key Words: home automation, domotic systems, web services, AJAX.

1. INTRODUCTION

Domotics systems are intelligent systems for houses and apartments which integrate and control devices or services, such as communications, security, lighting, climate control and household devices. These systems provide benefits as comfort, security and optimization of energy consumption [3,4,5]. Currently there are several companies engaged in the implementation of these systems, providing specific commercial solutions.

The overall structure of the domotic system varies according to manufacturers and the specific domain. In general, the main components are identified as *controllers* (to operate the system), *sensors* (to perceive the environment) and *actuators* (to modify the environment). Usually, these are different kind of devices, but sometimes a single device has the necessary intelligence to measure a physical variable, to process it and act accordingly. However, most solutions distinguish sensors from actuators in order to provide more flexibility and lower price. This is a very important issue, especially when the system is designed for family houses, where the budget is limited.

A Web service [1,6] is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-understandable format (specifically WSDL). Other systems interact with the Web using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. Thus, a Web Service allows the execution of a remote method and the reception of its result as if it were a local method. It makes possible the communication between, for instance, an ASP .NET system executing on IIS with another system developed on Java and executing on Solaris server.

Ajax is an acronym of *Asynchronous JavaScript And XML* [9], and it is a simple programming technique to create interactive web applications. Part of the application is executed on the client side (the user’s browser), and it maintains in background an asynchronous communication with the server. It is possible, for example, to make visual changes on the same page without reloading or sending explicit server requests. This leads to an increase of interactivity, speed and usability of the web interface.

This paper presents an implementation of a controller-actuator domotic system, which was developed using Web Services for remote access and AJAX in the web interface. To achieve simplicity in its physical installation, the control of devices is conducted through wireless mode and also trough a wired connection.

This work is organized as follows. The first section provides the scenario and the premises for our proposal. The second section shows the main elements of the system architecture and the interaction between them. In the final subsections we discuss related works and future extensions to the system.

2. DEVELOPMENT SCENARIO

According to the particular requirements, a domotic system can be implemented in several ways. Naturally, its cost is determined by the features and capabilities it provides. In our case, the emphasis is put on this economic factor and thus the implemented system provides a specific feature: it allows inexpensive remote control and scheduling of different electric devices that are commonly located in a family house or a middle-size store. Several premises are important for the proposed architecture:

- The system must allow local control and also remote control of devices [7].
- The system must allow the centralized administration of several independent houses, (probably) by different users.
- The system must use technologies that are easily assimilated by the programming community. It must not require high costs of development and deployment.

Based on the above description, the proposed architecture requires a web server that centralizes control and provides the needed functionality to manage devices with remote access. It also requires a computer at home, connected with the desired controllable devices (lamps, TV, irrigation systems, etc.). For simplicity, the implementation does not require a fixed public IP number; therefore the home-located application will work exclusively as a client of the web server. Thus, the user does not need more than a classic domestic Internet access. In Figure 1 the simplified architecture is depicted. The computer installed in the house (called *Home PC* in Figure 1) provides an appropriate interface to manage every device directly connected to it. This is called the *in-situ controller* of a particular house. The external server has the same functionality, but it is intended to be accessed remotely. This is called the *remote controller* of several homes managed by each user. This implementation needs two databases, one with general information of all users and their houses, which is located in the web server, and the other one with specific local information, which is stored on the user's PC. This dual functionality (both in the house *and* the server), is strongly desired for accessibility. It requires, however, an additional effort to satisfy with certain restrictions, such as the consistency between the server-located database and the home-located database. For that reason it is necessary to develop a synchronization protocol that allows the preservation of both databases in a consistent state, making updates whenever a change occurs from any side. In order to implement that protocol, a simple web service is used. It is important to remark another benefit of this intrinsic duplication: the system allows the control of devices without Internet connection. In this case, the changes are made locally and later when the internet connection is restored, both databases will be correctly updated.

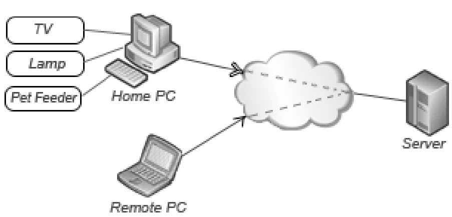

Figure 1. Main components of the architecture

The hardware used for communication between computer at home and the devices is connected to the parallel port. Orders from the port to devices are sent as wired or wireless signals, in order to ease the overall installation. This particular component that interconnects computer and domestic devices was developed and built by the authors of this paper and its structure will be explained later.

In the next section the software functionality is presented, as well as its operation mode and the hardware components.

3. APPLICATIONS COMPONENTS

The domotic system consists of several components. These are the *House Design Interface Software*, the *Device-Control Software* (for local deployment) and the *Device-Control Web Application* (for remote access).

3.1 House Design Interface Software

The human-computer interface is very intuitive and simple to use. This application provides the necessary tools for the creation of a graphical interface simulating the user's home. The designer can select the images representing rooms that compose the home, each one marked by a name. These room representations can be graphically arranged in a way that emulates the real structure of the house. In every room there is also a graphical representation of different devices to be controlled in that room. For the sake of extensibility this application gives the client the possibility to add more rooms later, as well as the ability to edit or delete rooms. The owner must define a username and a password,

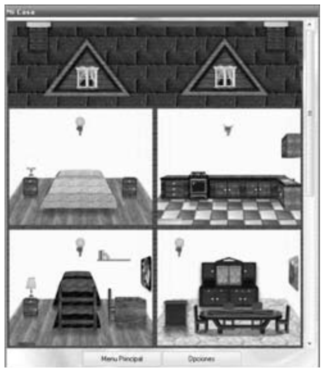

Figure 2. Graphic Interface of controlled house

which will be used to start future sessions in the house's design software or to start local or remote sessions to manage devices. Once the user has finished the house design, these changes will be reflected in both databases. In the Figure 2 a screenshot of an already configured house is shown. There is a kitchen, two bedrooms and a living room, each one with a lamp on it. The location of rooms and the elements included (lamps, television, radios, etc) are fully configurable. The application allows the characterization of external devices too, as the outdoor lights or garden sprinklers. The essential goal of this interface is to be easily accessed by not experimented users.

3.2 Device-Control Software

The main function of this separate application is to provide access to different components modeled at the previously explained *Interface Design* stage. It is the main access to manage all the configured devices in the house. Here, the user is able to easily change the actual state of any devices. For example, it is possible to turn on the lights of a room, the television, alarm sensors or to activate the pet feeder. Moreover, the user can also enter pre-programmed routines specifying a device, the state that the device will take when the routine is executed and the execution period (daily, once a week, etc). If the user is away from home these routines allow the simulation of activity in the house. It is also useful to set classic schedulable tasks as the watering of the garden or making coffee. Again, any change reflected here (the change of state of any device or the insertion, edition or deletion of some routine) will be reflected both in the local database as well as in the remote database, which is located in web server. This process is explained later.

3.3 Device-control web application

Finally, this application allows the user system to start sessions and to do exactly the same tasks that can be done with the previously described local application. That is, to turn on/off different devices and also to insert, edit or delete pre-programmed routines. Significantly, the user only could initiate sessions on the Web when the local application is connected to the Server. This means that the computer at home has an active connection to the Internet and the application is currently on execution. This restriction is necessary for the particular home and server interaction. Otherwise, it could happen that the user tries to remotely change the state of some device when it is impossible to achieve this change. An important aspect to remark is that remotely triggered orders are not directly communicated to home computer, which is required to ask the server about recent changes. This unidirectional communication is explained in the next section.

4. LOCAL AND REMOTE ACCESS

The system access functionality is based in two types of communication to achieve consistency in both databases and to reflect locally or remotely started events. It must consider both sides as initiators of events: one of these communications is done when the changes

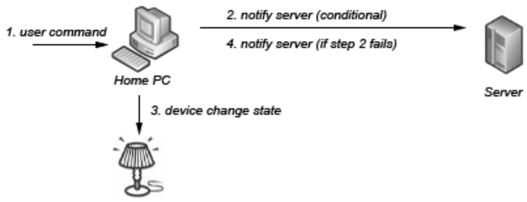

Figure 3. Ordering changes from home computer.

occur from the local application and other one is done when the events occur from the web application.

In the case that the changes are demanded from the local application, either on some device or a particular routine, the information about this event is gathered and sent to the server. The application uses a web service [6], which performs the update on the server database. Subsequently, the change is reflected on the local database and then the local application changes the status of the specified device or routine. In the Figure 3 the general steps of this process is shown. When the user does not have Internet access, the changes are only recorded on the local database and then the selected action is applied to the specified devices or routines. This generates a temporal inconsistency with the server database, but the synchronization is scheduled to be initiated right after the application is connected to the server again.

In this manner the local use of the system is not prejudiced if it occurs any problem with the Internet provider (ISP) or any other cause that can disable the communication. If changes are made from the web page, there are a few differences, mainly because the web site does not communicate with the local application that is executing on the user's machine. In fact, the local application communicates with the web application. The one-way access is also safe and simple to use in usual family homes. Figure 4 shows the general steps of this process. The user decides to modify the state of some device (or change the configuration of any routine) by remote access (the web application) and this change is registered in the server only. The local application then periodically makes a query to the server, which implements a web service indicating whether there are recently ordered changes.

This periodic query made by the local application to the server is achieved by two separate threads, one for devices and the other for routines. These are called `DeviceControlThread` and `Routine-ControlThread`. The first one controls changes in

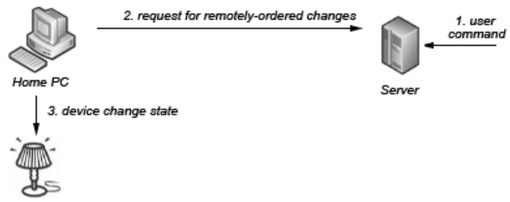

Figure 4. Changes from the web Server

the state of any device and the second one controls modifications of any routine. In the case of `DeviceControlThread`, it only uses a web method returning an array with the identities of active devices that are stored on the server database. Later on, these devices are activated and the rest are disabled, committing the changes on the local database. Right after that, actual changes on the devices through are achieved through parallel port. For changes on any routine, the `RoutineControlThread` requests a web method returning an array with the active routines (that is, the identities of enabled routines) and an indicator to control the existence of new routines or changes on existing routines. In this case, the thread requests a new web method asking for information about these changes. These new (or updated) routines are registered on the local database. Later on, the routines in the array are activated and the rest is disabled. Figure 5 shows this interaction between the client and server.

The procedures described above are repeated periodically. The parameter indicating the period of update can be configurated by the user. Basically, a low

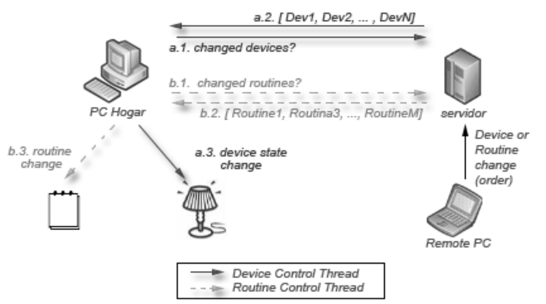

Figure 5. Devices and Routines Threads

value means a more frequent asking for changes, with the disadvantage of an increase in the number of request to the web services. A high value, however, decreases the numbers of request to the server, but there is a longer gap of time between requests, and thus the inconsistency between the databases may be temporarily noticeable.

4.1 Visual Interface to manage devices

The web interface of this domotic system is very intuitive given the use of AJAX technology [9], This provides a look and feel similar to the application located in the house. Several images are representing the rooms and devices installed in the house (as shown in Figure 2) and the partial refresh of the page greatly improves the use of the system. As a turned-on device has a diferent image than a turned-off device, then at the moment an action is performed on that device, updating only the small associated image provides greater performance when the system is accessed through the web. The same effect is achieved when inserting or editing pre-programmed routines. All of the selectable options in the interface are obtained from the database using AJAX. Only the selected control is refreshed and not the entire page as it would happen under a classical web implementation. In the Figure 6 the main interaction in this scenario is depicted. There are two levels of client-server communication. The inner arrow indicates a command sent to the server to handle devices. This command occurs when the user clicks on some device (marked with squares on the interface). The outer arrows indicate a periodically AJAX interaction that examines the state of the house and then visually reflects every change on the interface. This interaction avoids the need to refresh the entire page. It is appropriate to maintain a constant, periodical asynchronous exchange of data with the server to reflect the state of the house and its devices.

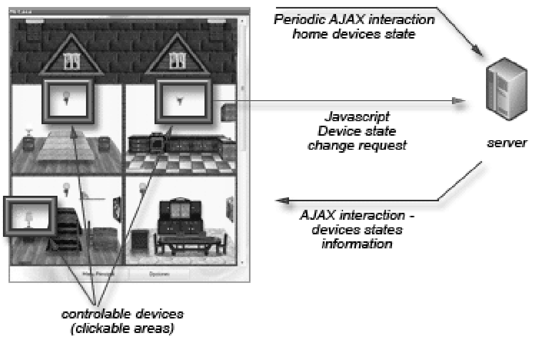

Figure 6. Visual Interface and AJAX

4.2 Implementation and test of the application

The system was implemented using .NET technology, in order to quickly obtain a prototype. Both the Device Control Application and the Interface Design Application were made with *WinForms* using C#. The web application is built using ASP and *Code Behind* in C#, the same is true for the web services. It was used *SQLServerExpress* for the databases. Regarding security, the password is encrypted and kept in both databases using the encryption algorithm provided by SQL Server.

Hardware was constructed considering two alternatives for a typical home: wired and wireless connections. The last one provides more flexibility when installing the system as the use of cable may be inappropriate in some rooms. This implementation uses a transmitter module called *TWS-434* and a receiver module called *RWS-434*. Both are very simple to use, its size is small and they use a 433.92 Mhz radio frequency within the electromagnetic spectrum. The modulation is ASK and can be used in other applications too.

The inner operation of the hardware is very simple. When there is an attempt to change the state of some devices, the corresponding code to activate or deactivate such devices is sent to the parallel port. For example,

- Code `00000001` means "*activate only the device related to output 0 of parallel port*".
- Code `00000101` means "*activate the devices related to output 0 and output 2 of parallel port*".

The output of the parallel port feeds the encoder, which in turn generates the input for transmitter *TWS-434*. When the signal is sent, it is received by the *RWS-434 receiver*, which must convert the signal into a similar original code. Relays are connected to this encoder, which will act as an electric switch, activating or deactivating the corresponding devices. Clearly, the wired option does not pass through the transmitter. It simply activates the corresponding relays, with the data taken from the parallel port.

The system was successfully tested in a real environment. We put special emphasis on maintaining the consistency in both databases, as well as the response time of the methods provided by the web service. The test was achieved with high load on the net and the result was very satisfactory. The databases were in an inconsistent state for a short period of time (according to the appropriate interval settings mentioned in Section 4) and the traffic generated by the constants requests from the application did not influence in the performance of the network. We also tested that the pre-programmed routines are executing correctly and we check that at the end of its execution they are updated properly. In all the cases, orders were sent from home as well as from remote locations.

Another interesting point that was evaluated is the appropriate frequency of requests to the web service. This frequency is established by a specific parameter. These requests are needed to communicate to the Control Application that there are recent changes in the house; if there is a low bandwidth Internet connection it is appropriate to reduce the frequency of the request. This is achieved by setting a higher value to the parameter representing the time between requests. In some tests, with low available bandwidth, a value between 25 or 30 seconds is admissible. In those cases where high bandwidth is available, it is possible to set requests more frequently. This is achieved by setting a low period of time between requests. This leads to a higher refresh rate in the databases, thus allowing faster updates.

Finally, another important point that was evaluated in the tests is the visibility of devices in wireless mode. We reached 070 meters in the interior of a house with several obstacles between the computer and the device (walls and doors). Some tests were made inside a big building and there may be distance of four floors, with acceptable response time. This shows that the developed hardware components are quick enough to not influence in the overall performance of the system, despite the general activity of the network.

5. EXTENSIONS

Some planned extensions are under development. The first extension is related to programmed routines and is intended to provide intelligence to the system. The idea is to observe and learn the normal activity in the house, in order to produce similar routines.

Automatic generation of routines

Our application allows the definition of routines to control home devices. The configuration of these routines can be done both locally and remotely, as described previously. However, it is interesting to implement an intelligent module for automatic routines generation. The system may observe the use of multiple devices over a reasonable period of normal activity at home and then it may automatically generate routines to simulate these activities. For example, the outside light of the front door of a family house is usually used at night, and the TV is mostly turned on in primetime hours. In a teenager's room, the music player is not likely to be used early in the morning, but it is always turned on during the first hours of the night. This, of course, depends on the daily routines of the family or home owner, and undoubtedly it will present small variations from day to day. However, in general terms, the use of household electrical appliances follows a general pattern within a long period of time. Thus, the domotic system may include a module for *learning routines*.

There is a previous project, faced by the authors of this paper, in which an application looks for different patterns in a specific database and indicates the possible outcomes given certain events. It was developed in Java and it uses *Weka* libraries [10] to resolve the searching of various patterns in the database. *Weka* is a collection of learning algorithms for automated data mining. These algorithms can be directly applied to the data set and they can be called from any Java code. *Weka* contains tools for pre-processing of data, classification, *clusterization*, rules of association and visualization, and also is a good tool for the development of automated learning. The integration of this Weka-based application can be easily done if every action performed on any device is stored in a database or log file. It is necessary to know the device, the day of operation and the time of the event, both for the actions carried out manually and for those driven from the user´s computer. Once the generated log is big enough, the data mining and learning module will determine, given a device or a group of devices, the frequency rate of operations and the particular actions (on/off). Then the module suggests a routine suitable for those devices. The most difficult aspect of this learning process is the constant observation of the indoors activity. This is because not every device is operated by PC. Thus, for those devices manually operated, a specific sensor must be installed. Our application, however, is suitable to log every change ordered in the (local or remote) PC interface. This module belongs to the local application and it requires the ability to detect and collect the real state of some devices. As stated before, this can be easily done when the changes are applied using any PC as an access point to the system. However, it is also possible to implement the automated routine generation through a web service, which should interpret the results of the data mining and translate this information into a model understandable by the application that controls devices. This Web Service could be consulted periodically, such as once a week or once a month. Then the results of this service will be displayed to the user as suggestions for possible routines for to the system.

Handling telephone devices

Another planned extension is the control of devices through the telephone line or a cell phone with Internet access. For the first possibility, one solution is to add a DTMF (Dual-Tone Multi-frequency) module to the telephone line, acting in a similar way as an answering machine. DTMF is a method for instructing a telephone switching system of the telephone number to be dialled or to issue commands to switching systems or related telephony equipment. Once this module answers the call, it stays listening for instructions. The instructions are sent by pressing different phone numbers in the calling phone and thus the module can link each number to previously configured activities. This implementation is very practical, because it just makes a phone call to the house to take control of some devices. Using this module it is possible, for example, to feed pets or turn on outdoors lights on command. The second possibility is to control the house through a mobile device with Internet access. The idea is simple, as it just requires the implementation of WAP version of the web server, offering the same services.

6. RELATED WORK

Several authors propose domotic solutions focused on specific issues. In this paper, the emphasis is placed on the universality of the technologies chosen in order to be widely accessible for typical users. Thus, only a few physical devices and an Internet connection are required. An interesting work is done by Marco Aiello [1], which describes various scenarios in which they have developed domotic systems. According to the classification given by Aiello, our implementation is an instance of *Domotic Scenario S3 (open-server hierarchy)*, providing very good heterogeneity and an acceptable scalability. The only risk of this scenario is centralization, which may become a bottleneck using web services. Importantly, our implementation also allows control offline of devices, which also corresponds to a *Domotic scenary S1* as indicated in [1]. Another important work is done by Araujo *et al* [5], in which he proposes an implementation to wireless management of devices. Our system has some similarities with the *Radio-Hotel* mentioned in [5]: air conditioning in the room is turned on or off manually, but it also allows to change its state from the Hotel Reception. Our system can work perfectly in an Intranet, with or without a central server. An important difference is that the proposed Araujo *et al* captures the state of devices or sensors and operates automatically according to the detected state, something missing in our proposal because the cost of needed sensors. Vittorio Miori et. al. also developed a work [3] which proposes the use of Web Services to link different communication protocols or different types of networks. An administrator interprets the outcome of the web service and then it transmits its results to the underlying architecture. Our work uses web services, and thus it is suitable to be applied according to the proposal in [3], since it is possible to manage devices using *JINI* or *Komex*, creating the appropriate controller without further modification to centralized system. In [4] Bonino and Garbo present an interesting application to control devices, intended to be used by handicapped persons. It can be manipulated through eye movements or head, the main feature which governs the overall design. The interesting thing in relation to our work is that it provides interoperability with different domotic existing networks, as *Bticino MyHome* [8].

7. CONCLUSIONS

The Domotic Systems are intelligent systems for homes and apartments, which integrates and controls communications, security, climatic adjustments and handling of electric appliances. This kind of system generates benefits in terms of comfort, safety and energy savings. This paper presented an implementation of a system to control home devices using web services and a common domestic Internet connection. The devices can be connected to a computer at home and they can be manipulated or programmed either *locally* or *remotely* by accessing a web server.

The web application is implemented using AJAX to optimize the overall operation. The Web Service based architecture allows the independence between the local application that controls the devices from home and the server that allows remote access.

For the physical connection of devices, we built a hardware module connected to the parallel port of a local computer. Its interior has relays to handle wired devices, and also contains an emitter with the respective encoder for turning on or off wireless devices.

8. REFERENCIAS BIBLIOGRAFICAS

[1] *The Role of Web Services at Home.* Marco Aiello. Proceedings of the Advanced International Conference on Internet and Web Applications and Services AICT/ICIW 2006. IEEE Computer.

[2] *Seguridad en Redes y Criptografía.* Dante I. González Sánchez. Tesis de Maestría. Instituto Tecnológico y de Estudios Superiores de Monterrey. 2004.

[3] *DomoNet: a framework and a prototype for interoperability of domotic middlewares based on XML and Web Services.* Miori, Tarrini, Manca y Tolomei. International Conference on Consumer Electronics, 2006 ICCE '06. pp. 117-118. 2006 Digest of Technical Papers-IEE.

[4] *An accessible control application for interacting with domotic environments through Gaze and Head Movements.* Dario Bonino, Alessandro Garbo. First International Conference on Ambient Intelligence Developments, September 2006, Sophia-Antipolis, pp.11-27. Ed. Springer-Verlag, ISBN-10: 2-287-47469-2

[5] *Domotic Platform Based on Multipurpose Wireless Technology with Distributed Processing Capabilities.* Araujo, Fraga, Moya y Nieto-Taladriz. 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. pp 3003-3007.

[6] *Web Services Architecture.* W3C Working Group. Editado por David Booth, Hugo Haas et al. Artículo on-line. http://www.w3.org/TR/ws-arch/.

[7] *The Networked Home: An Analysis of Current Developments and Future Trends.* Alladi Venkatesh, Erik Kruse, Eric Chuan-Fong Shih. Cognition, Technology and Work, 5 (1), 23-32.

[8] *Bticino MyHome System.* http://www.bticino-myhome.it

[9] *AJAX: a new approach to web applications.* Jesse James Garret. Adaptive Path Essay. Febrero de 2005. Artículo on-line. http://www.adaptivepath.com/ideas/essays/archives/000385.php

[10] Weka: Data Mining Software. http://www.cs.waikato.ac.nz/ml/weka/