Architecture based on Distributed Webservices Integrated Through Triggers Associated with Managed Resource States

Márcio André Da Costa Alencar
Tecnologia em Redes de Computadores
Centro Universitário do Norte, Laureate UNINORTE
Manaus-AM, Brazil

Walter Charles Souza Seifert Simões, Vandermi João Da Silva, Vicente Ferreira De Lucena Junior
Instituto De Computação, ICOMP
Universidade Federal Do Amazonas, UFAM
Manaus-AM, Brazil

***Abstract*— Research on smart homes show that more and more electronic devices are being linked to each other that allows you to program commands and settings those devices as a unique system. However, many electronic device manufacturers have not yet reached a standard that allows this integration much simpler. Therefore, this paper proposes the construction of a software layer for communication with electronic devices and independent automation systems. The methodology used was to build middle-tier software (middleware) based in web architecture to send HyperText Transfer Protocol (HTTP) requests, associated to resource state (e.g.: lamp on/off), to communicate with others devices used in the SmartHome. Results showed that can integrate devices to any, local or remotely via cloud, system able to address HTTP requests. The main contribution of this paper is to present a software architecture of electronic devices and can be used to enhance human interaction and customization of how the devices will interact with other systems.**

Keywords—Smarthomes; Distributed; Webservices; Triggers; Integrated

I. INTRODUCTION

Currently the society has advanced significantly in the areas of consumer electronics through the concept of smart homes that integrates security systems, entertainment and comfort. [1] and [2] smart homes are a new approach to offer comfort, improving the quality of life in a self-configurable and intelligent environment. In Bhagyalakshmi [3] the authors concluded that the popularity of remote control systems for homes achieved significant growth due to the availability and simplicity of the internet. However, even with expansion of internet, access facility and variety solutions, the bad experience of users, incompatibility between devices and absence of standard are features which smart homes haven't become too accepted [4].

This paper presents an architecture for smart homes based on independent systems able to receive and send HyperText Tranfer Protocol (HTTP) requests to control a specific feature of the residence, such as lamps, doors, windows, sensors and others. These systems already have an interface, which enable control of resources through these requests and they can receiving, processing and responding to the demands generated.

In this scenario, we sought to explore the interaction of resources via HTTP requests so that it was possible to give more flexibility and speed in setting up as independent automation systems interact and what actions will be taken by each resource according to their status feature. In Section III is detailed as this integration will occur.

II. RELATED WORKS

The study made by [5], presents a ARM9 device, 16/32 bits with a embedded system connected to an Ethernet network and offers to user a friendly web interface, through it's possible control all peripherals devices (cameras, relays, sensors and others) directly connected to this device. Despite being effective to simultaneously manage multiple devices a fault in the microprocessor makes the system inaccessible to the user.

A research conducted by [6] proposes a flexible, autonomous and low-cost system to the smart home. As a differential, this solution presented a security system capable of sending an email to the user when the alarm is triggered, however, also limited to this single interaction with an external system.

Lin and colleagues [4] proposed a standard for intelligent homes through the integration of home automation systems with intelligent routers. In this scenario the home automation is built over a Wi-Fi (Wireless Fidelity) network that adds the functionality of managing the resources of the residence and offers a basic set of services to local network connectivity. As a limiting factor, this proposal requires router with unconventional resources such as the possibility of adding hardware (peripheral components) and software (management interface for peripherals) which raising the cost of the solution.

El Shafee and Hamed [7] presented a paper in home automation which also uses Wi-Fi networks and consists of two parts: the first is a web server that manages, controls and monitors the residence and the second is a hardware module with an interface providing appropriate mechanisms for communication between sensors, actuators and server. Although it has a well structured system dependency hardware interface modules in relation to the Web server exposes the entire asset management system and also fails if the server is not available.

Related work describes solutions that use web server embedded in controllers able to connect to a local network, whether wireless or not. Its centralized architectures limit their interoperability with other systems since only the central unit can control resources. Beyond this limitation, the centralization presented as critical point of the system since, in case of failure, disables the management of resources house.

The proposed architecture defines that each device acts as a standalone webserver managing for a home resource. Such devices are dependent solely on their connectivity to the local network where the failure of one of the automation system does not prevent the execution of the activities of others. In addition, the resource management over the internet is facilitated by redirection of external requests received by router to the desired resource.

III. THE ARCHITECTURE

Fig. 1 illustrates the proposed architecture in this work which is comprised of: one component responsible for managing a LAN (network controller) and independent automation systems (actuators and sensor devices). The devices, when connected to the controller network, act as web servers performing the treatment of HTTP requests and share the same network with other nodes such as desktops, notebooks and smartphones.

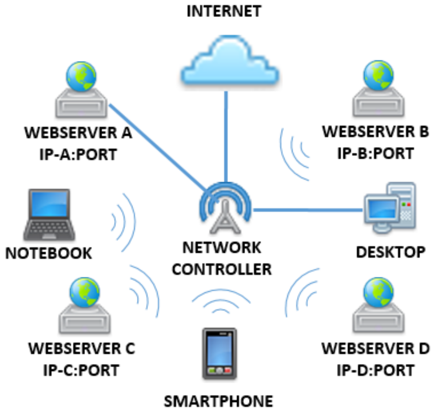


Fig 1. Main Scenario.

Although physically are arranged in a star topology, where a network controller is responsible to connect the nodes of the network, the system are independent of each other.

Each device is responsible for managing a specific feature of the residence (light, motion sensor, rain sensor, alarm etc.) and not dependent on another device to perform its activities. These devices have Internet Protocol, IP and define a specific port to receive HTTP requests. These requests are interpreted and enforced in accordance with the device interface specifications and respond to the node that generated the request. These responses were in HTTP-Response format following the Request for Comments (RFC) 7231 specifications, defined by the Internet Engineering Task Force (IETF).

Each residence resource has one or more states, for example, on or off lamp. The change of resource state might or might not generate new HTTP requests, depending on how the user wants to occur interaction between devices. Such triggers are logical routines that perform HTTP requests according to device state. This requests should be perform to different devices of their origin and are defined by the user through the automation system interface that manages the resource. An event may generate trigger activation sequence in different devices as illustrated in Fig. 2.

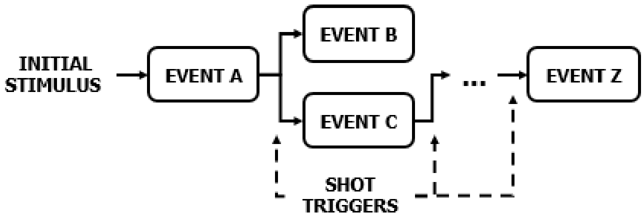


Fig 2. Event chain generated by triggers

This sequence of events is defined as a chain of events, where a start stimulus (Event A) shoots a trigger, generating a request to another device(s) and so on. This event chain extends until one of the devices does not have a trigger (Event Z) to be shot. This behavior can be best enjoyed in Item 5.

IV. DEVELOPMENT

For running tests were developed three prototypes with different features: lighting system, motion sensor and messenger system. Each system has different characteristics, but all act as web servers handling HTTP requests to control or monitor a particular resource. Each system has an IP address allowing its communication with other network devices.

In those implementations, the prototypes have been configured to receive HTTP requests on port 80 (default), but the configuration can also be made according to the application criteria using the ports above 1023.

A. *Motion Sensor*

The motion sensor (Fig. 3), reads the environment through a Passive Infrared sensor (PIR). This sensor can have two states: detected motion or no motion. This device does not allow the execution of an activity as presented in the lighting system. This component limited to collecting data through the sensor and carry out requests shot by triggers.

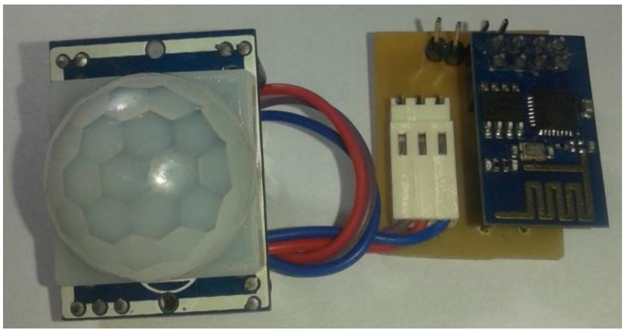


Fig 3. Lighting Control System Prototype

B. *Lighting System*

This system consists of monitoring the status of a lamp can be via relay by pressing a button or logic, through HTTP request. Both forms allow you to toggle the on and off states. An application for Android OS that allows the management of this system (Fig. 3a) was developed. In Fig. 3b you can see the prototype made of the lighting system and the manual override button.

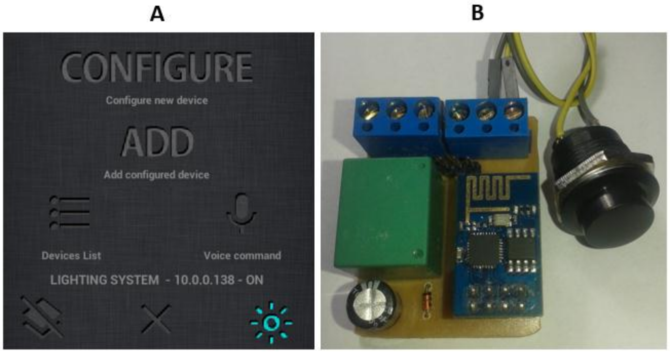


Fig 4. A- Android App. B-Lighting Control System Prototype

C. *Messenger System*

This system allows incoming requests are processed and sent to the user's mobile phone through an application to exchange online messages, the Telegram. This system has the unique feature in order to exploit the flexibility of the architecture proposed. Differentiating itself from other systems, that was implemented on a Raspberry Pi B+ (Fig. 5).

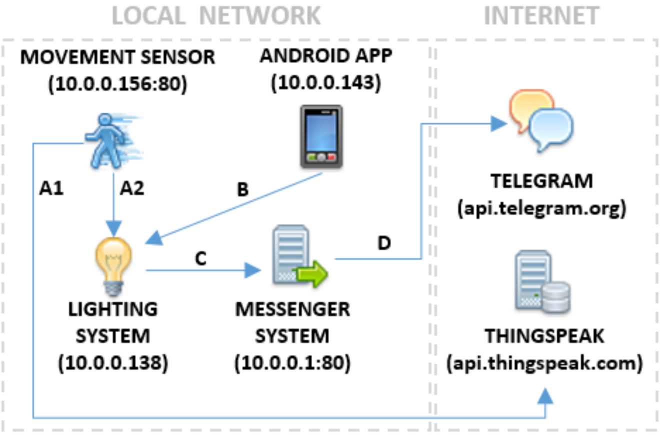
Fig 5. Raspberry pi: Model B+ (Source: https://www.raspberrypi.org/)

V. TESTS

The tests were aimed primarily to testing the interaction between the various systems through triggers, to do so, set up the devices following illustration in Fig. 5.

Each system connects to the LAN and obtained its IP address via DHCP (Dynamic Host Configuration Protocol), these IP's were linked to physical addresses of network components of each system.

In addition was explored ability to integrate with other systems on the internet such as ThingSpeak (database in the cloud that records the information generated and plots a dynamic graph of the stored values) and the Telegram (cloud-based messaging platform which provides an API for sending messages to users and groups registered)


LOCAL NETWORK
INTERNET
MOVEMENT SENSOR (10.0.0.156:80)
ANDROID APP (10.0.0.143)
A1
A2
B
C
D
LIGHTING SYSTEM (10.0.0.138)
MESSENGER SYSTEM (10.0.0.1:80)
TELEGRAM (api.telegram.org)
THINGSPEAK (api.thingspeak.com)

LEGEND

A1: Register event on Thingspeak
A2: Requests to turn lamp on/off (Sensor)
B: Request to turn lamp on/off (Android APP)
C: Request to inform Messenger when lamp changes state
D: Request to notify user via Telegram API

Fig 5. Scenario for test the triggers and systems interaction

In the presented scenario it is showed the integration between the smartphone, the lighting system and the messenger. The change of lamp status (through the app) shoots a trigger, which generates a request to the messenger, this one in turn, shoot a new trigger sending a request to the Telegram API (Application Programming Interface) notifying about the status of the lamp.

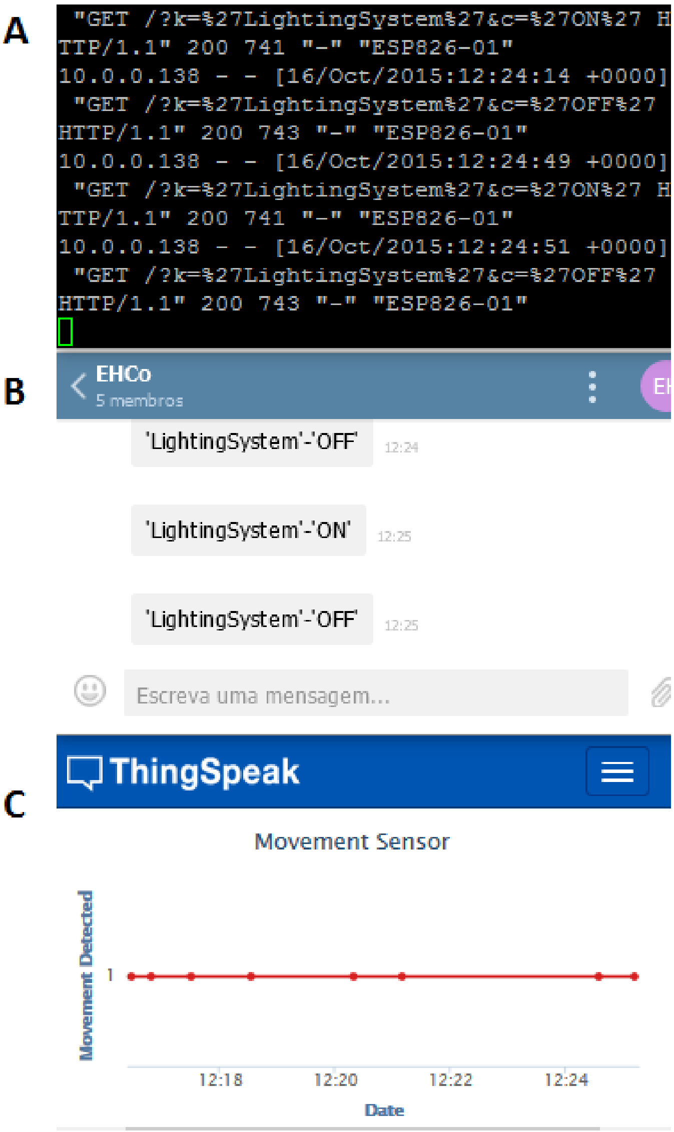
Now the motion sensor generates two requests to detect a movement: one for the lamp (to turn on lamp) and the other directly to Thingspeak (recording the time on movement detection).

VI. RESULTS

During the tests, by through monitoring the logs and analysis of data traffic in the network controller could be observed the system interaction as proposed by the architecture. The systems presented a collective behavior when your triggers are configured correctly.

Fig. 6 shows the system behavior when the lamp switched to on or off. The lamp may changes its states thought Android App (Fig. 4), manually or through trigger shot on movement sensor. This actions begins an event chain, as defined in Section III, which integrates the following systems: Motion sensor lighting system, messenger, Telegram and Thingspeak.

The detection of a movement fired a trigger, which generated a request to turn on the lamp, another trigger on motion sensor generated a request to Thingspeak (Fig 6.c), recording the movement detected in the cloud database. In addition to these, the triggers defined in the lighting system follow the behavior presented in the context of Fig 4, by shooting a trigger to the messenger (Fig. 6.a) who sends a notice to the user on his smartphone via the Telegram API(Fig. 6.b).



```
 "GET /?k=%27LightingSystem%27&c=%27ON%27 H
TTP/1.1" 200 741 "-" "ESP826-01"
10.0.0.138 - - [16/Oct/2015:12:24:14 +0000]
 "GET /?k=%27LightingSystem%27&c=%27OFF%27
HTTP/1.1" 200 743 "-" "ESP826-01"
10.0.0.138 - - [16/Oct/2015:12:24:49 +0000]
 "GET /?k=%27LightingSystem%27&c=%27ON%27 H
TTP/1.1" 200 741 "-" "ESP826-01"
10.0.0.138 - - [16/Oct/2015:12:24:51 +0000]
 "GET /?k=%27LightingSystem%27&c=%27OFF%27
HTTP/1.1" 200 743 "-" "ESP826-01"
```

EHCo
5 membros

'LightingSystem'-'OFF' 12:24

'LightingSystem'-'ON' 12:25

'LightingSystem'-'OFF' 12:25

Escreva uma mensagem...

Fig 6. A-Messenger receive HTTP request from Lighting System. B-Notification received on Telegram Account. C- Movement registered on Thingspeak.

It is important remember that URL length should be in accordance to storage limitation of source device. Not following these limitations should result in incomplete request storage and mal function on integration of independent system.

VII. CONCLUSION

Even on a small-scale test showed favorable results in the interaction of independent systems for home automation through web services. In the tests performed, an event in the presence of sensor started a chain of events generating new actions in three other independent systems (lighting,, messenger and Telegram), as expected by the architecture.

The flexibility offered by this architecture enables the interaction between home automation systems with any other systems capable of handling HTTP requests. This protocol, highly widespread, facilitates application and integration with existing systems avoiding development costs with a new standard of communication for it.

As it is an architecture that works in the application layer from TCP/IP standard, the technologies used to connect devices to the network (cable, radio, light) and location (local or internet network) do not prevent the interaction of systems, respecting only the logical limitations as firewall filters and network routing.

A critical featuring of this system are the loops, that occurs when an event chain has a triggers defined to fire a request to de same chain. These loops create a cyclic event that will not stop until one of element do not fire or receive a request.

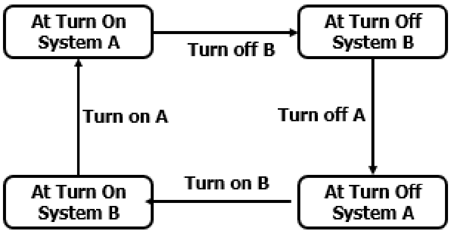


Fig 7. Loop between two lighting system

In Figure 8 illustrates two lighting systems (A and B) that have triggers defined as follows: System A (When switched on, turn off B. Switched off, turn on B) and System B (When switched on, turn on A). This triggers configurations creates an event chain which keeps turn on a turn off both system. That loop stops when on of system do not receive/perform a request from/to other system or manually by user.

Although the architecture has proved viable in your application, some topics are important points of future research such as security application on the data traffic, mechanisms for detecting and preventing loops; name resolution for services on the Internet (DNS-Domain Name Server).

The continuity of the studies also seek to carry out further tests exploring the combination of systems with more robust platforms and more efficient in order to offer a better quality of life for people through the application of this architecture in smart homes.

REFERENCES

[1] SAKER, Subhamay, CHAKRABORTY, Mithun, BANERJEE, Anindita. Low cost embedded system/ Android based smart home automation system using wireless networking. International Journal of Eletronics and Communication Engineering. V. 7, N. 2, p. 175 – 186, 2014.

[2] PIYARE, Rajeev. Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone. International Journal of Internet of Things. P. 5-11, 2013

[3] BHAGYALAKSHMI, P., DIVYA G. ARAVINDA, N. L. Raspberry Pi and Wifi Based Home Automation. International Journal of engineering Research and Applications. P. 57 – 60, 2015

[4] LIN, Hui, LIN, Jianbiao, JI, Ke, WANG, Jingjie, LIN, Feng. Promote the industry standard of smart home in china by inteligente routers technology. ArXiv, 1504.03912v1, Abril, 2015

[5] ROHITH, D. MOUNICA, G. VINOD, K. Design and Implementation of an Embedded Web Controller for Automation. International Journal of Emerging Technology and Advanced Engineering. V. 4, I. 9, P. 564-569, 2014

[6] KUMAR, Shiu. Ubiquitous Smart Home System Using Android Application. International Journal of Computer Networks & Communications. V. 6, N. 1, P. 33-43, 2014

[7] ELSHAFEE, A., HAMED, Karim A. "Design and implementation of a WiFi Based Home Automation System. World Academy of Scrienc, Engineering and Technology 68", p. 2177 – 2183, 2012.