

An infrastructure for smart hospitals

Gennaro Della Vecchia · Luigi Gallo ·
Massimo Esposito · Antonio Coronato

Published online: 7 January 2011
© Springer Science+Business Media, LLC 2011

Abstract In healthcare facilities most of the daily activities require strict coordination between clinicians, who often operate under heavy workloads and minimal workforce conditions in environments filled with increasingly complex technology. Ubiquitous Computing applications constitute a suitable solution for both reducing medical costs and improving patient safety by better supporting clinical processes. In this study we introduce an intelligent infrastructure for smart hospitals which implements basic services to optimize medical staff/patient interactions and grants ubiquitous and transparent access to clinical data stored in standard clinical databases. This infrastructure relies on the integration of Radio Frequency IDentification (RFID) and photosensor technologies to identify, locate and track clinicians and patients equipped with mobile devices and wearable RFID tags.

Keywords Pervasive access · eHospitals · RFID systems · Photosensors · Context-aware services · Service Oriented Architecture

1 Introduction

Recent developments in new ICT technologies are pushing the definition of new models and patterns for processes and services, where the wide diffusion of wireless technologies, along with new devices and sensors, is paving the way towards better

G. Della Vecchia · L. Gallo (✉) · M. Esposito · A. Coronato
ICAR-CNR - National Research Council of Italy, Via Pietro Castellino 111, 80131, Naples, Italy
e-mail: luigi.gallo@na.icar.cnr.it

G. Della Vecchia
e-mail: gennaro.dellavecchia@na.icar.cnr.it

M. Esposito
e-mail: massimo.esposito@na.icar.cnr.it

A. Coronato
e-mail: antonio.coronato@na.icar.cnr.it

and cheaper applications. Among the domains in which such achievements can be profitably exploited, healthcare has turned out to be one of the most suitable since it characterizes a domain where the importance of making correct decisions based on obtaining the right information at the right time is absolutely critical [25].

Pervasive healthcare is an emerging discipline which consists in the application of wireless, mobile and intelligent technologies to healthcare. It is related to the development of applications of up-to-date pervasive computing technologies—ubiquitous computing, context-aware computing, ambient intelligence, etc.—for healthcare, health and wellness management [6], with the definitive goal of making healthcare available to anyone, anytime, and anywhere according to the original vision of the pervasive computing paradigm. A wide range of innovative applications can then be conceived, from the remote monitoring of elderly people or ill patients to new environments like advanced surgery rooms, intelligent spaces for medical consultations or *smart hospitals*.

Nowadays, traditional hospitals are filled with increasingly complex technology, and most of the daily activities require coordination and collaboration between practitioners who are becoming ever more mobile and technology savvy. Indeed, evidence has been shown that pervasive healthcare applications, besides making expert care more accessible and personalized, can dramatically reduce medical costs and improve the quality of service [6, 28]. Moreover, hospital staff are often working under heavy workloads and with a minimal workforce. Clinicians have to manage a large amount of data that needs to be enriched with contextual information to be profitably used. Therefore, it is clear that pervasive healthcare applications constitute a suitable solution both to improve patient safety by better supporting clinical processes and to provide just relevant data to different medical practitioners by enriching and filtering that information.

Furthermore, it has recently been observed and reported that the duties performed by hospital workers are usually fragmented due to the fact that they need to track people or documents necessary to accomplish their activities, or to take care of emergencies [15]. This fragmentation forces hospital workers to continuously switch between tasks. For instance, while nurses are monitoring a patient's condition, they constantly need to move throughout the hospital premises looking for clinical files, thus interrupting the activity being performed to gather information. These aspects further increase the demand for high quality healthcare service at anytime and for anyone regardless of location and other constraints. By giving medical professionals appropriate and complete information, it is expected that the medical staff will be able to deliver better care that is tuned not only to the situation but also to the patient's history.

Nonetheless, we are still far from a productive application of pervasive computing in daily clinical practice. By its very nature, pervasive healthcare requires appropriate enabling technologies dependent on the distinctive features of the environment. Ideally, a seamless integration between the different devices should allow to view the system as a whole, therefore simplifying the development of high level applications. In fact, many segmented solutions, which do not interoperate well, are commonly found in hospitals. As a consequence, the gap between the technological reality and the application requirements may be considered the most limiting factor for a widespread deployment of pervasive applications in clinical scenarios.

In this study we introduce an intelligent infrastructure for smart hospitals which provides basic services to optimize medical staff/patient interactions and grants ubiquitous and transparent access to clinical data stored in standard clinical databases.

The rationale which drove the design of this infrastructure derived from the basic concept of allowing a context-aware access to the underlying environmental resources of a typical healthcare facility while keeping in mind feasibility, flexibility and extendibility. In more detail, our research efforts have been primarily focused on the design of a service-based framework that is capable of easily interoperating with legacy software and databases usually present in hospitals. Even though the approach we propose can be easily extended to integrate different low level technologies, in this work we have considered Radio Frequency IDentification (RFID) and photosensor technologies to identify, locate and track doctors, patients and devices.

The remainder of the paper is organized as follows. Section 2 introduces some related work in the field of pervasive healthcare, particularly focusing on RFID-based systems. Section 3 describes the infrastructure architecture and details the main features of services and systems. Section 4 describes the implementation and testing of the infrastructure in a real-world scenario. Finally, section 5 concludes the paper.

2 Related work

In recent years, many pervasive systems for healthcare have been proposed, discussed and realized. Pervasive healthcare is highly multi-faceted, with many applications focusing on the interoperability with legacy hospital assets, the security and privacy of confidential information and the usability of user interfaces.

Among the solutions focusing on interoperability, it is worth mentioning HERMES [22], which is a distributed middleware following a type- and attribute-based publish/subscribe model in an event-based architecture; however, rather than concentrating on supporting heterogeneous hardware and on providing low level services to simplify the design of pervasive applications, this work deals with routing algorithms and fault-tolerance mechanisms to develop scalable and robust distributed systems. Satoh, in [23] and [24], presents a general-purpose infrastructure to build and manage location-aware applications in ubiquitous computing settings, with the goal of providing people, places, and objects with computational functionalities to support and annotate them. In the same context, middleware mechanisms have been proposed to handle pervasively the execution of user-defined tasks through a network of heterogeneous mobile resources [9].

In [4], Bång et al. present NOSTOS, a paper-based UbiComp healthcare environment to support data capture and collaboration; it combines several technologies to enhance common paper-based clinical work environments; in this work the focus is on the interface, since the authors consider the paper-based interaction as the most productive one in healthcare scenarios. Different sensors, such as wireless accelerometers and infrared cameras, have also been used in medical scenarios to provide clinicians with novel inspection methods of medical data visualized on large screen displays [17], both with uni-modal and multi-modal interfaces [16].

With the goal of achieving a transparent integration of smart sensors, a middleware that supports component-based systems and event-based communication has been proposed in [7]; however, such a solution does not consider spatial and random relationships between events necessary to build context-aware services. MOBILEWARD [27] is a context-aware mobile prototype aimed at supporting work tasks in hospital wards; this prototype exhibits a certain degree of context awareness since it is able to recognize the position of nurses in the hospital to give them information and functionality. In [11], an approach based on the semantic integration of different positioning systems suitable for clinical environments is described; it also enables reasoning on location information in order to obtain higher context information or to resolve inconsistencies or conflicts due to sensing errors.

Using RFID technologies in health facilities is not a novel idea. In the past few years several pilot research projects, investigating the feasibility of RFID for tracking patients, assets, pharmaceuticals and personnel, have been conducted in hospitals [2, 3]. In [21] and [1], the general requirements for introducing RFID technology in healthcare environments are described and discussed. To address the demand of clinicians for a fast access to information, also when moving between different work places, in [19] the authors describe tracking systems suitable for hospital scenarios.

Bardram et al. in [5] use RFID technology to identify hospitalized patients and control drug assumption. In particular, beds in the wards are equipped with various RFID sensors which identify approaching tagged drugs in order to reduce medication errors by verifying that the right drugs are given to the right patient. In [30], Yu et al. introduce WISH, an application which is intended to facilitate the work of physicians and medical personal through RFID, mobile devices technology and multimedia streaming with the goal of optimizing the daily activities of the hospital staff.

De et al. in [12] describe a location-sensing prototype system that uses RFID technology to locate objects inside buildings, thus proving that active RFID is a viable and cost-effective candidate for indoor location sensing. An Access&Location service for pervasive grid applications which uses RFID and WiFi technologies to grant accesses and locate mobile objects in a pervasive way within a computer grid infrastructure is described in [10].

Other research works further describe successful projects concerning the effective adoption of RFID technologies in clinical environments. In [29], RFID was used to locate and track a group of patients, whereas in [18] RFID was used to track medical equipment, patient beds, devices and staff, leading the authors to claim that this technology could replace bar codes in the near future. In [13], an RFID-based application to guarantee the appropriateness of the clinical workflow of patients injected with radioactive substances in a nuclear medicine department is discussed.

Finally, RFID technology has also been used to improve the safety of highly critical environments such as operating rooms [26]. In [8], both RFID and wireless networks are used to enhance an operating room management information system; specifically, RFID systems are used to support clinical staff in choosing appropriate blood bags and medications in shorter times.

3 An infrastructure for smart hospitals

The proposed infrastructure consists in a Service Oriented Architecture [14] and exhibits a set of services and components as shown in Fig. 1 and described below.

3.1 Event service

Each time a change in the state of the RFID and photosensor systems is detected, an appropriate event is generated which has to be suitably processed. The Event Service implements event notifications throughout the infrastructure. It uses a software component, the BrokerComponent, which acts as an asynchronous communication broker providing a notification mechanism based on the event publish/subscribe paradigm. In particular, the BrokerComponent adopts the Topic-based subscription model, where notifications are grouped in topics and subscribers will receive all the notifications related to specific topics. Topics represent a class of events of interest to subscribers, and a service can publish a set of topics that clients can subscribe to, thus becoming an event producer.

By subscribing to a topic, a client service automatically receives notifications of all the events generated for that topic, thus becoming an event consumer. Notifications

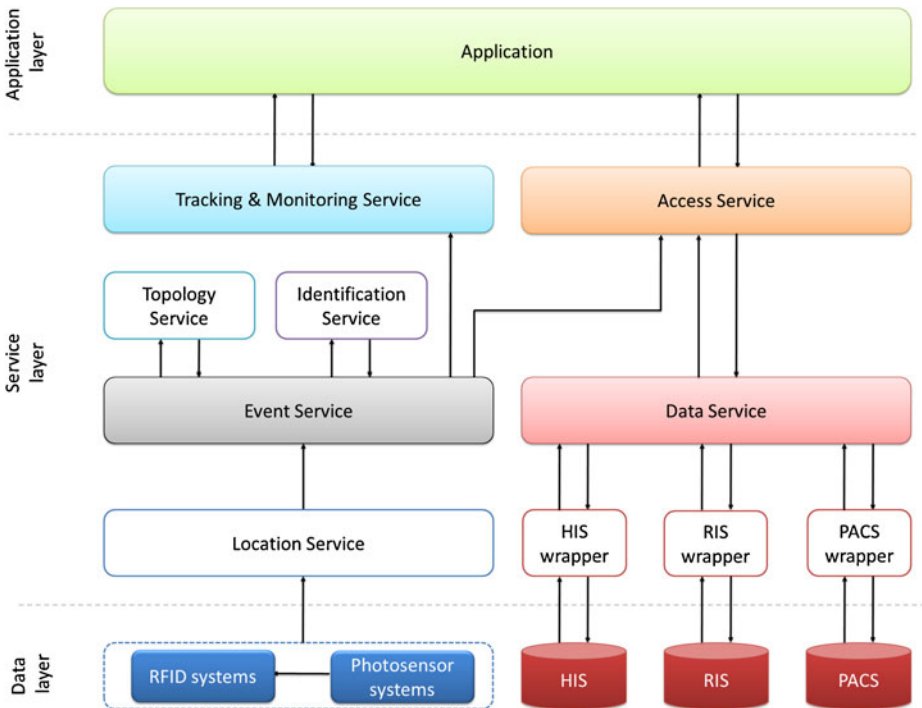


Fig. 1 The infrastructure of the system

Table 1 Topics synopsis

| TOPIC | Published by | Subscribed by |
|------------|--|---------------------------------------|
| NEW | Location Service | Topology Service |
| LOCATION | Topology Service | Identification Service |
| NEW_PERSON | Tracking&Monitoring Service | Identification Service |
| REL_PERSON | Tracking&Monitoring Service | Identification Service |
| LOCATED | Identification Service | Tracking&Monitoring Access Service |
| EXCEPTION | Location Service Topology Service Identification Service | Tracking&Monitoring Service |

are delivered from the producer to the consumer through the broker, which allows a communication between publishers and subscribers, at the same time keeping them decoupled, i.e., the producer registers with the broker and publishes its topics there, the consumer subscribes through the broker and receives, again through the broker, a notification when it is produced. Finally, such services can behave both as a producer and as a consumer.

The BrokerComponent exposes functions to register, subscribe and notify events related to the topics handled by the services of the infrastructure, as reported in Table 1.

The semantics associated with each topic reflects the functionalities yielded by the service which publishes it, and will be addressed in the respective section. The flow of the events related to these topics is illustrated in Fig. 2.

3.2 RFID/photosensor systems

An RFID system is composed of a reader connected to one or more antennas and covers a specific RFID sensed area. A number of RFID systems are installed in

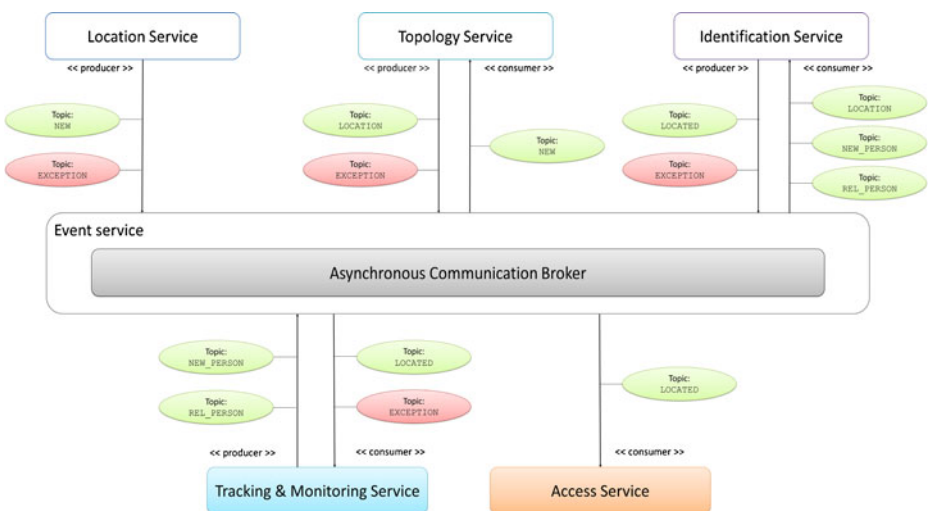


Fig. 2 Flow of topic notifications

the physical locations constituting the environment, and each system is univocally identified by its ReaderID. A physical location with an RFID system is said to be an RFID location. An RFID location is accessible through a watched gate which relies on an oriented couple of photoelectric sensors to reveal accesses. The time sequence of events caught by photosensors is used to infer the orientation and calculate the velocity of moving objects crossing the gate, in order to detect way-in and way-out movements. The calculation of the velocity is useful to reveal some anomalies such as, for example, an object obstructing the gate or inconsistencies like those arising when two or more people are so close to each other while crossing the gate that one or more of them might be hidden.

Moreover, the photosensors are also utilized to trigger on/off RFID antennas, in this way reducing RF exposure and power consumption.

3.3 Location Service

The Location Service handles low-level information coming from the RFID/Photosensor systems, such as RFID tagged objects detected by a specific RFID reader or events signaled by photosensors. Whenever a new condition occurs, the service generates an event related to the topic NEW which is then dispatched to the Event Service.

The Location Service uses a software component, the LocatingComponent, which is in charge of locating RFID tagged objects. The LocatingComponent uses the photosensors to assess velocity and orientation of moving objects, in order to infer whether a person is entering or leaving the room, and collects data coming from the readers, in order to maintain a list of the tags detected in the RFID location. There are as many LocatingComponents as RFID locations, each component being associated with a specific reader. Whenever a person passes through a watched gate, that person's movement is detected by the couple of photoelectric sensors, the RFID antennas are switched on and the reader senses the tags.

This low-level event is caught by the LocatingComponent which recognizes the movement direction, compares the tags just read with the list of tags previously detected in that location and updates the list. In this way, the component is able to infer that a moving object, with or without an RFID tag, is entering or leaving the gate, and generates the event NEW (RFIDLocation, TagID, EventType) with this information. For instance, if in the location identified by the RFIDLocation=01 the TagID=1AFF7 is detected leaving the location, the LocatingComponent generates the event NEW (01, 1AFF7, 'OUT'), while NEW (15, NULL, 'EXC_NT_IN') would mean that an object with no tag has entered the location RFIDLocation=15, thus signaling an exception. Possible values assigned by the Location Service to EventTypes are listed in Table 2.

3.4 Topology Service

The Topology Service is a service that relies on a logical model to map physical localizations, as detected by one or more positioning systems, to semantic locations. Specifically, a physical localization defines the position of a mobile object and is characterized by different granularities and scopes, depending on the particular positioning system used. In contrast, a semantic location defines the meaning of a

Table 2 Events recognized by the Location Service

| NEW EventType | Meaning |
|---------------|---|
| IN | A tagged object entering the location |
| OUT | A tagged object leaving the location |
| EXC_NT_IN | An object with no tag entering the location |
| EXC_NT_OUT | An object with no tag leaving the location |
| EXC_GEN | A generic exception has occurred |

location in a specific context and may include many physical locations. For instance, physical localizations can be GPS coordinates or an RFID location (see Section 4), whereas semantic locations can be a building, an office inside a building, or just an area in front of a wall-monitor.

The association between physical localization and semantic location is essential to the execution of the higher level services of the infrastructure. In fact, such services are location-aware, which means that they require high-level location information, while positioning systems are only able to collect raw location information. The Topology Service has been devised precisely in order to fill such a semantic gap.

The logical model adopted in designing the service defines physical locations in terms of proximity to well-known spots, rather than using an absolute reference frame. This technique is applicable to indoor environments equipped with sensors, which define physical locations in terms of their coverage areas, denoted as sensed areas. In addition to physical locations, a set of semantic locations can be defined and linked to the corresponding physical locations by decomposing the environment into atomic locations and then specifying both physical and semantic locations in terms of such atomic units.

Let us describe this approach by referring to the simple hospital layout we will adopt as an experimental scenario (see Section 4). Figure 3a shows the target topology. This specific context suggests a natural partition of the environment in terms of

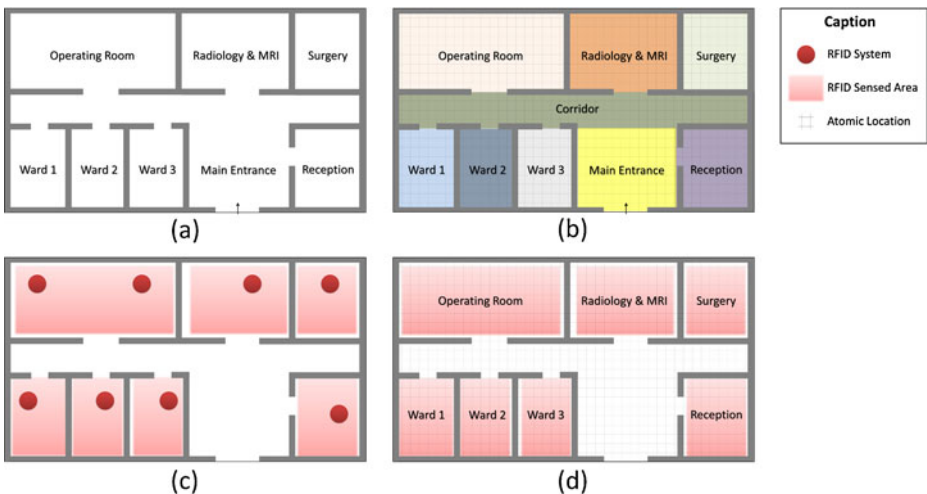


Fig. 3 Association between physical localization and semantic location

semantic locations such as Operating Room, Surgery, Ward 1, etc. Then, in order to formalize this semantics, a grid of atomic locations is put into correspondence with the layout, establishing in this way a local reference frame which leads to the *Atomic Location Perspective* shown in Fig. 3b. On this basis, a semantic location can be formally expressed as a set of atomic locations. It is worth noting that the size of the atomic location is a designer-dependent factor and that the semantic designer does not have to follow a strict one-to-one mapping of the layout. Different granularities can be defined and used, the only rule to follow being to map each atomic location to a semantic one.

The sensed areas covered by the positioning system, which in our implementation is based on the RFID technology, are RFID sensed areas, and correspond to RFID locations as previously defined, leading to the *RFID Location Perspective* shown in Fig. 3c. Finally, raw information, i.e. sensed areas, is used to attain high level information, i.e. semantic locations. This mapping is realized through the transformations *RFID Location Perspective* \rightarrow *Atomic Location Perspective* \rightarrow *Semantic Location Perspective*, as depicted in Fig. 3d.

This approach makes semantic locations independent of the positioning system used. Therefore, any change in the positioning system technology would not require a redefinition of the semantic locations. Moreover, if existing semantic locations are modified or new ones are added, the positioning system does not need to be relocated or recalibrated.

The Topology Service subscribes to the topic NEW and publishes the topics LOCATION and EXCEPTION. The service uses a software component, Semantic-Locator, which holds information about the target topology and the *RFID Location Perspective*. When an event NEW (RFIDLocation, TagID, EventType) is notified, the SemanticLocator proceeds with the transformation RFID location \rightarrow semantic location and generates an event the topic of which depends on the value of the EventType. If this latter denotes an exception (see Table 2), the event EXCEPTION (Where, TagID, EventType) is generated; otherwise the Semantic-Locator raises the event LOCATION (Where, TagID, EventType).

For instance, if the event NEW (01, 1AFF7, 'OUT') is notified and the Semantic-Locator finds that the RFIDLocation=01 corresponds to the Operating Room, the event LOCATION ('OPERATING_ROOM', 1AFF7, 'OUT') is generated, meaning that an object with TagID=1AFF7 is leaving the operating room, while NEW (15, NULL, 'EXC_NT_IN') would raise the event EXCEPTION ('WARD_6', NULL, 'EXC_NT_IN'), meaning that someone with no tag has entered Ward #6 (assuming that RFIDLocation=15 corresponds to Ward #6). Other kinds of exceptions can be raised, for example if for some reason the SemanticLocator is not able to find a location in the target topology.

3.5 Identification Service

The Identification Service provides features to identify patients and medical staff during their movements in the facility. It subscribes to the topics NEW_PERSON, REL_PERSON, LOCATION and publishes the topics LOCATED and EXCEPTION, as described below.

Whenever an event related to the topic LOCATION is notified, meaning that a tagged person has been detected at a specific location, the Identification Service

is in charge of recognizing that person. This functionality is implemented through a software component, `IdentifyPerson`, which relies on a list of recognized tagged people to process the topic. When events like `NEW_PERSON(Person, TagID)` or `REL_PERSON(Person, TagID)` are caught, the component alters the list accordingly. When an event `LOCATION(Where, TagID, EventType)` is notified, `IdentifyPerson` uses `TagID` to query the list of recognized tagged people, and upon a successful match the event `LOCATED(Where, Who, EventType)` is generated. Should the query fail to find anybody in the list, `EventType` is set accordingly and the event `EXCEPTION(Where, TagID, EventType)` is raised. For instance, if the event `LOCATION('OPERATING_ROOM', 1AFF7, 'IN')` is notified and `IdentifyPerson` finds that `TagID=1AFF7` corresponds to Dr. House, the event `LOCATED('OPERATING_ROOM', 'Dr. House', 'IN')` is generated, obviously meaning that Dr. House is entering the operating room. In contrast, if that `TagID` does not match with anybody, the event `EXCEPTION('OPERATING_ROOM', TagID, 'EXC_UN_IN')` is raised, meaning that an unknown tagged person is coming into the operation room.

3.6 Tracking&Monitoring Service

The Tracking&Monitoring Service is set at the highest level in the service layer along with the Access Service. It is in charge of (i) registering in the infrastructure patients admitted in the hospital and releasing them when discharged; (ii) keeping track of the movements of patients and medical staff and (iii) monitoring accesses to specific locations. Further tasks of this service are to detect inconsistencies and anomalies that may arise during the workflow activities in the infrastructure and to handle exceptions. The service directly communicates with the application (see Fig. 1) in order to ensure a suitable interaction with the medical staff.

The Tracking&Monitoring Service subscribes to the topics `LOCATED` and `EXCEPTION` and publishes the topics `NEW_PERSON` and `REL_PERSON`, as described below.

3.6.1 Registering and releasing people

The Tracking&Monitoring Service uses a software component, the `RegRelComponent`, to create and maintain a local database containing information about people recognized by the infrastructure along with their attributes. Routinely, when a new patient is checked in at a traditional hospital, the medical staff creates a new record in a standard medical database, storing the patient's personal and clinical historical data, and a unique `PatientID (PID)` is assigned to the patient.

`RegRelComponent` has features to initialize a wearable RFID tag—which has its own `TagID`—with the new patient's `PID` so that the patient is univocally associated with that tag. `RegRelComponent` then updates the database with this information, and the new patient is registered. The tag is eventually given to the patient to be worn throughout her/his stay in the facility, thus enabling the infrastructure to monitor her/his movements. Upon completion of the registration process, the event `NEW_PERSON(Patient, TagID)` is generated which will be processed by the Identification Service.

Likewise, when a registered patient is discharged from the hospital, she/he has to be dismissed from the infrastructure as well. With this aim, `RegRelComponent`

releases the patient by updating the related record in the database and issuing a “kill” command—a peculiar feature of the EPC protocol (see Section 4)—to the patient’s tag so that it can never be used again. Upon completion of the releasing process, the event `REL_PERSON(Patient, TagID)` is generated which will again be processed by the Identification Service.

The medical staff are also registered through a similar procedure, each member having her/his own wearable RFID tag.

3.6.2 Tracking people

The Tracking&Monitoring Service uses the TraceComponent to keep constantly updated a log of the movements of registered people within the health facility. Essentially, an entry of this log consists in the triplet (*who, where, what*) marked by a timestamp, so that whenever an event `LOCATED(Where, Who, EventType)` is notified, TraceComponent appends the proper triplet to the log. In this way it is possible to know at every time where someone stays, whether the person is a patient or a medical staff member, and to keep track of her/his movements.

For example, requests may be issued at any given moment by the personnel to know where Dr. House is or whether a certain patient has visited the Surgery. Moreover, Trace Component has also the capability of resolving collisions, which might occasionally occur if for some reason a tag is detected simultaneously by more than one reader (this may happen in the boundary regions of adjacent RFID Locations, but always after a tag has been first detected in the true location). In fact, on processing a `LOCATED` topic, TraceComponent checks the consistency of the event by analyzing the timestamp sequence related to that specific event: two or more `LOCATED` events referring to the same person notified within a short time window (e.g. 1 s), thus indicating a collision, are masked and not further processed.

3.6.3 Monitoring environments

In order to carry out a constant monitoring activity of the physical environments covered by the infrastructure, the Tracking&Monitoring Service uses the MonitorComponent. This component maintains for each semantic location a list of the people currently staying in that place; when a `LOCATED` topic is notified, the proper list is updated accordingly. In this way it is possible to know at every time who the people present in a specific room are, e. g. a function `Get_People(Surgery)` would return a list of patients and/or doctors currently staying in the Surgery by accessing the list maintained for the Surgery.

In addition, MonitorComponent implements mechanisms to enforce the correctness of the workflow of patients and/or medical staff. Generally, in health-care facilities some policies are adopted to regulate the access to critical environments; for example, only authorized personnel may be allowed in Radiology, or patients may enter the Surgery only in the presence of medical staff, etc. The Tracking&Monitoring Service exhibits a reactive and context-dependent behavior as automatic actions can be triggered in response to patients and medical staff movements. This behavior can be described by means of a set of rules conforming to the Event-Condition-Action (ECA) architectural pattern. A typical ECA rule has the form `when<event>if<conditions>then<action>`, where `<conditions>` specifies the circumstances that must be verified for the `<action>`

to be carried out whenever $\langle \text{event} \rangle$ occurs. In our case $\langle \text{conditions} \rangle$ can be any logical combination of patients/doctors staying in one specific room while $\langle \text{action} \rangle$ may be the notification of the application of a specific alert through the GUI. Each semantic location in the health facility has its own set of ECA rules specifically devised, reflecting the precise conditions to be met for that location. In fact, on processing a *LOCATED* topic, *MonitorComponent* implements the right set of rules and acts accordingly. For instance, if a patient enters the Surgery and no medical staff member is present, the component raises an alert. This behavior can be formalized by the following ECA rule associated with the Surgery:

```

WHEN (LOCATED('SURGERY', A_Patient, 'IN'))
IF ( $\exists$  doctor  $\in$  Get_people(Surgery)) THEN
  <no action>
ELSE
  <warn that A_Patient is in the Surgery but no staff
    are present>

```

3.6.4 Handling exceptions

A software component of the Tracking&Monitoring Service, *ExceptionHandler*, executes the task of gathering and processing all the events related to the topic *EXCEPTION*. When such a topic is notified, the component recognizes the type of exception and formats the appropriate alert message which is then forwarded through the GUI of the application to the user, i.e., the medical staff. A few examples of possible exception types are reported in Table 3.

3.7 Access Service

The Access Service is set at the highest level in the service layer along with the Tracking&Monitoring Service and, like this latter, directly communicates with the application. This service provides the medical staff with ubiquitous access to medical databases, on the basis of the information supplied by the services in the lower service layer.

The Access service also exhibits a reactive and context-dependent behavior dictated by a set of ECA rules associated with semantic locations. In this case, $\langle \text{action} \rangle$ consists in a series of operations aimed at formatting and generating a suitable query to the clinical databases in order to retrieve the data of a single patient and show them in the proper place with the proper format. Depending

Table 3 Examples of *EXCEPTION* types

| <i>EXCEPTION</i> type | Meaning | Notified by |
|-----------------------|---------------------------------------|------------------------|
| EXC_NT_IN | A person with no tag found entering | Topology Service |
| EXC_NT_OUT | A person with no tag found leaving | Topology Service |
| EXC_NO_LOC | Unable to map RFID/semantic locations | Topology Service |
| EXC_UN_IN | Unknown tagged person found entering | Identification Service |
| EXC_UN_OUT | Unknown tagged person found leaving | Identification Service |
| EXC_GEN | A generic exception has occurred | Any |
| ... | ... | ... |

on the context, i.e. which patient and which environment, the appropriate kind of medical information is retrieved and shown according to the resources available in the environment. In fact, the Access Service does not directly execute the query, but it relies on the Data Service (see below) to gain access to the requested data, thus being independent of the particular legacy clinical DBMS. The Access Service subscribes to the topic *LOCATED* to be notified of the movements of patients.

Let us proceed with an example of a typical feature of this service. If a patient is called into the Surgery, her/his medical record has to be readily available to the medical staff. In this situation, when the patient goes into the Surgery, the service is notified of the event and proactively acts to retrieve the patient's data. Then, since the Surgery is equipped with a wall monitor, the data are suitably formatted and visualized on that monitor.

This behavior can be formalized by the following ECA rule associated with the Surgery:

```

WHEN (LOCATED('SURGERY', A_Patient, 'IN'))
IF (∃ doctor ∈ Get_people(Surgery)) THEN
  <query the Data Service for A_patient's data>
  <get the data>
  <get Surgery's resource information>
  <format patient's data>
  <display data on the wall monitor>
ELSE
  <warn that A_Patient is in the Surgery but no staff are
    present>

```

As a further example, let us consider a Ward hosting some patients, say Ward no. 7. When a doctor comes into this Ward for a routine visit, queries to access the patients' clinical summaries are proactively generated and the results sent to the doctor's mobile device so that he/she can select a patient summary. The ECA rule in this case assumes the form:

```

WHEN (LOCATED('Ward#7', A_doctor, 'IN'))
IF (∃ patient ∈ Get_people(Ward#7)) THEN
  <get A_Doctor's mobile resource information>
  <∀ patient ∈ Get_people(Ward#7) retrieve the clinical
    summary>
  <launch a dialog box on on A_Doctor's mobile device>

```

3.8 Data Service

The Data Service provides features to access the clinical databases, namely HIS, RIS and PACS. HIS (Hospital Information System) is the information system designed to cover all the information transactions occurring in the clinical practice, from administrative and financial issues to medical data handling. RIS (Radiological Information System) is the information system used by radiology departments to store, manipulate and distribute radiological data and images. PACS (Picture Archiving and Communication System) is the information system in charge of storing, retrieving, distributing and presenting medical images.

HIS and RIS are typically compliant with the HL7 (Health Level 7) standard, which provides message-based protocols for the exchange, management and integration of clinical and administrative electronic health data. PACS is typically compliant with the DICOM (Digital Imaging and COmmunications in Medicine) standard, which defines a file format for medical images and provides a protocol for their exchange based on a set of commands.

The Data Service uses three WrapperComponents which implement handlers to communicate with HIS, RIS and PACS through their respective standards. Specifically, such components allow to (i) formalize messages or commands in the standards HL7 ver. 3.0 and DICOM ver. 3.0, and (ii) send and receive messages, commands and data to and from HIS, RIS and PACS.

4 Experimental implementation and testing

An experimental scenario, depicted in Fig. 4, has been devised as a testbed of the proposed infrastructure. It consists in a simple hospital layout composed of a number of physical environments, as described below. Each environment is accessible through a watched gate (except the Reception) and is equipped with an RFID reader and antennas; a desktop PC to access system functionalities is also installed.

| | |
|-----------------|--|
| Reception | Here the patient is accepted for hospitalization. A desk operator (medical staff member) registers the patient (i.e., a new record is created and all required data are input), a fresh wearable RFID tag is associated with the patient’s PID and given to the patient. |
| Wards | Rooms where patients reside. |
| Operating room | This is the room where medical staff provide surgical services to patients. |
| Radiology & MRI | This is the setting where radiological tests are carried out. This room is equipped with a wall monitor to display the medical images produced. |
| Surgery | In this room doctors perform specific examinations. |

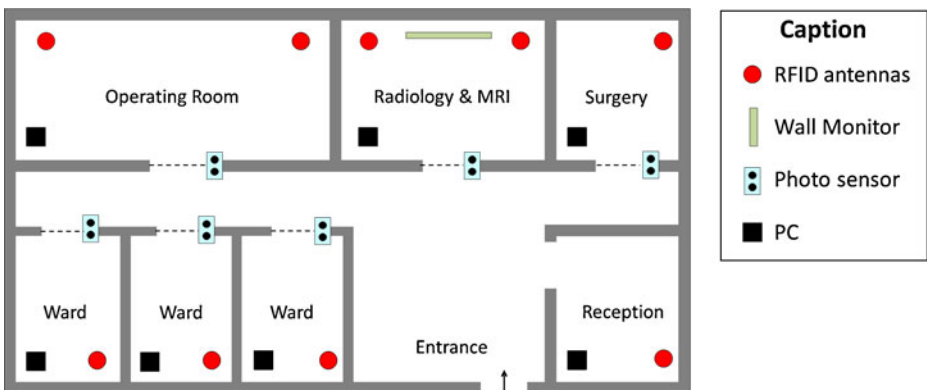


Fig. 4 Hospital layout

An experimental prototype for this scenario has been realized with the aim of giving a proof of concept of the principles underlying the design of the infrastructure. This prototype implements the following functionalities that directly come from the service oriented architecture previously described:

| | |
|-------------------------------|--|
| Initialize & Release Patient | These functionalities enable the medical staff at the Reception to assign an RFID wearable tag to a new patient and void the tag upon the patient discharge. The Initialize functionality includes accessing the HIS to retrieve the patient's data. |
| Locate Person | This functionality allows the medical staff to locate medical personnel and patients. |
| Get people | This functionality is intended to obtain a list of people located at a specific location. |
| Visualize data | This functionality provides medical staff with ubiquitous access to the patient's data. This functionality includes accessing HIS, RIS and PACS in order to retrieve clinical reports and images of the patient. |
| Track & Monitor | This functionality enables medical staff to obtain reports on people's movements and room accesses. |
| Generate Location Information | This functionality yields location information gathered through the detection of people's movements. It includes the identification of medical personnel and patients and is capable of notifying possible unauthorized accesses. |

These functionalities are modeled in the use case diagram shown in Fig. 5.

4.1 Prototype implementation

The prototype mainly consists of a set of front-end clients, a back-end centralized server and a positioning system which integrates RFID and photosensor technologies. The front-end clients run on a number of PCs with a LAN connection to the back-end server and expose a Graphical User Interface consisting in a window application program developed in Java language.

The back-end server consists in a workstation which runs the software components described in Section 3 and hosts local databases to store information about patients and to simulate RIS and PACS access; the LAN connection also allows communications between front-end clients and the RFID/photoelectric sensor network. The software components have been developed as standard OSGi modules. They are fully portable Java entities, with the exception of the Location Services, which also integrates, as native code, Microsoft Visual C++ 6.0 control drivers provided by the RFID technology and photoelectric sensor manufacturers.

The positioning system used by the prototype relies on RFID readers belonging to FEIG's Passive RFID Long Range Reader series, model ISC.LRU2000-A. These readers are able to interact with EPC Class 1 Gen 2 transponders to operate frequencies in the UHF range 865–870 MHz and can drive up to four UHF antennas;

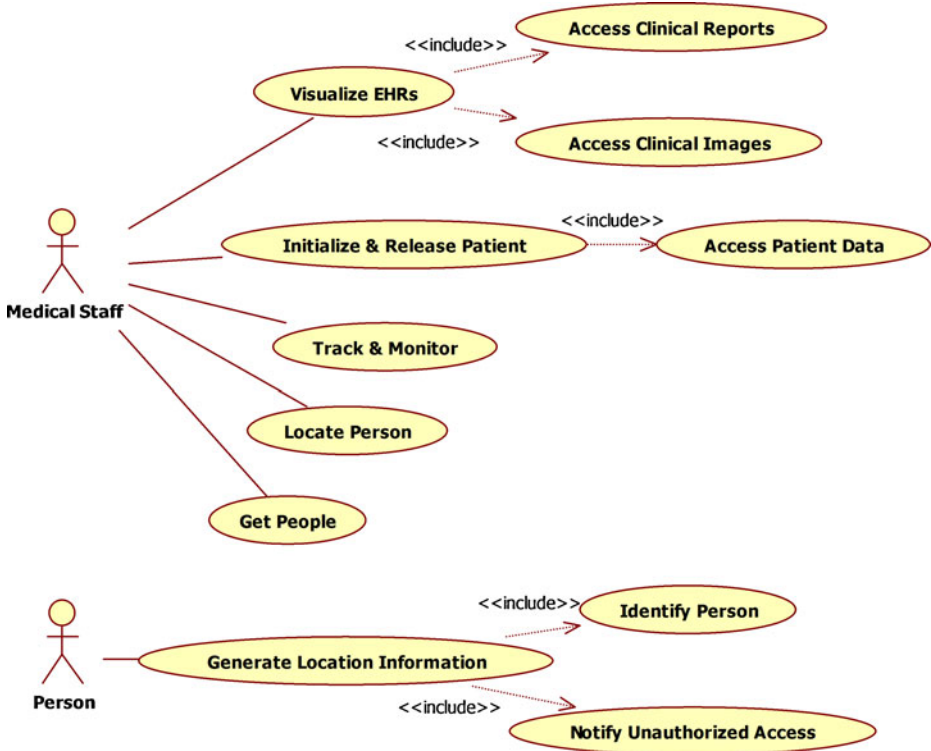


Fig. 5 System use cases

with at least two antennas a reading range of up to 10 meters (approx. 33 feet) can be covered. In addition, FEIG's UHF antennas model ID ISC.ANT.U250/250-EU have been installed. These antennas are circular polarized for operating frequencies in the UHF range 865–870 MHz. Omron photoelectric retro-reflective sensors, model E3F2-R4B4-E 2M have been also embedded in the system.

The experimental prototype has been implemented at the Institute for High Performance Computing and Networking (ICAR) of CNR—the National Research Council of Italy. It has been set up in a test environment simulating the hospital layout depicted in Fig. 4. As previously mentioned, each room constitutes an RFID location and is accessible through a watched gate which permits one single access at a time. A watched gate is realized by equipping the entrance door with a pair of oriented photoelectric sensors, namely two photoelectric retro-reflective sensors mounted in a such way that their beams are parallel and lie in a plane which is parallel to the floor. Each beam is assigned a conventional order, i.e. one precedes the other in a given direction. This oriented pair is positioned on a side of the entrance with beams crossing the doorway.

The antennas in the RFID location are positioned on two opposite walls in the room. They are mounted face-to-face, tilted with an angle of approx. 45° downward.

Up to four antennas are driven by a single reader and powered with 2 W ERP, in compliance with UE RFID regulations. Such a configuration has proved to be the most effective to ensure at least one read for each RFID tag staying in the room and to reduce collisions due to simultaneous readings in contiguous settings. Moreover, the events caught by photosensors are also used to trigger on and off the power to the antennas, in this way avoiding unnecessary RF exposure and reducing power consumption.

4.2 Assessment of testing activities

Tests have been performed on the experimental prototype to assess its behavior and, consequently, the feasibility of the infrastructure in a real scenario application. We focused mainly on the quality of service in term of dependability, i.e., “the trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers” [20]. Therefore the dependability of the prototype was evaluated by

- measuring response times, i.e. the time interval elapsed between changes in the state of the target scenario and the completion of the required actions, e.g., proper alert notifications, answers to enquiries made by clinicians or suitable queries to retrieve clinical data. It should be stressed that the figures reported below do not take into account the times required to access true HIS, RIS and PACS, which depend on the legacy clinical DBMS used in a real application.
- counting undetected/misdetected changes in the scenario, e.g., unread tagged people, trespassing, unresolved collisions, etc.

In undertaking testing activities, we adopted initially a scaled-down approach where only six patients, three members of the medical staff and an untagged visitor were simulated in the healthcare facility. Consequently, we reproduced a scenario where these people were randomly involved in the workflow of the hospital, either remaining idle or moving around the rooms and simulating various situations. In this framework, the prototype proved to be able to monitor the workflow of 100% of the people by providing typical response times $\ll 10$ s to display clinical data, with a peak response time < 1 s in the worst case, i.e., when all six patients entered the same location. Also, typical response times $\ll 1$ s were measured to generate alert messages, with a peak response time < 1 s in the worst case. No collisions were detected during this test case, nor was any undetection/mis-detection noticed. Then, dynamic tests were performed along a simulated time span of five days with an increasing number of patients and/or personnel, simulating the presence of up to twenty patients and five members of the medical staff plus five untagged visitors. This latter case reported typical response times ≤ 1.5 s to display clinical data, with a peak response time < 2 s in the worst case. Also, typical response times ≤ 1 s were measured to generate alert messages, with a peak response time < 2 s in the worst case. By a comparison with the scaled-down test case, it can be noted that no meaningful variations occurred in the readiness of the service, even for a threefold population in the facility. However, in the latter case as many as nine collisions were detected but only eight were properly masked. Additionally, three people were not correctly localized at the first attempt. All these errors were due to RF glitches. It is

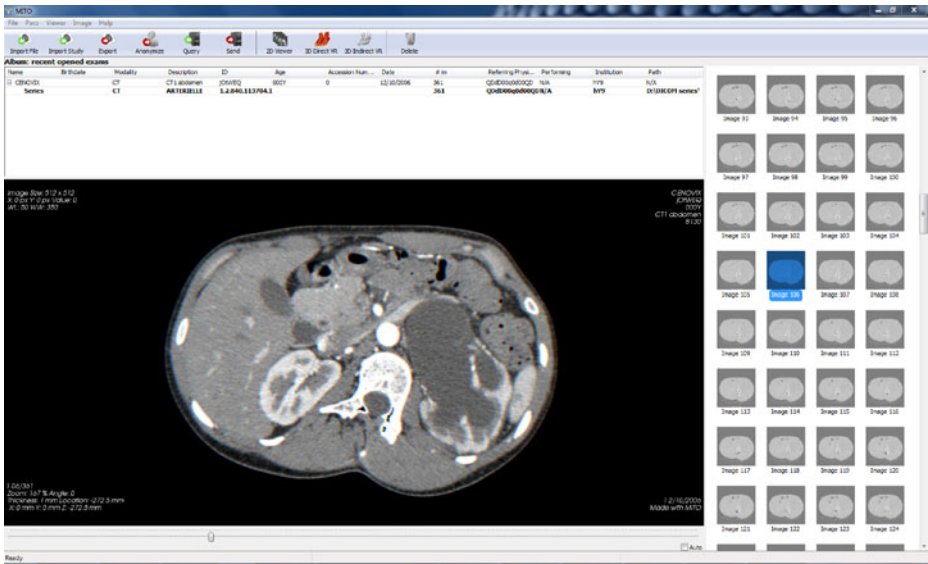


Fig. 6 A screenshot of typical radiological images output by the application

a fact that communication through RF is prone to interferences and random noises, so that a real application would require a convenient number of RFID readers and antennas to ensure a full coverage of the sensed areas.

In summary, we can state that the assessment carried out on the experimental prototype gave a proof of the feasibility of the proposed infrastructure, suggesting that a scaled-up implementation could be effectively deployed in a real Smart Hospital application.

An example of the clinical data retrieved during one of the above mentioned tests, a screenshot of typical radiological images output by the application implemented in the experimental prototype, is displayed in Fig. 6. This figure shows X-Ray images pertaining to a patient detected in the Radiology & MRI room as they are displayed on the wall monitor there installed. The patient has been identified, the proper query to the PACS has been performed and the retrieved images have been visualized, all this being carried out automatically without any human intervention.

5 Conclusions

As Ubiquitous Computing technologies constitute a powerful tool to address the complex management of healthcare institutions, the gap between the technological reality and the application requirements may be considered the factor which most seriously limits a widespread deployment of pervasive-based solutions in clinical scenarios. In fact, there is still more work that has to be carried out to effectively enforce viable, productive applications in daily clinical practice, where heterogeneous devices, resources and infrastructures are involved.

As an effort in this direction, this paper presents an intelligent software/hardware infrastructure which is intended to shorten the patent gap existing between health-care domain constraints and pervasive enabling technologies. The rationale of this infrastructure's design derives from the need to simplify a context-aware access to the underlying environmental resources of a typical healthcare facility while keeping flexibility and extendibility at a reasonable level. For this reason, our research has been primarily focused on the design of a service-based framework that is capable of easily interoperating with the legacy software and databases usually present in hospitals, offering at the same time features to raise the quality of healthcare services anytime to anyone regardless of location and other constraints. Specifically, we have introduced a three-layered, event-based and service-oriented architecture which, through the integration of RFID and photosensor technologies, yields functionalities to identify, locate and track clinicians and patients equipped with mobile devices and wearable RFID tags. Additionally, ubiquitous and transparent access to standard medical databases is granted.

A very simple but meaningful hospital layout has been devised as a testbed of the proposed infrastructure, for which an experimental prototype has been realized in Naples (Italy) at the Institute for High Performance Computing and Networking (ICAR) of the National Research Council of Italy (CNR). Using information on people's movements supplied by passive RFID and photosensor devices, the prototype runs a set of basic services implemented as standard OSGi modules. Moreover, handlers to communicate with legacy HIS, RIS and PACS through their respective standards have been integrated in the system's software architecture, achieving the final aim of allowing a pervasive and transparent access to clinical data.

An assessment of the experimental results gathered during suitable testing activities carried out on the prototype has provided a proof of the feasibility of the proposed infrastructure, suggesting that a scaled-up implementation could be effectively deployed in a real Smart Hospital application.

References

1. Al Nahas H, Deogun JS (2007) Radio frequency identification applications in smart hospitals. In: CBMS '07: twentieth IEEE international symposium on computer-based medical systems. IEEE Computer Society, Washington, DC, pp 337–342
2. Bacheldor B (2007) HCA North Florida expands its RTLS to track patients. <http://www.rfidjournal.com/article/articleview/3615/1/1/>
3. Bacheldor B (2007) Tags track surgical patients at Birmingham Heartlands Hospital. <http://www.rfidjournal.com/article/articleview/3222/1/1/>
4. Bång M, Larsson A, Eriksson H (2003) NOSTOS: a paper-based ubiquitous computing health-care environment to support data capture and collaboration. In: Proceedings of the AMIA annual symposium. Washington, DC, pp 46–50
5. Bardram JÈ (2004) Applications of context-aware computing in hospital work: examples and design principles. In: SAC '04: proceedings of the 2004 ACM symposium on applied computing. ACM, New York, pp 1574–1579
6. Borriello G, Stanford V, Narayanaswami C, Menning W (2007) Guest editors' introduction: pervasive computing in healthcare. *IEEE Pervasive Computing* 6:17–19
7. Casimiro A, Kaiser J, Verissimo P (2004) An architectural framework and a middleware for cooperating smart components. In: CF '04: proceedings of the 1st conference on computing frontiers. ACM, New York, pp 28–39

8. Chen PJ, Chen YF, Chai SK, Huang YF (2009) Implementation of an RFID-based management system for operation room. IEEE Computer Society, Washington, DC, pp 2933–2938
9. Coronato A, De Pietro G, Gallo L (2008) An agent based platform for task distribution in virtual environments. *J Systems Archit* 14(9):877–882
10. Coronato A, Della Vecchia G, De Pietro G (2006) An RFID-based access and location service for pervasive grids, vol 409. Springer, Berlin, pp 601–608
11. Coronato A, Esposito M (2008) Towards an implementation of smart hospital: a localization system for mobile users and devices. In: PERCOM '08: proceedings of the 2008 sixth annual IEEE international conference on pervasive computing and communications. IEEE Computer Society, Washington, DC, pp 715–719
12. De P, Basu K, Das SK (2004) An ubiquitous architectural framework and protocol for object tracking using RFID tags. In: MobiQuitous '04: proceedings of the first annual international conference on mobile and ubiquitous systems. IEEE Computer Society, Los Alamitos, pp 174–182
13. Della Vecchia G, Esposito M (2009) A pervasive system for nuclear medicine department. *Wirel Pers Commun* (2009). doi:10.1007/s11277-009-9789-x
14. Erl T (2009) SOA design patterns. Prentice Hall, Upper Saddle River
15. Favela J, Tentori M, Castro LA, Gonzalez VM, Moran EB, Martínez-García AI (2007) Activity recognition for context-aware hospital applications: issues and opportunities for the deployment of pervasive networks. *Mobile Networks and Applications* 12(2–3):155–171
16. Gallo L, De Pietro G, Coronato A, Marra I (2008) Toward a natural interface to virtual medical imaging environments. In: AVI '08: proceedings of the working conference on advanced visual interfaces. ACM, New York pp. 429–432
17. Gallo L, Minutolo A, De Pietro G (2010) A user interface for VR-ready 3D medical imaging by off-the-shelf input devices. *Comput Biol Med* 40(3):350–358
18. Halamka J (2003) Early experiences with positive patient identification. *J Healthc Inf Manag* 20(1):25–27
19. Holzinger A, Schwabinger K, Weitlaner M (2005) Ubiquitous computing for hospital applications: RFID-applications to enable research in real-life environments. In: COMPSAC '05: proceedings of the 29th annual international computer software and applications conference. IEEE Computer Society, Washington, DC, pp 19–20
20. IFIP WG 10.4 on dependable computing and fault tolerance. <http://www.dependability.org/wg10.4/>
21. Lahtela A, Hassinen M (2009) Requirements for radio frequency identification in healthcare. In: MIE '09: proceedings of the XXIInd international congress of medical informatics in a united and healthy Europe. IOS Press, Amsterdam, pp 720–724
22. Pietzuch PR, Bacon J (2002) Hermes: a distributed event-based middleware architecture. In: ICDCSW '02: proceedings of the 22nd international conference on distributed computing systems. IEEE Computer Society, Washington, DC, pp 611–618
23. Satoh I (2004) Linking physical worlds to logical worlds with mobile agents. In: MDM '04: proceedings of the IEEE international conference on mobile data management. IEEE Computer Society, Los Alamitos, p 332
24. Satoh I (2005) A location model for pervasive computing environments. In: PERCOM '05: proceedings of the third IEEE international conference on pervasive computing and communications. IEEE Computer Society, Washington, pp 215–224
25. Satyanarayanan M (2001) Pervasive computing: vision and challenges. *IEEE Pers Commun* 8:10–17
26. Schimpff SC (2007) Improving operating room and perioperative safety: background and specific recommendations. *Surgical Innovation* 14:127–135
27. Skov B, Høegh T (2006) Supporting information access in a hospital ward by a context-aware mobile electronic patient record. *Personal and Ubiquitous Computing* 10(4):205–214
28. Varshney U (2003) Pervasive healthcare. *Computer* 36(12):138–140
29. Wang SW, Chen WH, Ong CS, Liu L, Chuang YW (2006) RFID application in hospitals: a case study on a demonstration rfid project in a taiwan hospital. In: HICSS '06: proceedings of the 39th annual Hawaii international conference on system sciences. IEEE Computer Society, Washington, DC, p 184.1
30. Yu W, Ray P, Motoc T (2006) A RFID technology based wireless mobile multimedia system in healthcare. In: HEALTHCOM 2006: the 8th international conference on e-health networking, applications and services. IEEE Computer Society, Los Alamitos, pp 1–8



Gennaro Della Vecchia received his M.Sc. in Electronic Engineering from the University of Naples “Federico II” in 1984. He is a scientist researcher at the Naples branch of Institute for High-Performance Computing and Networking (ICAR) of the National Research Council of Italy (CNR) and a long time contract professor of Computer Engineering at the University of Naples “Federico II”. His research activity has been mainly focusing on parallel architectures, interconnection networks and distributed computing and, more recently, on their applications in multimedia and pervasive healthcare.



Luigi Gallo received his M.Sc. in Computer Engineering from University “Federico II” of Naples in July 2006 and the Ph.D. degree in Information Technology Engineering at the University of Naples “Parthenope” in April 2010. Currently he is a research fellow at the National Research Council of Italy (CNR)—Institute for High-Performance Computing and Networking (ICAR), where he is involved in the design of interactive systems for VR-enhanced 3D medical imaging environments. His fields of interest involve Virtual Reality and Human-Computer Interaction, specifically considering healthcare application scenarios. He is a professional member of ACM, ACM SIGGRAPH, ACM SIGCHI and SIGCHI-Italy.



Massimo Esposito was born in Naples, Italy, in 1979. He received his M.Sc. in Computer Science Engineering from the University of Naples “Federico II” in 2004. He is a scientist researcher at the Naples branch of Institute for High-Performance Computing and Networking (ICAR) of the National Research Council of Italy (CNR) and a Ph.D. student in Information Technology Engineering at the University of Naples “Parthenope”. His research interests involve pervasive computing and decision support systems applied to healthcare for supporting medical activities.



Antonio Coronato received the degree in information engineering at the University “Federico II” of Naples in the 1998. He is a researcher at the Institute for High Performance Computing and Networking (ICAR) of the National Research Council of Italy (CNR). He is grant professor of Software Engineering at the University “Federico II” of Naples and Object Oriented Programming at the Second University of Naples. His research interests are in the field of pervasive computing, middleware and component-based architectures and dependable computing.