A Framework for Context-Aware Home-Health Monitoring

Alessandra Esposito, Luciano Tarricone, Marco Zappatore, Luca Catarinucci, Riccardo Colella, and Angelo DiBari

University of Salento, Lecce, Italy
{alessandra.esposito,luciano.tarricone,
luca.catarinucci}@unile.it

Abstract. This paper presents a proposal for a context-aware framework. The framework is organized according to a general purpose architecture, centred around an ontological context representation. The ontology provides the vocabulary upon which software agents interoperate and perform rule-based reasoning, in order to determine the system response to context changes. The system components and their coordinated operations are described by providing a simple example of concrete application in a home-care scenario.

Keywords: context-aware, ontology, rule, logic, agent, ubiquitous, health.

1 Introduction

Progress in mobile devices, wireless networks and software technologies is bringing healthcare a new generation of systems, which are known under the name of pervasive and context-aware systems. The term 'pervasive' [1] refers to the seamless integration of devices into the user's everyday life. Appliances vanish into the background to make the user and his tasks the central focus rather than computing devices and technical issues. Context-aware systems adapt their operations to the current context without explicit user intervention. These kinds of applications pose several technological challenges. First of all, a pervasive application is made up of heterogeneous and dispersed components which must be able to interoperate in a transparent manner. Moreover, in order to adapt to context variations, the system must elaborate raw data sensed by context sources, such as sensors, and extract high level context information from them. Both requirements lead to the identification of the core component of the system: a robust, reusable and sharable context representation. However, the definition of a shared context model provides the basis for 1) allowing the system components to cooperate and build together the system reaction to incoming events, 2) structuring sensed data in a meaningful and interpretable form. Currently, there is no standard modelling option for context [2]. Therefore, available frameworks adopt proprietary approaches. Among them, ontology-based ones [3,4,5] are more and more recognized as the most promising [6], as they provide a rich formalism for specifying contextual information and are reusable and sharable. The adoption of ontologies is encouraged by their capability of being embedded within multi-agent and rule-based systems. Indeed, designing the application in a multi-agent fashion allows one to organize it around the

F.E. Sandnes et al. (Eds.): UIC 2008, LNCS 5061, pp. 119–130, 2008.

coordinated interaction of autonomous reasoning entities, which wrap the "natural" components of the pervasive system (such as sensing devices and consumer services). Rule-based logic supports agents in implementing advanced reasoning and in deriving high-level concepts from sensed information thus opening the application to sophisticated adaptation and pro-active behaviour.

This work proposes a framework for context-aware pervasive systems built around the three above mentioned technologies: ontology representation, multi-agent paradigm and rule-based logic. The amenability of these technologies for context-aware pervasive environments has been demonstrated in a number of recent works. Among them, we recall CoBrA [5], an agent-based framework for intelligent meeting rooms centred around an OWL ontology. Gu et al. [6] propose SOCAM which is an ontology-based middleware for context-aware services in smart space domains. An ontology-based context model and context reasoning for ubiquitous health-care is presented in [4]. Health-care is the focus of Paganelli and Giuli [3] work as well, which describes a system built around OWL and rule-based inference. A further impulse to the integration of the three technologies is provided with our work, which is the result of an effort to enhance their interoperability. As a result, the ontology provides the knowledge codification needed to support both agent reasoning and communication. The effectiveness of the adopted model is shown by using a simple and concrete example in a home-care scenario.

The paper is organized as follows. Section 2 introduces the general-purpose framework and its prototypal implementation. Section 3 is centered around context modeling. First the ontology representation, then the rule-based reasoning are illustrated. Section 4 focuses on how agent-based reasoning supports context elaboration. Section 5 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 is adopted throughout the entire paper as specific example of application and practical result.

2 System Architecture

The architecture of our system (Fig. 1) follows a widely accepted abstraction [7], according to which context-aware systems are organized into three layers: context sources. Context management middleware and context consumer level

Context sources include entities providing raw context data. They are conceptually partitioned into two groups: physical and virtual sources [8]. Physical sources include all 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.

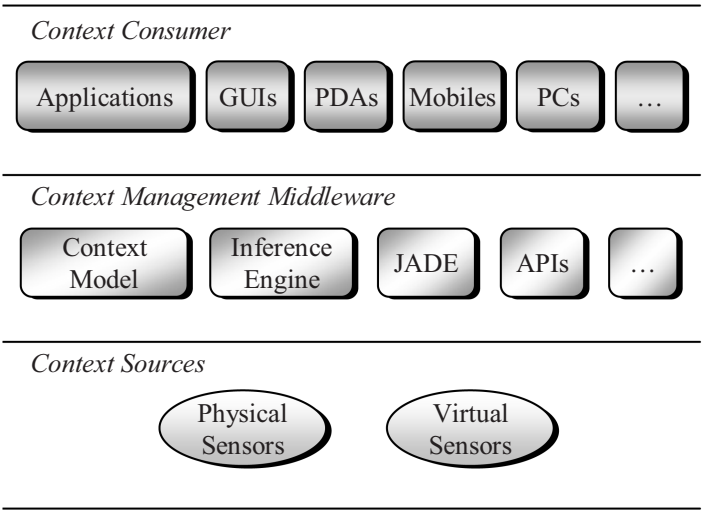

Fig. 1. The three-layer system architecture. The lowest layer includes physical and virtual sensors, i.e. devices and software entities forwarding context data. The middle level hosts middleware components, such as software development kits and APIs. The context model is the core of the system and belongs to the middle level as well. End user devices and applications are positioned at the highest level.

The prototypal implementation of the framework is drawn in Fig.2. As shown in the figure, the distributed system is made up of a team of interoperating agents, which share a common knowledge representation and reason through an inference engine. The agents are dispersed over several nodes and play different roles. As explained in Section 4, Context Provider Agents (CPA) wrap context sources and are in charge of converting raw context data into high level context information. CPAs cooperate with Context Interpreter Agents (CIA) which are responsible for managing high level context information and of identifying the set of actions to be triggered as a consequence of an emergence. Finally, Context Consumer Agents (CCA) forward the message/signal describing the alarm situation to the most suited destination. Input data are provided by a physical context source obtained by integrating sensors with an RFID device (see Section 5) and by a virtual context source containing static information.

In the following sections, the system components and their way of operating are described, with reference to a home-care scenario.

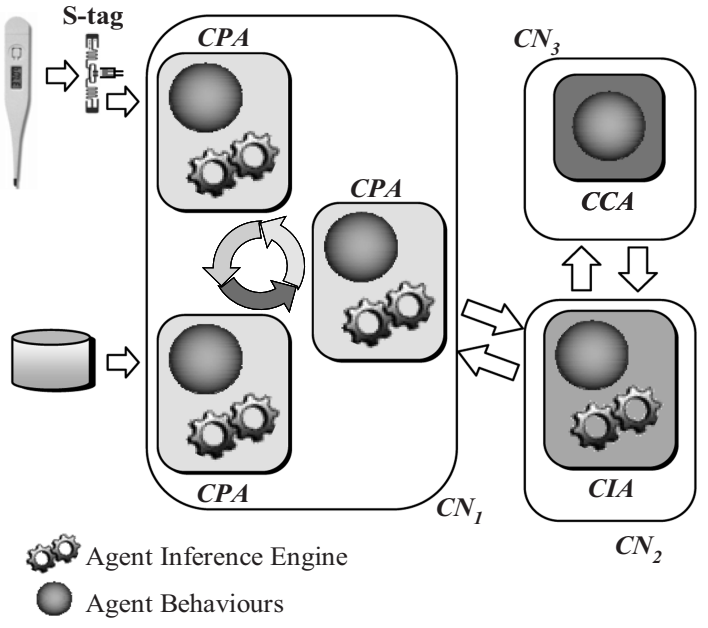

Fig. 2. System Prototype. The system was implemented on three nodes (*CN*) connected by a local area network. It follows a multi-agent paradigm. Context Provider Agents (*CPA*) filter and integrate data coming from physical and virtual sensors, thus converting raw context data into high level context. Context Interpreter Agents (*CIA*) process high level context information to identify the actions and the actors best suited for managing an emergency. Context Consumer Agents (*CCA*) forward the alarm information to the final destination.

3 Context Representation

An application is context-aware if it is able to adapt its behaviour to context by suitably reacting to context variations. In other terms, a context-aware application must exhibit a sort of "situation awareness", i.e. it must manage and interpret context and its real-time evolution. Moreover, a context-aware application is generally made up of several components, possibly deployed on dispersed nodes, which need to interact with one another. Therefore it is fundamental for such components to share a common representation of context knowledge.

These premises cast a twofold approach to context modelling.

First of all, the fundamental entities, objects and relations of a situation must be represented. This is needed in order to ground system knowledge on a common basis and provide a unified concept taxonomy to the several components interacting in the application framework. Secondly, context changes originated by meaningful data variations must originate recognizable "high-level" events, so that situation switching can be interpreted and suitably managed by the system. In other terms, the system must be enabled to deduce high-level, implicit context from the low-level, explicit context directly acquired from sensors. This operation often requires reasoning over complex combinations of different data and context information. For example, an

increase of blood pressure (low level context) exceeding the patient threshold (low level context) may produce a switching to an "alarm" situation (high level context), which, on its turn may produce a series of operations finalized at identifying and invoking the most suited available and reachable doctor.

Our system attacks the former need, i.e. machine representation of context entities, by the use of ontologies which enable to describe meaningful events, objects and relations of context and support several fundamental forms of reasoning, such as concept satisfiability, class subsumption, consistency and instance checking.

The latter need, situation switching management, is approached by implementing logical rules and by embedding context events into facts. In this way, an iterative process is activated every time rules are matched. Indeed, fired rules infer new facts, i.e. new scenario switchings, which on their turn may fire other rules. This process allows the system to convert low level context into high level context, with the final result of producing the system reaction to the context switching which originated the process. For example the occurrence of low level events sensed by physical and virtual sensors, may originate the "diastolic blood pressure of patient X=100" and "threshold of patient X=90" facts. These facts may fire a rule inferring the high level fact "alarm for patient X=true". The alarm situation, combined with facts concerning the availability of doctors, may produce the fact "doctor Y has to be called", which generates the system response to the detected patient anomaly.

As explained in the following subsections, the ontologies were implemented in the OWL language, whilst the rule-based domain knowledge was implemented with Jess on top of OWL ontologies.

3.1 Context Ontology

Several context modeling techniques [9] exist such as key value, mark-up scheme, graphical, object-oriented, and ontology-based modeling. According to [9], the ontology-based approach fits well with the requirements of ubiquitous/context-aware computing. Indeed. ontologies facilitate knowledge sharing and reuse. Knowledge sharing is enabled by providing a common knowledge model to computational entities, such as agents and services, which need to interoperate with one another. Knowledge reuse is promoted by ontology amenability to be extended to different domains and to be integrated within wider ontology-based frameworks.

Many ontology languages exist including Resource Description Framework Schema (RDFS) [10], DAML+OIL [11], and OWL [12]. OWL is a key to the Semantic Web and was proposed by the Web Ontology Working Group of W3C. Therefore, it was chosen for our prototype.

A common practise, when developing ontologies, is to adopt a *top level* (upper) shared conceptualization [13] 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.

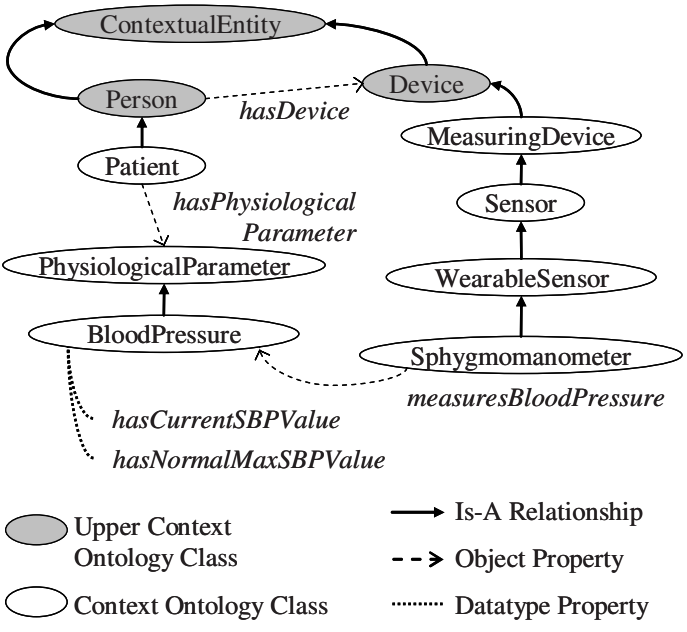

Fig. 3. Some classes from the context domain ontology referring to the "patient environment"

Therefore, we divided context ontology into two levels: the top-level context and the domain context ontology. The top-level context ontology includes concepts which refer to context-aware computing, independently from the application domain. The domain context ontology refers explicitly to the health-care domain.

The vocabulary related to context entities was defined starting from the widely accepted definition of context, provided in [14]: "Context is any information that can be used to characterize the situation of an entity." Therefore, an entity is a person, place, computational entity, or object which is considered relevant for determining the behavior of an application. Therefore, as shown in Fig.3 and Fig.4, "person", "device" and "TriggeredAction" are contextual entities which specialize into different con?epts depending on the context subdomain they belong to. For instance, a person can be a patient, a relative of the patient or a health operator. Devices can be of different types as well. For the sake of brevity, the figures show just small ontology fragments referring to the example used throughout the remaining part of the paper.

3.2 Context Rules

The proposed framework utilizes Jess [15] to structure knowledge in the form of declarative rules. Jess is a widely adopted rule engine and scripting environment written in Java. It adopts the Rete algorithm [16] to implement efficiently the rule matching.

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

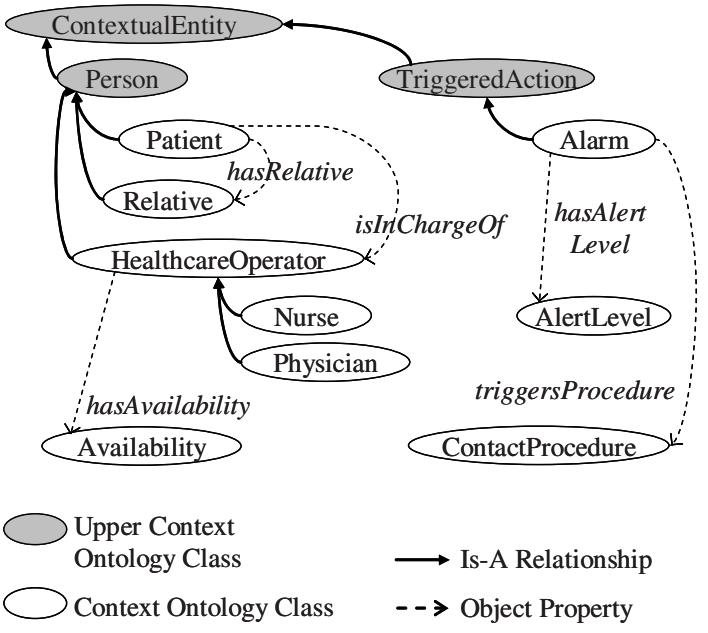

Fig. 4. Some classes from the context domain ontology referring to the "application consumer environment"

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 w*orking memory* are updated. A set of first-order rules determines if an alarm has to be triggered and which alarm level should be activated, according to measurement values and corresponding thresholds. Jess *pattern matcher* 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.

For instance, the following example shows a rule activating an alarm when the systolic blood pressure ("sbp-c" variable) exceeds the patient threshold ("spb-max" variable). When the rule is fired, the fact "status abnormal is true" ("sbp-s" variable) is inferred and the action "notify the abnormal event" is activated ("sbp-anomaly-notification" action):

```
(defrule verify-SystolicBloodPressure
   (measurement
      (hasPID ?m-pid)
      (hasMeasurementDate ?d)
      (hasMeasurementTime ?t))
   (patient  (hasPatientID ?pid)
      (hasCurrentSBPValue ?sbp-c)
      (hasNormalMaxSBPValue ?sbp-nmax)
```

```
      (SBPStatusAbnormal ?sbp-s))
   (test (> ?sbp-c ?sbp-max))
  =>
 (bind ?sbp-s true)
 (sbp-anomaly-notification)
)
```

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 a rule fired as a consequence of an abnormal status due to both systolic and diastolic blood pressure. The rule infers the fact “alarm is true” and the action “find-available-physician”

```
(defrule set-alert-level
   (patient   (hasPatientID ?pid)
      (SBPStatusAbnormal ?sbp-s)
      (DBPStatusAbnormal ?dbp-s)
      (HighAlertLevel ?hal))
  (test (eq ?sbp-s ?dbp-s true))
  =>
  (bind ?hal true)
  (find-available-physician)
)
```

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 Agent-Based Reasoning

All the agents are implemented by using the Java Agent Development Environment (JADE). JADE [17] is a software framework to develop and run agent applications in compliance with the FIPA specifications [18] 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. As shown in Fig.5, the semantics of agent messages and reasoning is built over OWL concepts and predicates, which are matched with Jess and JADE vocabulary.

Figure 6 shows the proposed multi-agent framework, which assigns three fundamental roles to agents:

Context provider agents (CPA). These agents wrap context sources to capture raw context data and instantiate the ontology representation. CPAs may encapsulate single

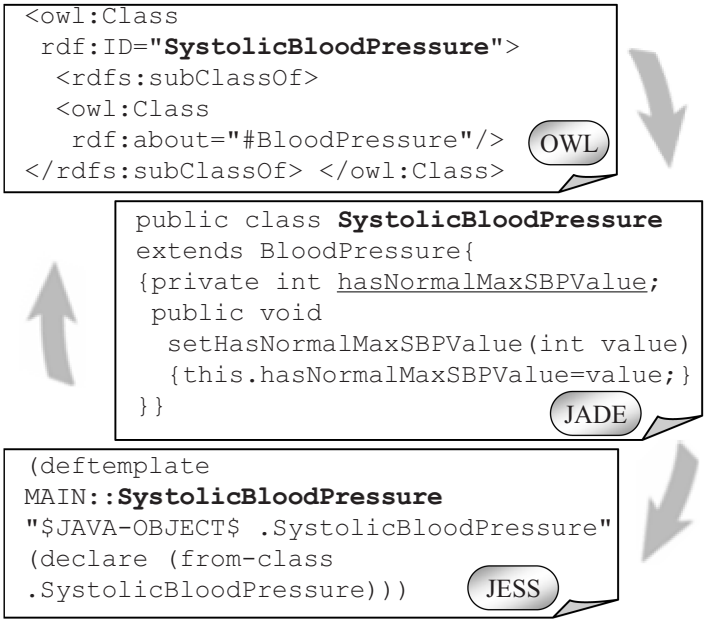

Fig. 5. The system implementation is based on the matching between OWL vocabulary with agent inner context representation (in the form of Java classes) and Jess facts codification

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, [19] 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.

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.

5 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. Nevertheless, in order to be effectively and conveniently integrated in the scheme proposed in Fig.2, the capability to measure a physical value (such as temperature, blood pressure or oxygen saturation), is only one of the several

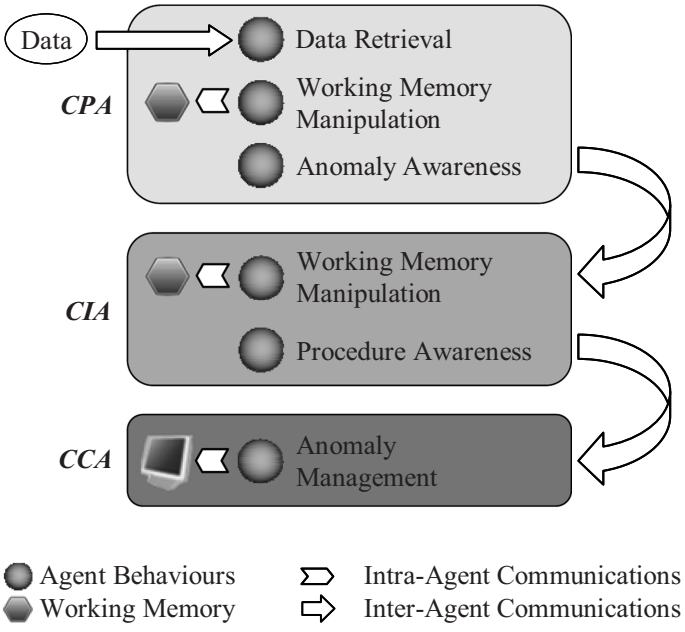

Fig. 6. The multi-agent framework. Context Provider Agents (CPA) are responsible for inferring a potential "anomaly awareness" situation from data provided by context sensors. Context Interpreter Agents (CIA) process high level knowledge provided by CPAs to acquire a sort of "procedure awareness". Context Consumer Agents (CCA) forward signals and messages to the suited destination as requested by CIAs (anomaly management). The three categories of agents embed a Jess rule engine, update the working memory of Jess and interoperate by using ACL messages.

requirements a physical source should satisfy. The measured data, for instance, should be sent to a data collector by using a wireless connection, and the choice of the most adequate one is not univocal: wi-fi, Bluetooth, GPRS, UMTS, GSM are only a few of the many possible candidates. In order to allow the indispensable capillary diffusion of physical sources, though, the ratio benefit/cost cannot be left out of consideration, thus imposing the choice of a cost-saving wireless technology. Moreover, the capability to be easily interfaced with Internet could be another added value.

On such basis, the integration of sensor-networks with RFID systems appears to be the most practicable way. RFID technology, in fact, is quite inexpensive (passive RFID tags are as cheap as few euro-cents) and naturally compatible with Internet [20]. Moreover, as demonstrated in the following, it could be slightly modified in order to transmit sensor-like data.

Indeed, the scheme proposed in Fig.7 represents the actually designed and realized (patent pending) general purpose Sensor-Tag (S-Tag) connected to a generic sensor. The architecture of the S-Tag does not substantially differ from standard RFID systems, thus allowing us to maintain the compatibility between this system and devices already available and internationally standardized. The working principle is as

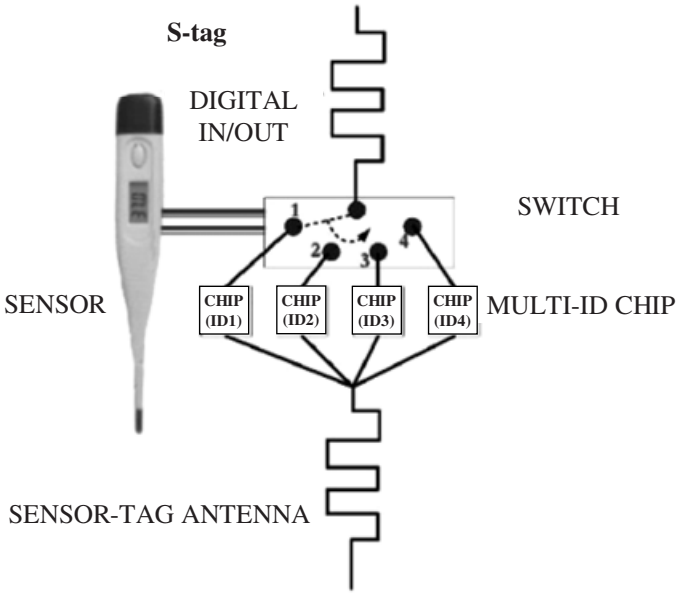

Fig. 7. A simplified scheme of the RFID sensor tag (S-tag)

easy as effective: data measured from a generic sensor are used as input to the S-Tag; when the Tag is in the region covered by the RFID reader, it sends back a signal containing a different combination of several 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 waveform.

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 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.

6 Conclusions

In this paper we presented a framework for context-aware computing and its prototypal implementation to a home-care scenario. The framework is based on a context codification obtained by integrating an ontology model and a rule-based representation. The main components of the proposed system are its multi-agent

behaviour, as well as the harmonization of heterogeneous technologies, such as agents, ontologies and rule-based inference engines, combined with the low-cost and flexible properties of RFID systems.

The overall system is now available and tested in real-life situations in home-health applications.

References

1. Weiser, M.: The computer for the 21st century. Scientific American, 94–104 (1991)
2. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. Int. J. Ad Hoc and Ubiquitous Computing 2(4), 263–277 (2007)
3. Paganelli, F., Giuli, D.: 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 2007) (2007) 0-7695-2847-3/07
4. Ko, E.J., Lee, H.J., Lee, J.W.: Ontology-Based Context Modeling and Reasoning for U-HealthCare. IEICE Trans. Inf. & Syst. E90–D(8), 1262–1270 (2007)
5. Chen, H., Finin, T., Joshi, A.: An ontology for context-aware pervasive computing environments. Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review (2003)
6. Gu, T., Pung, H.K., Zhang, D.Q.: A service oriented middleware for building context-aware services. Journal of Network and Computer Applications 28, 1–18 (2005)
7. Indulska, J., Sutton, P.: Location management in pervasive systems. In: CRPITS 2003 Proceedings of the Australasian Information Security Workshop, pp. 143–151 (2003)
8. Strang, T., Popien, C.L.: A context modeling survey. In: Workshop on Advanced Context Modeling, Reasoning and Management as Part of UbiComp 2004, The 6th International Conference on Ubiquitous Computing, pp. 33–40 (2004)
9. W3C, RDFS (RDF Vocabulary Description Language 1.0: RDF Schema), Recommendation (February 10, 2004), `http://www.w3.org/TR/rdf-schema/`
10. DAML site, http://www.daml.org
11. W3C, OWL Web Ontology Language Overview, Recommendation, (February 10, 2005), `http://w3.org/TR/2004/RDC-owl-features-20040210/`
12. Guarino, N.: Formal Ontology and Information Systems. In: Guarino, N. (ed.) Proceedings of the 1st International Conference on Formal Ontologies in Information Systems, FOIS 1998, Trento, Italy, pp. 3–15. IOS Press, Amsterdam (1998)
13. Chen, H., Finin, T., Joshi, A.: An ontology for context-aware pervasive computing environments. In: The Knowledge Engineering Review, vol. 18, pp. 197–207. Cambridge University Press, Cambridge (2003)
14. Dey, A.K., Abowd, G.D.: Toward a better understanding of context and context-awareness, GVU Technical Report, GIT-GUV-99-22 (1999)
15. Ernest Friedman-Hill “Jess In Action”, Edited by Manning
16. http://herzberg.ca.sandia.gov/jess/docs/52/rete.html
17. Jade, http://jade.cselt.it
18. Fipa, http://fipa.org/repository/index.html
19. Dockhorn Costa, P., Ferreira Pires, L., van Sinderen, M.: Architectural patterns for context-aware services platforms. In: 2nd International Workshop on Ubiquitous Computing (IWUC 2005), in conjunction with ICEIS 2005, Miami, USA (2005)
20. Finkenzeller, K.: RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification. John Wiley and Sons Ltd, Chichester