An Intelligent System for Distributed Patient Monitoring and Care Giving

Alessandra Esposito, Luciano Tarricone, Marco Zappatore and Luca Catarinucci

University of Salento, Lecce, Italy

Abstract. This paper presents a proposal for a context-aware pervasive framework. The framework follows a general purpose architecture, centered around an ontological context representation. The ontology provides the vocabulary upon which software agents communicate and perform rule-based reasoning, in order to assess the system reaction to context changes. The system components and their coordinated operations are described by providing a simple example of a concrete application in a realistic home-care scenario.

1 Introduction

Pervasive and context-aware systems are very appealing in a large number of application fields, such as health-care, robotics and industrial monitoring and control. Nevertheless, several technological challenges are still to be attacked before such systems can be adopted in daily activities. First of all, several heterogeneous and dispersed components must be enabled to interoperate continuously and in a transparent manner. Moreover, in order to adapt to context variations, the system must elaborate and interpret raw data sensed by variegated hardware and software sources, such as sensors and archives. Both requirements cannot be satisfied unless a robust, reusable and sharable context representation is available. Several approaches to context modeling are adopted in literature [2], ontology-based ones [3-5] being more and more recognized as the best suited [6]. In a pervasive context-aware system, ontology appealing features may be fully exploited by integrating them within a multi-agent and rule-based framework. Multi-agent paradigms organize applications around the 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.

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. These technologies have already been exploited and differently combined in other relevant works. Among them, the CoBrA [5] framework for intelligent meeting rooms is worth being mentioned. Similarly to our system, CoBrA adopts the agent paradigm and is centred around an OWL ontology. An ontology-based middleware for context-aware services (SOCAM) in smart space domains is proposed by Gu et al. [6]. Ubiquitous health-care is the focus of the ontology-based context model and context reasoning system presented in [4]. OWL and rule-based

Esposito A., Tarricone L., Zappatore M. and Catarinucci L. (2008).
An Intelligent System for Distributed Patient Monitoring and Care Giving.
In *Joint Proceedings of the 5th International Workshop on Ubiquitous Computing (IWUC 2008) 4th International Workshop on Model-Driven Enterprise Information Systems (MDEIS 2008) 3rd International Workshop on Technologies for Context-Aware Business Process Management (TCoB 2008)*, pages 12-21
DOI: 10.5220/0001739300120021

inference find an integration in the work of Paganelli and Giuli [3], which is tested in health-care environments as well. 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.

2 The Architecture

The system architecture (Fig. 1) follos a well known structure [7], which organizes context-aware systems into three levels: context sources, context management middleware and context consumer level. Context sources are conceptually partitioned into two groups: physical and virtual sensors [8]. Physical sensors identify hardware devices sensing context data, such as RFID, positioning systems. Virtual sources consist in software entities responsible for storing meaningful context data, such as GUIs for user preferences input, databases. Context representation utilities, modeling choices and software environments belong to the intermediate level. Finally, the context consumer layer includes all the entities, such as mobiles, Web interfaces, laptops, which interact with final users.

Figure 2 depicts our agent-based implementation of the above described architecture. A team of interoperating agents, which share a common knowledge representation and reason through an inference engine, is the system core. As shown in the figure, input data are provided by both physical and virtual context sources.

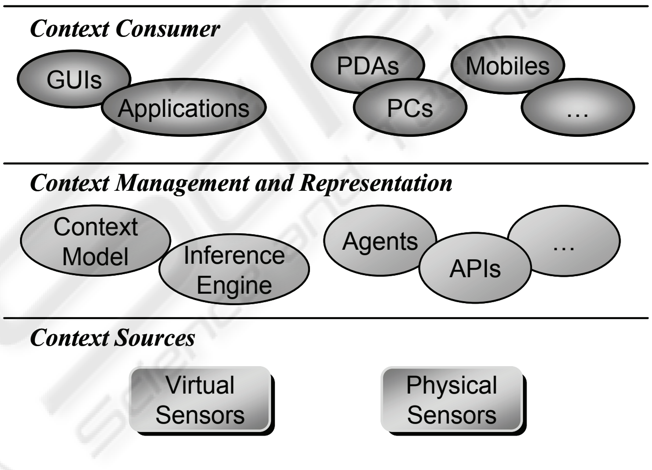

Fig. 1. Layered system architecture.

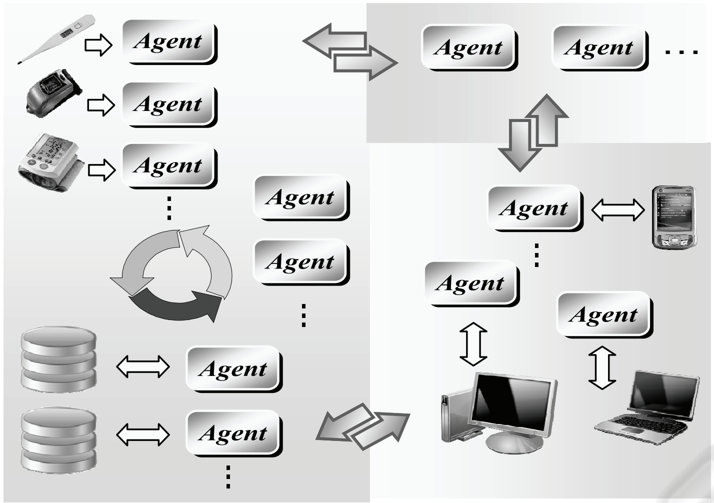

Fig. 2. System Prototype. The system follows a multi-agent paradigm and runs on distributed nodes connected by a local area network.

3 Implementation Choices

Many ontology languages have been defined such as Resource Description Framework Schema (RDFS) [9], DAML+OIL [10], and OWL [11]. 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 implemented our ontology by using the Protégé [12] graphical tool and adopted the Pellet [13] reasoner. The rule-based knowledge was implemented with Jess [14]. All the agents are implemented by using the Java Agent Development Environment (JADE) [15]. Inter-Agent communication is based on the FIPA ACL [16] which specifies the encoding and semantics of messages.

The three technologies are glued together by the OWL codification. The semantics of agent messages and their reasoning are built over OWL concepts and predicates, which have been matched with Jess and JADE vocabulary (Fig. 3). This process was made by taking advantage of the Bean Generator Protégé Plugin [17].

4 The Ontology

A common practise, when developing ontologies, is to adopt a *top level* (upper) shared conceptualization [18], on the basis of which domain ontologies are defined. This way of structuring knowledge facilitates sharing and reuse of ontologies in autonomous application domains. In our framework, the ontology provides the knowledge representation upon which agents reason and exchange messages. Therefore, the top

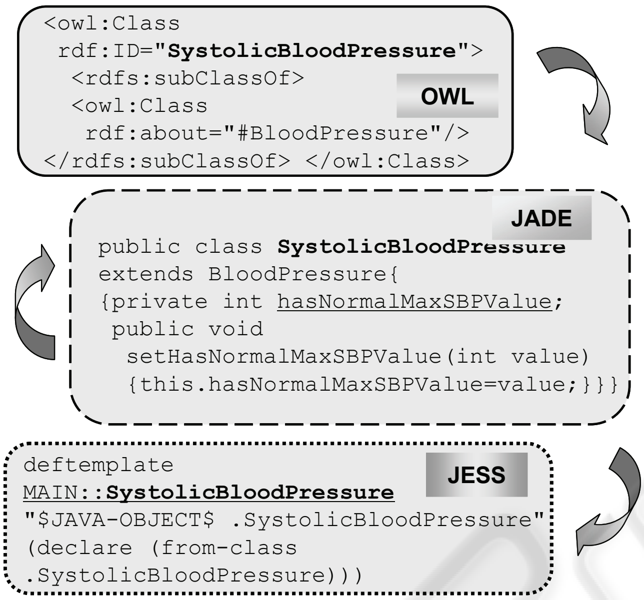

Fig. 3. The OWL vocabulary has been aligned with both agent inner context representation (in the form of Java classes) and Jess facts codification.

level concepts of our taxonomy reflect the JADE codification of ACL message "slots", as represented in the publicly available ontology named "OWLSimpleJADEAbstractOntology" [19]. This ontology allows agents to give a semantics to message components. It makes a distinction between "predicates" and "terms" (Fig. 4). The former can assume "*true*" or "*false*" values and are used to describe the current status of codified entities (e.g. "hasBloodPressure"). The latter are specialized into "Agent Actions" and "Concepts" classes. Agent Actions indicate the actions being performed by agents (e.g "sendNotification"). Concepts are used to characterize the entities upon which predicates and actions operate (e.g. "patient").

The vocabulary related to context entities was defined by starting from the widely accepted definition of context, provided in [21]: "Context is any information that can be used to characterize the situation of an entity." For example, "person", "device" and "TriggeredAction" (Fig.4 and Fig.5) are contextual entities which specialize into different concepts depending on the context subdomain they belong to.

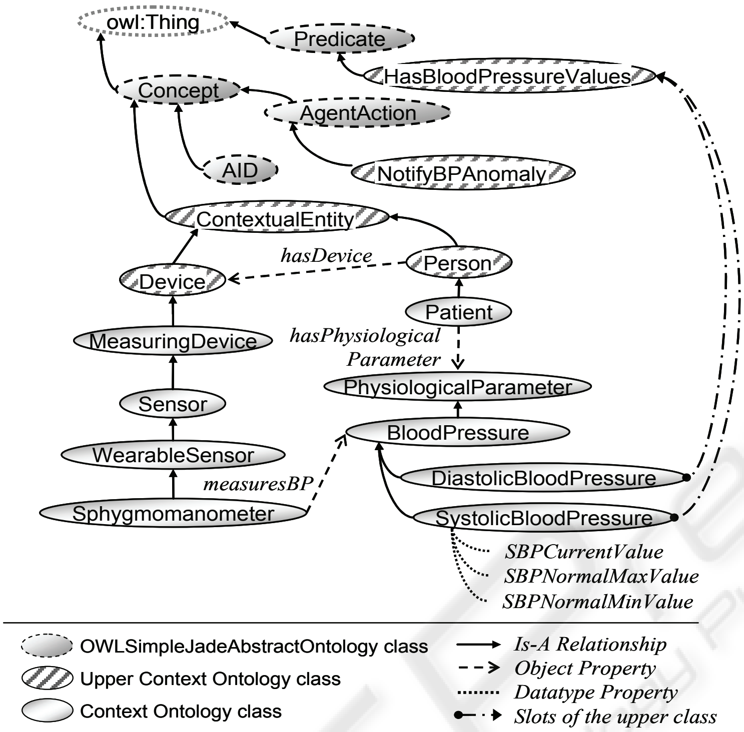

Fig. 4. Hierarchically structured ontology.

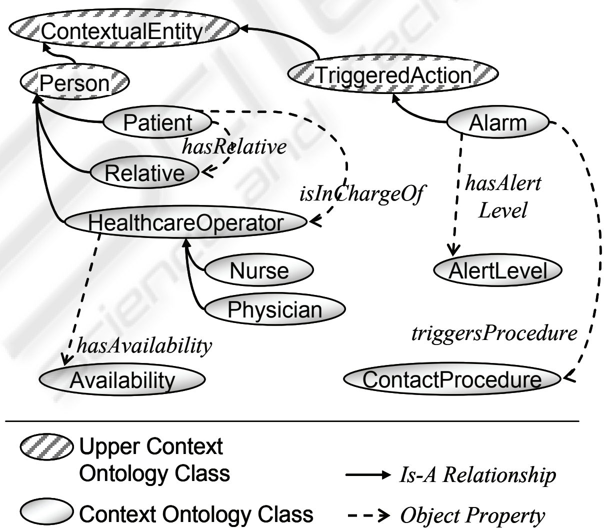

Fig. 5. Classes codifying some “application consumer environment” concepts.

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. First of all, 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") exceeds the patient thresholds ("spb-max" or "sbp-min"). When the rule is fired, the fact "status abnormal is true" ("sbp-s") is inferred and the action "notify the abnormal event" is activated ("send-bp-alert" action).

```
(defrule verify-SystolicBloodPressure
  ?f <- (SystolicBloodPressure
          (SBPCurrentValue ?sbp-c)
          (SBPNormalMaxValue ?sbp-max)
          (SBPNormalMinValue ?sbp-min)
          (SBPStatusAbnormal ?sbp-s))
   (test
     (or (> ?sbp-c ?sbp-max)
         (< ?sbp-c ?sbp-min)))
  =>
  (bind ?sbp-s true)
(send-bp-alert)
(retract ?f))
```

Afterwards, the context switches to a new situation, which we may call "procedure awareness", where it knows the activities to be performed in order to manage the alarm situation. 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
   ?f <- (patient   (hasPatientID ?pid)
       (SBPStatusAbnormal ?sbp-s)
       (DBPStatusAbnormal ?dbp-s)
       (HighAlertLevel ?hal))
  (test (eq ?sbp-s ?dbp-s true))
  =>
  (bind ?hal true)
(find-available-physician)
(retract ?f))
```

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

6 The Multi-Agent Framework

The proposed multi-agent framework (Fig. 6) partitions agents into three groups, having specific roles and responsibilities. Each agent embeds a Jess inference engine.

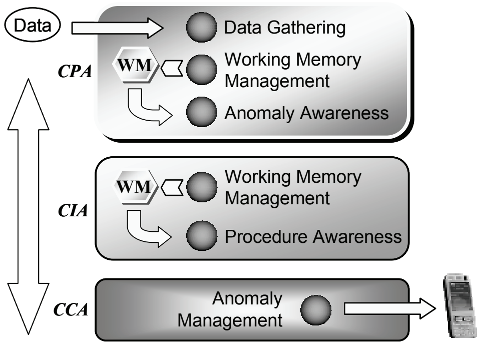

Fig. 6. The multi-agent framework. Each agent embeds a Jess Inference Engine.

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 [20], they interact with single domain CPAs, in order to aggregate context information coming from different sources. Both kinds of CPAs are also in charge of making low level context inference (in order to acquire "anomaly awareness") and putting relevant context information into the rule engine as facts.

Context Interpreter Agent (CIA). These agents are responsible for observing context changes sensed by CPAs, and, as consequence of these changes, to identify the set of actions to be performed by context consumer agents ("procedure awareness").

Context Consumer Agent (CCA). Context consumer agents are responsible for performing the actions identified by CIAs ("anomaly management"). Such actions provide the system reaction to context changes, and may assume diverse forms, such as the generation of a signal, the delivery of a notification or a web services request.

7 Agents Interaction

A simple example of agents interaction is shown in Fig. 7. A logical CPA gathers data coming from the sensor device and from the patient archive, thus being able to identify alarm events. When an alarm is identified, the CPA communicates the event to the CIA by means of an "inform" ACL message. The CIA, on its turn, requests information about the availability of physicians to the CPA responsible for the archive. Finally, an "inform" ACL message is forwarded to the suited CCA in order to communicate the alarm to the selected doctor.

```
(INFORM
 :sender  (agent-identifier :name
cpa_phys_bp@marco:1099/JADE)
 :receiver  (set ( agent-identifier :name
cpa_logic@marco:1099/JADE ) )
```

```
:content "( (HasBloodPressureValues ((DiastolicBloodPressure
:DBPNormalMinValue 50 :DBPNormalMaxValue 90
:DBPCurrentValue 95) (SystolicBloodPressure
:SBPNormalMinValue 90 :SBPNormalMaxValue 160
:SBPCurrentValue 97))
(p_id :27)
(bp_timestamp 1205453552515) )"
 :language  fipa-sl  :ontology  UHC)
```

The above code snippet shows how ACL fields (highlighted in bold and italic characters) are perfectly aligned with the concepts codified in our ontology.

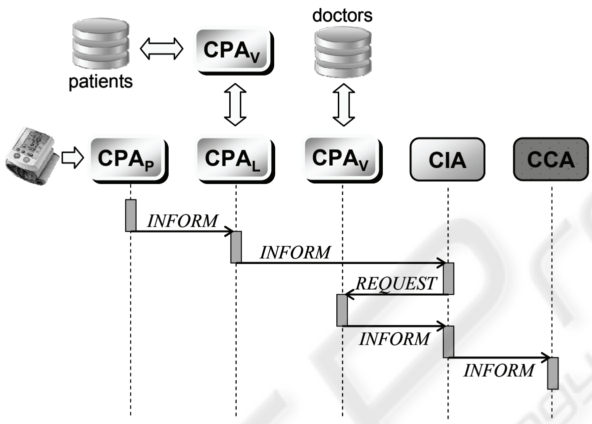

Fig. 7. An example of agents interaction.

8 Physical Sensors

The capability to measure a physical value is only one of the requirements a physical source should satisfy. The measured data, for instance, should be sent to a data collector by using a wireless connection, the choice of which is not univocal: wi-fi, Bluetooth, GPRS, UMTS, GSM are only a few of the possible candidates. Moreover, in order to guarantee a capillary diffusion of physical sources, the benefit/cost ratio must be considered. Finally, the capability to be easily interfaced with Internet could be another added value. On such basis, the integration of sensor-networks with RFID systems seems to be the most practicable way. RFID technology, especially in the UHF band, is 1) quite inexpensive (passive RFID tags are as cheap as few eurocents), 2) naturally compatible with Internet [22], and 3) guarantees a reading range (a few meters) which is adequate for the proposed application. Moreover such technology can be slightly modified in order to transmit sensor-like data.

Figure 8 represents the actually designed and realized (patent pending) general purpose Sensor-Tag (S-Tag). 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 easy as effective: data measured from a generic sensor are used as

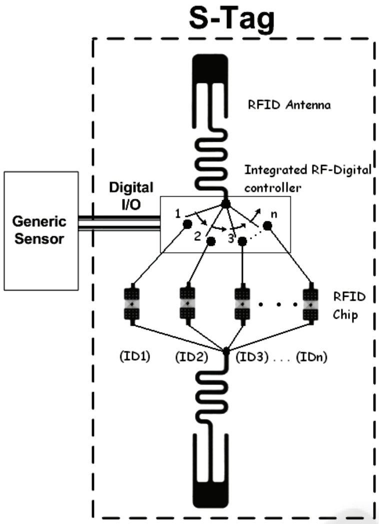

Fig. 8. A simplified scheme of the RFID sensor tag (S-tag).

input to the S-Tag. When the Tag is in the region covered by the RFID reader, it sends back a signal containing a combination of identity codes (IDs) depending on the value of the input, thus facilitating the transmission of sensor data. As apparent from the picture, the sensor is an external unit. In such a way, generic sensors, with the only requirement of being equipped with 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, also the implemented technological innovation is reasonably inexpensive, thanks to an accurate electromagnetic design of the tag antenna and of the microwave circuit (microcontroller, RF-switch and so on).

9 Conclusions

In this paper we presented a framework for context-aware computing and its implementation in a home-care scenario. The framework is based on a context codification obtained by integrating an ontology model and a rule-based representation. The proposed system is the result of an effort of harmonizing heterogeneous technologies, such as agents, ontologies and rule-based inference engines, combined with the low-cost and flexible properties of RFID systems. The system is now available in a prototypal implementation and is going to be tested in large scale real-life situations. Indeed, we are currently working to improve the system by 1) enhancing the rule-base with historical data about patient care, 2) increasing the number and the heterogeneity of input sources in order to definitely assess the system scalability, 3) enrich the system components from the consumer side in order to enable a pleasant and easy interaction with end-users.

References

1. Weiser, M.: The computer for the 21st century, *Scientific American*, 94–104 (1991)
2. Baldauf, M., Dustdar, S. and Rosenberg, F.: A survey on context-aware systems, *Int. J. Ad Hoc and Ubiquitous Computing, Vol. 2, No. 4, 263—277, (2007)*
3. Paganelli, F. and Giuli, D.: An Ontology-based Context Model for Home Health Monitoring and Alerting in Chronic Patient Care Networks, AINAW '07 (2007)
4. Ko, E. J., Lee, H. J. and Lee, J. W.: Ontology-Based Context Modeling and Reasoning for U-HealthCare, IEICE Trans. Inf. & Syst., Vol.E90–D, N.8, 1262--1270, (2007)
5. Chen, H., et al.: An ontology for context-aware pervasive computing environments, *The Knowledge Engineering Review*, Cambridge University Press, Vol.18, 197–207 (2003)
6. Gu, T., Pung, H. K. and. Zhang, D. Q.: A service oriented middleware for building context-aware services, Journ. of Network and Computer Applications, 28, Elsevier , 1--18 (2005)
7. Indulska, J. and Sutton, P.: Location management in pervasive systems, *CRPITS'03 Proc. of the Australasian Information Security Workshop*, 143–151, (2003)
8. Strang, T. and Popien, C.L.: A context modeling survey, Workshop on Adv. Context Modeling, Reasoning and Management, UbiComp 2004, pp.33–40 (2004)
9. RDFS W3C Recommendation 10, Feb. 2004, http://www.w3.org/TR/rdf-schema/
10. DAML+OIL Web site http://www.daml.org (last accessed March 2008)
11. OWL W3C Recommendation, http://w3.org/TR/2004/RDC-owl-features-20040210/
12. Protégé Web Site: http://protege.stanford.edu (last accessed March 2008)
13. Pellet Web Site: http://www.mindswap.org/2003/pellet/ (last accessed on March 2008)
14. Friedman-Hill, E. "Jess In Action", Edited by Manning (2003)
15. JADE web site http://jade.cselt.it (last accessed March 2008)
16. FIPA web site http://fipa.org/repository/index.html (last accessed March 2008)
17. Bean Generator Web Site: http://protege.cim3.net/cgi-bin/wiki.pl?OntologyBeanGenerator
18. Guarino, N.: Formal Ontology and Information Systems. *Proc. of the 1st Int. Conf. on Formal Ontologies in Information Systems*, FOIS'98, Trento, Italy, pp 3-- 15 (1998)
19. OWLSimpleJADEAbstractOntology http://jade.tilab.com/dot/CLOntoSupport.pdf
20. Dey, A.K. and Abowd, G.D.: Toward a better understanding of context and context-awareness, GVU Technical Report, GIT-GUV-99-22, (1999)
21. Dockhorn Costa, P. et al.: Architectural patterns for context-aware services platforms. *2nd Int. Workshop on Ubiquitous Computing (IWUC 2005)*, ICEIS 2005, Miami, USA, (2005)
22. Finkenzeller, K. "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification" Publisher: John Wiley and Sons Ltd (2006)