Do-it-Yourself Smart Experiences
ITEA 2 project 08005

Requirements specification and state-of-the-art
D2.1

Editor:
Universidad Politécnica de Madrid, Spain

Contributors:
Vicente Hernández Díaz, UPM
Mario Lopez-Ramos, Thales
Guillermo Miranda Álamo, I&IMS
Diego Cansado Mansilla, UAH
María Ángeles Sanguino González, ATOS Origin
Yacine Gharmi-Doudane, ENSIIE
Claudio Forliviesi, ALU
Marisa Escalante, ESI

Security: Public
Version: 1.2
Date: December 22, 2009
Number of pages: 81

History

Version	Date	Person, Partner	Comment
0.1	07.10.2009	Vicente Hernandez, UPM	TOC for WSN.
0.2	12.11.2009	Mario Lopez-Ramos, Thales	Global TOC, first iteration.
0.3	13.11.2009	Marisa Escalante, ESI	TOC Modification
0.4	13.11.2009	Miguel S. Familiar, UPM	TOC Modification and Abstract
0.5	16.11.2009	Diego Casado Mansilla, UAH	TOC Modification and UAH contributions
0.6	26.11.2009	Yacine Gharmi-Doudane, ENSIIE	TOC Modification
1.0	22.12.2009	Vicente Hernández, UPM	1st Draft for reviewing
1.1	24.04.2010	Vicente Hernández, UPM	2nd Draft for reviewing
1.2	03.06.2010	Mario Lopez-Ramos, Thales	Formatting updates

Abstract

This is the SoA of devices and actuators technologies, capabilities, drawbacks, innovative approaches and challenges that are closely related to DiYSE. Smart environment systems, for achieving its goals, must gather information about objects surroundings by means of sensors and must also be able to make such surroundings evolve to the desired conditions by means of actuators. This document shows present sensors and actuators technologies capabilities related to DiYSE as well as the challenges and requirement specification that DiYSE must meet.

TABLE OF CONTENTS

1 Introduction

1.1 Scope of this document and link with other deliverables

The DiYSE project has three State of the Art (SOTA) documents covering the different tools, techniques, methods and environments that may be used to provide a DiYSE platform. These documents present the same pool of elements from different points of view. Due to this SOTA partition, it will be needed to link some of the sections from one of the documents to other sections on some of the other two documents. This is really important in Section 3 of the D1.1 and D4.1 that will try to present the same topics from the requirements and the interfaces point of view: web technologies, mobile technologies, platforms, devices, etc…

In WP1 (Use cases and requirements), deliverable D1.1 will focus on which requirements will be covered by web technologies, mobile technologies, system platforms and toolsets, i.e. how the users will access the smart experiences by using these systems.

This document will present the SOTA of current applications, systems platforms and business models relating to DiYSE. This includes Ambient Experience applications, features of toolsets and the business models and ecosystems that are working at this moment in similar proposals.

The document also includes a PEST (Political, Economical, Social, Technological) analysis. It is important to know why people are motivated to produce and share services, devices, etc.

In WP2 (Interaction with the environment), deliverable D2.1 will focus on the state of the art of:

- electronic devices that can retrieve data from the users' environment and produce physical outputs,
- algorithms to extract information from them (such as identification or location) and the functionalities those devices can provide in DiYSE,
- networking technologies to interconnect them.

In particular, D2.1 will put a special emphasis on the following kinds of devices:

- existing ready-made devices available in the market,
- networks of tiny battery-powered programmable wireless sensors,
- open hardware platforms used by DiY hobbyists.

In WP4 (Interactive Experience Creation), deliverable D4.1 analyses similar elements to D1.1, but from the user/developer point of view: how the users will use the elements of the DiYSE ecosystem.

The review of existing application creation approaches will identify technologies that may be supportive for the envisioned creation of applications and services in smart spaces. It is expected that lessons learnt from methods empowering users in the world wide web to contribute content or even applications to communities may provide a good base. Also the issue of actually do-it-yourself versus do-it-together (or have the community do it for you) and crowdsourcing will be addressed. The document reviews how to create interactive experiences.

1.2 Objectives of work package 2 within the DiYSE project

The objective of the whole Do-it-Yourself Smart Experiences (DiYSE) project is to enable people to transform their everyday environment into a highly personalized meaningful communication/interaction experience that can span the home and city domains. The project aims to create a marketplace for user-generated application components, in which non-technically-skilled people can participate, re-using components designed by savvy users.

Within the DiYSE project, work package 2 aims at bridging the gap between software and the physical world. It will do so by enabling end users to connect to the system all kind of input and output devices and smart objects. They will enable to gather raw data from the environment, process it and enrich it to deliver meaningful information about surrounding phenomena and more generally interact with the environment.

Work package 2 has the challenge of enabling non-technical users to customize their environment by installing and configuring themselves devices they can buy or easily assemble. We will produce software and documentation both for expert users wanting to assemble hardware and customize software to tailor it for their usage, and also for non-technical users wanting to reuse final hardware and software components.

1.3 Objectives and structure of this document

The objective of this document is to study the state of the art on several technical challenges in heterogeneous Internet-of-things device interconnection and derive high-level requirements for the architecture design phase.

The document is structured as follows:

- Section 2 lays the context for non-technical users on the kind of devices we are addressing, the interactions they provide and the realistic scenarios where they can be used.
- Section 3 focuses on the technical challenges of the integration of heterogeneous devices in a plug-and-play fashion, so that they can be easily discovered, identified, configured, controlled, etc. no matter the network protocols they use.
- Section 4 deals with techniques and algorithms to extract meaningful information (such as identification or location) from raw data provided by devices and the way this information is delivered to applications.
- Section 5 zooms into a specific technology for the interaction with the environment, Wireless Sensor and Actuator Networks. In particular, we will identify the potential and the major challenges raised by its use in DiYSE applications, and analyze the associated requirements and state-of-the-art.
- Section 6 provides a brief survey on the existing Do-it-Yourself hardware platforms and identifies the technical challenges that need to be addressed to simplify the exposure of DiY devices in the DiYSE platform.

2 DiYSE devices and usages

This section aims at providing partners that are not involved in WP2 with information about the devices that may be available to the project and the kind of general functionalities that those provide. It will also identify a few representative scenarios involving devices, which will be used as guiding vectors for WP2 works.

2.1 Inventory of existing classes of devices

In order to provide a general view of the kind of devices that WP2 could integrate, a categorized list is provided below:

- Simple sensors (measuring level-based magnitudes):
  - Environment sensors: temperature, humidity, air quality, CO2...
  - Resource consumption meters: power, water, gas...
  - Building sensing: status of lights, appliances, doors, windows...
  - Health and wellness monitoring: blood pressure...
  - Sport and fitness
  - Geopositioning and tracking: GPS/Galileo (may be coupled with accelerometers)

- Simple actuators:
  - Building automation: controlling lights, appliances, heating, ventilation, air conditioning, shades...

- Multimedia sensors and actuators:
  - Sound: microphones, loudspeakers, speech/music recognition…
  - Image: photo camera, picture frame, face/gesture recognition…
  - Video: webcam, screen...

- Sensor-enabled mobile phones (Freerunner, Android phones, Limo phones, MMH-MultiModeHub)
  - location (GPS, network functionality)
  - acceleration (builtin accelerometer)
  - any device attachable to USB host port

- Computer and console controllers:
  - Voice-based: voice & speech recognition...
  - Motion-based: Wiimote, wearable sensors...
  - Image-based: motion detection, facial recognition, time-of-flight
  - Natural User Interfaces: Microsoft Surface, Xbox 360 Project Natal, NeuroSky...

- Other computer peripherals:
  - USB gadgets
  - Homebrew peripherals: Phidgets…

- Programmable standalone devices:
  - Homebrew devices: Arduino...
  - Programmable ambient devices: Nabaztag, Tux Droid...
  - Autonomous mobile robot kits: Lego Mindstorms...

- Ubiquituous wireless sensor networks
- Identification readers:
    - Bar codes, QR codes, RFID tags, NFC...
- IP Video cameras
    - HD [1].
    - Wireless with integrated microphone/speakers (Axis M1031-W) [2].
    - Domo with PTZ control (Axis P3301) [3].

2.2 Types of device usages and interactions

Devices involved in DiYSE applications (other than computers, smartphones and the like) generally provide functionalities that fall into one or more of the following categories:

- Data acquisition (or sensing): device is used to capture data from the physical world (e.g.: acceleration, temperature…) and provide information to the system.
- Actuation: device is used to act on the physical world (e.g.: controlling lighting, sound, movement…).
- Localization: specific type of data acquisition providing information on the location of an object, a person, etc. (e.g.: using a GPS receiver).
- Identification: specific type of data acquisition providing the unique identity of an object (e.g.: using barcodes, smartcards, RFID tags, etc.).
- User interaction: specific combination of sensing (and possibly actuation) intended for a user to deliberately interact with the system.

2.2.1 Data acquisition

Sensors are devices that measure a physical quantity, such as:

- Physical properties: temperature, pressure (including sound), light (including imagery), humidity, flow…
- Motion properties: position, velocity, angular velocity, acceleration…
- Contact properties: strain, force, torque, slip, vibration…
- Presence: tactile/contact, proximity, distance/range, motion…

Sensors can sample data periodically (eg: sound recording) or trigger events when a given condition is met.

For instance, IP cameras might be employed to read the lighting level, so that some component may ask “Is it dark?” and the reply would be true or false. They can also be used to recognise some objects (i.e.: a game where a child has to find a picture hidden in a room, and then show it to the camera so it can continue to the next step) and they can record and or analyse sound upon request (for example, recognise when someone is crying).

2.2.2 Actuation

Actuators are devices that transform an input signal into a physical property (such as light, sound, motion, etc.) often by triggering an electrical component (such as a lamp, a loudspeaker or a motor).

A typical DiYSE scenario is home automation, for instance controlling lights, appliances, heating, ventilation, air conditioning, shades…

Some IP cameras have integrated speakers that can be used to play sounds (waveforms). These can be used to play a given sound when a certain message is received. Finally, there are cameras with a small LED light that might be used to provide information to the user.

2.2.3 Localization

There are several devices for locating and positioning people and objects. Most of them use hybrid techniques.

For instance, mobile phones can be used for locating purposes and GSM / GPS devices can be used for locating people indoor and outdoor too. In the later case, when GPS signal is lost, GSM and last GPS position are used to locate people, some commercial devices such as Navento [4] and Senda GPS [5] use this approach. Smart phones supported with GPS, Bluetooth, digital compass and WIFI technology provide a variety of techniques to perform user location. Laptops, netPCs may use WIFI connection to locate users. In the other hand, there are other kinds of devices such as RFid tags and magnetic tags that can be applied for locating purposes. Finally QR-codes and others patterns can be also used to locate people and objects.

porque no siempre puedes estar cerca...
...pero si tenerlo localizado en todo momento...
...también desde tu móvil

Navento [4]

2.2.4 Identification

When the presence and identity of people and objects matters but their exact location does not, then the identification process appears. There are a lot of devices and techniques in order to identify people and objects, the most frequently used being RFID tags. Each tag provides information about the object or people who carry it. Some examples of use can be found in parking or room access control systems. Another kind of identification system uses visual recognition of some patterns. QR-codes can be used to provide identification information too.

In the other hand, wireless connection devices can provide identification information via the communication technology they use, i.e., Bluetooth, Zigbee, and WLAN can provide MAC addresses in order to identify devices or people who carry or wear them. Finally, smart and mobile phones can provide identification information depending on the cell they are connected at each time.

2.3 References

[1] Axis Communications. (2010, Feb.) Axis Q1755 Network Camera. [Online]. http://www.axis.com/products/cam_q1755/index.htm

[2] Axis Communications. (2010, Feb.) Axis M1031-W Network Camera. [Online]. http://www.axis.com/products/cam_m1031w/index.htm

[3] Axis Communications. (2010, Feb.) Axis P3301 Netwrok Camera. [Online]. http://www.axis.com/products/cam_p3301/index.htm

[4] Navento. Grupo Avanzit. (2010, Feb.) Welcome to Localization Era. [Online].

[5] SEINCO. (2010, Feb.) SANDA GPS. [Online]. http://www.seinco.es/public/index.php?pid=senda

3 Integrating devices in the DiYSE platform

This section lists the high-level requirements for the integration of devices in the DiYSE platform and identifies for each of them the existing standards and technologies.

3.1 Device connectivity

How to physically connect devices to a network, both using existing protocols for legacy devices and proposing

Directly connected to the Internet (IP) or using a gateway, or using a device controller behind a gateway:

- RFID, RuBee
- NFC (NBC)
- WSN (IEEE-based or Proprietary Standards)
- Bluetooth (WiBree)
- UWB (W-USB)
- Hybrids Communication Technologies
- ZigBee
- X10
- IEEE 802.11 a/b/g/n
- Ethernet
- WiMax
- USB
- IEEE 1394

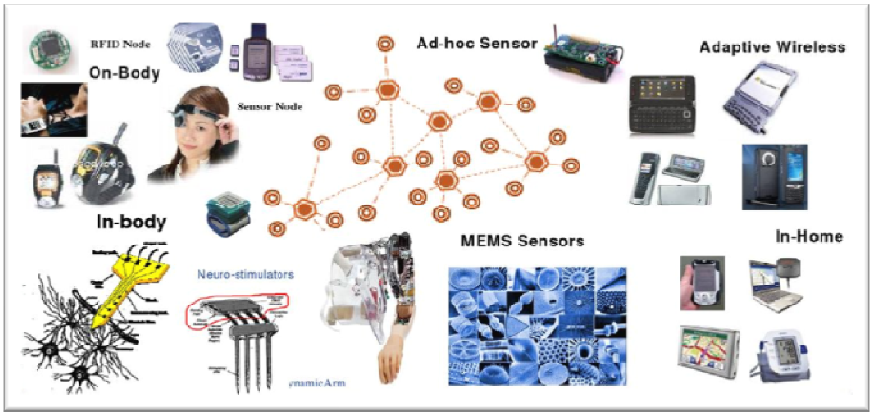


Figure 1 A set of heterogeneous devices using different communication technologies.

This subsection is devoted to overview the different communication technologies which could be used by devices in any of the proposed DiYSE scenarios. Despite on those scenarios most communication technologies are hard-wired (due in part to the lack of appropriate, reliable, and cost-effective wireless solutions), in this subsection we will only focus on wireless alternatives. The main advantages of a wireless solution are its ease of installation and deployment, the system flexibility, a dynamic network formation and the low cost which are mandatory issues for the proposed scenarios. At the end of the overview, we have included two comparative tables between some of the presented technologies which at present have become standards. Moreover, this subsection finishes with a survey of the most commonly used approaches to provide the coexistence and the interoperability of the related standards[1].

3.1.1 Wireless Sensor Networks Communication Standards

Generally Wireless Sensor Networks (WSNs) are originally standalone networks, where sensor readings are usually disseminated towards the sinks or gateways located in the boundaries. However, numerous WSN applications lead to the need of interconnecting WSNs to the external networks (e.g. Internet) to introduce WSN applications into different domains. The interconnection of WSNs with Internet or other communication networks also relaxes the control and management tasks of WSNs under dynamic changes of the application environment (see Section 3.8). Next we review the different WSN communication technologies and we briefly sketch their features towards their future interconnection:

3.1.1.1 IEEE 802.15.4

The IEEE 802.15.4 standard, designed specifically for remote monitoring and control applications, defines the characteristics of the physical and MAC layers for Low-Rate Wireless Personal Area Networks (LR-WPAN). The advantages of an LR-WPAN are ease of installation, reliable data transfer, short-range operation, extremely low cost, and a reasonable battery life, while maintaining a simple and flexible protocol stack [1].

Several leading radio manufacturers have implemented IEEE 802.15.4, which specifies a wireless link for low-power personal area networks. It is widely used in embedded applications, such as environmental monitoring to improve agricultural yields, structural monitoring to track building and bridge integrity, industrial control to provide more sense points and control points at lower cost. These applications generally require numerous low-cost nodes communicating over multiple hops to cover a large geographical area, and they must operate unattended for years on modest batteries. Such requirements target a very different set of applications than do WPAN technologies such as Bluetooth, which eliminate wiring for headsets, game controllers, and personal devices. Accordingly, 802.15.4's capabilities are more limited than other WPANs and WLANs – they have small frame sizes, low bandwidth, and low transmit power. Additionally, the microcontrollers typically coupled with LR-WPAN radios have limited memory and compute power. These constraints led many LR-WPAN vendors to embrace proprietary protocols and link-only solutions over 802.15.4 such ZigBee or open standards like 6LoWPAN.

3.1.1.2 ZIGBEE

The ZigBee Alliance [2] is an association of companies working together to develop standards (and products) for reliable, cost-effective, low-power wireless networking and it is foreseen that ZigBee technology will be embedded in a wide range of products and applications across consumer, commercial, industrial and government

[1] Note that on DiYSE environments there will be many heterogeneous devices which must communicate to each other as the Figure 1 shows.

markets worldwide. ZigBee was designed for reliable wirelessly networked monitoring and control networks.
This stack specification builds upon the IEEE 802.15.4 standard, hence it only defines the network and security layer, handling star and peer-to-peer network topologies, and providing a framework for application programming in the application layer.

3.1.1.3 6LowPAN

6LowPAN is the name of a working group in the internet area of the Internet Engineering Task Force (IETF). The 6LowPAN group has defined encapsulation and header compression mechanisms that allow IPv6 packets to be sent to and received from over IEEE 802.15.4 based networks [3]. Because of the potential of direct compatibility with the existing Internet infrastructure, 6LowPAN can be viewed as a significant factor in future sensor networks. It is also the most profound RFC [4] clearly breaking the OSI layered model and it exploits cross-layer information to minimise protocol overhead. It uses information in the link and adaptation layers to compress network- and transport-layer headers. The 6LowPAN may be connected to other IP networks through one or more border routers that forward IP datagrams between different media. Connectivity to other IP networks may be provided through any arbitrary link, including Ethernet, Wi-Fi, GPRS, or satellite as the next figure shows.

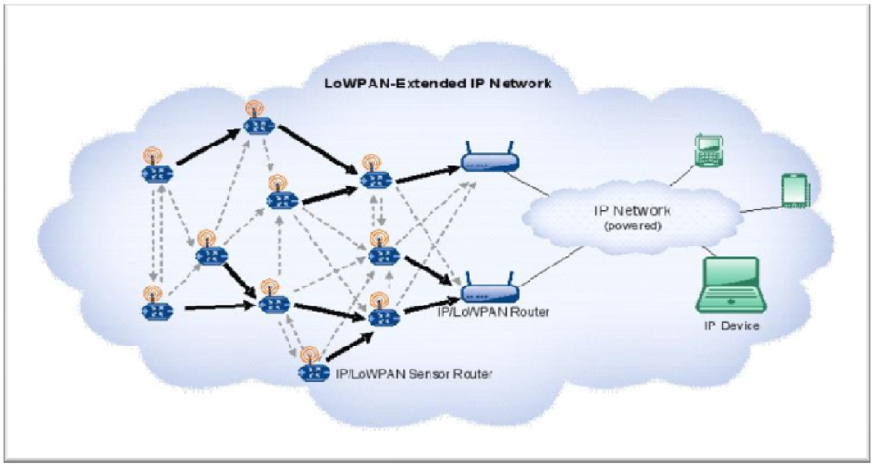


Figure 2 6LoWPAN gateway-less architecture.

3.1.1.4 Wireless-Hart

It is an open-standard wireless networking technology developed by HART Communication Foundation. The protocol utilizes a time synchronized, self-organizing, and self-healing mesh architecture and it currently operates in the 2.4 GHz ISM Band upon IEEE 802.15.4 standard but specifying new Data-link (including MAC), Network, Transport, and Application layers. W-HART was defined specifically for the requirements of process field device networks and industrial automation and to interoperate with the widely existing HART technology [5].

3.1.1.5 Z-Wave

It is a proprietary mesh network standard developed by ZenySys and standardized by the Z-Wave alliance (including Intel and Cisco) which is intended for home automation, residential and light commercial environments. Some of its applications embedded in consumer electronics products are remote controls, smoke alarms or security sensors such household appliances.

The Z-Wave RF system operates in the sub Gigahertz frequency range (900 MHz) and it is optimized for low-overhead commands and reliable communication [6].

3.1.1.6 ONE-Net

ONE-NET is an open-source standard which defines the physical and network layers for wireless networks. It was designed for low-cost, low-power control networks for applications such as home automation, security & monitoring, device control, and sensor networks. ONE-NET is not tied to any proprietary hardware or software, and can be implemented with a variety of low-cost off-the-shelf radio transceivers and micro controllers from a number of different manufacturers. ONE-NET is the only wireless control network that is based on the Open-Source philosophy (no royalties, freedom for use and modify, BSD license and Lots of design choices with open design standard). It operates in the 900 MHz band but additional frequencies are also possible [7].

3.1.1.7 Wavenis

It is a proprietary solution devised by Coronis System in 2001 but standardized as open by the Wavenis-OSA. Wavenis technology provides a platform to deploy Machine-to-Machine (M2M) applications such telemetry, industrial automation, home applications, etc. Its main strengths are ultra-low power consumption and long-range small amounts of data communications without the needed of Line of Sight (LOS) using the 900Mhz and 433Mhz band [8].

3.1.1.8 Dash7

DASH7 (ISO 18000-7) is a new, trade alliance with the goal of increasing the market size for ultra-low-power wireless products. Like ZigBee Alliance, DASH7 partners affectively address interoperability as well as the development of improved functions into the standard. DASH7's range of more than 1 kilometer (433Mhz), multi-year battery life, and ability to penetrate walls and water make it preferable in several WSN applications. DASH7 can be used with a variety of devices, from stand-alone DASH7 "tags" that monitor goods to mobile phones that allow consumers to monitor the energy usage in their own home [9].

3.1.1.9 Mi-Wi

Mi-Wi and MiWi P2P are proprietary wireless protocols designed by Microchip Technology that uses small, low-power digital radios based on the IEEE 802.15.4 standard. The technology aims to get low-data rate and short range distances by reducing the complexity of others WSN technologies (e.g. ZigBee) and reducing the footprint for constrained memory devices [10]. The environment of its applications are industrial monitoring, home and building automation, lighting control and automated meter reading.

3.1.1.10 INSTEON

INSTEON is a robust, redundant dual-mesh network that combines wireless radio frequency (RF) with the home's existing electrical wiring. INSTEON is less susceptible than other single band networks to the kind of interference and noise commonly encountered within the home (900Mhz band). It leverages the latest digital technology to create a true peer-to-peer mesh network. Because every INSTEON devices are flat, they do not require network supervision (network controllers and routing tables are not required). On the power-line, INSTEON devices are compatible with legacy X10[2] [11].

3.1.1.11 Bluetooth

IEEE 802.15.1 (Bluetooth) is another wireless link technology that falls under the WPAN classification. Intended to serve as a cable-replacement technology, Bluetooth supports relatively high throughput for a limited number of nodes within a small range. The key features of Bluetooth technology are robustness, low power, and low cost. The Bluetooth specification defines a uniform structure for a wide range of devices to connect and communicate with each other [12].

Bluetooth technology has achieved global acceptance such that any Bluetooth enabled device, almost everywhere in the world, can connect to other Bluetooth enabled devices in proximity. Bluetooth enabled electronic devices connect and communicate wirelessly through short-range, ad hoc networks known as piconets. Each device can simultaneously communicate with up to seven other devices within a single piconet. Each device can also belong to several piconets simultaneously. Piconets are established dynamically and automatically as Bluetooth enabled devices enter and leave radio proximity.

Bluetooth technology operates in the unlicensed ISM band at 2.4 to 2.485 GHz, using a spread spectrum, frequency hopping and full-duplex signal.

3.1.1.12 WiBree - Bluetooth Low Energy

It is a digital radio technology (intended to become an open standard of wireless communications) designed for ultra low power consumption (button cell batteries) within a short range (10 meters) based around low-cost transceiver microchips in each device.

Wibree is not designed to replace Bluetooth, but rather to complement the technology in supported devices. Wibree-enabled devices will be smaller and more energy-efficient than their Bluetooth counterparts. It operates in the same ISM band with a bit rate of 1 Mbit/s as its “big brother”. Main applications include devices such

[2] X10 is an international and open industry standard for communication among electronic devices used for home automation, also known as domotics. It primarily uses power line wiring for signaling and control, where the signals involve brief radio frequency bursts representing digital information.

as wrist watches, wireless keyboards, toys and sports sensors where low power consumption is a key design requirement [13].

3.1.1.13 Ultra Wide Band (UWB)

UWB is defined as any radio technology having a spectrum that occupies a bandwidth greater than 20 percent of the center frequency, or a bandwidth of at least 500 MHz. UWB communications allows for high data throughput with low power consumption for distances of less than 10 meters which can satisfy most of the multimedia applications such as audio and video delivery in home networking and it can also act as a wireless cable replacement of high speed serial bus such as USB 2.0 and IEEE 1394 [14]. It is touted as the next big thing for personal area networking where many devices are involved, low power is a must and high data rates are important (e.g. medical monitoring) [15]. Since UWB operates at very high frequencies it has very high penetration loss which will significantly affect the performance and size of the network nodes. Although UWB was claimed very low power initially in the literature, the attempts of such technology in the integrated circuits have exhibited power consumption more than that of the conventional narrowband short range wireless chips.

A major drawback to date with UWB has been the standards issue. In January 2006 the IEEE abandoned its efforts for standardization or the 802.15.3a Task Group (TG3a). The two groups developing UWB technology failed to come to agreement on a single solution.

3.1.1.14 Wireless Local Area Networks (WLAN) – Wi-Fi

Wireless Fidelity (Wi-Fi) includes IEEE 802.11 that is an evolving family of specifications for WLANs developed by the IEEE working group. There are several specifications in the family and new ones are occasionally added (e.g. 802.11a/b/g/n). All the 802.11 specifications use the Ethernet protocol and Carrier Sense Multiple Access with collision avoidance (CSMA/CA) for path sharing. The modulation used in newer 802.11 specifications is complementary code keying (CCK). The newer modulation methods provide higher data speed and reduced vulnerability to interference which permit users to surf Internet at broadband speeds when connected to an Access Point (AP) within the BSS (about 100m far) or in Ad hoc mode on the IBSS[16].

3.1.1.15 Radio Frequency IDentification (RFID)

RFID is the most pervasive communication technology mainly used in the called Internet of Things (IoT)[3], in fact, the International Telecommunication Union (ITU) considers RFID a key enabler of the IoT concept. That is because its good standardized status and low price of the simplest units.

Main components of RFID technology are a transponders, interrogators and middleware. A transponder (tag) is attached to or implanted in an object. Data is transferred between a tag and an interrogator (reader) device, one-way or on both

[3] The Internet of Things consists of everyday physical objects which have been given, at least, an electronic identity [31].

directions. RFID tags can be passive, semi-passive or active in nature, and information on tags can be read-only, read-write or rewritable[17].

The communication between RFID tag and reader is performed using magnetic inductance (contactless). This mechanism can also provide enough power to allow passive tags to operate without internal power source when they are exposed to the reader's magnetic field. Passive tags have very limited range, but some active ones can offer a range of up to 100 meters.

Information from the RFID objects is usually fed through some middleware application which forwards data to back-end systems for further processing or storage. Normally one is not going to access objects directly, but utilize the data through back-end applications.

3.1.1.16 Near Field Communication (NFC)

NFC is an extension of the ISO 14443, standardized in ISO 18092 and ISO 21481 using the frequency band of 13.56 MHz. NFC basically combines the operation of RFID transponder and interrogator (the reader) into one unit. The range of operation is very limited, about 10 centimeters (via magnetic field induction), which allows high density of objects in a given space so that they do not interfere with each other [18].

NFC is a combination of RFID contactless communication technology and wireless networking technology. The main difference between NFC and RFID is that in the NFC is possible to have a bidirectional transmission of information and NFC readers are primarily aimed at its usage in mobile phones. The principal applications of this technology are the peer-to peer communication between NFC enabled devices, payment and ticketing applications on mobile phones (this was one of the drivers for the creation of the NFC standard) and services or communication initiation[19].

3.1.1.17 Broad Band Technologies

These can be wired or wireless connections providing connectivity with remote services, central processing units or data aggregation facilities. Examples would include WSNs connected back to environment monitoring information servers which are located in a monitoring centre. Often this link is referred to as 'backhaul' and it is the data pipe that brings the sensed data back to a centre where it can be processed. Although narrowband solutions could be used if the data rates are low and network latency is permissible, the key breakthrough many times is the availability of Broadband technologies such as xDSL, WiMAX, 2.5 G (GPRS), 3G(UMTS) or satellite communications [20].

3.1.2 Comparison of Communication Standards

We consider that it is relevant to seamlessly compare the main well know wireless communication standards since it is important to get an idea of which of them we are going to use in the DiYSE scenarios. In fact, choosing a technology or another, we will likely obtain more or less advantages regarding coverage, topology, deployment, and so on. Selecting the most appropriate networking technology for a specific application can be challenging, and one size does not fit all. However, once the application’s communication requirements are clearly defined and the various attributes of the networking alternatives understood the most appropriate networking solution is usually easy to identify. In some cases, a hybrid approach may be the best option, as we later show in the interoperability subsection. For instance, a low-power, short-range subnet, such as ZigBee or 6LoWPAN aggregating sensor data for wide area communications across a GPRS/UMTS or Wi-Fi network. With a growing selection of wireless networking alternatives, users are no longer confined to wired installations, and with cost-effective and reliable wireless products emerging based on global standards, users are no longer restricted to proprietary wireless approaches. In Figure 3 a comparative graphic with the most well known wireless communication standards is presented.

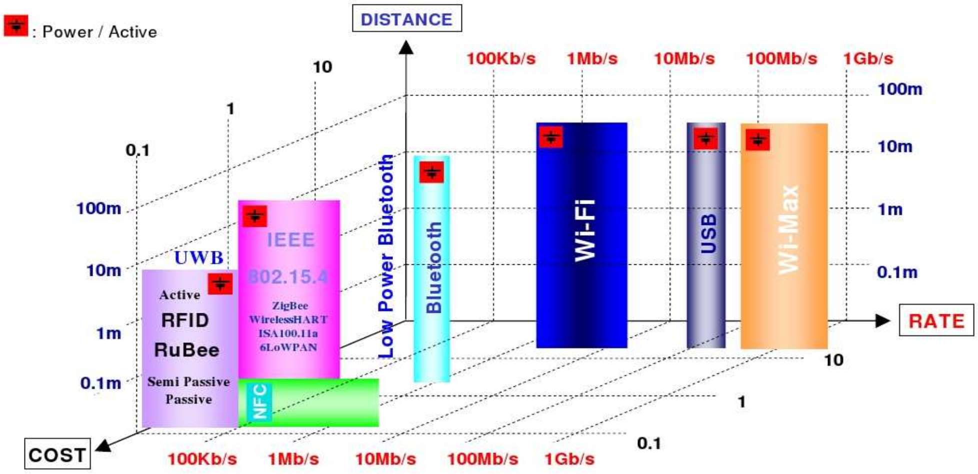
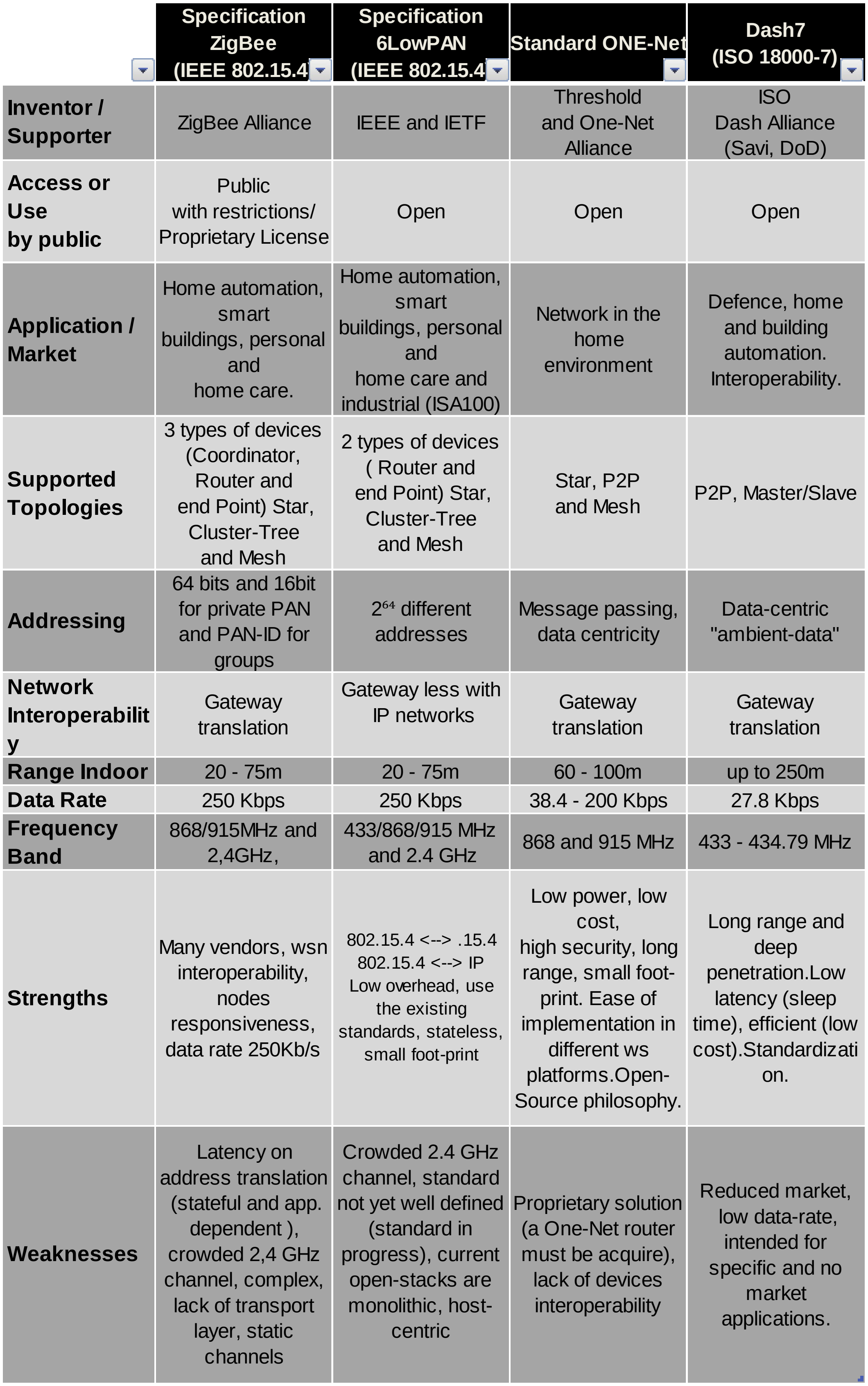


Figure 3 A comparative graphic (distance, cost and rate) of the wireless communication standards.

3.1.3 Wireless Sensor Networks

In section 3.1.1. we have presented an overview of the WSN communication technologies which could be found in the current market. As we described there are some of them which have become standards and others that are emerging quickly and are intended to get the standardization soon.

For sensor-based systems that require the flexibility of a wireless network, and which can tolerate modest message latency, users can select between proprietary and standards-based solutions.



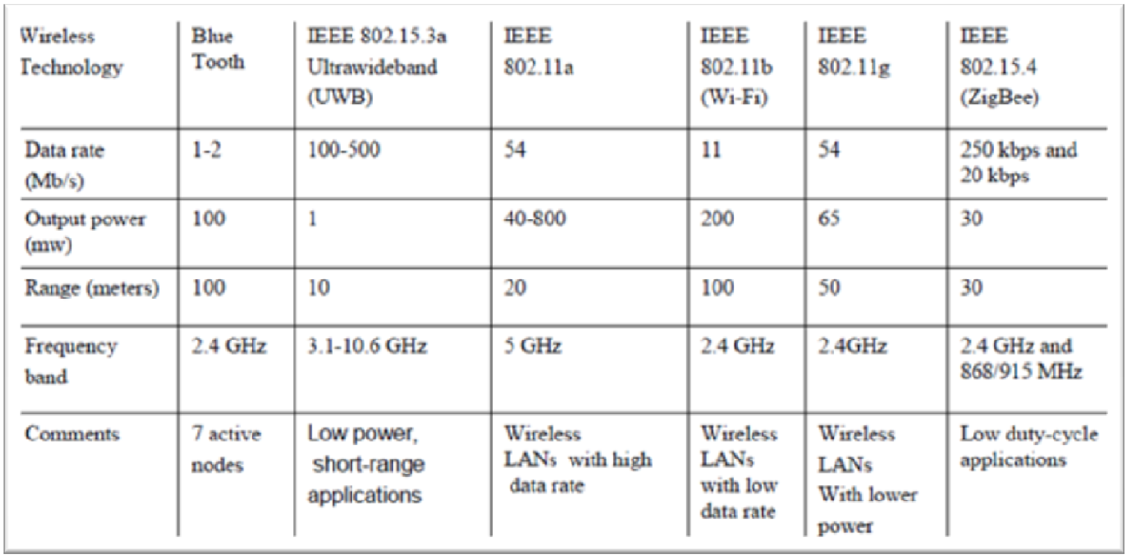
	Specification ZigBee (IEEE 802.15.4	Specification 6LowPAN (IEEE 802.15.4	Standard ONE-Net	Dash7 (ISO 18000-7)
Inventor / Supporter	ZigBee Alliance	IEEE and IETF	Threshold and One-Net Alliance	ISO Dash Alliance (Savi, DoD)
Access or Use by public	Public with restrictions/ Proprietary License	Open	Open	Open
Application / Market	Home automation, smart buildings, personal and home care.	Home automation, smart buildings, personal and home care and industrial (ISA100)	Network in the home environment	Defence, home and building automation. Interoperability.
Supported Topologies	3 types of devices (Coordinator, Router and end Point) Star, Cluster-Tree and Mesh	2 types of devices (Router and end Point) Star, Cluster-Tree and Mesh	Star, P2P and Mesh	P2P, Master/Slave
Addressing	64 bits and 16bit for private PAN and PAN-ID for groups	2^{64} different addresses	Message passing, data centricity	Data-centric "ambient-data"
Network Interoperability	Gateway translation	Gateway less with IP networks	Gateway translation	Gateway translation
Range Indoor	20 - 75m	20 - 75m	60 - 100m	up to 250m
Data Rate	250 Kbps	250 Kbps	38.4 - 200 Kbps	27.8 Kbps
Frequency Band	868/915MHz and 2,4GHz,	433/868/915 MHz and 2.4 GHz	868 and 915 MHz	433 - 434.79 MHz
Strengths	Many vendors, wsn interoperability, nodes responsiveness, data rate 250Kb/s	802.15.4 <--> .15.4 802.15.4 <--> IP Low overhead, use the existing standards, stateless, small foot-print	Low power, low cost, high security, long range, small foot-print. Ease of implementation in different ws platforms.Open-Source philosophy.	Long range and deep penetration.Low latency (sleep time), efficient (low cost).Standardization.
Weaknesses	Latency on address translation (stateful and app. dependent), crowded 2,4 GHz channel, complex, lack of transport layer, static channels	Crowded 2.4 GHz channel, standard not yet well defined (standard in progress), current open-stacks are monolithic, host-centric	Proprietary solution (a One-Net router must be acquire), lack of devices interoperability	Reduced market, low data-rate, intended for specific and no market applications.

Table 1 A comparison between four WSN communication standards

Since proprietary systems are usually customized to their application, they can offer benefits in transmission range, very low power consumption and per unit cost. However, they are not generally more secure than standards based systems, and their proprietary nature means that they can't achieve the high unit volumes and aggregated industry investment of standards-based systems [21]. Because the standardization is paramount to simplify and assure the broad use and applicability of the WSN technology, hereafter a comparison table of four of them (which we consider are probable candidates for the DiYSE scenarios) are presented[4]. In the table it has been showed their main technology features and on the bottom their weaknesses and strengths that should help us to decide which protocol we will apply in the each of the proposed scenarios.

3.1.3.1 Standards-based Wireless Networks

In the next table we roughly compare IEEE 802.15.1 (Bluetooth) wireless link technology, which falls also under the WPAN classification and IEEE 802.15.3 that pushes WPAN capabilities further, with greater throughput and support for more nodes. Although both are intended for battery operation, they only target lifetimes of several days to several weeks. In contrast, 802.15.4, which was before compared in **Error! Reference source not found.** is intended for low data-rate applications in which numerous nodes (up to 64,000 nodes) must be low-cost and have multiyear lifetimes on modest batteries. Finally, the IEEE 802.11 standards (including Wi-Fi which is designed to substitute wires between devices) is also included in the comparative table.



Wireless Technology	Blue Tooth	IEEE 802.15.3a Ultrawideband (UWB)	IEEE 802.11a	IEEE 802.11b (Wi-Fi)	IEEE 802.11g	IEEE 802.15.4 (ZigBee)
Data rate (Mb/s)	1-2	100-500	54	11	54	250 kbps and 20 kbps
Output power (mw)	100	1	40-800	200	65	30
Range (meters)	100	10	20	100	50	30
Frequency band	2.4 GHz	3.1-10.6 GHz	5 GHz	2.4 GHz	2.4GHz	2.4 GHz and 868/915 MHz
Comments	7 active nodes	Low power, short-range applications	Wireless LANs with high data rate	Wireless LANs with low data rate	Wireless LANs With lower power	Low duty-cycle applications

Table 1 A comparison between the most common wireless standards. Retrieved from [22].

In addition to this comparison, we can find many related studies in the literature. For ZigBee and Bluetooth, *Baker et al.* [23] studied their strengths and weaknesses for industrial applications, and claimed that ZigBee over 802.15.4 protocol can meet a wider variety of real industrial needs than Bluetooth due to its long-term battery

[4] Note that besides the previous presented approaches, there are many others communication solutions such as POPNet, SNAP, Mi-WI, EnOcean,Synkro, etc..but all of them are proprietary and we have discarded them on the SoA for such reason.

operation, greater useful range, flexibility in a number of dimensions, and reliability of the mesh networking architecture. For Bluetooth and Wi-Fi, *Ferro and Potorti* [24] compared their main features and behaviours in terms of various metrics, including capacity, network topology, security, quality of service support, and power consumption. In [25], *Wang et al.* compared the MAC of IEEE 802.11e and IEEE 802.15.3. Their results showed that the throughput difference between them is quite small. In addition, the power management of 802.15.3 is easier than that of 802.11e. Finally, *Jin-Shyan Lee et al.*[16] compared the four standards and shows its gains in terms of transmission time, data coding efficiency, protocol complexity and power consumption.

3.1.3.2 Coexistence

Since Bluetooth, ZigBee, 6LoWPAN, Wi-Fi and others use the 2.4GHz band, the coexistence issue between these standards must be dealt with. Basically, Bluetooth and UWB provide adaptive frequency hopping to avoid channel collision, while ZigBee and Wi-Fi use dynamic frequency selection and transmission power control. IEEE 802.15.2 discussed the interference problem of Bluetooth and Wi-Fi. Also, *Sikora and Groza* [26] provided quantitative measurements of the coexistence issue for ZigBee, Bluetooth, Wi-Fi, and microwave ovens. *Shuaib et al.* [27] focused on quantifying potential interferences between Zigbee and IEEE 802.11g by examining the impact on the throughput performance of IEEE 802.11g and Zigbee devices when coexisting within a particular environment. Regarding WSNs there are several studies towards the coexistence of this emerging technology. *Gang Zhou et al.* [28] devised solutions towards the cooperation of different WSN using the same crowded spectrum and *Musaloiu et al.* [29] studied the communication interferences caused by the exponential growing of 802.15.4-based WSN. In Figure 4 we can observe the range, band and data rate of the surveyed standards.

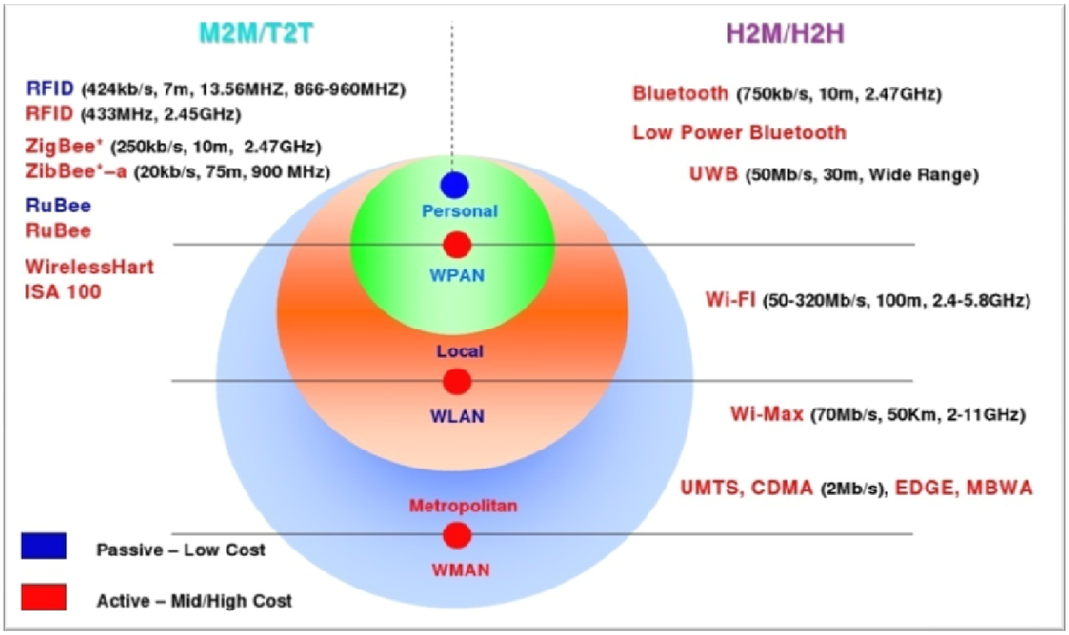

M2M/T2T
H2M/H2H
RFID (424kb/s, 7m, 13.56MHZ, 866-960MHZ)
RFID (433MHz, 2.45GHz)
ZigBee⁺ (250kb/s, 10m, 2.47GHz)
ZibBee⁺–a (20kb/s, 75m, 900 MHz)
RuBee
RuBee
WirelessHart
ISA 100
Bluetooth (750kb/s, 10m, 2.47GHz)
Low Power Bluetooth
UWB (50Mb/s, 30m, Wide Range)
Wi-Fi (50-320Mb/s, 100m, 2.4-5.8GHz)
Wi-Max (70Mb/s, 50Km, 2-11GHz)
UMTS, CDMA (2Mb/s), EDGE, MBWA
Personal
WPAN
Local
WLAN
Metropolitan
WMAN
🟦 Passive – Low Cost
🟥 Active – Mid/High Cost

Figure 4 A graphic with the wireless standards organized by their network used band. Retrieved from [30]

3.2 Device identification and addressing

3.2.1 IP Networking

Applications might access devices through the usage of hierarchical names, using naming systems such as DNS or LDAP. DNS in particular is interesting because it allows to record service and device descriptions along with addressing information.

Using DNS, it is very well possible to obtain information concerning a single device by making use of different names. This has two important benefits:

Different applications can make use of different naming schemes, so that naming hierarchies can reflect the logic of an application. This enables application to build names according to their requirements.

The same device becomes reachable using different categorizations. This means that it become possible to use different criteria for addressing a device. Location-based addressing for instance could compose DNS names by combining the names of different locations. Likewise, manufacturer-based addressing could compose the serial number of a device with the name of the manufacturer (e.g. the same webcam could be addressed by the following names: entrancecamera.mainbuilding.newyork.acme.org and sn0123456.cam_model_1234.manufacturer.org.)

It is possible to store device-specific information in DNS records such as:

> PTR Records: They are used to create an alias for a DNS name with another DNS name that contains the actual information. This is useful to map a device under several categories. Typically PTR names contain the network protocol used to communicate with a device. For instance, if a device implements a SOAP stack and is reachable as a HTTP server, it may have a PTR record under the name _http._tcp.domainname.org.
>
> A Records: These records are used to map the name of a device into an IP or IPv6 address. The same name could be mapped into several IP addresses, which is ideal for multihoming. In case of devices that are not directly connected to an IP network, this record will contain the address of the gateway through which it is possible yto reach the device.
>
> SRV Records: These records are useful to add extra service information beyond a name-address mapping. In fact SRV records may contain the TCP or UDP port number at which a particular service on a device is listening. This is useful whenever a device exposes more than one interface.
>
> TXT Records: These records contain text. An application or device can use them to store specific device description informations, or links to them. For instance, a service running inside a device could store a web link to its WSDL descriptor file. Applications that are based on onthologies could store links to their OWL descriptors.

3.2.2 Non IP Networks

Devices that do not have direct access to an IP network will need to be accessible through a gateway. The gateway must either be always on and enabled to receive incoming requests from the network, or may use existing network protocols to notifiy its devices to external parties. It could use a fixed scheme that maps each device to a different IP address/port/context path and provide the routing mechanism towards the internal network.

Typically it will have to detect changes inside the internal network (e.g. a wireless device connecting or disconnecting) and map them into the IP connectivity layer. In general it will need a mechanism to notify presence and reachability of the internal devices. Several techniques are available, such as:

- SIP: Session Initiation Protocol is a protocol that enables communicating parties to notify presence information and to establish/disestablish communication sessions

- DPWS: Device Profile for Web Services is a set of specifications that enables devices to embed web service interfaces on them. Moreover, DPWS contains a multicast-based discovery mechanism

- mDNS/Bonjour: it is another multicast-based discovery mechanism that is based on top of the DNS specification

- Dynamic DNS: It is possible to dynamically send updates to the DNS server(s) whenever the situation of the internal network changes. In this way the DNS information are always up-to-date.

- REST: Representational State transfer is an architectural style that describes how to treat and manipulate stateful resources in a way that is similar/compatible with the behaviour of the World Wide Web. REST can be used by gateways to provide Web Applications with a representation of the internal state of a device.

3.3 Device discovery, presence and lifecycle

Section 3.2.2 contains a list of protocols that may be useful to track the availability and lifecycle of devices, even when they are not directly connected to IP networks.
Normally each non-IP mechanism contains provisions specifying how to detect signals about the presence of a device. Device disconnection or unreachability is more difficult to track instead, and in this case applications must rely on keep alive mechanisms.

For instance, if Dynamic DNS is used to keep IP applications informed about the presence of a device, then every DNS update will have to contain a valid time to live entry that dictates when that DNS name has to expire. The Time-To-Live must be as close as possible to the duty cycle interval at which keep-alive messages are exchanged inside the internal network. The gateway itself could be stateless w.r.t. presence information, and could simply convert a keep-alive response message into a DNS update for a specific device. Whenever a device stops sending keep-alive messages its DNS entry will expire and applications will be informed that it has become unreachable.

Similar considerations apply to protocols like SIP. Instead, protocols that are based on multicast discovery schemes like DPWS need to provide a way to propagate multicast messages outside the internal network.

Finally, if REST is adopted, it become possible to create applications that do not need to rely on presence information. In fact it is possible under REST to create a stateful resource in the network which acts as a remote agent for a specific device. Under this model, the agent is always online and available for other devices and applications to communicate with even when its physical device is offline. It is also possible to express presence as a resource, if needed.

3.4 Device description and modeling

Approaches that are based on UDP like DPWS or DNS must use a way to describe a device that does not take too much space in terms of size. This is because the protocol adopted to send these information have inherent issues in sending large blocks of data. Under these models it may be useful to put the device description aside (for instance, as a file on a web server) and propagate only the link to it. This is what happens in DPWS Discovery, where a device sends only links to its description and not the description itself.

Approaches such as REST allow a resource or device to provide a link to its description in the device state itself.

The description of a device might differ from a protocol to another. Devices implementing SOAP interfaces may provide a WSDL file that describes the messages they accept/send. A REST exposure of a device, on the contrary, could be self describing using tags but in general would not require the definition of an interface and rather rely on self-describing data types (usually through the specification of a MIME-Type).

Devices Descriptions

- The *Devices description working group* has developed a core vocabulary to adapt content in Mobile Web. There are “Aspect” that are the type of components (device, web browser, network connection…) and “Properties” to refer a specific “Aspect”. The vocabulary refers to [1] is a set of properties to define two specific aspect “web browser” and “devices”. Some of this properties are “Vendor”, “Model”, “Version”, …

Devices Ontologies

- The *UWA Ontology* [2] is recommended to extend the above vocabulary with other Properties. This ontology has a “Device” class which represents a device in the deliverable context and some of its properties.
- *HYDRA Device Ontology* [3] The core device ontology contains taxonomy of device types and basic device and manufacturer information. The description

of the device properties and capabilities is divided into four interconnected modules:

- *Device Malfunctions Ontology*. It is divided in the taxonomy of type of error according to the severity (error, warning, fatal…)
- *Security Capabilities Ontology* represents the security properties of devices and the services, such as protocols, policies, mechanisms or objectives. Based on NRLOntology [4].
- *Device Services Ontology* presents the semantic description of device services on a higher, technology independent level. The HYDRA service model enables the interoperability between devices and services, employing the service capabilities and input/output parameters and may be automatically created by SAWSDL annotations [5].
- *Device Capabilities* represent the extended device information. The device capabilities are divided into three modules:
  - *Hardware* related device properties such as connection and communication protocols (e.g. Bluetooth or various network bearers, etc.), description of hardware interfaces (such as camera, display, etc.).
  - *Software* module includes various software platforms, operating systems, etc.
  - The state machine model representing the concepts of *states and transitions*, which are updated in the run-time and represent the device/service actual status.

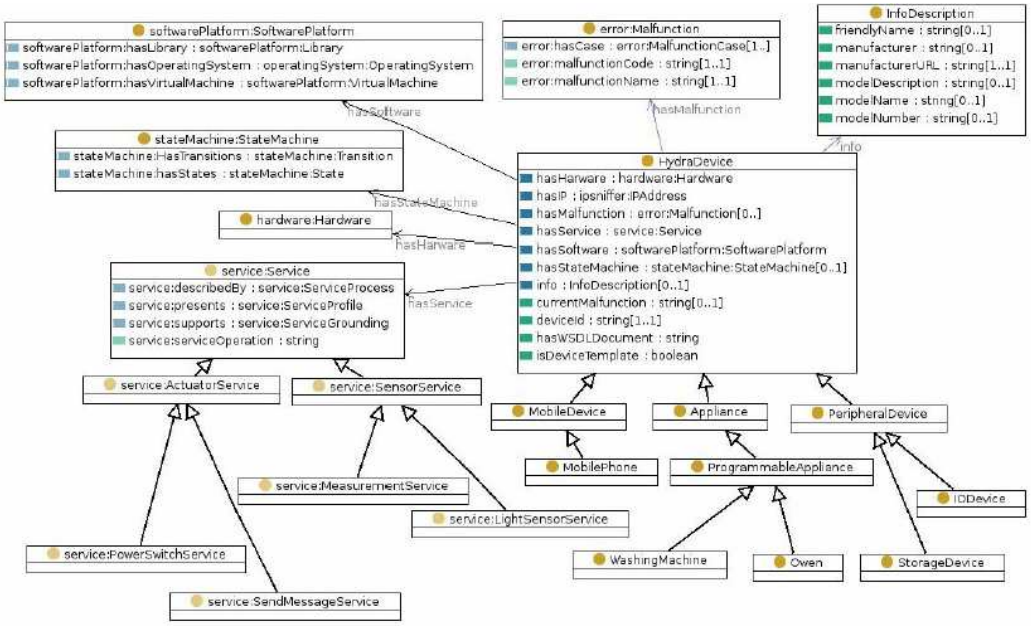
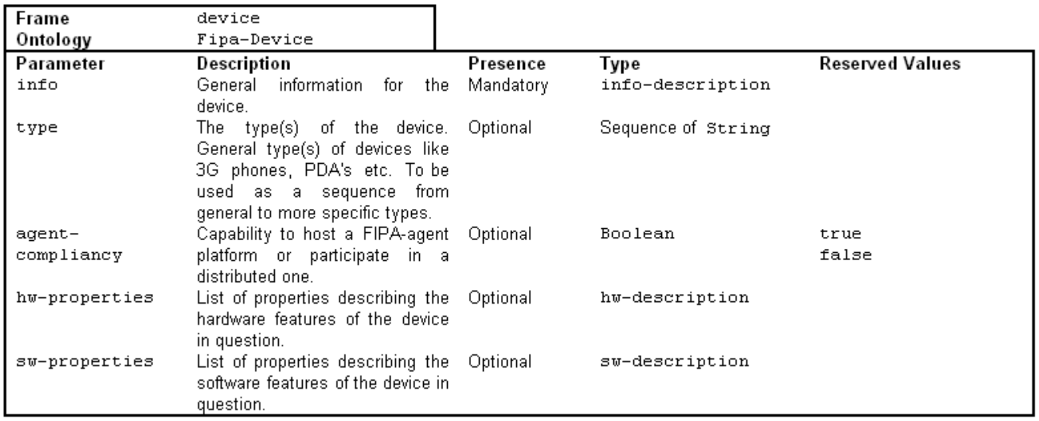

softwarePlatform:SoftwarePlatform
softwarePlatform:hasLibrary : softwarePlatform:Library
softwarePlatform:hasOperatingSystem : operatingSystem:OperatingSystem
softwarePlatform:hasVirtualMachine : softwarePlatform:VirtualMachine
error:Malfunction
error:hasCase : error:MalfunctionCase[1..]
error:malfunctionCode : string[1..1]
error:malfunctionName : string[1..1]
InfoDescription
friendlyName : string[0..1]
manufacturer : string[0..1]
manufacturerURL : string[1..1]
modelDescription : string[0..1]
modelName : string[0..1]
modelNumber : string[0..1]
hasSoftware
hasMalfunction
info
stateMachine:StateMachine
stateMachine:HasTransitions : stateMachine:Transition
stateMachine:hasStates : stateMachine:State
hasStateMachine
hardware:Hardware
hasHarware
HydraDevice
hasHarware : hardware:Hardware
hasIP : ipsniffer:IPAddress
hasMalfunction : error:Malfunction[0..]
hasService : service:Service
hasSoftware : softwarePlatform:SoftwarePlatform
hasStateMachine : stateMachine:StateMachine[0..1]
info : InfoDescription[0..1]
currentMalfunction : string[0..1]
deviceId : string[1..1]
hasWSDLDocument : string
isDeviceTemplate : boolean
service:Service
service:describedBy : service:ServiceProcess
service:presents : service:ServiceProfile
service:supports : service:ServiceGrounding
service:serviceOperation : string
hasService
service:ActuatorService
service:SensorService
MobileDevice
Appliance
PeripheralDevice
MobilePhone
ProgrammableAppliance
IODevice
service:MeasurementService
service:LightSensorService
service:PowerSwitchService
WashingMachine
Owen
StorageDevice
service:SendMessageService

Figure 1: Part of the HYDRA Device Ontology

- *FIPA Device Ontology Specification* for FIPA [6]. FIPA is an internationals organization to promoting the industry of intelligent agents by openly developing specifications. Device ontology specification is part of these specifications. The following table illustrates how define a devices with its most general properties.



Frame	device
Ontology	Fipa-Device

Parameter	**Description**	**Presence**	**Type**	**Reserved Values**
info	General information for the device.	Mandatory	info-description	
type	The type(s) of the device. General type(s) of devices like 3G phones, PDA's etc. To be used as a sequence from general to more specific types.	Optional	Sequence of String	
agent-compliancy	Capability to host a FIPA-agent platform or participate in a distributed one.	Optional	Boolean	true false
hw-properties	List of properties describing the hardware features of the device in question.	Optional	hw-description	
sw-properties	List of properties describing the software features of the device in question.	Optional	sw-description	

Figure 2: Device Description

Ontology-based Context Models. These models usually have a devices description as a fundamentally part of a context.

- An Ontology-based Context Model in Intelligent Environments [7].This context ontologies are divided into upper ontology (high-level ontology which captures general context knowledge about the physical World, figure 3) and domain-specific ontologies (collection of low-level ontologies which define the details of general concepts and their properties in each sub-domain).

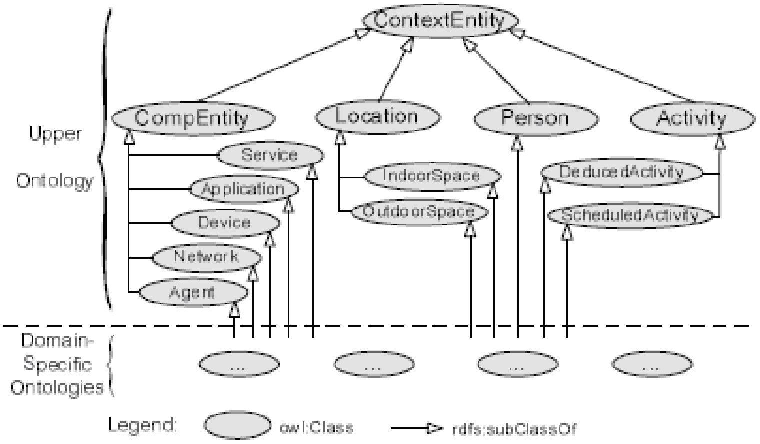


Figure 3: Class hierarchy diagram for our context ontologies

The following figure shows an OWL/RDF graph in one scenario where CellPhone-John and Fridge-Kitchen are type of the class Device and are related with other elements in the context.

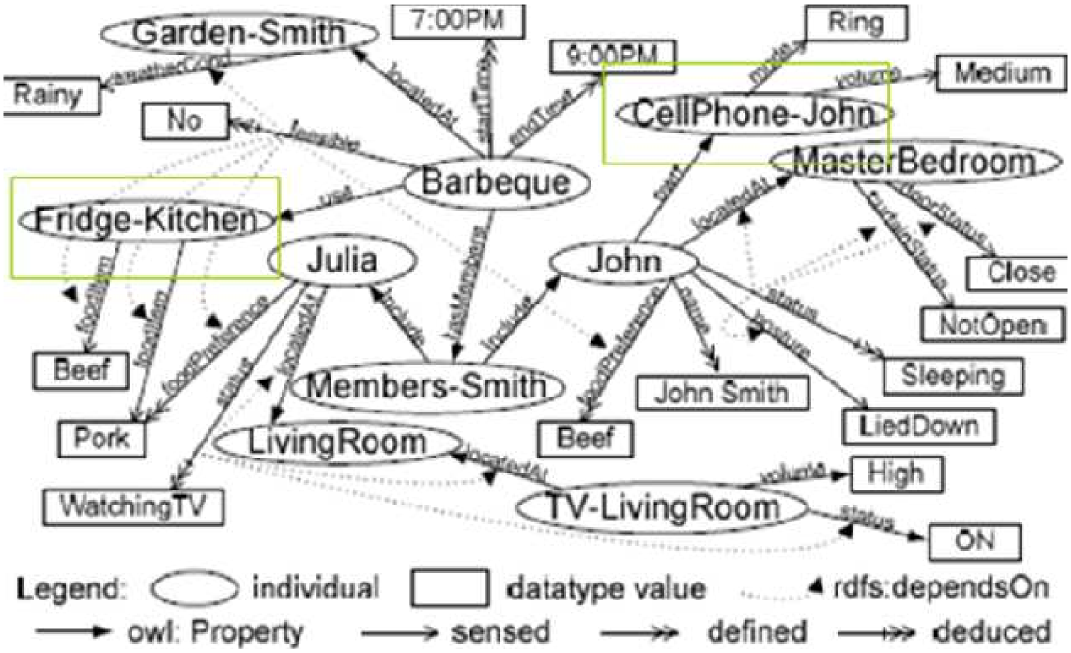


Figure 4: OWL/RDF Graph

- Context Studio [8]. Context Studio is an application personalisation tool for semi-automated context-based adaptation. Context Studio has context ontology, with an enhanced vocabulary model, is utilized to offer scalable representation and easy navigation of context and action information in the UI. The ontology vocabulary hierarchy is transformed into a folder-file model representation in the graphical user interface.

  Each context (object) is described using six properties: Context type, Context value, Source, Confidence, Timestamp, and Attributes [7]. Defining context vocabularies concerns defining sets of Context types and Context values

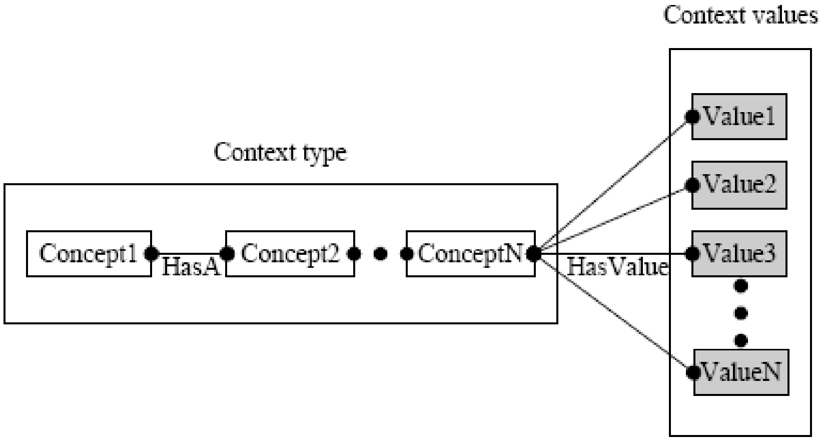


Figure 5: A model for creating vocabularies consisting of Context types and Context values

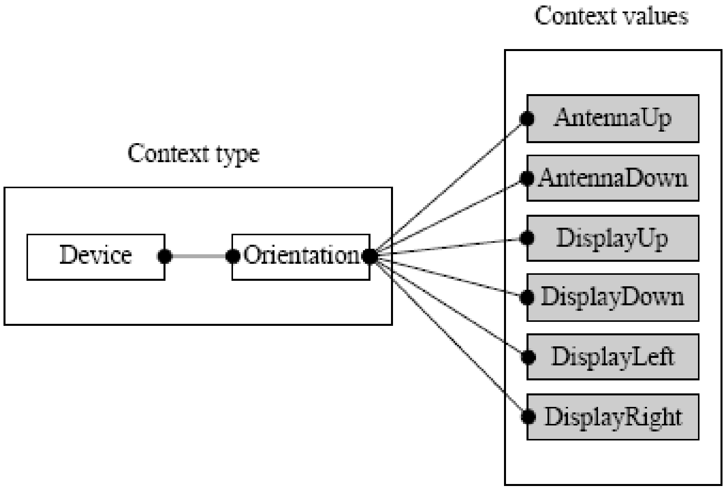
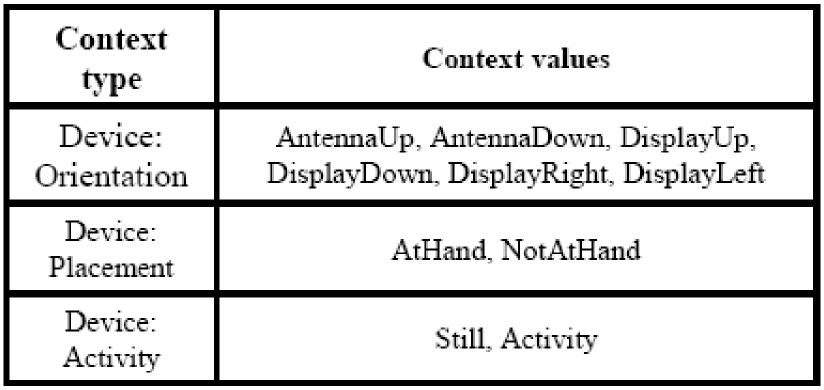


Figure 6: An example Context type and a set of Context values

The figure 7 presents an example of a context vocabulary describing device category contexts that are abstracted from acceleration and touch sensor data

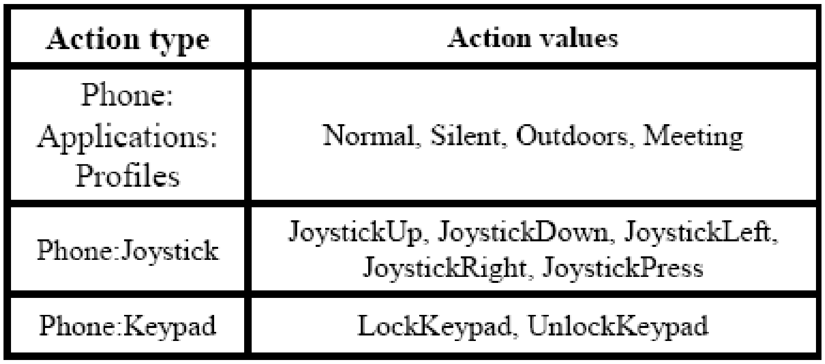


Context type	Context values
Device: Orientation	AntennaUp, AntennaDown, DisplayUp, DisplayDown, DisplayRight, DisplayLeft
Device: Placement	AtHand, NotAtHand
Device: Activity	Still, Activity

Figure 7: Device category sensor-based context vocabulary

Actions are defined with two properties, Action type and Action value, which describe actions as Context type and Context value describe contexts.

Figure 8 presents an example of an action vocabulary. Moreover, external devices can announce their actions, which can dynamically be included as Context Studio action vocabularies



Action type	Action values
Phone: Applications: Profiles	Normal, Silent, Outdoors, Meeting
Phone:Joystick	JoystickUp, JoystickDown, JoystickLeft, JoystickRight, JoystickPress
Phone:Keypad	LockKeypad, UnlockKeypad

Figure 8: Action vocabulary

3.5 Device data acquisition and control

- Flows:
    - Message-based
    - Stream-based
- Patterns:
    - Request/response
    - Publish/subscribe

3.5.1 Device-specific networking

When an IP network is set up, it is possible to retrieve data from and manipulate devices through their IP interfaces. Whenever devices do not support IP, it is necessary that a gateway in between the devices and IP networks understands and converts information between the two interfaces.

It is important to observe that the application space should not be polluted with issues which are specific to a certain connectivity technology or else the application gets locked into it. An example of this is a network of ZigBee sensors that communicate with an IP-based application through a ZigBee-To-IP gateway. If the data from the sensors are tunneled to the IP entity as they are they may contain ZigBee-specific information (like the strength of the radio signal) which are “just there” although they may not be related with the measurement data at all. This would require applications to be able to handle transport-specific data, and it should be avoided or the application would lock into that transport technology (ZigBee in this case).

On the other hand, some transport parameters should be made available for configuration through the IP interface. However the manipulation of these parameters should be made through a specific interface that should not be intertwined too much with the one used for data operations.

In general, pollution of the application space should be avoided especially when devices from different networking technologies are expected to communicate together (they will usually do that, either directly or indirectly through an IP network).

3.5.2 IP-based data acquisition and device control

At the IP level it is possible to use several protocol to exchange data and control information. The protocols described in the previous sections all fit well under different scenarios:

- SIP: the SIP protocol is ideal whenever data must be acquired in a conversational manner. It is the ideal when data producers operate by maintaining streams of information where clients subscribe and unsubscribe. Using the SIP protocol a stream producer could set up a SIP session and register itself into it as a stream producer. Consumers could just subscribe to that stream through a SIP URL. The SIP URL of a given stream could, if not previously known, be registered under a DNS name using a DNS TXT record.

- XMPP: The XMPP protocol is ideal when data and control data structures are sent using messages. Instead of a session-based communication, XMPP favours a one-to-one approach, where endpoints send messages directly to each other. XMPP URLs resemble those of emails, and, like emails, it is possible to keep messages in a storage memory when an endpoint is offline. Therefore XMPP is better suited for loosely-coupled interaction, where device control may happen asynchronously.

- SOAP: SOAP is a protocol used to exchange data with a web service. It is possible (using software stacks like DPWS) to embed web service interfaces within devices. In case SOAP is used a device will expose its interface through a description language called WSDL. This description language relies on XSD (XML Schema Definition) in order to describe the structure of the data that can be exchanged with a device. Given the client-server nature of SOAP, it is ideal for devices that can behave like servers, meaning that have a very high availability and are capable of servicing a great amount of traffic. However, given the faulty nature of hardware, SOAP clients should not assume the services to be always up and running, and provide a buffer solution for downtimes.

- REST: With REST, it is possible for every device to maintain its own resources inside the network, and to store its exported data over there. This has the advantage that data is available even when a device is offline. Likewise, control information could be put inside a resource by a device-controlling application, and have the device periodically read them (or being notified through HTTP Server Push), especially after it goes online. Therefore REST is a compelling solution when having to manage loosely-coupled systems, where the different system components and devices have different uptimes.

3.6 Device authentication and authorization

3.6.1 Authentication and authorization in a IP network

IP-based security is a broad topic that has been covered extensively over the years, and here we provide just the basics of it.

Almost all of the authentication and authorization technologies in the IP world are based on data encryption. This in turn is based in general on PKI (Public Key

Infrastructure) where a trusted certificate authority provides the infrastructure to create and verify digital signatures of messages. This enables entities to understand whether a message is original or fake, and to protect messages with a varying degree of protection.

Authorization is in general based on roles. Every entity in the system has one or more roles associated with it (which are normally stored somewhere, most of the times in a LDAP repository) which determines which actions it can performs and where. This allows devices to check whether a request message is coming from a trusted source, and if that source has the rights to ask that request to be performed. Also, the device will be able to encrypt the response in a way that only the intended destination will understand.

3.6.2 Non-IP networks and security

Particular care must be done at the boundary between IP and non-IP networks. In fact, most of the times the security mechanisms that have been in place in an IP network won’t work outside of it.

A possible way to ensure that end-to-end security is achieved is to encrypt messages at the level of the native device network (USB, ZigBee, X10 and so on) using the same PKI infrastructure in use at the IP level. This ensures that data are encrypted end-to-end. However, this requires the gateway to act as a simple tunnel for the data packets, which means that there is the risk of polluting the application space with device-dependent data.

Another constraint to this solution is given by the limitations of the devices themselves. In fact most of them have limited memory and processing power, meaning that supporting encryption at the native level is not always possible. This makes in fact end-to-end security impossible. However, the problem could be mitigated if wired networking is used and the wired network is situated at a secure location. Too bad, most native technologies (a part IEEE 802.11, which comes with an advanced and flexible security stack) do not provide more than simple data integrity checks (like parity checks or CRCs) but with no built-in security. This implies that security at the native level would require additions to existing standards, which may be impractical, especially for the sake of interoperability.

3.7 Device interoperability

Devices could communicate between entities using different networking protocols only if a protocol translation occurs.

However, given the sheer number of connecting technologies, it is not scalable to think in terms of protocol-to-protocol conversion. Instead, a common language or framework needs to be set up. This framework is IP, which is the building block of every networking protocol on the Internet.

Schemes for translating native protocols into IP can be implemented at the gateway level as defined in section 3.5.2. However, as mentioned in section 3.5.1, exposure of device interfaces at the IP level must try to shield as much as possible the application space from items appearing at the “lower levels” of the communication stack, in order to ensure that no unnecessary dependencies between applications and devices are created.

Simple and inexpensive objects, as WSN devices are, will not be equipped with large-scale active intelligence, so the data they produce and the communication methods they use to transmit the data are also relatively simple. Objects may transmit information between similar kinds of objects quite easily, but when data needs to be transferred upstream into real Internet or through another neighbour network, it is not feasible to make every object handle all the burden of the TCP/IP stack, to give an example. However, repeaters (w.r.t broadband communications) could be used to amplify or re-transmit wireless transmissions to allow more distance between objects:

- **Gateways** are used to transfer simple data transmissions into more complex networks or systems, or between the systems;

- **Proxies** can be used to hide a network of objects under one identity and aggregate the traffic;

- **Middleware** applications can be used to relay the object data to back-end systems or databases [50].

On the other hand, if we desire to merge two wireless networks in order they can cooperate (e.g. interconnect WSNs with the Internet), there are many approaches in the literature that have proposed solutions. These approaches include:

1. **Application-level gateway** that is realized as the implementation of the function which is able to perform the protocol translation (dominant approach or internetworking WSNs with the Internet) [51][52];

2. **Overlay network** is usually built on top of the Internet, and uses late address binding to achieve the independence of the underlying bearer protocols and addressing schemes. As an example, the Internet and WSN interconnection can be done using a Delay-Tolerant Network (DTN) on top of the two networks [53];

3. **Modified TCP** allows running TCP/IP protocol suite directly in the WSNs [54]. Thus, this approach provides the interconnection between the WSN and the Internet without requiring any proxies or gateways (recall 6LoWPAN) [55][56][22].

Regardless these approaches are focused on merging Internet with WSN, they could be also applied to interconnect other wireless networks. Currently there are many efforts towards the micro and nanotechnology, but nowadays it is obvious that mobile devices cannot bear with many embedded communication interfaces (e.g. Wi-Fi + 802.15.4 + Bluetooth + ..) due to the high cost, devices size and batteries. Nevertheless there are several projects that aim to combine the wireless standards:

- On MIMOSA Project the personal mobile phone is chosen as interface to Ambient Intelligence and a gateway between the sensors (RFID), the network of sensors, the public network and the Internet [57].

- µSensorial is another project which combines the wireless technologies on three different ways for a cluster-based approach: single-gateway architecture, architecture using mobile sinks, architecture using mote-gateways [58].

- SENSEI project which combines several communication devices through the mobile phone or Body Sensors and their scopes are similar to ours [59].

- AlarmNet is another research project which integrates heterogeneous devices, some wearable on the patient and some placed inside the living space to monitor the ambient changes from an outer data centre (Bluetooth, tags, WSN, Wi-Fi, backbone) [60].

3.8 References

[1] Device Description Repository Core Vocabulary. [Online]. http://www.w3.org/TR/2008/NOTE-ddr-core-vocabulary-20080414/#sec-properties

[2] Delivery Context Ontology. [Online]. http://www.w3.org/TR/dcontology/#introduction

[3] Hydra Project: Applications of Semantic Technologies in AmI. [Online]. http://www.hydramiddleware.eu/hydra_papers/Applications_of_Semantic_Technologies_in_AmI.pdf

[4] Naval Research Lab. Nrl security ontology. [Online]. http://chacs.nrl.navy.mil/projects/4SEA/ontology.html

[5] Semantic Annotations for WSDL and XML Schema. [Online]. http://www.w3.org/TR/sawsdl/

[6] FIPA Device Ontology Specification. [Online]. http://www.fipa.org/specs/fipa00091/PC00091A.html#_Toc511707119

[7] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang, "An Ontology-based Context Model in Intelligent Environments".

[8] Using context ontology in mobile devices application personalisation. [Online]. http://ia.ucpel.tche.br/~lpalazzo/Classificar/Estudos/Ontology/Pervasive/Utilising%20Context%20Ontology%20in%20Mobile%20Device%20Application%20Personalisation.pdf

[9] P. Mockapetris, ISI, RFC 1035 DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION, November 1987

[10] P. Vixie, Editor, ISC, S. Thomson, Bellcore, Y. Rekhter, Cisco, J. Bound, DEC, RFC 2136 Dynamic Updates in the Domain Name System (DNS UPDATE), April 1997

[11] P. Harsh, R. Newman, CISE, University of Florida, A Hierarchical Multicast Session Directory Service Architecture, November 16, 2009

[12] R. Fielding, UC Irvine, J. Gettys, Compaq/W3C, J. Mogul, Compaq, H. Frystyk, W3C/MIT, L. Masinter, Xerox, P. Leach, Microsoft, T. Berners-Lee, W3C/MIT, RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1 June 1999

[13] Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, Anish Karmarkar, Yves Lafon, SOAP Version 1.2, 27 April 2007

[14] Erik Christensen, Francisco Curbera, Greg Meredith, Sanjiva Weerawarana, Web Services Description Language (WSDL) 1.1, 15 March 2001

[15] P. Saint-Andre, Ed., Jabber Software Foundation, RFC 3920 Extensible Messaging and Presence Protocol (XMPP): Core, October 2004

[16] P. Saint-Andre, Ed., Jabber Software Foundation, RFC 3921 Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence, October 2004

[17] J. Rosenberg, dynamicsoft, H. Schulzrinne, Columbia U., G. Camarillo, Ericsson, A. Johnston, WorldCom, J. Peterson, Neustar, R. Sparks, dynamicsoft, M. Handley, ICIR, E. Schooler, AT&T, RFC 3261 SIP: Session Initiation Protocol, June 2002

[18] Roy Thomas Fielding, Architectural Styles and the Design of Network-based Software Architectures, 2000

[19] C. Adams, Entrust Technologies, S. Farrell, SSE, March Internet X.509 Public Key Infrastructure Certificate Management Protocols, 1999

[20] K. Holger and A. Willig, Protocols and Architectures for Wireless Sensor Networks. John Wiley & Sons, 2005.

[21] Z. Alliance, "ZigBee Specification, "Document 053474r17"," April 2007.

[22] J. W. Hui and D. E. Culler., "6LoWPAN: Extending IP to Low-Power, Wireless Personal Area Networks," *IEEE Internet Computing*, vol. 12, no. 4, pp. 37-45, Jul. 2008.

[23] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," Network Working Group RFC 4919, 2007.

[24] T. Lennvall and S. Svensson, "A Comparison of WirelessHART and ZigBee for Industrial Applications," in *IEEE International Workshop on Factory Communication Systems, WFCS 2008.* , Dresden, Germany, 2008.

[25] M. Knight, "Wireless security - How safe is Z-wave?," *Computing & Control Engineering Journal*, vol. 17, no. 6, pp. 18-23, Dec. 2006.

[26] One-Net, "An Overview of the One-Net standard," [on-line] http://www.one-net.info/, Accessed: Decembre 2009.

[27] C. Systems, "Wavenis Technology Overview," [online] Accessed: November 2009 http://www.coronis.com/en/wavenis_technology.html.

[28] J. Norair, "Introduction to DASH7 Technologies," DASH7 Alliance [online] http://www.dash7.org/DASH7%20WP%20ed1.pdf, 2009.

[29] D. Flowers and Y. Yang, "MiWi Wireless Networking Protocol Stack ," Microchip Tech. [online], 2008.

[30] S. Technology, "Insteon the datails," Insteon [online] http://www.insteon.net/pdf/insteondetails.pdf, 2005.

[31] H. Monson, "Bluetooth Technology and Implications," Internet Communication [online] http://www.sysopt.com/features/network/article.php/3532506, 2001.

[32] BLUETOOTH-SIG, "What is Bluetooth low energy technology," [online] http://www.bluetooth.com/NR/rdonlyres/B6E6A7BE-F7FD-4EEA-A96C-158E77768961/0/WhatisBluetoothlowenergytechnology.pdf.

[33] IEEE-Std-1394-2008., "IEEE Standard for a High-Performance Serial Bus," ISBN 978-0-7381-5771-9, 2008.

[34] D. Porcino and W. Hirt, "Ultra wideband radio technology: Potencial and challenges ahead.," *IEEE Communication* , vol. 41, no. 7, pp. 66-74, Jul. 2003.

[35] J.-S. Lee, Y.-W. Su, and S. Chung-Chou, "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee and Wi-Fi," in *33rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Taipei, Taiwan, 2007.

[36] I. W. 2. K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, 2nd ed. Wiley, 2003.

[37] I. O. f. Standards, "ISO/IEC 18092: Near Field Communication -- Interface and Protocol (NFCIP-2)," ISO/IEC, 2005.

[38] I. R. &. T. plc, "Near Field Communication in the real world – white paper," nfc-forum [online] www.nfc-forum.org/resources/white_papers/Innovision_whitePaper2.pdf.

[39] M. Sauter, *Communication Systems for the Mobile Information Society*, 1st ed. Wiley, 2006.

[40] G. Karayannis, "Standards-Based Wireless Networking alternatives," *Intelligent Systems*, Dec. 2003.

[41] C. 7. E. Project, "Comparison of the internetwroking technologies," http://www.capsil.org/capsilwiki/index.php/Comparison_of_the_internetwroking_technologies.

[42] B. N., "ZigBee and Bluetooth: Strengths and weaknesses for industrial applications," *IEEE Computing & Control Engineering*, vol. 16, no. 2, pp. 20-25, Apr. 2005.

[43] E. Ferro and F. Potorti, "Bluetooth and Wi-Fi wireless protocols: A survey and a comparison," *IEEE Wireless Commun*, vol. 12, no. 1, pp. 12-16, Feb. 2005.

[44] X. Wang, Y. Ren, J. Zhao, Z. Guo, and a. R. Yao, "Comparison of IEEE 802.11e and IEEE 802.15.3 MAC," in *IEEE CAS Symp. emerging Technologies: Mobile & Wireless Commun,* , Shanghai, China,, May, 2004, pp. 675-680.

[45] A. Sikora and V. F. Groza, "Coexistence of IEEE802.15.4 with other systems in the 2.4 GHz-ISM-Band," in *IEEE Instrumentation & Measurement Technology Conference*, Ottaw, 2005, pp. 1786-1791.

[46] K. Shuaib, M. Boulmalf, F. Sallabi, and A. Lakas, "Co-existence of Zigbee and WLAN: A performance study ," in *IEEE/IFIP Int. Conf. Wireless & Optical Communications Networks*, Bangalore, India, 2006.

[47] G. Zhou, J. Stankovic, and S. Son, "Crowded spectrum in wireless sensor networks," in *3rd Workshop on Embedded Networked Sensors*, 2006.

[48] R. Musaloiu and A. Terzis, "Minimising the effect of WiFi interference in 802.15.4 wireless sensor networks," *Int. J. Sen. Netw.*, vol. 3, no. 1, pp. 43-54, 2007.

[49] O. Vermesan, "Ubiquitous Wireless Sensor Networks," SINTEF, Norway [online presentation], 2009.

[50] International Telecommunication Union, "The Internet of Things," 2005.

[51] J.-H. Kim, D.-H. Kim, K. H-Y, and Y.-C. Byun, "Address internetworking between WSNs and Intenet supporting Web services ," in *IEEE MUE'07*, Seoul, Korea, 2007.

[52] J. Kim and D. Choi, "Esgate: Secure embedded gateway system for a wireless sensor network ," in *IEEE ISCE* , Algarve, Portugal, 2008.

[53] A. Dunkels, J. Alonso, H. Voigt, Ritter, and J. Schiller, "Connecting wireless sensornets with TCP/IP networks ," *LNCS WWIC*, vol. 2957, pp. 143-15, Feb. 2004.

[54] A. A. Dunkels, J. Alonso, and T. T. Voigt, "Making TCP/IP viable for wireless sensor networks ," in *EWSN* , Berlin, Germany, 2004.

[55] M. D. e. al., "Making sensor networks IPv6 ready ," in *ACM SenSys'08*, Raleigh, North Carolina, USA, 2008.

[56] K. Mayer and W. Fritsche, "IP-enabled wireless sensor networks and their integration into the Internet ," in *ACM InterSense'06*, Nice, France, 2006.

[57] M. Project. Microsystems platform for MObile Services and Applications. [Online]. HYPERLINK "http://www.mimosa-fp6.com/" http://www.mimosa-fp6.com/

[58] uSWN. Solving Major Problems in MicroSensorial Wireless Networks. [Online]. HYPERLINK "https://www.uswn.eu/j/" https://www.uswn.eu/j/

[59] S. project. Integrating the Physical with the Digital World of the Network of the Future. [Online]. HYPERLINK "http://www.sensei-project.eu/" http://www.sensei-project.eu/

[60] AlarmNet. Wireless sensor networks for smart healthcare. [Online]. HYPERLINK "http://www.cs.virginia.edu/wsn/medical/" http://www.cs.virginia.edu/wsn/medical/

[61] G. Karayannis, "Standards-based wireless networking alternatives," *Sensors Magazine*, Dec. 2003.

4 Processing data from devices

This section addresses the challenges and state-of-the-art technologies for enriching raw data from heterogeneous devices to provide high-level information to WP3 reasoning.

4.1 Virtual sensors

On the proposed DiYSE scenarios there will be a huge number of heterogeneous devices (sensors, actuators, PDA's, smart phones, etc.). Several of them are resource constrained, specially the individual devices in a WSN. They have limited processing speed, storage capacity, and communication bandwidth [4]. The WSN are closely coupled to a changing physical world, then, the nodes forming the network will experience wide variations in connectivity and will be subject to potentially harsh environmental conditions. Thus, the sensed data could be lost. In order to avoid this issue, the referred nodes have substantial processing capability in the aggregate, but not individually, so we must combine their many vantage points on the physical phenomena within the network itself. This is one of the primal objectives of what we have named "**virtual sensors**". On the other hand, by using the combined information, the virtual sensors have also the capability of reasoning by using fusion techniques[5]. The aim of this latter case, is to offer to requester nodes a high and enriched information that is collected from the surrounding sensors.

It is important to stress that for us, a virtual sensor has all the properties of a real sensor, with respect to its capability to communicate the sensed data, but the information that it offers to a requester node is derived from information already processed in other surrounding nodes. However, the information derived from a virtual sensor itself can be treated in the same fashion as a real one. What is challenging of our proposal is that the fused information should be retrieved from many different and heterogeneous real sensors.

For instance, let us think to an actor that wants to know if a specific person is inside a room. Instead of interrogate every potential sensor inside the room (wasting time and resources –recall the constrained features of sensors), the actor only needs to retrieve the information that it desires from the virtual sensor. This virtual sensor which has better resources than others, has beforehand merged and combined the surrounding information (i.e. people presence, people identification, people localization, etc.) in order to expose straightforwardly the requested information and services to actors. Even more, the supposed actor should not need to get into the room to retrieve the referred information and therefore it should not have the necessity of discover every sensor instance within the room since the virtual sensor does its best for it.

The idea of merging heterogeneous sources to get an enrich information and the concept of virtual sensor are not new. In fact, we can find some related examples in the literature: *Stefano Piva et al.* [1] contend that the application of multiple and

[5] Data fusion has been defined as the seamless integration of data coming from disparate sources and related information from associated databases to achieve improved accuracies and more specific inferences than could be achieved by the use of a single sensor alone [5]. Currently, data fusion systems are used extensively for target tracking, automated identification of targets, and limited automated reasoning applications [2].

heterogeneous sensors offers several possible performance benefits over traditional single-sensor based approaches (cost, complexity, and interface requirements) and that there are many additional benefits, depending on sensor type, fusion methodology, and the environment the system is operating in. In their work, the authors provide a solution to allow for sets of heterogeneous sensors, namely CCD video cameras and WLAN 802.11b radio devices, to be integrated in order to extract biometric information (position and ID) regarding objects in a given environment of interest.

Grabowski et al. [3] address the coordination of multiple, heterogeneous robots by developing the concept of a “virtual sensor”. Robot teams have the advantage that they can collectively share information. Then, they are able to fuse range information from a variety of different platforms to build a global occupancy map that represent a single collective view of the environment. A virtual sensor is simply an abstraction of the team’s occupancy map [3].

Kabadayi et al. [7] have developed virtual sensors, which enable collecting low-level sensed data and transforming it to a more abstract measurement to relay to the user (in-network aggregation). Their virtual sensors abstraction connects users directly to sensors in the local environment. Virtual sensors can be deployed on small devices, operate independently of heavyweight infrastructure, and provide on-demand, real-time connection to information that enables users to complete their tasks quickly, safely, and with the best possible information.

The same authors in [6] and [1] define a kind of virtual sensors that abstract a set of physical sensors and the operations that are performed on them, providing a new way of extracting data from heterogeneous wireless sensors. Moreover, their virtual sensors also offer a way to tailor a generic sensing environment to specific applications, which will be especially necessary as sensor networks become more widespread and general purpose. We will follow their work trying to go beyond it since their contributions have a major impact for the DiYSE virtual sensors.

Also in the literature we have found virtual sensors from the point of view of their implementation. In [9], a set of programming abstractions that allow a programmer to interact with several nodes (specified in a declarative way) as if they were a single virtual node is presented. That is achieved by relieving programmers from the details of data collection, allowing them to focus on the application logic.

To conclude this section, in the literature we have also found approaches using the concept of virtual sensors towards the seamless interoperability of different communication protocols and to simplify the applications development for integrated services involving multiple types of sensors [10].

4.2 In-door localization

In-door localization is about to determine the node's location (position) within the network [11] .It means for a node to determine its physical position (with respect to some coordinate system) or its symbolic location. Localization is not only required to understand the sensed data in the spatial context of the WSN, but also for navigation, a key feature on mobile sensor networks.

In this section we will review the techniques to compute the node's localization by presenting initially the methods that are used to estimate the distance to beacon/anchor nodes[6] and then the types of signal and models used to get those estimations.

4.2.1 Estimation Distance Techniques

4.2.1.1 Time of Arrival (ToA)

This first technique uses the time of transmission, the known signal propagation speed and the time of arrival of such signal in order to compute the distance between two nodes in the WSN. This technique is simple and efficient but it presents problems such exact time synchronization among nodes, reflections, and overhead. However the main drawback is that it is difficult to precisely record the arrival time of radio signals, since they travel close to the light speed. Therefore, it works best with an acoustic source as is proposed in [12] and [40].

4.2.1.2 Time Difference of Arrival (TDoA)

This approach uses two different signals with different propagation speed (e.g. ultrasound and radio signal). To estimate the distance it computes the difference between signals arrival time. The signals propagation speed must be known but it improves upon the ToA by eliminating the need to know when the signal was transmitted. Problems: Limited coverage (3-15m) and high dependency of the density of nodes within the WSN (number of nodes and connectivity). However the performance could be improved with signal processing (noise, peaks, filtering, etc.) as it is demonstrated in [13] and [14].

4.2.1.3 Angle of Arrival (AoA)

It is a method for determining the propagation direction of a radio-frequency wave incident on an antenna array. The AoA determines the direction by measuring the Time Difference of Arrival (TDoA) at individual elements of the array -from these delays the AoA can be calculated – usually the antennas array uses a direction of reference. Problems: Calibration, expensive/energy-intensive hardware, high cost.

4.2.1.4 Received Signal Strength Indicator (RSSI)

The nodes which use this method send out a signal of known strength (using the signal attenuation) and after they use the received signal strength and the path loss coefficient to estimate the distance to the known location . Another use for RSSI is profiling, in which a map of RSSI values is constructed during an initial training phase. Sensors then estimate their position by matching observed RSSI values with the training data. Presented problems: Highly error-prone process, inaccurate estimations, transmission power should be known, low coverage (10%Error d < 20m) [15][16].

[6] The beacon or also called anchors are nodes which know their own position to help sensors to determine their position within the WSN. Those nodes are very used in the distance estimation techniques but it has its shortcoming. It does not scale well in large networks and problems may arise due to environmental conditions.

4.2.1.5 Received Signal Frequency

Recently, there have been several published techniques that determine the position of a node based on the observed frequency of a signal [17][18]. Signal frequency will undergo Doppler-shift when the transmitter and receiver are moving relative to one another. The observed Doppler-shift at multiple infrastructure nodes can be used to derive the position and velocity of the mobile node.

4.2.1.6 Signal Modality

In the last section we have reviewed the techniques to estimate the node location or its position within a wireless sensor network. In most of them it has been specified the use of different types and modes of signals for the estimation purposes. Next, we summarize these signals and the used methods as well as we address the research papers and projects where they have applied.

4.2.1.7 Acoustic Ultrasound

Ultrasound is a cyclic sound pressure with a frequency greater than the upper limit of human hearing that is approximately 20 KHz.

These systems use both ultrasound and radio signal. The localization is hence performed by calculating the difference of the ToA of the referred signals [19][19][20] and [21] use this method.

4.2.1.8 Infrared (IR)

Infrared radiation (IR) is an electromagnetic radiation with a frequency range between 1 THz and 430 THz. The infrared-based localization systems use infrared light to perform their calculations to get the node localization by measuring the characteristics of the received signals [22][23]. These systems require are constrained to the device Line of Sight (LoS) and a proper orientation to the transmitters.

4.2.1.9 Radio frequency (RF)

Radio frequency systems employ a transmitted signal that is received by some mobile devices within the network. These systems are commonly used since almost all type of WSN communication standard use this carrier. The RF is used in several estimation techniques such as the received signal strength intensity, angle of arrival, time of flight and after it is employed by triangulation to calculate the node’s position [24][25].

4.2.1.10 Visible light

Visible light communication systems can provide solutions to the frequency band communications since light is not interfering with the crowded radio frequency band. This technique is used in [26].

4.2.1.11 Acoustic Signals

These systems employ sound signals instead of ultrasound signal to avoid the signal attenuation. Usually these acoustic signals are used in distance estimation techniques that compare the differences on arrival of two or more signals with different propagation speed.

4.2.1.12 Specific Radio Systems

They use Ultra high Frequency (UHF) and Very High Frequency (VHF) signals to perform their calculations. UHF designates a range of electromagnetic waves with frequencies between 300 MHz and 3 GHz while VHF uses the radio frequency band from 30 MHz to 300 MHz [27].

4.2.1.13 Ultra Wide Band

Ultra-wideband (UWB) is a radio technology that can be used at very low energy levels for short-range high-bandwidth communications by using a large portion of the radio spectrum. The ultra wide band systems employ a high frequency signal, above 3 GHz in order to avoid Non-LoS and the multipath. Such systems use either RSSI or TOA to calculate the position by triangulation [28].

4.2.1.14 WLAN

A Wireless Local Area Network (WLAN) links devices through a wireless distribution method (typically spread-spectrum or OFDM radio), and usually provides connection to the Internet using Access Points (APs).

WLAN location systems employ the exiting Wi-Fi network infrastructure access points. The systems measure the received signal strength to perform calculations to get the estimated position of nodes within the range or cell. These systems must deal with multipath and Non-LOS problems. WLAN systems often use probabilistic models to mitigate such problems as is demonstrated in [29] and [30].

4.2.1.15 Bluetooth

Bluetooth is an open standard and a communication protocol primarily designed for the exchange of data in low power consumption networks with short range of communication (i.e. Personal Area Networks - PANs). In such networks (PicoNets) the Bluetooth localization systems use the received signal strength intensity to calculate the position of the devices inside the network [31]. Because the devices use a radio (broadcast) communications system, they do not have to be in the LoS of each other and the synchronization is avoided.

4.2.1.16 Zigbee

ZigBee is a specification for a suite of high level communication protocols using small, low power digital radios based on the IEEE 802.15.4-2003 standard for wireless personal area networks.
The systems which make use of this technology, as the Bluetooth ones, employ the received signal strength intensity to calculate the position of the mobile devices [32].

4.2.1.17 RFID

Radiofrequency identification (RFID) technique uses tags applied to or incorporated into something (objects, people, animals, etc.) for identification and tracking purposes using radio waves. RFID location systems deploy a number of readers in the area where the tags will surely move around. By reading and identifying the tags (Assuming that the readers are placed in a well known position) the distance or the placement of nodes estimation is calculated. The RSSI intensity is used again as is demonstrated in [33] and also in the *WhereNet project [34] .*

4.2.1.18 Artificial vision

Most of these systems use printed codes or patterns that can be recognized by Webcams or IP cameras. The code gives information about the position or the number of objects that could be in a room. Another proposed technique attached to artificial vision is the named “schema analysis”. By taking snapshots with a camera and the proper software, it is possible to determine the position of the objects in the schema or to know if a new object has appeared or has been detected [35].

4.2.1.19 Global Positioning System (GPS)

The well known GPS system is composed of three parts: between 24 and 32 satellites in Medium Earth Orbit, four control and monitoring stations on Earth and the actual navigation devices. GPS satellites broadcast signals from space that GPS receivers use to provide three-dimensional location (latitude, longitude, and altitude) plus the time.
Then a GPS receiver calculates its position by precisely timing the signals sent by the GPS satellites [36][37]. The main drawback of this technique is its limited use in outside environments. However with the appearance of systems like Assisted-GPS (A-GPS) the trend go towards the use of this technology also over indoor environments.

4.2.1.20 Magnetic tracker

These systems use magnetic readers and magnetic tags. By reading the magnitude of the magnetic field can calculate the relative position of the tag to the reader [38][38].

Using the distance estimation methods with different signal modalities previously sorted, it is possible to get accurate node's location within the WSN by applying any of the next position estimation techniques.

4.2.1.21 Trilateration

This first technique estimates a node position by computing the node's distances to three non-linear points with their location is beforehand known. To compute the distances to anchors, the trilateration make use of the previously reviewed distance estimation techniques. For a 2-Dimensions space, three anchors are enough but for 3-d position another anchor in a different plane is needed. This method is used in the several localization works such [12] and [39].

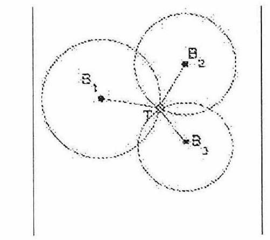


Figure 1: Trilateration - the three dash points permit to the white one (T) to compute its 2D position.

4.2.1.22 Trilateration – Multihop Range Estimation

When a node desires to compute its position but the anchors are not in the node's range, it makes use of its neighbor nodes as relays to reach the beacon nodes and eventually compute its current position. There are two methods to perform this computation:

1. Count the number of hops assuming that the length of one hop is known (DV-Hop). Start by counting hops between anchors and divide by the known distance;

2. If there exist a range estimate between neighbors, use them to improve the total length of route estimation in the previously defined method (DV-Distance).

4.2.1.23 Triangulation

It is similar to Trilateration, but rather than distances, angles are used to determine a node position. Generally, to compute the position two angles plus a distance between two well known anchors are needed (2 Dimensions - for 3 Dimensions also the azimuth is used). This estimation technique is very useful in a network of phased antennas array where different receptors with a well-known distances measure the ToA of the target signals and compute the angle.

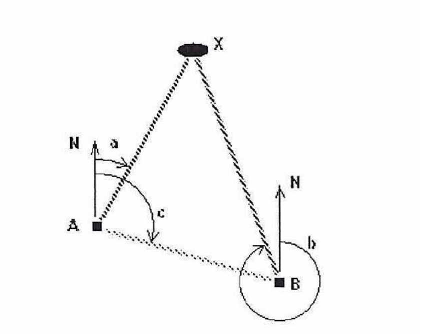


Figure 2: Triangulation - two angles and the distance between the two beacons.

4.2.1.24 Multilateration

Multilateration, also known as hyperbolic positioning, is the process of locating an object by accurately computing TDoA of a signal emitted from that object to three or more receivers. It also refers to the case of locating a receiver by measuring the TDoA of a signal transmitted from three or more synchronized transmitters. Multilateration should not be confused with trilateration, which uses distances or absolute measurements of ToA from three or more anchors.

4.2.1.25 Iterative Multilateration

It makes use of neighbor nodes to determine their position and then those act as beacons for other nodes. For instance, assume that some nodes can hear at least three anchors (to perform triangulation), but not all. Let more and more nodes compute position estimates and spread their position knowledge in the network. The infrastructure is created in an Ad-hoc manner but the problem mainly arises on errors propagation and high computation in every node.

4.2.1.26 Scene Analysis

Another technique used overall in the field of image and pattern recognition is the scene analysis where features of an observed scene from a particular vantage point are used to infer the location. It can be done in two ways:

1. Static: observations matched to features recorded in a database with corresponding locations;

2. Differential: examine differences between two successive scenes to calculate location.

The main drawback is that this technique requires compiling a database of features, thus extensive infrastructure.

4.2.1.27 Proximity-based.

This technique aims to determine when an object is near of a well-known location instead of compute the specific or accurate space locations. Three methods are used to determine such proximity:

1. Physic contact: it is the basic method using pressure, capacitance or touch sensors;

2. Monitoring or polling wireless Access Points (APs). This method is intended to know whether a mobile device is in one or more cells coverage within a wireless network;

3. Monitoring or observing identification techniques. This third method aims to combine the use of different identification techniques such as credit cards, logins, registers, and footprints, IDS to know the approximate node location.

It is important to remark that these three methods could be applied in any of the proposed DiYSE scenarios since most of them looks for objects or people tracking

4.2.1.28 Dead Reckoning

This technique is overall used in Robotics. Robots obtain their current velocity from wheel encoders or other means, and use this information in conjunction with the amount of time that has elapsed since the last update to derive current position and heading [41][41]. The major drawback of this approach is that the position estimation accrues error over time, primarily because of noisy encoder data due to uneven surfaces, wheel slippage, dust, and other factors [42].

Determining location or position is a vitally important function in WSN, but fraught with many errors and shortcomings:

1. Range estimates often not sufficiently accurate;

2. Many anchors are needed for acceptable results;

3. Anchors might need external position sources (GPS);

4. Multilateration problematic (convergence, accuracy).

In the DiYSE project we will aim to solve some of these issues and to go beyond the presented literature approaches. Nevertheless we will use and combine the basis of localization in WSN in the proposed scenarios to reach our targets.

4.3 Identification

A very important concern related with localization is people and objects identification. In many cases when a person or an object is located it is also identified but most of the times in an inaccurate way. There exits hence several techniques to locate and identify people and objects inside a WSN in a proper way.

The identification systems intended for monitoring and tracking people in both indoor and outdoor areas have become increasingly important. The efficient tracking of many people motion in either large buildings or outdoor areas are still relatively difficult tasks. There are several standard systems working in the Identification (ID) of low frequency or high frequency rates, but during the ID process, the tracked person or the object must somehow touch or be near to the reader or even insert an ID card into a reader.

Next we present some of the most well know identification systems:

4.3.1 RFID Systems

RFID Systems use tags with a microchip as a data-carrying device. These tags have a Unique IDentification (UID) code. The RFID systems are completed with a tag reader that interrogates the mobile tags to get theirs UID and an interface to a host application such as a computer to allocate such UIDs. In the referred systems, the targets that are being monitored are not required to perform any action during the ID process, that is, to be computationally headless [43][44][45] and [45].

From [46] a tracking system for a constant flow of targets in which the speed of the targets is determined using previous sensor measurements is presented. The new obtained data is continuously updated by using the current sensor measurements dependent of the previous ones. In [47][47] another tracking system is presented, the authors modify the layer two MAC protocol of the RFID standard to get as faster as possible the tags ID.

4.3.2 WLAN

The systems based in the wireless LAN technology use the MAC address to perform the connection to the network. Hence, it is straightforward to get the device ID since MAC addresses are unique and therefore the nodes are perfectly indentified.

4.3.3 Bluetooth

Bluetooth is an open wireless protocol for exchanging data over short distances from fixed and mobile devices, creating personal area networks called piconets or scatternets which are limited to seven devices. As in WLAN, the network identification use unique MAC addresses [12][12].

4.3.4 Zigbee

ZigBee is a specification for a suite of high level communication protocols using small, low power digital radios based on the IEEE 802.15.4-2003 standard for wireless PANs. Zigbee has not the Bluetooth limitation of the number of nodes belonging to a network. Moreover, Zigbee provides two ways of identifying nodes, by using the devices MAC address and a two bytes address to identify the node into the current wireless sensor network. Both methods could be used to indentify uniquely the nodes within the network [32].

4.3.5 Artificial Vision

Most of these systems use printed codes or patterns that can be recognized by Webcams. These codes can give information about the position or the number of objects or some information which permits identify the object as in QR codes

4.4 Information Exposition to Application Layer

There are several research projects with the main focus put on the management of context and knowledge information. The gathering of that information involves three main different aspects: the description, the storage and retrival, and the acquisition of information. This section is going to cover two of these issues: the description and the storage due these two aspects are the most important ones in order to communicate the information to the application layer (WP3).

Among all the projects related to these areas of study:

- DANAE project [49] aimed at providing a complete framework for not only describing multimedia information but also for transporting this information to the user with the best QoS taking into account the session context information.

- ITACITUS project [50], where a distributed cultural resources repository is to be offered to the visitor or researcher.

- AMBIESENSE project [51] the end user is a mobile citizen surrounded by an intelligent environment which senses the contextual information and offers personalized services. In this case context tags are used to implement the context-aware technologies.

- HYDRA project [52] relies on the middleware abstraction to construct an intelligent ambient supported by heterogeneous devices and technologies.

- DAIDALOS [53][53]: The DAIDALOS vision is to seamlessly integrate heterogeneous network technologies that allow network operators and service providers to offer new and profitable services, giving users access to a wide range of personalised voice, data, and multimedia services or

- MIDAS Project [54]. The main objective of the project is to define and implement a platform to simplify and speed up the task of developing and deploying mobile services, making it commercially feasible for the wider IT industry (not just telecom companies) to provide such services.

With respect to the storage and retrieval of context information, many content-aware multimedia services have been presented in the literature, such as content-based retrieval, content-based navigation and real-time video streaming. For instance:

- **A Classification Framework for Storage and Retrieval of Context** [55]. This paper presents a classification framework for the storage of the context model and the retrieval of the context information from the context model. The context model is used to add context meaning to the raw monitoring data, and is thus a necessary element for most context-aware systems. After the context model adds meaning to the raw monitoring data, the resulting context information is used by the system.

- **Adaptive storage and retrieval of large compressed images** [56]. Enabling the efficient storage, access and retrieval of large volumes of multidimensional data is one of the important emerging problems in databases. This paper presents a framework for adaptively storing, accessing, and retrieving large images. The framework uses a space and frequency graph to generate and select image view elements for storing in the database. By adapting to user access patterns, the system selects and stores those view elements that yield the lowest average cost for accessing the multi-resolution sub region image views.

- **Context-Aware GRID Services: Issues and Approaches** [57]. This paper contains a section related to context management system. The context management service uses the context gathering and publishing service, the context retrieval service and the context storage service. Context information is obtained from a wide plethora of applications, services and sensors (collectively called context sources) spread all over the network. The job of the context gathering service is to collect this information and put it into a common model understood and accessible by all components of the context service. This highly distributed characteristic introduces a great challenge for the manageability of all this information. Hence there is a need for a common context information model and a context publishing protocol. The context is then stored in a context information base for access by context consumers through the context retrieval service. The relationship between the context sources and consumers are shown in figure 1 below.

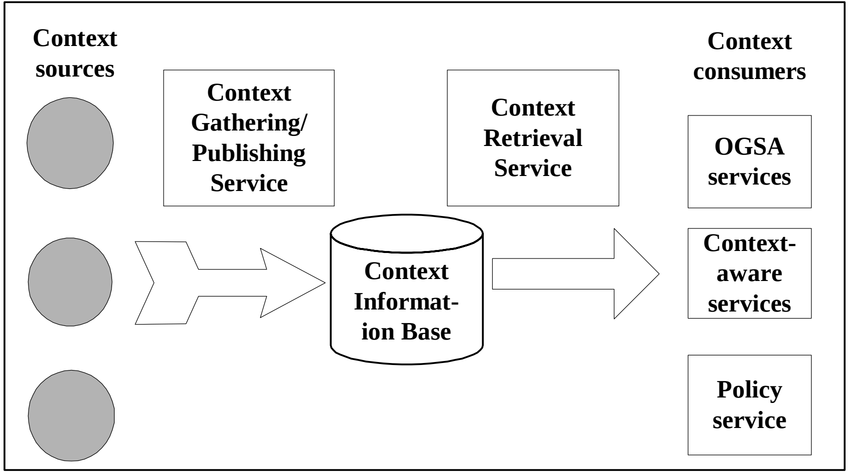


Figure 5: Relationship between context sources and context consumers

- **Towards a Conceptual Model for Context-Aware Adaptive Services** [58] Recent advances in both portable devices and wireless networks make mobile computing a reality. Embedded and invisible computing resources are paving the way to a new paradigm known as pervasive computing. More attention needs to be paid to the development of intelligent services in such a highly dynamic environment. This paper aims to present a conceptual model for context awareness based service adaptation methodology.

4.5 References

[1] S. PIVA, M. GANDETTO, R. SINGH, and C. S. REGAZZONI, "HETEROGENEOUS SENSORS DATA FUSION ISSUES FOR HARBOUR SECURITY," in *Harbour Protection Through Data Fusion Technologies*. Springer Netherlands, 2008, pp. 233-241.

[2] H. D. and J. Llinas, "An introduction to multisensor data fusion," : Proceedings of the IEEE, vol. 85, no. 1, pp. 6-23, 1997.

[3] R. Grabowski, P. Khosla, and H. Choset, "Development and Deployment of a Line of Sight Virtual Sensor for Heterogeneous Teams," in *International Conference on Robotics and Automation (ICRA 2004)*, 2004, p. 3024–3029.

[4] D. Culler, D. Estrin, and M. Srivastava, "Overview of Sensor Network," *IEEE Computer Society*, Aug. 2004.

[5] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery, "Information Fusion for Wireless Sensor Networks: Methos, Models, and Classifications.," *ACM Computing Surveys*, vol. 39, no. 3, Aug. 2007.

[6] K. Sanem, A. Pridgen, and J. C., "Virtual Sensors: Abstracting Data from Physical Sensors," in *International Symposium on a World of Wireless, Mobile and Multimedia*, Washington, DC, USA, 2006, pp. 587-592.

[7] S. Kabadayi, A. Pridgen, and C. Julien, "Virtual Sensors: A Demonstration," University of Texas at Austin TR-UTEDGE-2007-003, 2007.

[8] K. Sanem, A. Pridgen, and J. C., "Remotely Deployed Virtual Sensors," University of Texas at Austin TR-UTEDGE-2007-010, 2007.

[9] P. Ciciriello, L. Mottola, and G. P. Picco, "Building virtual sensors and actuators over logical neighborhoods," in *international workshop on Middleware for sensor networks*, Melbourne, Australia , 2006, pp. 19-24.

[10] P. Evensen and H. Meling, "Sensor virtualization with self-configuration and Flexible Interactions," in *Casemans '09: Proceedings of the 3rd ACM International Workshop on Context-Awareness for Self-Managing Systems*, New York, NY, USA, 2009, pp. 31-38.

[11] HOLGER KARL and ANDREAS WILLIG. “Protocols and Architectures for Wireless Sensor” Networks. John Wiley & Sons, 2005.

[12] KOTANEN, A., et al.“Experiments on Local Positioning with Bluetooth” ICT 2003, 10th International Conference on Telecommunications. Feb 2003. Volume 2. Pp 297-303

[13] PRIYANTHA, N. B. ; CHAKRABORTY, A.; BALAKRISHNAN. H. “The Cricket location-support system”. In Mobile Computing and Networking, pp. 32–43, 2000

[14] SALLAI, J; BALOUGH, G; MAROTI, M.; LEDEZI, A.; and KUSY, B. “Acoustic ranging in resources constrained sensor networks”. International Conference on Wireless Networks. 2004

[15] NICULESCU, D and NATH, B. “Ad hoc positioning system (APS) using AoA”. IEEE Infocom. April, 2003

[16] PENG, R. and Sichutiu, M. L. “Angle of arrival localization for wireless sensor networks”. Sensor and Ad Hoc Communications and Networks. SECON’06. 2006

[17] PORRETTA, M.; NEPA, P.; MANARA, G. and GIANNETTI, F. “Use of Deterministic Propagation Models for Testing Wireless Networks Location Techniques” IEEE Vehicular technology magazine. June 2008

[18] RANA, M. and JAGANNATHAN, S. “R-Factor: A new parameter to enhance location accuracy in RSII based real-time location systems”. 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09.

[19] HEXAMITE, “Hexamite Homepage” http://www.hexamite.com. 2007

[20] MCCARTHY, M.; DUFF, P.; MULLER, H. L.; RANDELL, C. “Accessible ultrasonic positioning” IEEE Pervasive Computing. Volume 5, issue 4. Octubre-Diciembre 2006. Pp. 86-93.

[21] GONZALEZ, J. R. and BLEAKLE, C. “High-precision robust broadband ultrasonic location and orientation estimation”. IEEE Journal of selected topics in signal processing. Vol 3. #5. October. 2009

[22] LEE, C et al. “Indoor positioning system based on incident angles of infrared emitters” 30th annual conference of IEEE on Industrial Electronics Society. IECON 2004. Noviembre 2004. Volume 3, pp. 2218-2222

[23] SAYEEF, S.; MADAWALA, U. K.; HANDLEY, P. G.; SANTOSO, D. “Indoor personnel tracking using infrared beam scanning” IEEE on position location and navigation symposium. Abril 2004. PLANS 2004

[24] GUSTAFSON, D. E.; ELWELL, J. M.; SOLTZ, J. A. “Innovative indoor geolocation usin RF multipath diversity” IEEE on position, location and navigation symposium. Abril 2006. PLANS 2006

[25] SERTTHIN, C.; TSUJI, E.; KUWANO, S and WATANABE, K. “A switching estimated receiver position scheme for visible light indoor positioning systems.” . 4th International Symposium on Wireless Pervasive Computing, 2009. ISWPC 2009

[26] JEITA (Japan Electronics and Information Technoloy Industries Association). Standar number CP-1222

[27] LORINCA, K., WELSH, M.: MOTETRACK. “A robust, decentralized approach to rf-based location tracking”. Proceedings of the International Workshop on Location and Context-Awareness LoCA’05. 2005

[28] GIGL, T.; JANSEN, G. J. M.; DIZDAREVIC, V.; WITRISAL, K.; IRAHHAUTEN, Z. “Analysis of a UWB indoor positioning system based on received signal strength” 4th workshop on positioning, navigation and communication 2007. WPNC’07. March 2007. Pp: 97-101

[29] CIURANA, M.; BARCELÓ, F.; CUGNO, S. “Indoor tracking in WLAN location with TOA measurements” Proceedings of the international workshop on Mobility management and wireless access MobiWac’06

[30] KUSHKI, A.; PLATANOTES, K. N.; VENETSANOPOULOS, A. N. “Kernel-bases positioning in wireless local area netowrks” IEEE Transactions on Mobile Computing. Volume 6, issue 6. Junio 2007. Pp: 689-705

[31] ZHOU, S.; POLLARD, J. K. “Position measurement using Bluetooth” IEEE Transaction on Consumer Electronics. Volume 52, issue 2. May 2006. Pp. 555-558

[32] NORRIS, M. “Single-Chip Zigbee for Indoor Mobile Telemetry”. The IEEE Seminar on Telemetry and Telematics. 2005. Ref. No. 2005/11009. 11 april 2005. pp 10/1-10/4

[33] HINSKE, S. “Determining the position and orientation of multi-tagged objects using RFID technology” Fifth annual IEEE international conference on pervasive computing and communications workshop 2007. PerCom Workshop’07. March 2007. Pp: 377-381

[34] WhereNet. “Wherenet homepsge”. http://www.wherenet.com. April ‘07

[35] KRAMP, G. “Hyoerfloor” Proceedings of the third Nordic conference on Human-computer interaction NordiCHI’04. October 2004

[36] DJUKNIC, G. M. RICHTON, R.E.“Geolocation and Assisted GPS”, IEEE Computer, 34, 2, pp. 123-125.. 2001

[37] SCHMID, A. “Positioning accuracy improvements with differential correlation” IEEE Journal of Selected Topics in Signal Processing. Vol 3. #4. August 2009

[38] BURLINGTON, VT.”Technical Description of DC Magnetic Trackers”, Ascension Technology Corp., 2001

[39] HIGHTOWER, J.; BORRIELLO, G. “Location sensing techniques” 2001 Universidad de Washington. UW CSE Technical Report, 2001

[40] HEIDARI, M.; NAYEF, A. A. and PAHLAVAN, K. “UDP identification and error mitigation in TOA-based indoor localization systems using neural network architecture” IEEE Transactions on Wireless Communications, vol 8, # 7. July 2009

[41] WEI-WEN, K. “Integration of GPS and dead-reckoning navigation systems” Vehicle Navigation and Information Systems Conference, Volume: 2, On page(s): 635- 643. 1991. October. 1991

[42] VON DER HARDT, H.-J. WOLF, D. HUSSON, R. “The dead reckoning localization system of the wheeled mobile robot ROMANE” International Conference on Multisensor Fusion and Integration for Intelligent Systems, 1996. IEEE/SICE/RSJ. Pp. 603-610. December 1996

[43] POLIVKA, M.; SVANDA, M.; HUDAC, P. and ZVANOVEC, S. “UHF RF identification of people in indoor and open areas” IEEE Transactions on microwave theory and techniques. Vol 57. #5. May 2009

[44] THAMAE, L. Z.; WU, Z. and KONRAD. W. “Propagation characteristics of a 2,45 GHz microwave radio frequency identification system”. IET Microwaves, Anttennas & Propagation. Volume 3. Issue 1. Pages 32-39. February 2009

[45] POLIVKA, M.; SVANDA, M. and ZVANOVEC, S. “UHF RF Identification distance in indoor areas” 3rd Conference on Antennas and Propagation. EuCAP 2009. Pages 2318-2320. March 2009

[46] GEYIC, S. C. and SZYMANSKI, K. “Multi-target tracking and identifiction by a vector of sensor” IEEE Military Communication Conference 2008. MILCOM 2008. November 2008

[47] PANICHPANIBOON, S. “Adaptative frame length selection scheme for RFID object identification” IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications 2007. PIMPC 2007. September 2007

[48] Josh Rouillard, "Contextual QR Codes," The Third International Multi-Conference on Computing in the Global Information Technology, iccgi, pp.50-55, 2008

[49] DANAE project “Dynamic and Distributed Adaptation of scalable multimedia content in a context-Aware Environment”: http://danae.rd.francetelecom.com/

[50] ITACITUS project “Intelligent Tourism And Cultural Information Through Ubiquitous Services”:http://itacitus.org

[51] AMBIESENSE project “Ambient, personalised, and context-sensitive information systems for mobile users”: http://www.ambiesense.net/

[52] HYDRA project “Networked Embedded System middleware for Heterogeneous physical devices in a distributed architecture”: http://www.hydramiddleware.eu/hydra_documents/Hydra_brochure.pdf

[53] DAIDALOS: “Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services”, FP6-IST project: http://www.ist-daidalos.org/

[54] MIDAS, FP6-IST project: http://www.ist-midas.org/documents/project_summary_midasv11.pdf

[55] B.I.J. Siljee, I.E. Bosloper, J.A.G. Nijhuis “A Classification Framework for Storage and Retrieval of Context”

[56] J. R. Smith, V. Castelli, C.S. Li: "Adaptive storage and retrieval of large compressed images," in Storage & Retrieval for Image and Video Databases, VII, M.M Yeung, B.L. Yeo and C. A. Bouman Eds. Proc. SPIE, vol. 3656, pp. 467-487, January 1999.

[57] Kerry Jean, Alvin Tan, Alex Galis “Context-Aware GRID Services: Issues and Approaches”

[58] Jun-Zhao Sun and Jaakko Sauvola, “Towards a Conceptual Model for Context-Aware Adaptive Services”

5 Wireless Sensor and Actuator Networks for DiYSE

This section zooms into a specific technology for the interaction with the environment, Wireless Sensor and Actuator Networks. Given the broad spectrum of research subjects in that area, we will identify the potential and the major challenges raised by its use in DiYSE applications, and analyze the associated requirements and state-of-the-art.

5.1 Middleware for wireless sensor networks (UPM)

5.1.1 Distributed Systems and Middleware

Coulouris et al [1] define a distributed system "as one in which hardware or software components located at networked computers communicate and coordinate their actions only by passing messages". Wang in [1] claims that one of the main challenges of distributing computing comes from the conflict between the contexts of distributed computing and the embedded sensor devices. Distributed computing should support scalability, reliability, dependability, and heterogeneity, but this demands the careful the design under the context of resource limited devices and dynamic network topology.

In order to provide the above mentioned high level services, while providing support for the existing heterogeneity in Wireless Sensor Networks architectures, a middleware layer is required. Middleware for sensor networks can be considered as a software infrastructure that glues together the network hardware, operating systems, network stacks, and applications. A complete middleware solution should contain a runtime environment that supports and coordinates multiple applications, and standardized system services, such as data aggregation, control and management policies adapting to target applications. Also, middleware software architectures should offer mechanisms to achieve adaptive and efficient system resources use, in order to prolong the sensor network's life.

5.1.2 Middleware approaches for Sensor Networks

Different middleware approaches were found, García in [3] has classified these approaches, taking into account the programming models in sensor networks, see Fig. 1.

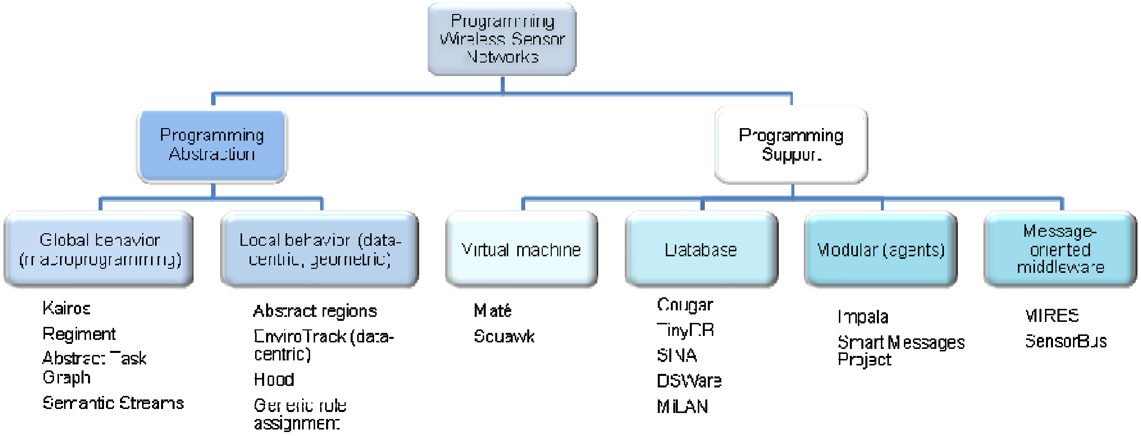


Fig. 1 Middleware approaches taking the programming model used into account [Adapted from [1].

Programming sensor networks includes two major classes: programming support and programming abstraction.

Programming Abstraction

Manage to the way a sensor network is viewed and presents concepts and ideas of sensor nodes and sensor data. There are two main approaches for programming abstraction classes the global behavior and the local behavior approaches.

Global Behavior

This first programming abstraction approach, the sensor network is programmed as whole rather than writing low-level software to drive individual nodes. A global WSN's behavior is programmed at a high-level specification that enables node concerned about dealing with low-level. Some examples of this approach are: Kairos [4], Regiment [5], Abstract Task Graph [6] and Semantic Streams [7].

Local Behavior

This second programming abstraction approach deals with the behavior of the sensor network nodes from a local point of view in a distributed computation. The local behavior approach focuses on the nature of the sensed data and, in particular, on a specific location in a sensor network. Some examples of this approach are: Abstract Regions [8], EnviroTrack (data-centric) [9], Hood [10] and Generic role Assignment [11].

Programming Support

Manage the providing systems, services, and run-time mechanisms, such a reliable code distribution, safe code execution, and application-specific services.

The programming support class consists of five approaches (see Fig.1): virtual machine-based, modular programming-based, database-based, application-driven, and message-oriented middleware.

Virtual Machine

This approach consists of virtual machines, interpreters and mobile agents. Its main characteristic is flexibility, allowing developers to write applications in divided small modules, which are injected and distributed through the network by the system using tailored algorithms and then interpreted by the virtual machine. Some examples of this approach are: Maté [12], Squawk [13].

Modular Programming (Mobile Agents)

The use of mobile code facilitates the injection and distribution through the network and leads to application modularity. Less energy is necessary when broadcasting small modules instead of the complete application. Some examples of this approach are: Impala [14], Smart Messages Project [15].

Database

This approach observes the entire network as a virtual database system, offering an easy-to-use interface that permits the user extract data of interest and issue queries about the sensor network. The database is one of the earliest examples of high-level abstractions for sensor network programming. Some examples of this approach, TinyDB [16], SINA [17], DSWare [18] and MiLAN [19], which in addition to these features, provides a data service that features QoS support.

Message-Oriented Middleware (MOM)

This approach is quite suitable in pervasive environments such as Wireless Sensor Networks, where most applications are based on events. Message-oriented middleware uses the publish-subscribe mechanism to facilitate message exchange between nodes and the sink nodes. Some examples of this approach are: MIRES [20] and SensorBus [21].

5.1.3 Context-Aware Middleware Approaches

Many applications, which are deployed over Wireless Sensor Networks, are context-aware. Therefore, it is necessary to found mechanisms in order to get context information from the environment in a structured way. Besides, this information has to be meaningful from the application point of view. In this sense, the middleware layer in Wireless Sensor Networks has to implement some mechanisms in order to reach efficient deployments of context applications over Wireless Sensor Networks. Since each application interprets the underlying sensor network differently according to their objectives, middleware layer has to manage different contextual requirements. In the following paragraphs the main contextual middleware approaches as well as two proposals for context information presentation in Wireless Sensor Networks, will be briefly described.

The middleware layer proposed in [22] is based on an execution Framework. That Framework is able to manage contextual information by using an architecture divided in three sub-layers: Context Provider, Context Process and Context Adapter. The middleware's life cycle is divided in three phases: acquisition of context data, interpretation of context information and adaptation according to identified situation. The Context Provider layer provides "crude" data about the environment and sensor status. The Context Process layer filters and aggregate the crude data from Context Provider. The higher layer, Context Adapter, is able to take decisions about the convenience of performing an adaptation. In this proposal there are context nodes which provide context information by using five primitive components: Context Process, Context Reasoning, Context Configuration, Activity Manager, and Message Manager.

In [23] a middleware for contextual agents was proposed. This middleware layer was thought with the purpose of compose an execution Framework suitable for agents in ubiquitous computing environment. The contextual model implemented by this proposal allows using different reasoning mechanisms like first order logic and temporal logic. The types of agents and services coexisting in this middleware are the following ones: Context Providers, Context Synthesizer, Context Consumer, Context Provider Searching Service, Historic Context Service and Ontology Service.

The middleware proposed in [24] attempt to solve some problems identified in WSNs, majorly three:

- The solutions in WSN are usually designed and implemented for a specific objective and a single platform.
- The lack of a standard allowing the communication between different WSN technologies.
- The most of WSN solutions are based on arrays of homogeneous sensors.

To solve the three major problems in WSNs mentioned above, a Semantic Sensor Network (SSN) was proposed in [24]. This approach allows semantically tagging the sensed data from a heterogeneous distributed sensor network in order to ease the managing of contextual data in a large scale network.

In [25] a Context Aware Sensornet (CASN) was proposed. CASN integrates the contextual computing theory [26] with sensor networks concept. In CASN, the node's context is most important than the human context. This approach implies several challenges as a suitable behavior abstraction or the technologies required for context representation in an energy-efficient way. The middleware's framework is composed of four main components: Context Representation Component, Context Interpreter Component, Contextual Service Component, and the Kernel of the node. The Context Representation Component uses a lightweight ontology called µSONG (Micro Sensornet Ontology) which provides a simple and flexible way of presenting the context. The context interpretation is achieved by using an interpreter based on fuzzy rules called CIBFR (Context Interpreter Based on Fuzzy Rule).

In [27] a rule based Middleware called MIDSEN was proposed. This proposal includes two major algorithms: event detection algorithm (EDA) and context aware service discovery algorithm (CASDA). Both algorithms are implemented by inference engine. MIDSEN defines sensors and applications as services. EDA takes an input as sensor readings and makes an event primitive. A primitive event is built by event detection time and event format. By matchmaking, CASDA discovers services, which match with given service request.

The Framework-based middleware proposals mentioned above integrates mechanisms to manage contextual information. However, each proposal used its own language to represent that information. Currently, there is not a standard to formalize the representation of information which is managed in resource constrained systems as Wireless Sensor Networks. In this sense, several representation models have been proposed to be used in Wireless Sensor Networks. Between them, we can found WISNO (WIreless Sensor Networks Ontology) [28]. WISNO includes an ontology divided in two levels: high and low. The high level ontology is used to perform a fine analysis of contextual information. The low level ontology is used to characterize the data from sensors which are deployed around the Wireless Sensor Networks. Reasoning rules based on descriptive logic and SWRL [29] have also included in WISNO specification. Another proposal, which is based on formalized representation system of sensor information, is [30]. In this proposal each sensor provides an energy level as well as its status. The condition of every sensor integrated into the node is described by the following quadruple: <t, m, e, a>, where t is the sensor type, m is the operator type, e is the energy consumption of that sensor, and a is the sensor precision. The dynamic information of each sensor can be summarized in tuples like <E, {S}>, where E is the remaining energy level and {S} is the set of one or more quadruples which describe the status of the sensors in the node.

5.2 In-network reasoning and data fusion

Sensors and actuators use to have a very small process capability. In any case, sensor processors are improving this aspect, and there is a trend to include inside some control algorithms. These devices are starting to deploy real distributed systems.

In this sense, the concept of artificial intelligence is starting to be included into sensors and actuators [31]. Different paradigms have been used to perform this intelligence, like knowledge based systems [32], fuzzy logic [33][34] or artificial neural networks [35].

Deckneuvel [31] reported an analysis of intelligent sensor and purposed a language specifically developed for the design of these systems. Benoit et al. [36] presented a modeling of intelligent sensor and proposed three large categories when intelligence is applied to sensor: intelligence of the perception, reasoning, and social intelligence. Lately, hybrid systems, which are composed of fuzzy logic and neural network, have been proposed Averkin and Belenki [37]. In [38], the use of a distributed rule based fuzzy logic engine designed for collaborative WSN has been described. This approach uses fuzzy logic:

1. To fuse individual and neighbourhood observations in order to produce a more accurate and reliable result;

2. As cooperative algorithm to compensate the resource limitations and the lack of reliability.

5.3 Services Management in Wireless Sensor Networks (UPM)

Wireless Sensor Network consists of a multitude of tiny embedded devices that are capable of sensing information continually and transmit data from one device to another via a wireless ad hoc network. Such networks are characterized by their ease of deployment and being self-configuring. Nowadays, the applications of WSN technology have been broken down into two main categories: Monitoring and Tracking.

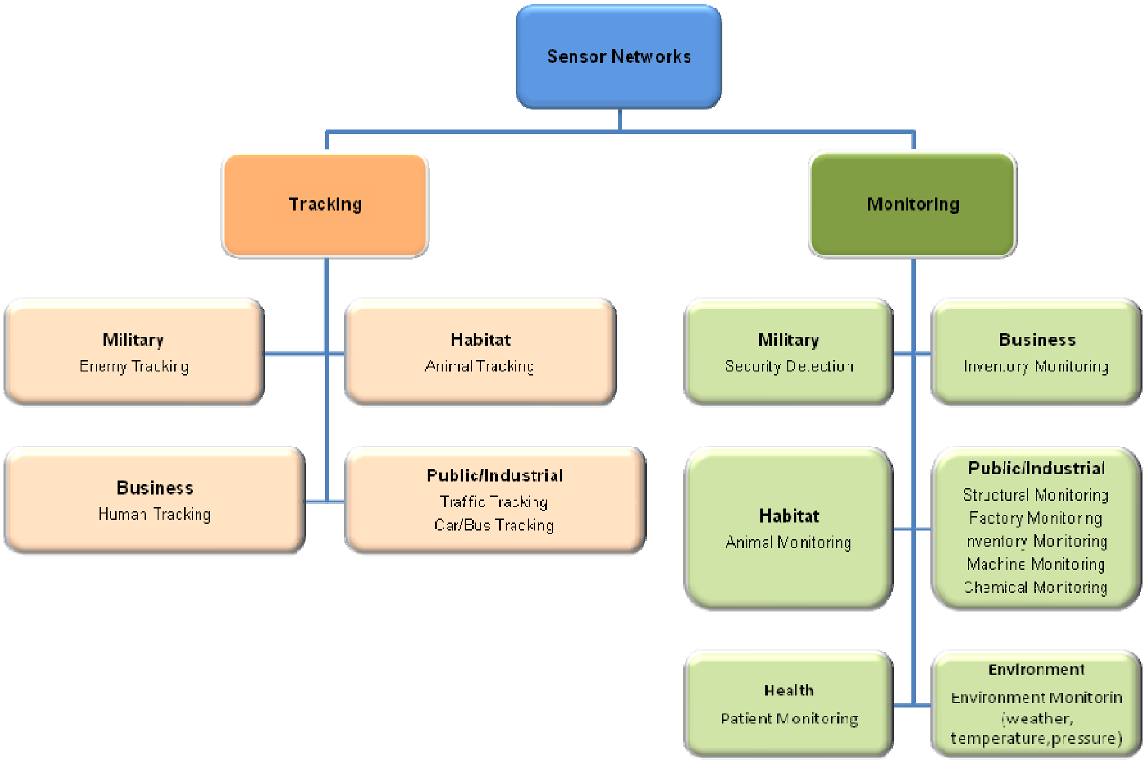


Fig. 2 Overview of Sensor Network applications [39].

One of the most important features of the WSN for DiYSE is that they can be completely heterogeneous characteristics, for example, nodes may have multiple types of sensors, different power and processing capabilities and can interact with other network through a gateway.

Powerful devices can perform complex operations, but are more expensive and power-hungry consume much power. Otherwise, weak WSN devices enable higher deployment densities and increase network lifetime as they are cheaper and consume less power. By integrating devices with different resources and capabilities, a heterogeneous WSN can combine the advantages of both powerful and weak devices.

The heterogeneity of the network presents significant challenges for service provisioning. New programming models are necessary to simplify WSN application development and increase overall network utility.

Service-oriented computing can simplify application development by hiding platform-specific capabilities behind services. These services are dynamically discovered and used at run-time, enabling applications to be platform-independent and adapt to network dynamics. While service-oriented computing is widely used on the Internet, adopting it to WSNs is non-trivial due to the extremely limited resources available and highly dynamic nature.

5.3.1 Service Provisioning in Sensor Networks

Nowadays, Wireless Sensor Networks are systems that have a limited amount of resources. Therefore, service provisioning in Sensor Networks is a huge challenge.

The classical SOA-based approaches are not currently feasible to be used over Wireless Sensor Networks because of the intrinsic resource limitations of that kind of technologies. The concept *Services Oriented Architecture* (SOA) refers to a set of software components that together perform a certain task or provide a service [40], [41].

The SOA standards such as XML, HTTP, SOAP, WSDL and UDDI are majorly related with web services provisioning by using no resources-constrained computers so they are not recommended to be applied in WSN. There are some proposals which try to solve the service provisioning in Wireless Sensor Networks by using SOA-based technology.

In [42] an iterative SOA-based design process was proposed. Services-oriented architecture suits particularly well for WSNs as the development of the whole network can directly be mapped to service, simple or complex. The proposed design process is based on agile design technologies [43]. The authors of [42] chose this methodology since the WSN development is iterative and short what suits with the agile methods. However, it is structured according a waterfall model [44]. The waterfall model includes eight stages: *gathering of the requirements, their analysis, the design of the solution, development of the software architecture, development of the code, testing, deployment and post implementation.*

As it has been commented in previous section in [45] sensors and applications are modeled as services. This proposal includes a service discovery algorithm called context CASDA. This algorithm takes input as service request (*SR*) and available services (*S*). For filtering purpose, only those services that belong to service requester category are managed. This algorithm returns degree of similarity between service request and available services. To perform the comparison between requested and available service some factors are taken into account such service's inputs and outputs, and required contextual information.

Fok in [46] proposed a middleware when the applications are implemented as task, which are platform-independent application processes that contain code, state, and service specifications. Services are able to maintain state, provide multiple methods, and have their own thread of control, enabling them to operate in parallel with task. Servilla provides two light-weight programming languages tailored to support service provisioning in WSNs. The first, ServillaSpec, is used to create service specifications and descriptions that enable flexible matching between tasks and services. The second, ServillaScript, is used to create tasks and is compiled into bytecode that runs on a Virtual Machine. Services are implemented in NesC on TinyOS and compiled into native binary code for run-time efficiency. An important feature of Servilla lies in its capability to support coordination and collaboration among heterogeneous devices inside a WSN.

5.3.2 Execution Environments

Wireless sensors networks, as the field of matures, needs support more complex applications and collaboration among them in order to provide services. For this propose, it's necessary more powerful programming methods, monitoring and control, both hardware and software, during its operation. So, specifying the program after deployment and changing it during operation it´s necessary, since the application may be somewhat changed during the sensor operation to ensure adequate service provision. Some solutions have been propose to allow reprogramming sensor networks in the field.

The ability of loading and updating applications after deployment is one of the factors that it will be the local sensor networks usable. Lately there have been some interesting proposals in this regards.

SensorScheme [47] is a platform for dynamic program loading and execution, based on the semantic of the Scheme programming language and designed to meet the demands of sensor networks applications. This platform focused on efficient code transport, minimizing its size, by separating the format while transmitted from the in-memory code storage while executing, optimizing the communication channel and energy consumption.

Interactive and Extensible Framework for Execution and Monitoring of Wireless Sensor Networks (ISEE) [48] is an environment for the execution and monitoring of sensor network services. Is supports verification and testing of sensor network services, whether simulated, emulated or real. So, this framework can be used during all process in wireless sensor network, development, deployment and real use. This framework is based in previous work like EmStar [49] and TOSSIM [50], a simulator for TinyOs Networks.

Also, related whit this are the Virtual Sensor Networks. They are a collaborative Wireless Sensor Networks to provide protocol support for the information, usage, adaptation and maintenance of subsets of sensors collaborating on specific tasks. The main target is to enhance applications in which subsets of sensors, varying dynamically, must to achieve the desired outcomes, while relying on the remaining nodes connectivity, deployment and resources constrains.

Now, the objective is to get an execution environment that it allows to configure dynamically a service. But, usually, a service shall consist of a group of applications that, using a collaborative way among then, they will provide a service. Therefore, it's necessary to develop an environment that configures each application and the relation with the others applications that provide the service.

5.4 Wireless Sensor Networks Management

The function of Wireless Sensor Network Management Systems is to provide monitoring and controlling capabilities functionalities. This kind of ubiquitous networks presents several peculiarities that make more difficult the management task performing over them, where can be identified open issues like the constrained-resources of nodes, dynamic network topology, variable channel capacity and prone to fail [51]. Due these limitations, main efforts in management procedures for sensor networks are mainly focused on monitoring and controlling tasks, in order to optimize the network operation and maintain the network performance [52].

The network management protocols and frameworks designed for Wireless Sensor Networks had take into account the properties of sensor nodes. In this way, suitable characteristics of network management for Wireless Sensor Networks can be identified as follows: light-weight and event driven communication paradigm, robustness and fault tolerance, adaptability and responsiveness, minimal resource usage and scalability [53], [54]. In next Subsection, foundations of main approaches for sensor networks management will be presented, classifying them into protocols and management frameworks.

5.4.1 Management protocols

RRP (Register mechanism Routing Protocol) [55] is a cluster-based mobile routing algorithm aimed to improve the network life-time, using for this a system's load balancing schema, which defines a set of flooding-zones for the data forwarding decisions, in order to perform the data aggregation tasks. RRP proposes a hierarchical deployment based on three area types: manufacturing, warehouse and service. Acquisition of data is carry out in the manufacturing area, delivering the processed data to the warehouse and service area. The main advantages of RRP are that zone flooding ensures low message overheads, and adjusting the size of flooding zone, it ensures high reliability. The main lacks of RRP are that it requires a GPS device attached to the sensor nodes, in order to implement the zone-flooding protocol.

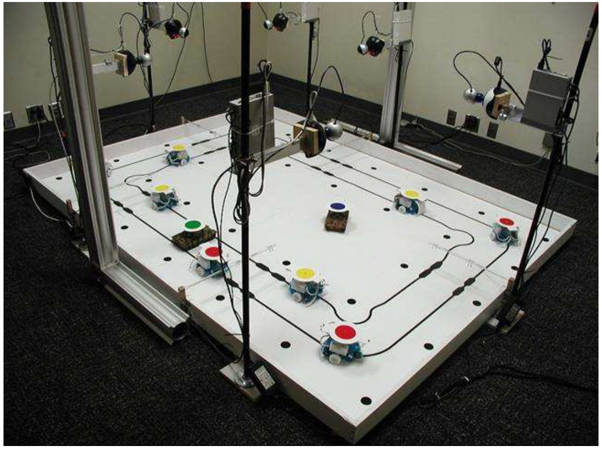
SNMS (Sensor Network Management System) [56] is an interactive system for sensor network health monitoring. It provides a query-based network health and event logging functions. SNMS supports collection and dissemination of traffic patterns. Collection traffic pattern is used to obtain health data from the network, while dissemination traffic pattern is used to distribute management messages, commands, and queries. To achieve the previous exposed goals, SNMS develops a gathering tree to collect network health information, introducing a minimal impact on memory and network traffic. SNMS further minimizes energy consumption merging multiple queries into a single message. On the other hand, SNMS network management function is limited to passive monitoring only, requiring human managers to submit queries and perform post-mortem analysis of management data.

WinMS (Wireless Network Management System) [57] proposes an adaptive policy-based sensor network management system, which provides self-management for network performance maintaining, adapting the network behavior according to the traffic conditions. WinMS architecture defines a schedule-driven MAC protocol, to collect and disseminate management data, form and to the sensor nodes in a gathering tree. Also, it implements a local network management scheme, providing autonomy for wireless sensor nodes to perform management functions, and a central network management scheme, to perform preventive and corrective maintenance. It is worth nothing, however, that the initial setup cost for building the gathering tree is proportional to network density.

5.4.2 Management frameworks

BOSS (Bridge Of the SensorS) [58] defines a service discovery management approach for Wireless Sensor Networks. It supports network state information retrieval from the Wireless Sensor and Actuator Network, including sensor node device description, the number of sensor nodes in the network, and the network topology. The localization service provides positioning information for each sensor node in the network. The synchronization service is focused for clock synchronization among sensor nodes in the network. The power management service offers support for checking remaining battery and changing the sensor's operation mode. BOSS offers dynamic adaptation for sensor network topology changes, supporting proactive network management. On the other hand, BOSS requires human interaction to analyze the network states, taking management actions accordingly.

MANNA (Management Architecture for Wireless Sensor Networks) [59]is a policy-based management architecture designed for gathering dynamic management information, mapping it into sensor networks models, performing management functions and services based on wireless sensor network. It defines the MANNA Network Management Protocol (MNMP), which is a light-weight protocol designed for management information exchange between management entities (i.e., cluster-heads, nodes and manager). Some of the management procedures covered by MANNA are related with coverage area supervision, networking parameters configuration, network topology and connectivity discovery, energy map generation, and node localization. Also, MANNA Framework performs coverage area maintenance, reducing the network overhead, packets collision, and energy consumption, turning off redundant nodes in the Wireless Sensor Network.



MARWIS (Management Architecture for Heterogeneous Wireless Sensor Networks) [60] is a management framework which defines support for common management tasks, such as network monitoring, reconfiguration, and updating program code, in sensor networks composed by heterogeneous platform mote architectures and heterogeneous sensor types. This approach propose a network deployment based on clusters, called SSNs (*sensor sub-networks*), which contains sensor nodes of same type, in order to handle large, heterogeneous Wireless Sensor and Actuator Networks. To interconnect different SSNs is proposed the use of gateways. In addition, this approach proposes the use of MS (*Management Stations*), a laptop or a remote workstation, which is connected to the Internet, and where the network topology can be visualized.

5.5 Multimedia Wireless Sensor Networks (ENSIIE)

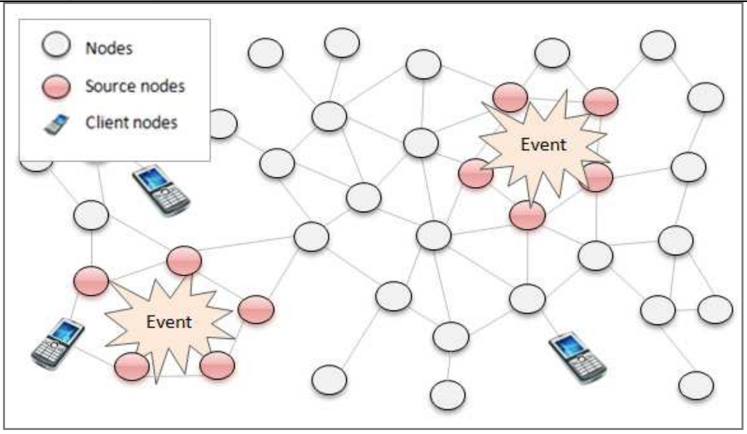
Multimedia Wireless Sensor Networks (MWSNs) can be considered as a specific application of WSNs. Nodes in this case are miniaturized multimedia acquisition devices (cameras, image sensors, audio recorders … etc.) connected via wireless communications, they produce both video and audio streams to provide an efficient coverage in a specific area to guarantee different services, such as: traffic management, home supervision, telemedicine or military surveillance. The benefit of using such architectures is the deployment and maintenance facility induced by the inherent plug-and-play and self-organized nature of wireless sensor networks.

Several academic organizations and corporations are working on the development of new devices, protocols and architectures for MWSNs. Authors of [61] give an extensive overview of the existing algorithms, software and hardware for multimedia wireless sensor networks. Studies are performed in the nodes placement strategies for efficient 3D coverage [62] and also in the image acquisition processing of multi-resolution streams [63] Crossbow Technologies for example, provides a multimedia board for their Imote2 sensors platform that allows for capturing images, video and audio. Yale University has developed SOS [64] an operating system that employs video sensor for behaviour interpretation using distributed sensing. Georgia Institute of Technology deployed an experimental MWSN testbed [65] based on heterogeneous sensors (scalar, low-res cameras, high-res cameras) for different studies at the MAC, Network, Transport and the Application layers.

5.6 Event-Centric Wireless Sensor Networks

Event-centric wireless sensor networks (EC-WSNs) differ from common WSNs, where the communications are triggered either by an on-demand or a sink-based process. Indeed, in this case, the communication is triggered only when an event is detected in the immediate sensing-range of a node. Final users, i.e. users interested in an event occurrence, can subscribe to one or more event interest, and get updates from the sensor network. For example, clients (digital billboards, cell phone, laptops, PDA) interested in parking spots availability in a specific area may subscribe to this service and get updates each time the sensor network detects that places are available. The intent is to provide a pseudo real-time view of all the occurring events of the network.

Examples of application of EC-WSNs can be envisioned in urban environment, where final users may connect to a large scale EC-WSN and subscribe to various event-interests like: parking lots or public bikes availability, queues and lines status in different offices (postal office, supermarkets, fuel stations, etc.), or even pollution levels in parks and public places.



Event Centric WSN Overview

5.7 Security in Wireless Sensor Networks

5.7.1 Vulnerabilities and Security Requirements in WSN

A wireless sensor network is vulnerable due to its characteristics as an open medium, with a network topology that dynamically changes, which employs cooperative algorithms, it lacks an element for managing and monitoring the network and it also hasn’t a perimeter defence [66][68][70] clearly defined. The main vulnerabilities in wireless sensor networks are [67][69]: wireless links, auto configuration mechanisms, limited resources, reduced communication and unattended operation.

The security problems in a wireless sensor networks become clear with the assumption that radio links are insecure, which makes communication in wireless sensor networks insecure too. An attacker could easily listen to the channel as it is a broadcasting one, inject whatever data, and even play some bit packet headers that were sent before. In addition, an attacker could gain control of one or more motes and attack from them the whole sensor network. It is also assumed that information from the mote (keys, data, etc.) will be compromised if an attacker has access to it.
Attacks on wireless sensor networks can be made from outside the network or can be originated inside it. Attacks from malicious motes that are not part of the network and try to join it without authorization, are often done applying cryptanalysis techniques and attacking physically the device.

Motes are part of the network and they may attack if they have been compromised by an attacker that has manipulated them [66]. The most common attacks that often occur in motes that are part of the network are: flood, alteration or replications of routing information, selective transmission, sink attack, Sybil attack, wormhole attack, HELLO flood attack, spoofing recognition, passive listening, denial of service and subversion of a node.

5.7.2 Security approaches

Symmetric and asymmetric cryptosystems approached for WSN

Security in wireless sensor networks is currently provided exclusively through symmetric key cryptography but there is some studies which purpose ultra-low power hardware implementations of public key algorithms.

Several public key schemes can be used to provide the security services described above. In [6] we take a closer look at Rabin's Scheme, NtruEncrypt and Elliptic Curve Cryptosystems (ECC) as the most promising candidates for low power implementations.

Although these algorithms are more powerful and secure than those of symmetric key, we must take into account the time they take to encrypt and decrypt, the number of bits to be added when we encrypt a message and the energy consumption are much higher than that of symmetric key algorithms. Wireless sensor networks are unique in this aspect due to their size, mobility and computational/power constraints [72].

RC5 (*Rivest Cipher-5*) is an encryption block algorithm [73], which may be one of the cryptographic algorithms more suitable for WSN due to their good behavior in devices with low memory capacity.

RC6 (*Rivest Cipher-6*) is an improvement of RC5. Although RC6 is an algorithm with a security level something greater than RC5, it performance is significantly lower in WSN [74].

Another important algorithm used in Wireless Sensor Networks is **TEA** [75] (*Tiny Encryption Algorithm*), TEA is a block cipher algorithm which requires little memory space.

XTEA (*eXtended* TEA) is a block cipher algorithm, which appeared to correct the weaknesses of TEA. It was designed by engineers at Cambridge Computer Laboratory in 1997 and it has not been patent yet [76]. The small size of the implementation of this algorithm has provided an option that is highly recommended on systems with very high memory restrictions (such as embedded systems or wireless sensor networks).

Some studies [77], concluding that the memory requirements imposed by XTEA in sensor networks are a quarter of those required by AES (Advanced Encryption System). Thus also studies made by our researchers [78], has been shown that the use of the AES algorithm reduces half the batteries life.

The traditional **DES** (*Data Encryption Standard*) uses many computational resources and therefore it isn't recommended [79] for use in WSN.

TinyECC and **WMECC** [80] are two implementations of public key cryptography on the TinyOS operating system. Both include cryptographic primitives based in elliptic curve cryptography optimized for wireless sensor networks, such as models *Micaz*, *Telosb* and *Imote2*. One of the greatest advantages is that the primitives are already included a specific operating system for wireless sensor networks, allowing developers use this type of primitive easily. TinyECC and WMECC perform for the first time deployments based in public key cryptography and introduce the concept of *digital signature* in devices designed for use on WSN. However, the main disadvantage of this type of cryptography, is the high run time (with temporary magnitude scales around a second) required for data encryption or for processing or verifying digital signatures.

The .NET Micro Framework Microsoft architecture [81] also included in version 2.0 a *namespace* (package or classes group) oriented to the security for very low capacity devices, as employees in wireless sensor networks. This space name, named *Microsoft.SPOT.Cryptography* [82] incorporates two cryptographic algorithms, one based on symmetric key cryptography and the other based on public key cryptography. The symmetric key algorithm is XTEA in a version optimized for WSN. The public key-based algorithm is RSA. Although cannot speak about a security model, .NET Micro Framework provides the necessary tools to create a complete security model.

Data privacy

Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data was the first document where confidentiality of communications was guaranteed.

Directive 97/66/EC of the European Parliament and of the Council of 15 December 1997 concerning the processing of personal data and the protection of privacy in the telecommunications sector translated the principles set out in Directive 95/46/EC into specific rules for the telecommunications sector.

Directive 2002/58/EC [83] of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications).

In the case of public communications networks, specific legal, regulatory and technical provisions should be made in order to protect fundamental rights and freedoms of natural person. Measures should be taken to prevent unauthorized access to communications in order to protect the confidentiality of communications, including both the contents and any data related to such communications, by means of public communications networks and public available electronic services.

In digital mobile networks, location data giving the geographic position of the terminal equipment of the mobile user are processed to enable the transmission of communications. The processing of such data for value added services should only be allowed where subscribers

On 25 June 2008, the European Parliament's Standing Committee on Civil Liberties, Justice and Home Affairs asked for measures to correct the European Commission's proposal to amend the Directive on Privacy and Electronic Communications (called *ePrivacy Directive*).

Peter Hustinx, the European Data Protection Supervisor (EDPS), adopted, on 14 April, an Opinion on the European Commission's proposal amending, among others, the ePrivacy Directive. The EDPS basically supported the EC proposal giving a few recommendations such as the obligation to notify any breach of security not only from providers of public electronic communication services in public networks but also from providers of information society services which process sensitive personal data.

5.7.3 Security Services Infrastructure

When the number of nodes in the WSN grows and new services more complex are implemented, it is necessary to include authentication and authorization services in order to verify the data source and to allow taking authorization decisions.
In a model of communication between two parties, the data authentication can use symmetric mechanisms. Sender and receiver share a secret key to generate and verify a MAC (*Message Authentication Code*) [72].

MAC mechanisms can be integrated into a sensor network through the protocol family **SPINS** (*Security Protocols in Networked Systems*) [84]. This solution presents as advantage that only adds 6 bytes in the packet *payload* and the energy consumption is only the 20 % of the total energy use.

A widely authentication method used in WSN is the Authentication *broadcast.* This authentication requires an asymmetric mechanism. Unfortunately, the asymmetric cryptography mechanisms require a great computing process, a great communication rate and storage, as already mentioned in previous paragraphs. The **uTesla** protocol solves this problem by introducing asymmetry through the delayed revelation of symmetric keys, allowing an efficient authentication broadcast scheme. However [85] explains that uTelsa is not suitable for the authentication traffic between nodes because it provides authentication delayed in time. In this protocol generated keys are applyed to generate messages MAC sequentially, but are released with a delay once the packets are received. uTesla requires to distribute some information based on *unicast* between the base station and each sensor before authentication of *broadcast* messages. Therefore, authors as [86], propose the replacement of the initial distribution *unicast* using *broadcast*-based techniques.

We can found some researches about the way to provide entities authentication in hierarchical sensor networks. The authors of [87] propose the use of a certificate called **TESLA**, which can be use by low capacity devices (such as WSN nodes) to provide entity authentication. Its *framework* authenticates new nodes on the network, while supports trust relationships.

The authors of [88] propose a scheme based on *Elliptic Curve Cryptography (ECC)*. The idea of this scheme is to use PKI (Public *Key Infrastructure*). In this solution, there is a base station placed in a safe environment with increased capabilities of processing and storage. This base station serves as CA (*Certification Authority*). A certificate of a legitimate user (U) is signed by the CA. This schema requires more processing on the encryption and signature verification as on the decryption and signature. The authors claim that the use of ECC is feasible in this type of networks. However it may cause a bottleneck process from the sensor nodes if there is excessive traffic on the network. A weakness of this protocol is that an attacker can gain false certificates and signatures. In addition *DoS* attacks could be received by continuous sending of certificates that exhaust the memory and the battery of these nodes.

There is also the possibility of a source authentication system *broadcast* based on sending multiple messages MAC [89] called **MultiMAC**. What is new in this mechanism is to provide a key distribution combinatorial and deterministic, providing scalable authentication with few key storage requirements. This service authentication is implemented as a security component of **TinyOS**. This mechanism based on multiple MAC messages and requires that network nodes have a **key ring.** To authenticate a message, the source node generate a list of MAC based on their keys, and added them to the message. The receiving node will verify the message based on the MAC that has generated using keys that are shared with the source node. To meet the WSN restrictions must be designing an appropriate key ring.

5.8 References

[1] G. F. Coulouris, *Distributed Systems: Concepts and Design*. Addison Wesley*,* 2005.

[2] M. M. Wang, J. N. Cao, J. Li, and S. K. Dasi, "Middleware for Wireless Sensor Networks: A Survey," Journal of Computer Science and Technology, vol. 23, pp. 305-326, May 2008.

[3] A.B. García, J.F. Martínez, J.M. López, A. Prayati, and L. Redondo, “Problem Solving for Wireless Sensor Networks”. Springer-Verlag, 2008.

[4] R. Gummadi, O. Gnawali, and R. Govindan, "Macro-programming Wireless Sensor Networks Using Kairos," in Distributed computing in Sensor Systems, S. Berlin/Heidelberg, vol. 3560, pp. 126-140. 2005.

[5] R. Newton and M. Welsh, "Region Streams: Functional Macroprogramming for Sensor Networks," in First International Workshop on Data Management for Sensor Networks, 2004.

[6] A. Bakshi, V. K. Prasanna, J. Reich, and D. Larner, "The Abstract Task Graph: A Methodology for Architecture-Independent Programming of Networked Sensor Systems," in Workshop on End-to-End, Sense and Respond Systems, Applications and Services, pp. 19-24. 2005.

[7] K. Whitehouse, F. Zhao, and J. Liu. (2005, Apr.) Microsoft Research. [Online]. http://research.microsoft.com/apps/pubs/default.aspx?id=70161.

[8] M. Welsh and G. Mainland, "Programming Sensor Networks Using Abstract Regions," in First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI '04), 2004.

[9] T. Abdelzaher, et al., "Enviro Track: Towards an Environmental Computing Paradigm for Distributed Sensor Networks," in 24th IEEE International Conference on Distributed Computing Systems (ICDCS'04), 2004.

[10] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks," in Fifth Symposium on Operating Systems Design and implementation (OSDI'02), Boston, 2002, pp. 136-146.

[11] K. Römer, C. Frank, P. J. Marrón, and C. Becker, "Generic role assignment for Wireless Sensor Networks," in 11th Workshop on ACM SIGOPS European workshop, Leuven, 2004.

[12] P. Levis and D. Culler, "Maté: a tiny virtual machine for sensor networks," in 10th International conference on Architectural support for programming languages and operating systems, San Jose, California, 2002, pp. 85-95.

[13] D. Simon, C. Cifuentes, D. Cleal, J. Daniels, and D. White, "Java™ on the bare metal of wireless sensor devices: the squawk Java virtual machine," in 2nd International conference on virtual execution environments, Ottawa, 2006, pp. 78-88.

[14] T. Liu and M. Martonosi, "Impala: a middleware system for managing autonomic, parallel sensor systems," in ninyh ACM SIGPLAN symposium on Principles and practice of parallel programming, San Diego, California, 2003, pp. 107-118.

[15] P. Kang, et al., "Smart Messages: A Distributed Computing Platform for Networks of Embedded Systems," The Computer Journal, vol. 47, no. Special Issue on Mobile and Pervasive Computing, pp. 475-494, 2004.

[16] S. Madden, J. Hellerstein, and W. Hong. (2003, Sep.) TinyDB: In Network Query Processing in TinyOS. http://telegraph.cs.berkeley.edu/tinydb/tinydb.pdf.

[17] C.C. Shen, C. Srisathapornphat, and C. Jaikaeo, "Sensor information networking architecture and applications," Personal Communications, IEEE, vol. 8, no. 4, pp. 52-59, Aug. 2001.

[18] S. Li, Y. Lin, S. H. Son, J. A. Stankovic, and Y. Wei, "Event Detection Services Using Data Service Middleware in Distributed Sensor Networks," Telecommunication Systems, vol. 26, pp. 351-368, Dec. 2004.

[19] A. L. Murphy and W. B. Heinzelman, "MiLAN: Middleware Linking Applications and Networks," University of Rochester, NY, Technical Report, 2002.

[20] E. Souto, et al., "A message-oriented middleware for sensor networks," in 2nd Workshop on Middleware for pervasive and ad-hoc computing, Toronto, 2004, pp. 127-134.

[21] A. R. Ribeiro, F. C. Silva, L. C. Freitas, J. C. Costa, and C. R. Francês, "SensorBus: a Middleware model for Wireless Sensor Networks," in 3rd International IFIP/ACM Latin American conference on Networking, Cali, 2005, pp. 1-9.

[22] A. Taherkordi and R. Rouvoy, "A Self-Adaptive Context Processing Framework for Wireless Sensor Networks," in *Middleware Conference. Proceedings of the 3rd international workshop on Middleware for sensor networks*, Leuven, Belgium, 2008.

[23] A. Ranganathan and R. Campbell, "A Middleware for Context-Aware Agents in Ubiquitous Computing Environments in Ubiquitous Computing Environments," in *CM/IFIP/USENIX International Middleware Conference*, Brazil, 2003.

[24] L. Ni, Y. Zhu, and M. Jian, *Semantic Sensor Net: An Extensible Framework*. Heildelberg: Springer, 2005.

[25] H. Quin and X. Zhou, "Integrating Context Aware with Sensornet," in *Proceedings of the First International Conference on Semantics,* 2005.

[26] A. Dey and G. Abowd, "Towards a Better Understanding of Context and Context-Awareness," in *Workshop on The What, Who, Where, When, and How of Context-Awareness, as part of 2000 Conference on Human Factors in Computing Systems (CHI 2000)*, Hague, Holland, 2000.

[27] P. Patel, J. Sunil, S. Chaudhary, P. Ranjan, and D. Ambani, "Context Aware Middleware Architecture for Wireless Sensor Network," in *IEEE International Conference on Services Computing*, 2009.

[28] Y. Hu, Z. Wu, and M. Guo, "Ontology Driven Adaptive Data Processing In Wireless Sensor Networks," in *Infoscale '07*, Suzhou, China, 2007

[29] I. Horrocks et al., "SWRL: A Semantic Web Rule Language Combining OWL and RuleML". W3C member submission, 2004.

[30] S. Avancha, C. Patel, and A. Joshi, "Ontology-driven adaptive sensor networks," in *MOBIQUITOUS*, 2004.

[31] E. Dekneuvel., *Intelligent Sensor: Analysis and Design*, T. a. F. G. R. Zurawski, Ed. Embedded System Handbook, 2006.

[32] J. Cañada-Bago, M. A. Gadeo, J. A. Fernández, and V. J.R., "A Knowledge Based Wireless Sensor Network. ," in *European Wireless Sensors Network (EWSN09)*, Cork, Ireland, 2009.

[33] T. Srinivasan, R. Chandrasekar, and V. Vijaykumar, "A fuzzy, energy-efficient scheme for data centric multipath routing in wireless sensor networks. ," in *IFIP International Conference on Wireless and Optical Communications Networks.*, Bangalore, India,, 2006.

[34] C. X., H. T., R. R., and A. Elmaghraby, "A swarm-based fuzzy logic control mobile sensor network for hazardous contaminants localization. In Proceedings of the . ," in *1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'04).*, Fort Lauderdale, , 2004, p. 194–203.

[35] A. Kulakov and D. Davcev, "Tracking of unusual events in wireless sensor networks based on artificial neural-networks algorithms," in *IEEE International*

Conference on Information Technology: Coding and Computing, Washington, DC, USA, 2005, pp. 534-539.

[36] E. Benoit, R. Dapoigny, and L. Foulloy, "Fuzzy-Based Intelligent Sensor: Modelling, Design, Application," in *8th IEEE International Conference on Emerging Technologies,*, 2001, pp. 401-407,.

[37] A. Averkin and B. A.G., "Soft Computing in Wireless Sensors Networks," in *EUSFLAT*, Ostrava, Czech Republic., 2007, pp. 387-390.

[38] M. Marin-Perianu and P. Havinga, "D-FLER: A distributed fuzzy logic engine for rule-based wireless sensor networks ," in *International Symposium on Ubiquitous Computing Systems (UCS)*, Twente, Enschede, 2007, p. 86–101.

[39] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless Sensor Network Survey," *Computer Networks*, vol. 52, no. 12, pp. 2295-2330, Aug. 2008

[40] S. C. and J. T., "Web 2.0 and SOA: Converging concepts enabling the internet of services," *IT Proffesional*, vol. 9, no. 3, pp. 36-41, 2007.

[41] P. M. P., "Service-oriented computing: Concepts, characteristics and directions," in *Proc. of WISE*, Washington DC, USA, 2003.

[42] E. Meshkova, J. Riihijärvi, F. Oldewurtel, C. Jardak, and P. Mähönen, "Service-Oriented Design Methodology for Wireless Sensor Networks: A View through Case Studies," in *IEEE International Conference on Sensor Networks, Ubiquitious, and Trustworthy Computing*, 2008.

[43] M. R., *Agile Software Debelopment: Principles, Patterns, and Practices*. NJ, USA: Prentice Hall PTR Upper Saddle River, 2005.

[44] S. I., *Software Engineering*. Addison Wesley, 2006.

[45] P. Pankesh, J. Sunil, C. Sanjay, and R. Prabhat, "Context Aware Middleware Architecture for Wireless Sensor Network," in *IEEE International Conference on Service Computing*, 2009.

[46] C. L. Fok, G. C. Roman, and C. Lu, "Enhanced Coordination in Sensor Networks through Flexible Service Provisioning," in *11th International Conference on Coordination Models and Languages*, Lisboa, 2009, pp. 66-85.

[47] Evers, L., Havinga, P.J.M., Kuper, J., Lijding, M.E.M.; Meratnia, N., “SensorScheme: Supply chain management automation using Wireless Sensor Networks”, IEEE Conference on Emerging Technologies and Factory Automation, 2007. ETFA.

[48] M. Ivester, A. Lim, “Interactive and Extensible Framework for Execution and Monitoring of Wireless Sensor Networks”, First International Conference on Communication System Software and Middleware, 2006. Comsware 2006.

[49] Elson, J., Girod, L., Estrin, D., “EmStar: Development with high system visibility”. IEEE Wireless Communications, 2004.

[50] Levis, P., Lee, N., “TOSSIM: A simulator for TinyOS networks”. http://www.cs.berkeley.edu~pal/pubs/nido.pdf. 2006.

[51] B. Zhang, G. Li, “Analysis of Network Management Protocols in Wireless Sensor Network”, Proceedings of the International Conference on MultiMedia and Information Technology, pp. 546 – 549, 2008.

[52] W. L. Lee, A. Datta, and R. Cardell-Oliver, Network Management in Wireless Sensor Networks, to appear in Handbook on Mobile Ad Hoc and Pervasive Communications, edited by M. K. Denko and L. T. Yang, American Scientific Publishers, 2006.

[53] I.F. Akyildiz, T. Melodia and K.R. Chowdhury, “A Survey on Wireless Multimedia Sensor Networks”, Computer Networks Elsevier. 51, pp. 921–960, 2007.

[54] L. Frye and L. Cheng, “Network management of a WSN”, Technical Report LU-CSE-07-003. Lehigh University. 2007.

[55] W. Liu, Y. Zhang, W. Lou, and Y. Fang, “Managing Wireless Sensor Network with Supply Chain Strategy,” Proceedings of the IEEE Quality of Service in Heterogeneous Wired/Wireless Networks, 2004.

[56] G. Tolle and D. Culler, “Design of an Application-Cooperative Management System for Wireless Sensor Networks,” Proceedings of the Second European Workshop on In Wireless Sensor Networks, 2005.

[57] W. Louis Lee, A. Datta, and R. Cardell-Oliver, “WinMS: Wireless Sensor network-Management system, An Adaptive Policy-based Management for Wireless Sensor Networks”. Technical Report UWA-CSSE-06-001. University of Western Australia, 2006.

[58] H. Song, D. Kim, K. Lee, and J. Sung, “Upnp-Based Sensor Network Management Architecture,” in Proceedings of the International Conference on Mobile Computing and Ubiquitous Networking, 2005.

[59] L.B. Ruiz, J.M. Nogueira, and A.A.F. Loureiro, “MANNA: A Management Architecture for Wireless Sensor Networks,” IEEE Communications Magazine, Vol. 41, No. 2, pp 116–125, 2003.

[60] G. Wagenknecht, M. Anwander, T. Braun, T. Staub, J. Matheka and S. Morgenthaler, “MARWIS: A Management Architecture for Heterogeneous Wireless Sensor Networks”. In Proceedings of the International Conference on Wired/Wireless Internet Communications. LNCS 5031, pp. 177–188, 2008.

[61] I. Akyildiz, T. Melodia, and K. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, 2007, pp. 921-960.

[62] S. Soro, W.B. Heinzelman, and On, "the coverage problem in video-based wireless sensor networks, in: Proc," *of the IEEE Intl. Conf. on Broadband Communications, Networks, and Systems (BroadNets), Boston, MA, USA, October*, 2005.

[63] M. Rahimi, D. Estrin, and J. Villasenor, "Energy-Aware High Resolution Image Acquisition via Heterogeneous Image Sensors," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, 2008, pp. 526-537.

[64] D. Lymberopoulos and A. Savvides, "XYZ: a motion-enabled, power aware sensor node platform for distributed sensor network applications," *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, Ieee, , pp. 449-454.

[65] I. Akyildiz, T. Melodia, and K. Chowdhury, "Wireless Multimedia Sensor Networks: Applications and Testbeds," *Proceedings of the IEEE*, vol. 96, 2008, pp. 1588-1605.

[66] F. Anjum and P. Mouchtairs, *Security for wireless AD Hoc networks*. Wiley-Interscience, 2007.

[67] K. N. Randall and C. L. Pano, *Seguridad para comunicaciones inalámbricas*. McGraw Hill, 2003.

[68] M. Saraogi, *Security in Wireless Sensor Networks*. Department of Computer Science University of Tennessee, 2004.

[69] A. Perrig, J. Stankovic, and D. Wagner, *Security in wireless sensor networks*. Communications of ACM, 2004.

[70] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, *Wireless Sensor Network Security: a Survey*. CRC Press, 2006.

[71] Gunnar Gaubatz, Jens-Peter Kaps, Erdinç Öztürk, Berk Sunar. “State of the Art in Ultra-Low Power Public Key Cryptography for Wireless Sensor Networks”.

Cryptography & Information Security Lab. Worcester Polytechnic Institut. http://www.crypto.wpi.edu/Publications/Documents/GaubatzKapsPerSec05.pdf

[72] John Paul Walters, Zhenqiang Liang, Weisong Shi, Vipin Chaudhary. “Wireless Sensor Network Security: A Survey”. Department of Computer Science. Wayne State University. http://www.cs.wayne.edu/~weisong/papers/walters05-wsn-security-survey.pdf

[73] Ronald L. Rivest. “The RC5 Encryption Algorithm”. MIT Laboratoy for Computer Science. Marzo 1997. http://people.csail.mit.edu/rivest/Rivest-rc5rev.pdf

[74] Yee Wei Law, Jeroen Doumen, Pieter Hartel. “Survey and Benchmark of Block Ciphers for Wireless Sensor Networks”. Faculty of Electrical Engineering, Mathematics and Computer Science. University of Twente. http://doc.utwente.nl/57007/1/000000eb.pdf

[75] Vikram Reddy Andem. “A Crpyptanalysis of the Tiny Encryption Algorithm”. Department of Computer Science. University of Alabama. 2003. http://cs.ua.edu/SecurityResearchGroup/VRAndem.pdf

[76] Wen Chan Shih, Wen Hu, Peter Corke, Wen Chan Shih, Leslie Overs. “secFleck: A Public Key Technology Platform for Wireless Sensor Networks”. Autonomous Systems Laboratory. CSIRO ICT, Australia. http://www.cse.unsw.edu.au/~wenh/shih_tpm_poster.pdf

[77] Jens-Peter Kaps. “Chai-Tea, Cryptographic Hardware Implementations of xTEA”. Volgenau School of IT&E. George Mason University. http://www.springerlink.com/content/w1788t1644215uv1

[78] López L., Hernández V., Maján A., Martínez J.F., García A.B. y Dasila A. “Análisis de consumo energético y tiempo de proceso en cifrado AES en Redes Inalámbricas de sensores”. JITEL 2009 ISBN:978-84-96997-27-1 (pp.451-454).

[79] Germano Guimaraes, Eduardo Souto, Djamel Sadok, Judith Kelner. “Evaluation of Security Mechanisms in Wireless Sensor Networks”. Informatics Center. Federal University of Pernambuco. Proceedings of the 2005 Systems Communications (ICW ‘05). 2005. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1515560

[80] Rodrigo Roman, Cristina Alcaraz. “Applicability of Public Key Infrastructures in Witeless Sensor Networks” http://www.lcc.uma.es/~roman/files/roman-europki07.pdf

[81] The .NET Micro Framework References. [En línea http://msdn.microsoft.com/en-us/library/cc533014.aspx].

[82] Microsoft.SPOT.Cryptography Namespace. [En línea http://msdn.microsoft.com/en-us/library/cc531757.aspx].

[83] Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications). http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0058:EN:HTML

[84] A. Perring, J. Tygar. “Secure Broadcast Communication in Wired and Wireless Networks”. Kluwer Academic Publishers. 2003.

[85] Y. Yang, X. Wang, S. Zhu, G. Cao. “SDAP: A Secure hop-by-hop data aggregation protocol for sensor networks”. Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc. Florencia. Mayo 2006. http://www.cse.psu.edu/~xinrwang/papers/sdap.pdf

[86] L. Donggang, N. Peng. “Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks”. Proceeding of the 10th Annual Network and Distributed System Security Symposium. San Diego. 2003. http://www.isoc.org/isoc/conferences/ndss/03/proceedings/papers/17.pdf

[87] B. Mathias, T. Wade. “An Authentication Framework for Hierarchical Ad Hoc Sensor Networks”. Proceedings of ACM Workshop on Wireless Security (WiSE ‘03). San Diego. 2003. http://www.springerlink.com/content/r5g81j5169m61q76

[88] Z. Benenson, R. Gedicke. =. Raivio. “Realizing Robust User Authentication in Sensor Networks”. Workshop on Real-World Wireless Sensor Networks. REALWSN05. http://www.sics.se/realwsn05/papers/benenson05realizing.pdf

[89] Taojun Wu, Yi Cui, Brano Kusy, Akos Ledeczi, Janos Sallai, Nathan Skirvin, Jan Wermer, Yuan Xue. “A Fast and Efficient Source Authentication Solution for Broadcasting in Wireless Sensor Networks”. Institute for Software Integrated Systems (ISIS) and EECS. Vanderbilt University. http://www.truststc.org/pubs/206.html

6 Do-it-Yourself devices

Task 2.4 aims at providing savvy users the means to assemble and program hardware that can be used by end-users to customize their smart environment. This section will provide a brief survey on the existing Do-it-Yourself hardware platforms and identify the technical challenges that need to be addressed to simplify the exposure of DiY devices in the DiYSE platform.

6.1 Existing hardware platforms

A list of existing devices that can be connected to DiYSE has been provided in section 2.1. This section describes a few illustrative examples of inexpensive and open hardware platforms with a strong community support that hobbyists can use to create their own DiY devices.

6.1.1 Phidgets kits

Just as widgets make GUIs easy to develop, Phidgets [1] (or physical widgets) are building blocks that make the new generation of physical user interfaces easy to develop.

The “Phidgets Interface Kit” is a board powered and controlled by a computer’s USB port. It features a number of analog inputs and outputs to which different kinds of low cost sensors and actuators can easily be plugged. The interaction with the “Phidgets” is done through very simple APIs, consistent across a large number of programming languages, ranging from Java to Microsoft Excel.

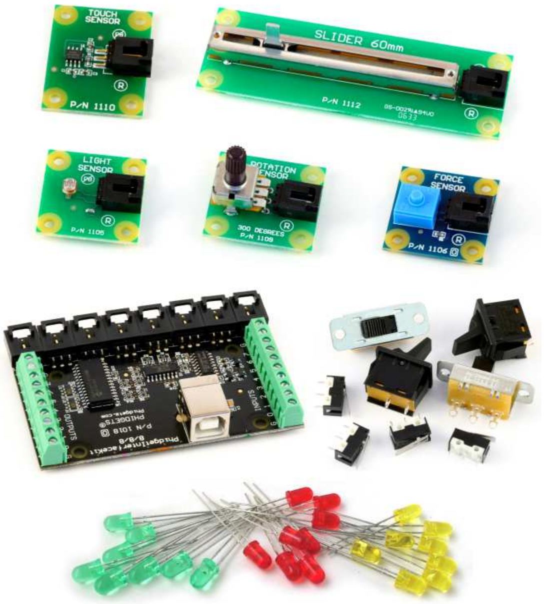


Figure 6 - The Phidgets Interface Kit Package #1 ($130)

6.1.2 Arduino microcontroller kits

A number of DiY microcontroller projects and ready-to-use kits exist, such as Arduino[2], MAKE Controller Kit [3], Parallax BASIC Stamp [4], NetMedia BasicX [5] Amongst all of them, Arduino is the one that has succeeded in drawing the attention of the Internet community, partly due its low cost, to its open nature that has motivated a number of clone boards (*duino) and especially to its ease of use by non-experts in electronics or programming.

Arduino is a simple open hardware design based on the Atmel AVR family of microcontrollers. It can easily be programmed using Wiring, a simple programming language, and the associated Arduino IDE.

Arduino features add-on modules called “shields”, which allow to add preassembled circuits to the main board to control motors, add Ethernet, Bluetooth, ZigBee, etc.

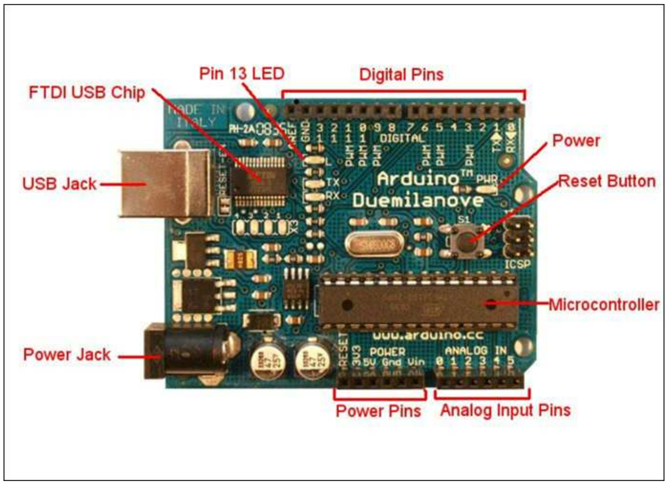
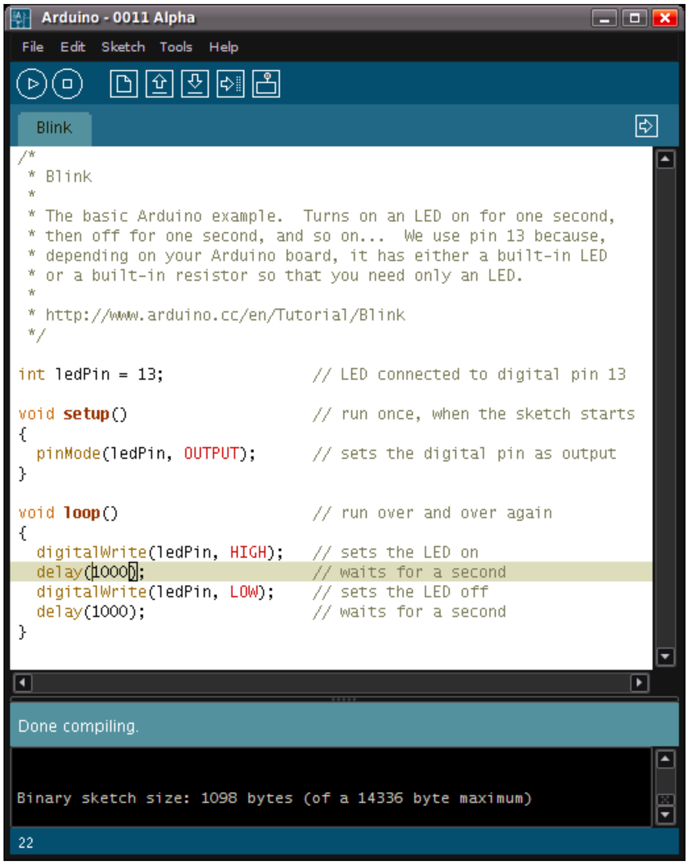


Figure 7 - Arduino board in its USB version, codename "Duemilanove" ($30)



```
/*
 * Blink
 *
 * The basic Arduino example.  Turns on an LED on for one second,
 * then off for one second, and so on...  We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;                // LED connected to digital pin 13

void setup()                    // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()                     // run over and over again
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(1000);                  // waits for a second
}
```

```
Done compiling.

Binary sketch size: 1098 bytes (of a 14336 byte maximum)
```

Figure 8 - Arduino IDE with a simple LED blinking program

Arduinos can be used to create devices that operate in different ways:

- **Standalone non-networked device**: it can interact only through input and output sensors and actuators, not with other devices. An example is a “DiY mood lamp” that changes color randomly.
- **Computer peripheral**: device is connected to a computer through serial, USB, or Bluetooth. It is programmed to act as a slave of a program running in the computer, just like Phidgets (Bitlash [6] or Firmata [7] libraries are useful for this). A typical project using this approach is a computer-controlled railway model. This is the simplest communicating setup as all microcontrollers have at least a serial port. The drawback is that in order to operate the device, the computer needs to be running.
- **Networked device**: Arduino boards can participate in a network, either using Ethernet or ZigBee shields with the corresponding libraries. In this set up, different Arduino-based devices can talk to each other, to computers in the LAN or even to servers on the Internet (a gateway is still required for ZigBee to IP bridging). This is the most interesting configuration for DiYSE as it enables interaction between the application creation environment and the devices.

Extensive support for the usage of Arduino is available online or in the “Making things talk” book [8].

6.1.3 Embedded GNU/Linux-based devices

A huge number of embedded devices today run *NIX-like operating systems such as GNU/Linux and are open or hacked so that they can run custom software. Their low cost, small size, low consumption and lack of noisy fans makes them perfect platforms for DiY IP devices. Indeed, peripherals such as Phidgets can be attached thus turning them into IP ambient devices that can be placed in buildings, vehicles or outdoor, powered by AC, batteries or even solar panels.

A few examples of such devices include:

- Single-board computers (PhidgetSBC, Gumstix, Beagle Board…)
- A large amount of inexpensive wireless routers (Linksys, LaFonera…) that can reprogrammed with open firmwares (OpenWRT, DD-WRT…)
- Game consoles such as PlayStation 2 and 3, Xbox and Xbox 360
- Mobile terminals with open firmwares and USB host capabilities (OpenMoko, Maemo, Android…)

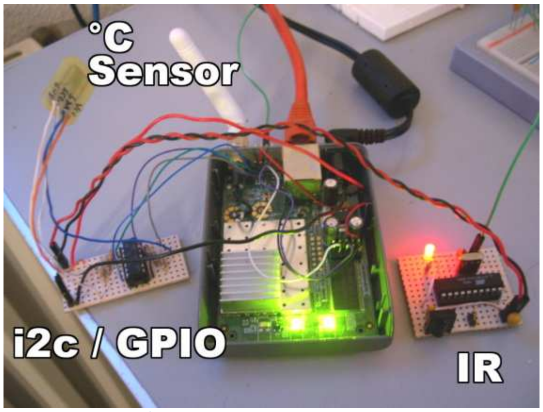


Figure 9 – Hacked Fonera WiFi router ($30) connected to thermometer and infrared receiver (source: lefinnois.net)

6.2 Challenges for the ease of integration

This section will focus on how to easily connect self-programmed or self-assembled devices to a common infrastructure so that they can be accessed from the DiYSE application creation environment.

From the connectivity point of view, Do-it-Yourself hardware platforms fall into one of the following categories:

- IP devices
- Networked non-IP devices
- Computer peripherals

Non-networked devices or devices that communicate with each other without interacting with computers or IP networks at all (e.g. a WiiMote controlling an Arduino-based Bluetooth lamp) are out of the scope of the DiYSE project because of the lack of obvious means to link them to the DiYSE application creation environment.

6.2.1 IP devices

This category includes products that are intended to be almost permanently connected to an IP network, either through LANs or cellular networks (using GPRS or 3G). Some examples are:

- Open IP camera platforms (Axis…)

- Open ambient consumer electronics (Nabaztag, Tux Droid, Chumby, Roomba robot…)
- Open hardware platforms (Bug Labs…)
- Open or hacked Linux-powered devices, such as listed above.
- TCP/IP enabled microcontrollers (such as Arduino with Ethernet shield) or "modems" (such as Lantronix XPort and WiPort [9] or the Digi Connect family [10])

In general, IP devices host servers exposing their built-in functions, either through TCP or UDP servers, HTTP servers (possibly using a RESTful resource model [11]), UPnP [12] or DPWS [13]. Often, they are open platforms where new functionalities can be implemented using Software Development Kits. In that case, user-added functions should also be exposed as services in the network.

As IP devices, they can directly participate in network interactions without intermediaries, either on the local network or through the Internet. For a device to be directly accessible through the Internet, a number of issues, which are common to final products and DiY devices, need to be addressed:

- **Addressing**: the device needs to have a public IP on the Internet and NAT traversal techniques are required to allow inbound connections.
- **Discovery**: the device needs to be discovered, either through a dynamic lookup or in a directory. Most dynamic discovery protocols, such as the ones used by Zeroconf, UPnP and DPWS, rely on multicast UDP datagrams, that don't propagate outside a LAN without a proxy.
- **Security**: authentication and authorization will be necessary if device is publicly accessible.

UPnP and DPWS protocol stacks fully match the needs of plug-and-play IP devices and open-source tools enable the development of the service software for many embedded platforms. However, the required effort make it more suited for industrial companies than for hobbyists.

Using an HTTP server (preferably using a RESTful resource model) is a good compromise, as HTTP servers are available in binary format for all kinds of IP devices and programming can be simply done through server-side scripts. However additional effort needs to be done to address the following missing functionalities:

- **Discovery**: DiYSE devices should be easy to plug-and-play in order to avoid non-experts unnecessary configuration, such as manually entering IP addresses.
- **Description**: in addition to discovery, the protocol enabling to control the device should be described or the software component enabling its usage (driver) made available in order to avoid users manually describing the interaction protocol.
- **Eventing**: if devices are meant to receive commands and notify events, a callback mechanism should be implemented. For instance, the Pachube system uses web hooks,

6.2.2 Computer peripherals

This category groups devices that will only work when connected to computers through wired connections (such as serial, USB or FireWire) or wireless personal area networks (such as the IEEE 802.15 family: Bluetooth, ZigBee, UWB, etc.). It can include:

- Wired peripherals such as Phidgets, modified USB gadgets or game controllers
- Bluetooth peripherals such as WiiMote
- Microcontroller working as an input or output serial peripheral

These devices will need a device-specific software "driver" running on the computer which will hide the specificities of the peripheral to the controlling application, which is independent of the underlying communication protocol.

Assuming that the semantics and coordination of the device interaction is ensured by the upper layers, the remaining challenge for a Do-it-Yourself usage of this category of devices is the seamless search and deployment of the appropriate device drivers.

In order to achieve a sense of interaction without computers, these peripherals can be connected to the one of the abovementioned embedded GNU/Linux-based computers, which can run permanently, silently and hidden. In this case, a network service exposing the peripheral to remote user applications would be required.

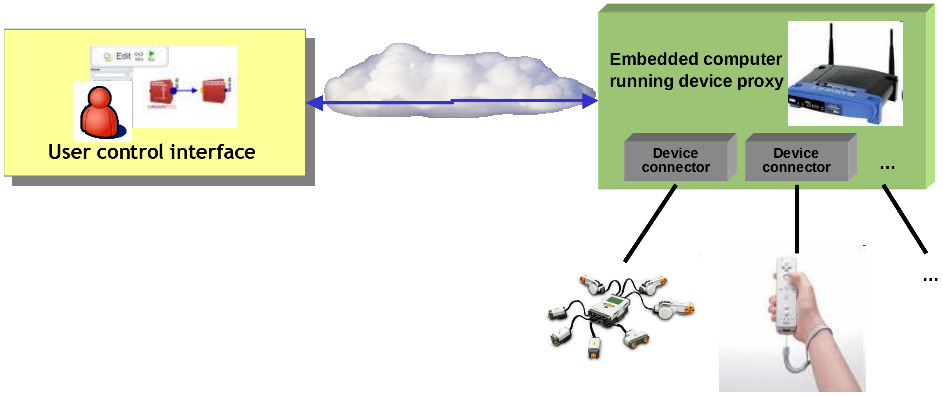


Figure 10 - High-level architecture of a embedded computer running a device proxy

6.2.3 Networked non-IP devices

This category groups devices that communicate using networks that do not use full TCP/IP stacks. As an example, an increasing number of hobbyists that want wireless battery-powered devices buy ready-to-use ZigBee modules [14]. These modules are easy to use as they behave like simple modems (communication uses serial and configuration is done through Hayes AT commands). They form wireless networks with star or mesh topologies that self-configure and route messages.

In general terms, two options are available for the integration of such networks with IP networks:

- Having a computer ("gateway") run a program that controls the whole network, hides its underlying complexity and exposes it as a service or "virtual device". The drawback is that this program would be usage-specific. The advantage is the solution is simple to implement, as it is equivalent to exposing a peripheral.
- Building a solution in which the gateway is generic (processing takes place in the routing layer but not in the application layer), so that new types of unforeseen nodes can join the network and should work and any application in

the IP network can access each node in the network. The state-of-the-art solution to implement this on top of low-power devices is 6loWPAN [15], an IETF specification for the usage of IPv6 with compressed headers on top of low-power devices (both wired and wireless). In this case, the role of the gateway is solely to compress and uncompress the IPv6 headers. UDP and ICMP protocols are already supported in a number of devices. The remaining issue is that running a TCP stack on such low-power devices, even if feasible, is over-engineering for its actual usage. In the application layer, current works, such as 6lowAPP [16], propose the usage of simple protocols similar to REST. The development of such protocols and development libraries for non-experts remains a challenge to be addressed.

6.3 References

[1] Phidgets website: shop, libraries and documentation
http://www.phidgets.com/

[2] Arduino community website
http://arduino.cc/

[3] MAKE Controller Kit website
http://makezine.com/controller/

[4] Parallax online store
http://www.parallax.com/

[5] Home of the BasicX microcontroller
http://www.basicx.com/

[6] Bitlash library for Arduino
http://bitlash.net

[7] Firmata library for Arduino
http://www.firmata.org/

[8] Tom Igoe, “Making things talk”, O’Reilly, ISBN 978-0-596-51051-0

[9] Lantronix XPort and WiPort product family
http://www.lantronix.com/device-networking/embedded-device-servers/

[10] Digi Connect product family
http://www.digi.com/products/embeddedsolutions/digiconnectme.jsp

[11] REST Wikipedia page
http://en.wikipedia.org/wiki/Representational_State_Transfer

[12] UPnP Wikipedia page
http://en.wikipedia.org/wiki/UPnP

[13] DPWS Wikipedia page
http://en.wikipedia.org/wiki/DPWS

[14] XBee family of products
http://www.digi.com/products/wireless/point-multipoint/

[15] 6loWPAN specification, IETF RFC 4944
http://en.wikipedia.org/wiki/6loWPAN

[16] IETF 6lowAPP working group
http://6lowapp.net/