



Università degli Studi di Milano-Bicocca
Dipartimento di Informatica, Sistemistica e Comunicazione (DISCo)
Dottorato di Ricerca in Informatica – XXII Ciclo
Anno Accademico 2008-2009

TRACKING WITH HIGH-DENSITY, LARGE-SCALE WIRELESS SENSOR NETWORKS

Davide Merico

Thesis supervisor: Prof. Roberto Bisiani

Thesis tutor: Prof. Raimondo Schettini

Ph.D. Coordinator: Prof. Stefania Bandini

*You remind me,
define me.
Incline me.*

If you died I'd.

To Alessandra with endless Love.

Acknowledgments

First of all, I would like to thank, with deep gratitude, my thesis advisor Prof. Roberto Bisiani. Roberto has been more than a perfect advisor during my research project. He has been a mentor, a guide, and a good friend.

My special thanks go to Alessandra Mileo and Torsten Schaub for inspiring discussions when working on the logic-based techniques and for their continuous support.

Without Julia Weekes this thesis would probably have been written in Italian. Julia, your English course is truly awesome!

I wish to express my gratitude to prof. Davide Diamantini and prof. Guido Martinotti for their precious support during these years.

I would also like to thank current and past colleagues Massimo Ballerini, Daniela Battagin, Simone Belleli, Raffaele Briganti, Francesco Colleoni, Roberto De Nardo, Matteo Dominoni, Emanuel Doneda, Fabio Malizia, Glauco Mantegari, Sara Manzoni, Luca Marinosci, Valerio Minetti, Michelle Pieri, Stefano Pinardi, Andrea Pozzali, Elisa Ribauda, Maurizio Pustorino, Giorgio Riva, Luigi Riva, Sergio Ruocco, Davide Tasca, Giuseppe Vizzari, in the NOMADIS Lab and DISCo for providing an exciting working environment and for their precious help.

Last but not least, I would like to say thanks to my beautiful Alessandra for her love, patience and encouragement.

To all of you, thanks! This work would not have been possible without your support.

Milan, October, 2009

Davide Merico

Let proportion be found not only in numbers and measures, but also in sounds, weights, times, and positions, and what ever force there is.

– Leonardo Da Vinci –

Contents

1	Introduction	1
1.1	The Goal of the Thesis	2
1.2	Contributions	2
1.3	Organization of the Thesis	3
2	People and Object Tracking State-of-the-Art	5
2.1	Basic Concepts	6
2.1.1	Ranging	6
2.1.2	Localization	10
2.1.3	Tracking	12
2.2	Evaluation of Localization and Tracking Systems	13
2.2.1	Evaluation Results and Further Considerations	17
2.3	Chapter Summary	19
3	Large Scale Wireless Sensor Networks	21
3.1	WSN Node Architectures	22
3.2	Communication Protocols	23
3.3	Localization in Wireless Sensor Networks	25
3.4	Chapter Summary	27
4	The DiGS Data Gathering System	29

4.1	Introduction to DiGS	29
4.2	DiGS Data Gathering Devices	32
4.2.1	The JN5139 Wireless Microcontroller	32
4.2.2	DiGS Fixed nodes	35
4.2.3	DiGS Mobile Nodes	39
4.3	DiGS Data Gathering Software	43
4.3.1	Hierarchical Network Organization	43
4.3.2	The DiGS-MB Algorithm	44
4.3.3	The DiGS-OB Algorithm	45
4.4	DiGS Evaluation	47
4.4.1	Time Synchronization	48
4.4.2	Evaluation of In-network Position Computation	50
4.4.3	Current Consumption Calculation	52
4.5	Chapter Summary	55
5	Target Tracking	57
5.1	Introduction	57
5.2	The BTTF Framework	59
5.2.1	BTTF Tracking Engine	60
5.2.2	The SIRS Algorithm	61
5.2.3	BTTF Particle Filter. Movement Model Details	62
5.2.4	BTTF Particle Filter. Sensor Models	64
5.3	BTTF Evaluation	65
5.4	Chapter Summary	70
6	Situation Assessment	71
6.1	Introduction	71
6.2	Data Generation with Repast Symphony	74

6.2.1	Agent-Based Simulation Frameworks	74
6.2.2	Definition of the Simulation Scenario	78
6.2.3	Context and Projections	81
6.2.4	Event Scheduling	83
6.2.5	Agents and other Objects	85
6.3	Logic-based Situation Assessment	86
6.3.1	Answer Set Programming: some notions	88
6.3.2	Definition of a Context Model	92
6.3.3	Intelligent Support to Localization	97
6.3.4	Discussion	102
6.4	Chapter Summary	104
7	A Case Study, the SINDI System	105
7.1	Introduction	106
7.1.1	System Requirements and Design Issues	108
7.2	Knowledge Representation of the Home Healthcare Domain	112
7.3	The Reasoning Capabilities	119
7.4	Evaluation	125
7.5	Chapter Summary	127
8	Conclusions	129
Appendix A		
	Schematics of Data-Gathering Nodes	131
Appendix B		
	Document Type Definition of the Data Generation Tool	135

List of Figures

2.1	Time of Arrival and Time Difference of Arrival	7
2.2	Dead Reckoning	10
2.3	Trilateration method with an ideal situation	11
2.4	Evaluation of Localization Systems. Testing Environment. . .	14
2.5	Experiments of Indoor Dead Reckoning with GyroDRM and DRC.	16
3.1	Main Components of WSN Nodes	22
3.2	APIT and Min-Max algorithms	26
4.1	Hierarchical Network Organization: the Infrastructure nodes and the Mobile nodes	30
4.2	Jennic JN5139 Block Diagram.	33
4.3	DiGS Fixed Node hardware schema.	35
4.4	Base Board of the DiGS fixed nodes.	36
4.5	Sensor Board of the DiGS fixed nodes.	36
4.6	DiGS Fixed Node with Enclosure.	37
4.7	Hardware schema of the DiGS Mobile Node first prototype. . .	40
4.8	The First Prototype of the DiGS Mobile Node.	40
4.9	Hardware schema of the DiGS Mobile Node current implemen- tation.	41
4.10	DiGS final prototype main components and layout.	42

4.11	DiGS-OB Broadcast Message Payload.	45
4.12	The DiGS-OB Scheduling Algorithm	46
4.13	Time synchronization error between two nodes. The Time Synchronization was stopped after 30 minutes.	48
4.14	RSSI values and distance	49
4.15	Radiation Pattern for JN5139 Ceramic Antennas.	49
4.16	Radiation Pattern for JN5139 Plastic-External Antennas.	50
4.17	Radiation Pattern for JN5139 Ceramic Antennas.	51
4.18	Effects of Sleep Period on Battery Life, Normalised to 1 Year	55
5.1	The Main Components and the Data Flow of the BTTF Framework.	59
5.2	Detail of the BTTF Tracking Engine	60
5.3	Inertial-Aided Motion Model	63
5.4	Generic Motion Model	64
5.5	RSSI Sensor Model	65
5.6	Rangefinder Sensor Model.	65
5.7	Evaluation of the BTTF Framework. Floor Plans of the Testing Environments with Sensor Positions.	67
6.1	Levels of Situation Awareness from the Endsley's Model	72
6.2	A Taxonomy of Selected ABM Tools	75
6.3	Nested hierarchy of Swarms	76
6.4	Interaction between Contexts and Projections	77
6.5	Projections used by the Repast S model and their relations	82
6.6	The Repast S toolkit showing the main Projection, walls (black squares) and the person (red star).	87
6.7	The Environmental Data Projection showing the light cell (green circles), environmental sensors (red crosses) and cells (purple squares).	87

6.8	Example: Modelling a Bedroom	94
7.1	Size of the Italian population by age groups: each age group is separately plotted and normalized to the group size in 1950 (Source ISTAT and IRP-CNR).	106
7.2	Data flow.	108
7.3	Architecture overview.	111
7.4	Information flow across levels.	114
7.5	Caregiver interface to access context data.	116
7.6	Example 7.3.1. ADL dependencies, inference process with re- spect to prediction.	123
7.7	Example 7.3.2. ADL dependencies, inference process with re- spect to prediction.	124
1	DiGS Fixed nodes: Base Board	132
2	DiGS Fixed nodes: Sensor Board	133
3	DiGS Mobile nodes	134

List of Tables

2.1	Average Errors	15
2.2	Evaluation of Localization Systems	18
4.1	JN5139 DC Current Consumption (VDD=2.2 to 3.6V, -40 to +85° C)	34
4.2	Performance of Sensirion SHT11 (25°C and 3.3V)	39
4.3	Evaluation Results of the In-network Position Calculation . . .	51
4.4	Activities of the Base Node and Related Current Consumption and Time Duration	53
4.5	Activities of the Broadcast Node and Related Current Con- sumption and Time Duration	53
4.6	Activities of the Mobile Node (Tag) and Related Current Con- sumption and Time Duration	54
5.1	Comparison between BTTF and Ubisense	68
5.2	Comparison between BTTF and Ubiquicom	69
6.1	Generic Spatial Relations Among Entities.	93
6.2	Attributes representing information about the Person entity. .	95
6.3	Attributes representing information about Room and Area en- tities.	95
6.4	Attributes representing information about the Object entity. .	96
7.1	Logical encoding of patient's profile.	113

7.2 Logical encoding of dynamic profile. 113

7.3 ASP reasoning performance (time is expressed in seconds). . . 127

Listings

6.1	XML-based Scenario Definition: Walls, Rooms, Areas, Passages.	79
6.2	XML-based Scenario Definition: Localization Sensors.	80
6.3	XML-based Scenario Definition: Environmental Sensors.	80
6.4	XML-based Scenario Definition: Lights and Heat.	81
6.5	XML-based Person Schedule Definition: Activities and Actions.	84
6.6	An Short Example of Schedule Definition for Daylight, Temperature and Humidity.	85
6.7	Partial Description of an Example Environment in terms of Propositional Predicates	97
6.8	ASP Encoding for the Case When Results from Bayesian Filters and Associated Likelihoods are Available	99
6.9	Example of the use of inertia law	101
6.10	Consistency Constraint for Interaction with Multiple Objects .	101
6.11	Example of Combination of Object Attributes	102
1	Document Type Definition of the Scenario XML File	135
2	Document Type Definition of the Person Schedule XML File .	138
3	Document Type Definition of the Daylight, Temperature, Humidity Schedule XML File	139

Chapter 1

Introduction

“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it”.

This is the incipit of Mark Weiser’s frequently-cited masterpiece written in September 1991 and entitled *“The Computer for the 21st Century”* [1]. In this work, Weiser envisions living environments pervaded by a high number of invisible technological devices affecting and improving all aspects of our lives. This paradigm is commonly described as ubiquitous computing, pervasive computing, ambient intelligence, or, more recently, “everyware” [2].

Given the continuous technological advances in computing and communication, it seems that we are rapidly heading towards the realization of Weiser’s vision.

It is easy to justify the need, by these emerging paradigms, of knowing the physical location of users. Outdoor location-aware applications are already widespread today, their growing popularity showing that location-awareness is indeed a very useful functionality.

Less obvious is how the growing availability of these locations and tracks will be exploited for providing more intelligent “situation-understanding” services that help people.

My work is motivated by the fact that, thanks to location-awareness systems, we are more and more aware of the exact positions of the users but unfortunately we are rarely capable of exactly understanding what they are doing.

Location awareness should rapidly evolve and become *situation-awareness* otherwise the ubiquitous-computing vision will become impracticable.

1.1 The Goal of the Thesis

The goal of this thesis is devising alternative and innovative approaches to the problem of indoor position estimation/assessment and evaluating them in real environments. These approaches should be based on:

- a low-cost and energy-aware localization infrastructure;
- multi-sensor, statistically-based, localization algorithms;
- logic-based situation assessment techniques.

The need of a low-cost and energy-aware localization infrastructure can be argued by considering that we will soon be forced to better control our buildings in order to reduce energy consumption while wireless and battery-operated sensors will be used (i.e. for sensing and actuation of all the parameters that influence the state of a building). Therefore, it will be common to have a large number of wirelessly connected sensors deployed in every building.

Given these particular assumptions, none of the current approaches or localization systems have a fully satisfactory performance.

1.2 Contributions

The main contributions of this dissertation are following:

- the algorithm invention, design, implementation and evaluation of the hardware and software components of a hierarchical data-gathering systems called *DiGS*.
- the algorithm invention, design, implementation and evaluation of a novel target tracking framework that fuse a number of different data sources into a robust tracking algorithm called *BTTF*.

- the extension of target tracking to “situation assessment” where the system brings to bear a number of different and extremely varied sets of information and processes them within a logical framework instead of a statistical one.
- the design and implementation of an agent-based simulator used to generate large sets of environmental data. This simulator is an important contribution in itself and its results are more general and flexible than what would have been strictly necessary to support this thesis.

1.3 Organization of the Thesis

Besides the necessary problem descriptions and state-of-the-art analysis the thesis is organized as the description and solution of increasingly difficult problems all the way to the forefront of what is possible today.

Chapter 2 describes the general background of tracking and outlines related work and the state-of-the-art. Furthermore, the performance of the relevant existing systems is compared.

Chapter 3 examines large scale Wireless Sensor Networks (from now on called WSN) since this technology, albeit developed for different purposes, offers a good opportunity for solving the problem of implementing a pervasive tracking infrastructure.

Chapter 4 describes new algorithms for implementing a hierarchical data-collection wireless network and is one of the novel contributions of the thesis. This chapter also describes a new kind of WSN node designed and built expressly for this work because nothing suitable was commercially available.

Chapter 5 is both a review on existing statistical tracking techniques and the description of a novel framework for fusing a number of different data sources into a robust tracking algorithm.

Chapter 6 extends tracking to a new level of “awareness” bringing to bear new ambient data and opens the possibility of “reasoning” not only on simple tracking but also on the situation at hand. A large part of Chapter 6 is dedicated to the description of an agent simulator that has been built to generate environment data on a large scale (both in number of sensors and in length of time). This simulator is an important contribution in itself and its structure is more general and flexible than what would have been strictly necessary to support this thesis.

Chapter 7 describes in detail a system that makes use of all the hardware, software, algorithms and techniques developed in this thesis in order to show how the general idea of tracking can evolve into situation assessment and then be used for a wider range of applications like monitoring and prevention.

Chapter 8 wraps it all up.

Chapter 2

People and Object Tracking State-of-the-Art

This chapter surveys various localization and tracking systems and the techniques they use to compute locations and to track users.

The relevance of a localization system depends on the applications it enables and this, in turn, depends on some of the technology's features (precision in 2D and 3D, frequency of position acquisition, size of the tag on the mobile entity, size of the infrastructure system, cost and power constraints).

Almost all localization technologies require some sort of infrastructure. For example, GPS [3] and Galileo [4] rely on a set of satellites, while most of the others rely on fixed ground stations. Since the position is computed by exchanging radio signals that are easily blocked or deflected by physical structures, the satellite-based systems are almost useless indoor.

The position is commonly computed by acquiring the distance from three or more points of known position and then applying trilateration. The distance can be computed by measuring variations of signal intensity or by timing the latency from transmission to reception.

All radio-positioning technologies can be augmented by some form of dead reckoning. For example speed and direction can be used to get a fix when the positioning signal is not available.

Section 2.1 gives a general overview of the basic concepts and techniques that can be used for localization and tracking. Section 2.2 introduces different

existing systems and evaluates their performance.

2.1 Basic Concepts

Electromagnetic waves and sound waves are widely used for the determination of the location of objects. The localization and tracking process relies on ranging. This section introduces the basic concepts and techniques that can be used for localization and tracking.

2.1.1 Ranging

To position an object or a device, the basic step is to use a reference point to determine the distance and angle between the device and the reference point. This subsection describes several of these basic approaches.

Proximity Proximity-based techniques involve determining when an object is in the neighbourhood of a known location. The presence of this object is sensed using some physical phenomenon with limited range.

When only mere connectivity information is available, these are the simplest ranging techniques that can be used in order to obtain useful positioning measurements. The key advantage of these techniques is that they do not require any dedicated hardware and time synchronization among nodes since the connection information is already available in wireless devices.

In cellular systems, proximity sensing is known by the terms Cell of Origin (CoO), Cell Global Identity (CGI), or simply Cell-ID.

Time of Arrival (ToA) In the Time of Arrival (ToA - sometimes Time of Flight, ToF) approach, signal traveling time is used to estimate the distance between a device and the reference point. Such systems typically use signals that move at a slower speed, such as ultrasound, to measure the time of signal arrival. Figure 2.1(a) illustrates this idea. An ultrasound signal is sent from the transmitter to the receiver; in return, the receiver sends a signal back to the transmitter. After this two-way handshake, the transmitter can infer the distance from the round-trip delay of the signals.

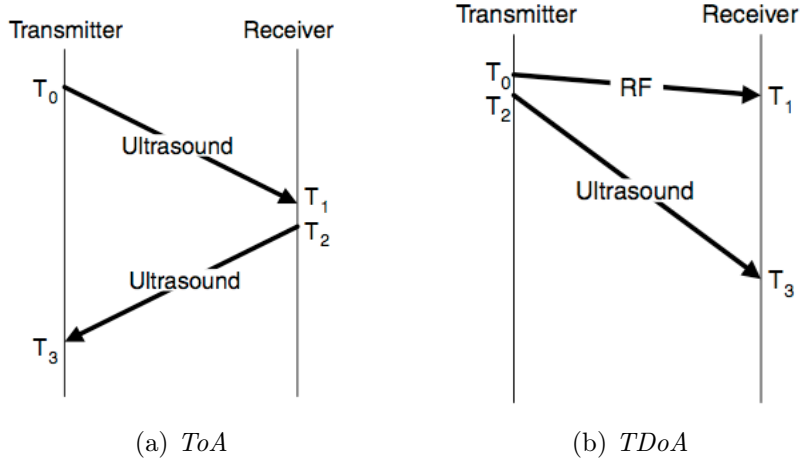


Figure 2.1: Time of Arrival and Time Difference of Arrival

Time Difference of Arrival (TDoA) Another distance estimation technique is the Time Difference of Arrival (TDoA). Although similar to the ToA scheme, this method uses two signals that travel at different speeds, such as electromagnetic waves (RF) and ultrasound. Figure 2.1(b) shows how TDoA works; transmission in one direction is sufficient. At time T_0 the transmitter sends an RF signal, followed by an ultrasound signal at time T_2 . The receiver can then determine its distance to the transmitter by computing following equation

$$((T_3 - T_1) - (T_2 - T_0)) * \left(\frac{V_{RF} * V_{US}}{V_{RF} - V_{US}} \right) \quad (2.1)$$

where V_{RF} and V_{US} are the traveling speeds of RF and ultrasound signals, respectively. For TDoA, in addition to errors caused by processing time, the receiver must also know the precise value of $(T_2 - T_0)$ in order to determine the distance.

Received Signal Strength (RSS) Besides using the signal traveling time, this ranging technique exploit the property of radio signal to degrade while traveling in a space. Because signals traveling in a space typically reduce in strength with respect to the distance that they travel, the Received Signal Strength (RSS) can be measured at the receiver's side.

RSS is an interesting a largely investigated techniques since no additional hardware is necessary and distance estimates can even be derived from com-

munication that is taking place anyway.

Assuming that the transmission power P_{tx} , the path loss model, and the path loss coefficient α are known, a mathematical propagation model [5] can be derived and the receiver can use the received signal strength P_{rcvd} to solve for the distance d in a path loss equation like the following:

$$P_{rcvd} = c \frac{P_{tx}}{d^\alpha} \Leftrightarrow d = \sqrt[\alpha]{\frac{c P_{tx}}{P_{rcvd}}} \quad (2.2)$$

The path loss exponent α usually ranges from 2 to 4 and it is heavily influenced by the specific deployment environment (e.g. by walls, furniture, and so on) or by other simultaneous transmissions. Therefore, RSS values are not constant but can oscillate, even when sender and receiver do not move. Therefore, several radio propagation models for indoor environments have been studied. For a multi-wall path loss model, see [6, 5]. For a general assessment of the behaviour of RSS see [7].

Angle of Arrival (AoA) The Angle of Arrival (AoA) approach is another commonly used method for positioning [8]. Instead of providing information about distance among nodes, such approach requires an antenna array or an array of ultrasound receivers, which can determine the angle and orientation of signals.

Pattern Matching Pattern Matching (also known as the fingerprinting) is a technique that uses the signature of sensor data to draw conclusions about the location of the observer or the position of objects in a scene. Instead of estimating the distance between a beacon and a device, this approach tries to compare the received signal pattern against the training patterns previously stored in a database and determine the likelihood that the device is currently located in a certain position. Typically, Pattern Matching techniques involve two phases:

- *Training Phase.* Given a known localization infrastructure, the purpose of the training phase is to collect signals from all the reference nodes at each training location. The received signal strengths are recorded. Each entry in the database has the format: $[(x, y), (rss_1, rss_2, \dots, rss_n)]$, where (x, y) is the coordinate of the training location, and $(rss_1, rss_2, \dots, rss_n)$, are the signal strengths received at the training location.

- *Real-time Phase.* With a well-trained positioning model, one can estimate a position of a mobile node given the signal strengths collected by the device from all possible base stations. The positioning model may identify a number of locations, each associated with a probability.

A clear disadvantage of this approach is that performing the training phase for complex environments can be really cumbersome.

Dead Reckoning Dead Reckoning (DR) is the process of estimating the current position of an object based upon a previously determined position, and advancing that position based upon known or estimated speeds over elapsed time, and course. The term “Dead Reckoning” has been established as an abbreviation for *deduced reckoning* and is sometimes also referred to as *inertial navigation*.

The main idea of dead reckoning is shown in Figure 2.2. Let (x_0, y_0) denote the known position of an object with regard to a reference 2D coordinate system, α the angle (or direction) of motion, and L the distance covered in a certain amount of time. Applying the dead reckoning process, the estimated position (x_1, y_1) is shown in the following Equation.

$$x_1 = x_0 + L \cos \alpha, \quad y_1 = y_0 + L \sin \alpha \quad (2.3)$$

If the distance L is not known, it can be obtained from the velocity v of the target and the travel time Δt , as shown in the following Equation.

$$L = v\Delta t \quad (2.4)$$

Distance, direction, and speed of motion can be essentially obtained in two ways: (i) by inferring them from the previous position, or using sensors on the object.

The most used sensors for dead reckoning are:

- **accelerometers**, for measuring the acceleration and velocity of the object;
- **gyroscopes**, for computing the direction of the motion;
- **inclinometers**, for obtaining inclination and direction of the motion;
- **magnetometers**, for measuring the Earth’s magnetic field and using it as a three dimensional compass to get an absolute attitude.
- **pressure sensors**, for obtaining the altitude in respect to the sea level and using it to recalibrate the other sensors.

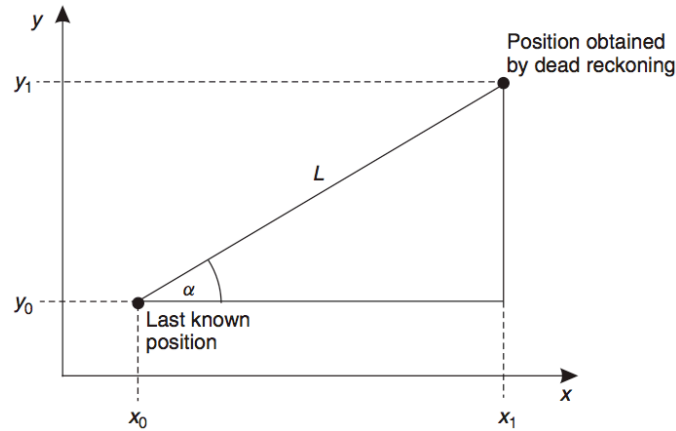


Figure 2.2: Dead Reckoning

A detailed introduction to these sensor technologies can be obtained from [9]. The accuracy of dead reckoning always degrades with the increase of time and with error accumulation. For further information and an evaluation of dead reckoning see Section 2.2.

2.1.2 Localization

The concept of *location* is not limited to the geographic representation of physical location with sets of latitude, longitude, and altitude; it is also applicable to symbolic location in a non-geographic sense such as location in time or in a virtual information space such as a data structure or the graph of a network. Therefore, it is very difficult to find a single algorithm that can function and satisfy all the localization needs in every situations.

A large number of localization approaches were proposed. However, almost all of them rely on some sort of localization infrastructure. The typical performance metrics used to compare different algorithms are:

- **accuracy.** The degree to which the random variation is centred on the true value (how close is an estimated position to the real position?);
- **precision.** It is related to the dispersion of the position estimation error (how often is a given accuracy achieved, for repeated position determinations?).

Evidently, accuracy and precision values only make sense when considered together, for an evaluation of localization systems in terms of accuracy and

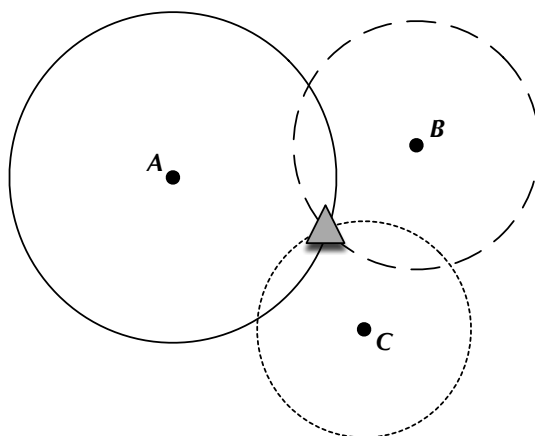


Figure 2.3: Trilateration method with an ideal situation

localization see 2.2. Other performance indexes are the robustness to errors of the algorithm (e.g. range measurement errors), the coverage, the localization infrastructure costs, the global energy consumption and the percentage of nodes with estimated position.

Furthermore, localization can be described as an optimization problem that minimizes the error of the solution for a given set of constraints [10], as follows.

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}_i)^2 \quad (2.5)$$

where n is number of measurements, x_i is the measured range, and \bar{x}_i is the corresponding range computed from the localized positions. σ then tells us how well the given solution fits the constraints.

Therefore, if the unknown node is in the range of a sufficient number of anchor nodes, its initial location can be found using several methods, the most used being *trilateration* (triangulation).

Trilateration is a well-known technique in which the positioning system has a number of beacons at known locations. These beacons can transmit signals so that other devices can determine their distances to the beacons based on the strength of the signals received. In an ideal situation, if a device can hear at least three beacons, its location can be estimated as follows. Figure 2.3 shows how trilateration works; A, B, and C are beacons with known locations. From their signals, a device (triangle) can be located using

the intersection of the three estimated-location circles.

However, as mentioned earlier, distance estimation always contains errors that will, in turn, lead to location errors. In practice, the three circles do not intersect in a common point. In this case, further optimization techniques (see equation 2.5) are needed to estimate the correct position.

For further details on WSN-specific localization algorithms see 3.3.

2.1.3 Tracking

Tracking algorithms use previous location and ranging data to track the movement of mobile objects. Until now, we considered the problem of localizing an object without taking into account any previous position knowledge. Given the current and previous estimated positions, it is possible to use movement models for improving the estimation of the tracked object expected position.

The most used techniques found in the literature are the *Bayesian Filters* [11]. *Kalman filters* and *Particle filters* are the most studied implementations of Bayesian Filters:

- **Kalman Filters.** The Kalman Filter (KF) [12] is an efficient recursive filter that estimates the state of a linear dynamic system from a series of noisy measurements. Opportunely modeling the motion of the mobile object, KFs can be used as post-processing filters for removing the effects of Gaussian measurement errors and therefore for better predicting the mobile object location.
- **Particle Filters.** The Particle Filters (PF) [13] discretize the probability distribution function (PDF) of the position of the mobile object instead of controlling the whole position space. They do not approximate the probabilities using only Gaussians, like KFs do. Besides the possible positions of the mobile object are represented as a set of particles, each of which has an associated weight. The distribution in space of these particles represents the most probable positions.

Tracking is the problem of finding trajectories of possibly moving targets. It is a complex problem, involving multiple sub-problems. For further details on the tracking techniques used see Chapter 5.

2.2 Evaluation of Localization and Tracking Systems

The precision of positioning systems varies from a few meters in 2D to a few tens of centimeters in 3D (a very fine mesh of ground stations is necessary in order to obtain the very best precision).

The range of available technologies varies from GPS and Galileo (satellite based, precision to a few meters, mainly outdoor-only), IEEE 802.11- and IEEE 802.15.4-based (can use an existing infrastructures, precision to a few meters, indoor and outdoor), dedicated RF systems (require a dedicated infrastructure, meter-precision), UWB-based (require a fine-meshed dedicated infrastructure, precision to a few centimeters, indoor and outdoor), Bluetooth and RFID-based (require dedicated beacons, no real localization but rather coarse proximity).

Rather than listing the existing localization and tracking systems, this subsection details the localization-system testbed I deployed. For a comprehensive survey of the most used localization systems see [14].

The testing environment is composed of two rooms divided by a large metal cabinet and it is shown in Figure 2.4 (a). Several system have been deployed in this environment. In particular, I evaluated the systems described in the following list. The data-gathering tests consist in placing the mobile tag of every system in a marked point and reading the measurements. This was repeated several times and in different points for all the systems.

- **Ubisense** The Ubisense [15] system requires a long and detailed deployment and calibration process. The localization infrastructure is composed of seven Ubisense Sensor placed as shown in Figure 2.4 (b).
- **WiFi-Based** Three different WiFi-based localization systems (AeroScout [16], Ekahau [17], Ubiquicom [18]) have been compared. They use three WiFi access points deployed as shown in Figure 2.4 (c).
- **WSN-Based** Fifteen nodes that use the Chipcon (now Texas Instruments) CC2431 Localization Engine [19] have been deployed as shown in Figure 2.4 (d). Moreover the Cricket [20] system has been tested with an infrastructure of 15 Cricket nodes deployed as shown in Figure 2.4 (e).

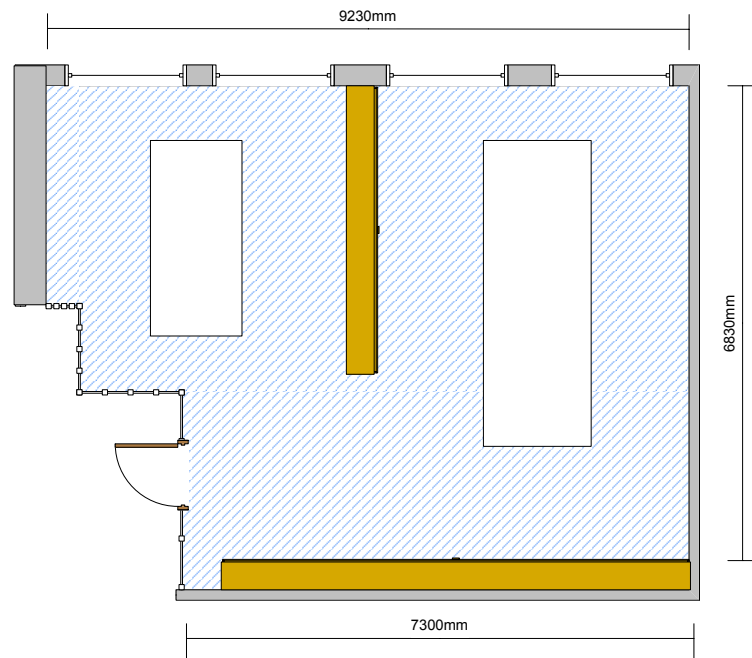
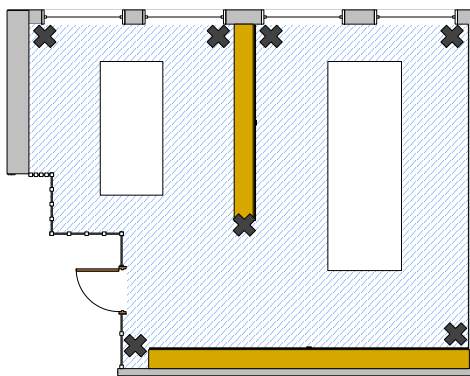
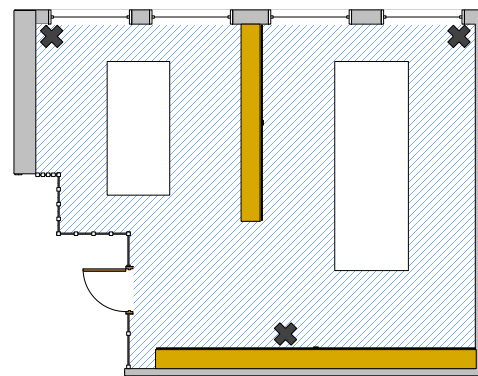
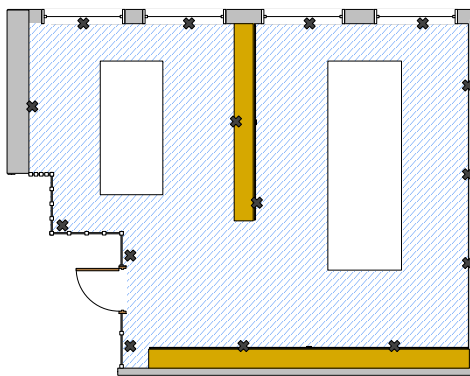
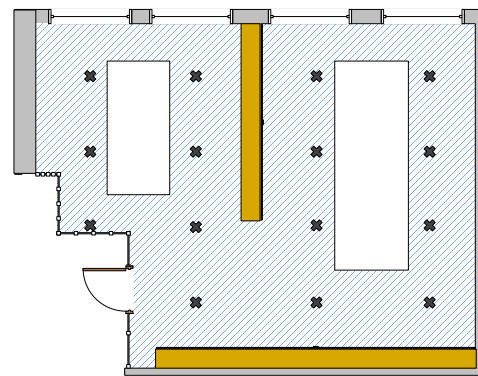
(a) *Map and the size of the Testing Environment*(b) *Positions of the Ubisense Sensors*(c) *Positions of the WiFi Access Points*(d) *Positions of the WSN Nodes*(e) *Positions of the Cricket Nodes*

Figure 2.4: Evaluation of Localization Systems. Testing Environment.

Table 2.1: Average Errors

Mean errors (meters)				
	10m	20m	30m	50m
GyroDRM	1.01m	1.68m	2.46m	3.12m
DRC	0.91m	1.38m	1.67m	2.21m

Moreover, I evaluated two inertial units: the Honeywell *GyroDRM* [21] and the Vectronix *Dead Reckoning Compass* (DRC) [22]. The GyroDRM is a gyro-aided dead-reckoning module for personnel on foot that includes a GPS receiver, an altimeter and a single axis gyro. The DRC uses accelerometers and magnetometers to deliver a continuous 2D position. I implemented several tests and found a very similar behavior to the experiments described in [23]. A good description of problems and performance of dead-reckoning systems can be found in a paper [24] and in a more detailed report [23] by NIST.

The experiments have been carried on in a university building and consist of pedestrians following a specific track at an even pace. The longest stretch between turns is 120 meters long while the shortest is 30 meters long. The path has been chosen to mimic some of the typical routes in our buildings. From the starting point on the lower left of the graph the pedestrians walk around the loop and then return to the starting point. The graph in Figure 2.5 (a) shows the tracks recorded by the dead reckoning devices averaged over a number of passes and without any recalibration.

The precision is not sufficient for a reasonable navigation: in some cases the error is greater than 10 meters. In particular, the error grows when the route has several turns or when the user has an irregular walking style. This happens because the tested DR devices are not more than advanced step counters. Moreover, the errors cancel out when the pedestrians return to the starting point, this is not a characteristic of the DR devices and should be considered an artifact of our experiments since it is probably due to the characteristics of the building. Therefore, DR must be aided by some form of recalibration. When used outdoor it is only natural to use GPS to recalibrate (as a matter fact one should look at it from the opposite point of view: DR is used to complement GPS when the GPS signal is not good enough). Indoor one must use some other recalibration techniques.

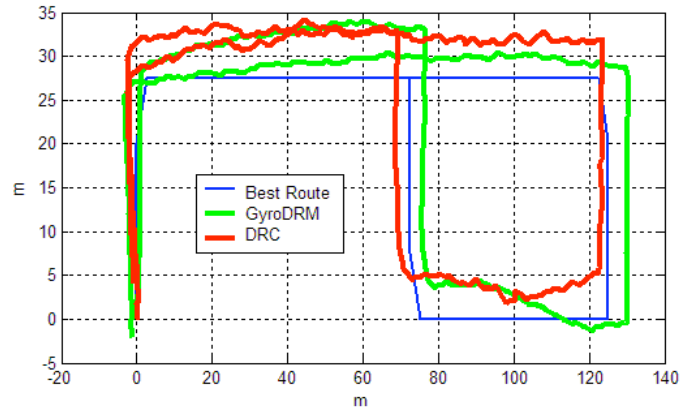
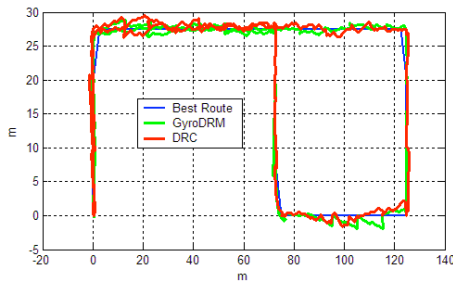
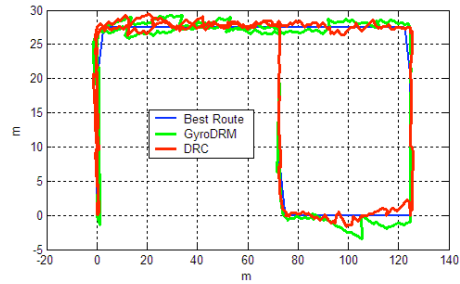
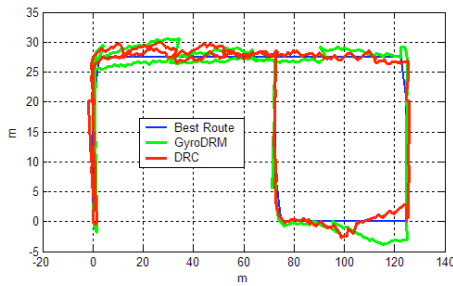
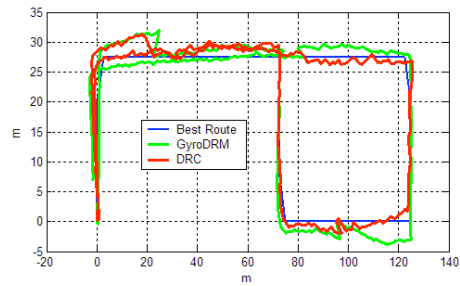
(a) *GyroDRM and DRC tracks without recalibration*(b) *10 meter recalibration*(c) *20 meter recalibration*(d) *30 meter recalibration*(e) *50 meter recalibration*

Figure 2.5: Experiments of Indoor Dead Reckoning with GyroDRM and DRC.

The key parameter is how often to normalize the DR output in order to offset the growing error of a DR. Figure 2.5 (b-e) shows a number of experiments where the DR's have been recalibrated every 10, 20, 30 and 50 meters. The quality of the result depends on the length of a track and it is apparent that sometimes, but not always, recalibration every 10 meters is necessary. The averages of the errors computed over the whole route are shown in Table 2.1.

These data represent the performance of specific hardware costing today on the order of 1,000 euros. One can use much cheaper hardware sensors for inertial navigation and more investigation must be able to characterize DR systems that have a better cost-performance ratio.

2.2.1 Evaluation Results and Further Considerations

The results of the evaluation have been summarized in Table 2.2.

The average localization performance of the Ubisense system is very interesting: an accuracy of 25 cm in 3D is attained for the 95% of the measurements whereas an accuracy of 14 cm is attained 50% of the times. Unfortunately, the infrastructure and deployment costs of such a system are very high and therefore its use is limited to a few controlled applications that require high accuracy.

Moreover, accuracy and precision of WiFi-based systems are quite low in respect to the infrastructural costs. They have been used in several situation but they are not sufficiently precise for many situation.

WSN-based system are clearly a good starting approach for tackling the localization and tracking problem. However, these techniques cannot be used alone because of the lack of precision. Therefore, I propose to integrate the WSN-based approach with dead reckoning techniques.

As detailed in the following chapters, the combined use of efficient data-gathering algorithms and advanced tracking frameworks can be the key for a substantial improvement of localization and tracking systems.

Table 2.2: Evaluation of Localization Systems

Category	Accuracy	2D/3D	Precision	User Cost	Infrastructure Cost	Limitations
Ubsense	25 cm	3D	high	low	high	cost of the infrastructure
WiFi-based	3-5 m	2D	low	low/medium	medium	precision, cost of the infrastructure
WSN-based	1-3 m	2D	low	low	low	precision
Ultrasounds TDOA	1-3 m	3D	medium	low	medium	cost of the infrastructure
Dead Reckoning	5-10 m	2D	high	low (future)	zero	precision

2.3 Chapter Summary

This chapter examined the general principles underlying the design of location systems. This chapter also described various indoor and outdoor location systems, related work on node localization, and systems for obtaining orientation information.

Moreover, this chapter has compared the performance of all the research and commercially-available tracking system outlining their problems and relative weaknesses and strengths.

Chapter 3

Large Scale Wireless Sensor Networks

Wireless Sensor Networks (WSN's) [25] consist of nodes that are capable of interacting with the environment by sensing and controlling physical parameters. Each node usually includes: processing capability, multiple types of memory, wireless transceivers, power sources and accommodate various sensors and actuators. They normally use packet radio communication to exchange data. These networks are typically used to collect data for long periods of time. After being deployed, the nodes have often self-organization capabilities and can operate without any assistance.

The common understanding of WSN probably originates from its initial research context, supported among others by numerous projects in the United States (e.g., SmartDust [26]). The vision was then of a kind of self-organized, homogeneous, tiny, multi-hop, and resource-constrained set of devices.

Specific scenarios for WSN's include habitat monitoring, industrial control, embedded sensing, medical data collection, building automation, fire detection, traffic monitoring, etc.

Currently, WSNs are beginning to be deployed at an accelerated pace. It is reasonable to consider that in 10-15 years the world will be covered with large-scale wireless sensor networks accessible via Internet. This can be thought of the Internet becoming a physical network. Therefore, we can envisage scenarios in which these networks will be pervasively available in several environments: networks of thousands or even tens of thousands nodes

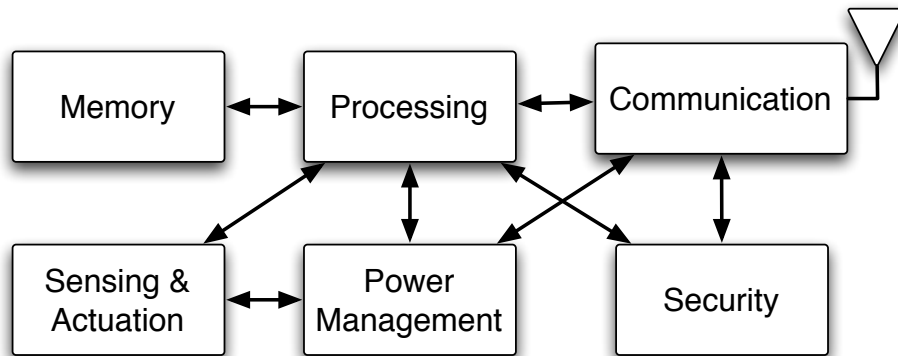


Figure 3.1: Main Components of WSN Nodes

are forthcoming!

This chapter presents an overview of some of the key areas and research in WSNs and it is organized as follows. Section 3.1 describes the most common node architectures. Section 3.2 gives a brief overview of the most used communication protocols for WSN. In conclusion, Section 3.3 describes several approaches used to tackle the problem of nodes localization in a WSN.

3.1 WSN Node Architectures

Every WSN node is a complex device that includes several dedicated hardware and software components. Several nodes can then cooperate in order to build either simple (e.g., star, tree) or very complex mesh networks, which can potentially be bridged to other types of information networks such as Internet. In the remainder of this section I will give a brief description of these components whereas network architectures will be described in Chapter 4.

A standard WSN node suitable for every application does not exist. WSN nodes have to meet the requirements that came from a given application. For example, an application may require a huge number of cheap nodes with low computational power, whereas another may need less but more powerful nodes and so forth. Moreover, some applications often need to adapt their behavior to particular situations. Therefore, each of the components of a WSN node has to cooperate for finding the right way to meet the specific application requirements.

WSN nodes have several components, the following description details

the most interesting ones whereas Figure 3.1 shows their interactions.

Processing. The processing is mostly done using *microcontrollers*. These controllers are the core of a wireless sensor node. They collect and process data from the sensors, control the actuators and manage ingoing and outgoing communications.

Sensing and Actuation. A *sensor* is a physical device that can observe (sense) environmental properties of its surroundings and convert them into electrical signals. Conversely, an *actuator* typically accepts an electrical signal and converts it into a physical action. Sensors and actuators belong to the broader family of transducers.

Communication. *Transceivers* are the components that turn nodes into a network devices capable of sending and receiving information over a wireless channel. They usually implement completely the PHY layers along with some critical part of the MAC layers. Transceivers appropriate for WSNs are available from many manufacturers (see Section 4.2 for further details). The communication device must be carefully chosen because its power consumption is often far from negligible. Currently, standard transceivers typically require 40-50 mA in both transmit and receive mode (at 3V, 0 dBm). Future generations shall get down to 10-15 mA [27].

Power Management. The power supply is a crucial component for WSN nodes. There are essentially two approaches: *energy management*, used to preserve as much as possible the available energy and *energy scavenging*, used for collecting energy from external power sources. The energy management problem is widely treated in the literature, however energy-aware data-gathering algorithms will be described in Chapter 4 and their power consumption detailed in Section 4.4. Several energy scavenging sources exist [28], however the main ones up to now are: kinetic (from vibrations to electricity through piezoelectric, electrostatic, or magnetic induction effects), thermal (using heat flux), and electromagnetic (from light or from RF radiation).

3.2 Communication Protocols

IEEE 802.15.4 In the past few years, hundreds of Physical (PHY) and medium-access (MAC) layer approaches have been proposed (for a

complete reference see [29]), however the most important one is the IEEE 802.15.4 standard [30].

The IEEE 802.15.4 working group defined the PHY and MAC layers for low-complexity, low-power consumption, low bit-rate Wireless Personal Area Networks (WPAN) connectivity.

The physical layer is mostly concerned with modulation and demodulation of digital data; this task is carried out by so-called transceivers. In sensor networks, the challenge is to find modulation schemes and transceiver architectures that are simple, low cost, but still robust enough to provide the desired service.

A global RF standard for WSN's is fundamental because it accelerates the technology evolution by identifying leading directions, and it could yield benefits such as the lowering of design costs and the interoperability at the communication level.

Network Layers The most used WSN network layers are: ZigBee [31] and 6LoWPAN [32].

ZigBee is a specification for a suite of high level communication protocols based on the IEEE 802.15.4 standard.

6LoWPAN is a specification that introduces an adaptation layer for enabling efficient IPv6 communication over IEEE 802.15.4 links. Extending IP to low-power, wireless personal area networks has recently been considered a practical way to implement the so called *Internet of Things* [2].

Topology control and Routing The routing algorithms for wireless sensor networks can be broadly divided into three types: flat-based routing, hierarchical-based routing, and adaptive-based routing. Flat-based routing assumes that all sensor nodes perform the same role. On the contrary, nodes in hierarchical-based routing have different roles in the network, which can be static or dynamic. Finally, Adaptive routing changes nodes' behavior according to different application and network conditions such as available energy resources. These routing protocols can be further classified into multipath-based, query-based, or negotiation-based routing techniques depending on the protocol operation.

In the remainder of this dissertation I will consider only hierarchical approaches. Topology control consists in deliberately restricting the set of neighboring nodes of a given node in order to minimize network

traffic and maximize node power. An example of hierarchical topology control is described in [33].

Approaches to hierarchical routing like those described in Leach [34], Teen [35] and APTeen [36] are particularly interesting for their approach to node organization in clusters.

3.3 Localization in Wireless Sensor Networks

Node Localization, the problem of determining the geographical location of a node in a network is a complex and fundamental problem for WSNs.

Although existing positioning systems can provide precise location information, their adoption without modifications has often several problems in respect to the constraints of a WSN node. For example, the deployment of a GPS receiver in every sensor node is expensive and unaffordable for most WSN applications.

Dardari et al. [37] propose the following classification of the most known and used localization algorithms.

- **Range-based.** Based on techniques that measure low-level spatial relations between nodes (distances, distance differences or angles). For further details see Section 2.1.1.
- **Range-free.** Based only on available connectivity information.
- **Anchor-based.** Anchor nodes are special nodes equipped with special positioning devices. Other nodes try to determine their own position relative to anchors. Once their positions have been estimated, regular nodes may in turn become anchors, etc. Anchor-based localization algorithms need another positioning system in place to provide initialization information.
- **Anchor-free.** Since none of the nodes knows its position, all collaborate with each other (usually only with their neighbors) in order to determine a relative map. Some further (post-)processing is needed in order to convert the relative map into an absolute position information otherwise only relative coordinates (virtual coordinates) can be found.

In extension to those described in Section 2.1.2, several interesting examples of localization are reported in the following list.

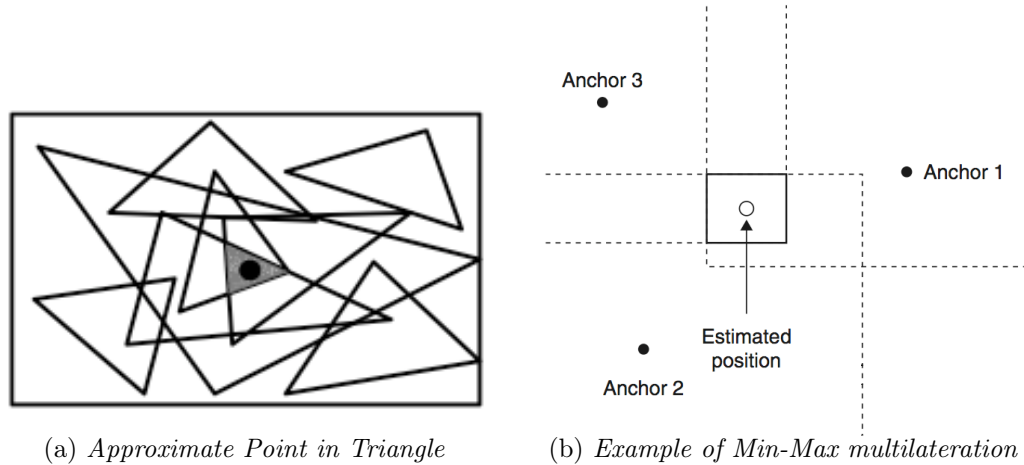


Figure 3.2: APIT and Min-Max algorithms

Overlapping Connectivity Bulusu et al. [38] describe an example of positioning system that operates without any numeric range measurements. Instead, it tries to use only the observation of connectivity to a set of anchors to determine a node position. The underlying assumption is that transmissions (of known and fixed transmission power) from an anchor can be received within a circular area of known radius.

Approximate Point in Triangle In [39], the authors suggest using this Approximate Point in Triangle (APIT) test for range-free localization. For any triplet of landmarks that a node can hear, if the node passes the APIT test (i.e., for each one of its neighbors at least some landmark sounds stronger) with respect to these landmarks, then the node is declared to be in the triangle defined by the landmarks. Thus the node can be localized to lie at the intersection of all landmark triangles that are known to contain it. See Figure 3.2 (a) for an example.

Min-Max Savvides et al. [40] proposed a simply localization algorithm that requires only low complexity sum and comparison operations, called *Min-Max*. As shown in Figure 3.2 (b), the key idea is to construct a bounding box starting from each known position (x_i, y_i) and distance measurement d_i . The estimated position is obtained as the centre of the intersection of these bounding boxes and moreover the final position is evaluated as the average of both corner coordinates.

Radio Interferometric Ranging Researchers at Vanderbilt University have

proposed and demonstrated a surprisingly simple and powerful method for ranging that produces centimeter ranging accuracy at ranges up to 160 meters [41, 42].

This technique, known as Radio Interferometric Ranging, exploits electromagnetic interference to obtain observations that are function of the locations of four nodes involved in the measurement. As such, it does not directly produce pairwise distance measurements, but rather a linear combination of four of the possible six pairwise distances. This fact renders localization algorithms that rely on pairwise distances unable to capitalize on this new technique.

The Radio Interferometric Ranging technique requires two sensors, a and b , transmitting simultaneously at slightly different frequencies, f_a and f_b . The combined signal exhibits a beat frequency, $|f_a - f_b|$, with a phase that is a function of receiver location. The difference of phase at two different receiving sensors c and d is a function of the coordinates of the two transmitters and two receivers.

Unfortunately, RF multipath effects distort the measurements and makes this approach difficult to use in indoor environments. However, the most remarkable feature of this technique is that it can be implemented with coarse time synchronization on inexpensive radios found on widely available sensor network devices.

Other Approaches Niculescu and Nath [8] proposed the *DV-Hop* and *DV-Distance* localization approaches based on network flooding, similar to the operation of a distance vector (DV) routing protocol.

Multi-hop lateration approaches have been proposed by Savvides et al. [43] and similarly by Savarese et al. [44].

3.4 Chapter Summary

This chapter has had two purposes: give an introduction to Wireless Sensor Networks per-se and outline the state-of-the-art of localization techniques using WSN's.

The first goal was rather daunting: there are literally tens of books, in average 500-pages long, that deal with this subject. The reason is that WSN's raise all the issues of *normal* networks and then some more.

Localization in WSN is also a much worked-on problem and presents new angles. For example the *range-free* and *anchor-free* algorithms described in this chapter are specific of this kind of networks.

Therefore, this chapter is the springboard for the thesis contributions that are all based on the existence of a pervasive WSN.

Chapter 4

The DiGS Data Gathering System

This chapter describes a **Data Gathering System** called **DiGS**.

After a general overview of the system in Section 4.1, Section 4.2 describes the DiGS hardware design and implementation. Section 4.3 describes the main software components of the DiGS system, and in particular it details the hierarchical network organization (division in zones or “digs”) from which the name DiGS arises. In conclusion, Section 4.4 describes the evaluation of different components of the DiGS system.

4.1 Introduction to DiGS

DiGS is a Data Gathering System that can be used for gathering localization information in order to track mobile nodes (or tags) and for acquiring environmental data in the surrounding of the tags.

Each node in the DiGS system is part of a Wireless Sensor Network and it is a small hardware system consisting of a wireless microcontroller with several different kinds of sensors.

The WSN used in DiGS is organized hierarchically. The area where the data gathering will be performed is divided into several zones each containing a cluster of nodes.

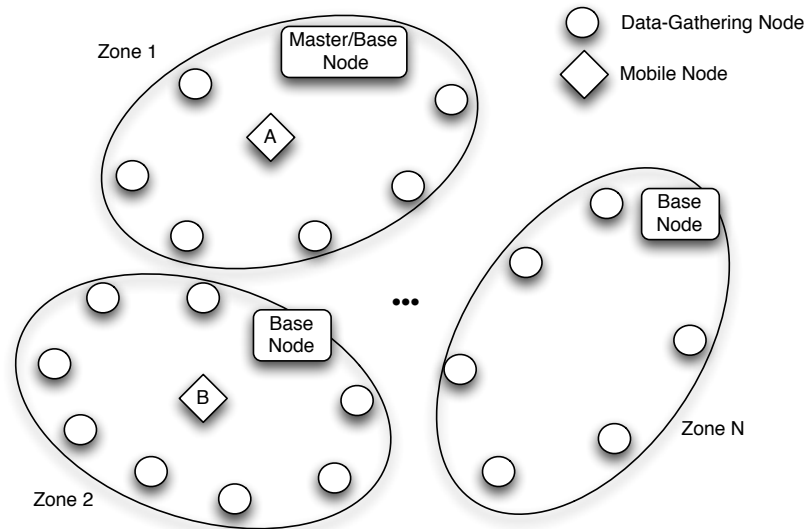


Figure 4.1: Hierarchical Network Organization: the Infrastructure nodes and the Mobile nodes

As shown in Figure 4.1, every zone contains at least:

- a base node, in a fixed and known location, mostly used for network coordination but also with its own sensing capabilities;
- several data-gathering nodes (also fixed) used for localization and for improving the accuracy of the data gathering;
- zero to many mobile nodes (the “tags”);

One of the base nodes is connected to a host computer and is called Master Node and behaves as a gateway towards the outside world.

Moreover, the DiGS system has been realized taking the following main requirements into consideration.

Energy consumption As most sensor nodes have a limited energy capacity, all protocols and algorithms must be energy-efficient and save as much energy as possible. Since the most energy is consumed during the wireless communication, the radio must be turned off most of the time. But also the transmission of data should be energy-efficient in order to

minimize the number of sent and received packets. DiGS deals with the energy-consumption problem trying to reduce the network traffic as much as possible using synchronization between nodes and appropriately scheduling the active time for every node.

Synchronization As far as synchronization is concerned, DiGS uses most of the ideas introduced by the Flooding Time Synchronization Protocol (FTSP) [45]. FTSP synchronizes (time-wise) of a sender node with multiple receivers using a single radio message time-stamped at both sender and receiver sides. It achieves a network-wide time synchronization with error in the micro-second range and scalability up to thousands of nodes. Moreover, a precise synchronization is needed in order to reduce the duty cycle of the nodes as much as possible.

Scalability A WSN is potentially composed of thousands of sensor nodes, densely deployed in an regional area. Protocols must thus scale well with the number of nodes. This is often achieved by using distributed and localized algorithms, creating clusters that let nodes mostly communicate only with their neighbours [36]. DiGS adopts the same approach adding a hierarchical organization of these clusters that improves the Cluster-based Hierarchical Time Synchronization protocol (CHTS) [46].

Adaptiveness Sensor nodes are prone to both mechanical and electrical failure. On the other hand, an high resiliency of the whole system is usually required, be it because of the importance of the task or simply because repair is difficult and costly. Therefore, in case of node failures, the network must be able to reorganize itself and if needed reconfigure its application tasks.

Self-organization The network must operate with the least possible need of manual configuration. For example, communication paths throughout the network should be established automatically. Also the cooperation between nodes and cluster formation must be organized in an unattended manner in order to achieve the global application task.

A final consideration regarding the localization-data gathering algorithms of DiGS. According to the number of mobile nodes tracked, the system can have two different approaches is necessary. The first approach, called DiGS-MB (see Section 4.3.2), can be used when a single mobile node is tracked by system. In this approach the tag broadcasts at regular intervals a localization

message to the fixed nodes. They extract the localization information (e.g. the RSSI parameter) and report the measurement to their base node.

The second approach, called DiGS-OB (see Section 4.3.3) deals with situations in which several mobile nodes need to be tracked. In the DiGS-OB approach the mobile nodes receive the localization information from the fixed nodes that broadcast this information following a precise schedule.

4.2 DiGS Data Gathering Devices

One of the contributions of this thesis is the design, the implementation and the prototyping of new WSN devices specifically conceived for people tracking and therefore equipped with heterogeneous sensors.

In particular, the fixed nodes include motion and range-finder sensors used to improve the accuracy of the RSSI-based localization and tracking. Moreover, the mobile node includes a complete six-deegrees-of-freedom (6DOF) Inertial Measurement Unit (IMU) that is used to track the user movements and to improve the behaviour of the target tracking algorithms (for further details see Chapter 5).

In the following sections, I will give more details about the microcontroller and the sensors used by the DiGS Data Gathering Devices.

4.2.1 The JN5139 Wireless Microcontroller

The WSN nodes of DiGS use the Jennic JN5139 IEEE 802.15.4 wireless microcontroller. It includes a RISC CPU, on-chip memory, a wireless transceiver, and an extensive range of peripherals (see Figure 4.2).

The CPU of the JN5139 is a 32-bit load and store RISC processor. It features a linear 32-bit logical address space with unified memory architecture, accessing both code and data in the same address space. Registers for peripheral units, such as the timers, UARTs and the baseband processor are also mapped into this space.

Moreover, the JN5139 includes 192 kBytes of ROM and 96 kBytes of RAM. The ROM contents include a bootloader, IEEE 802.15.4 MAC, a default interrupt vector table, an interrupt manager and several APIs for interfacing to the MAC and hardware peripherals. The RAM is used for both

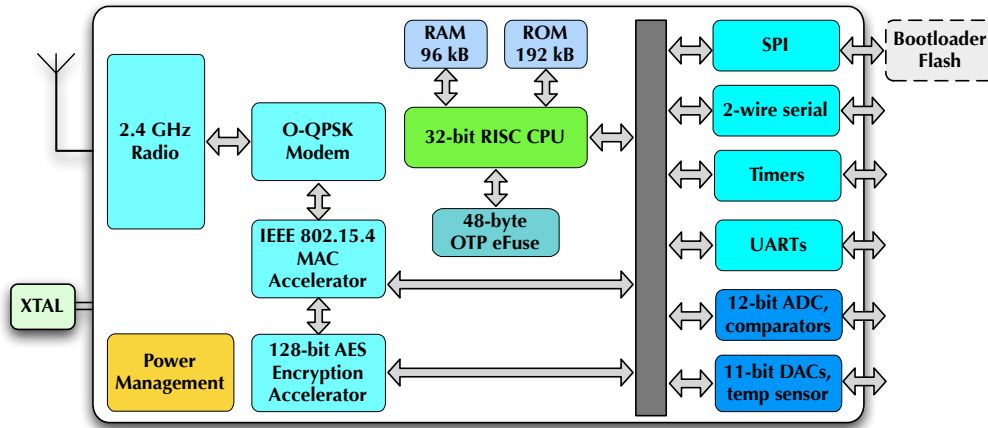


Figure 4.2: Jennic JN5139 Block Diagram.

code and data storage and is accessed by the CPU in a single clock cycle. The firmware is generally stored into an external memory connected to the SPI port, and it is loaded by the bootloader into RAM at reset time.

JN5139 has two system clocks. A 16MHz clock is used by the transceiver, processor, memory and peripherals and it is generated by a crystal-controlled 16MHz oscillator. The microcontroller uses a lower speed 32kHz clock during the start-up phase and the sleep periods. Due to manufacturing variations of its timing components, this oscillator runs nominally at $32\text{kHz} \pm 30\%$. For accurate wake-up from sleep a frequency calibration factor must be applied. The calibration factor is computed by software using the more accurate 16MHz clock. Further details can be found in section 4.4.

JN5139 has a IEEE802.15.4 (see section 3.2) standards-based wireless transceiver. It includes a 2.45GHz radio, a PHY controller, a baseband processor, an O-QPSK modem and a security coprocessor and PHY controller. Further details about radio propagation can be found in section 4.4.

In order to maximize the battery life and carefully control its power consumption, the microcontroller provides the following three operating modes:

- **Active Processing Mode.** All of the peripherals are available to the application. However there is still the option to doze the CPU but keep the rest of the chip active; this is particularly useful for radio transmit and receive operations, where the CPU operation is not directly required.

Table 4.1: JN5139 DC Current Consumption (VDD=2.2 to 3.6V, -40 to +85° C)

Mode	Action	Typical	MAX	Unit	Notes
Active Processing Mode	CPU processing	7.5	12.2	mA	SPI, DIOs enabled, code executing in RAM
	Radio transmit	38	50	mA	CPU in software doze, radio transmitting
	Radio receive	37	48	mA	CPU in software doze, radio in receive mode
Sleep mode	I/O wakeup	0.1		μ A	Waiting on I/O event
	RC Oscillator Timer	1.2		μ A	As above but waiting on timer event
Deep sleep mode		60	250	nA	Waiting on chip RESET or I/O event

- **Sleep Mode.** In the Sleep mode, several internal chip functions are turned off. However the state of DIO pins is retained allowing the reaction to different wake-up events (from timers, comparators or DIOs). For example, any (used as input) DIO pin can generate a wake-up event. This means that external devices communicating over the UART can wake-up a sleeping device by asserting the RTS signal.
- **Deep Sleep Mode.** The lowest power consumption is achieved using the Deep Sleep mode. Almost every peripheral is turned off and several functions including the 32kHz oscillator are stopped. This mode can be exited by a power down, a hardware reset, or a particular DIO reset event.

Table 4.1 gives an overview on the typical and maximum current consumptions for the three operating modes.

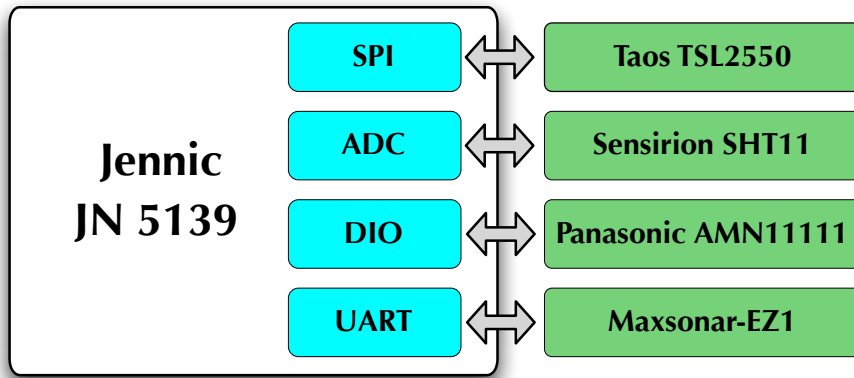


Figure 4.3: DiGS Fixed Node hardware schema.

4.2.2 DiGS Fixed nodes

The DiGS Fixed node is composed of two boards. The base board has the JN5139 microcontroller, the rechargeable battery and its recharge circuit. Moreover, it provides two expansion connectors, the first one used for generic input/output and for connecting external sensors whereas the other is used for connectioning the sensor board.

As previously said, the fixed nodes of the DiGS system have several sensing capabilities. The sensor board provides motion, range finder, humidity, temperature and ambient-light sensors.

Figure 4.3 shows the logical hardware schema and lists sensors and their connection to the JN5139.

Figure 4.4 and Figure 4.5 show the placement of the main components in the hardware implementation.

Figure 4.6 shows the final version of the fixed nodes in their enclosure.

Details about the sensors of the sensor board will follow.

Passive Infrared (PIR) Motion Sensor

The Panasonic AMN11111 [47] is a motion sensor that can detect changes in infrared radiation. This is particularly useful to catch the movements of a person (or object) that has a different temperature with respect to the

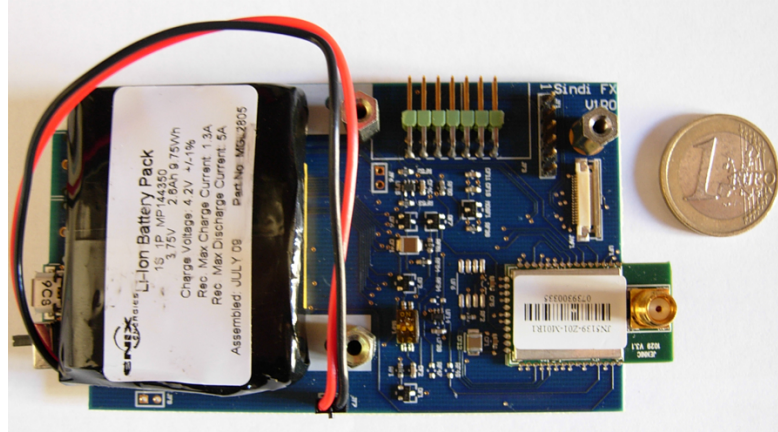


Figure 4.4: Base Board of the DiGS fixed nodes.

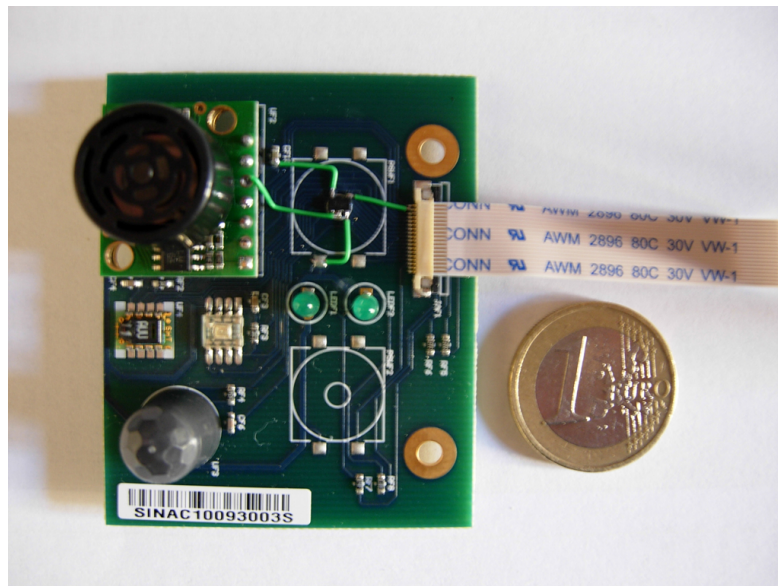


Figure 4.5: Sensor Board of the DiGS fixed nodes.



Figure 4.6: DiGS Fixed Node with Enclosure.

surroundings.

The sensor has a typical rated detection distance of 5 m when the difference in temperature between the person and the background is more than 4°C. The detectable movement speed ranges from 0.5 to 1.5 m/s. The horizontal detection range (in degrees) is 100°, whereas the vertical one is 82°.

The current consumption is very low: the typical maximum current consumption is 60 μ A in Standby mode, whereas it is 100 μ A when in Detecting mode.

Sonar Range Finder

The Maxbotix LV-MaxSonar-EZ1 [48] sonar range finder is a detection and ranging sensor that can provide very short to long-range measurements.

The LV-MaxSonar-EZ1 detects objects from 0 cm to 6.45 m and provides sonar range information from 15.24 cm out to 6.45 m with 2.54 cm resolution. Objects from 0 cm to 15.24 cm range as 15.24 cm.

The sonar range finder operates at 42KHz and readings can occur up to every 50ms (20-Hz rate). The typical current consumption is 2mA. Therefore

the sensor must be frequently put into sleep mode in order to minimize the device power consumption. Executing the ranging process only when a movement is detected by the PIR motion sensor is a good method to achieve a proper compromise between accuracy and consumption.

The LV-MaxSonar-EZ1 sensor needs to perform a calibration cycle in order to compute ranges correctly. Unfortunately, environmental variations in the surroundings of the sensor determine loss of calibration data randomly causing false close readings. Therefore, if the temperature, humidity, or power supply voltage change during operation, the sensor may require recalibration to reacquire the calibration data.

In order to avoid as much as possible these false readings, the humidity and temperature sensor (see the following paragraph) is used to monitor environmental variations and start the recalibration cycle for the LV-MaxSonar-EZ1 sensor when it is needed.

Moreover, false readings happens also when multiple ultrasonic rangefinders operate at the same time in the same environment. Therefore, they must be synchronized and follow a precise schedule for limit any possible interference.

In indoor environments, range finder sensors could be also used in passages or openings in order to acquire higher level information about the user movement (e.g. opening or closing doors and windows).

Humidity and Temperature

The Sensirion SHT11 [49] is a surface-mountable relative-humidity and temperature sensor. Humidity is measured using a capacitive sensor element whereas temperature with a band-gap one. Besides the humidity and temperature sensors the chip contains A/D converter, OTP memory, an amplifier and a digital interface. Table 4.2 shows performance test results for the SHT11.

Ambient Light

The TAOS TSL2550 [50] is a digital-output light sensor that provides light measurements over a dynamic range with a response similar to that of the human eye. The use of two photodiodes, the first (channel 0) sensitive to both visible and infrared light, whereas the second (channel 1) sensitive only

Table 4.2: Performance of Sensirion SHT11 (25°C and 3.3V)

Relative Humidity					
Parameter	Condition	min	typ	max	Units
Operating Range		0		100	%RH
Resolution		0.4	0.05	0.05	%RH
Accuracy	Typical		±3.0		%RH
Temperature					
Parameter	Condition	min	typ	max	Units
Operating Range		-40.0		123.8	°C
Resolution		0.04	0.01	0.01	°C
Accuracy	Typical		±0.4		°C

to infrared light makes it possible to approximate the human eye response with the commonly used unit of illuminance, Lux.

An integrating ADC converts the photodiode currents to channel 0 and channel 1 digital outputs. Channel 1 digital output is used to compensate for the effect of the infrared component of light on the channel 0 digital output. The ADC digital outputs from the two channels are used in Equation 4.1 to obtain a value measured in Lux.

$$Light\ Level\ (lux) = (Ch_0 - Ch_1) \times 0.39 \times e^{(-0.181R^2)} \quad (4.1)$$

where $R = Ch_1\ ADC\ Reading / (Ch_0\ ADC\ Reading - Ch_1\ ADC\ Reading)$. Formula 4.1 is obtained by optical testing the sensor with fluorescent and incandescent light sources.

4.2.3 DiGS Mobile Nodes

I realized two different prototypes of the DiGS Fixed nodes. The first one used an off-the-shelf inertial unit produced by Sparkfun, whereas the second one includes all the inertial sensors directly on-board.

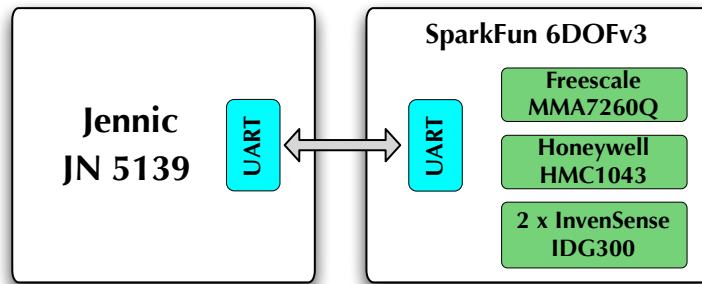


Figure 4.7: Hardware schema of the DiGS Mobile Node first prototype.

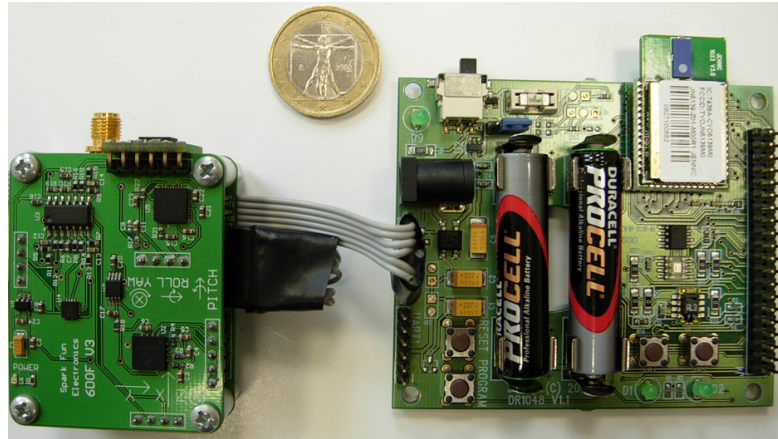


Figure 4.8: The First Prototype of the DiGS Mobile Node.

First Prototype of DiGS Mobile Node

Figure 4.7 shows the logical hardware schema of the simple connection existing between JN5139 and the Sparkfun unit whereas Figure 4.8 shows the main components and their layout.

Sparkfun 6DOFv3 IMU The Sparfun 6DOFv3 [51] is a complete inertial unit that uses the following sensors:

- Freescale MMA7260Q triple-axis accelerometer;
- 2 × InvenSense IDG300 gyroscopes;
- Honeywell HMC1043 dual-axis magnetometer.

All sensor readings are available in either ASCII or binary format over a serial connection and could be easily read through any terminal emulator.

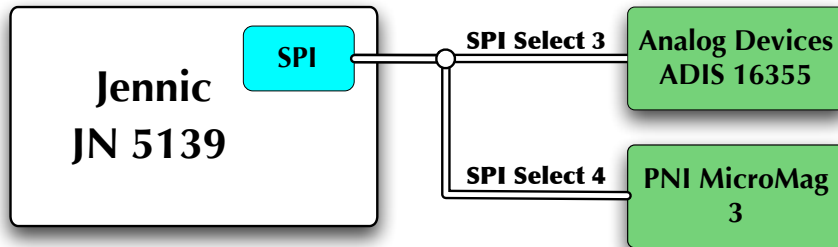


Figure 4.9: Hardware schema of the DiGS Mobile Node current implementation.

Moreover, all sensors are internally temperature compensated.

Freescale MMA7260Q The Freescale MMA7260Q [52] is a low cost capacitive MEMS accelerometer that consists of two capacitive sensing cells and a signal conditioning ASIC contained in a single integrated circuit package. Depending on two logic input pins, the device internal gain can be changed allowing it to function with a 1.5g, 2g, 4g, or 6g sensitivity. The device current consumption is $500 \mu\text{A}$ in active mode and $3 \mu\text{A}$ in sleep mode.

InvenSense IDG300 The InvenSense IDG300 [53] is an integrated dual-axis gyroscope. It uses two sensor elements with vibrating dual-mass bulk silicon configurations that sense the rate of rotation about the X- and Y-axis. This technique is called “in-plane sensing”. It incorporates X- and Y-axis low-pass filters and an EEPROM for on-chip factory calibration of the sensor. The device current consumption is $9.5 \mu\text{A}$ in active mode.

Honeywell HMC1043 The Honeywell HMC1043 [54] is a three-axis surface mount sensor array designed for low field magnetic sensing. This sensor measures the direction and the magnitude of Earth’s magnetic fields, from tens of micro-gauss to 6 gauss. The device current consumption is $10 \mu\text{A}$ in active mode.

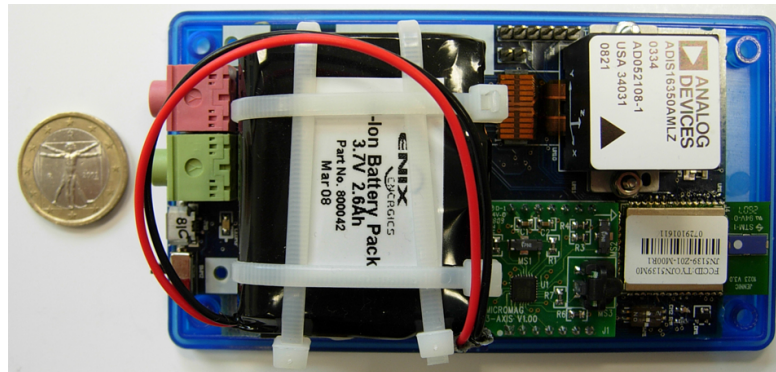


Figure 4.10: DiGS final prototype main components and layout.

Final DiGS Mobile Node

Analog Devices (AD) produces an highly integrated IMU with 3-axis accelerometers and 3-axis gyroscopes in a very low-dimension package. The availability of such a product lead me to use slightly different sensors for the final prototype of the DiGS mobile nodes. Moreover, AD will soon provide another IMU that also includes a 3-axis magnetometer in the same package.

Therefore, the final prototype of the DiGS Mobile Nodes uses:

- an **Analog Devices ADIS 16355**. The Analog Devices ADIS 16355 [55] is a complete triple-axis gyroscope and triple-axis accelerometer inertial sensing unit. Its dimensions are approximately $23\text{ mm} \times 23\text{ mm} \times 23\text{ mm}$, making it a very compact IMU. The device current consumption is 33 mA to 57 mA in active mode (depending on the reading speed) and $500\ \mu\text{A}$ in sleep mode.
- an **PNI MicroMag3**. The PNI MicroMag3 [56] is an integrated three-axis magnetic field sensing module that includes three magneto-inductive sensors and an ASIC controller on a single PCB, to provide a complete magnetic field sensing solution. The device current consumption is less than $500\ \mu\text{A}$ in active mode.

The idea is to substitute, when available, the AD ADIS 16355 with the newer and more complete AD's IMU, removing the PNI magnetometer.

Figure 4.9 shows the logical hardware schema of the final prototype of the DiGS Mobile node, whereas Figure 4.10 shows the main components and their layout.

4.3 DiGS Data Gathering Software

The software part of the DiGS data gathering system is based on a hierarchical organization of the network and on sensor data acquisition algorithms that are optimized for power consumption.

4.3.1 Hierarchical Network Organization

The area where the data gathering will be performed is divided into several zones (“*digs*”) that contain a cluster of nodes. As shown in Figure 4.1, every zone is controlled by one base node and contains several data-gathering nodes. All data-gathering nodes are within one hop of the base node of their zone. The mobile nodes can move from zone to zone without loss of connectivity.

Data-gathering nodes are typically battery-operated while base nodes are either connected to the power grid or use a more powerful battery. One of the base nodes (called master node) behaves as a gateway towards the outside world and, specifically, towards a server that runs the localization algorithm which is implemented with a particle filter (see Chapter 5 for further details).

The hierarchical network organization simplifies the network management. The master node manages the entire network using several network services that are part of the DiGS system: topology-control, hierarchical routing, localization, data aggregation, and communication. Therefore the DiGS includes well-known WSN algorithms [34, 57, 43].

The DiGS system requires a deployment and configuration step to build the clusters in the field. In this step the nodes are synchronized for the first time and receive the schedule that will let them communicate with minimal interference.

With the automatic creation of clusters, some base nodes could end up disconnected from each other. This cannot be avoided beforehand since the location of base nodes has to adhere to some physical constraints, e.g. the presence of a power socket. These situations have to be fixed by adding base nodes and clusters where necessary.

The algorithm FTSP [45] synchronizes two connected nodes by computing the difference between the sender and the receiver time and allowing the receiver node to estimate and compensate its clock skew. The algorithm is extended to a multi-hop network by starting from an arbitrary node and its neighbour and then propagating this pairwise synchronization to all the

other nodes.

Network synchronization limits the number of collisions and retransmissions while a tight schedule lowers the global power consumption by keeping nodes powered off when they are not needed. The synchronization between clusters is based on the exchange of timestamped messages between base nodes starting from the master node. Every base node is then responsible for the synchronization of all the broadcast nodes that are associated with it.

The schedule is built in a centralized way and is based on a neighbours' map created by each node with a gossiping (or controlled flooding) algorithm. In the initialization phase all the network nodes exchange information with their one-hop neighbours to build the neighbour map.

The master node then creates a "firing sequence" for the zones that guarantees that zones whose nodes might interfere with one another are never scheduled at the same time (see Section 4.3.3). Creating a schedule is very expensive both in time and power but it needs to be performed only when the network infrastructure is first deployed or changed. Neighbour maps are only recomputed when base nodes are added or removed.

The hierarchical organization and the presence of a fixed schedule simplifies the routing algorithm that is used to send the localization data to the master node. Messages are exchanged only when indicated in the schedule and according to the neighbourhood maps.

The DiGS system assumes that the location of both base and data-gathering nodes is known. This may make the physical deployment of large networks very complicated. In this case, the position of some of the base nodes can be computed using localization algorithms for WSN, e.g. collaborative multilateration [43] (see also 3.3). However, this lack of information will likely reduce the precision of the localization.

4.3.2 The DiGS-MB Algorithm

The DiGS-MB is a data gathering algorithm that can be used when a single mobile node is tracked by system. In this approach, the tag broadcasts at regular intervals a localization message to all the nodes that are within reach. They compute the localization information (e.g. the RSSI parameter) and report the measurement to their base node.

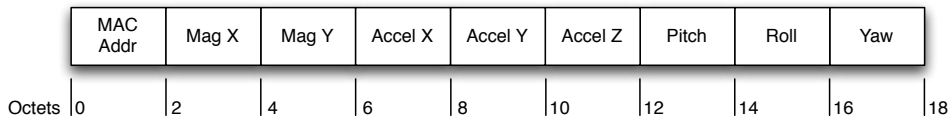


Figure 4.11: DiGS-OB Broadcast Message Payload.

The DiGS-MB data collection algorithm is executed in a three-phase cycle:

1. *Broadcast.* The mobile node sends a broadcast message with hop count equal to one. As shown in Figure 4.11, the broadcast message contains its MAC address and several inertial measurements. This message is received by the base and data-gathering nodes that are within reach. The size of the message is 18 bytes.
2. *Cluster Reporting.* Every data-gathering node computes the RSSI value of the received broadcast message and sends a report message with the localization information to its base node.
3. *Routing.* After having collected all the messages from its data-gathering nodes, the base node sends the localization information all the way to the master node connected to a server where a centralized localization algorithm is executed (for further details see Chapter 5). Only the base node forwards inertial data.

The cycle is normally executed at regular intervals to capture the movement of the mobile node. However, if the node does not move there is no need of broadcasting messages. The movement of the mobile node can be easily captured using its inertial sensors.

4.3.3 The DiGS-OB Algorithm

The DiGS-OB data gathering algorithm is power-efficient because the nodes are used in an opportunistic way and the zones where there are no tags remain active for brief periods. The algorithm lets the tag measure the RSSI between itself and all the broadcast nodes that it can reach.

The DiGS-OB data collection algorithm is executed in a four-phase cycle:

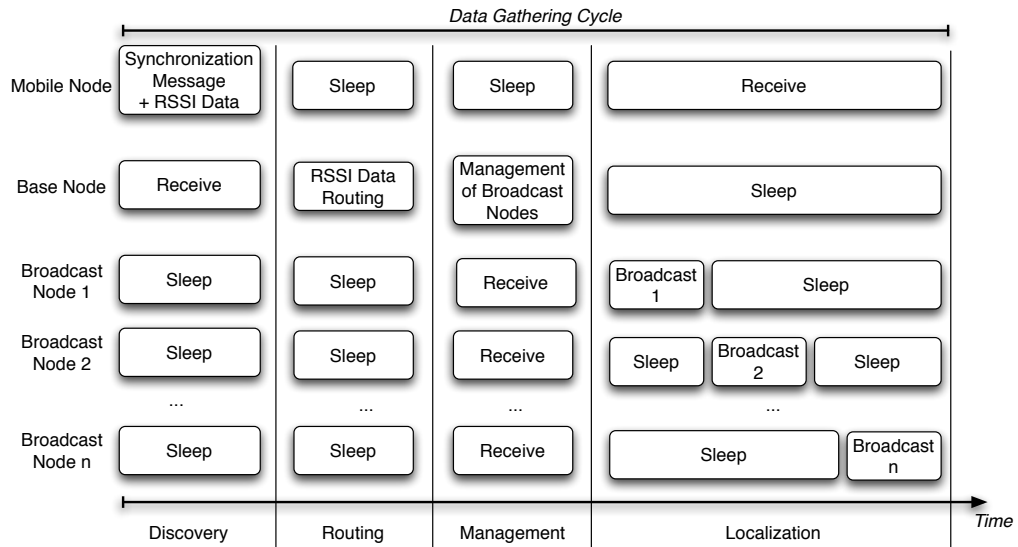


Figure 4.12: The DiGS-OB Scheduling Algorithm

1. *Discovery.* The mobile node sends a broadcast message with hop count equal to one. The message contains tag identification information and the RSSI measurements of previous messages from the broadcast nodes. This message is received by the base nodes that are within reach and is used to identify the zones that might contain the tag.
2. *Routing.* The base nodes that have received the mobile-node broadcast-message route the message content towards the master node.
3. *Management.* The base nodes send a broadcast message to the broadcast nodes asking for a localization step.
4. *Localization.* The broadcast nodes, one by one according to the agreed schedule, send localization messages to the tag. The base node can also participate in this step.

Figure 4.12 shows the four phases of the data collection algorithm for a single zone.

The cycle is executed at regular intervals according to the zone activation schedule agreed to during the configuration phase. This schedule and the fact that all nodes are synchronized avoid collisions across zones. Moreover, all the messages sent in a cycle are timestamped and can be used to keep the system synchronized.

The mobile node computes its position immediately by triangulating the distances from the base nodes that it can compute from the RSSI. The accuracy and the precision of this approach are limited by the RSSI propagation characteristics which are influenced by the environment in rather complex ways. On the other hand, this approach gives better results than a simple proximity computation and can be good enough when a tag is moving and we only need to know which zone it is in.

4.4 DiGS Evaluation

I performed a preliminary evaluation of the DiGS system testing its global behaviour in a simulated large-scale network using the Java-based simulator SIDnet-SWANS (Simulator and Integrated Development Platform for Sensor Networks Applications). SIDnet-SWANS allows the users to interact with various components of the network at different levels of granularity (e.g. node, region, routing structure). SIDnet is simply a graphical user interface wrapper for the well-known JiST-SWANS [58].

The simulation was very useful in order to test the behaviour of the algorithms using a high-level level language without getting lost in implementation issues. However, it is very difficult to evaluate several networking issues of the DiGS system using only a simulator.

Therefore, I realized a test environment to evaluate the algorithms for time synchronization and the in-network position computation. The test bed included 50 Jennic JN5139 nodes [59] and used the unslotted IEEE 802.15.4 MAC protocol. It was also used to estimate the average current consumption of every kind of DiGS node.

The DiGS system has been implemented over the most common PHY/-MAC protocol for WSN, IEEE 802.15.4. Moreover, I used the unslotted version of the standard that implements the carrier sense multiple access with collision avoidance (CSMA/CA) mechanism. IEEE 802.15.4 does a good job if contention is low but adds quite a bit of overhead if too many nodes are eager to transmit. The protocol has to cleverly exploit the scheduling in order to save the power of the network nodes.

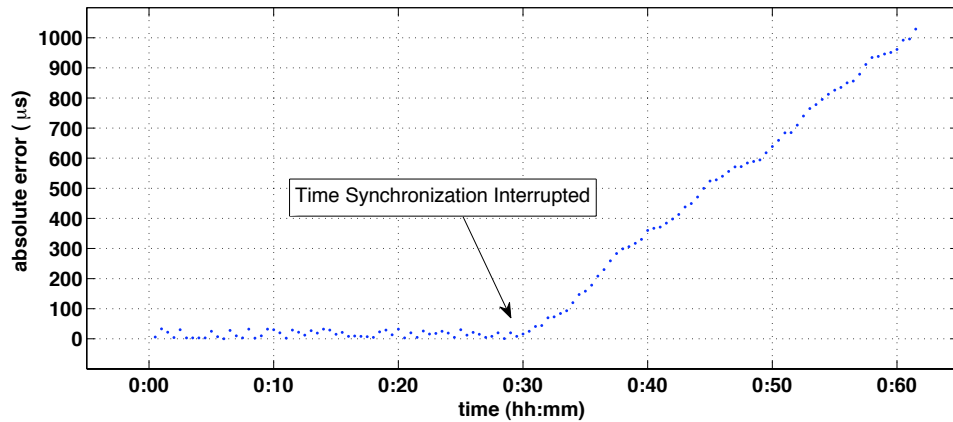


Figure 4.13: Time synchronization error between two nodes. The Time Synchronization was stopped after 30 minutes.

4.4.1 Time Synchronization

The DiGS system implements the time synchronization between network nodes using the FTSP protocol. Therefore, every message exchanged has a time-stamp that is used for synchronization.

I do not use the IEEE 802.15.4 MAC-level synchronization because it would prevent tight control of the sleep time of the nodes and raise power consumption.

An implementation and a preliminary evaluation of the FTSP synchronization algorithm has been performed by configuring a test environment composed of two base nodes.

The synchronization error of the Jennic nodes can be computed by taking into consideration that two separate oscillators are used to provide system clocks: a crystal-controlled 16MHz oscillator, using an external crystal, and an internal RC-based 32kHz oscillator.

The wake-up timers run at a nominal 32kHz but in practice they may run up to 30% faster or slower depending on temperature, supply voltage and manufacturing tolerance and drift up to $33.2\mu\text{s}$ per second.

The evaluation test consisted in sending synchronization messages between two base nodes every 15 seconds for 30 minutes. After 30 minutes we interrupted the synchronization mechanism to measure the time synchro-

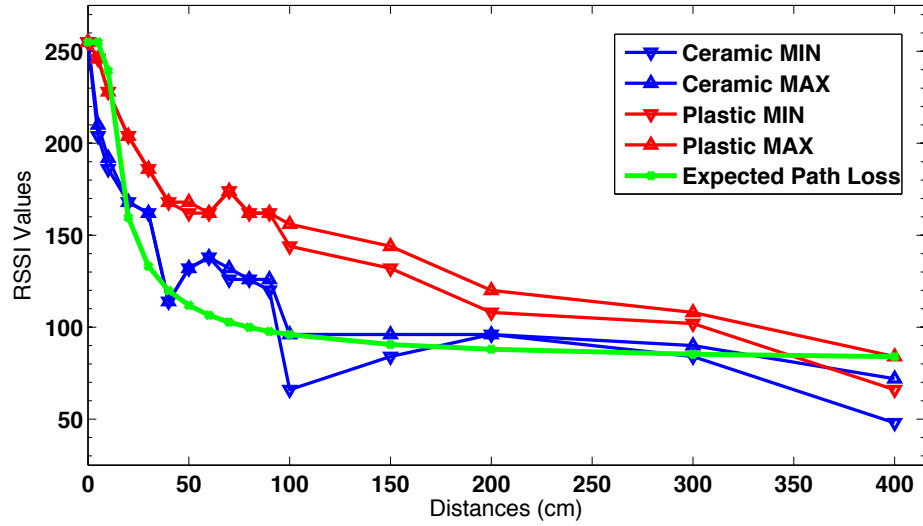


Figure 4.14: RSSI values and distance

nization error.

Figure 4.13 shows that our implementation of the FTSP protocol using Jennic nodes has a very similar behavior to what shown in [45].

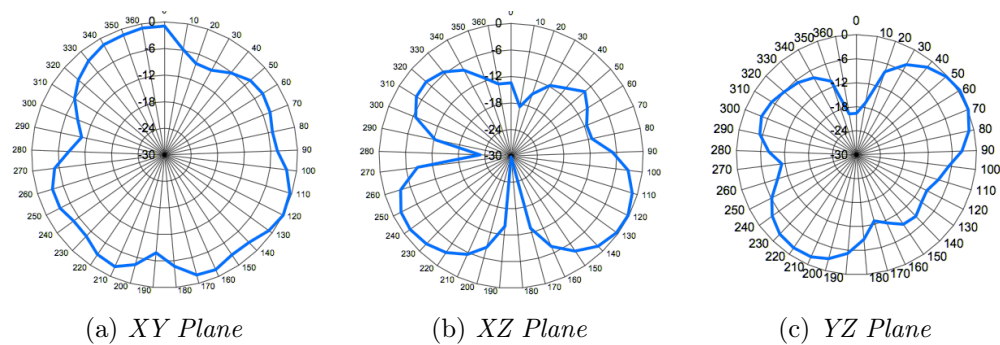


Figure 4.15: Radiation Pattern for JN5139 Ceramic Antennas.

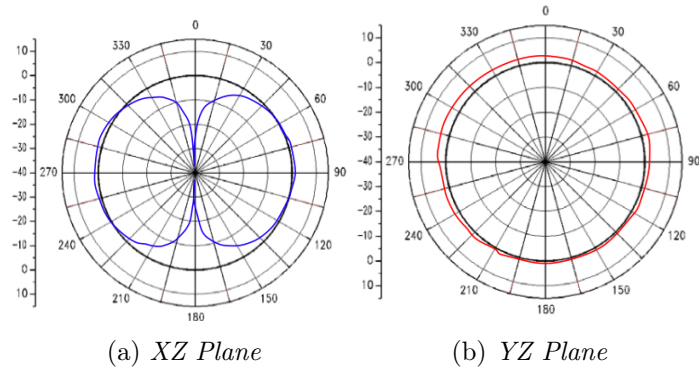


Figure 4.16: Radiation Pattern for JN5139 Plastic-External Antennas.

4.4.2 Evaluation of In-network Position Computation

The first step of the evaluation of in-network position-calculation algorithms is a correct characterization of the RSSI parameter for the Jennic JN5139. For a general assessment of the behaviour of RSSI see also [7].

Therefore, I recorded the RSSI values of many messages between two nodes of the network at various distances (10 cm, 20 cm, 30 cm, 40 cm, 50 cm, 60 cm, 70 cm, 80 cm, 90 cm, 100 cm, 150 cm, 200 cm, 300 cm, 400 cm). The nodes have two different types of antenna (ceramic and plastic) and different battery charge-levels (high-medium-low).

The test results are shown in Figure 4.14. This test shows that:

- the RSSI values are highly influenced by environmental conditions (such as location of the sensors in the room, presence of people, and so on);
- the RSSI values are substantially independent of battery charge-levels;
- the type of antenna considerably affects the RSSI values.

Starting from the last statement of the previous list, it is important to exactly characterize the radiation pattern for both ceramic and external-plastic antennas.

Figure 4.15 shows the radiation pattern for JN5139 ceramic antennas whereas Figure 4.16 shows that of external-plastic ones.

Taking the previous considerations into account, a preliminary evaluation of the DiGS system has been done configuring two similar test environments each composed of a master node, a mobile node, two base nodes and eight broadcast nodes assigned to two logical zones but with different antennas.

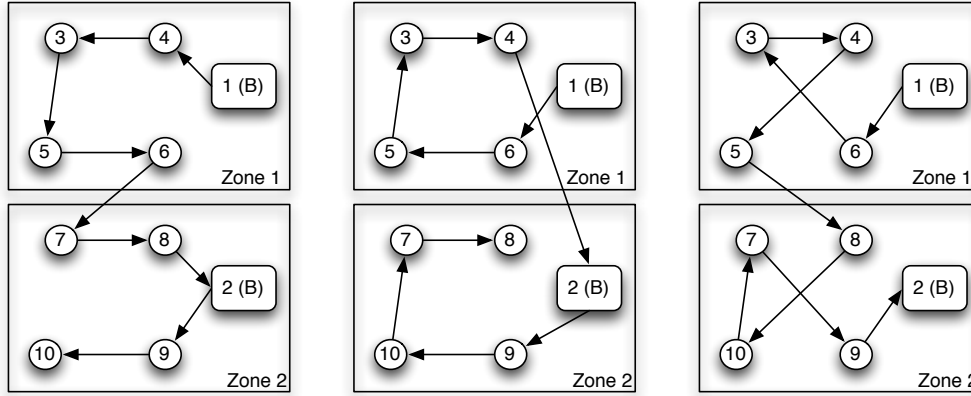
(a) *Clockwise*(b) *Counterclockwise*(c) *ZigZag*

Figure 4.17: Radiation Pattern for JN5139 Ceramic Antennas.

Table 4.3: Evaluation Results of the In-network Position Calculation

Route	Ceramic Antenna		External Antenna	
	Mean Accuracy	Acc. Variance	Mean Accuracy	Acc. Variance
Clockwise	212 cm	94 cm	183 cm	89 cm
Counterclockwise	224 cm	90 cm	193 cm	102 cm
Zig-zag	251 cm	118 cm	220 cm	93 cm

Figure 4.17 shows the three different routes of the experiment. In every route the starting point is the base node of the first zone, that in Figure 4.17 is labeled as 1 (B), where (B) means “Base Node”. We recorded for several times the RSSI data of the mobile nodes using the DiGS data-gathering algorithms taking as ground-truth the position given by a deployment of the Ubisense system (see 2.2). Table 4.3 summarizes the results of the evaluation. The techniques described in Chapter 5 significantly improve this results.

4.4.3 Current Consumption Calculation

In order to calculate the average current consumption of every kind of node of the WSN in the test bed we analyzed the behaviour of the unslotted CSMA/CA mechanism specified by the IEEE 802.15.4 protocol and computed the average current consumption of the Jennic nodes in every activity of the data-gathering algorithm showed in Figure 4.12. The algorithm has a *Total Cycle Time* that can be computed as follows:

$$\textit{Total Cycle Time} = \textit{Total Active Time} + \textit{Total Sleep Time} \quad (4.2)$$

The *Total Active Time* is the sum of the times required to complete every phase of the data gathering algorithm and can be computed as follows (for more details please see [60]).

In the unslotted version of IEEE 802.15.4 CSMA/CA, a Clear Channel Assessment (CCA) is performed to determine if the channel is currently in use before a data frame is transmitted. The CCA takes 8 symbol periods (0.128 ms) to complete.

Moreover, unslotted CSMA/CA is based on back-off periods, where one back-off period is equal to 20 symbol periods.

The behavior of unslotted CSMA/CA is determined primarily by two parameters, the Backoff Exponent (BE) and the Maximum Number of Backoffs (NB). The BE parameter is related to how many back-off periods a device will wait before attempting to assess the channel.

Therefore, supposing that the channel is free, the time taken to complete a single back-off period can be calculated as follows:

$$\textit{Backoff Period} (ms) = \frac{(2^{BE} - 1) \times 20}{62.5} \quad (4.3)$$

The maximum size of a data frame is 127 bytes, of which 6 bytes form the Physical layer header and 13 bytes form the MAC layer header (when using 16-bit source and destination addresses).

The total time taken to transmit a single data frame can be calculated as follows:

$$\textit{DataFrameTransmPeriod}(ms) = \frac{(\textit{HeaderSize} + \textit{PayloadSize}) \times 8}{250} \quad (4.4)$$

Table 4.4: Activities of the Base Node and Related Current Consumption and Time Duration

Phase	Activity	Current (mA)	Time (ms)
Discovery	Receive	37.00	5.000
Routing	CCA back-off	12.20	0.960
Routing	CCA	37.00	0.128
Routing	Transmission	38.00	4.064*
Management	CCA back-off	12.20	0.960
Management	CCA	37.00	0.128
Management	Transmission	38.00	0.800
* worst case			

During the back-off period, the application is running and the transceiver is off drawing a current of only 12.2 mA. During a CCA, the radio transceiver is on and therefore the current drawn is 37 mA.

Using the data of Tables 4.4 and 4.5 the *Total Active Time* is 12.04 ms for the base nodes and 6.792 ms for the broadcast ones.

The *Total Active Time* for mobile nodes depends on the number of zones and nodes that can be used for data gathering and can be computed using the data of Table 4.6.

Table 4.5: Activities of the Broadcast Node and Related Current Consumption and Time Duration

Phase	Activity	Current (mA)	Time (ms)
Management	Receive	37.00	5.000
Localization	CCA back-off	12.20	0.960
Localization	CCA	37.00	0.128
Localization	Broadcast Transmission	38.00	0.704

Table 4.6: Activities of the Mobile Node (Tag) and Related Current Consumption and Time Duration

Phase	Activity	Current (mA)	Time (ms)
Discovery	CCA back-off	12.20	0.960
Discovery	CCA	37.00	0.128
Discovery	RSSI Data Transmission	38.00	4.064*
Localization	Receive	37.00	number of zones × (number of broadc. nodes × broadc. time)*
* worst case			

The Total Sleep Time of Equation 4.2 can be computed as follows.

$$SleepTime_{TOT} (ms) = ZoneN \times (Lat + (NodeN - 1) \times Time_{Broadc}) \quad (4.5)$$

where Lat (*Latency*) is the sleep time during the Routing and the Management phases of the data gathering algorithm, $ZoneN$ is the number of Zones and $NodeN$ is the number of Nodes.

The average consumption for each activity can be determined using the following equation:

$$Node I_{average} = \sum_{\forall activity} I_{Activity} \times \frac{ActivityTime}{ActivityTime + TotalSleepTime} \quad (4.6)$$

where $I_{Activity}$ is the instantaneous current for a given activity.

Equation 4.6 gives the average total current drawn by a node.

For the mobile node if $TotalSleepTime$ is 1 second and the sleep current for the device is $6\mu A$ then the total current drawn is 16.492 mA. This consumption is quite high but it can be drastically limited using motion sensors (e.g. accelerometers) in order to activate the mobile node only when it is moving.

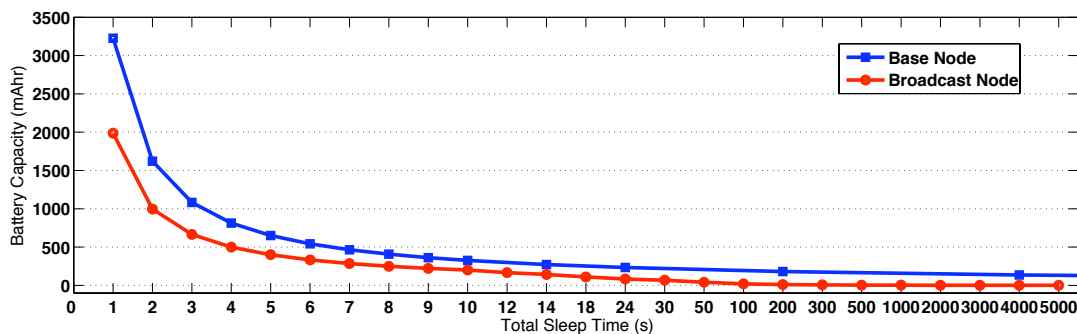


Figure 4.18: Effects of Sleep Period on Battery Life, Normalised to 1 Year

Similarly, using the data of Tables 4.4 and 4.5 and with the same *Total Sleep Time* of one second and sleep current of $6\mu A$ the average total current for the base node is $398.258\mu A$ and for the broadcast node is $226.666\mu A$.

Figure 4.18 shows a comparison between the effects of the sleep period on battery life for both base and broadcast nodes.

As you can see, even with short *TotalSleepTime* the DiGS-OB algorithm reaches a good compromise between tracking speed and batteries life.

4.5 Chapter Summary

This chapter has dealt with all is necessary to support the localization and tracking techniques described in the following chapters. First of all, the chapter has dealt in detail with the hardware structure and behavior of the single nodes, including the problem of designing and building them. Secondly, a number of algorithms and algorithm modifications have been devised in order to implement a network of energy-efficient nodes that can be extended to cover large areas, e.g. an entire building.

The chapter describes in detail the process necessary to characterize the signal strength (RSSI) behavior and estimates (since there has been neither the time nor the resources to test a large-scale (many 100's nodes) deployment the energy consumption benefits of the DIGS algorithms.

Chapter 5

Target Tracking

This chapter describes a **Bayesian Target Tracking Framework** called **BTTF**. BTTF exploits the widespread sensing and fusion technique of particle filters to implement a cutting-edge target-tracking framework.

After a general overview of the tracking problem and the approach used for tackling it in Section 5.1, Section 5.2 describes the BTTF design and implementation. In conclusion, Section 5.3 describes the evaluation of the BTTF framework in comparison with two well-known commercial tracking systems.

5.1 Introduction

As we have seen in Section 2.1.3, the *target tracking* algorithms use previous location and ranging data to track the movement of mobile objects. Target tracking systems keep the location information history and therefore are considered a dynamic methodology for the localization problem.

A contribution of this dissertation is the design and the implementation of a target tracking framework that exploits the following techniques:

- Multi-sensor data fusion;
- (Bayesian) estimation and filtering;
- Kinematic state estimation.

The complex problem of managing the data flow of different sensors is usually called *multi-sensor (data) fusion*. This problem involves the combination of sensory data from more than one source in order to generate more accurate, more complete or richer information. As a matter of fact, associating multiple independent sources of data can help reduce uncertainty after intelligently fusing and removing the ambiguity of the data.

The data sources for a fusion process do not necessarily originate from identical sensors: the sensor data can be indifferently produced from heterogeneous or homogeneous sensors.

According to [61], *estimation* is: “the construction of an estimate, based on observed data, of an unobservable underlying probability density function (PDF). The unobservable density function is thought of as the density according to which a large population is distributed; the data are usually thought of as a random sample from that population”. Moreover, *Recursive Bayesian estimation* is a general probabilistic approach for estimating an unknown PDF recursively over time using incoming measurements and a mathematical process model. When the observed PDF changes in time, this approach is named:

- **Filtering.** When we estimate the *current* value given past observations;
- **Smoothing.** When estimating *past* values given present and past measures;
- **Prediction.** When estimating a probable *future* value.

Filtering is desirable in many situations in engineering and embedded systems. The BTTF is a Bayesian filtering framework. A tracking system can be considered Bayesian if it has the following characteristics:

- **Prior Distribution.** A prior distribution on the state of the targets must exist. If the targets are moving, the prior distribution must include a probabilistic description of the motion characteristics of the targets. Usually the prior is given in terms of a stochastic process for the motion of the targets.
- **Likelihood Functions.** Sensor measurements and observations must be characterizable with likelihood functions.

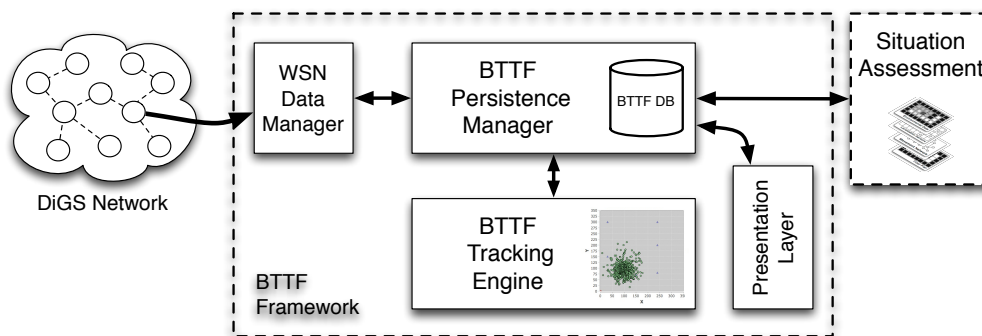


Figure 5.1: The Main Components and the Data Flow of the BTTF Framework.

- Posterior Distribution.** The basic output of a Bayesian tracker is a posterior probability density function (PDF) on the state of the target. The PDF at time t is computed by combining the motion updated prior distribution at time t with the likelihood function for the observations received at time t .

A *Particle Filter* (PF) [11, 13] is a non-parametric implementation of the Bayes filter. Its key idea consists of approximating the PDF by a finite number of parameters, representing this distribution by a set of random state samples drawn from it. The fact of being non-parametric is a clear advantage with respect to other approaches, in sense that PF's can represent a much wider distribution space than parametric (e.g. based on Gaussian distributions) techniques.

In addition, *Kinematics* is the branch of classical mechanics that describes the motion of objects without consideration of the causes leading to the motion. Dead reckoning techniques (see Section 2.1.1) make it possible the *Kinematic state estimation* of a moving target.

5.2 The BTTF Framework

In this section I will detail BTTF, a particle-filter based tracking framework. As shown in Figure 5.1, BTTF has the following main components:

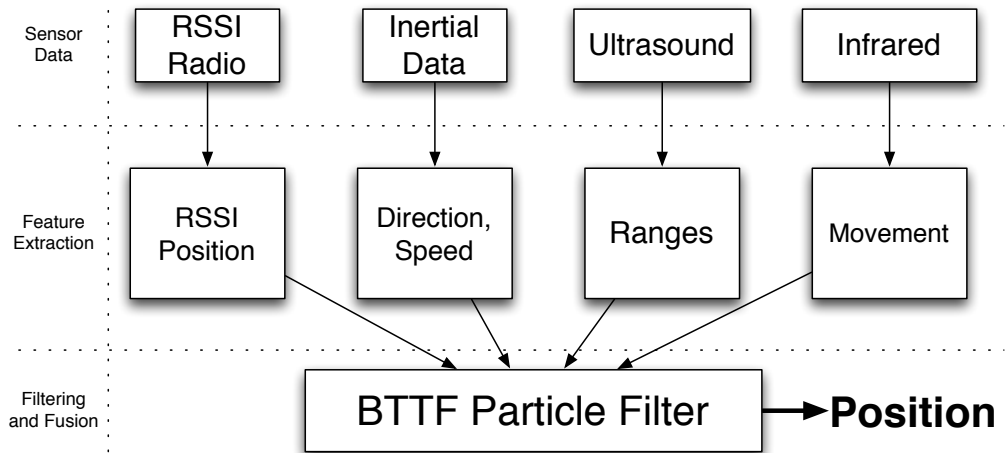


Figure 5.2: Detail of the BTTF Tracking Engine

- **WSN Data Manager.** An implementation of the DiGS system (see Chapter 4) is used for the data gathering. This component is responsible of the interaction of BTTF with the DiGS implementation;
- **BTTF Persistence Manager.** All the deployment configurations (sensor-node descriptions and positions, maps, etc.) and the sensor data are stored in a relational database. This component manages the persistence of these data;
- **BTTF Tracking Engine;**
- **Presentation Layer.** This component retrieves the computed positions and offers them to visualization clients.

In this context, the most interesting component is clearly the BTTF Tracking Engine. The remainder of this section will give further details about its implementation.

5.2.1 BTTF Tracking Engine

As detailed in Figure 5.2, the BTTF Tracking Engine is based on the following three layers:

- Sensor Data layer;
- Feature Extraction layer;
- Filtering and Fusion layer.

As we have seen in the introduction of this section, The BTTF Tracking Engine gathers data using an implementation of the DiGS System as localization infrastructure. Therefore, the sensor data sources used by the *Sensor Data* layer are substantially four: (i) Radio Signal Strengths, (ii) Inertial Data, (iii) Ultrasound, and (iv) Infrared. The sensors used by the DiGS System are detailed in Section 4.2.

The *Feature Extraction* layer is between the other two levels and it is used to transform sensor data into measurements meaningful for the filtering layer. Moreover it is used to optimize the sensor data by applying several heuristics for sorting, grouping or discarding sensor measurements. For example, RSSI measurements can be sorted in ascending order by their values in order to feed the PF with the best measurement first; furthermore, ultrasound ranges and infrared movements can be discarded if they are too noisy in respect to the previous measurements.

In addition, at this level the inertial data are filtered with a standard implementation of a Kalman filter in order to compute the direction and the speed of the moving target.

The BTTF Particle Filter of the *Filtering and Fusion* layer, aggregates the localization data and computes the target position using the algorithm described in section 5.2.2. Further details about the particle filter parameters will be given in sections 5.2.3 and 5.2.4.

5.2.2 The SIRS Algorithm

Many forms of particle filters are available in the literature. However, for the implementation of the BTTF Tracking Engine, I used the common Sequential Importance Sample with Resampling (SISR) algorithm. This implementation is quite common and it is a standard approach in several research communities (e.g. robotics, for robot tracking).

The SISR algorithm, differently from other filters (e.g. Kalman filter), is used to directly estimate the posterior probability density function (PDF) of the $\mathbf{x}(k)$ given the past observations $\mathbf{Z}(k)$ using the following equation [62]:

$$p(\mathbf{x}(k)|\mathbf{Z}(k)) \approx \sum_{i=1}^N w^i(k) \delta(\mathbf{x}(k) - \mathbf{x}^i(k)) \quad (5.1)$$

where $\mathbf{x}^i(k)$ is the i -th sampling point or particle of the posterior probability function and $w^i(k)$ is the weight of the particle.

In the current BTTF Tracking implementation, the state variable \mathbf{x}^i of every particle is a two dimensional hypothesis (x, y, speed, heading, weight) about the position of the object (the extension in three dimension is easy but not trivial).

The SISR algorithm comprises of the following steps [62]:

1. *Initialization*: Sampling N particles $\{\mathbf{x}^i(0), i = 1 \dots N\}$ according to the initial pdf $p(\mathbf{x}(0))$;
2. *Prediction Sampling*: For each particle $\mathbf{x}^i(k)$, get a new particle $\mathbf{x}^i(k+1)$ from the transition pdf $p(\mathbf{x}(k+1)|\mathbf{x}^i(k))$;
3. *Importance Sampling*: For each new particle $\mathbf{x}^i(k+1)$, calculate the weights $w^i(k+1) = p(\mathbf{z}(k+1)|\mathbf{x}^i(k+1))$.
4. *Normalization and Resampling*: The weights are normalized and finally resampled. In the resampling step, particles with low weight are deleted and particles with high weight are duplicated such that each particle has the same weight.

From the above description, we see that for particle filters, the transition density function $p(\mathbf{x}(k+1)|\mathbf{x}(k))$ and the update density function $p(\mathbf{z}(k+1)|\mathbf{x}(k+1))$ should be known to such an extent that enables to do prediction sampling and calculation of weights.

5.2.3 BTTF Particle Filter. Movement Model Details

The SISR approach, when used in the tracking context, involves the prediction of each particle movement using a motion model that implements its *Prediction Sampling* step. In order to do so, a human motion model that

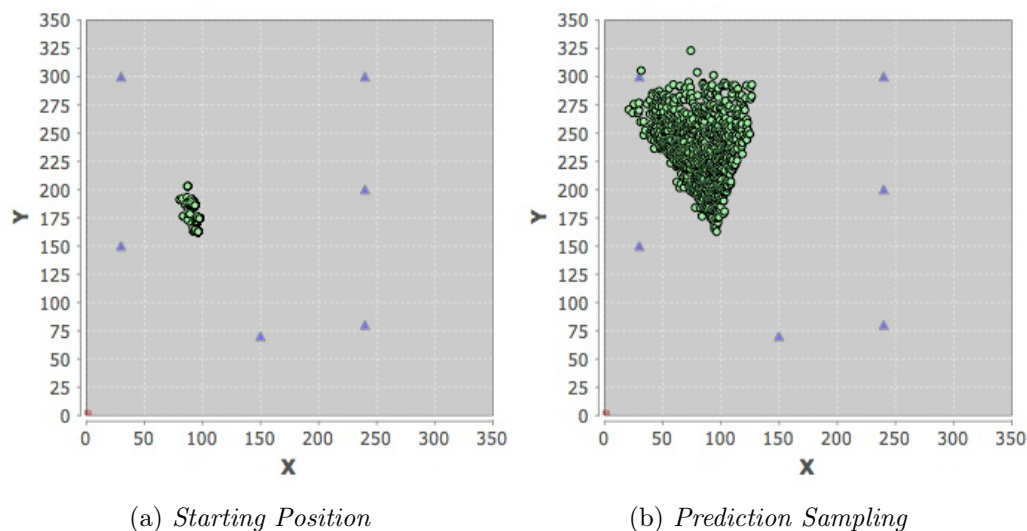


Figure 5.3: Inertial-Aided Motion Model

takes advantage of inertial data for computing speed and orientation of the target has been realized.

As we have seen, the state of every particle in the model includes speed and heading. During every prediction sampling step of the SISR algorithm, the BTTF Particle Filter updates, accordingly to the inertial data, the velocity and direction of every particle. In case of missing inertial data, an alternative approach consists of updating the state by adding or subtracting random values to speed and heading. After having updated their speed and heading every particle is moved in a new predicted position.

Figures 5.3 and 5.4 are useful to illustrate the two different approaches previously described.

Figure 5.3(a) shows the starting positions of the particles in an iteration of the SISR algorithm that exploits the inertial-aided motion model. The ground truth position is (95, 180). After the speed and heading update and the prediction sampling step, the resulting distribution of the particles is shown in Figure 5.3(b).

Figures 5.4(a) and 5.3(b) show the same thing for a generic motion model that does not use inertial data. In this second case, the particles can move in every direction (360°) with a random speed (0-10 m/s) and therefore, the accuracy of the tracking process is lower when compared to the first one.

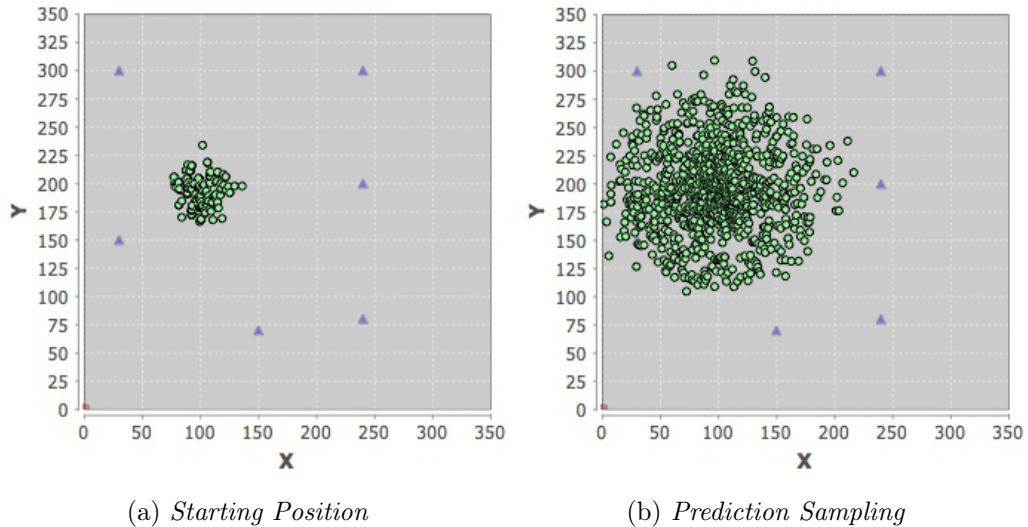


Figure 5.4: Generic Motion Model

5.2.4 BTTF Particle Filter. Sensor Models

In order to deal with heterogeneous sensor data, the BTTF particle filter must replicate the behaviour of every sensor (and its error sources) with an appropriate likelihood model. This likelihood can be modeled with the conditional probability $p(z|x)$ of a particular particle \mathbf{x}^i of being in a particular position (x, y, z) given the sensor measurement z .

Several experiments have been done in order to characterize the localization sensors behavior, as follows.

- **Movement Sensor** The passive infrared sensor does not provide distance information but only information about the presence of a person within a certain range. This sensor has a typical detection distance of 5 m. The horizontal detection range (in degrees) is 100° , whereas the vertical one is 82° . The output of this sensor is a boolean state “there is/there is not” a movement in the detection zone. Therefore, the characterization of its sensor model has been done considering that the orientation of the sensor is known and giving the maximum likelihood to particle positioned inside this zone.
- **RSSI Characterization** The RSSI behaviour of the Jennic JN5139

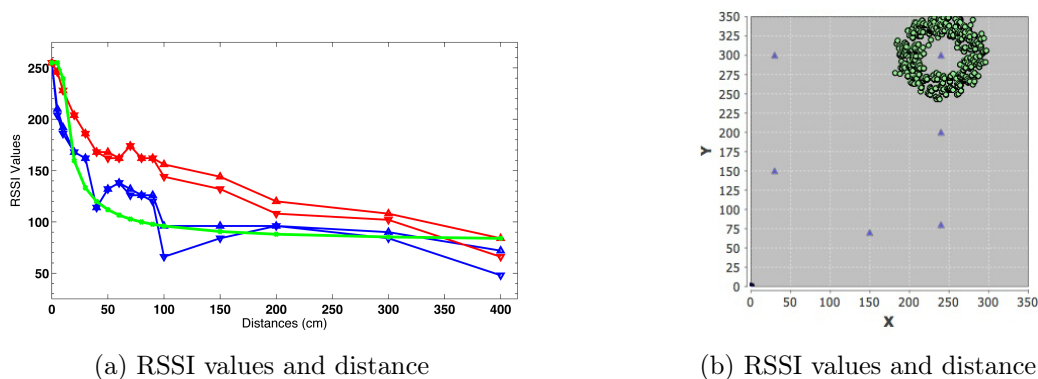


Figure 5.5: RSSI Sensor Model

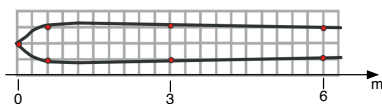


Figure 5.6: Rangefinder Sensor Model.

has been characterized with several tests for in-network position computation (see Figure 5.5 (a) and Section 4.4.2). This data source does not provide orientation information, therefore we must opportunely give high weights to the particles that are positioned in a circular region around the sensor, as shown in Figure 5.5 (b). The ring width represents the noise of this data source and it is modeled with a Gaussian distribution centered at the measured distance.

- **Rangefinder Sensor** As show in Figure 5.6, the rangefinder sensor detects objects from 0 cm to 6.5 m. The orientation of the sensor is known and therefore the particle are weighted according to a Gaussian distribution centered at the measured range.

5.3 BTTF Evaluation

This section presents the evaluation of the performance results of the BTTF framework.

To measure the localization performance in respect to a ground truth position in terms of accuracy (see Section 2.1.2), Broxton et al [63] proposed

a metric called Mean Absolute Error (MAE). MAE is very similar to the common Root Mean Square (RMS) and consist of computing the residual error between the estimated and actual node positions for every node in the network, sum them and average the result, as shown in the following Equation,

$$MAE = \frac{\sum_{i=1}^n \sqrt{(x_i - \bar{x}_i)^2 + (y_i - \bar{y}_i)^2 + (z_i - \bar{z}_i)^2}}{n} \quad (5.2)$$

where $(\bar{x}_i, \bar{y}_i, \bar{z}_i)$ are the coordinates of the target estimated position and (x_i, y_i, z_i) the ground truth ones.

The MAE metric was used to compare three configurations of the BTTF framework in three different environments, in comparison with two well-known commercial tracking systems. The tested BTTF configurations are the following:

- **RSSI only.** This configuration uses only the RSSI data source. Therefore, without inertial information available, it is forced to use a generic movement model;
- **RSSI + IMU.** This configuration also uses the only RSSI data source but unlike the previous one, the IMU inertial data are available and the inertial-aided motion model can be used;
- **RSSI + PIR + Range + IMU.** This configuration uses all the data sources available.

A DiGS localization infrastructure has been deployed in the following testing environments:

- *Office Environment U9.* This testing environment is composed of two rooms divided by a large metal cabinet and it is shown in Figure 5.7 (a);
- *Office Environment U14.* This setting is composed by three rooms and a part of a corridor. Figure 5.7 (c) shows CAD map of the area with the furniture;
- *Home Environment.* This testing environment is a real house composed by several rooms as shown in Figure 5.7 (b).

Two different evaluations of the BTTF framework have been completed. The first tests have been done using the Ubisense system as ground truth.

Ubisense developed and markets an UWB based localization system [15] that combines TDOA and angle of arrival (AoA) measurements to estimate tag location. The tags are equipped with 802.11b transceivers for data communication and proprietary UWB transmitters for localization.

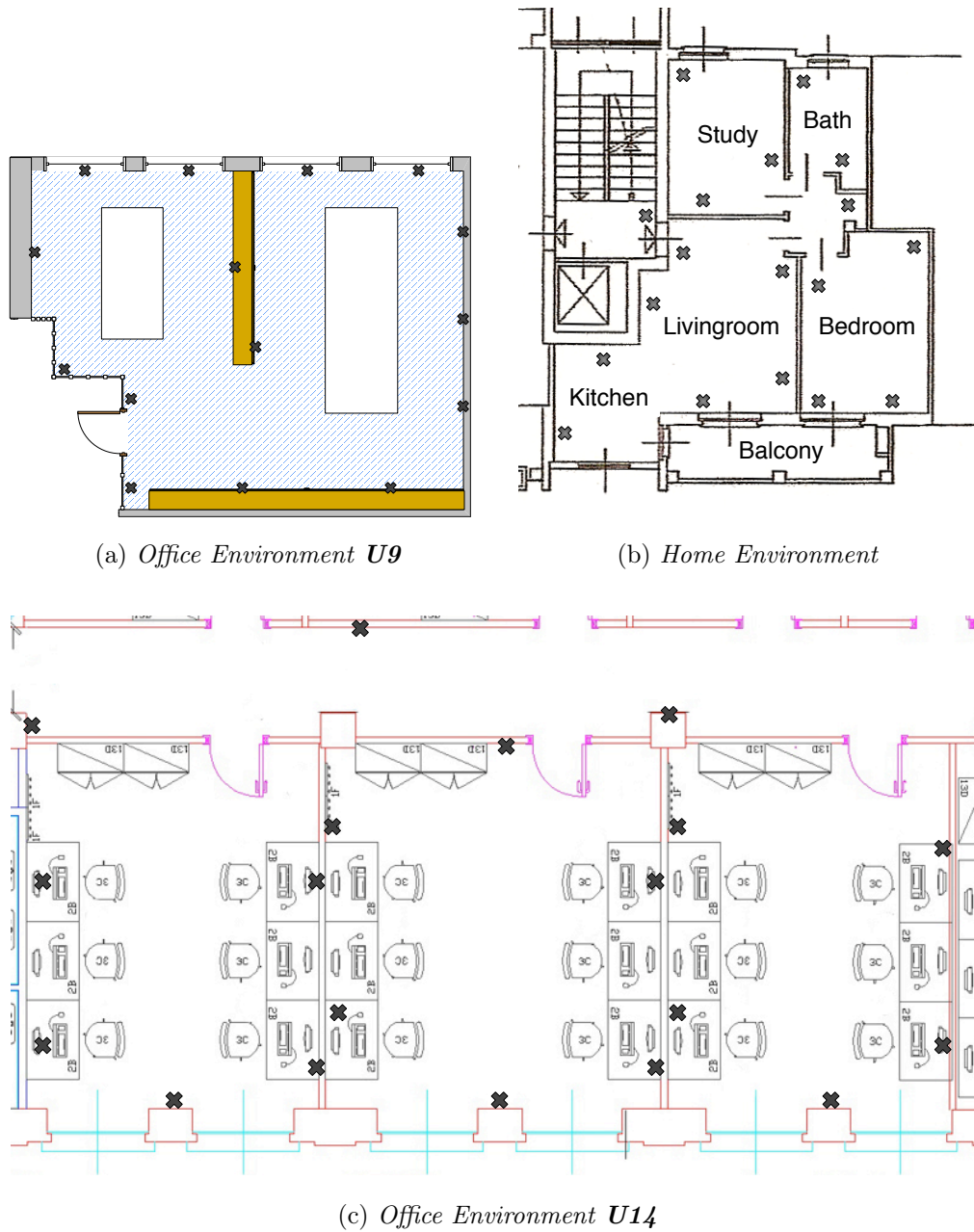


Figure 5.7: Evaluation of the BTTF Framework. Floor Plans of the Testing Environments with Sensor Positions.

Table 5.1: Comparison between BTTF and Ubisense

		Office Env. U9	
		MAE	Accuracy Variance
<i>Benchmark:</i> Ubisense (UWB)		28 cm	10 cm
<i>Prototypes</i>	RSSI only	123 cm	29 cm
	Variation	-340.9%	-188.2%
	RSSI + IMU	103 cm	25 cm
	Variation	-267.8%	-149.4%
	RSSI+PIR+Range+IMU	82 cm	22 cm
	Variation	-194.5%	-115.5%

The Ubisense system relies on time-synchronized and wired infrastructure and the installation process is complex, long and expensive (approximately 30,000€ for 500 m²). For these reasons it has been deployed only in the testing environment *Office Environment U9*.

In typical deployments, the location accuracy of 25 cm in 3D attained 95% of the times has been reported. I measured a slightly worse performance of 28 cm for the 95% of times.

In this tests, the person moved around the testing environments carrying a Ubisense tag and a DiGS Mobile Node at the same time. This was repeated several times, with different routes, and trying to cover the testing space as much as possible.

Table 5.1 summarizes the results of these tests. At first sight, the performance of BTTF seems poorer than Ubisense’s one. However, if we look at these results in terms of infrastructure and deployment cost difference, the performance of BTTF is reasonable. In particular, the performance of the **RSSI+ PIR+ Range+ IMU** configuration is very promising.

The second test phase has been done using the Ubiquicom Locator system as benchmark.

The Ubiquicom Locator [18] is an indoor-outdoor tracking system that functions in areas covered by WiFi signals. The declared location accuracy

Table 5.2: Comparison between BTTF and Ubiquicom

	Office Env. U9		Office Env. U14		Home Env.		
	Mean Accuracy	Variance	Mean Accuracy	Variance	Mean Accuracy	Variance	
<i>Benchmark:</i> Ubiquicom (WiFi)	142 cm	31 cm	135 cm	30 cm	150 cm	32 cm	
<i>Prototypes</i>	RSSI only Variation	123 cm +13.4%	29 cm +6.5%	120 cm +11.1%	28 cm +6.7%	133 cm +11.3%	30 cm +6.3%
	RSSI + IMU Variation	103 cm +27.5%	25 cm +17.9%	105 cm +22.2%	27 cm +10.0%	140 cm +6.6%	33 cm -3.8%
	RSSI + PIR + Range + IMU Variation	82 cm +42.0%	22 cm +29.1%	81 cm +40.1%	24 cm +21.1%	91 cm +39.7%	26 cm +19.0%

is 200 cm in 2D, attained 95% of the times.

Unfortunately, the Ubiquicom system cannot be directly used as ground truth because of its accuracy. Therefore, these tests have been organized as follows. Several ground truth positions have been marked around the testing environments. The person, besides carrying the two tags, follows a predefined route and uses a marker to indicate every ground truth position passed. This was repeated several times, with different routes, trying to appropriately follow them. This approach may be inaccurate if the routes are complex but it has been sufficient for these tests.

Table 5.2 shows the results of the second test phase for the three testing environments. The performance of the BTTF Framework is always better than Ubiquicom's one in every environment.

In conclusion, these tests show that the BTTF Framework has a very good performance in terms of better accuracy and lower costs for the localization infrastructure.

5.4 Chapter Summary

This chapter is the pivot of the thesis work: having described the problem and outlined a new and convenient infrastructure for the collection of data I start describing my solutions to the problem of target localization and tracking.

The chapter describes a framework that implements data fusion by means of the extension of classical statistical techniques. The described framework has been implemented and the results of an elaborated experiment comparing the performance of my tracking system to the performance of existing commercial systems is described. Given the sensitivity of such systems to the environmental conditions the experiment takes into account three very different environments ranging from an office to a household. The collected, preliminary, data suggests that the proposed techniques are worth introducing in real systems.

Chapter 6

Situation Assessment

In this chapter, I will describe an approach to *Situation Awareness* and in particular to the *Situation Assessment* (SA) techniques used to extend and furthermore improve the general behaviour of the previously described data-gathering and tracking frameworks.

The chapter is organized as follows. After an introduction to the SA approach in Section 6.1, Section 6.2 describes an agent-based data generator tool and gives a general overview of the data generation frameworks evaluated and used. In conclusion, Section 6.3 describes the context model, the logic-based techniques and gives a preliminary evaluation.

6.1 Introduction

The notions of *situation*, *situation awareness* and *situation assessment* have been formulated by many authors in various contexts.

In his frequently cited paper “*Toward a Theory of Situation Awareness in Dynamic Systems*” [64], Endsley presents a theoretical model of situation awareness based on its role in dynamic human decision making in a variety of domains. According to this model, *situation awareness* is the perception of environmental elements within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future.

In this paper, Endsley argues that “*it is important to distinguish the term situation awareness, as a state of knowledge, from the processes used to*

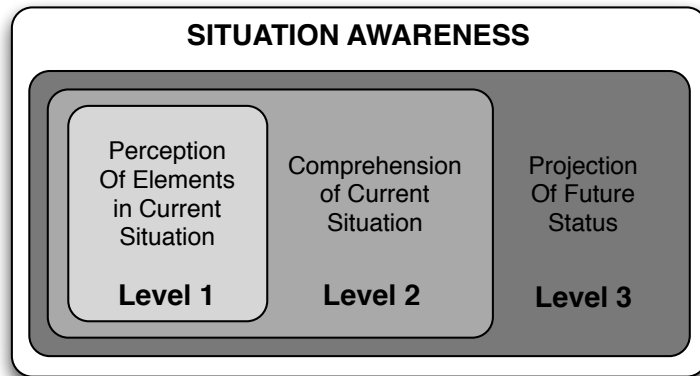


Figure 6.1: Levels of Situation Awareness from the Endsley's Model

achieve that state. These processes, which may vary widely among individuals and contexts, will be referred to as situation assessment or the process of achieving, acquiring, or maintaining the situation awareness."

According to Endsley's model, there are three high-level classes of situation awareness:

- **Level 1.** Perceiving *critical factors* in the environment.
- **Level 2.** Understanding what those factors mean, and in particular their relations in respect to the goal of the system.
- **Level 3.** Understanding what will happen with the system in the near future.

Figure 6.1 shows these levels and the existing relations among them.

The performance of existing localization and tracking systems can be really improved by extending them using situation awareness components. Being aware of what is happening in the deployment environment or achieving a complete knowledge about the current situation are the first steps in order to take decisions and make actions.

In this chapter, I will describe *Situation Assessment* (SA) techniques used to extend and therefore improve the global behaviour of the BTTF tracking framework (see Chapter 5). In particular, the existing framework can be extended, using the three levels of the Endsley's model, as follows.

The first level of the model consists on perceiving (or gathering) data about the environment. The DiGS data-gathering system has been extensively described in Chapter 4. As we have seen, the DiGS system can be used for gathering localization information for tracking tags and for acquiring environmental data in their surroundings. However, DiGS does not address the difficult problem of gathering or simulating large amounts of data (e.g. years) for real contexts.

In order to do that, I designed and realized an agent-based data-generation software using the Repast Symphony modeling toolkit. The data generator will be described in Section 6.2.

The second level of the model is related to the comprehension of the current situation. The situation assessment techniques commonly requires the recognition of the *objects* of a system. Moreover, these techniques involve building relationships between objects and object clusters of a system. The goal of this level is to derive higher-level inference, such as information about activities and other events.

Therefore, I realized a generic context model that can be used to describe the context in which the tracking will happen. This model can also be used to describe the existing relations between the user and the context. The context model is detailed in Section 6.3.2.

Logic-based techniques used to improve the tracking process taking advantage of the context model are described in Section 6.3.3.

The third level of the model is related to the projection of the current situation status in the future. Moreover, at this level, context interpretation techniques are usually used in order to highlight new information and to recognize particular events (e.g. daily activities, emergency situations and so on).

A complete case study of this level, the SINDI system is described in Chapter 7.

In conclusion, situation awareness is viewed as “*a state of knowledge*,” and situation assessment as “*the processes*” used to achieve that knowledge.

6.2 Data Generation with Repast Symphony

Another contribution of this thesis is the design and the implementation of an agent-based data-generation tool using the *Repast Symphony* toolkit.

This tool efficiently generates large data sets simulating several years of the household activities of an agent-modeled person exploiting the Repast S features.

The main agents involved in the simulation are the following:

- **Person.** This agent models the behaviour (movements, perceptions, etc.) of a person living in the simulation environment.
- **FloorCell, EnvironmentCell, LightCell and HeatCell.** These agents are used to model the behaviour of specific aspects of the simulation environment.
- **LocalizationSensor, EnvironmentalSensor.** These are the sensors deployed in the simulation environment.
- **DayManager, HeatManager.** These agents are used to model the daylight and the heat evolution during the data-generation.

After introducing the most used agent-based simulation frameworks, the remainder part of this section will describe the agent-based data-generation tool I designed and implemented. Further details about this agents can be found in Subsection 6.2.5.

6.2.1 Agent-Based Simulation Frameworks

A taxonomy of selected ABM Tools could be found in [65]. Following this classification and after having evaluated the environments described in the remainder of this subsection, I choose to design and implement my data generation tool using the Repast Symphony framework. Figure 6.2 shows the original taxonomy extended with the results of my evaluation.

Swarm

Swarm [66] is a platform for agent-based models (ABMs) that includes:

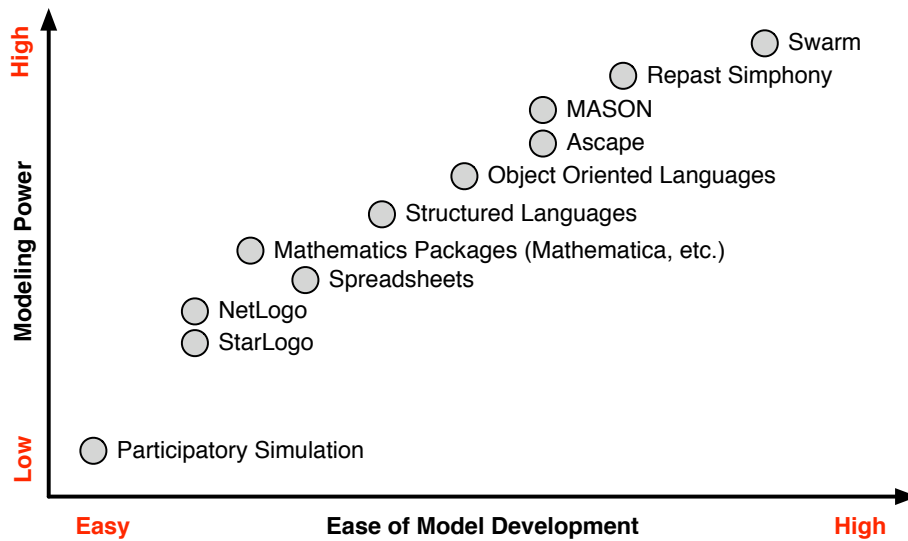


Figure 6.2: A Taxonomy of Selected ABM Tools

- A conceptual framework for designing, describing, and conducting experiments on ABMs;
- Software implementing that framework and providing many handy tools.

As shown in Figure 6.3, Swarm is based on a nested hierarchy of models with the integration of swarm schedules.

This basic architecture is the simulation of collections of concurrently interacting agents: with this architecture, we can implement a large variety of agent based models. Swarm is the first platform in which this hierarchical organization was introduced. Moreover, it is the first social complexity simulator, and many other ABMs (e.g. RepastJ, Ascape and MASON) take inspiration from it.

Swarm software comprises a set of code libraries which enable simulations of agent based models to be written in the Objective-C or Java computer languages.

Repast Symphony

The **R**ecursive **P**orous **A**gent **S**imulation **T**oolkit (*Repast Symphony* or *Repast S* for short) is a free and open source agent-based modeling toolkit.

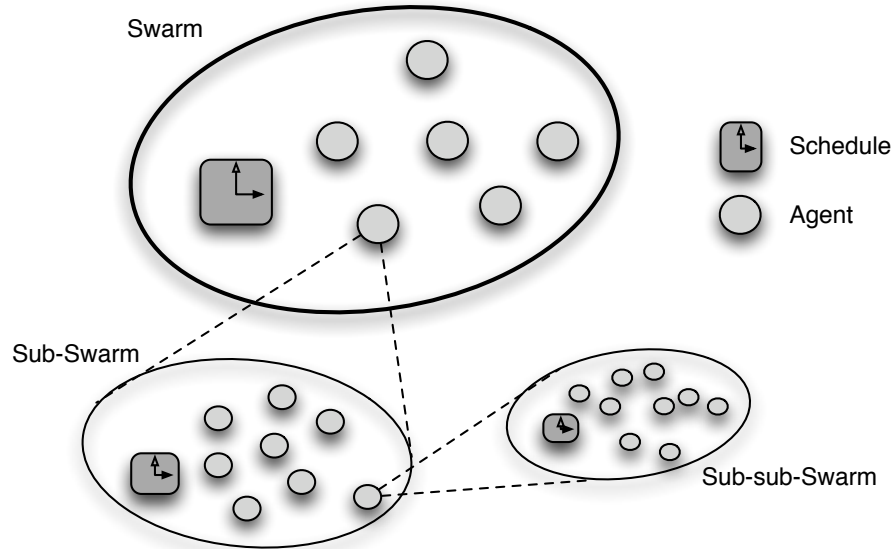


Figure 6.3: Nested hierarchy of Swarms

Similarly to other ABMs, Repast S provides a way to define the so-called “proto-agents”. Generally, the proto-agents are entities with a set of properties and behaviors but without a precise learning behavior. If proto-agents gain learning behavior, they become agents.

Conversely, the Repast S framework introduces new modeling structures: *Contexts* and *Projections*.

Contexts are the core data structure of Repast S. A Context is a simple container based on set semantics that represents an abstract population of proto-agents without providing any mechanism for their interaction.

Projections are data structures designed to define and implement relationships between proto-agents within a given Context. Projections take the meta-population as defined in a Context and impose a relationship on it allowing interaction between proto-agents. Each Context can have an arbitrary number of Projections associated with it.

The Figure 6.4 shows the interaction between Contexts and Projections. On the left you can see a Context filled with agents whereas on the right examples of Projections (a grid, a network and a map).

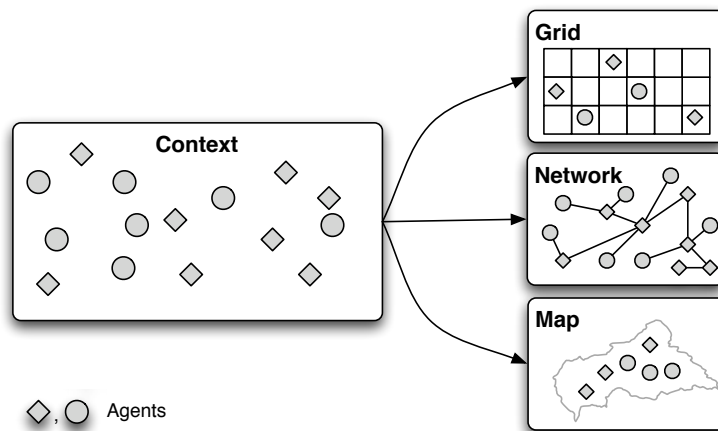


Figure 6.4: Interaction between Contexts and Projections

Other Agent-based Simulation Frameworks

Ascape *Ascape* [67] is a tool for developing and exploring general-purpose agent-based models. In *Ascape*, a high-level framework supports complex model design, while end-user tools make it possible for non-programmers to explore many aspects of model dynamics. *Ascape* is written entirely in Java.

MASON Multi-Agent Simulator Of Neighborhoods (*MASON*) [68] is a discrete-event multiagent simulation library core in Java, designed to be the foundation for large custom-purpose Java simulations, and also to provide more than enough functionality for many lightweight simulation needs. *MASON* contains both a model library and an optional suite of visualization tools in 2D and 3D.

NetLogo *NetLogo* [69] is a programmable modeling environment for simulating natural and social phenomena. *NetLogo* is particularly well suited for modeling complex systems developing over time. Modelers can give instructions to hundreds or thousands of agents all operating independently. This makes it possible to explore the connection between the micro-level behavior of individuals and the macro-level patterns that emerge from the interaction of many individuals. *NetLogo* is based on a previous modeling environment called *StarLogo*.

Other Not Agent-based Frameworks

In this subsection I do not pretend to compile an exhaustive list of simulation, data generation and verification frameworks that are not based on agent models and can be used for intelligent-environment data generation. However I will give a few significant examples.

The paper “*Increasing Reliability in the Development of Intelligent Environments*” [70] gives a good introduction to data generation and verification tools. The paper describes:

FADG. The **F**lexible **A**rtificial **D**ata **G**enerator (*FADG*) is a data-generator program. Its generic component deals with simulation aspects (event management, events, time stamping, etc.) whereas the domain specific component provides the knowledge of a particular environment (layouts, devices, etc.). It can generate events at any level of frequency with a good efficiency (several years of data in a few minutes).

Spin. Spin is a popular open-source software tool that can be used for the formal verification of distributed software systems. Modeling intelligent environments is possible using the **PRO**cess **ME**ta-**L**anguage (Promela) provided by Spin.

UPPALL. UPPAAL is a tool for automatic verification of safety and bounded liveness properties of networks of timed automata implemented in C. The tool could be used to model, simulate and verify intelligent environment systems.

Other interesting examples of data-generation frameworks can be found in [71], [72] and in [73].

6.2.2 Definition of the Simulation Scenario

Providing a way to define a simulation scenario is the first step towards the creation of a data-generation tool. In order to do so, I specified a XML-based scenario-definition file. This scenario file is loaded and parsed during the initialization phase of the Repast S simulator. As described by the following excerpts of the XML definition, it is used to define several aspects of the

```

<scenario>
  ...
  <walls>
    <wall_cell posX="x0" posY="y0" /> ... <wall_cell posX="yN" posY="yM" />
  </walls>
  ...
  <rooms>
    <room key="kitchen" name="Kitchen">
      <area key="kitchen" name="Kitchen_Area">
        <room_cell posX="x0" posY="y0" /> ...
        <room_cell posX="xN" posY="yM" />
      </area>
    </room> ...
  </rooms>
  ...
  <passages>
    <passage fromR="outside" fromA="lift" toR="outside" toA="landing">
      <passage_cell posX="x0" posY="y0" /> ...
      <passage_cell posX="xN" posY="yM" />
    </passage> ...
  </passages>
  ...
</scenario>

```

Listing 6.1: XML-based Scenario Definition: Walls, Rooms, Areas, Passages.

simulation context. For a more formal specification based on the Document Type Definition (DTD) format, see Appendix B.

Listing 6.1 shows the definition of the wall cells. If specified here, a `FloorCell` agent becomes a wall.

More interestingly, Listing 6.1 shows the division in Rooms and Areas of the simulation environment. The *Room* entity identifies rooms in the environment whereas the *Area* entity identifies disjoint areas of interest in a room. Note that the spatial inclusion of areas A_1, \dots, A_n in a room R is such that $\bigcup_{i=1}^n A_i \subset R$.

The hierarchical definition of room and areas is intentionally similar to the hierarchical organization used by the DiGS system (see Chapter 4) and allows to define the network clusters for data generation. In order to generate data sets consistent with those of DiGS, the data-generation tool replicates, for many aspects, the behavior of the DiGS data-gathering algorithms (in particular regarding the person's movements and RSSI computation).

Moreover, Listing 6.1 shows the definition of particular *Passage* cells that

```

<scenario> ...
  <loc_sensors>
    <loc_sensor key="8D:00:00:00:00:01">
      <position posX="x0" posY="y0" orient="E" />
      <pir>
        <pir_cell posX="x0" posY="y0" /> ... <pir_cell posX="xN"
          posY="yM" />
      </pir>
      <range>
        <range_cell posX="x0" posY="y0" /> ... <range_cell posX="xN"
          posY="yM" />
      </range>
    </loc_sensor> ...
  </loc_sensors> ...
</scenario>

```

Listing 6.2: XML-based Scenario Definition: Localization Sensors.

```

<scenario> ...
  <env_sensors>
    <env_sensor key="8D:00:00:00:00:01">
      <position posX="x0" posY="y0" orient="E" /> ...
      <sense_cell posX="x0" posY="y0" />
    </env_sensor> ...
  </env_sensors> ...
</scenario>

```

Listing 6.3: XML-based Scenario Definition: Environmental Sensors.

are used to combine different rooms and areas. Partitioning the environment in this way is particularly useful for the situation assessment techniques (see Section 6.3) in order to efficiently compute user's room changes and walking direction.

Listing 6.2 shows the definition of the *Localization Sensors*. In addition to the sensor position on the grid and its orientation, the definition specifies the behavior of the passive-infrared and movement sensors in terms of cells covered.

Listing 6.3 shows the definition of the *Environmental Sensor*. During the data-generation phase, these sensors can read the temperature, humidity and brightness values of the EnvCell agent that is positioned as defined by the `<sense_cell>` element.

```

<scenario> ...
  <light_sources>
    <light_source room="kitchen" key="kitchen-kitchen" type="natural">
      <light_cell posX="x0" posY="y0" />...<light_cell posX="xN" posY="yM" />
    </light_source> ... </light_sources> ...
  <heat_sources>
    <heat_source room="kitchen" key="kitchen-kitchen" type="fan-coil">
      <heat_cell posX="x0" posY="y0" />...<heat_cell posX="xN" posY="yM" />
    </heat_source> ... </heat_sources> ...
  <windows>
    <window key="kitchen" light_source="kitchen-kitchen"
      initialState="close" /> ...
  </windows>
  <light_switches>
    <light_switch key="kitchen" light_source="kitchen-lamp"
      initialState="turn-off">
      <position posX="14" posY="6" />
    </light_switch> ...
  </light_switches>
</scenario>

```

Listing 6.4: XML-based Scenario Definition: Lights and Heat.

In conclusion, Listing 6.4 shows the definition of *Light* and *Heat Sources*. These elements define the properties of the objects that are used to manage and propagate the state of light and heat throughout the simulation scenario. Moreover, Listing 6.4 shows the elements used to define the position and the initial status of *Windows* and *Light Switches* in the simulation environment.

6.2.3 Context and Projections

The simulation scenario is modeled by a Repast S model that describes a main Context associated with four Projections. Figure 6.5 shows the main Projections of the model. Three $N \times M$ grid projections are used to define:

- **Floor Plan.** Floors and walls are defined by a grid of FloorCell agents by appropriately setting their boolean property *wall*.
- **Localization Sensor Grid.** The second grid projection is used to position the LocalizationSensor agents in the scenario and furthermore during the localization data generation.

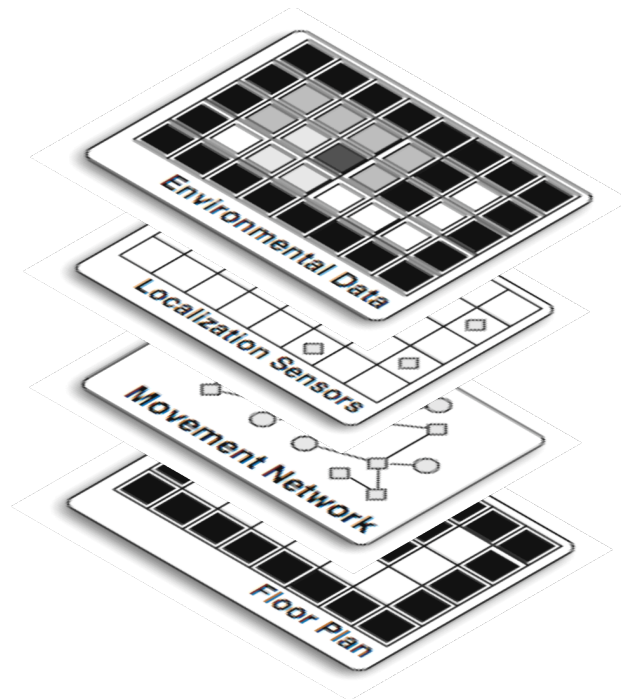


Figure 6.5: Projections used by the Repast S model and their relations

- **Environmental Data Grid.** The last grid project is used to position the EnvironmentalSensor, LightCell and HeatCell agents in the environment. Moreover the agents use the grid to simulate the propagation of environmental values such as humidity, temperature and brightness in the context.

The **Movement Network Projection** defines two undirected weighted graphs having FloorCell agents as nodes. The first graph is mainly used for the movement of the Person agent computed using Dijkstra's shortest path algorithm. Therefore, its weights are computed in order to avoid movements across wall cells.

The second graph is mainly used during the RSSI-data generation to verify if a direct line-of-sight between the person and the localization sensors exists. This information is useful in order to better simulate the RSSI propagation model. In order to do so, the weights of the second graph are computed without considering the walls. The line-of-sight information is computed comparing the results of shortest path algorithm between a localization sensor and the person obtained using the two graphs. These results

will differ only if there is a wall between the localization sensor and the person.

During the initialization phase, the context and the projections of the simulation model are dynamically populated reading the XML-based scenario file using a context-creation class, specifically defined in the Repast S framework. Furthermore, this is not the only method that can be used to configure the simulation environment: Repast S can load the required initialization data in several ways (e.g. using agent definitions directly stored in a relational database or in text files).

6.2.4 Event Scheduling

The data-generation is driven by two different event schedulers that are used to: (i) simulate the behaviour and the habits of the person, (ii) simulate light, temperature and humidity evolution during the day.

In order to simulate the behaviour and the habits of a person living and moving in a realistic home environment, I defined a XML-based file that contains a detailed list of both high- and low-level activities that the *Person* agent is required to execute during the data generation.

As also shown in Listing 6.5, the first part of the XML definition describes the high level activities. In addition to a starting and an ending time, every *Activity* is composed of one or more *ActionSets*. The same *ActionSets* are composed of several low-level *Actions* that indeed specifies what the *Person* agent should do.

The current version of the simulation tool requires a full description of the low-level actions. This is a simple and efficient approach, nonetheless it is not the right solution when dealing with complex day schedules. Enhancing the capabilities of the *Person* agent in order to independently compute the set of actions needed to fulfill a high-level *Activity* is the right way to tackle this problem. Despite that, this improvement of is out of the scope of this thesis.

The second event scheduler used by the data generation tool is the one controlling the evolution of the natural light during the simulated day. As shown in Listing 6.6 it is also based on a XML definition file.

```

<schedule descr="Normal_Day" startAt_HH="00" startAt_MM="00" endAt_HH="23"
  endAt_MM="59">
  <activities>
    <activity name="night-toilet" startAt_HH="05" startAt_MM="00">
      <actionSet name="night-toilet" />
    </activity>
    <activity name="wake-up" startAt_HH="08" startAt_MM="00">
      <actionSet name="wake-up" />
      <actionSet name="opening-windows" /> ...
    </activity> ...
  </activities>
  ...
  <actionSets>
    <actionSet name="night-toilet">
      <action type="turn-on" switch="bedroom" posX="48" posY="21"
        waitMM="0" descr="turn_on_bedroom_light" />
      <action type="turn-on" switch="passage" posX="36" posY="26"
        waitMM="0" descr="turn_on_passage_light" />
      <action type="turn-on" switch="bathroom" posX="30" posY="35"
        waitMM="0" descr="turn_on_bathroom_light" />
      <action type="goto" posX="35" posY="44" waitMM="5" />
      <action type="turn-off" switch="bathroom" posX="30" posY="35"
        waitMM="0" descr="turn_off_bathroom_light" />
      <action type="turn-off" switch="passage" posX="36" posY="26"
        waitMM="0" descr="turn_off_passage_light" />
      <action type="turn-off" switch="bedroom" posX="48" posY="21"
        waitMM="0" descr="turn_off_bedroom_light" />
      <action type="goto" posX="45" posY="15" waitMM="5" />
    </actionSet>
    <actionSet name="wake-up">
      <action type="goto" posX="49" posY="25" waitMM="3" />
    </actionSet>
    <actionSet name="opening-windows">
      <action type="open" window="bedroom" posX="33" posY="7" waitMM="1"
        descr="open_bedroom_windows" />
      <action type="goto" posX="27" posY="25" waitMM="0" />
      <action type="open" window="livingroom" posX="22" posY="8"
        waitMM="1" descr="open_livingroom_windows" />
      <action type="goto" posX="12" posY="7" waitMM="0" />
      <action type="open" window="kitchen" posX="12" posY="3" waitMM="1"
        descr="open_livingroom_kitchen" />
    </actionSet>
  </actionSets>
</schedule>

```

Listing 6.5: XML-based Person Schedule Definition: Activities and Actions.


```

<daylight_Temp_Hum_Schedule>
  <externalLight>
    <extLightSched fromHH="00" fromMM="00" toHH="04" toMM="59" light="0" />
    ...
    <extLightSched fromHH="12" fromMM="00" toHH="12" toMM="59" light="500"
      /> ...
    <extLightSched fromHH="20" fromMM="00" toHH="23" toMM="59" light="0" />
  </externalLight>
  <temperature>
    <extTempSched fromHH="00" fromMM="00" toHH="04" toMM="59" temp="16" />
    ...
    <extTempSched fromHH="20" fromMM="00" toHH="23" toMM="59" temp="15" />
  </temperature>
  <humidity>
    <extHumSched fromHH="00" fromMM="00" toHH="04" toMM="59" humid="33" />
    ...
    <extHumSched fromHH="20" fromMM="00" toHH="23" toMM="59" humid="50" />
  </humidity>
</daylight_Temp_Hum_Schedule>

```

Listing 6.6: An Short Example of Schedule Definition for Daylight, Temperature and Humidity.

6.2.5 Agents and other Objects

As previously described, several agents are involved in the data generation. This subsection details the main agents and other notable objects involved in the simulation.

The *Person* agent follows the previously described activity schedule moving in the FloorPlan grid projection and interacting with other agents (e.g. for opening doors and windows or for turning on and off switches). Therefore, it has the perception of the other objects that exists in the simulated house. In the current version of the simulator these perceptions are quite limited but sufficient for a rich data generation. I considered to further improve the perception model of the Person agent using a well-know human behaviour model called **Physical Conditions**, **Emotional States**, **Cognitive Capabilities** and **Social Status** (PECS) [74].

During every simulation tick, the *LocalizationSensor* agents compute the RSSI, passive infrared and movement measurements considering the position of the Person agent in the FloorPlan. The measurement computation takes clearly into account the model of every sensor. RSSI values are computed computing the distance between the localization sensor and applying

the radio propagation model, whereas PIR and Movement considering if the person is in one of the cells monitored by the sensor, as defined in the XML scenario file. *EnvironmentalSensor* agents generate temperature, humidity and light measurements periodically examining the state of the associated *EnvironmentCell* agent (as also described in the XML scenario definition). Furthermore, all the measurements are stored in relational database.

The *DayManager* and *HeatManager* agents are used to manage the evolution of daylight and external temperature and humidity during the data-generation. Using the previously described schedule they modify the state of *LightSources* and *HeatSources* objects. The same *LightSources* and *HeatSources* modify the state of the associated *LightCells* and *HeatCells*. *LightSources* and *HeatSources* are associated with *Door* and *Window* objects and with this double passage makes it possible to manage the propagation of light and heat according to the state of doors and windows (e.g. the internal light of a room is zero (dark) if its window is closed, even if the external light has high values).

The *FloorCell*, *EnvironmentCell*, *LightCell* and *HeatCell* agents are arranged in grid projections in order to replicate the behaviour of walls and temperature, humidity, light variations in the simulation environment. This value propagation is implemented using with a controlled flooding algorithm [75, 37]. Every agent propagates temperature, humidity and light changes only if it has not already received the same update request. Therefore, these update messages have, in addition to the new light or heat values, an indication of the simulation tick in which they occur.

Figure 6.6 and 6.7 show the Repast S toolkit in execution with different projection displays.

6.3 Logic-based Situation Assessment

In situation assessment, context-dependent data can arise from different sources; for example data may be gathered by sensors or collected from several knowledge-bases. The incompleteness and the heterogeneous nature of such data stresses the need for expressive reasoning techniques for Situation Assessment.

The Knowledge Representation and Reasoning framework we used is that of Answer Set Programming (ASP).

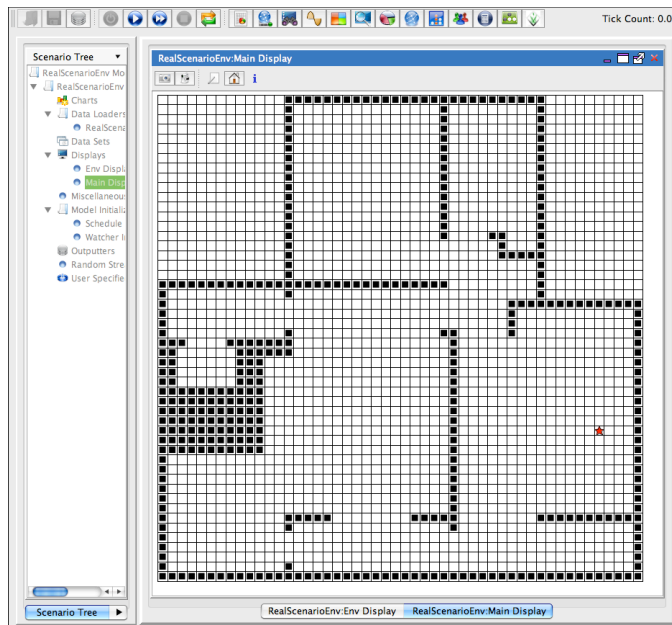


Figure 6.6: The Repast S toolkit showing the main Projection, walls (black squares) and the person (red star).

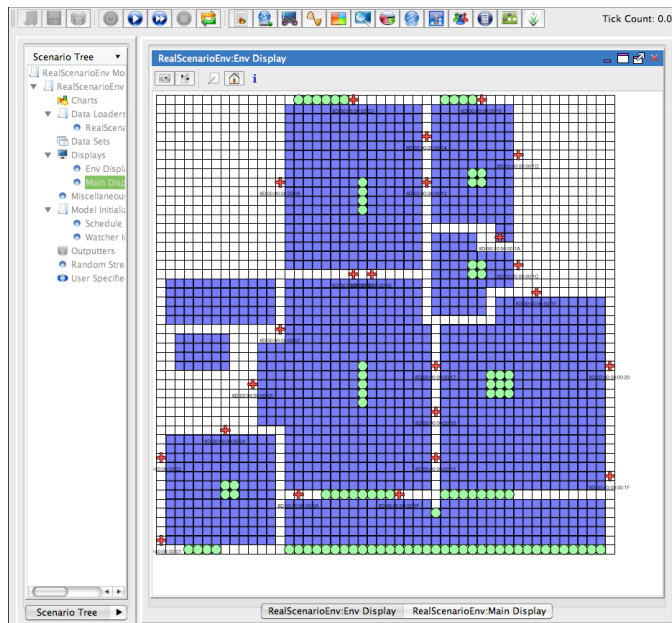


Figure 6.7: The Environmental Data Projection showing the light cell (green circles), environmental sensors (red crosses) and cells (purple squares).

Compared to pure statistical approaches, the support provided by logic inference based on ASP is highly expressive and computationally efficient because it can deal with first-order representations, which are much richer than the propositional ones characterizing probabilistic inference. Furthermore, ASP can deal with incomplete information and commonsense reasoning using defaults. Cardinality and weight constraints together with optimization techniques are also interesting features for this application, in that they can be used to model preferences and different degrees of uncertainty [76, 77, 78, 79].

The logic approach to problem solving based on ASP has the following desirable properties:

- *readability* of the problem specification;
- *flexibility* with respect to the possibility of combining heterogeneous sources of knowledge;
- *modularity* in the specification of the problem that describes properties of a desired solution;
- *expressivity* of the modeling language and computational efficiency of the inference engines.

In the remaining part of this section, I will provide some preliminary notions on ASP that can be useful in order to understand the encoding, the specification of a logic-based context model and a detailed description of how ASP techniques have been used to support localization and Situation Assessment.

6.3.1 Answer Set Programming: some notions

Answer Set Programming (also known as Stable Logic Programming or A-Prolog) is a methodology based on the Stable Model Semantics proposed in [80] for solving computational problems. Compared to the traditional Logic Programming approach, the major advantage of ASP is its simplicity and connection to constraint programming and deductive databases.

In ASP a given problem is represented by a logic program viewed as constraints on the solutions. The logic program is written in such a way that its solutions correspond to the solutions of the original computational problem, which are given in terms of *answer sets*.

A logic program P is a finite set of rules r_i of the form

$$r_i : L_0 \leftarrow L_1, \dots, L_m, \text{ not } L_{m+1}, \dots, \text{ not } L_n .$$

where L_i ($i = 0..n$) are literals, *not* is a logical connective called *negation as failure* and $n \geq m \geq 0$. We define $L_0 = \text{head}(r)$ as the *head* of rule r_i , and $\text{body}(r_i) = L_1, \dots, L_m, \text{ not } L_{m+1}, \dots, \text{ not } L_n$ as the *body* of r_i . Furthermore, let $\text{body}^+(r_i) = \{L_1, \dots, L_m\}$ and $\text{body}^-(r_i) = \{\text{not } L_{m+1}, \dots, \text{not } L_n\}$.

Rules r_i with $\text{head}(r_i) = \emptyset$ are called *integrity constraints*, while if $\text{body}(r_i) = \emptyset$, we refer to r_i as a *fact*.

An interpretation I is represented by the set of atoms that are true in it. A *model* of a program P is an interpretation in which all rules of P are true according to the standard definition of truth in propositional logic. Apart from letting ‘,’ stand for conjunction, this implies treating rules and default negation by *not* as implications and classical negation, respectively. Note that the (empty) head of an integrity constraint is false with respect to every interpretation, while the empty body is true with respect to every interpretation.

Intuitively, a stable model is a possible view of the world that is *compatible* with the rules of the program. Rules are therefore seen as constraints on these views of the world.

Let us start defining stable models of the subclass of positive programs, i.e. those where, for every rule r_i , $\text{neg}(r_i) = \emptyset$.

Definition 6.3.1 (*Stable model of positive programs*)

The stable model $a(\Pi)$ of a positive program Π is the smallest subset of IB_Π such that for any rule r_i in Π :

$$L_1, \dots, L_m \in a(\Pi) \Rightarrow A \in a(\Pi) \quad (6.1)$$

Clearly, positive programs have a unique stable model, which coincides with that obtained applying other semantics; in other words positive programs are unambiguous.

Definition 6.3.2 (*Stable models of programs*)

Let Π be a logic program. For any set S of atoms we built a reduct Π^S of Π as follows:

- For each rule r_i , if at least one of the L_{m+1}, \dots, L_n of r_i is in S , then the rule can not fire, so we throw the rule out of the reduct;
- For each rule r_i , if atom $L_i, i \in \{m+1, \dots, n\}$ is not in S , its negation “not L_i ” can be dropped from the body of r_i because it is always true; thus if none of the L_{m+1}, \dots, L_n of r_i is in S , then the rule can be rewritten as:

$$r_i : A \leftarrow L_1, \dots, L_m \quad (6.2)$$

Clearly, $[\Pi]^S$ does not contain not, so that its stable model is already defined. If this stable model coincides with S , then we say that S is a stable model of Π . In other words, a stable model of Π is characterized by the equation:

$$S = a(\Pi^S). \quad (6.3)$$

Answer sets of P are particular models of P satisfying an additional stability criterion. Roughly, a set X of atoms is an answer set, if for every rule r_i of the program, $L_0 \in X$ whenever L_1, \dots, L_m belong to X and no L_{m+1}, \dots, L_n belongs to X .

Formally, an *answer set* X of a program P is a minimal (in the sense of set-inclusion) model of

$$\{head(r_i) \leftarrow body^+(r_i) \mid r_i \in P, body^-(r_i) \cap X = \emptyset\}.$$

Although answer sets are usually defined on ground (i.e., variable-free) programs, the rich modeling language of ASP allows for non-ground problem encodings, where rules with variables (upper case names) are taken as a shorthand for the sets of all their ground instantiations. Grounders, such as *gringo* [81] and *lparse*, are capable of combining a problem encoding and a problem instance (typically a set of ground facts) into an equivalent ground program, which can then be then processed by one of the available ASP solvers. For the implementation of the situation assessment techniques described in this thesis, answer set programs are grounded using Gringo [79] and interpreted using the Clasp [82] solver.

For extending the expressive power of logic programs, I take advantage of three kind of rules [77].

The first type is the **cardinality rule**, which is of the form:

$$r_i : h \leftarrow l \{a_1, \dots, a_m\} u \quad (6.4)$$

where $C = l \{a_1, \dots, a_m\} u$ is a *cardinality constraint* and l and u are two integers giving a *lower* and *upper* bound, respectively, on the number of satisfied literals within the constraint.

For a cardinality constraint C as in the body of (6.4), we let $lit(C)$ denote its set of literals $\{a_1, \dots, a_m\}$ and let $lb(C) = l$ and $ub(C) = u$. C is *satisfied* by a set of literals S , if

$$lb(C) \leq |lit(C) \cap S| \leq ub(C) .$$

Whenever bound l or u is missing, it is taken to be 0 or $|lit(C)|$, respectively.

Rule 6.4 is interpreted as follows: if at least l and at most u literals in the set $\{a_1, \dots, a_m\}$ are satisfied by a stable model S , then atom h should be in S .

Example 6.3.1 “Once is a mistake, twice is a habit” is expressed by the rule:

$$habit \leftarrow 2 \{mistake_1, \dots, mistake_n\}.$$

The second type of rule is the **choice rule**, which is of the form

$$r_i : \{h_1, \dots, h_k\} \leftarrow body(r_i). \quad (6.5)$$

which implements a nondeterministic choice over the atoms in $\{h_1, \dots, h_k\}$ whenever $body(r_i)$ is true in the stable model. That is, the rule generates solutions including any number of atoms among $\{h_1, \dots, h_k\}$.

Example 6.3.2 Choose exactly one (different) complementary course for each school year.

$$\begin{aligned} & 1 \{course(subj_1, N), \dots, course(subj_k, N)\} 1 \leftarrow year(N). \\ & \leftarrow course(C, N), course(C, N_1), N \neq N_1. \end{aligned} \quad (6.6)$$

The third type of rule is the **weight rule**, which is of the form

If we associate a real-valued coefficient, i.e., a weight to each atom in the cardinality constraint, we obtain a generalization of rule 6.4, called *weight rule* of the form

$$r_i : h \leftarrow l \{a_1 = w_{a_1}, \dots, a_m = w_{a_m}\} u \quad (6.7)$$

where l and u are again two integers giving a *lower* and *upper* bound, respectively, on the sum of the weights of satisfied literals within the constraint.

The idea is that a stable model satisfies the constraint if the sum of the weights of the literals satisfied by the model is between l and u . This means that the head of rule 6.7 is included in a stable model S if $l \leq \sum_{a_i \in S} w_{a_i} \leq u$.

Whenever bound l or u is missing, it is taken to be $-\infty$ or $+\infty$, respectively.

Example 6.3.3 *A student is admitted to a school year, if he has passed at least 18 credits in the previous year.*

$$\text{admitted} \leftarrow 18 \{ \text{passed}(\text{course}_1, n_1) = n_1, \dots, \text{passed}(\text{course}_k, n_k) = n_k \}. \quad (6.8)$$

Answer sets of programs with cardinality and weight constraints are formally defined as in [83].

6.3.2 Definition of a Context Model

A model of context is as a suitable description of the domain knowledge we need to represent and describe elements in a context. A well designed model is crucial for any context-aware system. In the literature there is a rich variety of context models discussed and proposed for different purposes [84, 85].

While most of the implemented context models are domain-dependent and do not support powerful inference, logic-based models fail to provide a representation of context-dependent data that is both expressive and with good computational properties [86]. The logic model described in this section aims at being generic and computationally rich at the same time.

Previous investigation of context models has indicated that there are certain entities in a context that, in practice, are more important than others. These are location, identity, activity and time [87, 88]. In fact, in the context of home monitoring, the more intuitively relevant aspects of a context are: *where* you are, *who* you are (clinical profile), *which* resources you are using, *what* you are doing and *when*.

In order to represent this information in the model, the following entities have been identified:

Table 6.1: Generic Spatial Relations Among Entities.

Relation	Object	Reference Object	Relation Type
<i>person_in</i>	Person	{Room,Area}	Generic directional relation
<i>in</i>	Area	Room	Generic directional relation
	Object	{Room, Area}	
<i>under</i>	Object	Object	Generic directional relation
<i>connect</i>	Room	Room	Generic directional relation
<i>on</i>	Object	Object	Generic directional relation
<i>near</i>	Object	Area	Generic distance relation

- *Person* entity to identify the person and her movements;
- *Room* entity to identify rooms in the environment;
- *Area* entity to identify disjoint areas of interest in a room;
- *Object* entity to identify objects or resources the person can interact with.

A physical description of the environment is available in terms of coordinates in a bidimensional grid where each cell is identified by coordinates X, Y and has some properties (being in a room, being a wall, being a passage, etc.).

A set of generic spatial relations has been taken into account rather than describing geometric spatial relations between objects. This results in greater generality because a complete physical description of the environment in terms of absolute distances is not needed. These relations among entities are summarized in Table 6.1¹.

Generic spatial properties can be used in the inference process to better evaluate the feasibility of a solution. As an example, the system can infer that the person is in a given area A_j of a room R_i on the basis of information provided by motion sensors only. However, coordinates evaluation could return a different localization result, thus the system could deduce that the moving object could be a cat or something else.

All the other pieces of information are available at a higher level of detail and can be indexed as attributes of the context entities: while data gathered

¹Note that the spatial inclusion of areas A_1, \dots, A_n in a room R is such that $\bigcup_{i=1}^n A_i \subset R$.

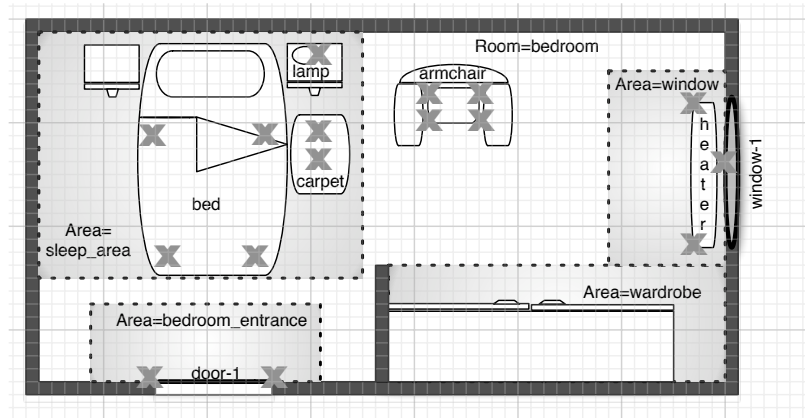


Figure 6.8: Example: Modelling a Bedroom

by the sensors are processed and aggregated according to specific algorithms for feature analysis, the information available at upper levels is filtered by the abstraction. This makes it possible to represent meaningful information as properties of objects, rooms or areas, keeping the model independent of sensors' characteristics and positioning. The resulting context specification is then mapped into a set of logic predicates in ASP.

An example of how a bedroom can be represented in our context model is illustrated in Figure 6.8, where sensors are represented just for the sake of completeness but they do not appear directly in the knowledge describing the context model: only the information associated with the entities is represented.

Attributes' value may come from (i) external knowledge (such as observed values for attributes of the Person entity), (ii) opportunely aggregated sensor data (such as attributes related to the environment) or (iii) results of the inference process (inferred values for attributes of the Person entity, when observed values are not available). Values of both attributes and spatial relations (except the inclusion of an area in a room which is static) are dynamic and need to be associated to an interval of time. In this way, the reasoning system can take into account their evolution during context interpretation. In order to reason about situation assessment, it is necessary to conceive a model that is able to support commonsense reasoning, reasoning about actions and qualitative preferences.

Occasional lack of specificity with respect to the spatial relations is compensated by the inference process: reasoning about relations and attribute values may help inferring new information.

Table 6.2: Attributes representing information about the Person entity.

Signature	Domain Values	Description
<wgt, Val>	Val={1..300}	Weight (kg)
<motion,Val,P ¹ >	Val={walk, still, zeroSignal ² }	Motion activity
<posture,Val,P ¹ >	Val={sit, lay, stand, zeroSignal ² }	Posture of the person
<orientation,Val,P ¹ >	Val={turn, straight, zeroSignal ² }	Direction of motion

¹ Parameter P={0..100} represents data reliability, and it is computed by the algorithms used for feature extraction.

² Value *zeroSignal* is related to the fact that no signal is received from sensors detecting movement.

Table 6.3: Attributes representing information about Room and Area entities.

Attribute Name	Domain Values	Description
ambientLight	{dark, ..., bright}	brightness of the environment
ambientLightType	{natural, artificial}	nature of the light
ambientHumidity	{dry, medium, wet, super-Wet}	humidity level
ambientTemperature	{cold, chilly, warm, hot, burning}	temperature
ambientSound	{mute, mild, medium, noisy}	noise level
presence	{yes, no}	presence of a moving entity
noxiousGas	{yes, no}	presence of noxious gas
smoke	{yes, no}	presence of smoke

Table 6.4: Attributes representing information about the Object entity.

Signature	Domain Values	Description
objectLight	{dark, ..., bright}	light produced by the object
objectLightType	{natural, artificial}	nature of the light
objectHumidity	{wet, dry}	humidity on object
objectTemperature	{hot, cold}	meaning depends on object
objectSound	{none, regular, loud}	meaning depends on object
switch	{open, closed}	state of doors and windows objects
state	{on, off}	state of on/off devices
filteredLoad	{0..300}	weight measurement from load-cells
loadVolatility	{stable, mild, unstable}	volatility of the load measurement
waterflow	{yes, no}	water flowing through the object
presence	{yes, no}	moving entity detected by the object

Attributes associated with entities included in our model and their values are detailed in Tables 6.2, 6.3 and 6.4. Most of the values for attributes associated to rooms, areas and objects are the result of a process that maps numerical sensor data into meaningful thresholds. The thresholds are derived both from objective considerations, e.g. a given temperature might be too hot for the human body to survive, and from patient-profile and environment related considerations, a southern Italian and a British person might have a very different idea of what is comfortably hot or cold.

The continuous measurements provided by sensors are stored in a relational database as well as results of reasoning. In the actual implementation, discrete time stamps represent seconds and numerical sensor values are mapped into thresholds. A new value for the attribute of an entity is extracted when its mapping changes from a given threshold to a different one. As an example, the attribute representing environmental light is aggregated as the average light in a room with a value $V \in \{0..500\}$, mapped in the following thresholds V_t according to the value of V :

$$\begin{aligned}
 V_t &= \textit{dark} & \textit{if } V &\in \{0..125\} \\
 V_t &= \textit{shadow} & \textit{if } V &\in \{126..250\} \\
 V_t &= \textit{clear} & \textit{if } V &\in \{251..375\} \\
 V_t &= \textit{bright} & \textit{if } V &\in \{376..500\}
 \end{aligned}$$

A logic fact of the form $\textit{attribute_env}(\textit{ambientLight}, \textit{Room}, V_t, T)$ is

generated whenever the new light value is mapped to a different threshold. I am currently working on a wrapper that will make it possible to use the ASP solver as a permanently running API to feed with aggregated sensor data as soon as they are available. This will make the reasoning process faster.

As an example, to model the environment illustrated in Figure 6.8 at time T , the partial description obtained by instantiating the first order specification in terms of propositional predicates, would be similar to that described in Listing 6.7.

```
obj(door). obj(lamp). obj(carpet). obj(heater).
obj(armchair). obj(window). obj(bed).
area(sleep_area). area(wardrobe). area(entrance).
room(bedroom).

in(sleep_area, bedroom, 1).
in(wardrobe, bedroom, 1).
in(entrance, bedroom, 1).

in(door, bedroom, 1). in(heater, bedroom, 1).
in(lamp, sleep_area, 1). in(carpet, sleep_area, 1).
in(armchair, bedroom, 1). in(window, bedroom, 1).
in(bed, sleep_area, 1).

objectLight(lamp, bright, 1).
attribute_env(ambientLight, bedroom, bright, 1).
switch(door_1, open, 1). switch(window_1, close, 1).
filteredLoad(armchair, 0, 1).
filteredLoad(carpet, 0, 1).
filteredLoad(bed, 0, 1).
...
```

Listing 6.7: Partial Description of an Example Environment in terms of Propositional Predicates

Provided this description, commonsense reasoning can be used for situation assessment even when only sparse sensor data are available, as illustrated later in this section.

6.3.3 Intelligent Support to Localization

In order to understand how reasoning helps in the interpretation of imprecise sensor data to support situation assessment, let us consider localization of the person in a position X, Y of the grid representing the home environment.

SINDI's localization component is based on the intensity variations of the radio signals exchanged between nodes, filtered by a bayesian filter. I referred to this processed values as to *BTTF results*. An additional aggregation process makes it possible to associate each cell of the grid with a numerical value that is proportional to the probability that the person is in that cell according to BTTF results. We refer to this numerical value as to a *likelihood*.

Unfortunately, it is not always true that the higher likelihood for a cell, the closest the person is to the cell.

Given proximity values returned by the bayesian filter with a certain likelihood P at a given time T (provided as facts of the form $in(X, Y, T, P)$), the ASP program takes available sensor data that can be used to validate these positions and reasons about several preference criteria to identify the best solutions.

When results of the bayesian filter are available at time T and likelihoods are provided for a set of cells, the first optimization criteria is represented by a cost function $reward(X, Y, T, V)$ combining the likelihood with the support obtained from other sensor data (namely range finder and passive infrared) for a given cell X, Y of the grid. This information about additional sensor data is represented by a set of facts of the form $sense(S, X, Y, T)$, where S =distance for the range finder and S =motion for the passive infrared sensor. Further criteria are *coherence*, representing the percentage of sensor data received with respect to the expected data for a given cell X, Y and *best move*, representing the closest position with respect to the previous one.

All possible positions are generated for a given time T and the optimal solutions are obtained by applying different combination of the preference criteria: when likelihood values are available, the most supported signal which is most coherent and best move is preferred. If no such position exists, then coherence is preferred to the best move criterion; when results of the bayesian filter are not available (no cells are provided as possible location), then a valid move that is coherent is the best candidate, finally we give up coherence if no such position exists.

Listing 6.8 shows the correspondent encoding for the case likelihoods are available².

²Note that the symbol “:-” in the encodings represents the implication “ \leftarrow ”.

Listing 6.8: ASP Encoding for the Case When Results from Bayesian Filters and Associated Likelihoods are Available

```

sense_domain(X,Y,T) :- in(X,Y,T,P) .
sense_domain(X,Y,T) :- sense(S,X,Y,T) .

has_data(T)          :- sense(S,X,Y,T) .
sensed_data(X,Y,T)  :- sense(S,X,Y,T) .
has_rssi(T)         :- in(X,Y,T,P) .
in_project(X,Y,T)   :- in(X,Y,T,P) .

count_sensed(N,X,Y,T) :- N = [sense(S,X,Y,T) : sensed_type(S)=1,
    in_project(X,Y,T)=1], sense_domain(X,Y,T) .
reward(X,Y,T,V) :- in(X,Y,T,P) , count_sensed(N,X,Y,T) , V=P+N*100 .

exists_better_rssi_a(X,Y,T) :- reward(X,Y,T,V) , reward(X1,Y1,T,V1) , V1>V,
    most_coherent(X1,Y1,T) , best_move(X1,Y1,T) .
exists_better_rssi_b(X,Y,T) :- reward(X,Y,T,V) , reward(X1,Y1,T,V1) , V1>V,
    most_coherent(X1,Y1,T) .
exists_better_rssi_c(X,Y,T) :- reward(X,Y,T,V) , reward(X1,Y1,T,V1) , V1>V,
    best_move(X1,Y1,T) .
exists_better_rssi_d(X,Y,T) :- reward(X,Y,T,V) , reward(X1,Y1,T,V1) , V1>V .

max_rssi_a(X,Y,T) :- not exists_better_rssi_a(X,Y,T) , in_project(X,Y,T) ,
    most_coherent(X,Y,T) , best_move(X,Y,T) .
max_rssi_b(X,Y,T) :- not exists_better_rssi_b(X,Y,T) , in_project(X,Y,T) ,
    most_coherent(X,Y,T) , not best_rssi_a(T) .
max_rssi_c(X,Y,T) :- not exists_better_rssi_c(X,Y,T) , in_project(X,Y,T) ,
    best_move(X,Y,T) , not best_rssi_b(T) , not
    best_rssi_a(T) .
max_rssi_d(X,Y,T) :- not exists_better_rssi_d(X,Y,T) , in_project(X,Y,T) ,
    not best_rssi_c(T) , not best_rssi_b(T) , not
    best_rssi_a(T) .

best_rssi_a(T) :- max_rssi_a(X,Y,T) .
best_rssi_b(T) :- max_rssi_b(X,Y,T) .
best_rssi_c(T) :- max_rssi_c(X,Y,T) .

%%% Physics invalid %%%
invalid(X,Y,T) :- wall(X,Y) .

%%% Invalid moves %%%
step(0;1;-1) .
valid_move(X1,Y1,T) :- person_in(X,Y,T-1) , X1=X+Z1 , Y1=Y+Z2 , step(Z1;Z2) ,
    poss_locate(X1,Y1,T) .
invalid_move(X,Y,T) :- not valid_move(X,Y,T) , poss_locate(X,Y,T) .

distance(X,Y,T,D) :- person_in(X1,Y1,T-1) , invalid_move(X,Y,T) ,
    poss_locate(X1,Y1,T-1) ,
    poss_locate(X,Y,T) , D=#abs(X1-X) + #abs(Y1-Y) .

```

```

distance(X,Y,T,1) :- valid_move(X,Y,T).

exists_better_move(X,Y,T) :- distance(X,Y,T,D), distance(X1,Y1,T,D1), D>D1.
best_move(X,Y,T)          :- distance(X,Y,T,D),
                           not exists_better_move(X,Y,T).

%%% Coherence %%%
is_passage(X,Y) :- passage(X,Y,R1,A1,R2,A2).
is_cell(X,Y) :- cell(X,Y,R,A).

p_support(X,Y,T,C) :- C = [sense(motion,X,Y,T), sense(distance,X,Y,T)],
                      poss_locate(X,Y,T).
coherence(X,Y,T,P) :- p_support(X,Y,T,C), P=(100*C)/2, C>0,
                      is_passage(X,Y).
coherence(X,Y,T,C) :- p_support(X,Y,T,C), C=0, is_passage(X,Y).
coherence(X,Y,T,100) :- p_support(X,Y,T,C), is_cell(X,Y).

most_coherent_time(T,M) :- M = #max [coherence(X,Y,T,C) :
                                     coherence(X,Y,T,C) = C], M>=0.
most_coherent(X,Y,T) :- most_coherent_time(T,C), coherence(X,Y,T,C).

%%% Generate positions %%%
max_rssi(X,Y,T) :- max_rssi_a(X,Y,T).
max_rssi(X,Y,T) :- max_rssi_b(X,Y,T).
max_rssi(X,Y,T) :- max_rssi_c(X,Y,T).
max_rssi(X,Y,T) :- max_rssi_d(X,Y,T).

max_data(X,Y,T) :- max_data_a(X,Y,T).
max_data(X,Y,T) :- max_data_b(X,Y,T).

static_in(T) :- not has_rssi(T), not best_data_a(T), not best_data_b(T).

poss_locate(X,Y,T) :- in_project(X,Y,T), not invalid(X,Y,T).
poss_locate(X,Y,T) :- sensed_data(X,Y,T), not has_rssi(T),
                    not invalid(X,Y,T).

1{person_in(X,Y,T) : poss_locate(X,Y,T)}1.

%%% Hard Constraints %%%
:- person_in(X,Y,T), not max_rssi(X,Y,T), not max_data(X,Y,T),
   poss_locate(X,Y,T), not static_in(T).

%%% Presence in a room %%%
loc_in(R,A,T) :- person_in(X,Y,T), cell(X,Y,R,A).

```

When plausible positions are not available for a few time stamps, the reasoning system should combine optimization criteria in a different way to complete the partial information over time.

In some cases, the system may need to rely on more sparse information to infer localization. In this case, it is computationally hard to consider all possible locations in terms of cells of the grid. However, commonsense reasoning can be used to characterize the current situation and use this inference results to localize the person in a given room. As an example, the inertia law on the relation *in()* can be used to detect where an object is at a given time, as shown in Listing 6.9:

Listing 6.9: Example of the use of inertia law

```

out (Obj,Room,T) :- in (Obj,R1,T) , R1!=R.
in (Obj,Room,T)  :- in (Obj,R,T-1) ,
                    not out (Obj,Room,T) .

```

In the simplest scenario, a person is supposed to interact with an object when the value of one attribute associated with the object changes. These changes are also propagated using the inertia law. Interaction with multiple objects can be consistently interpreted by generating all possible alternatives: the interaction with two distant objects at the same time makes it possible that the person is in both locations, unless one of the scenarios is supported by additional evidence; a consistency constraint can be used to specify that options should be treated separately (one location for each time step), as shown in Listing 6.10.

Listing 6.10: Consistency Constraint for Interaction with Multiple Objects

```

% inertia law
n_attribute_obj (Name,Obj,V,T) :-
    attribute_obj (Name,Obj,V1,T) ,
    V1!=V, range (Name,V) .

attribute_obj (Name,Obj,V,T) :-
    attribute_obj (Name,Obj,V,T-1) ,
    not n_attribute_obj (Name,Obj,V,T) ,
    attribute_def (Name,Obj) , time (T) ,
    range (Name,V) .

% possible locations
change_val (Obj,T) :-
    attribute_obj (Name,Obj,V1,T-1) ,
    attribute_obj (Name,Obj,V,T) , V!=V1 .

poss_in (R,A,T) :- change_val (Obj,T) ,
                  in (Obj,R,A,T) .

% generate possibilities

```

```
{ person_in(R,A,T) : poss_in(R,A,T) }.

% constraint on locations
:- person_in(R,A,T), person_in(R1,A1,T), A!=A1.
```

This support to localization is far less precise but can provide the system with plausible alternatives even when precise information from the bayesian filter are missing, such as during the night or when the person does not carry the localization tag. Situation assessment is even more crucial in these cases. Specific sequences of changes in the attributes of objects can help characterizing what is happening. As an example, a change in bed's load sensors followed by lights off in the bedroom may indicate that the person went to sleep, while a change in bed's and carpet's load sensors followed by switching the lights on in a room that is not the bedroom, indicates that the person got out of bed, as shown in Listing 6.11.

Listing 6.11: Example of Combination of Object Attributes

```
in_bed(T) :- person_in( bedroom , bed , T ) ,
             attribute_env( luce , bedroom , bedroom , dark , T ) ,
             attribute_obj( state , imud , off , T ) ,
             filteredLoad( bed , V , T ) , V > 0.

out_bed(T) :- person_in( bedroom , bed , T1 ) ,
             filteredLoad( carpet , V , T ) , V > 0 , T1 < T.

localized(T) :- person_in(R,A,T) .
awake(T)      :- not in_bed(T) , localized(T) .
awake(T)      :- out_bed(T) .
bad_sleep(T) :- attribute_obj( loadVolatility , bed , N , T ) , in_bed(T) , N != stable .

sleep_interrupt(T) :- in_bed(T1) , awake(T) , T1 < T.
back_to_bed(T)      :- sleep_interrupt(T0) , T0 < T1 , awake(T1) , in_bed(T) ,
                       T1 < T.

nightstart(T) :- in_bed(T) .
nightend(T)    :- sleep_interrupt(T) , T < T1
                 not back_to_bed(T1) , time(T) .
```

6.3.4 Discussion

The main contribution of this solution is that data are interpreted by a component that is able to draw from a knowledge database and make complex

inferences. Moreover, my approach to situation assessment has the following desirable characteristics:

Declarative Formalism. The declarative nature of the formalism makes the problem specification readable and easily adaptable to an assortment of situations: thresholds can be adapted to a particular setting, as well as the combination of criteria used to support localization can be targeted to the reliability of data sources;

Abstract Representation of Data. The abstract representation of collected data as properties of entities and relations among entities makes the model independent from the specific sensors used;

Expressivity. The expressivity of the modelling language and the computational efficiency of the inference engine put the basis for the application of our solution to more complex tasks including tracking multiple objects, real-time monitoring, planning and scheduling.

This modelling approach looks similar to what can be obtained by using an ontology [89], with the difference that the ASP reasoning enhances readability, expressivity and computational efficiency of the model. I am aware of the fact that research efforts are converging toward the combination of non-monotonic reasoning and ontology-based knowledge representation, but available implementations are still domain dependent and formal issues need to be further explored. For this reason, I decided to encode the contextual information directly into first order logic under the ASP framework. The domain description and inference rules can be easily combined with domain knowledge extracted from an (existing or new) ontology if needed.

The reasoning processes for context interpretation and evaluation takes advantage of the non-monotonic nature of ASP. If we consider reasoning support to localization, results are obtained quite fast (less than one second) on a grid of 50×50 cells of 25×25 centimeters, along 500 time steps. Correctness of the solution has been tested in a simulation environment: we generate ad-hoc sensor data by simulating the behavior of the patient with an agent simulator (see Section 6.2) and create specific situations that test the correctness of the inference with respect to situation assessment, such as localization.

Several tests have been done, evaluating the ASP programs by using *Gringo* [79] and *Clasp* [82], which supports constraints, choice rules and weight rules [83]: a location on the grid is provided 90% of the time even when the information from the sensors is conflicting and partial (such as when neither possible locations nor associated likelihood values are available), and the average distance of the solution from the real position is between 25 and

50 centimeters (one cell in my setting).

However, the computational complexity grows when the size of the ground instance becomes bigger, for example by incrementing the size of the grid or the number of time steps. In this case the bottom-up grounding of the experimental grounder *bingo* [90] can be an interesting solution to limit the size of the grounded program, while using the incremental version of the *Clasp* solver, *iclingo* [91] one can obtain faster results in computing state evolutions when some elements of a state at time i depend on the state at time $i - 1$.

Another interesting aspect is related to a more complex class of reasoning tasks, which include reasoning about sequences of states over time steps, such as for planning and scheduling. For these classes of problems, temporal constraints may negatively affect the size of grounding. To overcome this problem and maintain efficiency, one can take advantage of the integration of ASP reasoning with constraint solving techniques [92].

Measuring the accuracy improvement of situation assessment using logic-based reasoning techniques is difficult and depends on the correctness of the simulation or on the conditions of the real scenario. However, my work represents a preliminary attempt in this direction and it shows that it is worth considering the combination of statistic-based techniques and expressive logic-based inference to successfully support situation assessment.

6.4 Chapter Summary

This chapter is the logical (pun intended) extension of the previous chapter. It shifts the focus from mere target tracking to “situation assessment” where the system brings to bear a number of different and extremely varied sets of information and processes them within a logical framework instead of a statistical one. As a matter of fact, the boundary from statistical to logical is one of the issues of this chapter and the reader will have noticed that there is a level at which both techniques are suitable.

When the complexity and uncertainty of the data raises again the logical framework becomes the preferred one also because the benefits it yields extend beyond tracking into a full new understanding of what the target(s) is doing. At the same time, this chapter shows that the statistical techniques are the necessary underpinning of the tracking process because of their efficiency.

Chapter 7

A Case Study, the SINDI System

In this chapter, I will describe the SINDI System, a case study for the data-gathering, tracking and situation assessment techniques previously described.

SINDI is an intelligent home healthcare system characterized by a wireless sensor network and a reasoning component. The goal of the system is to allow constant and unobtrusive monitoring of a patient in order to enhance autonomy and increase quality of life. Data collected by the sensor network are used to support a reasoning component, which is based on Answer Set Programming, in performing three main reasoning tasks: (i) continuous contextualization of physical, mental and social state of a patient, (ii) prediction of possibly risky situations and (iii) identification of plausible causes for the worsening of a patient's health.

Starting from different data sources (sensor data, test results, inference results) the reasoning component applies expressive logic rules aimed at correct interpretation of incomplete or inconsistent contextual information, and evaluates correlation rules expressed by clinicians. The expressive power of ASP allows efficient enough reasoning to support prevention, while declarativity simplifies rule-specification and allows automatic encoding of knowledge. Preliminary evaluations show that the combination of an ASP-based reasoning component and a wireless sensor network is a good solution for creating a home-based healthcare system.

The chapter is organized as follows. After an introduction to the SINDI system and its system requirements in Section 7.1, The reasoning framework is formally described in Section 7.2, while details about the implementation of the Knowledge Representation and Reasoning Model are presented in Sec-

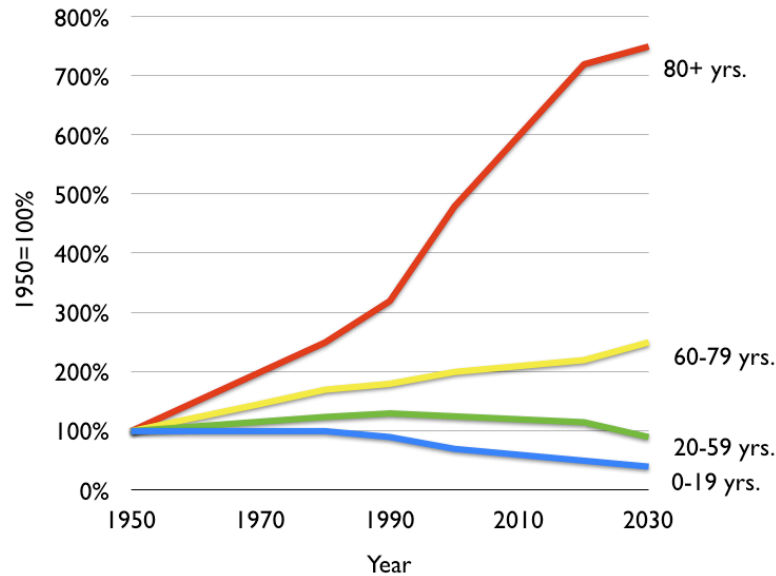


Figure 7.1: Size of the Italian population by age groups: each age group is separately plotted and normalized to the group size in 1950 (Source ISTAT and IRP-CNR).

tion 7.3. In conclusion, Section 7.4 gives a preliminary evaluation. Some of the material in this chapter has been published in a paper in the *Computer Journal* [93].

7.1 Introduction

In many countries the ratio between the number of old and young people is constantly growing. Figure 7.1, drawn from data on the Italian population, shows that the percentage of people “over 80” relative to the total population might increase eight-folds from 1950 to 2030. This means that the number of old people will be a sizeable percentage of all the population. We want for these people to have a very good quality of life while keeping the expenses as low as possible. In many cases this means helping people stay home as long as possible (no more than 5% of the elderly in Europe are institutionalized).

The system we are building is targeted to raising as much as possible

the age at which a person needs to be institutionalized. We believe that constant monitoring through pervasive technologies is essential to provide a more efficient health assistance at home. In fact, recent studies about the acceptance of technologies for the elderly [94, 95] show that while people tend to look for social relationships in activities such as cleaning or playing cards, in situations related to safety, health and personal care they are likely to rely on technological solutions.

For this reason there has been a strong development of computer technologies applied to specific fields of medical science in order to allow the delivery of clinical care outside of hospitals. For example, Telemedicine and Clinical Decision Support Systems have been used to collect complex clinical data and implement diagnosis at-a-distance. The practical use of these techniques in real contexts has shown that they work well for some very specific healthcare applications, such as medical prescriptions [96] or real-time transmission of clinical data. Our system complements these techniques by taking into account the contextual setting and the health evolution of patients over long periods of time.

In our study we address those elderly that are clinically stable although they might be affected by chronic diseases and physical decline (more than 90% of the population over 65 has more than one chronic disease). Since their health conditions do not require constant monitoring of complex biomedical parameters, these patients do not need, and are less tolerant of, invasive sensors.

Many user-centered systems that analyse user's behaviour and detect emergencies have been developed. They often cater to the identification of predefined patterns of behaviour rather than to the assessment of health in general, and they are mainly based on statistical analysis of data, thus needing substantial training to be adapted to a particular patient.

We use monitoring to support prevention, causal diagnosis and emergency detection in the same framework and provide a global representation and reasoning model for general health assessment, combining medical knowledge, patient's clinical profile and context evaluation through sensor data. These data are combined and interpreted by an inference engine to help caregivers detect patients' physical, mental and social status as it evolves.

Figure 7.2 shows a very high-level overview of the architecture of our system and the correlation between its components.

The presence of heterogeneous information makes it possible to both automatically adapt the results of the reasoning process when new information

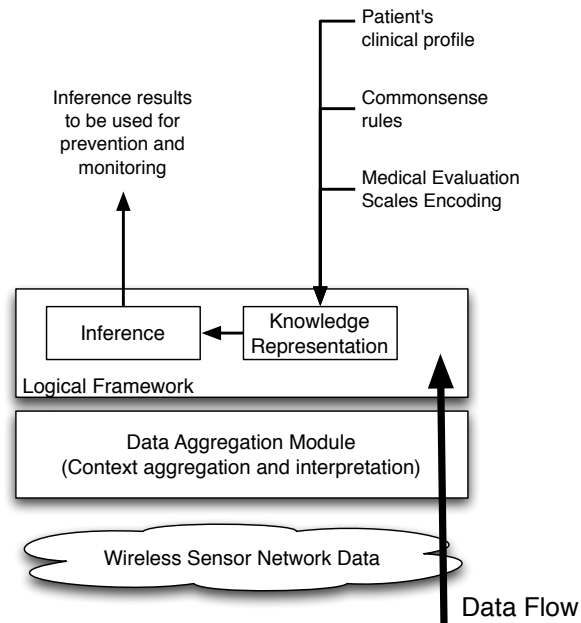


Figure 7.2: Data flow.

is available and deal with user and context-specific constraints. We use Answer Set Programming (ASP) because it constitutes a powerful declarative framework for knowledge representation and reasoning in this application context. ASP addresses many of the requirements listed in Subsection 7.1.1.

We do not focus on the use of robots for healthcare because, beyond their high cost of set up and maintenance, their presence is rather intrusive and the help they can really provide is marginal.

Starting from these considerations, we have developed the first prototype of the SINDI (Secure and INdependent Living) system. The principal requirements of the system and the way we address them are illustrated in Subsection 7.1.1.

7.1.1 System Requirements and Design Issues

As previously mentioned, the SINDI system has been designed to support caregivers in monitoring and providing health assistance to the elderly in their home environment by using wireless sensor technologies and automated reasoning capabilities, but it also interacts with the elderly to help them directly. The main goal is not to extend life but to enhance autonomy and

increase quality of life.

For this reason, the main requirements we have been focusing on are:

- *unobtrusiveness*: the monitoring system should not affect the lifestyle and habits of the person being monitored;
- *technological soundness*: the monitoring system should use what is already commercially available with respect to technology;
- *affordability*: costs should be kept low in order to be afforded by medium-income families; set-up and management should also be easy and cheap;
- *easiness*: the elderly may have problems in handling complex multiple devices, therefore the interface with the system should be as close as possible to what they are used to, and the interaction should be intuitive;
- *medical soundness*: though SINDI does not support complex medical diagnosis of specific diseases, the intelligent support should take into account the appropriate medical knowledge;
- *user-centrality*: each person has different needs, thus a system which is supposed to work in a specific home environment should consider the psycho-social and clinical setting of the patient when evaluating the evolution of his health status, rather than mapping his situation to similar medical cases;
- *adaptivity*: to address user-centrality, the system should incorporate mechanisms to adapt to different patients, both automatically and by explicit parameters setting;
- *context-awareness*: sensing activity and reasoning support should consider not only static user-specific needs, but also the evolving state of the patient and of the environment in order to give more accurate results when data are incomplete and dynamic;
- *reactivity*: for a long-term monitoring support, data manipulation and interpretation can be done offline; however, the system should be able to react in real time to specific triggers (such as emergency situations, user input, system feedback);
- *reliability*: collection and aggregation of data, as well as results of reasoning used to help caregivers in understanding patient's health evolution should be reliable enough to assure adequate support;
- *accuracy*: results of the reasoning process should be as close as possible to what caregivers expect, according to the available information.

In order to address these requirements, some technical and methodological choices turned out to be crucial in the design and implementation of the

SINDI system:

- To preserve unobtrusiveness, we decided not to use cameras in order to avoid the uncomfortable feeling of being constantly observed. Dynamic data about the person and the environment are unobtrusively captured by a Wireless Sensor Network (WSN), composed of several sensor nodes, a wearable monitoring device and a master processor.
- The use of commercial nodes readily available on the WSN market increases considerably the affordability of this kind of system, helping at the same time the reduction of overall costs and simplifying configuration and deployment.
- Although the elder generation is getting closer and closer to technology, they might have problems in dealing with complex devices such as a PDA or devices that force them to read from a small screen, such as portable phones. For this reason, SINDI allows interactions with the patient through the TV screen, controlled by a device that is similar to a TV remote.
- Context-awareness is another important requirement. The reasoning has its basis on the aggregation and interpretation of different kinds of information from heterogeneous sources (such as light, position, movement, localization, load cells). The idea is that additional information can help in characterizing the solutions of the reasoning process, identifying the most plausible ones, according to the available domain knowledge. This also enhances reliability since heterogeneous sources of information that can be interpreted may help in compensating errors and incompleteness of data.
- If and when new sensors are available, the information they produce can be easily taken into account by adding new rules.
- The need of making the system user-centered and medically sound lead us to include some medical knowledge in the reasoning phase. In this way it is possible to trace general habits and their correlation with the patient's well-being according to the evaluation methods of clinical practice. In particular, we want to address the fact that clinicians need to be supported in:
 1. understanding patients' physical, mental and social settings as they evolve,
 2. predicting what could follow with respect to particular changes in one or more aspects of patients' general health state and
 3. identifying correlated aspects that may be the cause for a negative change in the patient's general health state.

The first aspect is related to the contextualization of worsenings of the

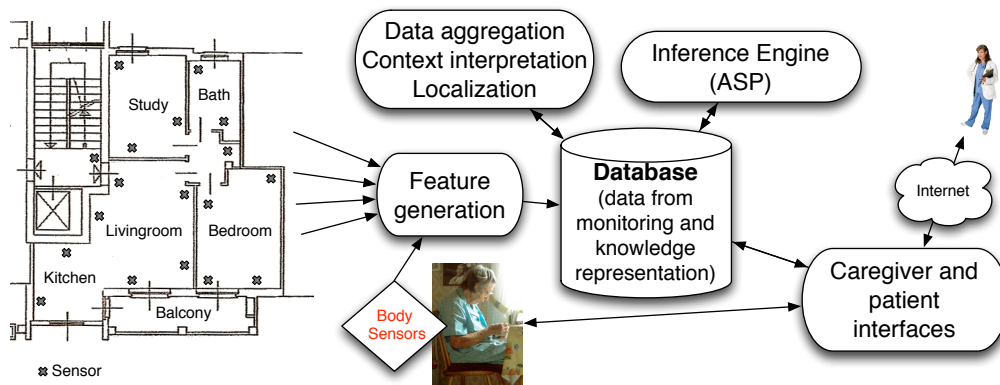


Figure 7.3: Architecture overview.

general health status of the patient, not only with respect to similar clinical cases, but giving more importance to aspects that turn out to be important for the patient at hand. The second aspect refers to prediction, i.e. the identification of health-related aspects (we refer to them as *items*) which deserve specific attention regarding to a worsening; this makes it possible to act *before* more serious side-effects are observed and to plan appropriate short- and long-term interventions, thus reducing risks. The third aspect is more similar to diagnosis, but it should be a *local* process rather than a case-based one, in that it must take into account patient's clinical and environmental settings and adapt to the specific patient.

- To perform these reasoning tasks and encode the relative knowledge into a common model, we believe ASP is the right framework because:
 1. the effectiveness of the implementation makes it possible to express deductions, default reasoning, constraints, choices and qualitative preferences;
 2. declarativity allows the automatic encoding of medical knowledge, thus making the system easily extensible and medically sound;
 3. the use of contextual information and the way new knowledge can be taken into account, makes it possible to deal with incomplete information and enhance context-awareness.
- Medical soundness and context-awareness can help obtaining better solutions because the combination of different sources of information (sensors, medical knowledge, clinical profile, user defined constraints) that change over time make the system more reliable (i.e. much better able to disambiguate situations, thus reducing false positives) and

adaptable (easily extended on the face of new available information).

- We also considered the fact that the reasoning process is run not only periodically but also according to specific triggers. These triggers can be associated to states of emergency and specific actions to be performed by the system. By adding appropriate logical constraints to the ASP program, emergencies can also be contextualized in almost real-time. This behaviour addresses reactivity.

The general architecture of SINDI and its main components are illustrated in Figure 7.3.

7.2 Knowledge Representation of the Home Healthcare Domain

A careful analysis of health care in home settings suggests that health-related items can be classified into three levels: Functionality level representing functional disabilities of the person monitored, Activities of Daily Living (ADL) level representing her dependence in performing daily activities, and Risk Assessment level characterizing risky conditions. Significant aspects of health assessment at each level (referred to as *items*) have been identified according to the medical practice in health assessment of the elderly [97] and encoded in our declarative framework.

A lower layer (State level) contains aggregated context data as well as static and dynamic evaluations of significant aspects of patient's clinical settings (referred to as *indicators*).

Static aspects of clinical profiles include stable pathologies characterizing elderly care and drug intake as well as results of specific complex tests performed periodically by the caregiver. Predicates used in logic rules to represent the static profile are illustrated in Table 7.1.

Dynamic aspects of the clinical profile include:

- indicators that can be evaluated through ad-hoc tests proposed by the system when needed;
- indicators evaluated by the WSN through continuous monitoring and data aggregation;
- indicators evaluated through logic rules (e.g. the quality of sleep).

Every time the inference process is run, the system compares values of in-

Table 7.1: Logical encoding of patient's profile.

Predicate	Description
test(Name,Value)	test results
drug(Name)	list of drugs
pathol(Name)	list of pathologies
profile(X,Name,V)	X={drug,pathol} V={yes,no}

Table 7.2: Logical encoding of dynamic profile.

Predicate	Description
lev(L,I)	association items-level
obs(I, V_i ,T1,T2)	evaluation of an item
obsind(Ind, V_i ,T1,T2)	evaluation of an indicator
link(I,Ind)	association item-indicator
range(Ind, V_i)	range of values
ord(Ind, V_i ,Num)	order of values

dicators from the previous inference with the actual values (either aggregated by the sensors or inferred by logic rules) and computes differential evaluations. Admissible values for each indicator are part of the medical knowledge and are encoded by the knowledge engineer.

At higher levels, each indicator can be associated with one or more items. Results of differential evaluations identify which indicators are subject to worsenings and which items are critical: the higher the number of worsenings associated with an item, the more the item is critical.

Predicates used in logic rules for evaluation are listed in Table 7.2, where I represents an item, L is a level, V_i are values, Ind represents indicators and $T1$, $T2$ are timestamps. Logic rules used to detect worsenings by using differential evaluations are presented in Section 7.4.

We want to point out that evaluations of items obtained by the system

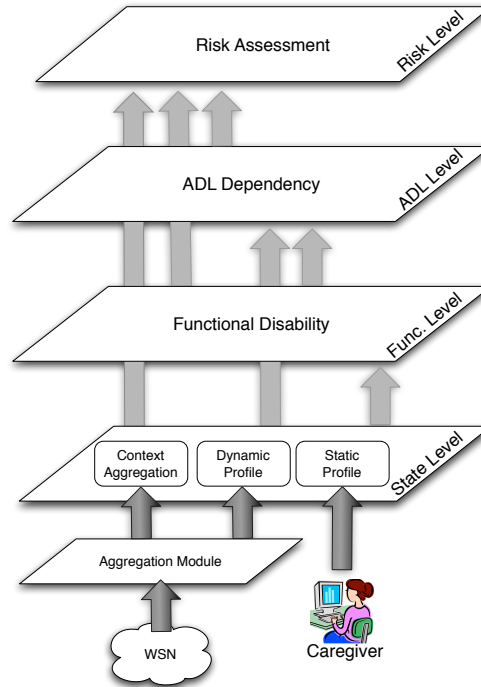


Figure 7.4: Information flow across levels.

through specific tests are only partial. In fact, computer-aided tests can include only a few of the aspects considered in the complete initial evaluation performed by the caregivers. The frequency with which these tests are proposed to the user depends on test scheduling done by the caregiver. As an example, if the quality of sleep gets substantially worse, this can raise a trigger and the system proposes a computer-aided version of the appropriate test; the worsening can also be observed by the caregiver that remotely schedules the test for the patient.

The reasoning process takes also into account medical knowledge about causal correlations between items, and combines them with results of differential evaluations to show how patient's health can evolve in terms of functional disability (Functionality level), dependencies in performing daily activities (ADL level) and risks assessment (Risk Assessment level).

Figure 7.4 illustrates the flow of information across levels: details about the person and the environment at the state level are provided by the WSN (eventually aggregated), while values of items at upper levels are hierarchically influenced by values of items at lower levels.

In the following subsections we give details about items at each level and indicators (at the state level) associated to them. Details about causal correlations and reasoning capabilities are presented in Section 7.3.

State Level The *state level* includes static and dynamic profiles as well as context-dependent information, which are the results of the aggregation of sensor data. The static profile includes:

- complete test results evaluated by the caregiver on periodic examinations using appropriate medical scales: cognitive state (Mini Mental Test [98] and Clock Drawing Test [99]), vision (optical tests), mobility (Tinetti-POMA scale [100]), affective state (GDS test [101]), nutritional state (Mini Nutritional Assessment [102]) and ADL dependency (Katz scale [103]);
- intake of drugs among which we consider ache inhibitors, benzodiazepins, psychotropes, neuroleptic and anti-parkinson;
- presence of specific age-related pathologies among which we consider reduction in visual acuity, hearing loss, osteoarthritis, cognitive decline, depression, alcoholism, vascular pathologies, arthritis, cardiac problems, parkinson, epilepsy, dentistry problems, disthyroidism, acute pathologies;

Interesting pathologies and drugs, as well as valid tests, have been identified together with the geriatrics of the S. Gerardo Hospital in Monza. The declarative nature of the ASP framework makes it easy to add new information. However, we are aware of the fact that a user-friendly interface should be available for clinicians to extend the medical knowledge of the system without the constant support of a knowledge engineer. We believe this is feasible since declarativity allows automatic encoding from a high level specification into ASP, through appropriate mapping. We are investigating this issue.

Context-dependent information includes:

- Personal details: biomedical parameters such as temperature and weight;
- Environmental properties: average light value, humidity, temperature, architectural barriers (such as presence of stairs or carpets in a given room or area of interest);
- Basic activities: movement activity can be easily captured by the wearable sensor, and we consider it as characterized by *motion* (walking, standing still), *position* (sit, lay, stand) and *orientation* (straight, turning);

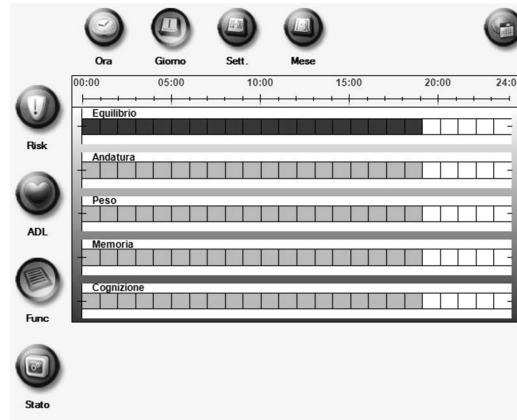


Figure 7.5: Caregiver interface to access context data.

- Localization: the way the patient moves from one room/area to another is traced;
- Interaction with objects: we consider two kinds of binary interaction according to sensors associated with the specific object, i.e. *pressure* (chair and bed objects) and *switch* (doors, windows and devices).

Context-related data can also be used to infer values of those indicators that are not directly available from aggregated sensor data. As an example, consider the indicator *quality of sleep*. To understand the quality of sleep it is necessary to reason about the night activity, taking into account consistent contextual information (location, state of objects, movement) and some auxiliary predicates (such as start/end of night time, getting up, going out of bed). A simplified version of the ASP code to infer quality of sleep is shown in Section 7.4.

Consistent interpretations of the context can also be crucial for caregivers in order to investigate the particular settings in which worsenings or emergencies are detected, since these values can be analyzed through a visual interface similar to the one illustrated in Figure 7.5.

Items at each upper level are characterized by an initial static evaluation (predicate *obs()* in Table 7.2), and a set of indicators used for differential evaluation (predicate *obsind()* in Table 7.2). As an example, a complete test performed by the caregiver is encoded by an *obs()* predicate, while partial (computer-aided) versions of the same test are encoded as indicators using *obsind()* predicates.

In the following subsection we identify preliminary tests used for initial evaluation and we give additional details about the association of indicators to items.

Functionality Level At this level, the system considers the following functional disabilities:

- balance and gait, initially evaluated through the appropriate parts of the Tinetti-POMA medical scale; indicators are represented by aspects of the scale that can be captured and evaluated through the wearable sensor, i.e. standing, sitting, turning and walking;
- nutrition, initially evaluated with the Mini Nutritional Assessment test; the indicator is the Body Mass Index (BMI);
- vision, initially evaluated through specific optometric tests; indicators are the level of light during the day and at the sunset (according to medical practice, keeping the light on when the shadows are opened and the natural light coming from the outside is up to a certain level, may indicate a problem);
- hearing sensibility, evaluated through audiometric tests; indicator is the reaction time to a ringing bell;
- mental and cognitive capabilities, evaluated through the Mini Mental and Clock Drawing tests; indicators are represented by a computer aided questionnaire, counting ability and quality of sleep;
- insomnia, evaluated through a questionnaire; indicator is the quality of sleep;
- emotional stability, initially evaluated through the GDS test [101]; indicator is the computer-aided version of the GDS test.

According to the literature, mobility remains one of the most important aspects to be assessed in order to protect the elderly from the negative consequences of a fall. The Tinetti-POMA test has been claimed to be the gold standard in assessing mobility dysfunctions in the elderly and an important fall risk assessment measure. This test was published by Tinetti in 1986 and it has been designed to evaluate the position changes and gait maneuvers used during normal daily activities.

With respect to complete tests and computer-aided versions of them, a concrete example is represented by the Mini Mental and Clock Drawing tests. These tests are not easily performed without the assistance of a person. Thus, the evaluation of the correspondent indicators is done by periodically (once a month) proposing a reduced version of the test to the user via TV screen: the Abbreviated Mental Test (AMT) consisting in a small set of questions [104].

Results of these simple tests are stored in the DB and they can be accessed by the caregiver at any time.

ADL Level At this level we consider the Activities of Daily Living (ADL) as evaluated in the Katz scale, in particular:

- transfer (mobility) has the same indicators as balance and gait;
- dressing has the same indicators as balance and visual functionalities;
- feeding has the same indicators as nutrition functionality;
- bathing has no indicators in the current version;
- toileting has no indicators in the current version.

ADLs that are not associated with any indicator cannot be evaluated to identify worsenings. For this reason, only prevention is possible, based on correlation rules (see Section 7.3 for details).

Mobility is evaluated through the Tinetti-POMA scale while other ADL's are evaluated according to the Katz index. We want to point out that reasoning at this level is not aimed at activity recognition through the identification of patterns of behaviour, as in other approaches to monitoring [105]. We rather concentrate on possible inter-dependencies that may arise in performing ADLs', according to correlations with items at other levels, because this is useful for prevention.

Instrumental Activities of Daily Living (IADL) from the Lawton scale [106] have not been included. This choice has been guided by the fact that their impact on other health-related items is less determinant and the evaluation with the state of the art sensor technology is too complex to be performed in a non intrusive way.

Risk Assessment Level Risks are identified by the potentially most dangerous situations for elderly people at home, namely:

- risk of falls, initially evaluated through the Tinetti-POMA scale; it has the same indicators as balance and gait functionalities;

- risk of depression, initially evaluated through the GDS scale; indicators are those of nutrition, balance, gait and sleep functionalities;
- frailty, initially evaluated through a combination of GDS test, Mini Mental Test and Katz evaluation; indicators are the same as those of nutrition, balance, gait, vision and emotional functionalities plus some additional ones like walking speed, age, number of pathologies, number of drugs and number of activities in which the patient needs help;
- risk of dependency, evaluated through the Katz index; it has the number of ADL's that cannot be easily performed as indicator;
- malnutrition, evaluated through the Mini Nutritional Assessment test with BMI as indicator;
- isolation, having the number of visits and the time spent out of the house as indicators.

7.3 The Reasoning Capabilities

In the knowledge representation model of SINDI we describe a home health-care scenario by a declarative representation of the domain at different levels.

At the state level, data provided by the sensors network can be noisy and imprecise, even after aggregation. SINDI uses the logic-based situation assessment techniques described in Chapter 6 in order to disambiguate unclear situations as much as possible.

At upper levels, inference is performed by separate logic programs in order to detect:

1. functional disabilities, every hour;
2. dependencies in performing Activities of Daily Living, every day;
3. risk assessment, every day.

Besides domain knowledge and consistent interpretation of the context, two more aspects are necessary in order to reason about the health status of the person monitored: *differential evaluations* and *correlation rules*.

Differential evaluation of an item I at level L through the indicators Ind_i is possible by comparing the value V_i^0 of each associated indicator at the beginning of the previous inference (time zero) and the (eventually aggregated) value V_i^1 of the same indicator at the time interval being evaluated (time 1).

For some indicators such as standing and sitting, several evaluations may be available for the time interval (hour or day) considered in the inference

process. Given that a single value has to be provided for each indicator in a given interval, the data extraction module taking data from the database and passing them to the ASP engine is in charge of computing the most frequent value for that interval. This choice can be motivated by the fact that the slow trend of physical and cognitive decline makes evaluations uniform in a short period of time such as an hour or a day, and isolated values that are far from the most frequent one can be due to occasional awkward movements rather than to a disability.

Though differential evaluations can also indicate improvements, we only consider worsenings, as they are much more relevant with respect to risk prevention. The reasoning system can be extended to consider also health improvements and use them to evaluate response to medical treatments.

Worsenings can be detected applying the following logic rule:

```
worse(L, I, Ind, T1, T2) :- obsind(Ind, V, T1-1, T1),
                           obsind(Ind, V1, T1, T2),
                           link(I, Ind), lev(L, I),
                           ord(Ind, V1, N1), T1 < T2,
                           ord(Ind, V, N), N < N1.
```

Correlation rules concern dependencies between a cause (I_1) and an effect (I). Different dependencies are allowed:

- only negative influence of an item I_j on another item I_k ;
- only positive influence of an item I_j on another item I_k ;
- directly proportional influence of an item I_j on another item I_k ;
- inversely proportional influence of an item I_j on another item I_k ;

Each of these correlations can be *strict* or *possible*. In the first release of the system we concentrated on strict and possible negative influence, since they are more significant for prevention and diagnosis. All other dependencies can be introduced and encoded in the system in a similar way, and we are considering this issue in the implementation of the following prototype of SINDI.

Correlation rules can be specified by clinicians and automatically mapped into ASP to express negative/possibly-negative influence of an item I_k on another item I_j , respectively encoded into logic predicates:

```
neg_influence(I_j, I_k).
poss_neg_influence(I_j, I_k).
```

Consider the structure of items and correlation rules as an oriented graph stratified into levels: items are nodes and correlations are oriented arcs connecting two nodes.

Each item I_j at a level L , $I_j(L)$, can be connected to another item $I_k(L')$ in two different ways:

- oriented arc from $I_j(L)$ to $I_k(L')$: if $I_j(L)$ gets worse, this has negative influence on $I_k(L')$;
- oriented dotted arc from $I_j(L)$ to $I_k(L')$: if $I_j(L)$ gets worse, this *may* have negative influence on $I_k(L')$.

In addition, the layered structure is used to avoid possibly infinite propagation of dependencies when the reasoning process investigates the search space.

As already mentioned, the inference process considers items at each level separately. No matter which level is being evaluated, the system first characterizes every item in the graph as being either *stable* (none of the indicators become worse for that item) or *unstable* (one or more indicators become worse for that item in the interval being evaluated). When at least one indicator becomes worse, the item is marked as unstable, otherwise it is stable:

$$\begin{aligned} \text{unstable}(L, I, T1, T2) & :- \text{worse}(L, I, \text{Ind}, T1, T2), \\ & \quad \text{lev}(L, I). \\ \\ \text{stable}(L, I, T1, T2) & :- \text{not unstable}(L, I, T1, T2), \\ & \quad \text{lev}(L, I). \end{aligned}$$

This distinction is crucial to determine the behaviour of the system when it reasons about each $I_j(L)$ at the specific level L :

- a) if $I_j(L)$ is *stable*, the system performs the following reasoning task:
 - makes predictions about the amount of risk for $I_j(L)$ to get unstable, as follows:
 - * investigates the direct connections determined by correlations rules, to identify items $I_k(L')$ that may influence $I_j(L)$;
 - * check each $I_k(L')$ to see whether it is unstable and, in this case, conclude that $I_j(L)$ can be at risk due to its correlation to $I_k(L')$;
- b) if $I_j(L)$ is *unstable*, the system performs three different reasoning tasks at once:

- identifies possible negative effects of the worsening of $I_j(L)$ on other items $I_k(L')$ according to correlation rules, represented by oriented arcs from $I_j(L)$ to $I_k(L')$ ¹;
- performs *local* diagnosis, detecting possible alternative causes of the worsening of $I_j(L)$ among the correlated items that have been marked as unstable;
- contextualizes the worsening of $I_j(L)$ providing alternative health-related contextualizations and values of items (both stable and unstable ones) included in each separate contextualization scenarios.

Note that diagnosis and contextualization reasoning tasks may generate alternative solutions. This is due to the fact that two items I_1, I_2 that are related to the item being investigated I , can be in the same path backwards from I or not. In the second case, they are part of two different solutions.

Just as an example, we report part of the encoding used for prediction of functional disabilities, with respect to a stable function F . This corresponds to point a) in the description of the algorithm. Logic rules make it possible to predict negative effects of the worsenings of I on function F :

```

poss_neg_pred(func ,F,L,I ,T1,T2) :-                                     % Prediction
    stable(func ,F,T1,T2) ,
    poss_neg_influence(F,I) ,
    unstable(L,I ,T1,T2) .
neg_pred(func ,F,L,I ,T1,T2) :-
    stable(func ,F,T1,T2) ,
    neg_influence(F,I) ,
    unstable(L,I ,T1,T2) .
```

To conclude this section, we present a simple example on how the reasoning tasks can support clinicians in health assessment of the elderly.

Example 7.3.1 Consider the graph in Figure 7.6 (a). Suppose that the reasoning system is investigating ADL dependencies (level 2) and items are stable at this level (light grey nodes). Suppose also that visual functionality is

¹Propagation of negative effects are not considered since the layered structure of the graph lets us identify them simply by investigating results of the inference for $I_k(L')$ when items at level L' are evaluated.

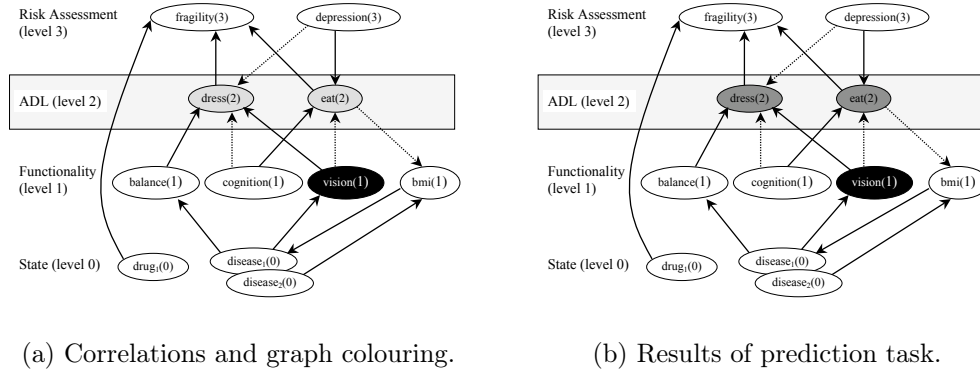
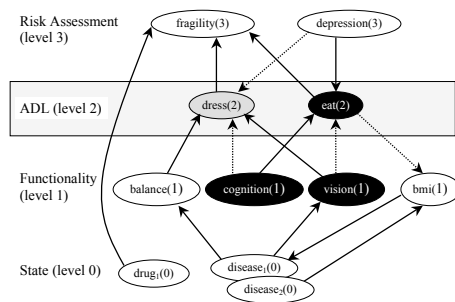


Figure 7.6: Example 7.3.1. ADL dependencies, inference process with respect to prediction.

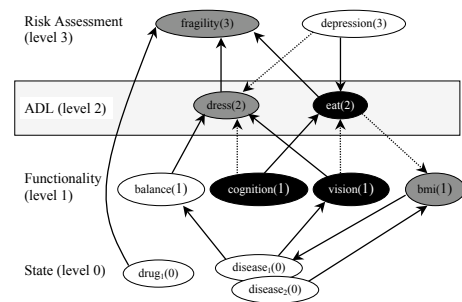
marked as unstable (black node). Results of the inference process with respect to prediction are illustrated in Figure 7.6 (b): ADLs' dress and eat are both at risk (dark grey nodes) due to the visual functionality, but in one case (for the ADL eat) the risk is only possible. This qualitative interpretation allows the association of worsenings to priorities and guides caregivers in planning interventions.

Example 7.3.2 Suppose now that, in the same setting, the initial evaluation of ADL dependencies is the one depicted in Figure 7.7 (a): ADL eat is unstable (i.e. there is an increasing level of dependency in performing eat), and visual and cognitive functionalities are unstable too. The inference process returns the following results:

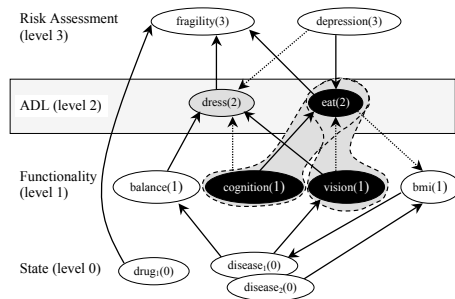
- prediction: risk of fragility and nutrition functionality (represented by the BMI) have to be monitored carefully since they may get worse due to dependency in performing eat (Figure 7.7 (b)); the system also returns that ADL dress is at risk due to cognitive and visual functionalities;
- local diagnosis: the increased level of dependency in performing eat can be due to a functional disability in vision and/or cognition (Figure 7.7 (c)); these possible explanations are treated as separate solu-



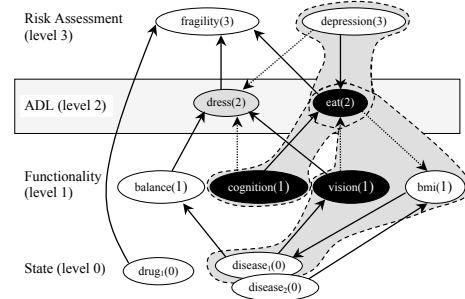
(a) Correlations and graph coloring.



(b) Results of prediction task.



(c) Results of diagnosis task.



(d) Results of contextualization task.

Figure 7.7: Example 7.3.2. ADL dependencies, inference process with respect to prediction.

tions since following a path from eat backward, leads to either one or the other functionality;

- contextualization: following the links backwards from the unstable ADL eat, three contextualization sets are identified for its worsening, as shown in Figure 7.7 (d); values of items and related indicators are provided to clinicians through appropriate interfaces, helping to identify the most plausible alternative contextualizations of the worsening of ADL eat (Figure 7.7 (d)) according to the context.

It is easy to figure out how, in a more complex schema of dependencies among items, loops and multiple paths can make reasoning a hard task. For this reason we decided to make the graph structure hierarchical, in that reasoning tasks for items at one level are performed separately from the reasoning tasks for items at other levels. This distinction is both conceptual and temporal, since inference is run every hour for functional evaluation and every day for ADL dependency and risk assessment. In dealing with complex graphs of dependencies, the expressive power of ASP can be crucial since it makes it possible to explore complex search spaces maintaining the computational complexity rather low.

7.4 Evaluation

The evaluation of an assisted-living system like SINDI is very difficult because it is hard to identify the correct metrics. The need for a common framework to identify the challenges and to suggest the metrics is clear. The work described in [107] proposes an *evaluation framework* for assessing the quality of assistive environments. This framework identifies a set of attributes that are considered critical for user adoption. The categories identified are the following:

- functionality (correctness, robustness - errors and faults, reliability - time of continuous operation);
- usability (ease of use, non obtrusiveness, accessibility);
- security and privacy (such as access modes and encryption);
- architecture (modularity, interoperability - standard interfaces to integrate components);
- cost (installation, maintenance);

This framework could be improved by using separate evaluation metrics for users and technical experts.

So far we have run the full system for short periods of time (days) in a mock-up environment without real users. Non-obtrusiveness stems from the design of SINDI. Details about the user and the environment are automatically collected by the WSN, and no complex statistical information or specific medical knowledge are needed to analyze possible evolutions of patient's health and to support understanding. Moreover, the interaction of SINDI with the patient and the caregivers is fully intuitive, as we deal with multimedia contents and the patient is provided with a device that looks like a remote control. Security and privacy are guaranteed by the use of security standards and techniques. Modularity and computational efficiency stem from the declarative nature of ASP and the availability of efficient solvers. Finally, the use of off-the-shelf components in SINDI considerably reduces overall costs.

In the first testing environment of our system we evaluated ASP programs by using *Lparse* as grounder and the *Clasp* solver [82, 79] as inference engine.

The Clasp solver supports constraints, choice rules and weight rules [83] and it can solve complex reasoning tasks very efficiently due to the heuristics used, combining ASP expressivity with boolean constraint solving.

In the testing phase of SINDI, we used Clasp both to generate the backlog (a few months of data) and to test the global performance of the system.

We did some tests on randomly generated instances with 20 to 70 correlation rules and 10 to 50 items, obtaining results in 0.65 to 2.75 seconds, once state level data had been aggregated and interpreted correctly. The worse cases have been observed for instances where the number of correlations was more than 6 times the number of items. These results are due to the high number of bidirectional correlations among items, derived by the random generation of instances. According to geriatric practitioners, similar cases are not common in real settings and, except for those instances, the reasoning process scales well. Time of execution on sample instances are summarized in Table 7.3.

Context aggregation and interpretation at the state level remains the harder task, since it requires to analyze up to 24 hours of data when reasoning about ADL dependency and risk assessment is performed. In evaluating indicators, delegating part of the aggregation process to the WSN nodes lowered the computational time up to 60 per cent for instances of medium complexity (i.e. for a person that is active from 30 to 40 per cent of the time

Table 7.3: ASP reasoning performance (time is expressed in seconds).

Items (num.)	State Level (time) WSN/ASP vs. ASP	Correlations (num.)	Upper levels (time)
20	140.05 vs. 205.07	30	0.69
		70	1.01
		130	2.03
30	169.71 vs. 481.22	120	1.03
40	183.48 vs. 487.17	160	1.11
50	201.32 vs. 589.03	200	1.37
70	241.11 vs. 603.16	250	1.68

in a day).

These computational costs do not apply to situations in which emergencies arise, since they are detected almost immediately by triggering events and managed by evaluating appropriate integrity constraints.

7.5 Chapter Summary

This chapter has been of a rather different kind when compared with the previous ones. It has described a specific system and, therefore, delved into topics that are somewhat alien to this thesis. This has been necessary in order to correctly present the application.

The aim was to show that the algorithms and techniques previously presented can all come together in a system that has great potential to be useful in real life. Therefore, it complements the mere quantitative evaluations of the previous chapters with a more qualitative evaluation of the potentials of the thesis contributions.

Chapter 8

Conclusions

This thesis tackles the problem of indoor position estimation and tracking of people and objects with a number of innovative approaches based on:

- a low-cost and energy-aware localization infrastructure;
- multi-sensor, statistically-based, localization algorithms;
- logic-based situation assessment techniques.

The thesis rests on a number of realistic but still arguable set of assumptions.

First of all it relies on the assumption that it will become common for a building to have a large number of (wirelessly) connected sensors. The will of people and government to move towards a “greener” society have strong implications on how we will build and use homes, offices and factories in the future. Better construction material and techniques will be critical but also energy management will have to bear its share. To implement a greener world we cannot, at least in the “old” countries, wipe away and rebuild all the existing buildings, be it for economical or for historical reasons. Therefore, better control (i.e. sensing and actuation of all the parameters that influence the state of a building) will have to be retrofitted: wireless and battery-operated will be the name of the game.

Secondly, the thesis relies on the constant technological improvement of both hardware devices and of the batteries to power them. The constant improvement of silicon-based devices is guaranteed for many years to come

because we are more and more becoming dependent on such devices in our lives, e.g. mobile telephones. Batteries are a more difficult subject because there are no clear winning solutions. Recently, though, new insights have been gained into building radioisotopes-based batteries that are as small as a coin and are almost an order of magnitude more powerful than chemical batteries. Moreover, sometimes energy-scavenging might do the trick and free us from batteries.

Finally, the thesis rests on the usefulness of tracking and, in particular, situation-understanding. It is easy to justify tracking, the growing popularity of (sometime controversial) location-aware applications shows that tracking is indeed a very useful functionality. Less obvious is how we will be able to harness situation-understanding to help people. The SINDI system described in the thesis is a good example but much could also be argued for the lack of privacy that pervasive situation assessment systems might generate.

The algorithms and techniques that are the outcome of this thesis have all been tested by implementing them and measuring the performance in the field. Both in a quantitative sense, e.g. accuracy and precision, and in a qualitative sense, e.g. within the SINDI system. Unfortunately, there were neither time nor resources for a really pervasive test using thousands of wireless nodes in a large building.

This will be the next challenge.

Appendix A
Schematics of Data-Gathering
Nodes

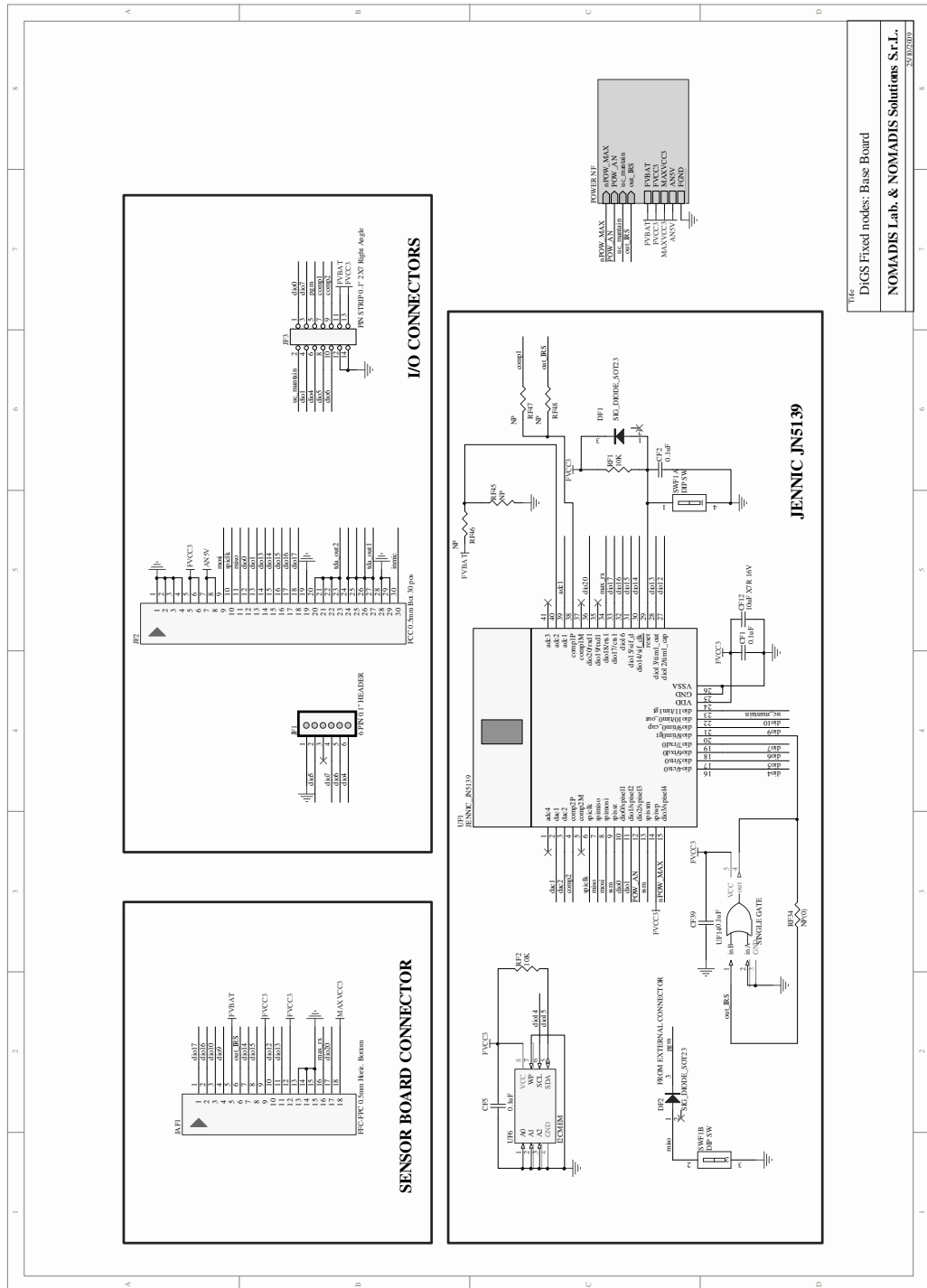


Figure 1: DiGS Fixed nodes: Base Board

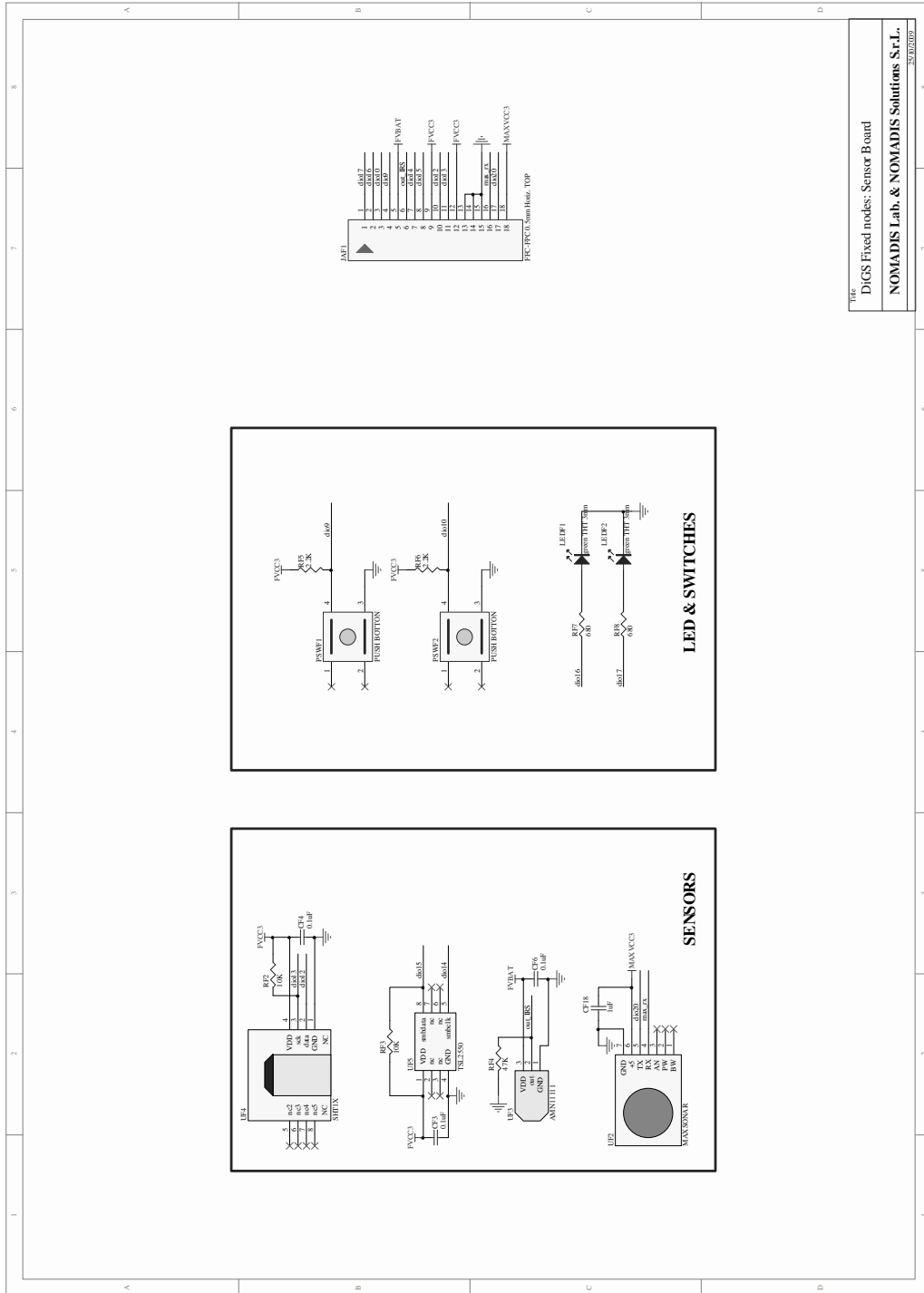


Figure 2: DiGS Fixed nodes: Sensor Board

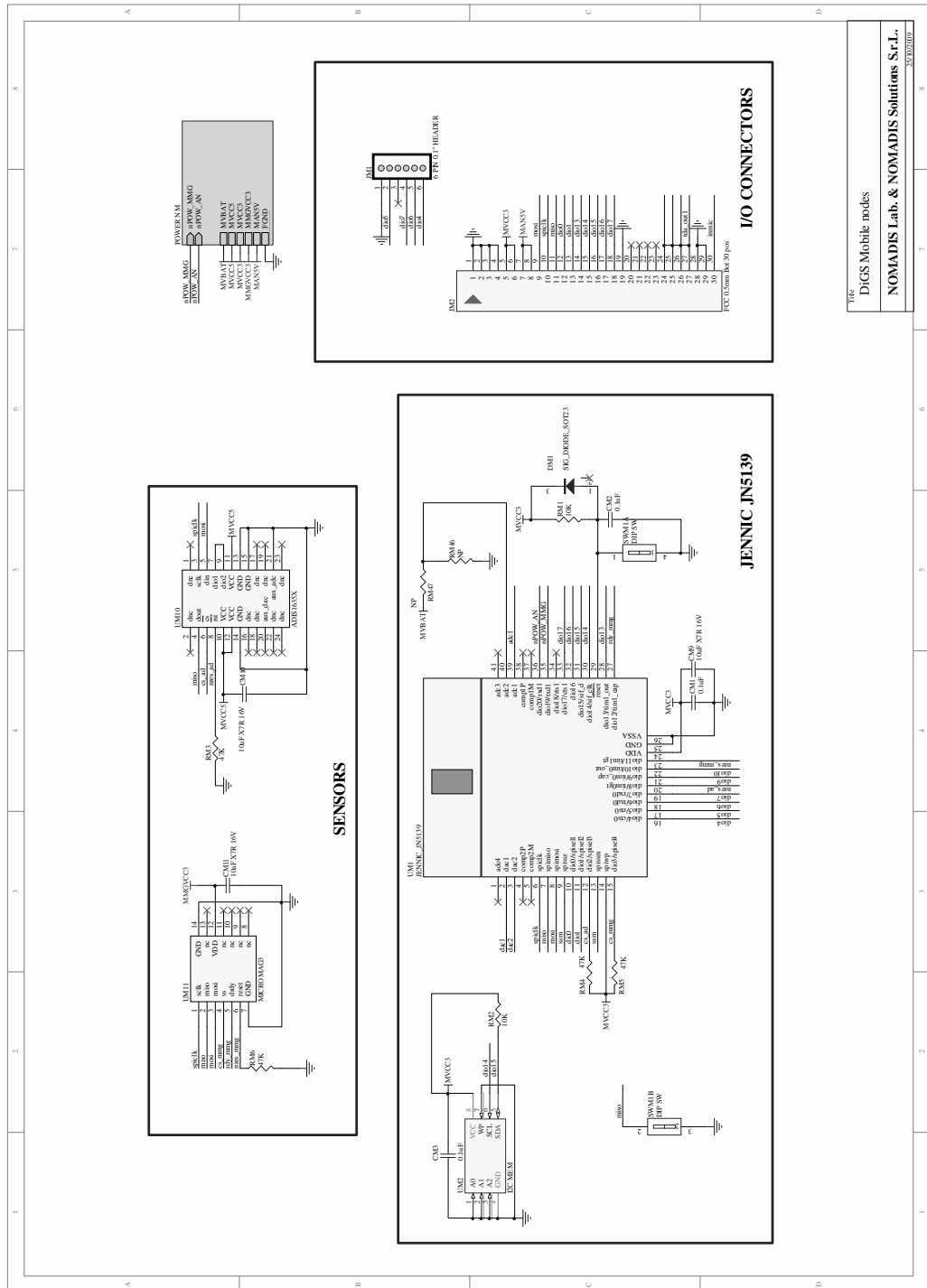


Figure 3: DiGS Mobile nodes

Appendix B

Document Type Definition of the Data Generation Tool

Listing 1: Document Type Definition of the Scenario XML File

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE scenario [
  <!ELEMENT scenario ((walls, rooms, passages, loc_sensors, env_sensors,
    light_sources, heat_sources, windows, light_switches))>

  <!ELEMENT walls ((wall_cell+))>
  <!ELEMENT wall_cell EMPTY>
  <!ATTLIST wall_cell
    posX CDATA #REQUIRED
    posY CDATA #REQUIRED
  >

  <!ELEMENT rooms ((room+))>
  <!ELEMENT room ((area+))>
  <!ATTLIST room
    key CDATA #REQUIRED
    name CDATA #REQUIRED
  >
  <!ELEMENT area ((room_cell+))>
  <!ATTLIST area
    key CDATA #REQUIRED
    name CDATA #REQUIRED
  >
  <!ELEMENT room_cell EMPTY>
  <!ATTLIST room_cell
    posX CDATA #REQUIRED
    posY CDATA #REQUIRED
  >
</>
```

```

<!ELEMENT passages ((passage+))>
<!ELEMENT passage ((passage_cell+))>
<!ATTLIST passage
  fromA CDATA #REQUIRED
  fromR CDATA #REQUIRED
  toA CDATA #REQUIRED
  toR CDATA #REQUIRED
>
<!ELEMENT passage_cell EMPTY>
<!ATTLIST passage_cell
  posX CDATA #REQUIRED
  posY CDATA #REQUIRED
>

<!ELEMENT loc_sensors ((loc_sensor+))>
<!ELEMENT loc_sensor ((position , pir , range))>
<!ATTLIST loc_sensor
  key CDATA #REQUIRED
>
<!ELEMENT position EMPTY>
<!ATTLIST position
  orient (N | S | W | E) #IMPLIED
  posX CDATA #REQUIRED
  posY CDATA #REQUIRED
>
<!ELEMENT pir ((pir_cell+))>
<!ELEMENT pir_cell EMPTY>
<!ATTLIST pir_cell
  posX CDATA #REQUIRED
  posY CDATA #REQUIRED
>
<!ELEMENT range ((range_cell+))>
<!ELEMENT range_cell EMPTY>
<!ATTLIST range_cell
  posX CDATA #REQUIRED
  posY CDATA #REQUIRED
>

<!ELEMENT env_sensors ((env_sensor+))>
<!ELEMENT env_sensor ((position , sense_cell))>
<!ATTLIST env_sensor
  key CDATA #REQUIRED
>
<!ELEMENT sense_cell EMPTY>
<!ATTLIST sense_cell
  posX CDATA #REQUIRED
  posY CDATA #REQUIRED
>

```

```
<|ELEMENT light_sources ((light_source+))>
<|ELEMENT light_source ((light_cell+))>
<|ATTLIST light_source
  key CDATA #REQUIRED
  room CDATA #REQUIRED
  type (natural | artificial) #REQUIRED
>
<|ELEMENT light_cell EMPTY>
<|ATTLIST light_cell
  posX CDATA #REQUIRED
  posY CDATA #REQUIRED
>
heat
<|ELEMENT heat_sources ((heat_source+))>
<|ELEMENT heat_source ((heat_cell+))>
<|ATTLIST heat_source
  key CDATA #REQUIRED
  room CDATA #REQUIRED
  type CDATA #REQUIRED
>
<|ELEMENT heat_cell EMPTY>
<|ATTLIST heat_cell
  posX CDATA #REQUIRED
  posY CDATA #REQUIRED
>

<|ELEMENT windows ((window+))>
<|ELEMENT window EMPTY>
<|ATTLIST window
  initialStatus (open | close) #REQUIRED
  key CDATA #REQUIRED
  light_source CDATA #REQUIRED
>

<|ELEMENT light_switches ((light_switch+))>
<|ELEMENT light_switch ((position+))>
<|ATTLIST light_switch
  initialStatus (turn-off | turn-on) #REQUIRED
  key CDATA #REQUIRED
  light_source CDATA #REQUIRED
>
|>
```

Listing 2: Document Type Definition of the Person Schedule XML File

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE schedule [
<ELEMENT schedule ((startingPosition , activities , actionSets))>
<!ATTLIST schedule
    descr          CDATA #IMPLIED
    endAt_HH       CDATA #REQUIRED
    endAt_MM       CDATA #REQUIRED
    startAt_HH     CDATA #REQUIRED
    startAt_MM     CDATA #REQUIRED
>
<ELEMENT startingPosition EMPTY>
<!ATTLIST startingPosition
    posX CDATA #REQUIRED
    posY CDATA #REQUIRED
>
<ELEMENT activities ((activity+))>
<ELEMENT activity ((actionSet+))>
<!ATTLIST activity
    name          CDATA #REQUIRED
    startAt_HH    CDATA #REQUIRED
    startAt_MM    CDATA #REQUIRED
>
<ELEMENT actionSets ((actionSet+))>
<ELEMENT actionSet ((action*))>
<!ATTLIST actionSet
    name CDATA #REQUIRED
>
<ELEMENT action EMPTY>
<!ATTLIST action
    descr CDATA #IMPLIED
    posX CDATA #REQUIRED
    posY CDATA #REQUIRED
    type (goto | turn-on | turn-off | open | close) #REQUIRED
    waitMM CDATA #REQUIRED
    window CDATA #IMPLIED
    switch CDATA #IMPLIED
>
]>

```

Listing 3: Document Type Definition of the Daylight, Temperature, Humidity Schedule XML File

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE daylightSchedule [
<ELEMENT daylight_Temp_Hum_Schedule ((externalLight , temperature ,
    humidity))>

<ELEMENT externalLight (extLightSched)>
<ELEMENT extLightSched EMPTY>
<ATTLIST extLightSched
    fromHH CDATA #REQUIRED
    fromMM CDATA #REQUIRED
    toHH CDATA #REQUIRED
    toMM CDATA #REQUIRED
    light CDATA #REQUIRED
>

<ELEMENT temperature (extTempSched)>
<ELEMENT extTempSched EMPTY>
<ATTLIST extTempSched
    fromHH CDATA #REQUIRED
    fromMM CDATA #REQUIRED
    toHH CDATA #REQUIRED
    toMM CDATA #REQUIRED
    temp CDATA #REQUIRED
>

<ELEMENT humidity (extHumSched)>
<ELEMENT extHumSched EMPTY>
<ATTLIST extHumSched
    fromHH CDATA #REQUIRED
    fromMM CDATA #FIXED "00"
    toHH CDATA #REQUIRED
    toMM CDATA #FIXED "59"
    humidity CDATA #REQUIRED
>
]>

```


Bibliography

- [1] Mark Weiser. The computer for the 21st century. *Scientific American*, February 1991.
- [2] Adam Greenfield. *Everyware: The Dawning Age of Ubiquitous Computing*. New Riders, 2006.
- [3] Ahmed El-Rabbany. *Introduction to GPS: the Global Positioning System*. Artech House, Norwood, MA, USA, 2002.
- [4] Ramjee Prasad and Marina Ruggieri. *Applied Satellite Navigation Using GPS, Galileo, and Augmentation Systems*. Artech House, Norwood, MA, USA, 2005.
- [5] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.
- [6] A.J. Motley and J.M.P. Keenan. Personal communication radio coverage in buildings at 900 mhz and 1700 mhz. *Electronics Letters*, 24(12):763–764, Jun 1988.
- [7] Vaidyanathan Ramadurai and Mihail L. Sichitiu. Localization in wireless sensor networks: A probabilistic approach. In *Proceedings of ICWN 2003*, pages 275–281, Las Vegas, Nevada, USA, 23-26 June 2003. CSREA Press, Washington, DC, USA.
- [8] D. Niculescu and Nath Badri. Ad hoc positioning system (aps) using aoa. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1734–1743 vol.3, March-3 April 2003.
- [9] D. Grejner-Brzezinska. *Positioning and Tracking Approaches and Technologies*, volume Telegeoinformatics - Location-Based Computing and Services, page 69–106. CRC Press, 2004.

-
- [10] Branislav Kusy. *Spatiotemporal Coordination in Wireless Sensor Networks*. PhD thesis.
- [11] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT Press, 2005.
- [12] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME – Journal of Basic Engineering*, (82 (Series D)):35–45, 1960.
- [13] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2001.
- [14] Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.
- [15] The Ubisense RTLS. <http://www.ubisense.net/>.
- [16] The AeroScout RTLS. <http://www.aeroscout.com/>.
- [17] Ekahau. Wi-Fi based RTLS tracking and communication solutions. <http://www.ekahau.com/>.
- [18] The Ubiquicom Locator RTLS. <http://www.ubiquicom.it/>.
- [19] Texas Instruments CC2431. <http://www.ti.com/corp/docs/landing/cc2431/>.
- [20] The Cricket Indoor Location System. <http://cricket.csail.mit.edu/>.
- [21] Honeywell GyroDRM. <http://www.magneticsensors.com/>.
- [22] Vectronix DRC. <http://www.vectronix.ch/>.
- [23] L. E. Miller. Indoor navigation for first responders: A feasibility study. Technical report, National Institute of Standards Technology, February 2006.
- [24] L. E. Miller, P. F. Wilson, N. P. Bryner, M. H. Francis, J. R. Guerrieri, D. W. Stroup, and L. Klein-berndt. Rfid-assisted indoor localization and communication for first responders. In *In Proc. of the Int. Symposium on Advanced Radio Technologies*, 2006.

- [25] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.
- [26] Brett Warneke, Matt Last, Brian Liebowitz, and Kristofer S. J. Pister. Smart dust: Communicating with a cubic-millimeter computer. *Computer*, 34(1):44–51, 2001.
- [27] Wolfram Kluge, Frank Poegel, Hendrik Roller, Matthias Lange, Tilo Ferchland, Lutz Dathe, and Dietmar Eggert. A fully integrated 2.4-ghz ieee 802.15.4-compliant transceiver for zigbee applications. *IEEE Journal of Solid-State Circuits*, pages 2767–2775, 2006.
- [28] Shad Roundy, Paul Kenneth Wright, and Jan M. Rabaey. *Energy Scavenging for Wireless Sensor Networks*. Springer, 2003.
- [29] I. Demirkol, C. Ersoy, and F. Alagoz. Mac protocols for wireless sensor networks: A survey. *IEEE Communications Magazine*, 44(4):115–121, 2006.
- [30] IEEE 802.15.4 WPAN-LR Task Group. <http://www.ieee802.org/15/pub/TG4.html/>.
- [31] ZigBee Alliance. <http://www.zigbee.org/>.
- [32] 6LowPAN - Wireless Internet of Things. <http://www.6lowpan.org/>.
- [33] Manish Kochhal, Loren Schwiebert, and Sandeep Gupta. Role-based hierarchical self organization for wireless ad hoc sensor networks. In *Proceedings of WSNA 2003*, pages 98–107, San Diego, CA, USA, 19 September 2003. ACM Press, New York, NY, USA.
- [34] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless sensor networks. In *Proceedings of HICSS 2000*, pages 8020–8029, Maui, Hawaii, USA, 4-7 January 2000. IEEE Computer Society, Washington, DC, USA.
- [35] A. Manjeshwar and D. P. Agrawal. Teen: A routing protocol for enhanced efficiency in wireless sensor networks. In *Proceedings of IPDPS 2001*, pages 2009–2015, San Francisco, CA, 23-27 April 2001. IEEE Computer Society, Washington, DC, USA.
- [36] A. Manjeshwar and D. P. Agrawal. Apteen: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor

- networks. In *Proceedings of IPDPS 2002*, pages 195–202, Fort Lauderdale, FL, 15-19 April 2002. IEEE Computer Society, Los Alamitos, CA, USA.
- [37] Roberto Verdone, Davide Dardari, Gianluca Mazzini, and Andrea Conti. *Wireless Sensor and Actuator Networks: Technologies, Analysis and Design*. Academic Press, 2008.
- [38] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low-cost outdoor localization for very small devices. *Personal Communications, IEEE*, 7(5):28–34, Oct 2000.
- [39] B. M. Blum J. A. Stankovic T. He, C. Huang and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pages 81–95, 2003.
- [40] Andreas Savvides, Heemin Park, and Mani B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. pages 112–121, 2002.
- [41] Miklós Maróti, Péter Völgyesi, Sebestyén Dóra, Branislav Kusý, András Nádas, Ákos Lédeczi, György Balogh, and Károly Molnár. Radio interferometric geolocation. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 1–12, New York, NY, USA, 2005. ACM.
- [42] Branislav Kusý, Ákos Lédeczi, Miklós Maróti, and Lambert Meertens. Node density independent localization. In *IPSN '06: Proceedings of the 5th international conference on Information processing in sensor networks*, pages 441–448, New York, NY, USA, 2006. ACM.
- [43] Andreas Savvides, Chih-Chieh Han, and Mani B. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, New York, NY, USA, 2001. ACM.
- [44] C. Savarese, J. Rabay, and K. Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *In Proceedings of the Annual USENIX Technical Conference*, 2002.
- [45] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. The flooding time synchronization protocol. In *SenSys '04: Proceedings of*

- the 2nd international conference on Embedded networked sensor systems*, pages 39–49, New York, NY, USA, 2004. ACM.
- [46] Hyunhak Kim, Daeyoung Kim, and Seong-eun Yoo. Cluster-based hierarchical time synchronization for multi-hop wireless sensor networks. *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on*, 2:5 pp.–, April 2006.
- [47] Panasonic AMN11111. <http://pewa.panasonic.com/pcsd/product/sens/>.
- [48] Maxbotix LV-MaxSonar-EZ1. <http://www.maxbotix/>.
- [49] Sensirion SHT11. http://www.sensirion.com/en/01_humidity_sensors/02_humidity_sensor_sht11.htm.
- [50] TAOS TSL2550. <http://www.taosinc.com/productdetail.aspx?product=20/>.
- [51] Sparkfun 6DOFv3 IMU. <http://www.sparkfun.com/>.
- [52] Freescale MMA7260Q. <http://www.freescale.com/sensors/>.
- [53] InvenSense IDG300. http://www.invensense.com/products/idg_300.html.
- [54] Honeywell HMC1043. <http://www.ssec.honeywell.com/magnetic/magnetometers.html>.
- [55] Analog Devices ADIS 16355. <http://www.analog.com/en/other-products/multi-chip/adis16350/products/product.html>.
- [56] PNI MicroMag3. <http://www.pnicorp.com/products/all/micromag-modules/>.
- [57] Wei Bo, Hu Han-ying, and Fu Wen. An improved leach protocol for data gathering and aggregation in wireless sensor networks. *Computer and Electrical Engineering, International Conference on*, 0:398–401, 2008.
- [58] Rimon Barr, Zygmunt J. Haas, and Robbert van Renesse. Jist: an efficient approach to simulation using virtual machines: Research articles. *Softw. Pract. Exper.*, 35(6):539–576, 2005.
- [59] Jennic Wireless Microcontrollers. <http://www.jennic.com/>.

- [60] Jennic Application Note: JN-AN-1055. http://www.jennic.com/files/support_documentation/JN-AN-1055-Using-Coin-Cells-1v1.pdf.
- [61] David W. Scott. *Multivariate Density Estimation*. John Wiley Sons, Inc., 1992.
- [62] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.
- [63] Michael Broxton, Joshua Lifton, and Joseph A. Paradiso. Localization on the pushpin computing sensor network using spectral graph drawing and mesh relaxation. *SIGMOBILE Mobile Computer Communication Review*, 10, 2006.
- [64] Mica R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37:32–64(33), March 1995.
- [65] Michael J. North and Charles M. Macal. *Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation*. Oxford University Press, Inc., New York, NY, USA, 2007.
- [66] The Swarm Agent-based modeling platform. http://www.swarm.org/index.php/Swarm_main_page.
- [67] The ASCAPE Framework. <http://ascape.sourceforge.net/>.
- [68] The MASON Framework. <http://cs.gmu.edu/~eclab/projects/mason/>.
- [69] The NetLogo modeling environment. <http://ccl.northwestern.edu/netlogo/>.
- [70] J. C. Augusto. Increasing reliability in the development of intelligent environments. In *Proceedings of 5th International Conference on Intelligent Environments (IE09)*, Barcelona, Spain, 20-21 July 2009. Ios Press.
- [71] A. Mendez-Vazquez, A. Helal, and D. Cook. Simulating events to generate synthetic data for pervasive spaces. *Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research*, April 2009.

- [72] Alexander Dupuy, Jed Schwartz, Yechiam Yemini, and David Bacon. Nest: a network simulation and prototyping testbed. *Commun. ACM*, 33(10):63–74, 1990.
- [73] I. Armac and D. Retkowitz. Simulation of smart environments. *IEEE International Conference on Pervasive Services*, pages 257–266, 15-20 July 2007.
- [74] B. Schmidt. *The Modelling of Human Behaviour*. SCS Publications, 2000.
- [75] Hedetniemi, Hedetniemi, and Liestman. A survey of gossiping and broadcasting in communication networks. *NETWORKS: Networks: An International Journal*, 18, 1988.
- [76] G. Brewka, I. Niemelä, and T. Syrjänen. Implementing ordered disjunction using answer set solvers for normal programs. In *Proceedings of JELIA 2002*, pages 444–455, Cosenza, Italy, 23-26 September 2002. SPRINGER-BERLIN.
- [77] P. Simons, I. Niemelä, and T. Soinen. Extending and implementing the stable model semantics. *Artificial Intelligence Journal*, 138(1-2):181–234, 2002.
- [78] N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, and F. Scarcello. The dlv system for knowledge representation and reasoning. *ACM Transactions on Computational Logic*, 7(3):499–562, 2006.
- [79] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. Conflict-driven answer set solving. In *Proceedings of IJCAI 2007*, pages 386–392, Hyderabad, India, 6-12 January 2007. AAAI Press, Massachusetts, USA.
- [80] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proceedings of ICLP 88*, pages 1070–1080, Seattle, Washington, 15-18 August 1988. MIT Press, Massachusetts, USA.
- [81] A user’s guide to Gringo, clasp, clingo, and iclingo. <http://potassco.sourceforge.net>.
- [82] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. clasp: A conflict-driven answer set solver. In *Proceedings of LPNMR 2007*, pages 260–265, Tempe, AZ, USA, 15-17 May 2007.

- [83] Ilkka Niemelä and Patrik Simons. Extending the smodels system with cardinality and weight constraints. In Jack Minker, editor, *Logic-based artificial intelligence*, pages 491–521. Kluwer Academic Publishers, Norwell, MA, USA, 2000.
- [84] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, pages 263–277, 2007.
- [85] Thomas Strang and Claudia Linnhoff-Popien. A context modeling survey. In *In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England*, 2004.
- [86] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung. Ontology based context modeling and reasoning using owl. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 18–22, Washington, DC, USA, 2004. IEEE Computer Society.
- [87] N. S. Ryan, J. Pascoe, and D. R. Morse. Enhanced reality fieldwork: the context-aware archaeological assistant. In V. Gaffney, M. van Leusen, and S. Exxon, editors, *Computer Applications in Archaeology 1997*, British Archaeological Reports. Tempus Reparatum, 1998.
- [88] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Proc. of the Workshop on Mobile Computing Systems and Applications*, pages 85–90. IEEE Computer Society, 1994.
- [89] Anand Ranganathan and Roy H. Campbell. An infrastructure for context-awareness based on first order logic. *Personal Ubiquitous Computing*, 7(6):353–364, 2003.
- [90] Bingo: an experimental bottom-up grounder based on semi-naive evaluation. <http://potassco.sourceforge.net/labs.net>.
- [91] iClingo: an incremental ASP system implemented on top of Clasp and Gringo. <http://www.cs.uni-potsdam.de/iclingo/>.
- [92] Veena S. Mellarkod, Michael Gelfond, and Yuanlin Zhang. Integrating answer set programming and constraint logic programming. *Ann. Math. Artif. Intell.*, 53(1-4):251–287, 2008.

- [93] R. Bisiani, D. Merico, A. Mileo, and S. Pinaridi. A logical approach to home healthcare with intelligent sensor-network support. *The Computer Journal*, page bxn074, 2009.
- [94] M. V. Giuliani, M. Scopelliti, and F. Fornara. Elderly people at home: technological help in everyday activities. In *Proceedings of ROMAN 2005*, pages 365–370, Nashville, TN, 13-15 August 2005. IEEE Computer Society, Washington, DC, USA.
- [95] E. Dishman. Inventing wellness systems for aging in place. 37(5):34–41, 2004.
- [96] J. Teich, J. Schmitz, G. Kuperman, and D. Bates. Effects of computerized physician order entry on prescribing practices. *Archives of Internal Medicine*, 160:2741–2747, 2000.
- [97] Kevin C. Fleming, J. M. Evans, D. C. Weber, and D. S. Chutka. Practical functional assessment of elderly persons: A primary-care approach. *Mayo Clinic*, 70:890–910, 1995.
- [98] M. F. Folstein, S. E. Folstein, and P. R. McHugh. Mini-mental state: A practical method for grading the cognitive state of patients for the clinician. *Journal of Psychiatric Research*, 12(3):189–198, 1975.
- [99] S. Yamamoto, N. Mogi, H. Umegaki, Y. Suzuki, F. Ando, H. Shimokata, and A. Iguchi. The clock drawing test as a valid screening method for mild cognitive impairment. *Dementia and Geriatric Cognitive Disorders*, 18:172–179, 2004.
- [100] M.E. Tinetti, T.F. Williams, and P. Mayewski. Fall risk index for elderly patients based on number of chronic disabilities. *American Journal of Medicine*, 80:429–434, 2002.
- [101] J.A. Yesavage, T.L. Brink, T.L. Rose, O. Lum, V. Huang, M. Adey, and V.O. Leirer. Development and validation of a geriatric depression screening scale: A preliminary report. *Journal of Psychiatric Research*, 17(1):37–49, 1982-1983.
- [102] Y. Guigoz, B. Vellas, and P.J. Garry. Mini nutritional assessment: A practical assessment tool for grading the nutritional state of elderly patients. *Facts and Research in Gerontology*, 2:15–59, 1997.
- [103] S. Katz, T.D. Downs, H.R. Cash, and R.C. Grotz. Progress in the development of adl. *Gerontologist*, 10(1):20–30, 1970.

- [104] S. Jitapunkul, I. Pillay, and S. Ebrahim. The abbreviated mental test: Its use and validity. *Age Aging*, 20:332–336, 1991.
- [105] Martha E. Pollack. Intelligent technology for an aging population: The use of ai to assist elders with cognitive impairment. *AI Magazine*, 26(2):9–24, 2005.
- [106] M.P. Lawton. Scales to measure competence in everyday activities. *Psychopharmacological Bulletin*, 24(4):609–614, 1988.
- [107] Vangelis Metsis, Zhengyi Le, Yu Lei, and Fillia Makedon. Towards an evaluation framework for assistive environments. In *Proceedings of PETRA 2008*, pages 1–8, Athens, Greece, 16–18 July 2008. ACM Press, New York, NY, USA.