Proceedings of Wireless HIVE Networks Conference, Austin, Texas, USA, August 07-08, 2008

Service-Oriented Middleware for Smart Home Applications

José M. Reyes Álamo, Johnny Wong, Iowa State University

Abstract—**Smart Homes uses different devices including sensors and actuators to provide services that assist in performing the activities of daily living. Currently these smart homes are developed in a manual and adhoc way. Manually configured smart homes create scalability, extensibility and cost problem. The need for a uniform platform for intercommunication among devices and a middleware abstraction using service-oriented technology is presented in this paper. A layered architecture for smart home developers to use is presented. Different demos that take advantage of the service-oriented middleware layer are shown.**

***Index Terms*—Actuators, Middleware, Sensors, Smart Homes,**

I. INTRODUCTION

The Smart Home is equipped with sensors, actuators and other technology to assist the resident performing the activities of daily living. Most of the research in smart homes focuses in assisting the elderly and persons with special needs [1]. The rich number of devices to use as well as the diversity of these technologies poses tremendous challenges at the time of development. Having all these devices working together is usually done in a manual, adhoc way [2]. Having a uniform way for developing and deploying smart homes applications becomes necessary in order to speed the process and reduce the cost. This uniformity will help ensuring that the smart home performs the tasks it is supposed to. It will also maintain the non-intrusiveness, making the resident feel comfortable within a technology pervasive space.

A smart home is implemented by combining and integrating technology already available. This allows providing a series of composed services that a single product cannot, adding more variety to the task that can be assisted by a smart home. Most services and applications in a smart home need some type of sensor such as temperature, light, or weight. These sensors are passive and are used to monitor the environment and measure phenomena. Services may use actuators such as light controller, appliance controllers, and motors to automatic open or close doors. Actuators are active devices that can control other devices and change their state. Smart homes combine sensors, actuators and applications providing a variety of individual and aggregate services and making the smart home a hybrid sensor network. This interaction among devices does not necessarily seek to facilitate the communication among human-to-human or human-to-machine but the device-to-device.

This paper shows how different sensors, actuators and application can be combined to make application for the smart home. It also shows the difficulties on integrating different technology into a single service. Based on these difficulties, important requirement are indentified to facilitate the development and deployment of such services and applications within a smart home. Several examples of current and proposed services and applications are explained. Based on the difficulties and the requirements found we proposed an architecture for smart home sensor applications especially sensor applications. This architecture has several layer one of them the software middleware layer. This software middleware is the key to improve development and interoperability among devices. Several applications on the smart homes that will benefit of using our proposed architecture and middleware are explained. Different demos that have been implemented using this service-oriented approach are also explained and one of these demos is explained more detailed.

The rest of the paper is organized as follows Section II contains related work. Section III defines the smart home as a device-to-device environment and presents two use cases. Section IV lists the requirements for this device-to-device environment. Section V presents the architecture for interoperation among different kinds of sensors based on the requirements. Section VI concludes the paper and shows future work.

II. RELATED WORK

A middleware solution for sensor networks have been studied in the past and some solutions have been proposed. For example [3], presents Atlas that is a middleware sensor platform. The idea of Atlas is to enable programmable pervasive spaces making the sensors plug-and-play. To accomplish this the sensors connect to an Atlas node, which is a piece of hardware, and the Atlas node has to be configured by indicating which sensors are connected to it. After this initial configuration the Atlas node is turned on and it will load a service to an OSGi framework and the sensors will be ready to use. This is a possible solution nevertheless using Atlas adds another piece of hardware which requires a constant source of power. The sensors supported are analog sensors which require wiring. Also the configuration of the Atlas nodes is still very manual and does not detect when a

978-1-4244-2849-6/08/$25.00 ©2008 IEEE

sensor is changed, you have to re-configure the node. Using current standards such as IEEE 802.11 for sensor networks is proposed in [4]. This have the advantage that these protocols are well defined, thoroughly tested and allow for interoperation with current devices that uses the standard. Even thought devices that use IEEE 802.11 consume less energy than in the past, using this standard for sensor networks needs more investigation. In [5] it is proposed a set of configurable devices to use for health monitoring which can be controlled by a Pocket PC. The problems is that the battery lifetime is too short and that the Pocket PC needs to be carried all the time, even inside the smart home for this application to work.

As we can see several devices and application has been developed to alleviate the problem of using wireless sensors for different applications. These same techniques can be integrated together and used in the smart home but this process needs to be simplified and standardized. In the next few sections we define the smart home as a device-to-device environment.

III. SMART HOMES AS A DEVICE-TO-DEVICE ENVIRONMENT

The smart home uses sensors, actuators and applications in order to assist the resident in performing activities of daily living. The emphasis is on the performance and communication among devices and application and not necessarily human-to-human or human-to-computer [6]. Therefore a smart home fits into the definition of a wireless hive network. We present two cases of applications that use different devices to provide a service and how smart home fits the wireless hive network definition.

A. *Medicine Management*

Consider the case of a smart home that assist the resident in the management of their medicines such as in [7]. This system integrates the doctor, pharmacy and the smart home and check prescriptions for conflicts with other medicines, food or medical conditions. In this system each prescription has a unique RFID for identification and tracking purposes. An RFID reader is needed to scan the medicine into the smart home system and to detect the location of the medicine within the smart home. Depending on the user preferences a reminder system can include an alarm, speaking a message or a pop-up message on the TV. Different actuators are needed to perform these tasks and remind the resident to take the medicines. This system has a conflict detection routine which needs to interact with other sensors, actuators and applications. As seen all these task require the interaction of several devices and applications but not with the resident. This data eventually need to be available to the resident, nevertheless the main purpose of the system is the device-to-device interaction and not the user-to-user or user-to-device interaction. Therefore an application like medicine management fit the definition of wireless hive networks.

B. *Food Management*

Food management using a smart refrigerator and a smart microwave is another important service provided by a smart home [8]. In a food management application, RFID tags are used to identify the food items. An RFID reader is used to detect and locate the food. A database with the food information such as name, nutrition facts, and expiration date is needed. Temperature sensors will help to ensure the food remains fresh and at appropriate temperature. Weight sensors are needed to detect when running out of food. A reminder and conflict checking system will also use information from the food management system. In the food management system the food information needs to be available to the user but the interaction among devices is more important. The operations of the sensors, actuators and other devices such as storing information and conflict checking routines needs to be optimized. Again the main focus of the food management service is the device-to-device interaction and not the user-to-device interaction.

These are just two examples of application within a smart home that uses and integrates several sensors and actuators whose interaction need to be optimized for that purpose. Another applications and services in the smart home includes location and tracking, opening and closing the door, wake up service, emergency detection, turning appliances on or off (such as stove or iron) and more. All these applications and services require close interaction among the devices but there is no standard way to do it at this point. Most of these services and applications are integrated in a manual and ad-hoc way. Therefore a set of requirement for sensors and actuators need to be defined so that the development time and the cost of development can be reduced. The next section lists these requirements.

IV. SMART HOME REQUIREMENTS

The smart home has a number of devices interacting with each other. There is a need for these devices to be able understand each other. Different sensors use different communication protocols and some protocols are proprietary. The use of wireless mesh networks [9] to allow communication among different protocols would enable devices to communicate with each other. A wireless mesh network uses wireless mesh router which are routers that understands different communication protocols and route from one protocol to another protocol. For example a router that understands Zigbee, Wi-Fi and Ethernet can be used to interconnect three different networks and exchange data among them. This interconnection is necessary and also that the data sent must be correctly understood by other nodes. To this purpose a software middleware layer is presented in which the raw sensor data is sent over the wireless mesh network, the wireless mesh network communicates with the middleware, and the middleware translate the data to a format understandable by the receiving node. For example the Phidgets sensors [10], returns raw data in numbers from 0 to 1000. In the case of a temperature sensor a returned value of 200 is not very useful. The middleware will take care of making the conversion from this raw value to the appropriate temperature in Celsius or Fahrenheit degrees. This middleware also will make easier application development as programmer can just query for temperature without worrying about which physical sensors are being used.

Other requirements especially for battery operated sensors, is that they must have a long life and be reliable. Some perpetual sensors can be plugged into a constant source of power and other can be wireless. Either way sensors should notify with plenty of time when running out of battery or failing. Also nodes should provide security by notifying when they are under attack. Sensor must ensure privacy and security by allowing only the appropriate parties query the reading of the sensors and access them. The architecture for smart home applications is presented in details in the next section.

The need for a solution that is extensible and scalable is needed as some services in the smart home might be temporary. For example imagine the case where the resident of the home is under certain treatment to determine if the resident has certain health condition. It might be possible that the resident needs to carry a set of wireless sensors for a just a few days. These devices will become new to the smart home. Configuration and testing should be minimal and transparent, a process that is not possible in today's smart homes. The resident might not want to alter the smart home configuration and re-program it. Having an easy way to add these sensors to the smart home and remove them when no longer needed, will be very helpful. This requires a common architecture which is presented in the next section

V. SMART HOME LAYERED ARCHITECTURE

The following is the proposed architecture for the smart home sensor applications. At the very bottom we have the Phenomena and Objects Layer. In this layer the phenomena to be sensed by the sensors and the objects controlled by the actuators are found. The next layer is the sensor and actuators layer that consist of sensor and actuators from different manufacturers. We assume that these devices implement different protocols. The next layer is the connectivity layer. This layer has the wireless mesh routers which enables communication among different protocol and allow the sensors to communicate among themselves. The next three layers will reside on the smart home host computer. The layer on top of the connectivity layer is the software middleware layer. This software middleware layer will be responsible for implementing the device specific details for each of the sensors and actuators and provide them as services. This will add a layer of abstraction in which instead of reading raw sensor data you will read significant values. These services provided by the software middleware layer will be made available to the service layer. The service layer is a collection of all sensor, actuators, and application services all under the same framework or platform. These services can be incorporated in service-oriented frameworks such as OSGi [11]. This service layer will provide the services to the applications layer. It is in the application layer where the different smart home operations are performed and the one with which the resident interacts with the smart home. In Figure 1 you can see a visual representation of this architecture. We have implemented several demos using this service-oriented approach and using sensors and actuators from different manufacturers in our smart home lab. A description of some of these follows in the next section.

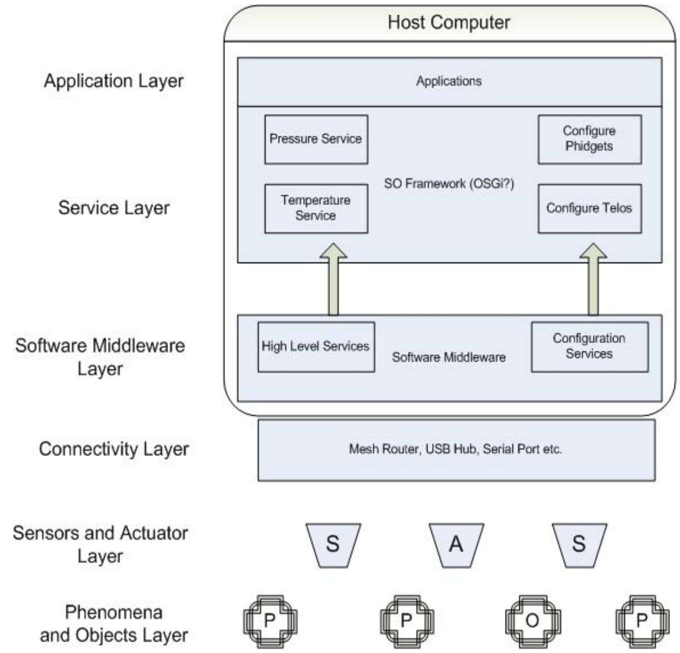

FIG. 1. SMART HOME LAYERED ARCHITECTURE.

VI. SMART HOME DEMOS

In our smart home lab we have several demos that use devices such as Phidgets sensors, rfid reader, and web cameras, and X10 appliance controller, and telos motes. Some of these demos interoperate with different technologies by invoking services provided by the service layer. In our case the middleware is also implemented as a service. Applications are therefore developed using only the services provided, abstracting the details on the particular hardware that do the readings. For example we have a temperature service which can either use Phidget sensor or a telos sensor. For the application it does not matter which sensor we use as it uses the temperature service which abstracts the particular hardware that makes the temperature readings. We have implemented several demos using this approach such as the medicine management, a smart fridge, a camera controller and an alarm system that turns on appliances at the time to wake up.

To show interoperability among different protocols we will describe in details one of the demos we have developed. This demo uses wireless sensors to collect the data and communicate with an application via the Internet. We have one telosb mote A which is sensing phenomena. We have another telosb mote B which is a base station node connected to the host computer over a USB port. Node A sends a packet to node B using the protocol IEEE 802.15.4 for Wireless Personal Area Networks. Node B communicates with an OSGi service which reads the packet from the telosb node. The raw data is transformed into meaningful data by the service. This data is then passed to another service which communicates with an application via the Internet. This application consists of a server that receives the data and reads is loudly using a text-to-speech module. Figure 2 provides a visual diagram of this demo. The response time between readings the data, send it to the service, send it to the speech-application via the Internet took less than two seconds in all the tests. This is very reasonable time frame for a smart home application where response time is not as critical as industrial applications. Also taking in considerations that the OSGi framework and the sensor network application are running on a VMWare Virtual machine running Xubuntu Linux, all the services are running in an OSGi framework and the speech application is running on a MAC reachable over the Internet. So even after all these layers of abstractions the response time is very reasonable for smart home type applications. The fact that service oriented approach can cross several platforms in a reasonable amount of time is also a plus.

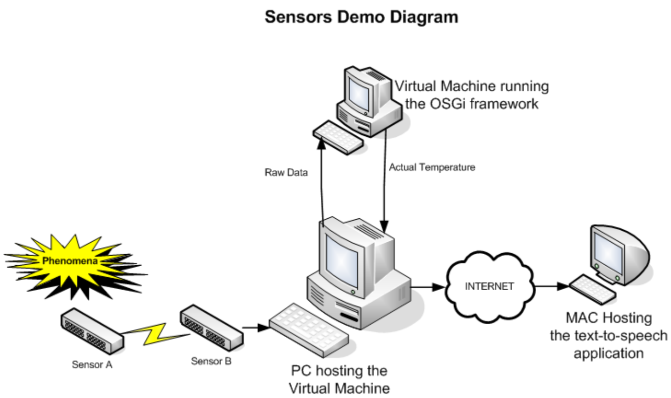

Fig. 2. Sensors Demo Diagram.

VII. CONCLUSION AND FUTURE WORK

A smart home has several sensors and actuators providing different services. Interconnecting them is difficult and is often done in a manual, adhoc way. In this paper we identify key requirements for smart home applications that use several sensors and an architecture that will ease the task of development and reduce the costs. The approach focuses on the device-to-device communication and the use of a software-layer middleware. We showed that this approach applies to smart homes in application such as medicine and food management as the tasks the devices need to perform are not designed specifically for human-to-device interaction. The architecture emphasizes a mesh layer which will make communication among different protocols possible and a service-oriented software middleware layer which will make possible to abstract the devices to the service they provide without worrying on the particular device. We mentioned how several demos on our smart home lab used this approach and gave the details on one of the demos that use different communication protocols and layers of abstractions to show the feasibility of this proposed architecture and middleware. The response time for these applications and demos is acceptable for smart home application. Future work includes refining the middleware layer and tests it with more devices. Making this middleware plug-and-play is another research goal.

REFERENCES

[1] N. Noury et al., “New trends in health smart homes,” enterprise networking and computing in healthcare industry, 2003. Healthcom 2003. Proceedings. 5th international workshop on, 2003, pp. 118-127.
[2] R. Bose et al., “Building plug-and-play smart homes using the atlas platform.”
[3] Jeffrey King et al., “Atlas: A service-oriented sensor platform: hardware and middleware to enable programmable pervasive spaces,” local computer networks, proceedings 2006 31st IEEE conference on, 2006, pp. 630-638.
[4] Lewis Adams, “Capitalizing on 802.11 for sensor networks.”
[5] I. Korhonen, J. Parkka, and M. Van Gils, “Health monitoring in the home of the future,” engineering in medicine and biology magazine, IEEE, vol. 22, 2003, pp. 66-73.
[6] The wireless hive networks manifesto.
[7] J.M. Reyes Alamo et al., “Miss: medical information support system in the smart home environment.”
[8] Sumi Helal, “The gator tech smart house: a programmable pervasive space,” mar. 2005.
[9] I. Akyildiz and X. Wang, “A survey on wireless mesh networks,” communications magazine, IEEE, vol. 43, 2005, pp. S23-s30.
[10] “Phidgets inc. - Unique and easy to use USB interfaces”; http://www.phidgets.com/.
[11] D. Marples and P. Kriens, “The open services gateway initiative: an introductory overview,” Communications Magazine, IEEE, vol. 39, 2001, pp. 110-114.