

DH Compliant Adapter to Integrate a Surveillance Robot in the Digital Home

V. García Sánchez, I. González Alonso

Abstract—Today, the use of service robots in the digital home has increased significantly. Therefore, our research group has developed the DHCompliant protocol in order to integrate in a transparent way robot tasks with the home automation services. In this research paper, it is described the development and running of an adapter for integrating a surveillance robot in a digital home. By using this adapter, we achieved an increase of interoperability with home automation robots and household items.

Index Terms—Adapter, Surveillance Robots, Digital Home, DHCompliant, UPnP, Interoperability.

I. INTRODUCTION

According to the publication World Robotics 2010 Service Robots of The International Federation of Robotics there has been more than 8.7 million service robots in the homes since 2009. It is believed that during the years 2010-2013 there will be an increase of 11.4 million robots [1]. Most of these service robots are cleaning, lawn and leisure robots, and with a lesser extent, passenger transport robots and security and surveillance robots [1]. With this background, it is expected a large increase in sales of surveillance robots including the one which has been used for this paper. The objective is to perform experiments in order to reduce the interoperability problems between these robots and other robots, and also with the digital home. To reduce this interoperability problem it has been developed an adapter compatible with the UPnP protocol [2] and DHCompliant [3][4] which allows to integrate the robot in the digital home. In section two there are stated some of the existing surveillance robots in the market and different experiments carried out to discover new features. Section three describes problems of interoperability between devices, and architectures and protocols which can reduce it. Section four describes the methodology applied to the experiment described. Sections five and six describe the experiments conducted and their results. Finally, conclusions and future work are presented.

II. MAJOR SECURITY ROBOTS AND ITS EVOLUTION

Today, the market offers several alternatives regarding security robots. The following TABLE I shows some of them and their characteristics.

TABLE I. SECURITY ROBOTS

Robot	Features
mSecurit [5]	Vigilance robot capable of patrolling outdoors in a semiautonomous form, detecting intruders and raising the alarm to the control center. This robot has been used in different studies [6].
Research PatrolBot [8]	Robotic platform designed for security tasks. Used primarily for research. Some of its features are: two drive and two directional wheels, reaches up to 2m/s, has 16 ultrasonic (sonar), up to 50 kg of cargo, prepared to work 24x7, fully programmable, webcam, etc. Its use can be seen in different experiments as [9].
Mosro	Mobile robotic and autonomous security robot for surveillance tasks on large indoor spaces. Some of its features are: fingerprint sensor, camera, siren and light of warning, radar sensor, detector of movement by infrared, speaker, gas measurement sensor, ultrasonic sensors for navigation, etc. More information can be seen in [9].
Ofro	Autonomous robot for external security. Some of its features are: it reaches 7.2 km/h, autonomy of 12 hours, 40 kg of cargo, infrared video camera and video camera CCD, gas sensor, infrared sensor, ultrasonic sonar, wireless and GSM/GPRS/UMTS, etc. More information can be seen in [9].
Rovio	Rovio is a telepresence and surveillance robot manufactured by WowWee. Some of its features are: three motorized wheels, a webcam, a microphone, LED lighting and a microphone.
Wifibot 4G [10]	It is a small robot designed for domestic surveillance. It is manufactured by WiFiBot. Some of its main features are: four drive wheels, wireless, two infrared sensors, webcam, etc. It has been used for several experiments [11][12].

These commercial robots have different characteristics and have been used for experimental studies in recent years. Some examples are the design of new security robots and the improvement of the services or the handling of these robots. Among the most significant advances are intruders

detection systems using webcams, thermal cameras, infrared, etc. [13][14][15], fire detection systems by incorporating sensors of smoke, temperature, flame, etc. [16], navigation systems and obstacles avoidance [14][17] and ultrasonic sensors with security systems, temperature, sound and light sensors [18]. Some of them also include internet and mobile phones management [15][18], systems capable of sending images to the internet and mobile phones by GSM [15][16], etc.



Fig. 1. Pioneer P3-DX

The adapter developed in this experiment allows the integration of a surveillance robot in the digital home, the Pioneer P3-DX (Fig. 1) which is a research robotic platform. The TABLE II shows its characteristics.

TABLE II. PIONEER P3-DX ROBOT

Robot	Features
Pioneer P3-DX [7]	It is one of the most widespread robotic platforms in research studies. Some of its features are: two drive and a rotating wheel, eight sonar ultrasonic, up to 25 kilograms of load, fully programmable, etc..

III. ARCHITECTURES AND PROTOCOLS FOR INTEROPERABILITY

The IEEE definition for interoperability is "the ability of two or more systems or components to exchange information and to use the information that has been exchanged". The main problem achieved between the interoperability systems is the wide variety of information systems used by companies; this entails the minimum requirements that are not usually established. However, it is important to note that the interoperability that is required is more than a simple exchange of information. In addition, companies have to facilitate the use of the same platforms and different languages, as well as important other aspects, such as the ability of interaction and implementation of simultaneous tasks. Throughout the history of distributed systems for the interoperability [19], it is very important to take into account the desire of cooperation and expansion in the networks that were already in use. The following sections describe the UPnP, HAVi and Jini architectures, and the DHCompliant protocol with the UPnP that is used

by the adapter. UPnP (Universal Plug and Play) The main feature of the UPnP technology [2] is the transparency when a device that has just been connected to the network is installed. All services of this device are available automatically without any configuration. A natural application of this technology will be home and build automation, which gives the possibility of creating distributed control architectures. The main idea of UPnP model is the use of two different components: the control point and the devices, so each device (a robot, a router, etc) gives access to the rest of the nodes in a local network (LAN).

A. HAVi (Home Audio Video interoperability)

HAVi [20] is an open distributed software architecture an independent platform and programming language. HAVi specifies a set of APIs designed for the direct interconnection of electricity consumption of digital video and audio of various types. The devices can meet the HAVi specifications between them although they are different. The connection, configuration and use of the devices will be simple and fast. HAVi APIs are able to automatically detect devices on the network and coordinate the functions of several devices connected to that network. The connection between the devices is via an IEEE 1394 digital network. These devices are HAVi at all levels and there is none that controls the rest, everyone can control everyone.

B. Jini

Jini [21] is a technology developed by Sun Microsystems. Its aim is to enable users to share services and resources in the network. It provides easy access to these resources from anywhere on the network, even if they switch places. It simplifies as much as possible the tasks of deploying, maintaining and managing the system of both devices and users. The services offered by objects that are part of the system can be used by any user/object connected to it. Services can be actions taken by both devices (hardware), software or combinations of both. They communicate with each other using a service protocol and a set of interfaces in Java. Jini supports any type of physical media: IEEE 1394, Bluetooth, IrDA, Ethernet, regardless of the operating system and hardware supported.

C. DHCompliant Protocol

It is an integration protocol that enables communication between digital home devices and existing robots [4]. It also provides a gateway to other protocols that can be found in a smart home environment. DHCompliant provides several modules identified by diverse services. These services allow the control of different aspects that a home automation can offer to the users:

- Allows users to schedule robot tasks
- Collaboration control tasks between the devices

- Design of rules that must be constantly checked by the devices, to meet the wishes of the user.
- Establish a security system that ensures confidentiality.
- Controlling the movements of the robots with a location system.

IV. METHODOLOGY

This section describes a number of features the adapter has to meet, the tools used in the development and testing, the structure of the experiment using a SysML block diagrams, software structure of the adapter and a brief description of the operation of the adapter.

A. Adapter features

The following TABLE III shows the features that must follow the adapter developed.

TABLE III .Adapter Features

FEATURE	DESCRIPTION
Supports UPnP and DHCCompliant	The robot using the adapter will be compatible with the UPnP and DHCCompliant protocol.
Generic	It will allow reuse.
Extensible	It will allow the implementation of new functionalities.
Abstract	Using UPnP and DHCCompliant protocols transparently to user.
Open Source	Open source to new developers.
Low consumption	Cpu and memory consumption should be low.
Fast processing	Fast execution time.

B. Tools

- The project has been developed with the Java programming language [22].
- The Intel Software for UPnP Technology [23] toolkit has been used to develop an UPnP device by using the Java programming language [22]. Within this set of tools, it also has been used the following ones:
 - Service Author (Intel Tools for UPnP Technologies): To create the XML file that describes the available services from the robot in an UPnP network.
 - Device Spy (Intel Tools for UPnP Technologies): To scan the available devices in the UPnP network. It was also used for testing the robot when performing different service invocations.
 - Device Builder (Intel Digital Home Device Code Wizard): To auto-generate the skeleton of the source code for creating UPnP devices in the Java language.
- To develop the UPnP control point it has been used the Cling library [24].
- The management of the robot was performed using the programming API ARIA [25].

In the DHCCompliant project official Web site [26] the

released version of the developed adapter can be downloaded.

C. Experiment Structure

The following SysML diagram (Fig. 2) show the structure of the experiment, where it can be noticed:

- Pioneer P3DX Block: the robot used for video surveillance tasks.
- Block Adapter DHC Pioneer P3DX: is the adapter developed in Java that is connected to the digital home network.
- Computer Onboard: is the computer included inside the robot which runs the adapter.

Finally, the devices needed to run DHCCompliant and the Wifi router to form a compatible UPnP network.

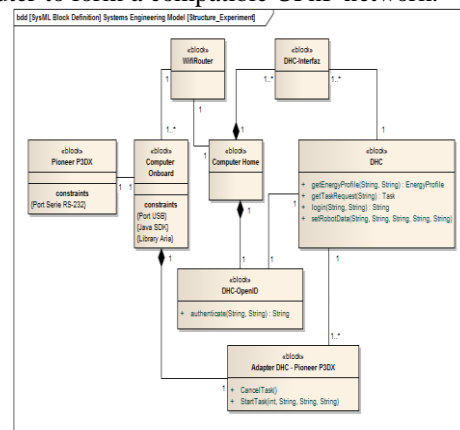


Fig. 2. System Structure for the Experiment in SysML

D. Adapter Structure

The adapter is structured in a series of packages, a short description can be seen below:

- **“Control Point” package:** Contains the whole logic to create an UPnP control point. Serves to discover and to subscribe to the services of the DHC devices and the DHC-OpenID.
- **“Device” package:** Contains the source code to publish the available services of the robot in an UPnP network. The main published services are:
 - Forward and rotation movements.
 - Start task: is the scheduled security task carried out by the robot.
 - Cancel task: cancels the active task.
- **“Pioneer” package:** Is the layer where it is implemented the whole robot control system.
- **“Useful” and “properties” package:** These are the packages where the functions to support the development and the configuration parameters are implemented.

E. Running the Adapter

The figure below (Fig. 3) shows a diagram representing in a SysML diagram of activity the overall implementation of the adapter. When the adapter is started, three independent processes will be launched. On the one hand, the adapter initiates the connection to the robot, on the

other side; it starts the module making the robot into a compatible UPnP device. Finally, the most important task is started, making the robot module compatible with the protocol and connecting the DHCompliant with the central devices of DHC.

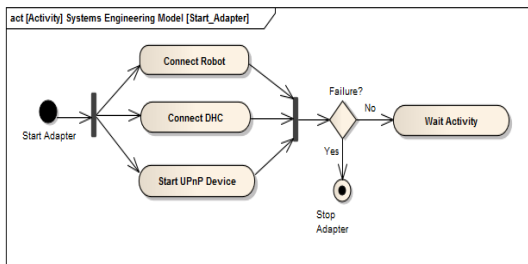


Fig. 3. Activity Diagram - Start Adapter

The connection with DHC devices are described in more detail in the following SysML activity diagram (Fig. 4). The first process that makes the adapter is to find the devices DHC and DHC-OpenID. If it finds them, it subscribes to them. Once subscribed to the service it authenticates the device authentication DHC-OpenID. After the authentication in the DHC-OpenID service, it is validated in a security service DHC device with the key obtained from the DHC-OpenID service. After being validated, the authentication services from both devices subscribe to other DHC services and begin to monitor the public state variables. When the adapter captures a change in the state variables for the available tasks and notes a safety task (Colaboration_Keyword = Security) asked the device DHC, DHC accepts the request when the adapter sends the command to the robot so that it can run. It will be running until the robot ends or the device itself DHC stops the orders and the order captures the adapter observing the state variables. In the other adapter it will be watching to see if they will be valid to perform new tasks in the future.

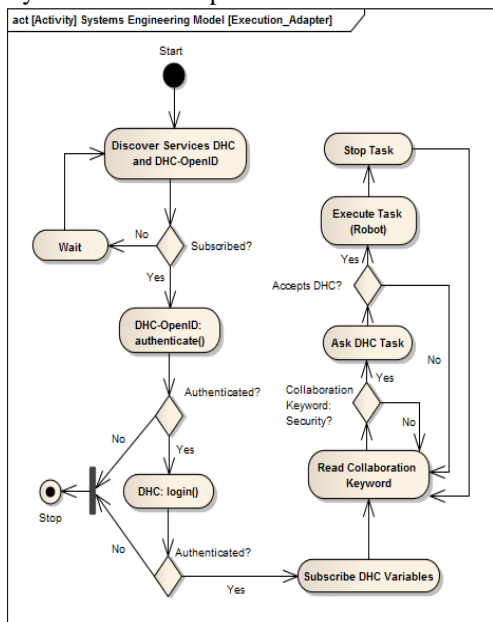


Fig. 4. Sysml Activity Diagram of the Adapter Execution

V. EXPERIMENTS

This section describes the experiments that were carried out to demonstrate the integration of a robot surveillance using the adapter in a digital home developed by the DHCompliant protocol. The experiments were focused on demonstrating the operation of the adapter and the compatibility of the UPnP architecture and DHCompliant protocol. We have also conducted performance tests measuring the time spent in each operation and its CPU and memory consumption.

For testing we have used the following hardware and software:

- Pioneer Robot P3DX onboard computer connected to a USB-Serial cable.
- The robot's onboard computer is a laptop with an Intel Celeron E3200 at 2.40GHz, 4GB of RAM and Windows Vista 64bit. Adapter development environment NetNeans v6.9.1.
- Devices DHC and DHC-OpenID run from Visual Studio 2008 on a computer with an AMD Athlon 64 X2 Dual 4000 + to 2.11 GHz, 3GB of RAM and Windows XP operating system.
- As a communications router between the two teams with a Linksys WRT54GL.
- Your Kit Java Profiler v9.5.4 to measure the CPU and memory adapter.

In addition, tests have been conducted in an ideal environment, assuming that the separate elements of the power unit is working properly (router, DHC and DHC-devices OpenID). The results will display any possible failures that may occur during the performance from these devices.

A. Connecting the adapter to the robot.

In this experiment (Fig. 5) it has been shown that the adapter is able to connect to the robot and it has been measured the average time it takes.

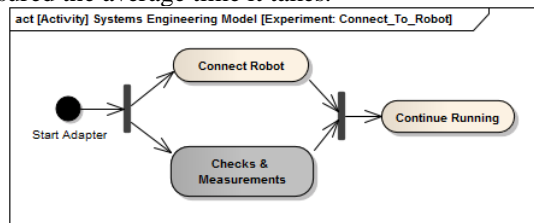


Fig. 5. Experiment 1: Connecting the adapter to the robot.

B. Upnp Service Publication of the Robot by the Adapter.

In this experiment (Fig. 6) it has been found that the adapter publishes all UPnP services developed for the robot and makes valid the UPnP device. The tests that were carried out were: to check how long it takes the adapter to be an UPnP device and the valid performance tests of some of the available UPnP services.

UPnP services available:

- **Connection service:** For connecting / disconnecting the robot.

- **DHCGroups Service:** Service for the DHCompliant protocol. In this service the tasks can be activated / cancelled.
- **Energy Service:** Energy Management Service of the robot. The energy profiles of the robot can be changed.
- **Movement Control Service:** For controlling the robot movement.
- **Sonar Service:** To use the robot sonar.

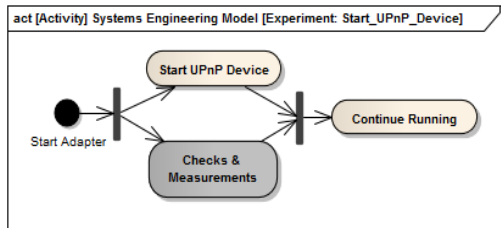


Fig. 6. Experiment 2: Adapter publishes the UPnP service of the robot

C. Discovery of the Devices DHC Openid and Dhcompliant by the Adapter.

In the experiment described in the following diagram (Fig. 7), it was found that the adapter is able to discover devices and DHC DHC-OpenID, and can validate the security of both devices, a prerequisite for being able to interact with Home automation and robotic elements of the digital home. In addition to verify that the adapter is connected to services and validates them, the average time it takes to make the whole process has been measured.

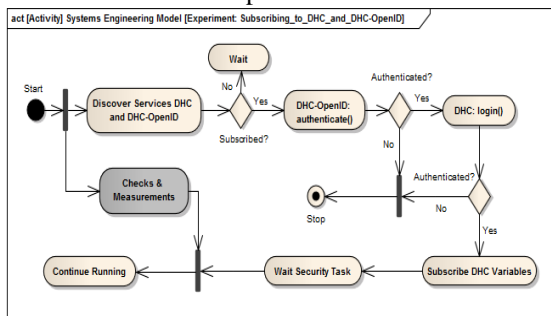


Fig. 7. Experiment 3: Adapter Discovers and Connects To the DHC and Dhcompliant Openid Services

D. Petition and Execution of a Security Task

This experiment (Fig. 8) has been to send a security task from the interface of the DHC to run the robot. For the robot to perform the task, it must recognize the services published by DHC, then the adapter must make a request to the device DHC suggesting that it is interested in running the task. DHC will see this request and confirm the robot for execution. Once confirmed by the DHC device, the adapter will observe that the task can be performed and it will send it to the robot to carry it out. The scheduled security task is a simple task where the robot only has to move up to a point and return to the starting point (Fig. 9). Thus, the robot performs the surveillance of the delimited area. The extent to where the robot must move is indicated

in the definition of the task set from the DHC interface. In this experiment, apart from checking that the robot has performed the task, the average time the robot takes to start the task was also measured.

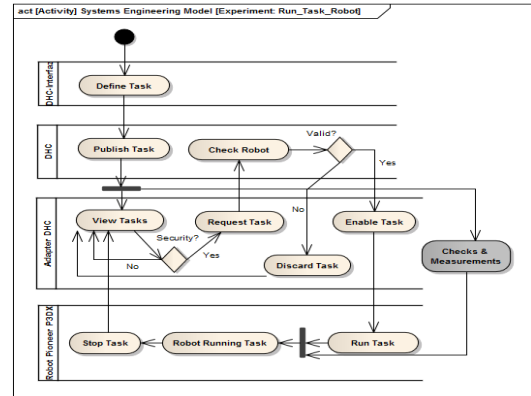


Fig. 8. Experiment 4: Request and Implementation of a Security Task



Fig. 9. Robot Movement during Security Task Experiment

E. Cancelling a Safety Active Task

In this experiment (Fig. 10), it has been checked that the adapter is able to detect the cancellation of the task that was active when the DHC system orders it. Measures have also been made in order to calculate the average time it takes the adapter to cancel the task.

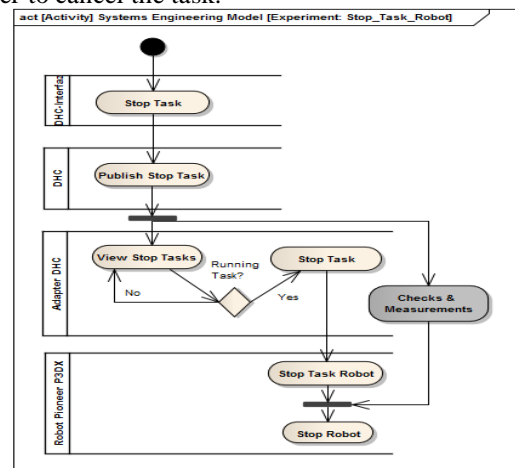


Fig. 10. Experiment 5: Cancellation of an Active Security Task

F. Monitoring CPU and Memory Consumption in the Execution of a Task.

In this last experiment (Fig. 11) it has been measured the CPU and memory consumed in the process leading from when the adapter is started until the robot executes a task and this one is canceled.

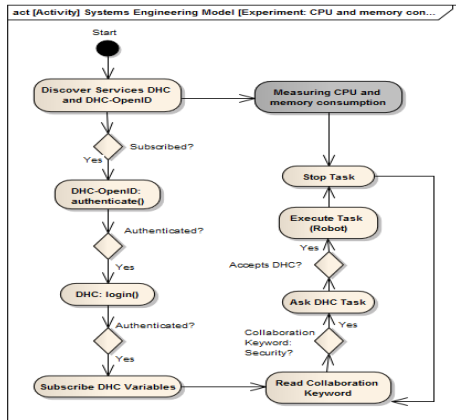


Fig. 11. Experiment 6: Measuring CPU and Memory Consumption

VI. RESULTS

This section shows the results of different experiments conducted to demonstrate the correct operation of the adapter.

A. Connecting the Adapter to the Robot

The results are shown in the following TABLE IV.

TABLE IV. RESULT CONNECTING THE ADAPTER TO THE ROBOT

N° Executions	Retrieved Result	Mean Time (milliseconds)	Errors
25	The adapter is connected correctly to the robot.	2877,67	There has been no error

B. Upnp Services Publication of the Robot by the Adapter

The results are shown in the following TABLE V.

TABLE V. RESULT UPNP SERVICES PUBLICATION OF THE ROBOT BY THE ADAPTER

N° Executions	Retrieved Result	Mean Time (milliseconds)	Errors
25	The adapter publishes all UPnP service robot.	5215,79	There has been no error

In the figure below (Fig. 12) the UPnP services that publish the adapter on the robot can be observed.



Fig. 12. Capture of the Robot Upnp Services

Besides, all the service methods have been performed to verify their proper operation. The result of the executions of two methods selected at random is shown below. Results of the method "getSonarRange ()" of the "Sonar" service are shown in the TABLE VI.

TABLE VI. RESULT INVOCATION METHOD "GETSONARRANGE()"

N° Executions	Retrieved Result	Mean Time (milliseconds)	Errors
25	The distance to the nearest object with a maximum distance of 5 meters is shown.	16,56	There has been no error

An example of the invocation from the program "Device Spy" is shown in the figure below (Fig. 13).

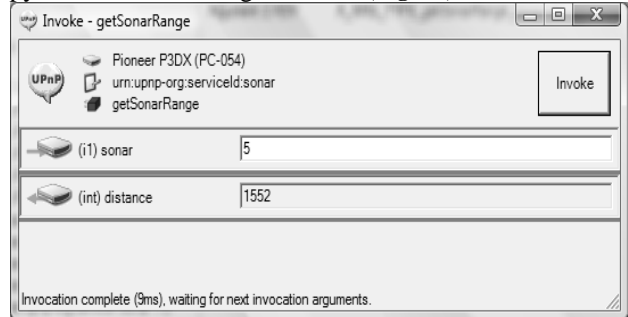


Fig. 13. Capture Of The Invocation By The Method Getsonarrange ()

The result of the method "is Connected ()" of the "Connection" service are shown the TABLE VII.

TABLE VII. RESULT INVOCATION METHOD "IS CONNECTED ()"

N° Executions	Retrieved Result	Mean Time (milliseconds)	Errors
25	It is warned if the adapter is connected to the robot. It was always connected.	12,48	There has been no error

C. Discovery Of The DHC-Openid and Dhcompliant Devices by the Adapter

The results are shown in the following TABLE VIII.

TABLE VIII. RESULT DISCOVERY OF THE DHC-OPENID AND DHCOMPLIANT DEVICES BY THE ADAPTER

N° Executions	Retrieved Result	Mean Time (millisecond s)	Errors
25	The adapter discovers the DHC and DHC-OpenID devices. It has been subscribed to the authentication services and has been validated by them. Later it has been subscribed to the rest of the DHC services.	3680,79	There has been no error

The figure below (Fig. 14) shows a capture of the execution log of the adapter, where it can be seen how the adapter originally discovers the DHC and DHC-OpenID devices, it subscribes to the login and authentication services, and it is authenticated and logged in each one. Subsequently, it subscribes to the rest of DHC services and waits by reading the state variables waiting future tasks to be executed.

```

Connected to robot.
Name: Hovi_2887
Type: Pioneer
Subtype: p3dx-eb
Device found -> OpenID: urn:schemas-upnp-org:device:DHCOpenIDProvider:1
Service discovered: (RemoteService) ServiceId: urn:upnp-org:serviceId:AuthenticateService
OpenID -> Established: uuid:2258b406-2649-4369-b239-948352a20c47-urn:upnp-org:serviceId:AuthenticateService-30
OpenID -> authenticate - return: http://localhost:4660/user.aspx/Pioneer0-0000-0000-0000-00000000000000000000
Device found -> GUIDHC: urn:schemas-upnp-org:device:GUI:1
Device found -> DHC: urn:schemas-upnp-org:device:DHC:1
Service discovered: (RemoteService) ServiceId: urn:upnp-org:serviceId:DHCSecurity
ActRobotPacketReceiver::receivePacket: bad packet, bad checksum
DHCSecurity -> Established: uuid:3a511de2-7af3-4650-4e93-9eb5331ca1d1-urn:upnp-org:serviceId:DHCSecurity-7
Pioneer -> Profile Energy: LOW
DHCSecurity -> Login - return: keyprueba
Service discovered: (RemoteService) ServiceId: urn:upnp-org:serviceId:DHCEnergy
Service discovered: (RemoteService) ServiceId: urn:upnp-org:serviceId:DHCGroups
Service discovered: (RemoteService) ServiceId: urn:upnp-org:serviceId:DHCLocalization
Service discovered: (RemoteService) ServiceId: urn:upnp-org:serviceId:DHCRules
DHCEnergy -> Established: uuid:4556a98a-44fc-4e9a-b109-737b0e66e1a-urn:upnp-org:serviceId:DHCEnergy-3
DHCLocalization -> Established: uuid:4556a98a-44fc-4e9a-b109-737b0e66e1a-urn:upnp-org:serviceId:DHCLocalization-1
DHCRules -> Established: uuid:4556a98a-44fc-4e9a-b109-737b0e66e1a-urn:upnp-org:serviceId:DHCRules-3
DHCGroups -> Established: uuid:4556a98a-44fc-4e9a-b109-737b0e66e1a-urn:upnp-org:serviceId:DHCGroups-3
DHCGroups -> Event: 0
Collaboration_Keyword: Wait
    
```

Fig. 14. Capture of the Adapter Log Discovering the DHC and DHC-Openid Services

D. Petition and Execution of a Security Task

The results are shown in the following TABLE IX.

TABLE IX .RESULT PETITION AND EXECUTION OF A SECURITY TASK

Nº Execution s	Retrieved Result	Mean Time (milliseconds)	Errors
25	In 20 runs the adapter has received the tasks correctly and has subscribed the robot for performing the task. DHC has accepted the request and the adapter has ordered the robot to run it. The robot has executed it successfully.	5422,63	In two runs the adapter has received incorrect task parameters. There were default values set in DHC. The error had occurred in the execution of the DHC device which published incorrect tasks.

In the figure below (Fig. 15) it is shown an execution log of the adapter with the parameters of the task being performed and the messages that are sent to the robot to start the task and the subsequent movements.

```

Collaboration Keyword: Security
CancelTask: 0
TaskID: Prueba_20
DHCGroups - GetTaskRequest -> destinationFloor - return: 1
DHCGroups - GetTaskRequest -> destinationRoom - return: 1
DHCGroups - GetTaskRequest -> taskPointCoordinates - return: 100;0
DHCGroups - GetTaskRequest -> escapePointCoordinates - return: 0;0;0
DHCGroups - GetTaskRequest -> SemanticInformation - return: none
DHCGroups - GetTaskRequest -> CollaborationKeyword - return: Security
DHCGroups - GetTaskRequest -> Hive - return: 1.0
DHCGroups - GetTaskRequest -> TimeTask - return: 2011-05-31T12:00:00
DHCGroups - GetTaskRequest -> Workspace - return: 1.0
DHCGroups - GetTaskRequest -> TimeToPerform - return: 2011-05-31T13:00:00
DHCGroups -> Missed events: 7
DHCGroups -> Event: 10
DHCEnergy -> EnergyProfile - return: HIGH
DHCEnergy -> Downtime - return: 0
Pioneer -> Change energy profile to: HIGH
Start the Safety Task: Prueba_20
DHCGroups -> Event: 13
Pioneer -> Start Task
Pioneer -> Energy Profile: HIGH
Pioneer -> Distance to cover: 100.0
Pioneer -> Rotation angle: 180.0
Pioneer -> Distance to cover: 101.0
    
```

Fig. 15. Capture of the Adapter Log Discovering the Security Task

E. Cancelling a Security Active Task

The results are shown in the following TABLE X.

TABLE X .RESULT CANCELLING A SECURITY ACTIVE TASK

Nº Executions	Retrieved Result	Mean Time (milliseconds)	Errors
25	The adapter has recognized that the task was carried out by the robot and told it to stop.	1ms to recognize that it must cancel the task. The time to completely stop the robot depends on where it was at that time, because it stops at the initial point of departure.	There has been no error

Below a screenshot is shown (Fig. 16) with the adapter log when it recognizes that the task must be cancelled and its subsequent cancellation.

```

TaskID: Prueba_20
Collaboration_Keyword: Security
CancelTask: 1
Cancel the task: Prueba_20
Pioneer -> Cancel the task
Pioneer -> Rotation angle: 180.86805143692197
Pioneer -> Distance to cover: 101.00495037373169
Pioneer -> Finished the task
Pioneer -> Change energy profile to: LOW
    
```

Fig. 16. Capture of the Log Adapter When the Task Is Cancelled

F. Monitoring CPU and Memory Consumption in the Execution of a Task

The figure (Fig. 17) shows a peak of 39% of CPU consumption. This is the moment when the adapter is connected to the robot and is also initializing UPnP services. Later, the consumption is minimal until the moment when consumption rises at 4%, that is the moment where a security task is received and the order is sent to the robot. While executing the task until the end and while closing the adapter, the consumption is close to zero.

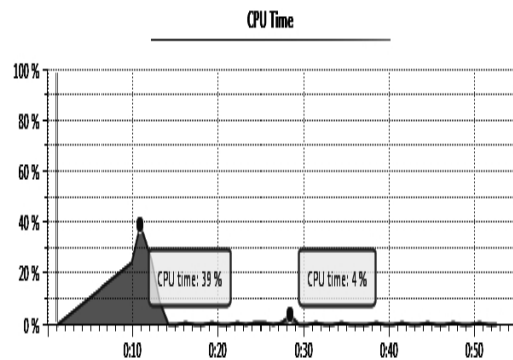


Fig. 17. CPU Consumption during the Execution of a Task

As shown in figure (Fig. 18) from the mark (A) all the class definitions and metadata have already been loaded in memory. It can be compared with the previous figure (Fig. 17) which corresponds to the moment where the CPU consumption is higher.

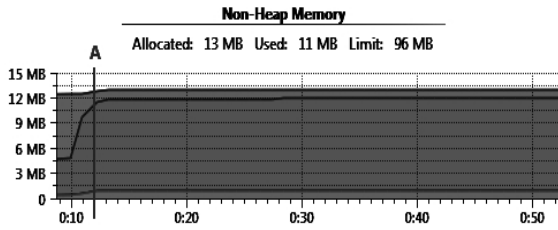


Fig. 18. Non-Heap Memory Consumption

In the figure (Fig. 19) it can be observed the Heap memory consumption. The consumption remains in a cycle, ranging from 2 megabytes to 4 megabytes, except when the adapter is initializing (interval A) and when the security task is received(interval B).

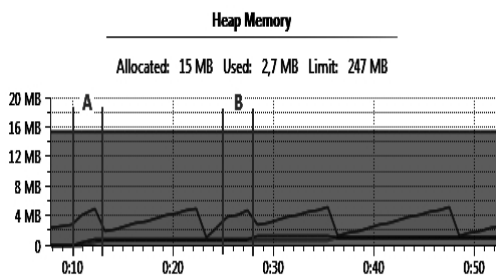


Fig. 19. Heap Memory consumption

In the figure (Fig. 20) it can be observed the number of threads loaded, every thread is loaded when the first mark is reached (mark A). When running the security task a new thread is started (mark B). In the last mark (mark C) the execution of the thread that was performing the security task is finished.

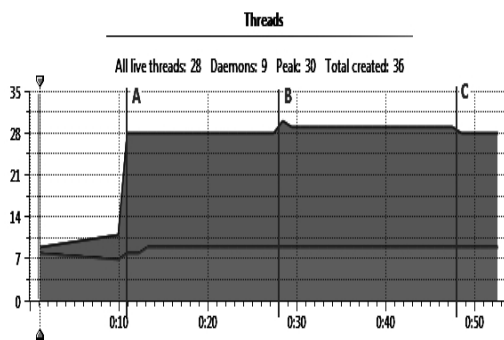


Fig. 20. Execution Threads

G. Possible Failures Independently Of the Adapter

Previous experiments have been conducted under ideal conditions, where all devices alien to the adapter were working correctly and even so, it have been seen that the services of the DHC device have failed in two runs. In the following TABLE XI, there are shown some of the possible failures that may affect the adapter and it is indicated how the adapter has responded to these failures. Apart from these shortcomings described, there are many more failures. Some examples are: robot is without power, the computer of the robot broke down or there are obstacles in the robot navigation, among others.

TABLE XI. FAULT POSSIBLE EXPERIMENTS

Device and Services in which the Failure Occurs	Failure Time	Answer of the Adapter / Robot
Router failure (no Wifi failure)	The adapter is not connected to the DHC-OpenID, or to DHC.	The adapter is connected to the robot, but not to the services DHC and DHC-OpenID.
Router failure (no Wifi failure)	The adapter is already validated in DHC and DHC-OpenID, and subscribed to all services. Currently it is not running any task.	The adapter has lost its connection with the services of DHC and could not receive new tasks.
Router failure (no Wifi failure)	The robot is running a security task.	The adapter could not get the cancellation of the task and only will end it if it was scheduled to be ended at a given time.
Failure on DHC and DHC-OpenID services	The adapter is not validated in the login and authentication services of DHC and DHC-OpenID.	The adapter is connected to the robot, but it failed to connect to the DHC-OpenID and DHC services.
Failure in DHC services, but not in DHC-Security	Adapter is not connected to any dispositive.	The adapter is connected and validated on DHC and DHC-OpenID-Security, but it was not able to subscribe to the rest of DHC services. The adapter cannot receive any task.
Failure in DHC services, but not in DHC-Security	The robot is performing the security task.	The adapter could not receive the cancellation of the task and can only end if it was scheduled for being ended.
Failure on DHC-OpenID	Before you login to the service.	The adapter failed to validate any device.

VI. CONCLUSION

This experiment has succeeded in developing an adapter for a surveillance robot (Pioneer P3DX) compatible with the UPnP technology and the DHCompliant protocol. The adapter has been developed with Open Source tools and is free published on the DHCompliant project website [26]. Using the adapter, the robot is capable of maintaining a high interoperability in a transparent way with the other automation elements and service robots integrated in the digital home thanks to the DHCompliant protocol. In the results it can be seen the proper operation of the adapter that allows the DHC system to send tasks to the robot. The adapter receives and transmits the tasks in a short period of

time and consumes little resources (CPU and memory) of the system in which it is embedded. Furthermore, it can be noticed that the robot, apart from being managed by the DHCompliant system, can be also handled through the UPnP services, as it becomes an UPnP compatible device. In future works we pretend to use this adapter as a basis for the development of future adapters for any type of service robot. We pretend to expand the functionality of the adapter by adding more features of the DHCompliant protocol, as encryption in the exchange of messages, the use of location and energy services. All the updates that will be added to the adapter will be posted on the DHCompliant [26] website as new versions.

ACKNOWLEDGMENT

We must acknowledge the continuous support given by the University of Oviedo, and also for providing all the resources that made possible our research. This paper was made under the grants DH Compliant (MITC-09-TSI-020100-2009-359) and DH Compliant 2 (MITC-11-TSI-020100-2011-313)

REFERENCES

- [1] Statistics - IFR International Federation of Robotics». [Online]. Available: <http://www.ifr.org/service-robots/statistics/>. [Accessed: 02-Jun-2011].
- [2] UPnP Forum, «UPnP Forum». [Online]. Available: <http://www.upnp.org/>. [Accessed: 28-Feb-2011].
- [3] I. González Alonso, M. R. Fernández Alcalá, J. Maestre Torreblanca, and M. P. A. García Fuente, Service robotics within the Digital Home: Applications and future Prospects, 1o ed. Springer, 2011.
- [4] González Alonso I, Álvarez Fres O, Alonso Fernández A, Gómez del Torno P, Maestre JM, García Fuente MDPA. Towards a new open communication standard between homes and service robots, the DHCompliant case. Robotics and Autonomous Systems. 2012. Volume 60, Issue 6 pages 889-900
- [5] MoviRobotics S.L., «Movirobotics. mSecurit.» [Online]. Available: <http://www.movirobotics.com/SPmsecurit.php>. [Accessed: 24-Feb-2011].
- [6] A. Fernández-Caballero, J. C. Castillo, J. Martínez-Cantos, and R. Martínez-Tomás, «Optical flow or image subtraction in human detection from infrared camera on mobile robot», Robotics and Autonomous Systems, vol. 58, n^o. 12, págs. 1273-1281, Dic. 2010.
- [7] MobileRobots, «MobileRobots Pioneer 3-DX (P3DX) differential drive robot». [Online]. Available: <http://www.mobilerobots.com/researchrobots/researchrobots/pioneer3dx.aspx>. [Accessed: 24-Feb-2011].
- [8] MobileRobots, «MobileRobots PatrolBot research robot platform». [Online]. Available: <http://www.mobilerobots.com/ResearchRobots/ResearchRobots/ResearchPatrolBot.aspx>. [Accessed: 24-Feb-2011].
- [9] T. Theodoridis y H. Hu, «Intelligent Security Robots: A survey», 2009.
- [10] Wifibot Networked Robotics, «Wifibot Networked Robotics». [Online]. Available: <http://www.wifibot.com/wifibot-oldproducts.html>. [Accessed: 28-Feb-2011].
- [11] L. Payá, A. Gil, O. Reinoso, M. Juliá, L. Riera, and L. M. Jiménez, «Distributed platform for the control of the WiFiBot robot through Internet», in ACE 2006. 7th IFAC Symposium on Advances in Control Education, 2006.
- [12] R. Severino and M. Alves, «On a Test-bed Application for the ART-WiSe Framework», IPP-HURRAY Technical Report, HURRAY-TR-0601103, 2006.
- [13] A. Treptow, G. Cielniak, and T. Duckett, «Active people recognition using thermal and grey images on a mobile security robot», in Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on, 2005, págs. 2103-2108.
- [14] Zeng DehuaI, Xu Gang, Zhong Jinming, and Li Li, «Development of a Mobile Platform for Security Robot», in Automation and Logistics, 2007 IEEE International Conference on, 2007, págs. 1262-1267.
- [15] J.N.K. Liu, Meng Wang, and Bo Feng, «iBotGuard: an Internet-based Intelligent Robot security system using Invariant Face Recognition against intruder», Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 35, n^o. 1, págs. 97-105, 2005.
- [16] R. C. Luo, T. Y. Lin, and K. L. Su, «Multisensor based security robot system for intelligent building», Robotics and Autonomous Systems, vol. 57, n^o. 3, págs. 330-338, Mar. 2009.
- [17] R. C. Luo, P. K. Wang, Y. F. Tseng, and T. Y. Lin, «Navigation and Mobile Security System of Home Security Robot», in Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on, 2006, vol. 1, págs. 169-174.
- [18] Y.-G. Kim, H.-K. Kim, S.-G. Lee, and K.-D. Lee, «Ubiquitous Home Security Robot Based on Sensor Network», in Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology, Washington, DC, USA, 2006, págs. 700-704.
- [19] I. G. Alonso, M. R. F. Alcala, and M. P. A. G. Fuente, «Interoperability Standard Used by Service Robots: Study to Develop an Open Standard By Infobotica Research Group», in Systems (ICONS), 2010 Fifth International Conference on, 2010, págs. 18-23.
- [20] R. Lea, S. Gibbs, A. Dara-Abrams, and E. Eytchison, «Networking home entertainment devices with HAVi», vol. 33, n^o. 9, págs. 35-43, Sep. 2000.
- [21] K. Arnold, R. Scheifler, J. Waldo, B. O'Sullivan, and A. Wollrath, Jini Specification, 1o ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.
- [22] J. Gosling, B. Joy, G. Steele, and G. Bracha, Java (TM) Language Specification, The (Java (Addison-Wesley)). Addison-Wesley Professional, 2005.
- [23] Intel Corporation, «Intel® Software for UPnP* Technology: Download Tools - Intel® Software Network», 24-Feb-2011.



ISSN: 2277-3754

ISO 9001:2008 Certified

International Journal of Engineering and Innovative Technology (IJET)

Volume 2, Issue 3, September 2012

[Online]. Available: <http://software.intel.com/en-us/articles/intel-software-for-upnp-technology-download-tools/>. [Accessed: 24-Feb-2011].

[24] Cling - Java/Android UPnP library and tools». [Online]. Available: <http://teleal.org/projects/cling/>. [Accessed: 24-Feb-2011].

[25] MobileRobots, «ARIA - MobileRobots Research and Academic Customer Support». [Online]. Available: <http://robots.mobilerobots.com/wiki/ARIA>. [Accessed: 24-Feb-2011].

[26] DHCompliant Consortium, «Downloads - Dhcompliant Web Site». [Online]. Available: <http://www.dhcompliant.com/index.php/page/10.html>. [Accessed: 20-Jun-2011].

AUTHOR'S PROFILE



Mr. Víctor García Sánchez is a B.Sc. in Computer Science by the University of Oviedo. He obtained the M. Sc. degree on Web Engineering in the University of Oviedo (2010). Nowadays he's working in Infobótica Research Group as technical staff. This research group belongs to the University of Oviedo and their main research field is the developing of robotic software.



Dr. Ignacio González Alonso is an associated professor at the University of Oviedo since 2005. He also takes part in Infobotica Research Group at University of Oviedo, developing projects in computer and robotics fields. He is collaborating in several national funded projects, for example TIC4Bot and DHCompliant. He also is the coauthor of twelve international contributions. He has a publication in the Inderscience publishers of distinguished academic, scientific and professional journals, and he also is going to be the co-author of the book "Service robotics within the Digital Home: Applications and future Prospects" by Springer-Verlang.