Smart Data Collection and Management in Heterogeneous Ubiquitous Healthcare

Luca Catarinucci, Alessandra Esposito, Luciano Tarricone,
Marco Zappatore and Riccardo Colella
University of Salento,
Italy

1. Introduction

The increasing availability of network connection and the progress in information technology and in hardware miniaturization techniques, are determining new computing scenarios, where software applications are able to "configure themselves" based on information coming from heterogeneous sources (sensors, RFID, GPS, databases, user input, etc.) which form the so called "context". In other terms, such applications are based on the representation and codification of different kinds of data, such as biomedical parameters, environmental data, device location, user preferences, resource availability, every time, from every location and through different modalities (pervasiveness), and on the provision of services and contents adapted to current context (context-awareness). Such computing scenarios find application in a large number of real-life domains, as environment monitoring, supply-chain management and so on. Health-care is perhaps one of the most relevant and promising. Indeed, the perspectives opened by such technologies are wide and variegate: they range from the home-care of mobility-impaired people to the harmonization and presentation to hospital workers of information gathered from distributed heterogeneous sources.

The implementation of context-aware systems is based on two distinct but strongly interlaced tasks: 1) monitoring and collection of sensorial data, with the related issues concerning data transmission, costs, enabling technologies as well as the heterogeneity and the number of data to be collected 2) processing and integration of data with available context information in order to activate decision processes which are in many cases not trivial.

Key points are the selection of the enabling technologies for the collection, transmission and smart management of data gathered from heterogeneous sources, and the design of a system architecture having the following characteristics:

a. simple to use;
b. low-cost and low-power consumption in order to make possible the implementation of systems with a high number of nodes;
c. interoperable with any type of sensors;
d. customizable to different kinds of application domains with a limited effort;
e. scalable with the number of nodes;

f. suitable response times so to be adopted also in emergency situations.

Based on the above considerations, we developed a cost-effective RFID-based device for the monitoring and collection of sensorial data, and a pervasive and context-oriented system which operates on sensor data and is based on an innovative, flexible and versatile framework. The description of the RFID device and of the software framework is provided in this chapter, which starts with an introduction to potential applications for healthcare and to emerging techniques for both data collection and smart data management, and concludes with the validation of both hardware and software solutions in the real-life use-case of patient remote assistance.

2. Potential applications

This paragraph proposes an overview of the possible applications of emerging technologies for data sensing, gathering and smart data elaboration with a special focus to the healthcare domain. We partition applications into four groups, even though some overlapping between such groups exist.

2.1 Knowledge sharing, availability and integration

Management and delivery of healthcare is critically dependent on access to data. Such data are normally provided by several heterogeneous sources, such as physiological sensors, imaging technology or even handwriting and often spans different organisations. Moreover they are generally stored by using different formats and terminologies, such as electronic health records, clinical databases, or free-text reports. If such a rich collection of health and community data could be linked together, we would improve enormously our capability of finding answers to health and social questions and of tackling complex diseases (Walker, 2005). Unfortunately, sharing such a complex aggregate of information and deriving useful knowledge from it is currently very difficult.

A very promising answer to this need is provided by the enabling technologies for smart data collection and management which are named "ontologies" [Section 3.4]. Ontologies define a formal semantically rich machine-readable specification of concepts and relationships which is unique and sharable among dispersed consumers. They provide coherent representations of biomedical reality thus enabling wide-area multi-organizational sharing of data by supporting publishing, retrieving and analysis of data in large-scale, diverse and distributed information systems.

Moreover, when integrated with other enabling technologies (such as those listed in Section 3.3 and 3.5) ontologies allow the identification of the correct information to be forwarded to health care professional depending on available information about context, such as time, place, patient current state, physiological data, and so on.

2.2 Independent living

The combination of sensors, RFID, wireless protocols and software solutions, is going to enhance aging population and impaired people capability of conducting a possibly *autonomous independent living*. Indeed, pervasive and context-aware applications are more and more recognized as promising solutions for providing continuous care services, while improving quality of life of people suffering from chronic conditions.

Support to independent living encompasses a wide range of applications ranging from the telemonitoring of vital parameters to scenarios involving home automation and domotics. In the former case (Paganelli, 2007), the system allows chronic patients, such as epileptics or persons with *chronic* mental illness, to conduct a substantially normal life as it makes a continuous measurement of vital data and/ or of the activities of daily living and social interactions. An immediate assistance is provided in emergency situations, when patients may be unconscious and unable to react. Indeed, the system is able to autonomously alert the best suited care giver and to forward her/ him a meaningful synthesis of real-time (e.g. physiological data) and static data (e.g. historical patient information).
Smart homes (SM4ALL, 2010), instead, consist in homes augmented with networked sensors, communicating objects, and information appliances. Ambient sensors acquire information about the home environment, body near sensors acquire information about the habitant. The combination of these sensors enables the creation of activity profiles which allow the constant monitoring of patient daily life. Also in these cases, adverse events are recognized by the system ("the habitant leaves the home while the cooker is still switched on", or "the patient has a fall") and specific assistance is quickly provided.

2.3 Tracking

The capability to trace, localize and identify a large number of heterogeneous objects in a complex and dynamic environment is essential in many sectors, including the healthcare one (Bacheldor, 2008, 2009), (Swedberg, 2010). Moreover, the widespread diffusion of the object/ people to be traced, forces the use of flexible, easy to use and inexpensive technologies. Such requirements are typical of RFID systems. The joint use of RFID tags and smart software infrastructures in identification badges for health care professionals, patients, and fixed assets, therefore, is more and more gaining momentum. Thanks to these technologies, physicians and nurses are tracked in real-time or near real-time, and contacted in case of emergencies, whilst specific medical devices, such as multi-channel infusion pumps, are associated with patients.
Tracking is also of utmost importance to control drug dispensation and laboratory results. Medical dosages and patient samples are validated, together with blood supplies and blood type matching.
In the pharmaceutical supply chain, where security and safety are required to prevent compromising the health of patients, smart labels and sensors are fundamental. Smart labels are attached to drugs, in order to track them through the overall supply chain. Sensors continuously monitor the status of items requiring specific storage conditions and discard them if such conditions are violated. Drug tracking and e-pedigrees also support the detection of counterfeit products and for keeping the supply chain free of fraudsters.
The smart labels on the drugs can also directly benefit patients in several manners, e.g. by storing the package insert, informing consumers of dosages and expiration date, by reminding patients to take their medicine at appropriate intervals and monitoring patient compliance, etc...

2.4 Smart hospitals

With the ability of capturing fine grained data pervasively, wireless sensors and RFID tags have numerous available and potential applications in hospitals (Wang, 2006). The quality

of patient care at the bedside is improved and errors reduced. This is accomplished by integrating the physical process of care delivery with medication information and software applications that provide clinical decision support, and quality and safety checks.
Without such systems, healthcare providers traditionally use a paper-based "flow chart" to store patient information during registration time, which is updated by different professionals and forwarded to the incoming staff at the end of each shift. As a result the paper report is not always accurate. RFIDs allow for transmitting and receiving data from patient to health professionals without human intervention. RFID wristbands are used to identify patients during the entire hospitalization period. They are used to store patient data (such as name, patient ID, drug allergies, blood group, and so on) in order to constantly keep staff informed. Medical professionals can easy access and update patient's records remotely via Wi-Fi connection using mobile devices.

3. Emerging technologies

Pervasive platforms derive from the merging of several technologies which recently had an impressive improvement, such as wireless communications and networking, mobile computing and handheld devices, embedded systems, etc.
As a result, a review of the emerging technologies which enable the development of pervasive systems may reveal a hard and tedious task. The focus of this section is therefore restricted on a subset of such technologies. We substantially concentrate on the technologies which will be cited in the course of the chapter and which we believe actually constitute, and will probably still constitute in the future, the core of pervasive context-aware systems.
Computing environments are inherently distributed and heterogeneous. They are made of components which can be mobile, embedded in everyday objects or in general hidden in the infrastructure surrounding the user (Weiser, 1991). The context data to be computed must be collected by spread, inexpensive and easy to use data collectors. Those based on the *RFID technology* (Section 3.1) guarantee both identification and localization. The *integration of sensors on traditional RFID tags* (Section 3.2) assure also the interaction with other physical values, mandatory for a realistic context reconstruction. Such heterogeneous and dispersed components have different requirements and capabilities and are expected to interact continuously and in a transparent manner.
"Interaction" and "transparency" in a "distributed" environment are well supported by the so-called *multi-agent paradigms*, which, as explained in Section 3.3, organize applications around the coordinated interaction of autonomous reasoning software entities, which wrap the "natural" components of the distributed system.
Pervasive systems must also exhibit some intelligent behavior in terms of context awareness, i.e. their components must be able to coordinate with one another in order to recognize and react aptly to environment modifications. "Coordination" and "reasoning" require a high level of (semantic) interoperability, i.e. the availability of a shared and agreed knowledge layer upon which agents base their inferences and cooperation.
For this purpose, *ontologies* (Section 3.4) and *rule based logic* (Section 3.5) seem the best combination of technologies for enabling knowledge representation and reasoning, the former provide a sharable, reusable model of reality, the latter efficiently support inferencing and event recognition.

3.1 RFID

Radio frequency identification is one of the emerging technologies that is having the most capillary diffusion, thanks to its low cost, its easiness and the very large number of applications where it can be crucial.

RFID can be seen as the enhanced version of the union between radio frequency (RF) anti-theft systems and bar codes.

RF anti-theft systems are based on passive devices, called tags, essentially constituted by a spiral antenna in parallel to a capacitor. If a tag is exposed to a certain electromagnetic field, such as those generated by the RF gates at the shop exits, the spiral antenna receives enough power to charge the capacitor. Once the capacitor discharges, a new signal is transmitted through the spiral antenna and received by the RF gate, generating the alert.

As well known, vice versa, a barcode is a series of bars and spaces through which information is encoded. The reading is then delegated to optical scanners which, however, work only a few centimetres away and only in line-of-sight condition.

RFID technology, therefore, combines the capability of barcodes to transmit complex information with the peculiarity of responding only when interrogated through a proper RF signal, characterizing passive anti-theft systems.

Substantially, if a book marked with an RF anti-theft tag passes through a RF gate, a nonspecific alert is generated (telling us that something has been stolen). A more detailed information about the book can be obtained only by reading its barcode through an optical scanner. When the same book is marked with an RFID tag, we can know which book is passing through the gate in terms of title, author, cost, number of pages, inventory number and so on. It is apparent, though, how this sophisticated anti-theft system is only one of the most simple applications which can take undeniable advantage from an RFID-based implementation.

A generic RFID system is, hence, mainly based on two entities: on one side there are the tags, which store in a microchip information in terms of electronic product code (EPC), and can receive and transmit signals by means of an aluminium or copper antenna. On the other side there are the readers and the reader antennas, which emit electromagnetic signals and wait for an answer. When a tag is in the region covered by a reader antenna, the tag microchip is powered and a signal containing the EPC, is sent back towards the reader antennas. The reader decodes such an identifier code (ID), telling a middleware system that a specific tag has been read by a certain reader antenna. In Fig.1 a schematization of an RFID system is reported.

The reasons for the spread diffusion of RFID technology, however, is not to be found in its working principle, but rather in its cost and its ease of use. The mere wireless transmission of an ID under certain electromagnetic solicitations, in fact, does not hide any particular technological impediment. On the contrary, the capability of using passive devices, which do not require maintenance and cost a few cents, is a peculiarity of RFID tags.

This promises a widespread use, for instance, also in applications where the cost should remain affordable and reusability is not guaranteed.

A comprehensive classification of the RFID technology in terms of working frequency, maximum working distance, tag powering method, cost, sensitivity, storage capacity etc., is certainly possible but is beyond the scope of this chapter. The interested reader is addressed to (Dobkin, 2007) for such a goal.

On the contrary, it is worth highlighting that, among the others, passive RFID systems in the ultra high frequency (UHF) band are those that provide the wider perspectives of applications.

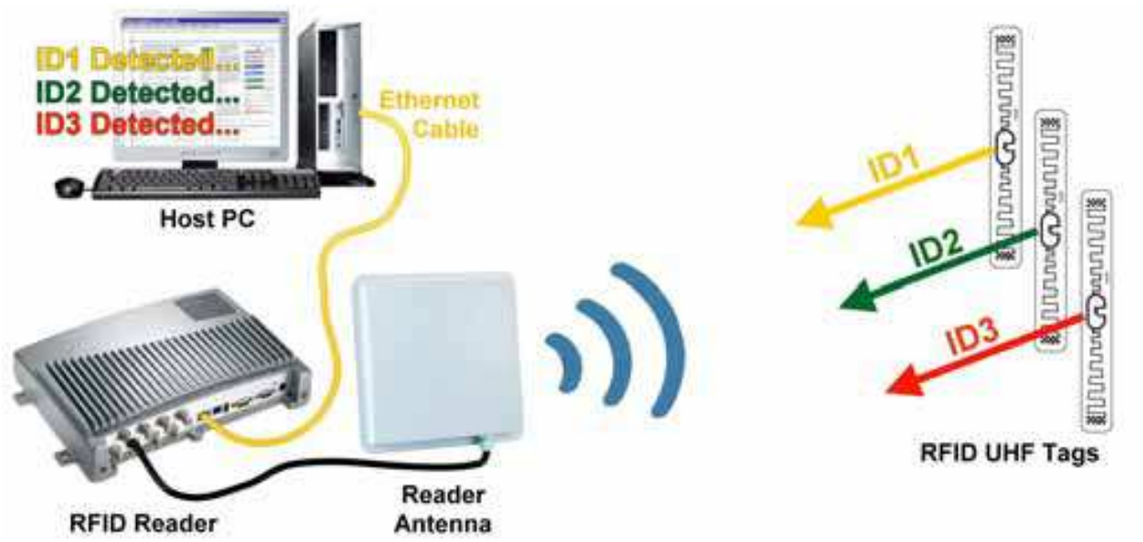

Fig. 1. Schematization of an RFID System

Their work frequency varies accordingly to individual state regulations; it ranges from a minimum of 860 MHz, for instance in Europe, to a maximum of 960 MHz, in Japan, via the 915 MHz of the United States (Dobkin, 2007). In Fig. 2 a prototype of a home-made UHF passive tag is shown. Both chip and antennas are clearly visible in the figure. RFID tags are generally bonded to a polyethylene terephthalate (PET) layer, which can be adhesive or dry. The prototype of Fig.2, on the contrary, is realized by cutting an adhesive sheet of copper by using a numerically controlled cutting plotter.

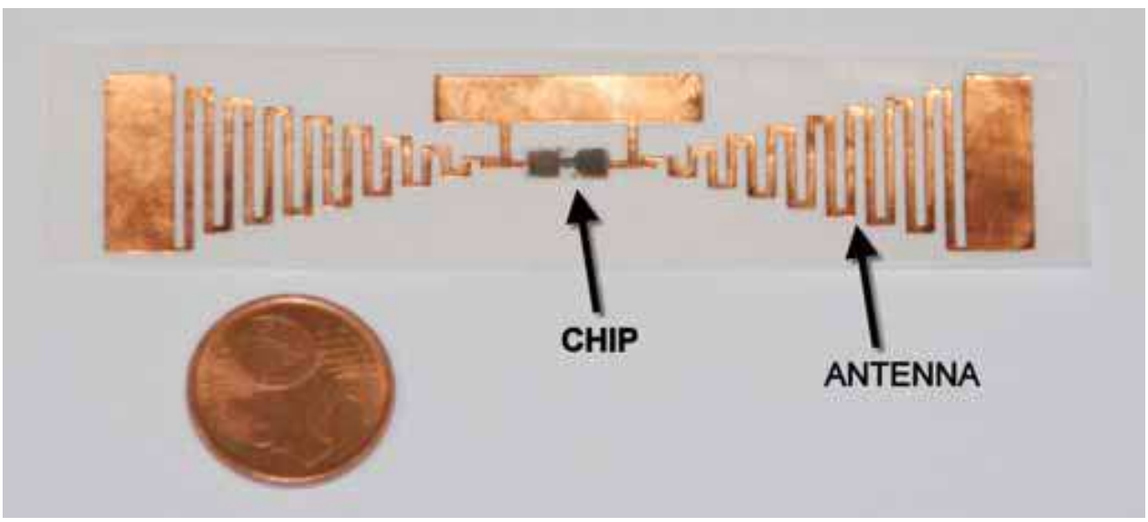

Fig. 2. Home-made prototype of a UHF RFID Tag

Although the apparent simplicity of the tag, it is important to stress that the antenna design plays a very crucial role. The chip exhibits a strong capacitive contribution, so that a conjugate matching technique, guaranteeing the maximum power transfer between chip and antenna, must be applied. Moreover, the need of a reduced size forces the use of miniaturization techniques (Dobkin, 2007). A well designed antenna guarantees reading

range up to 9 m, good enough to support the use of this technology in many applications based on auto-identification. Vice versa, when an application needs the management of more complex data, such as physical values representing the context around the tag, an integration between RFID and sensors is mandatory.

3.2 RFID and sensor integration

An evolution of RFID technology is represented by its integration with sensors, so that, along with the ID code, also the value measured by the sensor is sent towards the reader. In such a way, the collection and management of heterogeneous data, similarly to canonical wireless sensor networks, is made possible, with some advances in terms of benefit/ cost ratio.

Indeed, many different wireless technologies are mature for the transmission of data measured by generic sensors, much more than the inexpensive RFID. Compared with RFID technology, Wi-fi, Bluetooth, GPRS, UMTS and GSM guarantee wider transmission distances, higher bit-rates as well as larger amount of exchangeable data. Unluckily, the benefit/ cost ratio of such technologies, depending on the target application, is often not so appropriate. Therefore, when a capillary diffusion of physical sources is needed, a cheaper wireless technology is desirable. Moreover, the easy interfaceability with Internet could be another added value.

So far, the integration of wireless sensor networks with RFID-based sensor systems appears to be the most practicable way. UHF RFID technology, in fact, is quite inexpensive (passive RFID tags are as cheap as few euro-cents). It is also naturally compatible with Internet, and provides a reading range adequate for many applications. It is worth mentioning that some UHF RFID tags with embedded sensors of temperature or of pressure are already available on the market (Cho et al., 2005), being often inexpensive and rather accurate. Nevertheless the study of general purpose devices that support the integration of RFID and generic sensors, independently of the size and of the cost of the sensor itself, seems to be the new tendency.

Furthermore, the ability to gather in an easy and low-cost way a large amount of identification and context data, paves the way to the solution of ambitious problems that require the smart management of data. For such a goal, though, it is necessary to complement this hardware resources with new adequate software technologies, able to smartly support the decision-making processes.

3.3 Software agents

Software agents (Bradshaw, 1997) are (semi)autonomous programs which represent objects from the real world, such as persons, devices and services. Real entities delegate to agents the capabilities which are requested by ubiquitous context-aware systems, such as the interaction with the environment and with other entities (i.e. agents).

Agents are autonomous, i.e. they are distinct entities, separated, in some sense, from the world in which they exist. They are able to correctly operate with little or no direct human input or supervision, and have control of their internal state and behavior.

Agents interaction with the environment happens in a both reactive and proactive manner. Reactivity means that the agent is able to acquire information about the operating environment – through sensors, messages or interaction with humans – and adapt to perceived changes. Pro-activity means that the agent has a number of goals and is able to take the initiative to realize them.

Agent-based systems are used in many situations. For instance, they are worth to be adopted when the knowledge required to solve a given problem is spatially distributed, when different computational entities must cooperate to reach a common goal and when a given task can be subdivided in autonomous sub-tasks. Many problems in the healthcare domain have such features: knowledge is usually distributed in different physical locations; medical problem solving strategies quite always require collaboration between different healthcare operators, structures and devices; finally medical procedures are complex and are suited to be partitioned into different sub-tasks.
According to Nealon and Moreno (Moreno, 2003) the healthcare domain is a "vast open environment, characterised by shared and distributed decision making and management... requiring the communication of complex and different forms of information between a variety of clinical and other settings, as well as the coordination between groups of healthcare professionals with very different skills and roles". As agents may tackle distributed problems, are able to communicate among each other, can partition complex problems in sub-problems, take decision autonomously and provide information to end users reactively and proactively, they represent a good choice in the healthcare domain. As a result, agents-based systems are currently used in a wide series of medical-related problems (Moreno, 2003): patient scheduling, organ and tissue transplant management, community care, information access, decision support systems, home healthcare monitoring, medicine and device management, healthcare operators scheduling.
Agents coordinate and exchange information, have a representation of their goals, and determine their course of action on the basis of a model of the environment which is shared with other agents. As explained below, ontologies are actually recognized as the best instrument for defining such a model.

3.4 Ontologies

The advent of ontologies (Gruber, 1995) in computer science in the early nineties and the current effort in their settlement and development is making the "semantic interoperability dream" come true, i.e. ontologies are more and more recognized as the best suited instrument to harmonize complex and different sources of information and to get meaningful knowledge from them.
Ontology is the branch of philosophy that provides a formal foundation for specifying, and making statements about, what exists and how the things that exist are related to each other. In computer science ontologies are used to describe the logical structure of a specified domain of interest, its concepts and the relations among them, and to present them in an explicit and standardized way.
In order to encode knowledge in the form of an ontology an appropriate language must be used. The most recent emerging standard proposed by W3C is the so called Web Ontology Language (OWL) (W3C OWL, 2005). OWL represents reality as a set of classes, properties and instances. Classes stand for the concepts which must be encoded; instances are the concrete individuals populating classes, whilst properties (relationships) assert general facts about the members of classes and/ or about individuals. OWL provides also a certain number of operators (*constructors and restrictions*) for expressing complex concepts, constraints and dependencies which reflect the human model of reality.
An important feature of ontologies is that they can be processed by a reasoner, i.e. by a software tool able to extract new knowledge implied by explicitly codified knowledge. This is

very interesting for context aware computing. A simple example of reasoning can be given by the exploitation of "transitive relations", which are supported by OWL. Given three entities, such relations bring the conclusion of the existence of a relation between the first and third entity, when a relation between the first and second and between the second and third occurs. For example, the 'locatedIn' relation between an 'Entity' and a 'Location' is commonly marked as "transitive". As a result, if a user is 'locatedIn' the room 'Bedroom', which is in turn 'locatedIn' the 'Home', then the reasoner concludes that the user is located in 'Home'.
Indeed, context-aware systems require more advanced reasoning mechanisms. They are expected to be able to derive higher-level, conceptual context, such as "what the user is doing" from relevant low-level context, such as "the door is moving", which is provided by context sensing. This is the reason why ontologies are normally accompanied by rule based systems.

3.5 Rule based logic

Rule-based logic (Partridge, 1994) is an expressive way to define situations. Rule-based systems (RBS) differ from standard procedural or object-oriented programs as they do not clearly define the order in which code executes. Their behaviour is embedded in a set of *rules*, each of which encodes a small piece of codified knowledge and can be executed according to the current content of the so-called *working memory* (or agenda). The working memory stores the so-called *facts*, which represent the factual knowledge of the RBS.
Rules are similar to traditional if-then-else statements, but, differently from them, they are not executed in any predetermined order. They have a left hand side (the *if* part) and a right hand side (the *then* component). The left hand side contains information about facts and *objects* which must be true in order for the rule to potentially fire (i.e. execute).
The process of firing rules is carried out by the so-called rule engine (or *inference engine*)**:** a program (interpreter) that performs iteratively the so-called *Match-Resolve-Act* (MRA) cycle over the rule set and the working memory. In a MRA cycle, at any given time**,** rules are activated by matching them against the working memory elements (Match phase); then activated rules are chosen for execution according to some selection strategy (Resolve phase) and finally executed (Act or firing phase). In the Act phase, facts can be modified or deleted, and new facts can be added to the working memory as well.
In synthesis, RBSs select and execute available rules according to the current context they perceive. This makes RBSs particularly suitable for implementing context-aware architectures that must be modular and easily extensible: facts contain a symbolic and declarative description of a current situation and the rule set puts in relationship context conditions and devices' behavior and actions.

4. A real-life application: a context-aware infrastructure based on RFID sensor tags

4.1 Introduction

In this section, we propose a framework for context-aware pervasive systems built around the above mentioned technologies: RFID, sensors, ontology representation, multi-agent paradigm and rule-based logic.
The section is organized as follows. Section 4.2 introduces the general-purpose framework and its prototypal implementation. Section 4.3 synthesizes system implementation choices. Sections 4.4, 4.5 and 4.6 are centred around context modelling. First the ontology

representation, then the rule-based reasoning are illustrated. Section 4.7 focuses on data collection strategies. It describes "S-tag", a novel low cost device enabling the integration of sensor networks with RFID systems. A home-health scenario in the "emergency management" domain is finally provided in Section 4.8 as specific example of application and practical result.

4.2 The Architecture

The architecture of our system (Fig. 3) follows a widely accepted abstraction (Indulska, 2003) according to which context-aware systems are organized into three layers: context sources, context management middleware and context consumer level.

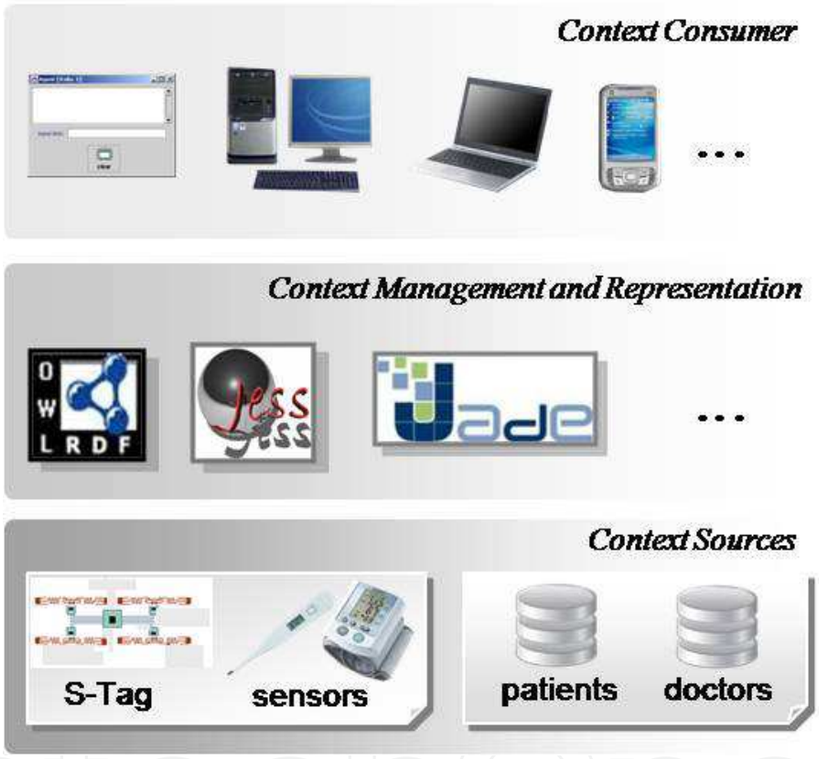

Fig. 3. Layered system architecture

Context sources include entities providing raw context data. They are conceptually partitioned into two groups: physical and virtual sources (Strang, 2004). Physical sources include hardware devices able to sense context data, such as RFID, sensors, positioning systems, etc. Virtual sources include software services able to gather context data, such as GUIs for user preferences input, databases, etc. Such data must be elaborated in an "intelligent" manner, so that the overall system reacts properly to context changes. This requires the availability of a machine-interpretable representation of context and of software components (*agents*) able to suitably process such knowledge. Both of them are conceptually situated at the intermediate level of the system architecture, the so-called middleware layer, as they provide the core building blocks of a context-aware application. Agents interoperate with one another thanks to the availability of a unified model of reality. Their behaviour is strongly influenced by data provided by context sources and substantially determines the

activities to be performed at the highest layer. Indeed, the context consumer layer includes all the entities, such as mobiles, Web interfaces, laptops, which interact with final users in response to meaningful context changes, thus determining the behaviour of the context-aware application as a whole.

Fig. 4 shows the fundamental components of the prototypal implementation of the framework. The system core is a team of cooperating agents, which share an ontology-based knowledge representation and reason by means of a rule-based inference engine.

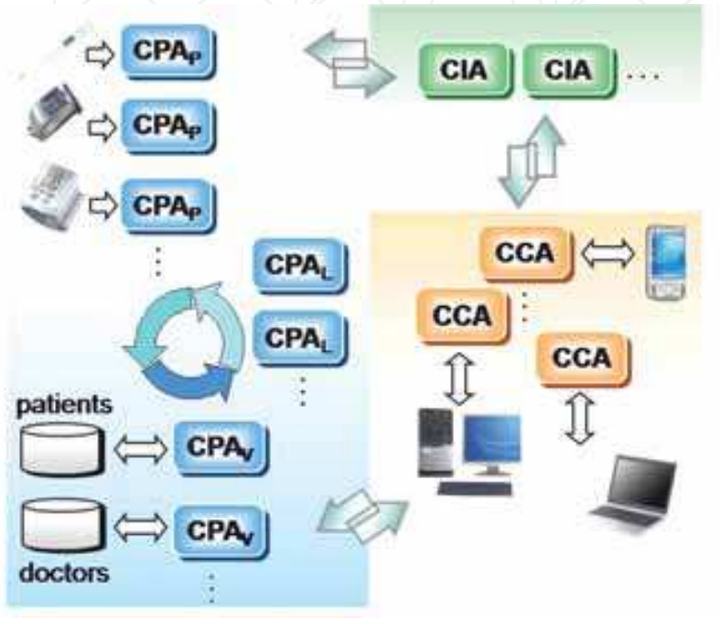

Fig. 4. The system follows a multi-agent paradigm and runs on distributed nodes connected by a local area network. Context Provider Agents (CPA) filter and integrate data provided by physical and virtual sensors. Context Interpreter Agents (CIA) identify the actions and the actors best suited for managing an emergency. Context Consumer Agents (CCA) perform the actions identified by CIAs

4.3 Implementation choices

Many ontology languages exist including Resource Description Framework Schema (RDFS) (W3C RDFS, 2004), DAML+OIL (DAML, 2010), and OWL (W3C OWL, 2005). OWL is a key to the Semantic Web and was proposed by the Web Ontology Working Group of W3C. It is much more expressive than other ontology languages such as RDFS. We chose OWL rather than DAML+OIL as the latter has been merged into OWL to become an open W3C standard. We took advantage of the Protégé (Protégé, 2010) graphical tool with the OWL plug-in and adopted Pellet (Pellet, 2010), a freeware, public domain ontology reasoner, to perform consistency checking.

The rule-based domain knowledge was implemented with Jess (Friedman-Hill, 2003) on top of OWL ontologies. Jess is a rule engine and scripting environment written in Java. Its rule engine uses the Rete algorithm (RETE, 2010) which has been shown to be fast and efficient especially for large data sets and it is widely used within a professional programmers' community.

Agents are implemented by using the Java Agent Development Environment (JADE).JADE (JADE, 2010) is a software framework to develop and run agent applications in compliance with the FIPA specifications (FIPA, 2010) for interoperable intelligent multi-agent systems. Inter-Agent communication is based on the FIPA ACL which specifies a standard message language by setting out the encoding, semantics and pragmatics of the messages.
The three technologies were glued together (Fig. 5) by taking advantage of a local adaptation of BeanGenerator (OntologyBeanGenerator, 2010), a Protégé plugin able to convert OWL codification into Java classes. As a result, the semantics of agent messages and their reasoning are built over OWL concepts and predicates, having been matched with Jess and JADE vocabulary (see next section).

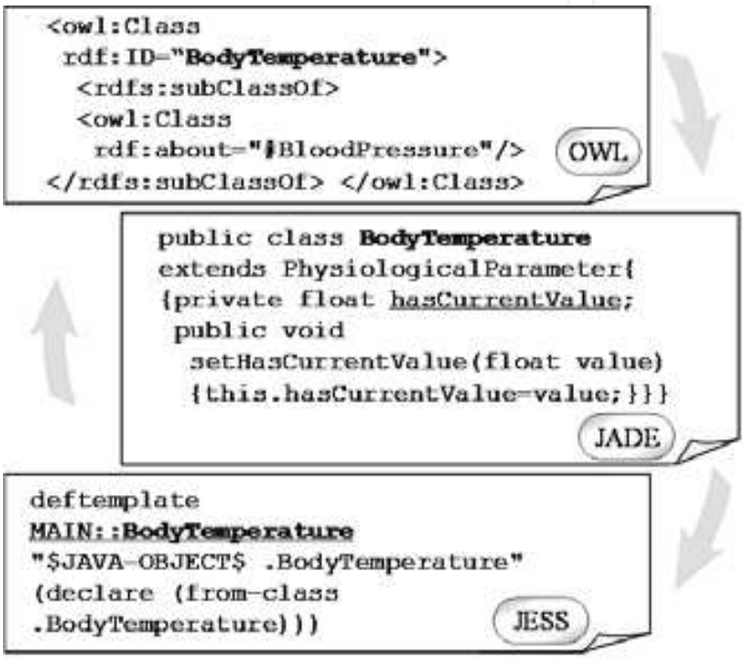

Fig. 5. The system implementation is based on the matching between OWL vocabulary with Jade agent inner context representation (in the form of Java classes) and Jess shadow facts codification

Finally, input data are provided by both physical and virtual context sources. Physical sources are obtained by integrating sensors with an RFID device (see Section 4.7), whilst virtual context sources consist in databases providing static information.
In the following sections, the system components and their way of functioning are described more in detail, with reference to a home-care giving scenario.

4.4 Context ontology

A common practise, when developing ontologies, is to adopt a *top level* (upper) shared conceptualization (Guarino, 1998) on top of which domain ontologies are built. Top level ontologies codify general terms which are independent of a particular problem or domain. Once the top level ontology is available, several *lower level* ontologies can be introduced, with the scope of incrementally specializing concepts from the high level generic point of view of the upper ontology to the low level practical point of view of the application. This

way of structuring knowledge promotes sharing and reuse of ontologies in different application domains.

The hierarchical architecture of our ontology is organized into three levels (see Fig. 6, left side). Indeed, the ontology of our framework provides the knowledge representation upon which agents reason and exchange messages. Therefore, the top level concepts of our taxonomy contain terms useful for codifying the multi-agent environment, thus facilitating their interoperation. These concepts can be reused in any agent-based environment, independently from the application domain.

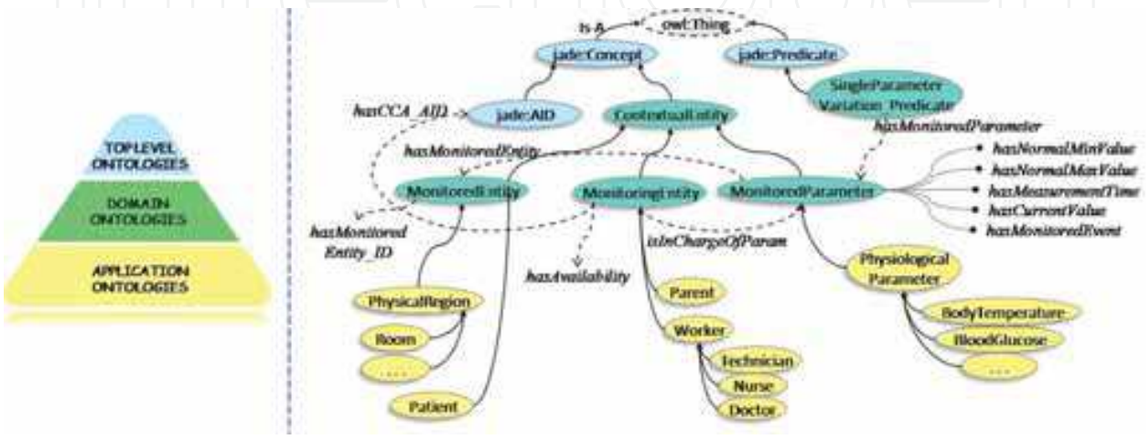

Fig. 6. A fragment (on the right) of the ontology, which is hierarchically structured in ontology modules (depicted on the left)

In order to fully exploit them in a context-aware system, a specialization is needed. In other terms, the context knowledge must be represented. This is done by implementing a "context middle level ontology" which specializes the top level ontology and codifies general concepts related to context. Terms specifically related to the application domain are introduced at the bottom level by making a further specialization. Some more details about the way our ontology is structured can be evinced by Fig. 6 (right side). Indeed, the top level ontology reflects the JADE codification of ACL message "slots", as represented in the publicly available ontology named "OWLSimpleJADEAbstractOntology" (JADE Tutorial, 2010). This ontology allows agents to give a semantics to message components. It makes a distinction between "predicates" and "terms". Predicates can assume "true" or "false" values and are used to describe the current status of codified entities (e.g. "*SingleParameterVariation_Predicate*"). Terms are specialized into "Agent Actions" (not depicted in Figure 4) and "Concepts" classes. Agent Actions codify the actions being performed by agents (such as "*NotifyAnomaly*"). Concepts are used to characterize the entities upon which predicates and actions operate (such as "*MonitoredParameter*", "*MonitoringParameter*", "*PhysicalRegion*", "*Worker*" and so on). Agents are themselves codified in the ontology by instantiating a suited subclass of the "Concepts" class, namely the "AID" class.

The vocabulary related to context entities was defined starting from the widely accepted definition of context, provided in (Dey, 1999): "Context is any information that can be used to characterize the situation of an entity." As a consequence, an entity is a person, place,

computational entity, or object which is considered relevant for determining the behavior of an application. Moreover, in order to satisfy the ontology reusability requirement, the "*MonitoredEntity*", "*MonitoredParameter*" and "*MonitoringEntity*" contextual entities have been introduced at domain level. On their turn, such entities specialize into different concepts depending on the context subdomain they belong to. For instance, a "*MonitoredEntity*" can be a patient, an object or an environment; a "*MonitoringEntity*" can be a relative of the patient or a health operator. Devices can be of different types as well.

4.5 Context rules

Jess rules are used to convert low-level information, given in a raw form by sensors, into high-level context. This is conceptually performed in an incremental fashion. When the system starts to work, the sensor network or other devices get data from physical world. Depending on the incoming events captured by sensors and context, the facts in the Jess *working memory* are updated. A set of rules determines if an alarm has to be triggered and which alarm level should be activated, according to measurement values and corresponding thresholds. Jess *rule engine* then searches automatically through the available combinations of facts to figure out which rules should be fired. Such rules, when matched, infer new facts which express the context switching to "situation of alarm". In other terms, the system acquires a sort of *anomaly awareness*, i.e. the raw data interpretation infers facts which express the occurrence of an anomaly.

In order to keep the system as general as possible (and easily customizable to different application scenarios) (Esposito, 2010), we adopted the so-called *normalized* rules (Williamson, 2010), i.e. rules that can be used in a variety of heterogeneous cases without changing their structure. For instance, rules concerning the identification of alarm events due to an anomaly behaviour of data provided by physical sensors have the same codification, independent from the specific monitored parameter.

The following example shows a rule activating an alarm when a parameter has a monitored value ("?par-c" variable) that exceeds the patient thresholds ("?par-max" and "?par-min" variables). When the rule is fired, the fact "Single Parameter Variation" ("?f-SPVp" variable) is inferred (as another predicate) and the action "notify that abnormal event" is activated ("send-single-alert" action). The "?pn" variable, finally, identifies the monitored parameter having generated the alarm.

```
(defrule verify-sensor-data
    ?f <- (SingleParameterMeasurement_Predicate
        (hasParamName ?pn)
        (hasParamCurrentVal ?par-c |: (> ?par-c ?par-max) |:
(< ?par-c ?par-min))
        (hasParamNormalMaxVal ?par-max)
        (hasParamNormalMinVal ?par-min)
        (hasParamMeasurementTime ?par-time))
      =>
; assert a Single Parameter Variation Predicate ?f-SPNp
      (send-single-alert ?f-SPVp))
```

The "anomaly awareness" facts may fire other rules, which may on their turn infer other facts. This determines the switching to the higher level context. In other terms, the context switches to a new situation, which we may call "procedure awareness", in which the activities to be performed in order to manage the alarm situation are known.

The following example shows how a rule is fired as a consequence of an abnormal status due to a couple of parameters. The rule infers the fact "Aggregated Parameter Variation" ("?f-APVp" variable) and starts the action "send-aggregate alert" in order to notify the anomaly to entities at a higher logical level.

```
(defrule aggregate-temporal-variations-for-2-params
 ?f0 <- (SingleParameterVariation_Predicate
         (hasParamName ?pn0)
         (hasParamMeasurementTime ?t0))
 ?f1 <- (SingleParameterVariation_Predicate
         (hasParamName ?pn1)
         (hasParamMeasurementTime ?t1))
 ?f-check <- (aggregation-pattern-for-2-params
        (pattern-time-width ?ptw &:
        (<= (abs(-?t0 ?t1)) ?ptw))
         (param-0 ?pn0) (param-1 ?pn1))
=>
; assert an Aggregate Parameter Variation predicate ?f-APVp
(send-aggregate-alert ?f-APVp))
```

Once that the procedures needed to manage the anomaly have been identified, the context consumers come into action by performing suited "anomaly management" actions.

As detailed in the following section, the kind of reasoning above described is carried out with the support of suited agents.

4.6. Agent-based reasoning

The following subsections respectively depict the multi agent organization of our system and an example of agents interaction.

4.6.1 The multi-agent framework

Fig. 7 shows the proposed multi-agent framework, which assigns three fundamental roles to agents:

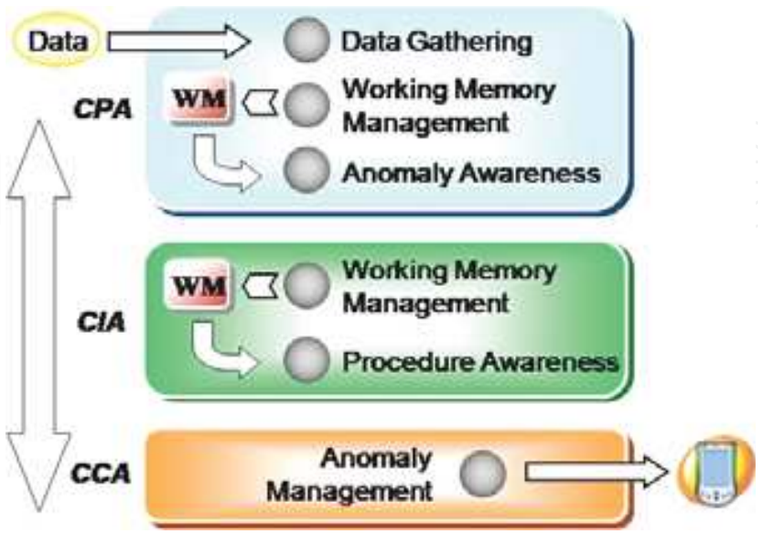

Fig. 7. The proposed multi-agent framework

Context Provider Agents (CPA). These agents wrap context sources to capture raw context data and instantiate the ontology representation. CPAs may encapsulate single sensors or multiple sources. In the former case ("single domain CPAs") they are mainly responsible for gathering and filtering data and info from sensor devices. In the latter case, (Dockorn Costa, 2005) they interact with single domain CPAs, in order to aggregate context information from various context sources (for instance sensed data must be aggregated with patient thresholds). Both kinds of CPAs are responsible also of making low level context inference and putting relevant context information into the rule engine as facts.

Context Interpreter Agent (CIA). Context Interpreter Agents are responsible for observing context changes sensed by CPAs, and, as consequence of these changes, to identify the set of actions that should be performed by context consumer agents. Substantially, they are responsible for identifying the monitoring entity best suited to manage an alarm situation, for contacting it and for forwarding context information to it.

Context Consumer Agent (CCA). Context consumer agents are responsible for performing the actions triggered by CIAs. Actions provide the application reaction to context information changes, which may assume diverse forms, such as the generation of a signal, the delivery of a notification or a web services request. Moreover, they may request further data to CPAs by contacting the suited CPA_v agent.

Registration Agents (RA). are in charge of managing agents registration and deregistration. Indeed, the system is dynamic in nature, as monitored and monitoring entities can be added and removed during its functioning. The JADE environment provides a service, namely the Directory Facilitator (DF), which supports runtime registration and service advertizing. However, DF has limited capabilities in service description and reduced performances. Therefore we implemented our own system by using the Jess virtual machine, which demonstrated to be much more efficient and supports a much richer description of agent capabilities and features.

4.6.2 Agents interaction

A simple example of agents interaction is provided in Fig. 8. A logical CPA, i.e. a CPA in

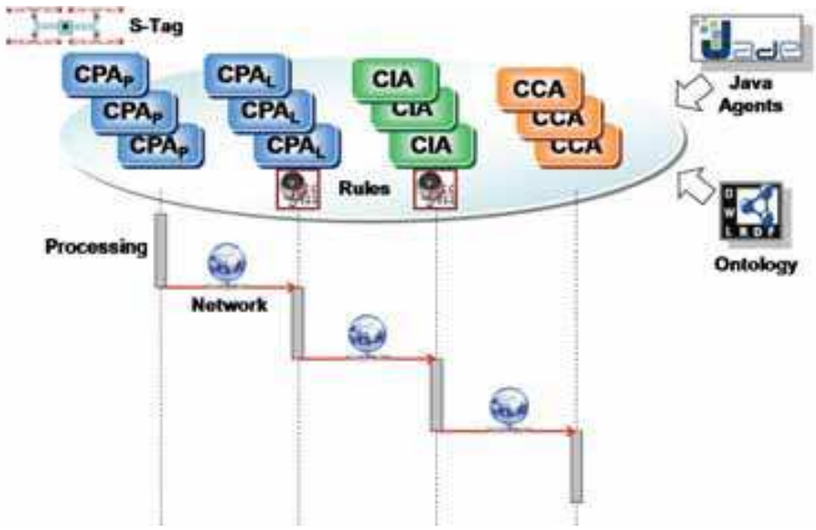

Fig. 8. A simple example of agents interaction

charge of aggregating data coming from different context sources (and their CPAs), gathers data coming from a sensor device and from the patient archive and identifies alarm events. When an alarm is issued, the CPA communicates the event to the CIA by sending an "inform" ACL message. The CIA, on its turn, requests information about the availability of physicians to the CPA being associated to the doctors archive. Finally, an "inform" ACL message is forwarded to the suited CCA in order to communicate the alarm to the selected doctor.

The code snippet below shows how ACL fields are perfectly aligned with the classes being codified in our "Ubiquitous Health-Care" (UHC) ontology (see Section 4.4). It contains an example of "inform" message sent by a physical CPA (namely "CpaP_BodyTemperature_Patient3" in the following code snippet) to a logical CPA (indicated as "CpaL_Patient3"). The UHC vocabulary is adopted to communicate sensed data of a patient (identified by his id "MonitoredEntity_ID" slot) measured at a certain timestamp ("timestamp" slot). The UHC "HasMonitoredParameter" predicate, together with its slots "NormalMinValue", "NormalMaxValue", "CurrentValue" (indicating the thresholds and the current sensed value of the monitored parameter), enrich the message with a deeper semantic meaning. Once that the message is interpreted, the identification of consequent actions is done by means of rules similar to those described in Section 4.5.

```
(INFORM
 :sender  (agent-identifier CpaP_BodyTemperature_Patient3)
 :receiver set ( agent-identifier CpaL_Patient3) )
 :content
  "( (HasMonitoredParameter (BodyTemperature
            :NormalMinValue 35.5
            :NormalMaxValue 36.5
            :CurrentValue 39.5))
    (MonitoredEntity_ID :Patient3)
    (timestamp 1205453552515) )"
:language  fipa-sl  :ontology  UHC)
```

4.7 Context sources

As previously stated, the input raw data of the proposed architecture is represented by the set of values, usually physical parameters, collected by the so-called physical sources. It has also been outlined that the realization of devices that integrate RFID technology with generic sensors has numerous advantages in terms of cost and easiness.

Indeed, the scheme proposed in Fig. 9 represents the actually designed and realized general purpose Sensor-Tag (S-Tag) connected to a generic sensor including, among the others, sensors for biomedical applications (patented). When Tag and sensor are connected one to another and are interrogated, the measured value can be read by a standard UHF RFID system and interpreted through an RFID middleware. In such a way, the system takes advantage from the standard RFID technology, preserving most of its peculiarities and maintaining the compatibility with all the devices already available and worldwide standardized.

The working principle is as easy as effective: data measured by the sensor connected to the S-Tag is used to control a set of integrated RF switches which select the appropriate IDs to be transmitted back towards the reader when the S-Tag is interrogated. Actually, as the S-

Tag is designed as a general-purpose device, the sensor signal must be preventively quantized through an ad-hoc circuit embedded in the sensor itself. Of course, the number n of bits used for the quantization must be equal to the number of available IDs on the S-Tag. As a result, the number of different combinations, excluding the one with all the bits equal to zero (which would cause interpretability problems) is $2^n - 1$.

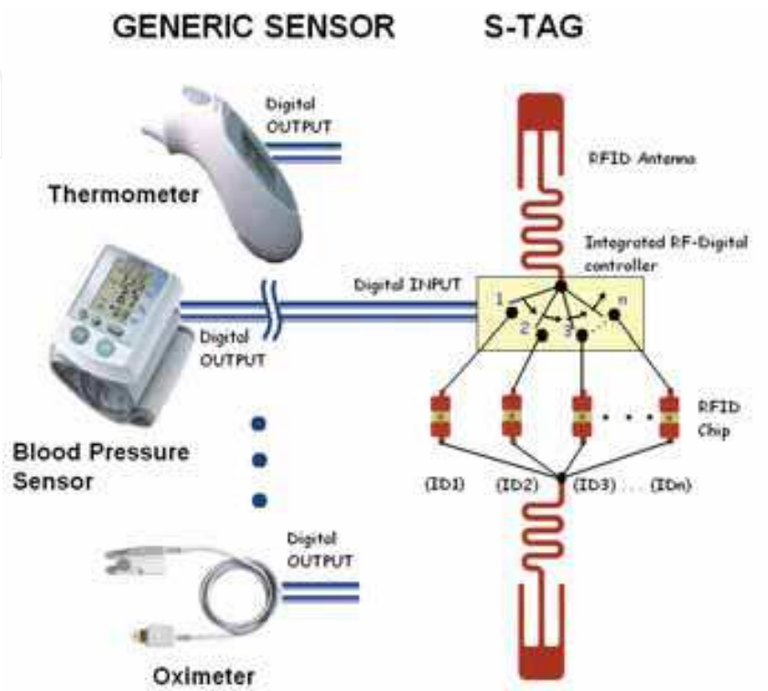

Fig. 9. A simplified scheme of the designed RFID sensor tag

Consequently, when the Tag is in the region covered by the RFID reader, it sends back a signal containing a proper combination of identity codes (IDs) depending on the value of the input itself, thus facilitating the transmission of sensor data. More specifically, the internal microwave circuit of the S-Tag samples the value at its input (which has been measured by the sensor) and quantizes it by using a number of bits equal to the number of available different IDs. For each bit with value equal to 1, the integrated micro-switch selects the corresponding ID to be transmitted; the combination of bits can be hence received by a standard RFID reader and easily decoded in order to rebuild the sensor-measured data.

For a better clarification of the working principle of the S-Tag, Fig. 10 reports a block diagram referred to a temperature sensor applied to a 4-ID S-Tag. The sensor is supposed to work in the range 35-42 °C and to be provided with a 4-bit binary digital output (15 valid combinations), guaranteeing a resolution of 0.5 °C. The measured temperature, for instance assumed to be 38.3 °C, after the quantization and the digitalization, will correspond to the binary symbol "0110", that will become, hence, the input signal of the S-Tag. Consequently, the internal logic of the S-Tag enables the transmission of the two IDs associated to the second and the third bit (those ending respectively in 752 and 753 in the figure), and disables the transmission of the others. When an RFID reader interrogates the S-Tag, hence, it receives back only the enabled IDs which are automatically associated to the logic state "1". The non-received IDs, instead, are associated to the logic state "0". The so-reconstructed

symbol is then reconverted, resulting in temperature value of 38.5 °C, slightly different from the actually measured one because of the quantization effect.
This is not the most adequate context for a more exhaustive explanation of the implementation issues of the S-Tag; we only would like to observe that, as apparent from the picture, the sensor is an external unit. In such a way, generic sensors, with the only requirement of a specific digital output, can be used. Such sensors are not integrated into the S-Tag, so that they do not influence the tag cost. Moreover, thanks to an accurate electromagnetic design of the tag antenna and of the microwave circuit (microcontroller, RF-switch and so on), also the implemented technological innovation is reasonably inexpensive.

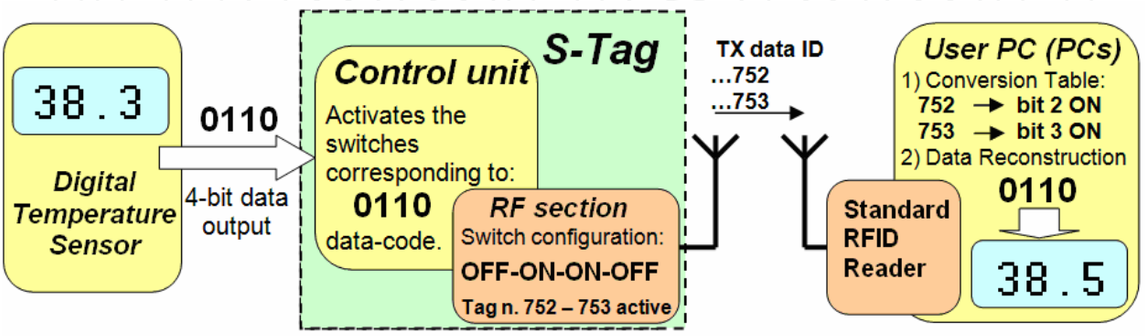

Fig. 10. Example of temperature data transmission through a 4-ID S-Tag. The binary symbol at the S-Tag input is used to discriminate the IDs to be transmitted. Vice versa, the information about whether or not an ID has been received by the RFID allows a straightforward symbol reconstruction. The difference between measured and received temperature value is due to the quantization

4.8 Results

This section provides a number of results obtained from a benchmarking phase performed to test performance and efficacy of both the S-Tag component (subsection 4.8.1) and the overall software platform fed by S-Tag data (subsection 4.8.2).

4.8.1 Benchmarking the S-Tag

In this section the proposed S-Tag is extensively tested in order to demonstrate its effectiveness and versatility. All the reported results have been obtained by using a 4-ID S-Tag receiving in input the four binary signals of the quantized measured value.
In the first presented results, the signal at the S-Tag input has been artificially generated in order to create extremely stressing conditions. Indeed, each of the four binary inputs is assumed to vary every two seconds at random. The random signal generator, hence, can generate 2^4 different uncorrelated combinations, thus covering the whole dynamic range of the S-Tag.
In such a situation, the S-Tag has been positioned at different distances (varying from 50 cm to 4 m) from a standard UHF RFID reader antenna (for instance, the Alien ALR-8610-AC). For each distance, the random signal generator has been connected to the S-Tag for a testing time of 10 minutes, in which the RFID reader has collected data and the signal has been reconstructed via an ad-hoc but very uncomplicated software. The difference between actual and received signal has then been evaluated.

In Fig.11, for instance, transmitted and received samples are compared at a Reader-Sensor distance of 3 m. It can be observed that, despite the presence of one spike, the transmitted signal is quite perfectly reconstructed. It is worth mentioning, however, that the longer is the interrogation distance, the highest is the number of recorded spikes. The spikes correspond to erroneous received samples which, obviously, should be reduced as much as possible. Because of the strong correlation between consecutively measured values in real situations, however, software filters capable to correct most of the read erroneous values can be easily implemented and applied. For a detailed quantification of the performance of the system, hence, the circled-dotted line in the graph of Fig. 12 reports, for different distances, the percentage error evaluated as ratio between number of erroneous samples and number of samples. A sample is a transmitted symbol of four bits and it is considered erroneous if at least one bit is not correctly received. Many observations result from the graph. For distances superior to 3.5 m, for instance, the measured signal cannot be correctly reconstructed.

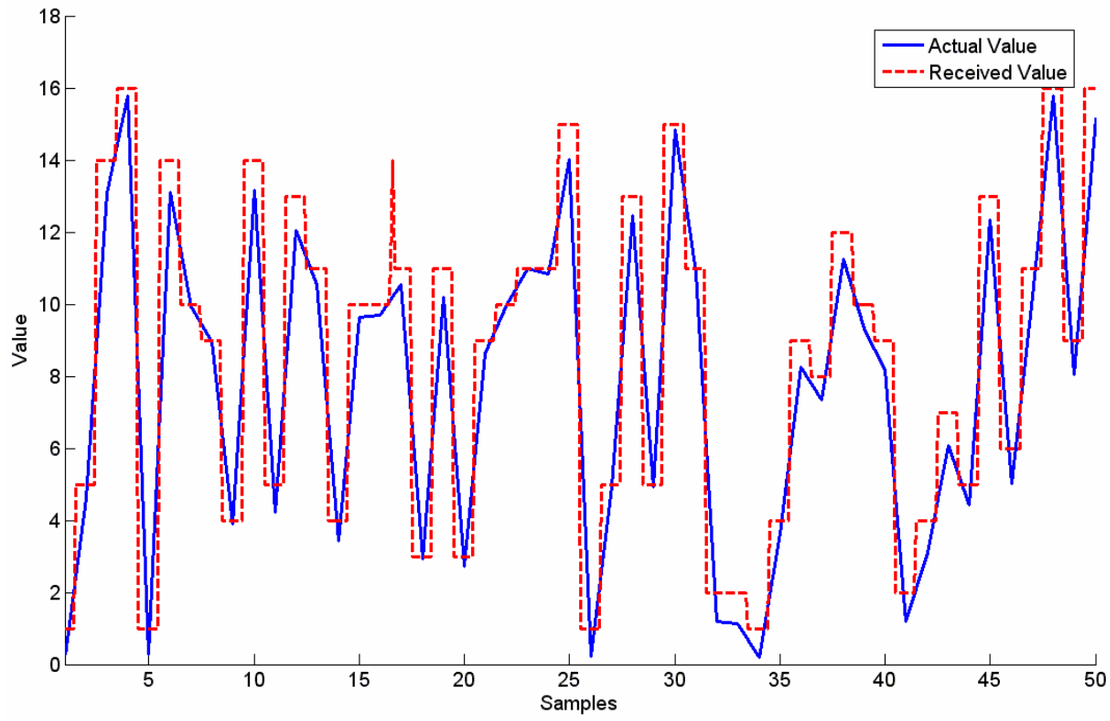

Fig. 11. Comparison between the actual random signal at the input of the S-Tag and the signal received by a RFID reader at a distance of 3 m

Even though passive RFID Tags can be read even at 9 m from the reader, in fact, the slight mismatch caused by the microwave circuits and the coupling among the S-Tag antennas, reduce the reading range. Work is in progress in order to improve such issue. For distances inferior to 3.5 m, however, the percentage error is definitely negligible, especially if it is considered that no software filter is applied to the received bits. In such a context, the squared-continuous line in the same Fig. 12 shows how the accuracy is improved after the use of a very simple filter that detects and removes the spikes. Since the RFID reader interrogates the S-Tag many times a second, the filter verifies the eventual difference among

samples referred to the same transmitted value, and where only one reading differs from the others, it corrects the erroneous sample.

In order to verify that the S-Tag can be used with generic sensors, once disconnected from the random generator, it has been connected to a glucometer. In this case, it is expected an improving of the S-Tag performance. It is worth recalling, in fact, that the physical parameters potentially monitored, show much slower variation in the considered time intervals than the previously used random generator. In the same Fig. 12, for instance, the reported results refer to the blood glucose monitoring in the range 4 – 12 mmol/ l. More specifically, results have been obtained by loading into the used artificial glucometer some realistic glycemia values of a diabetic patient recorded during a whole day. For such a goal, glucose values to be transmitted have been obtained from AIDA web site (AIDA, 2010), that is an on-line educational simulator designed to assist interested parties in obtaining a better understanding of how insulin and glucose interact within the body (Blanchard, 1998). More specifically, one of the sample cases proposed on the AIDA web site (the number 0015 referring to the virtual patient named "Edward Carllson"), has been chosen, and the virtual patient's blood glucose values have been charged on the artificial glucometer connected to the S-Tag. Results of Fig. 12 clearly show the optimal behavior of the system even at the maximum investigated distance both for the non-filtered and filtered received signals.

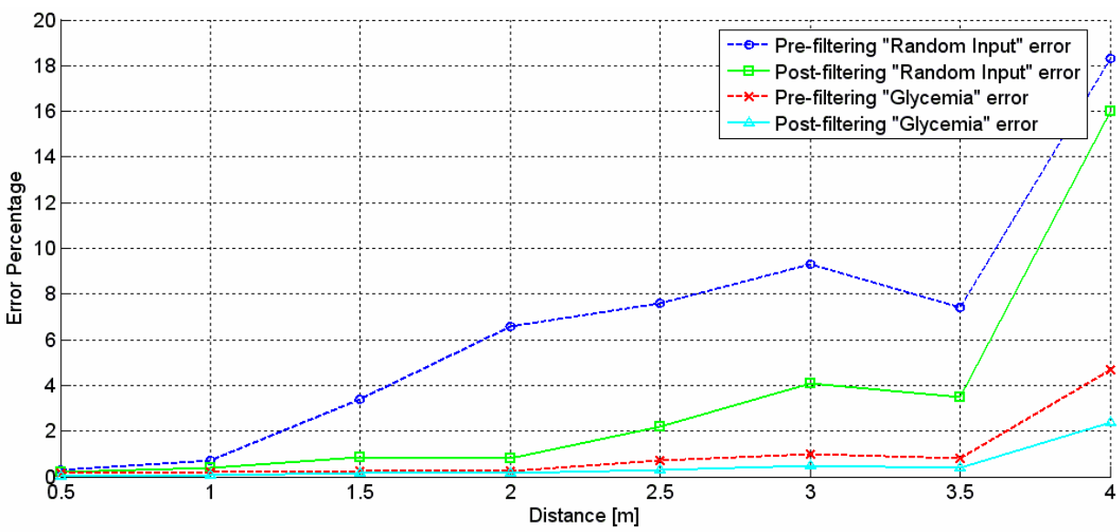

Fig. 12. Comparison among the evaluated error percentages by using two different inputs (random signal and glycemia) and by varying the distance between S-Tag and reader antenna . The same error is also reported after the software filter is applied

Finally, a more practical mobile scenario has been considered by positioning the S-Tag on a programmable robot that, moving itself in the RFID covered area, simulates patient motion. The robot has been programmed to alternate moments of stillness and moments of activity for a total test time of 24h, during which the blood glucose values are transmitted through a S-Tag each 15 minutes. After the reader has received the S-Tag signal, the blood glucose values are reconstructed both with and without the filtering process. In Fig. 13, for instance, the three reported graphs are referred to three types of signals: a) original transmitted blood glucose trend, b) reconstructed signal without filtering, and c) filtered signal.

Results show the optimum behavior of both non-filtered and filtered signals and demonstrate the very good matching between original transmitted blood glucose trend and received reconstructed signal. The slight differences among signals are due to the quantization, as described above.

In addition to the glucometer many other kinds of sensors have been tested and the S-Tag used as a source data for elementary control systems, thus showing the S-Tag capability to effectively transmit data measured by different kinds of sensors towards standard RFID readers.

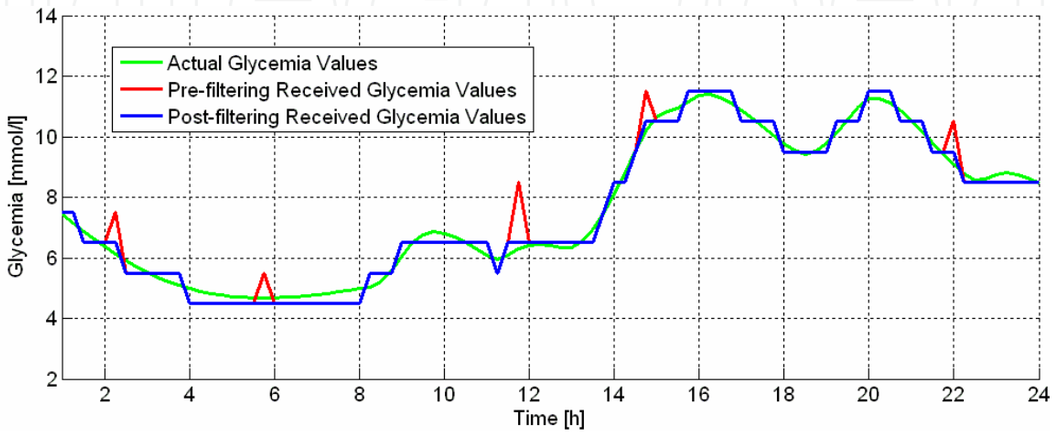

Fig. 13. Comparison of the actual glycemia values at the input of the S-Tag with the received non-filtered and filtered values

4.8.2 Benchmarking the overall system

The overall system was tested in the framework of a network of computers connected by a TCP/ IP-based 100Mb/ s network. Computers run under Linux operating system and have the following characteristics:

1. Intel Pentium, 3.00 GHz (single core); RAM: 512 MB;
2. AMD Athlon, 1.8 GHz (single core); RAM: 478 MB;

The experimentation consisted in collecting average alarm generation and transmission times in diverse configurations (Fig. 14). The simplest consists of a single patient monitored by one sensor and assisted remotely by four doctors. The most complex consists of 5 patients, each being monitored by 4 sensors and being assisted by 4 doctors. Results concern average times obtained over 30 runs. The system monitors data sensed by sensors, and, based on patient info (loaded once at system start-up) determines alarm events. Alarm events trigger actions which basically consist in identifying the best suited doctor and contacting him/ her (via mobile). Such doctors are the ones having the appropriate specialization to tackle the out-of-range monitored parameters (i.e. cardiologists for alarms concerning heart-rate and/ or blood pressure) and the ones being available in the time interval nearest to the alarm measurement time.

Measured times were taken by imposing four "stressing conditions" that simulate a real scenario: 1) each sensed parameter has always out-of-range values, 2) in the same run, all sensors send data at the same time, 3) the time-spacing between consecutive measurements is equal to two seconds, 4) only one CIA has been activated, in order to create a fictitious bottleneck in the system. Measured times consist in:

- the time elapsing at the CPA_P side between the reception of data from the sensor and the triggering of the corresponding alarm obtained by means of rule-based reasoning;
- the time elapsing at the CPA_l side between the reception of a message coming from CPA_P and the triggering of the corresponding alarm obtained by means of rule-based reasoning;
- the time elapsing at the CIA side between the reception of a message coming from CPA_l and the forwarding of a message to CCA.

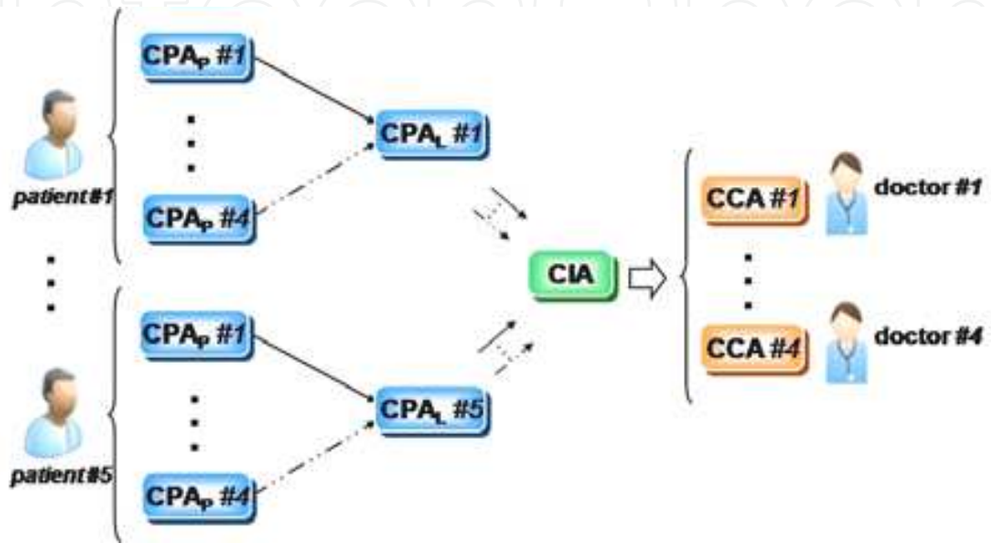

Fig. 14. Experimentation setup

As shown in Table I, CPA_P are responsible for filtering data coming from a single sensor, therefore their times do not change when the number of sensors and/ or of patients change. Each CPA_l, instead, elaborates data coming from sensors connected to a single patient. In our experimentation set-up, they elaborate data coming from up to four sensors and produce two kinds of aggregate alarms: the former associates "BodyTemperature" and "BloodGlucose" patterns, the latter combines "BloodPressure" and "HeartRate" patterns. Therefore, CPA_l times increase with the number of sensors attached to single patients. CIA times, finally, depend both on the number of sensors and on the number of patients (i.e. both on the number of CPA_P and on the number of CPA_l). Table 1 also illustrates data obtained by aggregating the average times obtained for each configuration. As shown in the table, processing times are very good, thus demonstrating the amenability of the proposed approach to attach also more complex monitoring problems.

Agent	1 sensor 1 patient	2 sensors 1 patient	2 sensors 2 patients	4 sensors 1 patient	4 sensors 2 patients	4 sensors 5 patients
CPA_P	3.68	3.68	3.68	3.68	3.68	3.68
CPA_L	/	4.74	4.74	5.96	5.96	5.96
CIA	2.24	2.87	2.93	3.28	6.97	24.49
Ttot	5.92	11.29	11.35	12.92	16.61	34.13

Table 1. Performance results [ms]

5. Conclusions

Context-aware ubiquitous systems open a very large number of appealing perspectives for healthcare. They promise huge increases in performance, accuracy and availability of treatment. The range of application domains potentially benefiting from these technologies is wide and variegate, as it includes different areas, such as remote monitoring of mobility-impaired patients and support to information flow among heath-care workers.

In order to facilitate the implementation of this kind of systems, scientific community is working hard to define standards and to implement middleware solutions. In spite of that, the available systems still lack in flexibility and often are not accompanied by a well documented characterization of customization effort and/ or of performance data. A number of enabling technologies seems to provide a solid building block for the design and development of flexible and robust context-aware pervasive systems. They were briefly resumed and commented in this chapter, together with a framework for context-aware computing based on such technologies.

The system harmonizes agents, ontologies and rule-based inference engines. The ontology provides the knowledge codification needed to support both agent reasoning and communication. The system elaborates data coming from heterogeneous sources, including an innovative RFID system, which embeds RFID and sensors in a reasonably inexpensive tag, thanks to an accurate electromagnetic design of the tag antenna and of the microwave circuit (microcontroller, RF-switch and so on).

The effectiveness of the system is demonstrated by reporting performance results obtained from validation in a real-life application in the home-care scenario.

The very promising results, in terms of easiness of use, reconfigurability and performance, make the proposed approach a very good candidate for the solution of even complex monitoring problems, belonging to the healthcare domain and generally to other scientific domains provided that an ontology describing them is available.

6. References

AIDA homepage: on-line free Web-based diabetes software simulator. (last accessed: 2010, September).

Bacheldor, B. (2008, February 20th). Brigham and Women's Hospital Becomes Totally RTLS-enabled". *RFID Journal* .

Bacheldor, B. (2009, May 29th). Philly Hospital Uses RTLS to Track Patient Flow, Care and Training. *RFID Journal* .

Blanchard, S., Novotny, C., DeWolf, D., Lehmann, E., Gotwals, R. J., & Rorhbach, R. (1998). AIDA On-Line: a Glucose and Insulin Simulator on the WWW. In: *Proceedings of the 20th Annual International Conference on the IEEE Engineering in Medicine and Biology Society*, 3, p. 1159-1162.

Bradshaw, J. (1997). An Introduction to Software Agents. In: *Software Agents* (p. 3-46). Cambdrige, MA, USA: MIT Press, J.M. Bradshaw Editor.

Cho, N.; Song, S.-J. et al. (2005). A 8-μW, 0.3 mm² RF-Powered Transponder with Temperature Sensor for Wireless Environmental Monitoring. In: *IEEE International Symposium on Circuits and Systems*, pp. 1-17.

DAML homepage. DARPA Agent Markup Language: http:/ / www.daml.org (last accessed: 2010, July).

Dey, A. a. (1999). Toward a Better Understanding of Context and Context-Awareness. In: *GVU Techincal Report*.

Dobkin, D. (2007). The RF in RFID: Passive UHF RFID in Practice. Newton, MA, USA: Newnes.

Dockorn Costa, P., & Ferreira Pires, L. a. (2005). Architectural Patterns for Context-Aware Services Platforms. In: *2nd International Workshop on Ubiquitous Computing (IWUC2005) in conjunction with ICEIS2005*. Miami, FL, USA.

Esposito, A., & Tarricone, L. a. (2010). A Versatile Context-Aware Pervasive Monitoring System: Validation and Characterization in the Health-Care Domain. In: *Proceedings of IEEE International Symposium on Industrial Electronics (ISIE2010)*, (p. 2791-2796). Bari, Italy.

FIPA repository. The Foundation for Intelligent Physical Agents: http:/ / fipa.org/ repository/ index.html (last accessed: 2010, September).

Friedman-Hill, E. (2003). *Jess in Action*. Manning.

Gruber, T. (1995). Toward Principles for the Design of Ontologies Used for Knowledge Sharing. In: *International Journal of Human Computer Studies* (43), 907-928.

Guarino, N. (1998). Formal Ontology and Information System. In: *N. Guarino Editor, Proceedings of the 1st International Conference on Formal Ontologies in Information Systems* (FOIS98) (p. 3-15). Trento, Italy: IOS Press.

Indulska, J., & Sutton, P. (2003). Location Management in Pervasive Systems. CRPITS03 – In: *Proceedings of the Australasian Information Security Workshop*. 21, p. 143-151. Adelaide (Australia): Johnson, C.; Montague, P and Steketee, C.

JADE homepage. Java Agent Development Environment: http:/ / jade.tilab.it (last accessed: 2010, September).

JADE Tutorial webpage. Application-Defined Content Language and Ontologies. From: *JADE website:* http:/ / jade.tilab.com/ doc/ tutorials/ CLOntoSupport.pdf (last accessed: 2010, September).

Moreno, A. a. (2003). Application of Software Agent Technology in the Health Care Domain. Basel, Switzerland. Birkäuser Verlag.

OntologyBeanGenerator webpage. From: *Protégé Wiki homepage*: http:/ / protege.cim3.net/ cgi-bin/ wiki.pl?OntologyBeanGenerator (last accessed: 2010, September).

Paganelli, F. & Giuli, D. (2007). An Ontology-based Context Model for Home Health Monitoring and Alerting in Chronic Patient Care Networks, In: *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)* 0-7695-2847-3/ 07.

Partridge, D. a. (1994). Knowledge Based Information Systems. Mc-Graw Hill.

Pellet homepage. http:/ / www.mindswap.org/ 2003/ pellet/ (last accessed: 2010, September).

Protégé homepage. http:/ / protege.stanford.edu (last accessed: 2010, September).

RETE Algorithm webpage. From: *JESS website:* http:/ / herzberg.ca.sandia.gov/ jess/ docs/ 52/ rete.html (last accessed: 2010, September).

SM4ALL homepage. http:/ / www.sm4all-project.eu/ (last accessed: 2010, September).

Strang, T. a.-P. (2004, September). A Context Modeling Survey. In: *Workshop on Advanced Context Modeling, Reasoning and Management*, (p. 33-40). Nottingham, England.

Swedberg. (2010, May 28th). Nyack Hospitals Tracks Medication Compliance. *RFID Journal*.

W3C. Recommendation 10 webpage. (2004, Febraury 10th). From: *RDFS (RDF Vocabulary Description Language 1.0: RDFS Schema) webpage*: http://www.w3.org/TR/rdf-schema (last accessed: 2010, September).

W3C. Web Ontology Language Overview, W3C Recommendation webpage (2005, February 10th). http://w3.org/TR/2004/RDC-owl-features-20040210 (last accessed: 2010, September).

Walker, J., Pand, E., Johnston, D., Adler-Milstein, J., & Bates, D. a. (2005, January 19). The Value of Healthcare Information Exchange and Interoperability. In: *Health Affairs, Web Exclusive*.

Wang, S.-W. et al. (2006). RFID Application in Hospitals: a Case Study on a Demonstration RFID Project in a Taiwan Hospital. In: *Proc. of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, 08, p. 184-194. Kauai, Hawaii (USA).

Weiser, M. (1991). The Computer for the 21st Century. In: *Scientific American - Special Issue on Communications*.

Williamson, J. JESS Wiki: Keep Your Rules Normalized. From: *JESS website*: http://www.jessrules.com/jesswiki/view?KeepYourRulesNormalized (last accessed: 2010, September).

Biomedical Engineering, Trends in Electronics, Communications and Software

Edited by Mr Anthony Laskovski

ISBN 978-953-307-475-7
Hard cover, 736 pages
Publisher InTech
Published online 08, January, 2011
Published in print edition January, 2011

Rapid technological developments in the last century have brought the field of biomedical engineering into a totally new realm. Breakthroughs in materials science, imaging, electronics and, more recently, the information age have improved our understanding of the human body. As a result, the field of biomedical engineering is thriving, with innovations that aim to improve the quality and reduce the cost of medical care. This book is the first in a series of three that will present recent trends in biomedical engineering, with a particular focus on applications in electronics and communications. More specifically: wireless monitoring, sensors, medical imaging and the management of medical information are covered, among other subjects.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Luca Catarinucci, Alessandra Esposito, Luciano Tarricone, Marco Zappatore and Riccardo Colella (2011). Smart Data Collection and Management in Heterogeneous Ubiquitous Healthcare, Biomedical Engineering, Trends in Electronics, Communications and Software, Mr Anthony Laskovski (Ed.), ISBN: 978-953-307-475-7, InTech, Available from: http://www.intechopen.com/books/biomedical-engineering-trends-in-electronics-communications-and-software/smart-data-collection-and-management-in-heterogeneous-ubiquitous-healthcare

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen