Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 5 (2011) 386–393
Procedia Computer Science

The 2nd International Conference on Ambient Systems, Networks and Technologies (ANT)

An Intelligent Object Framework for Smart Living

Sasa Savic and Hao Shi*

School of Engineering and Science
Victoria University, Melbourne, Australia

Abstract

Wireless embedded modules have increasingly been implemented on a range of devices, becoming a standard inclusion in mobile technology, entertainment systems and gaming consoles. It will not be far off before they are introduced to household appliances and industrial devices. However, the key challenge is that each manufacturer implements a proprietary set of protocols and standards, and in addition, a set of management applications, which add an unnecessary overhead of compatibility issues and protocol inconsistencies. The aim of the project is to propose an Intelligent Object Framework (IOF), a new communication standard over a wireless network without the existing complexity of a multiple set of architectural solutions. The Intelligent Object Framework consists of a framework design that enables devices of different platforms to communicate by a common data exchange model via a device management controller. In this paper, a prototype of the Intelligent Object Framework is implemented based on Microsoft .NET framework. A substantial amount of research work is carried out to create IOF Management Studio, Intelligent Objects devices, system integration and testing. It demonstrates that a single application can manage multiple Intelligent Object devices efficiently under the proposed Intelligent Object Framework.

Selection and/or peer-review under responsibility of Prof. Elhadi Shakshuki and Prof. Muhammad Younas.

Keywords: Intelligent Object Framework (IOF); Intelligent Object (IO) Devices; Ambient Systems; Smart Living; Protocol

1. Introduction

Embedded devices and programmable logic units are more powerful, with increased processing power, large storage capabilities, faster and varied communication mediums, pre-installed with a variety of Real-Time Operating System (RTOS) that ubiquitously support a range of communication protocol standards

* Corresponding author. Tel.: +61-3-9919-4060; Fax: +61-3-9919-4908.
E-mail address: hao.shi@vu.edu.au

Selection and/or peer-review under responsibility of Prof. Elhadi Shakshuki and Prof. Muhammad Younas. doi:10.1016/j.procs.2011.07.050

and application frameworks. Whilst this is still an evolving paradigm, discrepancies can be seen at application and protocol management levels. Manufacturers of systems constantly face daunting tasks of finding more efficient ways of managing their devices and must specify the chronology of the interactions between certain communication entities, be that a physical device or an application.

As design of data exchange models become critical in real-time environments, certain protocols have been implemented to overcome the compatibility issues due to lack of reliability in protocol design. One of those protocols is SNMP (Simple Network Management Protocol) [1]. Whilst being a transparent request and response protocol that communicates certain management information between two types of SNMP software entities, it fails to meet certain preemptive actions and response times. Although it is mainly implemented in network devices and secondary management interfaces, it has also been found on computers and printers. Despite its apparent compatibility across distributed systems, it is considered a highly complicated protocol to implement as it comes with numerous abstruse encoding rules and schemes. SNMP also lacks in efficiency as network bandwidth is exhausted with useless data. As described in RFC3411 [2], section 4.4.3, SNMP version is transmitted in every SNMP datagram. In addition to the unnecessary data overhead at each datagram, multiple length and data descriptors are broadcasted throughout each message. This inefficiency in packet encapsulation methodology yields superfluous data handles which are broadcasted across the network. This effect is not felt in small to medium networks, although, networks that consist of large scale distributed devices do. Furthermore, device failures are not detected in real-time and devices that succumb to network failures will not be noticed until the next polling cycle. Software applications managing the SNMP enabled devices must be aware of the available information that lies within the devices. If there is a discrepancy in the schema, this must also be changed in the management application.

Another kind of network management and control protocol that has emerged since the late 90s is xAP Home Automation protocol [3]. An open protocol intended for support and integration of telemetry and control devices within a home. While the framework and protocol definition [3] enables multi-device control over UDP protocol, it achieves this by multi broadcast messages. The approach must rely on independent applications and devices to drop unwanted packet data. Moreover, failure to address the issue of centralized function management yields in a complex software and hardware design. xAP approach relies on each device implementation to vary its schema, thus it does not conform to a framework standard, but rather rely on protocol specification itself.

There is an urgent need to have a standard communication protocol over wireless network to abolish the existing complexity of a multiple set of architectural solutions. In this paper, a new Intelligent Object Framework (IOF) is proposed to enable devices of different platforms to communicate by a common data exchange model via a device management controller. The rest of this paper is organized as follows. Section 2 briefly describes the related research work on wireless applications. Section 3 presents the proposed Intelligent Object Framework (IOF) architecture and its major components. Section 4 explains the IOF System Design regarding protocol layer and data layer. IOF Implementation including IOF Management Studio and experiments on Intelligent Objects (IO) using the framework as test cases are demonstrated in Section 5. Finally, conclusion and future work are addressed in the last Section.

2. Related Work

The use of Wi-Fi connectivity in non PC based devices such as MP3 players, dual mode cellular and Wi-Fi VoIP phones, video games, printers, smart phones, PDA's, tablets and televisions is rapidly growing [4]. It has been reported that 56 million cellular Wi-Fi phones had shipped [5]. It is up approximately 52 percent, around 48 million consumer electronics devices including consoles, digital televisions, set-top boxes, printers unit shipped which is up 51 percent and staggering 71 million Portable Consumer Electronic Devices that account for hand-held games, cameras, portable music player units shipped, up 33 percent, only in 2008 alone [6]. The advancements and modularity of the Wi-Fi modules have increased productivity and sales of electronics devices drastically.

The use of wireless applications however, stretches even further than conventional devices and has been implemented in varieties of different systems. One typical application outside the device spectrum has been seen in home automation systems. One particular system is Ambient Intelligence within a Home Environment [4] which aims to implement a collaborative system of wireless proportions to accommodate many devices to talk to a central control unit. The system has been implemented as an automated ambient home solution exhibiting devices such as the television, washing machine, heating system to be accessed wirelessly. There is a great advantage to the use of technology accordingly to its specifications, although the initial architecture does limit the user to be using only a certain set of appliances in accordance with the system. This exhibits future integration issues that are limited by the design.

Zigbee, a de facto standard for WSNs (Wireless Sensor Networks) becomes one of the most promising protocols for smart home and automation due to its low-power consumption, low cost, and supports for various ad hoc network configurations [4, 8, 9]. A Ubiquitous Home Services has recently been designed and implemented using dynamic integration of Zigbee devices into residential gateways based on the OSGi (Open Service Gateway Initiative) service framework [10]. The ubiquitous service systems not only can be used for home network service domains but also for a variety of service domains including automotive, office, and hospital services with enhanced security .WiiKey Smartphone application [11] is another interesting application that has been researched applying wireless technology and communication methodology. The research is focused around smart objects that act as wireless devices, which are controlled wirelessly by a controller application. The concept of WiiKey has been thought out quite well, although the fundamental framework is flawed. WiiKey does not define a framework to incorporate a standard set of protocols used for interconnecting these devices with a common application infrastructure, thus not providing the tools for future integrations with other devices or systems. The Ambient Assisted Living (AAL) for assisting elder people has been pursued recently and it has gained greater importance because of the increasing life expectancy worldwide [12]. The Wi-Fi technology fails short in accuracy of the location information. Further, the elder users have quite specific requirements and their relationship with augmented objects and modern interface technologies have yet to be carefully analyzed.

3. The proposed Intelligent Object Framework (IOF) architecture

The proposed Intelligent Object Framework (IOF) differs by accepting device changes via adding functions, storing them to a database by conforming to a common Intelligent Object (IO) data model. Once the data exchange occurs, the functions and its IO data types become available to any application, thus enabling the user to execute the new/amended function. The overview of the proposed IOF is illustrated in Fig. 1.

4. Intelligent Object Framework (IOF) Prototype

Prior to the design and implementation of an Intelligent Object Framework prototype, several design factors need to be considered:

- Operating system type which acts as a sandbox for development
- Type of Integrated Development Environment that the prototype solution will be designed in
- Chosing a stable database management system that supports relational database model
- Choosing the right set of embedded devices that act as an Intelligent Object in order to prove the concept of communication
- Type of protocol implementation that best suits the design model
- Defining chronological model of real-time interaction between communicating entities

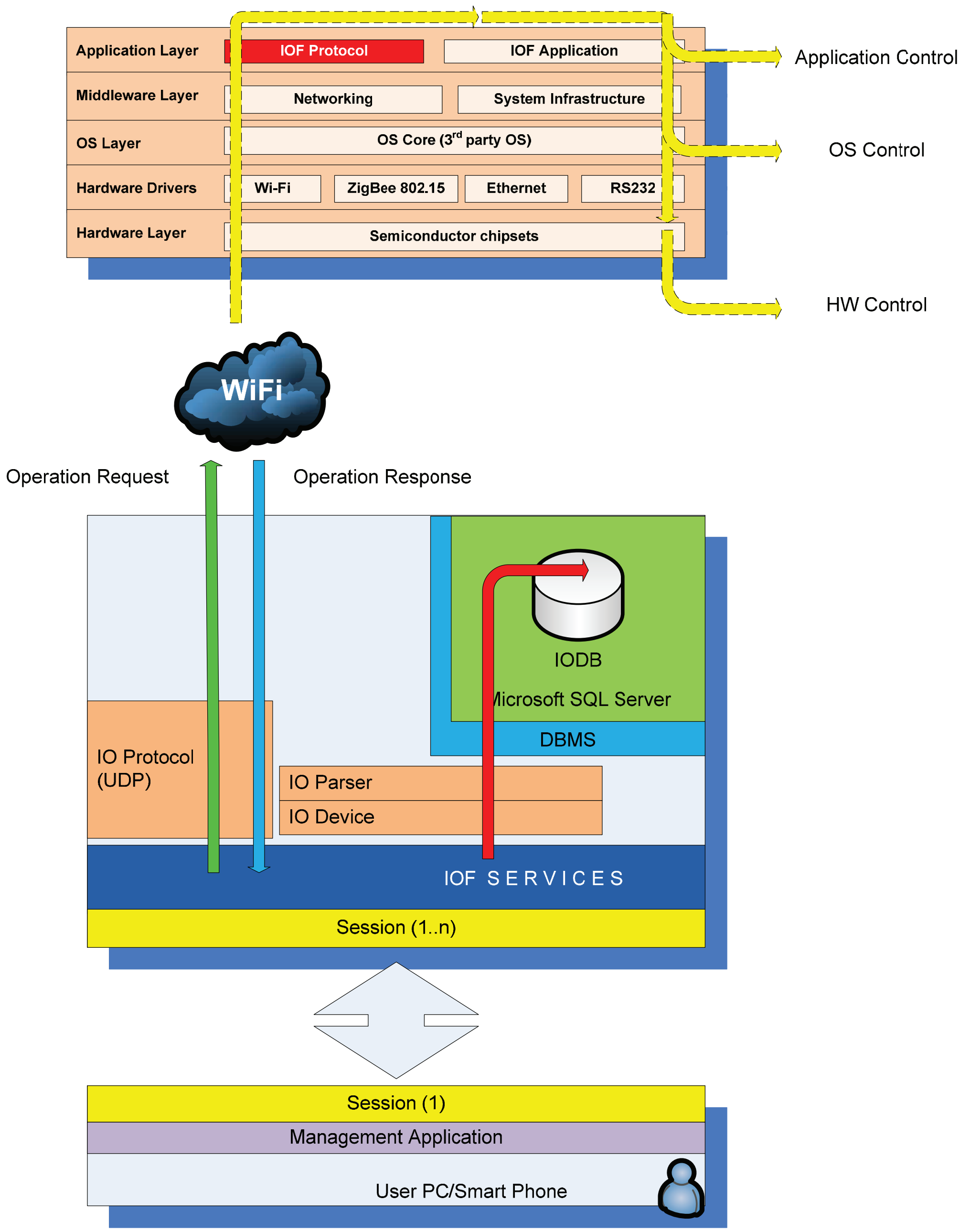

Fig. 1. Overview of the proposed Intelligent Object Framework (IOF

The high level architectural design of IOF is described in Fig. 2. IOF model consists of Services/ProtocoLayer, Device Layer which implements Parser Interface where Processing/Logic Layer is defined, and finally a Data Layer

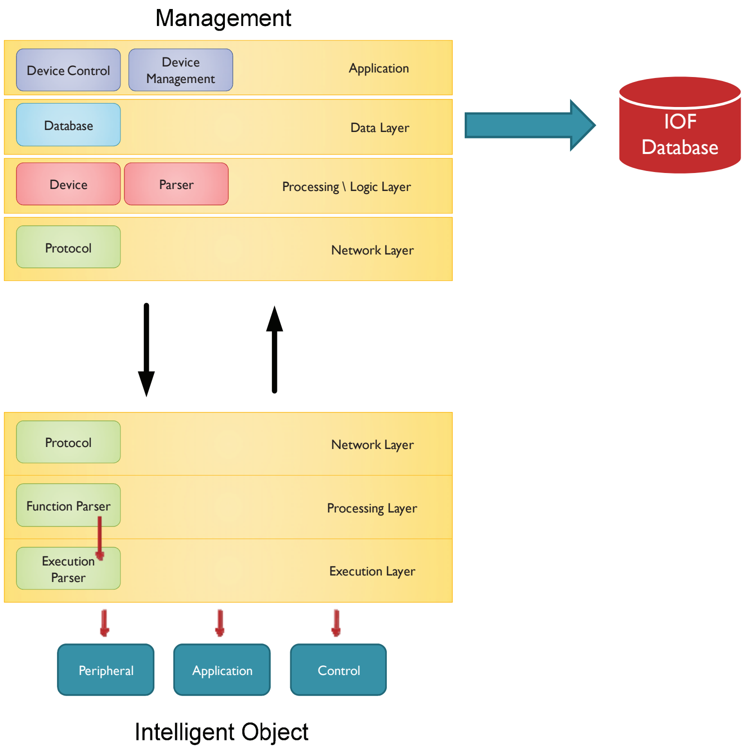

Fig. 2. IOF system architecture block diagram

The IOF consists of three main components from the service end:

- The Network Layer: defines the protocol and services,
- Processing and Logic Layer: describes the device object and the parser needed to evaluate the data coming from the Intelligent Object device, and
- Database Layer: interacts with the database for storage and retrieval of device functions.

The Intelligent Object comprises of similar layers as the framework itself, except it does not need data layer as storage of functions is only done on the service end. The Intelligent Object (IO) does include an extra Execution Layer which interacts with applications, peripherals or abstract controls. The interactions are defined as functions that are sent to the Intelligent Object services, defined by the protocol.

The IOF Services layer handles sessions from client side controllers. The session establishes communication with controlling client side by implementation of IOF protocol definition within IO Protocol. Any communication data that arrives via client or device side is placed through IO Parser. Once the data is evaluated, parser will hand over data chunks to corresponding handlers the IO Device handler, and IO Data which fetch device data from the database, and handle session to and from endpoints (devices, controllers).

5. Intelligent Object Framework (IOF) Implementation

The IOF prototype is based on Microsoft .NET Framework and Windows Mobile SDK. The design and implementation of IOF has proved to be a complex task as extensive research is carried out on communication models and protocol implementation specifications design. The right set of devices needed to be chosen, that have the right platform features allowing the implementation of core components, in order to design and develop the proof of concept.

Based on Microsoft .NET [13], the Intelligent Object Framework Management Studio is developed to provide an easy to use interface for managing devices attached to the network. Underneath the simplicity of the design lies an engine that dynamically creates the controlling environment based on the function content received from a particular device or set of devices. It shows the Management Studio platform with the basic set of features. It renders a mainstream application design with features like control panel, status screen and the device viewing window.

The Object Requests section, shown in Fig. 3 shows a device that is seeking a connection to the network. Important device information is displayed upon connection request. The user can see the device name and additional information. Another important factor that is displayed is the function number. This shows the number of functions the particular device has on offer. The data underneath is used in setting up the function iteration loop which receives the functions from the intelligent object. The Object Request information window also displays the MAC address as it is a primary identifier that is used in distinguishing different devices from one another within the framework itself and the database.

Fig. 3. Object request window under IOF management studio

5.1. Intelligent Objects

Intelligent Object (IO) is a device that conforms to Intelligent Object Framework (IOF) specification. It allows a developer to define abstract methods which map to certain device functions. These methods are sent to a management application services via a defined function exchange protocol which in turn store the functions into a database. These functions can be called up and accessed at any time from the database via the management application. This process creates a function execution request that is sent to the intelligent object that reads the request and processes it accordingly.

Intelligent Object is developed using Tibbo Integrated Development Environment (TIDE) and T-BASIC programming language that is loaded on an EM1026 Embedded Device Platform running Tibbo Operating System (TiOS). The backend database is developed in Microsoft SQL Server. In order to achieve platform independency between devices, two devices have established a heterogeneous wired and wireless network environment using an built-in Wi-Fi adapter. Interpretation of an Intelligent Object enabled device defines an embedded device that has the capability of communicating via IOF standards and that it also provides a function execution model for processing requests from the management application, which means if a request is sent to a device to open a door, the device must be able to interpret the message and act accordingly whilst providing some form of feedback to the caller.

In order for the device to conform to these standards, it must allow the software engineer to have the capability to program it, hence the two devices that are chosen to act as intelligent objects were HTC Mobile running Windows Mobile 6.1 OS as shown Fig. 4(a) and TIBBO EM1206 Evaluation Board with an GA1000 Wi-Fi addon module in Fig. 4(b).

Fig. 4. Intelligent Objects (a) HTC Diamond running Windows Mobile 6.1 (b) TIBBO EM1206 Evaluation Kit

TIBBO embedded device runs lightweight real time operating system, TiOS, that provides a programmer with modular set of high-level objects. This in turn allows for object oriented design approach as opposed to sequential approach. As embedded linux approach is far more complex to implement, TIBBO IDE allows quick implementation of network-enabled systems.

5.2. Test Cases

The system has been sucessfuly tested providing integration and management to an enbedded device and a smartphone device. The HTC Smartphone device shows three distinctive functions that have been derived. The process to conform the framework can be summed up by the sequence diagram that depicts the typical function exchange scenario. Once the process is complete, the user can access the device functions, thus being able to manage and monitor the device. The test case provided is of the most fundamental capability of the IOF framework. In one of these test cases, the user attempts to make a phone call (command the HTC Smartdevice to dial a phone number) as shown in Fig. 5.

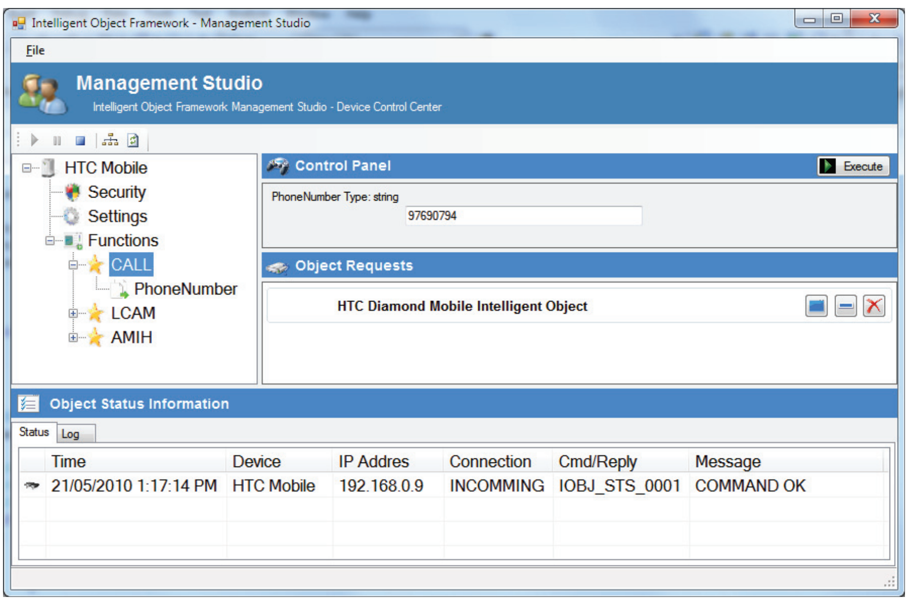

Fig. 5. Management Studio – HTC Mobile Data

6. Conclusion and Future Work

The primary aim of the project is to develop a new Intelligent Object Framework (IOF) to allow a single application managing multiple Intelligent Object (IO) devices. The research is focused on using the current set of technologies to provide the platform for communication and a set of protocols for device management and control. The implementation of the IOF framework prototype and its set of components have supported a bilateral communication protocol for multiple platform independent device monitoring and control. The successful testing of Intelligent Objects, each running different platform and environment, on the developed IOF Management Studio has demonstrated the IOF has fulfilled its objective. As such, the IOF presents a very promising future in platform independent device management of household and industrial devices and paves a way for a better device integration and management.

However, the currently developed system only supports the basic implementation of the services and protocol such as allowing functions to be downloaded from an Intelligent Object device and stored to a database. By following the framework definition, the IOF could be extended and scaled to allow developers to take the advantage and control their own Intelligent Objects range from mobile phones, embedded devices to an entire set of household devices. The IOF can be further extended by implementation of a secure protocol with token based authentication or equivalent, which would add an overhead on the UDP packet, although it would provide a safe communication channel. The protocol definition can further be improved by implementation of TCP protocol suite for critical devices that need a connection oriented communication. It would provide a logical connection that negotiates before sending data to one another. This addition to the current solution would provide a reliable communication between Intelligent Objects and Intelligent Management Services. The ultimate objective is to provide an effective solution for allowing different devices to be controlled and monitored via a single application using the platform independent framework.

References

[1] SNMP (Simple Network Management Protocol), http://www.snmplink.org/

[2] Harrington D, Presuhn R, Wijnen B. An Architecture for Describing SNMP Management Frameworks, The Internet Engineering Task Force (IETF), RFC3411, December (2002) 34-40.

[3] Ian Lowe et al. xAP an open standard for the home, http://www.xapautomation.org/index.php?title=Protocol_definition

[4] Graham C. Conformity, *Engineer's Reference Guide* 2009; **14**:178-181.

[5] Wi-Fi Alliance, http://www.wi-fi.org/

[6] Bielikova M. and Kraiovic T. Ambient Intelligence within Home Environment, http://www.ercim.eu/publication/Ercim_News/enw47/bielikova.html

[7] Callaway E. et al. Home networking with IEEE 802.15.4: a developing standard for low-rate wireless personal area networks, *IEEE Communication Magaine* 2002; **40**:70–7.

[8] Zigbee Alliance: Zigbee Vision for the Home: Zigbee Wireless Home Automation. Zigbee Whitepaper (2006)

[9] Zigbee Alliance: Zigbee Enables Smart Buildings of the Future Today. Zigbee Whitepaper (2007)

[10] Ha YG. Dynamic Integration of Zigbee Devices into Residential Gateways for Ubiquitous Home Services, *UIC* 2009:**LNCS 5585**:221–235

[11] Vo N, Shi H and Szajman J. WiiKey: An Innovative Smartphone Based Wi-Fi Application, Proceedings of International Multi-Symposiums on Computer and Computational Sciences 2008, Shanghai, China, pp. 91-97

[12] Avil´es-L´opez E, Antonio Garc´ıa-Mac´ıas J, Villanueva-Miranda I, Developing Ambient Intelligence Applications for the Assisted Living of the Elderly, Proceedings of the International Conference on Ambient Systems, Networks and Technologies (ANT2010), Paris, France, pp. 53-60.

[13] Microsoft MSDN: Introducing Visual Studio, http://msdn.microsoft.com/en-us/library/fx6bk1f4(v=VS.100).aspx