

VIRTUAL DOMOTIC SYSTEMS: A NEW 3D MULTI-USER, HIGHLY IMMERSIVE AND INTUITIVE INTERFACE FOR RESIDENTIAL GATEWAYS

Carmen Lastres Redondo, Francisco Javier Luque Cañones, Francisco Pedro Luque Oostrom, Iris Galloso Guitard, Asunción Santamaría Galdón
CeDInt-UPM. Universidad Politécnica de Madrid
Edif. CeDInt-UPM. Campus de Montegancedo. 28223 Pozuelo de Alarcón. Spain

Jordi Muria Portillo
T-Systems Iberia
CRV.C/ Llorens i Artigas, 4-6. 08028 Barcelona, Spain

ABSTRACT

Residential gateways are systems based on different control technologies that provide a unified way of communication for various devices and appliances present in homes and buildings. These devices include sensors (for the detection of people, temperature, lighting, security, flooding), actuators (switchers for heating systems, dimmers, control mechanisms) and electrical and electronic devices (appliances, audio and video devices, multimedia devices, personal computers, etc.). Residential gateways are used to control and optimize the energy consumption in buildings. Unfortunately, they frequently provide non-intuitive interfaces, thus making their operation complex to users. This paper presents the results of an R&D project that has shown the feasibility of using Virtual Reality environments to create new 3D interfaces for residential gateways. The main purpose of this project has been the design and development of a system that provides a more usable and intuitive interface to interact with commercial residential gateways. As added value, the Virtual Domotic System (VDS) offers multi-user control and the possibility of it being operated from immersive and non-immersive visualization systems. The VDS and the residential gateway are connected and communicate through the Internet.

KEYWORDS

User interface, intuitive, domotic, Virtual Reality, residential gateway.

1. INTRODUCTION

There is currently a wide variety of residential gateways on the market. They are systems based on different control technologies that provide a unified way of communicating and controlling domotic devices and appliances found in homes or buildings. These systems are becoming very important because they allow the energy consumption in buildings to be reduced by the user or by implementing Ambient Intelligent (AmI) algorithms. Residential gateways can be controlled in local mode from various devices such as touch screens, PCs with keyboard and mouse or from the television set. They also offer remote access through the Internet.

Depending on the technical solutions, residential gateways offer different user interfaces. Some gateways are based on command lines. In these cases, users first have to understand and then, learn how to control the devices. Some others have text-based menus through which users have to navigate to find a device or to launch a command. Moreover, the user usually receives limited feedback on the devices' status (lights, shutters, air conditioning systems) or the acknowledgment of commands sent to the domotic devices.

In fact, the lack of intuitive user interfaces is one of the reasons why users are reluctant to install residential gateways in buildings or even in their own homes. This is hindering the penetration of residential gateways in consumers' market.

This paper presents the results of an R&D project that has shown the feasibility of using Virtual Reality environments to create new 3D interfaces for residential gateways. The main purpose of the project has been

the design and development of a system that provides a more usable and intuitive interface to interact with commercial residential gateways (Molina et al. 2002).

When using the Virtual Domotic System, the user will no longer need command lines and complex text menus. The system provides direct interaction with a 3D model that accurately reproduces the building, including every appliance that can be controlled from the residential gateway. The user navigates inside the building in a similar way as in a video-game. The user does not need any previous knowledge on which appliances can be controlled, since these devices are automatically highlighted by the system, as soon as the user approaches them when navigating in the virtual model.

Another interesting feature of the Virtual Domotic System is the possibility of navigating through the house checking the real status of lights, heating systems, temperature sensors, etc., thanks to the synchronization between the virtual model and the real status of domotic devices. This is of particular importance for users with special needs such as those with mobility disabilities.

Moreover, the system is able to operate in multi-user scenarios, where several users can access the residential gateway simultaneously to control either the same room or different rooms of the same building. The system always maintains the synchronicity between the real and the virtual environments. Actions carried out in the real world are immediately updated in the virtual representation of the building. For instance, if a user enters and switches on a light, the virtual model shows a warning message and updates the virtual model with the current status of the lighting.

The VDS is the virtual representation of the real building. The residential gateway is the system that communicates and controls the real world. Both systems are connected to the internet and they communicate through it. Thus, the VDS does not need to be in the same place as the building (see Figure 1).

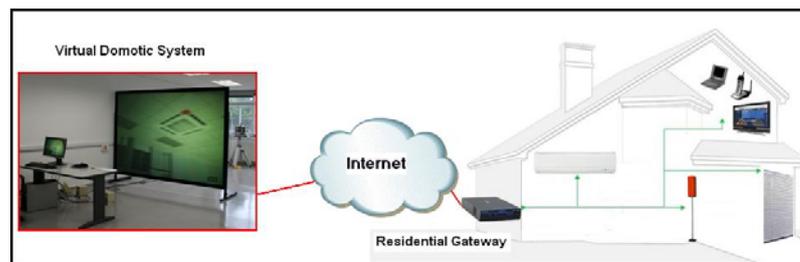


Figure 1. Virtual Domotic System. General Diagram

Finally, it should be highlighted that the application can be operated from all kinds of Virtual Reality sets, that range from simple non-immersive systems, such as a PC with a keyboard and a mouse, to sophisticated visualization immersive environments such as stereoscopic projection systems (stereowall, CAVE, etc.) (Brooks 1999, Kalawski 1993, Rusty 2004, Burdea 1994).

2. SYSTEM ARCHITECTURE

The architecture of the virtual domotic system comprises the following elements, as shown in Figure 2:

- *Domotic devices*. Set of devices used to control electronic appliances present at homes or buildings. Two main types of domotic devices can be found:
 - Sensors. Used to measure specific parameters (temperature, luminance, humidity, etc.).
 - Actuators. Used to control (open, close, switch, dim, etc.) electronic devices that operate on lights, shutters, blinds, heating/air conditioning, etc.
- *Residential gateway*. Platform to provide access and control to domotic devices. They are intended to manage a wide range of domotic devices (sometimes, even of different technologies). The residential gateway offers a unified user interface to send commands and receive information about the state of the domotic devices. This user interface is frequently neither friendly nor intuitive. The aim of the project presented here is to replace it with the VR visualization interface.
- *Information Exchange Module*. Software module that gathers and processes the information available in the residential gateway (configuration and status of devices). It converts this information into a format

compatible with the Visualization and Development Tool. Reversely, it translates any command provided by the VR user and sends it to the residential gateway.

- *Main application module.* Software module that virtually recreates the real environment. It generates the virtual interface for the user, providing menus for interaction and representing in a completely visual scenario the updated status of devices, rooms, etc.

- *Visualization interface.* It shows the virtual scene in both non-immersive systems (PC with desktop application and simple peripherals) and in immersive systems (with partial or total immersion).

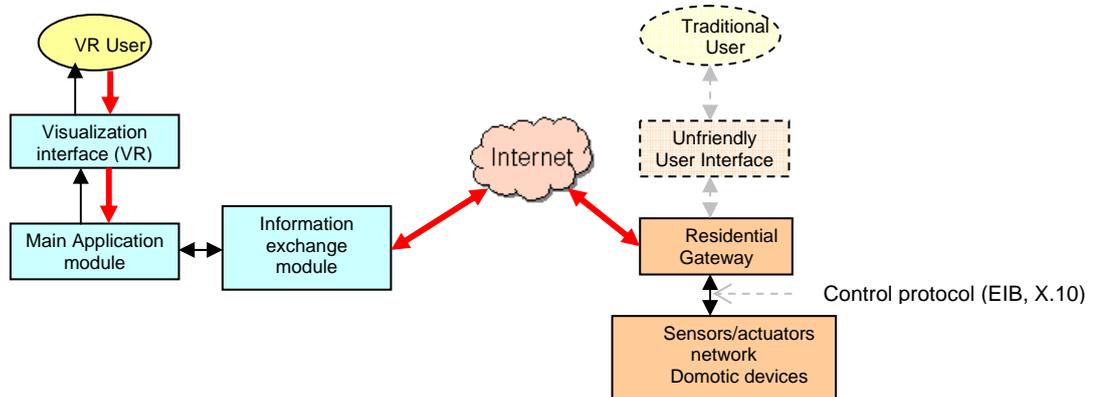


Figure 2. Architecture of the System

The VDS is managed by a control module, responsible for the control and synchronization of the whole system. As shown in **Error! Reference source not found.**, the use case analysis can be broken down into the following actors, use cases and dependencies among them:

- Actors:
 - Residential gateway: system that communicates with the domotic devices, using a domotic communication protocol.
 - Navigation interface: manages the user's navigation within the virtual model.
 - Initialization and synchronization routine: establishes the connection with the residential gateway, thus guaranteeing the correspondence between the virtual model and the real environment.
 - Network module: establishes the communication with the residential gateway.
 - Event listener server: stays permanently registered to the residential gateway waiting for the reception of events. It notifies those events to the control module, as soon as they happen.
- Use cases:
 - Receive events: the residential gateway notifies to the control module when a change of status takes place (change in a sensor's value, a device is activated, etc.).
 - Send actions: the user sends actions (commands) from the navigation interface of the virtual model or the user physically activates a device.
 - Routine execution: the model contains pre-defined scenes (executed as a set of actions).
 - Model configuration: when initialization takes place, the initial distribution of the registered locations and devices is downloaded from the residential gateway. It is also periodically activated (at the request of the synchronization routine).

The main innovation and contributions of the R&D project are the development of two blocks the Information Exchange Module and the Main Application Module. These two modules are presented in more detail in the following sections.

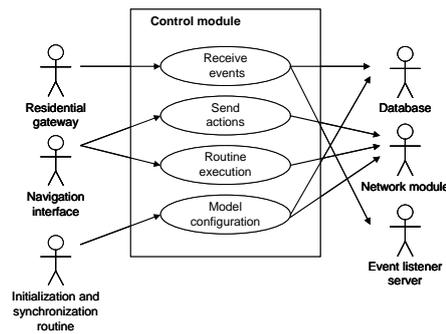


Figure 3. Use Case Diagram of the Control Module

3. INFORMATION EXCHANGE MODULE

The Information Exchange Module (IEM) block is in charged with establishing the communication between the residential gateway and the VDS. The main goal of the IEM is to achieve the synchronization of status and the transmission of events from the residential gateway to the virtual reality environment and vice-versa.

The module is based on the sequential exchange of XML messages (Rusty, 2004), thus eliminating the need to store the information on the devices' status in intermediate components (such as databases). This working principle guarantees the scalability and adaptability of the overall system.

Consequently, the Information Exchange Module has two main blocks:

- The connection and configuration module. It analyzes and builds the XML messages.
- The event listener server. This server must be registered at the residential gateway. To do so, the server sends a message to the residential gateway with its IP address and its communication port. The residential gateway is programmed to send its messages to all entities registered in it. So, once registered, the event listener server receives any incoming message from the residential gateway in real time. Every time a device status changes, the residential gateways sends a message each to its registered entities. This is the way to achieve multi-user functionalities. Several VDS applications can be registered in the residential gateway, in such a way that all them can communicate and operate with it.

One of the main challenges of this project is the integration of the network functionalities necessary for the development of this module. This has been achieved by programming separate functions using Visual C++ that are later used by the Visualization and Development Tool as described in section 3.2 of this document.

3.1 Connection and Configuration Module

The main task of the connection and configuration module is to perform the parsing of the XML message, according to the semantic rules of the document.

The connection and configuration module downloads the information on the locations and devices registered in the residential gateway from the real world. This process is carried out in the boot of the system in order to initialize the virtual model according to the real status of the devices, but also during runtime, thus ensuring permanent synchronization between the domotic devices and their virtual representation. With this process, changes in devices in the real world (for instance, a light is switched on) are immediately displayed on the VDS.

XML messages have the following information fields:

- Locations: name of places (rooms, areas) registered in the residential gateway and devices installed in each location.
- Devices: types of domotic devices and their current status.
- Action: description of the event that can be executed by a specific device.
- Routines: set of actions involving one or more devices (also called scenes).

As a result of the parsing process, the information obtained is stored in a set of tables (see description in Table 1). These tables are built dynamically when the connection between the residential gateway and the Virtual Domotic System is established. Moreover, they are asynchronously accessed by the control module and the visualization interface.

Table 1. Set of Tables Obtained after the Parsing Process

NAME	DESCRIPTION
Locations table	Stores the locations registered in the residential gateway at the moment of connection. Contains the list of the devices associated to them.
Devices table	Contains the list of devices working in each location, together with their id, category and type.
Device type table	Stores detailed information about the actions possible for each type of device (switches, dimmers, shutters, thermostats, etc.).
Scenes table	Consists of a set of scenes (or routines) that the residential gateway can translate in a given subset of device actions.

As soon as the system has read the XML messages and built the tables, it is ready to start its two main activities:

- *Invoke actions and scenes.* When the user selects a specific action or scene from the virtual interface, the order is translated into a given URL address and is sent to the residential gateway via HTTP.
- *Receive events.* The module interprets the events received from the residential gateway by the event listener server and modifies the configuration tables accordingly.

The aforementioned modules have been implemented using the visualization and development tool “Virtools” (McCarthy, 2004). This tool is based on a graphic interface, where the execution line is established through the interconnection of basic behavior functions called Building Blocks (BBs). These BBs can be grouped into Behavior Graphs and scripts in order to achieve more complex results.

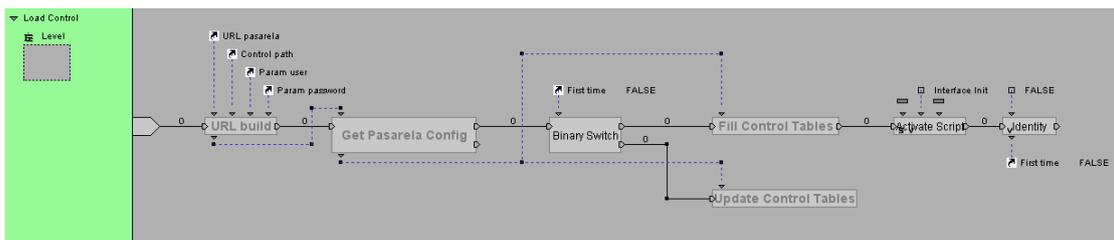


Figure 3. Example of Implementation. Building Blocks of the Control Routine

3.2 Event Listener Server

This module receives events related to the change of status of the devices. “Event” is understood as a message sent by the residential gateway with data describing the new status of a particular device. The server allows the system to be synchronized with the residential gateway as long as all messages notifying changes in status are received. As this hypothesis can only be assumed in an ideal environment, the complete synchronization is achieved by periodically polling the current situation to the residential gateway as described in section 3.1 for the boot process.

The residential gateway implements a service to register “listeners” interested in receiving status messages when a change happens. Messages are then sent to all interested users allowing the system to work properly in a multi-user environment. The structure of the Event Listener Server comprises the following main blocks:

1. Get events: implements the event listener and the parsing of information contained in the XML message.
2. Device finder: checks that the event refers to a device already stored in the tables. If the device is not found, then the message is considered as “not valid” and is ignored by the server.
3. Get parameters: processes the data in the messages, obtaining the values to be updated in the tables.
4. Update database: stores the new values in the tables and activates routines to modify the state of the devices in the virtual model.
5. Send notice: sends a message to the user interface to notify changes of status.

4. MAIN APPLICATION MODULE

The Main Application Module has three significant features:

- recreates the rooms and the distribution of the domotic devices in the real environment virtually,
- presents the information on the real status of the devices in a visual and intuitive way,
- provides interaction menus to configure the real environment.

4.1 3D Modeling of Home/Building Environment

The 3D model of the real home or building has been created, in this particular project, using 3DStudio Max 9 (Autodesk), a tool capable of designing any type of object that also provides files formats suitable to be imported with Virtools. 3DStudio translates .MAX files into .NMO format, compatible with Virtools.

The environment has been built starting from the CAD files of the building, containing detailed information about dimensions, position of walls, pillars, etc., of the building.

The next step was the recreation of objects' textures and materials, paying special attention to their illumination. Firstly, light processing was developed using Maxwell Render, a powerful tool for illumination, but finally the implementation of textures and light was carried out by means of *lightmaps*. Lightmaps are textures that automatically save the materials and light of objects. They reduce the load of processing during runtime, although they present a static shading of objects (invariant position of the light in the model).

Though 3D modeling may be a time-consuming task, this effort is nowadays minimized thanks to the widespread availability of buildings' CAD design. This fact, together with the possibility to create room/scene editors to simplify the 3D modeling task, may be of help when undertaking the commercial implementation of VDS.



Figure 4. 3D Model of CeDInt-UPM Building

4.2 Navigation Tools

The navigation system developed in the project allows total interaction with the virtual model just by using the mouse and cursors on the keyboard. Two navigation modules have been developed:

- Cursor navigation: declares the keys that can be used to navigate, as well as the speed of the movement and changes in direction (also implemented with W, X, A and D keys).
- Master navigation: controls the collision detection with obstacles. This is an important factor when realistic navigation is desired, since the user should not be able to go through walls or closed doors in the virtual environment.

4.3 Interface Menu Development

The interface menu provides a wide variety of features to make navigation easy and intuitive. These features are presented here through the description of the different scripts the module is made of:

- *Pointer script*: It shows a pointer in the centre of the screen. When the user moves around the scene and the pointer targets a domotic device, this specific device is highlighted in red and a sound is emitted. In this way, the user easily knows which objects are capable of being activated by the residential gateway.
- When the user selects a device, the menu of specific actions available for that device drops down and the user can choose an action to be launched. This sequence of events is depicted in **Error! Reference source not found.**

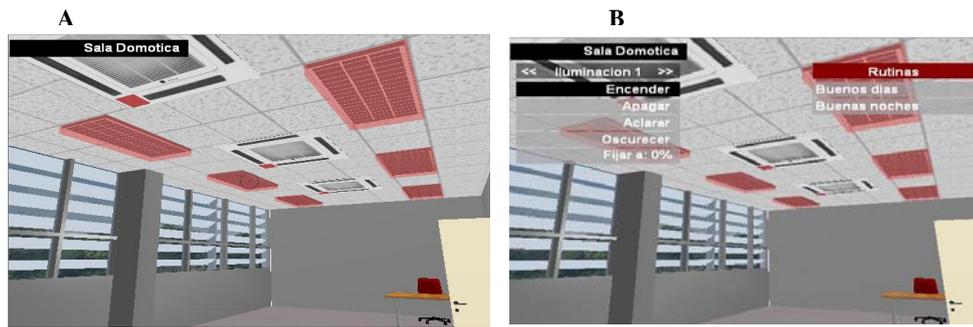


Figure 6. Pointer Script Features: Selection and Activation of Devices. A) The Pointer in the Center of the Screen Allows the Selection of Illumination Systems (Highlighted in Red). B) Specific Actions Menu for Illumination Devices

- *Interface init.* It composes the menu for any device in the room where the user is. In this way, the system is always ready for any sudden action the user may do.
- *Update menu.* It creates and deletes the buttons with the different actions possible for the selected device at every moment. It creates the buttons but does not show them on the screen.
- *Update scene menu.* It builds the buttons related to the scenes already pre-configured in the residential gateway (they cannot vary during runtime). For example, a scene called "Good Morning" can be configured to raise all shutters and turn off all lights in the building.
- *Parent sliding 2D frame script.* It contains all the action buttons of the selected device and shows/hides these buttons on/from the screen. The *Parent Sliding 2D Scene Frame script* performs the same function, associated to scenes.
- *Next and Prev scripts.* They allow moving forward and backwards within the list of devices in a room, providing an extra navigation tool.
- *Current room script.* It shows a label with the name of the room where the user is.
- *Event advertisement script.* It creates labels that show modifications in the state of the devices. This is especially important when the system is used in a multi-user environment, since it shows messages about changes the other users have made in other rooms. Moreover, another user could physically activate or adjust a device and the system will notify it to the VR user by means of these labels.
- *Temp Display Script.* In case a room has a thermostat-type device, the system shows the configuration temperature of the heating/air conditioning system in the room.

5. THE SYSTEM UP AND RUNNING

In the previous sections, the general structure of the system, its main building blocks and working principles have been presented. This section describes the specific characteristics of the Virtual Domotic System that has been successfully implemented and is currently up and running.

In order to implement a real system, the following technical solutions have been chosen, regarding the different elements of the architecture:

A set of KNX (previously EIB European Installation Bus) Merten devices have been installed and configured through the ETS3 tool (Instabus, 2003; Millán, 2001): blind controllers, switchers, dimmers, air conditioning control and RS-232 interfaces. These devices have been connected to an OSGi-based platform (OSGi, 2002; Haiges, 2004), from eNEO Labs.

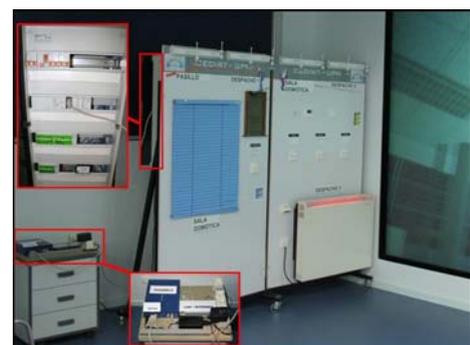


Figure 5. System Set-up: Stereowall, Domotic Devices Installed on a Show-panel and Residential Gateway.

The capacity of the system to report devices' failures will depend on the capability of the domotic technology to report this type of events. The KNX devices used in the project are able to report physical devices' status (on, off, level, etc.), but not devices' failures (light broken, motor failed, etc.).

The virtual environment modeled in the project corresponds to the domotic show-room, the corridor and two offices located on the first floor of the Centre for Integral Domotics of the UPM. As visualization infrastructures, an immersive stereowall and non-immersive PC environment have been tested.

Performance tests in terms of rendering frame rates have been performed over a PC with AMD processor Opteron 252, graphic card NVidia Quadro and 4GB of RAM memory. The tests have been carried out in a 3D scenario with 3 different configurations:

- High quality: high resolution textures and advanced lighting effects, at 50 to 70 frames per second (fps).
- Medium quality. Reduced textures and advanced lighting effects: 100 – 120 fps.
- Low quality. Reduced textures and basic lighting effects: 180 – 200 fps.

Therefore, in case of complex scenes, the three following major acceleration methods can be used: reduction in textures' quality, use of less advanced lighting effects and programming techniques based in portals that process only the objects of the current room.

The following figures show the system working.

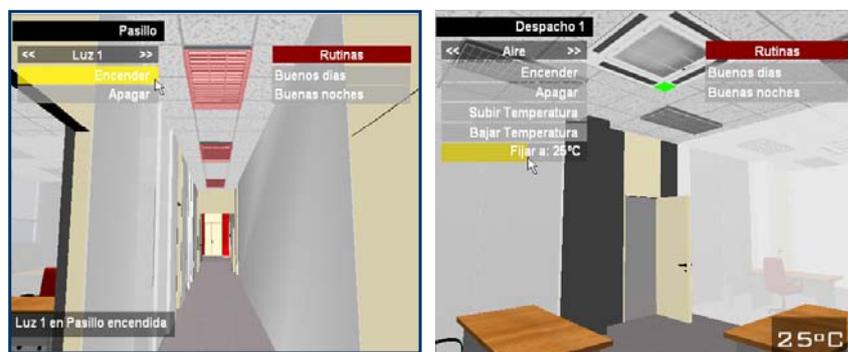


Figure 6. (left) Selection and Activation of Lights in the Corridor. (right) Selection of Heating System. Activation. Modification of Temperature



Figure 7. Domotic Show-room. Selection and Opening of Shutters

6. CONCLUSIONS AND FUTURE RESEARCH LINES

This paper presents the conceptual structure of a Virtual Domotic System and its successful implementation in a fully functional real system, thus showing the technical feasibility of the idea.

The VDS presented can control a real environment (building) by connecting to a residential gateway through an http connection. The residential gateway used in the implementation of this project is an OSGI compatible gateway working with a sensors/actuators network based on KNX Standard, commercially available at eNEO Labs (www.eneo.com).

The VDS supports multiuser features, in such a way that every registered user can use the residential gateway, controlling the real world. Data interchange between the VDS and the residential gateway is based on XML messages. So, VDS is easily exportable to other gateways with internet connection facilities.

The VDS has resulted in an easy-to-use application that may help to introduce domotic systems for energy management at buildings. This allows the potential users to extend to non-technical people that will find a very appropriate tool to control the energy consumption in buildings. People with limited mobility can also carry out vigilance tasks by checking the status of electronic devices using the VDS.

The usability of the VDS has been qualitatively tested with advanced users, mainly R&D professionals. The VDS provides enhanced intuitiveness (easiness to perform configuration tasks without previous knowledge of the virtual and real systems) compared to other residential gateways' interfaces. Nevertheless, further research is on progress to quantify the improvement in the usability for different types of users.

The VDS implements a user-password security layer that relies on the access control system of the residential gateway. Further development on this security layer may be necessary to ensure the necessary confidentiality and security in multi-user and internet access scenarios.

The reliability of the system in a multi-user environment has been tested with up to 5 users accessing simultaneously to the system. Testing of the system behavior under massive access is one of the future research lines of the project. The actions taken when several users compete for the same resource are determined by the residential gateway (order in which it receives the commands). Nevertheless, user profile based priority policies may be studied and implemented in future versions of the prototype.

Once the main tool has been developed, additional features can be integrated in VDS, such as a scene/room editor to facilitate 3D modeling; inclusion of video streaming devices (for information and surveillance purposes); support of different types of sensors (water/gas escapes, user presence, opened doors/windows, etc.), and the generalization of the system for any type of residential gateway.

ACKNOWLEDGEMENT

The work here presented is part of the R&D project "DVD – Nuevos Sistemas de Domótica Virtual para Discapacitados" (New virtual domotic systems for disabled), developed during years 2007 and 2008 in cooperation with the companies T-Systems Iberia and eNeo Labs.

Authors would like to thank the Spanish Ministry of Industry, Tourism and Commerce, for the support provided through the funded projects FIT-350300-2007-41 and TSI-020100-2008-402.

REFERENCES

- Brooks Jr., 1999. "What's Real About Virtual Reality?", *IEEE Computer Graphics And Applications*, 19(6), 16.
- Burdea, G. et al., 1994. *Virtual Reality Technology*. JohnWiley. New York.
- Haiges, S. 2004. *OSGi Tutorial*. A step by step introduction to OSGi Programming. October 2004.
http://www.knopflerfish.org/tutorials/osgi_tutorial.pdf.
- Instabus EIB commercial catalogue, 2003. Siemens, S.A.
- Kalawsky, R. S., 1993. *The Science of Virtual Reality and Virtual Environments: A Technical, Scientific and Engineering Reference on Virtual Environments*. Addison-Wesley, Wokingham, England .
- McCarthy, C. et al., 2004. *Virtools Dev User Guide*. 2004.
- Millán, J. 2001. *Técnicas y procesos en las instalaciones automatizadas en los edificios (Techniques and processes in automated building installations)*. Thomson Editores, Spain. Paraninfo S.A. ISB: 84-283-2801-3.
- Molina, A. et al., 2002. "La Realidad Virtual como Herramienta para el Aprendizaje de la Domótica" (Virtual Reality as a tool to teach on domotics). *6 Congreso Iberoamericano, 4 Simposio Internacional de Informática Educativa, 7 Taller Internacional de Software Educativo*. Vigo, 20-22nd November 2002.
- Rusty H.E., 2004. *XML 1.1 Bible*. 3rd Edition. John Wiley & Sons, Inc. London.