Flexeo: an architecture for integrating Wireless Sensor Networks into the Internet of Things

Juan Ignacio Vazquez, Aitor Almeida, Iker Doamo, Xabier Laiseca and Pablo Orduña

MoreLab-Mobility Research Lab

Tecnológico Fundación Deusto

Avda. de las Universidades 24, 48007 Bilbao,

{ivazquez, aalmeida, idoamo, xlaiseca, porduna}@tecnologico.deusto.es

Abstract. Wireless sensor networks are a hot topic in Ubiquitous Computing for implementing context-awareness scenarios. The connection of sensor nodes to the Internet leads to new ways for remote monitoring of human behavior in real-time. In this paper, we introduce Flexeo: a flexible architecture for implementing monitoring solutions based on wireless sensor networks, with distributed intelligence at different layers. In this way, sensor-populated scenarios may communicate with Internet-based facilities enabling the vision of an Internet of Things.

Keywords: Wireless sensor networks, Internet monitoring, distributed architecture, Internet of Things.

1 Introduction

Context awareness is one of the most important aspects of Ambient Intelligence and Ubiquitous Computing, with a remarkable amount of research that has been carried out during the last years. The most basic form of perceiving context information consists in deploying ubiquitous sensors in the environment, which are able to capture part of the existing data, correlate and synchronize these data, analyze them, and finally, carry out a reactive activity without user intervention.

One of the most popular technologies for creating these sensor-populated scenarios are Wireless Sensor Networks (WSN). The use of an energy-efficient wireless bearer such as IEEE 802.15.4/Zigbee [1], enables the rapid and seamless deployment of sensor nodes in any scenario, with minimum requirements from existing infrastructure. Small size and easy integration of physical sensors into these platforms leverages their flexibility to address different types of applications [2].

On the other hand, there is an existing trend about connecting physical objects to the Internet, acting as first-class citizens both for publishing information or retrieving data that may determine their future behavior. This approach has been dubbed the Internet of Things [3]. On the Internet of Things, real objects may benefit from all the existing knowledge that is available on the Internet for better fulfilling people's needs. For example, an Internet-connected umbrella may access a public weather information website, retrieve the weather forecast in a language such as XML, and switch on a LED to indicate this situation. In this way, users do not need to access the Internet through a different device (a computer) in order to take the decision whether to use or not a concrete object (the umbrella), achieving a better functional integration.

The other way around is also possible: creating physical objects that capture part of the real world information and publish it on the Internet. This approach enables the possibility of creating communities of Internet-connected devices that share real world information.

As we mentioned, wireless sensor networks are one of the most suitable technologies for capturing real world data. Therefore, connecting WSN to the Internet in order to publish contextual data in standard ways so that they can be shared with other entities, analyzing these data, taking decisions in remote premises, and finally implementing these decisions back in the real world through actuators, is a challenging activity where complementary technologies may be applied.

In this paper, we present Flexeo, a flexible architecture for connecting wireless sensor networks to the Internet, and distribute the intelligence and the decision making process through different layers. Section 2 analyzes previous work in the field of Internet-oriented wireless sensor networks. Section 3 presents the general architecture of Flexeo, while section 4 describes the different layers of the architecture. Section 5 introduces some of the prototypes developed for the system, and finally, section 6 presents some conclusions and future work.

2 Previous Work

Wireless sensor networks have been a hot topic in Ubiquitous Computing research during the last years, facilitating the deployment of Ambient Intelligence scenarios. Different types of platforms have been available such as Berkeley Motes and Smart-Its [4], and more recently Sun SPOT [5] and Intel Mote 2, which promise more powerful and easier application development tools, along with increased computing capabilities.

These wireless sensor platforms have been widely used to prototype a large number of applications, ranging from environmental monitoring [6] to medical monitoring in projects such as CodeBlue [7], Alarm-Net [8] or MyHeart [9]. It is noteworthy how medical applications, which require continuous monitoring of patients' vital signs, are a suitable niche for experimenting with non-intrusive small sensing nodes.

The main problem with all these experiences is that their architecture has been designed in a very *ad hoc* way for the problem being addressed, being difficult to adapt to other scenarios or applications, although the core problem is the same: remote monitoring and operation of dynamic sensor networks. Our previous experiences with WSN-based monitoring also featured this lack of flexibility, forcing us to redesign different aspects of the architecture depending on the concrete target application.

On the other hand, the Internet of Things (IoT) is a newly developed approach, initially promoted by ITU [3], being "*the next step in 'always on' communications, in which new technologies like RFID and smart computing promise a world of networked and interconnected devices that provide relevant content and information whatever the location of the user*". Any device or object can be directly or indirectly connected to the Internet and communicate with other objects, services or people, even forming communities of devices that mimic social behaviour [10].

One of the first experiences on integrating WSN into the Internet is SenseWeb [11], a P2P-like open architecture for sharing sensor data through the Internet. Using the SenseWeb API users can register their own sensors to publish information and create a common repository of sensor data. SensorMap is an example of end-user application consisting on a mash-up that combines sensor streams obtained from SensorWeb and maps from Virtual Earth.

The main limitation of SenseWeb from an architectural point of view is its centralized vision: all the decision process is carried out at a single central point called Coordinator, where all the sensor data are stored and analyzed. That is, all the intelligence in the system is located at a unique place and all the data must me sent through communication gateways to this point in order to analyze them and take the appropriate action.

From our experiences, we found that different decisions levels could be mapped onto different architectural layers, so that part of the analysis of sensor data and the determination of the reactive response could be also done at a local level, where the sensors are deployed. Under this approach, the communication gateway is transformed into an intelligent entity, able to perform reasoning based on existing local data, as well as forwarding the appropriate information (such as reporting taken decisions) to upper levels in the architecture. Of course, a central point of control still exists for global analysis of data and remote monitoring and operation.

In order to solve the two mentioned issues in existing architectures (lack of flexibility and centralized decision taking process) we created a flexible architecture for wireless sensor networks integration into the Internet, that could be customized in different ways, embedding intelligence at different layers, in order to accommodate to the disparate requirements of possible application scenarios with minimum redesign and recoding.

3 General Architecture

Flexeo architecture is divided on three different layers (see Fig. 1): the Sensors and Actuators Layer, the Coordination Layer and the Supervision Layer.

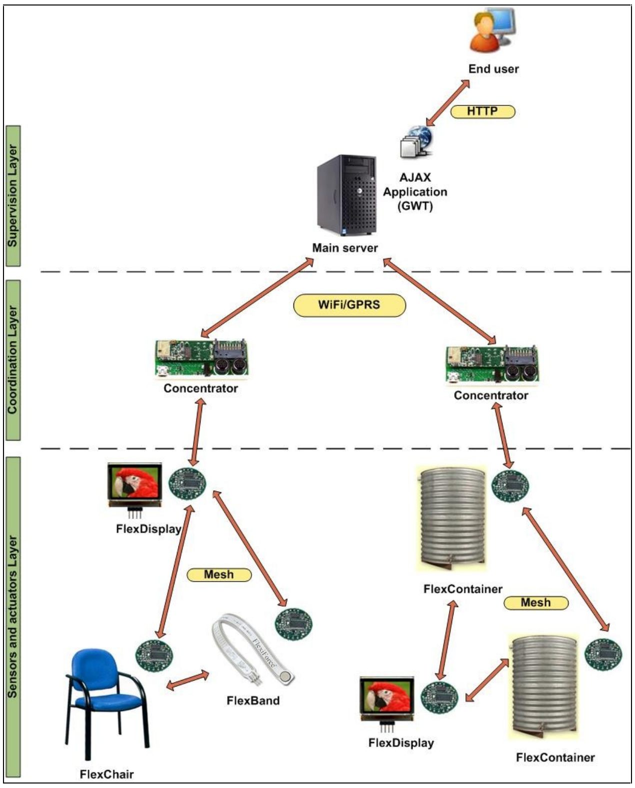

Fig. 1. Flexeo Architecture

The Sensors and Actuators Layer was formed by the sensors and actuators that interact with the environment. Every sensor/actuator was integrated on wireless nodes (Crossbow Mica2 Motes). These nodes form a mesh network and send the information gathered by the sensors to the Coordination Layer through a special node called the base node. Messages are routed from one node to another until they reach this base node. For instance, sensorized chairs may have two sensors (at the seat and the back, in order to know whether the user is correctly seated or not), while wireless displays have one actuator (an organic screen). Even if these devices have more than one sensor or actuator, only a single wireless node is needed to integrate them.

The Coordination Layer is responsible for the management of the data received from the sensor network and for taking decisions based on this information. This layer will store temporarily the gathered data in a buffer, which is sent to the upper layer once this buffer is full, in order to reduce the number of connections. Since GPRS is used in some scenarios, it is important to minimize the number of connections due to their cost. Preliminary work has been carried out to develop a protocol that uses missed calls to force the coordinating device (called *concentrator*) to establish a new GPRS connection before performing a web request. The use of GPRS is optional, since the concentrator also supports IEEE 802.11b (Wi-Fi). The concentrator is implemented on a Gumstix embedded device [12], powered with an Intel XScale processor and embedded Linux, which has more advanced computational resources compared to the sensor nodes found in the lower layer.

Finally, the Supervision Layer provides the concentrators with a Web server to publish on the Internet the data retrieved from the sensor network they manage. This layer stores the data retrieved from the Sensors and Actuators Layer in a database. The Supervision Layer also offers a Web interface for the end users to manage the sensor data and generate statistics, as well as an API for integrating the monitoring and control functions into any existing business application.

4 Layers and intelligence distribution

This section describes how intelligence is distributed among all the layers, along with involved technologies.

4.1 Sensors and Actuators Layer

The selected sensor platform for this layer was Mica2 provided by Crossbow. This platform was selected due to its stability over the other existing model, MicaZ, which was in testing phase when the project was developed. We used this platform in order to design and implement several sensorized objects (see section 5), and the base station that acts as a gateway between the sensor network and the coordination layer.

There is almost no intelligence at this level, due to the computational limitations of the wireless sensor nodes, although some forms of data aggregation strategies were performed in order to provide higher level information to the Coordination Layer.

4.2 Coordination Layer

The Coordination Layer is implemented through some entities called concentrators, which are embedded platforms based on Gumstix, powered with the OSGi framework [13] and different OSGi Bundles, offering a collection of services to access the sensor network. Each concentrator controls one sensor network at a concrete location.

OSGi

In order to provide a modular, loosely-coupled, device-independent framework in the concentrator, each node (sensor/actuator/device) must provide a programming interface with the methods to interact with the node itself. In this way, it is possible to perform high level queries to find and manage the nodes controlled by the concentrator. The OSGi Framework provides this level of abstraction and is available even for some embedded systems, so it was a good candidate for abstracting the sensor layer within the concentrator.

Due to the RAM (64 MB) and flash (16 MB) memory constraints, the selected OSGi implementation must have a small footprint and low platform requirements. In order to select one implementation, existing ones were tested to check how suitable they were for Flexeo. Table 1 shows the results of the tests (the tests were done in the Gumstix platform, using the JamVM Java Virtual Machine and GNU Classpath v0.90 libraries).

Table 1. OSGi Framework comparison

	RAM Memory	Flash Memory
Concierge R3 (without HTTP server)	2864 KB	116.8 KB
Concierge R3 + HTTP Oscar R4	12776 KB	773.4 KB
Concierge R3 + HTTP KF R3	9572 KB	384.8 KB
Concierge R3 + HTTP KF R4	*Invalid classpath*	
Equinox R4 (without HTTP server)	10324 KB	1292 KB
Knopflerfish R4	*Invalid classpath*	
Oscar R4 (without HTTP server)	8080 KB	788 KB

As the table indicates, the Concierge [14] with HTTP Knopflerfish R4 service and the Knopflerfish R4 [16] implementation could not be run in the Gumstix platform due to classpath incompatibility. Equinox R4 [15] and the Oscar R4 [17] implementations were discarded based on their high resource requirements. Finally the Concierge R3 implementation was selected. In order to provide the HTTP Service, Nano HTTPD [18], a tiny embeddable HTTP Server written in Java, is used since the resources it requires are more affordable than the ones required by the Knopflerfish R3 HTTP Service (it only uses 13 KB in Flash Memory and 186 KB in RAM Memory).

Flexeo Concentrator Services

The Flexeo HTTP Service provides a REST-based API to enable the access to the values gathered by the sensors, returning the results based on a given device type or device identifier.

The Network Manager Service deals with the network connections. This service can handle different types of network connections. Although GRPS and Wi-Fi are implemented, other type of connections could be added. In order to minimize the number of connections, the Buffer Management Service stores the values subject to be sent to the upper layer in a buffer. This data is periodically sent when the buffer is full.

Intelligence at the Concentrator

In Flexeo, system intelligence is mostly distributed along two levels in the architecture: the Coordination Layer and the Supervision Layer. The Gumstix-based concentrator is the entity that implements the intelligence at the Coordination Layer.

The strategy for implementing a basic form of intelligence at this level was through a series of domain rule sets that evaluated the current context information, obtained from controlled wireless sensor nodes against some predefined conditions, and triggered a reactive action when those conditions were met. This scheme enabled Flexeo to take rapid local decisions at the concentrator, when all the information required to take those decisions was provided by the directly controlled sensors, while still allowing global intelligence at the Supervision Layer, where these data were integrated with the rest of the information sent by concentrators in other locations.

4.3 Supervision Layer

The Supervision Layer offers a basic graphical interface for monitoring the system, as well as an API for integrating the information into the current business management solution. Using the basic graphical interface (see Fig. 2), users can retrieve the current values of the sensors filtering them by the device type and/or the concentrator ID (each sensor measure has a timestamp associated). Users can also retrieve the alerts generated in one concentrator; these alerts are generated by the domain rules programmed in each concentrator. Finally it is possible to retrieve a histograph with the values of a sensor over a period of time. This graphic can be customized filtering the sensors by concentrator and/or type and configuring the period of time to be represented in it.

The Flexeo Web Interface was developed using Google Web Toolkit (GWT) [19], based on AJAX technology. The applications are developed in the Java programming language and GWT transforms them into HTML and JavaScript code. GWT applications have shorter load times, the generated code is more compact, and applications are supported by Internet Explorer, Firefox, Opera and Safari browsers.

Intelligence at the Supervision Layer

The Supervision Layer has a global view of the whole system, centralizing the data provided by all the concentrators that manage the low-level sensor nodes at the different locations. As already mentioned, we also developed a basic API that enables easy integration of gathered data with any ERP, information system or database. In this way, organizations can benefit from the Flexeo monitoring system completely integrated with their current business monitoring or business intelligence solution.

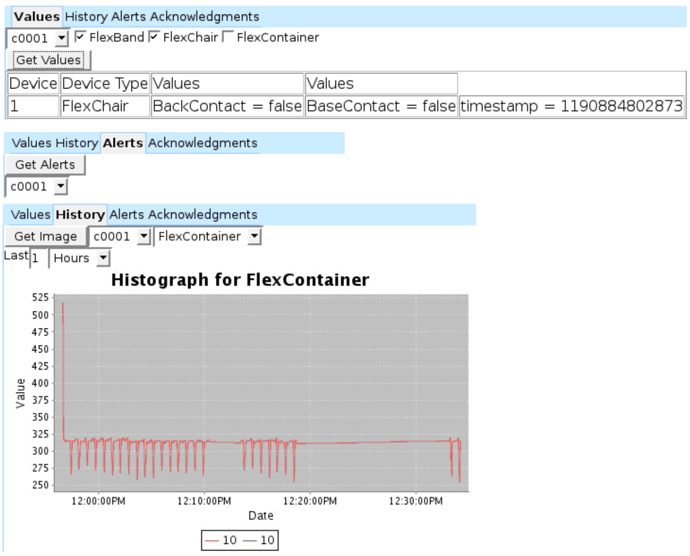

Fig. 2. Flexeo Web Interface at the Supervision Layer

5 Prototype devices and scenarios

We have designed several scenarios and devices using the Flexeo architecture, mostly for industrial monitoring and user activity monitoring. These are some of the prototyped devices, integrating wireless sensor nodes and providing data that flowed up all the levels in the architecture:

- *FlexContainer*: this device monitors the state of a chemical fluids container, by detecting the level of liquid. For experimental purposes, we assumed that the contained fluid conducted electricity, so we applied a low intensity electric current to the fluid and detected when the circuit is closed. This sensor node device was based in a MDA-100 device, which also provides temperature and light-intensity values.
- *FlexChair*: this sensorized chair includes a Mica2 mote and two FlexiForce sensors (see Fig. 3). These sensors located at the seat and back of the chair are

able to detect the user's position while seated. The FlexiForce sensors provide an intensity based information while the Mica2 wireless node only accepts voltage-based inputs, so in order to solve this problem a transformation function was developed and implemented in the sensor board. This information may be used for tracing the ergonomic behavior of users in a simple way: detecting whether they are leaning back often, or if they remain seated for too long without taking a small break, generating recommendations that may benefit their health.

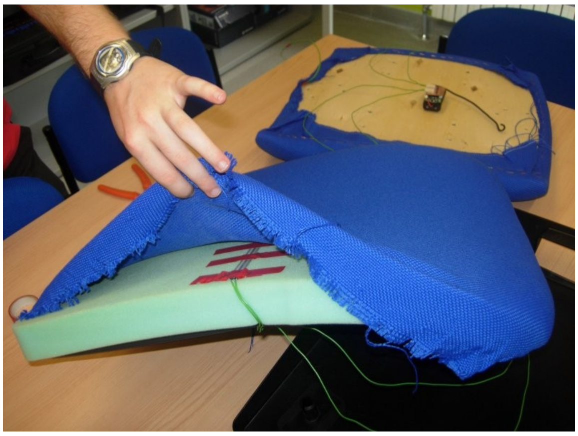

Fig. 3. Prototyping FlexChair

- *FlexDisplay*: this device is a bracelet with a display incorporated and wireless capabilities used to visualize the alarms of the system. It is based on a Mica2Dot mote, the smallest available wireless sensor node from Crossbow. The selected screen for displaying information is the uOLED64, developed by 4D-Systems. This screen has a USART interface compatible with any microcontroller. In order to manage the screen a driver was developed in the Mica2DOT node with three main functions: controlling the synchronization and the start status, controlling the image and text renderization and controlling the display visualization mechanism (through new functions such as DisplayOnOff, FadeToBlack and FadeFromBlack). The FlexDisplay can be used by security personnel or other monitoring staff in order to receive visual alerts about any facility being monitored, as well as further instructions from the monitoring center, via this energy-efficient graphical display.

6 Conclusions and future work

Wireless sensor networks are a very popular technology for monitoring user activities and implementing context-awareness scenarios. We introduced an architectural model for Internet monitoring of WSN, based on three layers that enable intelligence distribution and decision taking at different levels. One of the most remarkable aspects of Flexeo is its flexible application-agnostic orientation, which allows the system to be used for industrial purposes, health-at-home monitoring, human activity tracking or even environmental monitoring. We integrated state-of-the-at technologies such as wireless sensor networks, embedded computing platforms (Gumstix) and high-level programming kits (Google Web Toolkit) in a single coherent system.

Our future research will consist on moving more intelligence to lower-levels of the architecture to make it even more independent from the upper layers, particularly enhancing the intelligence at the wireless sensor network nodes by using fuzzy-logic and neural or neuro-fuzzy computational schemes, and semantic annotation of the gathered data.

Acknowledgements. This work has been partially supported by the Biscay Regional Government, Ekinberri program. The authors would like to thank all the people that took part in the development of Flexeo.

References

1. Zigbee Alliance, "Zigbee specification v1.0 ", http://www.zigbee.org/), 2004.
2. Sohraby, K., Minoli, D., Znati, T. "Wireless Sensor Networks: Technology, Protocols, and Applications", Wiley-Interscience, 2007.
3. ITU. "ITU Internet Reports 2005: The Internet of Things", ITU, 2005.
4. M. Beigl and H. Gellersen. „Smart-Its: An Embedded Platform for Smart Objects". In Proc. Smart Objects Conference (SOC 2003), Grenoble, France, May 2003.
5. Sun Microsystems. "Project Sun SPOT", http://www.sunspotworld.com/, 2008.
6. Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J. "Wireless sensor networks for habitat monitoring". In Proceedings of the 1st ACM international Workshop on Wireless Sensor Networks and Applications. WSNA '02. 2002. ACM, New York, NY, 88-97.
7. David Malan, Thaddeus Fulford-Jones, Matt Welsh, and Steve Moulton. "CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care". International Workshop on Wearable and Implantable Body Sensor Networks, April 2004.
8. A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, and J. Stankovic. "ALARM-NET: Wireless sensor networks for assisted-living and residential monitoring". Technical Report CS-2006-11, Department of Computer Science, University of Virginia, 2006.
9. J. Luprano, J. Sola, S. Dasen, J. M. Koller, O. Chetelat, "Combination of Body Sensor Networks and On-Body Signal Processing Algorithms: the practical case of MyHeart project,", pp. 76-79, 2006.
10. Vazquez, J.I., Lopez-de-Ipina, D. "Social Devices: Autonomous Artifacts That Communicate on the Internet". The Internet of Things. Lecture Notes in Computer Science. Volume 4952, 2008.
11. Kansal, A., Nath, S., Liu, J., and Zhao, F. 2007. "SenseWeb: An Infrastructure for Shared Sensing". IEEE MultiMedia 14, 4 (Oct. 2007), 8-13. DOI= http://dx.doi.org/10.1109/MMUL.2007.82
12. Gumstix Inc. URL: http://www.gumstix.com/. April, 2008.
13. OSGi Alliance. URL: http://www.osgi.org/. April, 2008.
14. Concierge OSGi – An optimized OSGi R3 implementation for mobile and embedded systems. URL: http://concierge.sourceforge.net/. April, 2008.
15. Equinox OSGi. URL: http.//www.eclipse.org/equinox/. April ,2008.
16. Knoplerfish Open Source OSGI. URL: http://www.knoplerfish.org. April 2008.
17. Oscar OSGi. URL: http://oscar-osgi.sourceforge.net. April 2008.
18. NanoHTTPD server. URL: http://elonen.iki.fi/code/nanohttpd/
19. Google Web Toolkit. URL: http://code.google.com/webtoolkit/. April, 2008.