2009 International Conference on Parallel and Distributed Computing, Applications and Technologies

# CASanDRA: A framework to provide Context Acquisition Services ANd Reasoning Algorithms for Ambient Intelligence Applications

Ana M. Bernardos, Paula Tarrío, José R. Casar
Universidad Politécnica de Madrid
Madrid, Spain
{abernardos, paula, jramon}@grpss.ssr.upm.es

***Abstract*— The development of ambient intelligence (AmI) applications usually implies dealing with complex sensor access and context reasoning tasks, which may significantly slow down the application development cycle when vertically assumed. To face this issue, we present CASanDRA, a middleware which provides easily consumable context information about a given user and his environment, retrieving and fusing data from personal mobile devices and external sensors. The framework is built following a layered service oriented approach. The output data from every CASanDRA's layer are fully accessible through semantic interfaces; this allows AmI applications to retrieve raw context features, aggregated context data and complex 'images of context', depending on their information needs. Moreover, different query modes –subscription, event-based, continuous and on-demand- are available. The current 'mobile-assisted' version of CASanDRA is composed by a CASanDRA Server, developed on an applications container and hosting the system intelligence, and CASanDRA Lite, a mobile client bundling a set of sensor level acquisition services. How an AmI application may be effortlessly built on CASanDRA is described in the paper through the design of an 'Ambient Home Care Monitor'.**

***Keywords-context-aware systems, ubiquitous frameworks, ambient home care systems, sensor networks***

## I. Introduction

Ambient Intelligence applications, those conceived to work in an environment populated with sensors and aiming at adapting their performance to their users' personal characteristics and needs, are intensive consumers of data coming from both physical and virtual sensing sources. Accessing those data and inferring useful and reliable information may be a complex process, which usually hinders the development of AmI services. For this reason, since 1994, when Schilit and Theimer coined the term 'context-awareness', different solutions to manage context data have been proposed.

In this paper, we describe CASanDRA, our framework to provide Context Acquisition Services and Reasoning Algorithms to be directly consumed by any type of application needing to handle context information. CASanDRA decouples the acquisition and inference tasks from the application development by offering a set of interfaces for information retrieval. The framework design is based on a data fusion-oriented architecture. CASanDRA has been designed to be easily scalable; it simplifies the integration of both new sensor access interfaces and fusion algorithms deployment, as it also aims at serving as a testbed for research.

CASanDRA is able to offer its services (presence, location, ambient data, agenda information, biometric diagnosis, etc.) under the assumption that a given infrastructure is available (WiFi, Bluetooth, ZigBee, pressure mats, RFID tags, etc.) and that the user is carrying a mobile device with connectivity capabilities and, if necessary, other personal wireless devices (such as biometric sensors). Nevertheless, the framework is able to adapt its service offer depending on the technologies availability.

Section 2 explains what makes CASanDRA different from some other frameworks in literature. In Section 3, CASanDRA's features are detailed. Section 4 describes the middleware architecture and the acquisition services currently available. Section 5 shows how CASanDRA may be used to build context-aware applications through an example. Conclusions and further work are in Section 6.

## II. Context-aware platforms: current situation and challenges

In the last decade, a good number of middleware proposals to handle context information have appeared in literature. As stated below, they differ in their architecture approaches, data modeling and reasoning systems.

One of the first frameworks was the Context Toolkit [1]. It is based on widgets, uses tuple data modeling and does not provide reasoning capabilities (although subsequent works [2] propose to use Dempster-Shafer theory to handle uncertainty and derive complex contexts). Gaia [3] aims at providing a meta-operative system, built on the concept of active space, which coordinates and combines information from heterogeneous devices in a physical space. It uses a Context File System, a virtual file hierarchy based on

978-0-7695-3914-0/09 $26.00 © 2009 IEEE
DOI 10.1109/PDCAT.2009.51

context information tags, which structures directories on the context predicates. Solar [4] is an event-based architecture, composed by stars and planets. Stars handle external application subscriptions to event and planets execute Solar's operators (filters, transformers, mergers and aggregators). Each application needs to be able to build an operation tree to gather and aggregate the context information it needs. CoBrA [5] is an agent-based architecture. The context broker agent maintains a shared context model for the rest of agents in the space. It performs rule-based inference on OWL language. The Context Manager Framework [6] has a central context manager which uses a blackboard and fuzzy logic for reasoning purposes. SOCAM [7] uses ontologies for context modeling and reasoning, as it aims at converting physical spaces into semantic ones. CORTEX [8] configures distributed sentient objects, independent software components capable of acquiring context data and performing inference. These components share information with nearby objects which are dynamically discovered. CoCA [9] and MARKS [10] faces context management in ad-hoc networks.

There are also some proposals of device-centric middlewares. MobiPADS [11] proposes an architecture based on mobilets, structures composed by a master (in the phone) and a slave (in a central server). Citron [12] is conceived to be fully operative in the mobile device: it uses a blackboard approach to handle information from workers (which access sensors), which is represented in tuples in the citron space.

Apart from these contributions, many other exists: ALICE, Aura, CARMEN, CARISMA, Capnet, Cooltown, EasyLiving, LIME, MiddleWhere, MiLaN, Mobiware, one.world, Oxygen, TSpace, CASS, TEA, WildCAT, XMIDDLE, MoCoA or CoSAr.

In spite of diversity, there is not an accepted methodology to handle the development of context-aware applications, even if it has been recognized that a unified treatment both of data models and architectures would accelerate the production of AmI context-aware services. CASanDRA is built following a service-oriented application-centric approach, aiming at providing simple semantic interfaces. That means that on-line access to its services, just by using http queries, making possible to connect a whole environment and its inhabitants transparently to the Internet.

## III. CASANDRA DESIGN PRINCIPLES

### A. *CASanDRA philosophy for 'cloud context aware services'*

CASanDRA aims at facilitating the deployment of AmI applications hosted by external service providers in remote servers. The framework provides a set of functionalities that simplify the access to context data which are gathered through mobile and personal devices and the networks around them. That means that third parties developing context-aware software do not have to deal with acquisition, communications and general inference tasks, but just define their context information needs to receive context data updates, or query the system for specific data. With this approach, we are assuming that context-aware services will not necessarily be natively installed in the device, but will be developed as Internet services, accessible through a data connection. Nevertheless, native applications may also profit from CASanDRA services.

### B. *Components*

The framework is shaped into two different components: CASanDRA Server and CASanDRA Lite. CASanDRA Server is the infrastructure component, which hosts inference, user registry and consumer application registry. The structure and interfaces defined for CASanDRA Server make its replication easy, so it is possible to have a 'cluster' of distributed servers to host different fusion and inference algorithms. Additionally, CASanDRA Lite is a client performing acquisition tasks enabled by the mobile device sensing capabilities (e.g. acceleration measurements from in-device inertial systems, nearby networks signal strength measurements, etc.). This component may also perform light inference functions (e.g. in-device activity estimation, etc.).

### C. *Service Oriented Architecture*

The architecture of CASanDRA is inspired in the SOA paradigm, following its key concepts of 'visibility', 'interaction' and 'effect' for the framework's capabilities and basing on 'services' to build these capabilities. As defined in the standard [13], a 'service' needs to be able to perform work for another, specifying the work offered for another and also offering to another to perform the work. CASanDRA services have interfaces to be externally invoked, and publish their functionalities for applications to use them.

### D. *User-centric approach*

In CASanDRA, the control of data acquisition and transfer procedures always remain in hands of the target user, as he/she will have control over CASanDRA Lite functionalities and will handle which applications can access CASanDRA services, and when and how they will be able to do it. Moreover, the user will be able to set external context providers to enrich his personal context data. CASanDRA is also prepared to deal with external sources of context data in a transparent way for data providers.

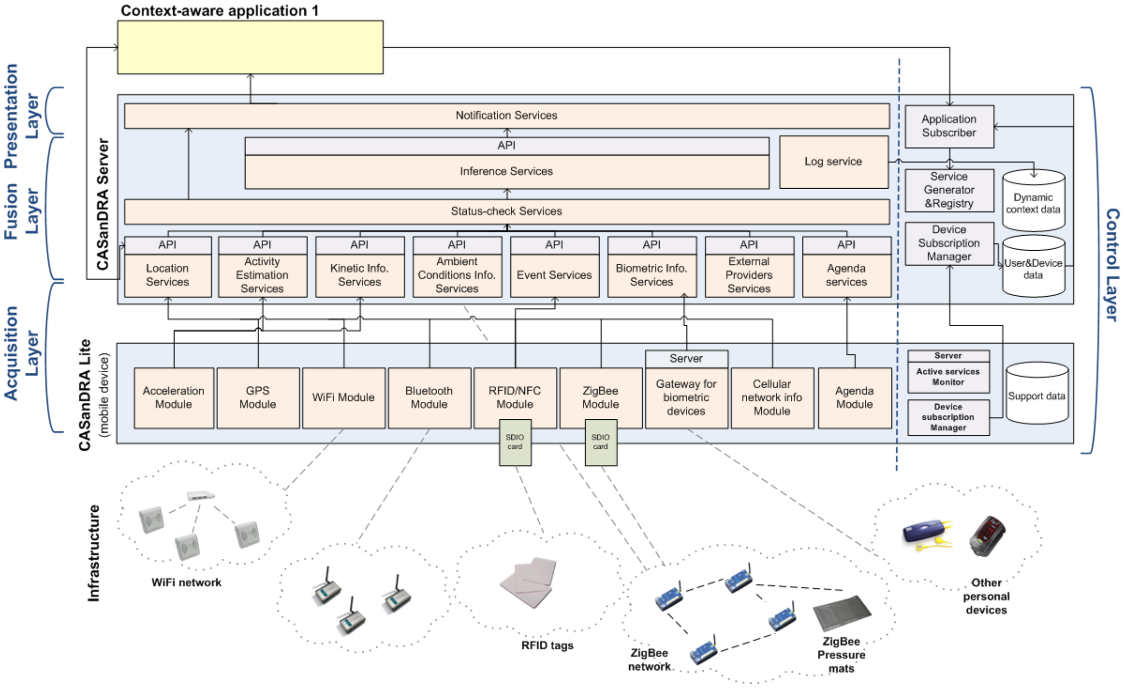


Figure 1. CASanDRA architecture.

### E. Fusion services

CASanDRA is conceived to host different data fusion services, which may compute both simple context features (such as location or activity) but also complex context 'images', describing, for example an 'emergency' status.

### F. Data interfaces

Standardized interfaces at different levels of abstraction allow applications to access context data in their raw form but also in an aggregated way. For example, it may be possible to access three-axis acceleration data coming from a mobile device to externally compute the user's activity, but also to use CASanDRA's user's activity estimation service. Semantic technology (XML) makes data access possible in a standardized way.

### G. Information retrieval methods

CASanDRA facilitates a consumer application to recover context information through different procedures. The first mode is by querying a particular service for information whenever the application requires it, in a continuous or in an occasional call. For example, the first mode is suitable to receive heart rate samples from a wireless heart monitor to build an electrocardiogram, while the latter is suitable to query for location in a given moment.

Apart from these modes, it is also possible to subscribe to a service in order to receive periodic updates for a given context information. Moreover, the application may also establish conditions for CASanDRA to evaluate and trigger notifications to the consumer applications when context changes (events) are detected.

### H. Quality of context

Dealing with uncertainty is a fundamental issue when handling context information, so being aware of the information quality is needed to allow the consumer applications correctly reason on their own purpose. As a consequence, quality of service is directly related to quality of context in context-aware applications. CASanDRA aims at handling quality of context by offering context data together with a reliability parameter. For example, see Figure 2b for the case of a proximity ZigBee location method providing its probability of estimation correctness.

## IV. The CASanDRA Framework

### A. Architecture description

CASanDRA structure is represented in Figure 1. It partially implements the fusion architecture described in [14] The CASanDRA stack is composed by three main layers: the acquisition layer, the fusion layer and the presentation layer.

The acquisition layer is divided into the Server and the Lite components, and it hosts the lowest access functions to sensors, preprocessing and signal adequacy functions. The fusion layer performs feature extraction and upper level context inference. The presentation layer includes the notification processes.

Additionally, a vertical control layer hosts the functionalities needed to handle the query methods (on-demand, event-based, subscription and continuous modes). Available services are implemented into servlets, which may be invoked and executed from and external consumers (see Figure 2b for an example).

The CASanDRA Service Registry makes easier to retrieve the information of the deployed services through XML interfaces: sensor type, service description, standard input, communication port for socket connectivity and information retrieval methods may be consulted by invoking <http://[server_ip]:8180/serviceregistry/getServices>.

CASanDRA's fusion layer is prepared to dynamically generate/activate the available services to check if a certain condition is fulfilled. When using subscription or continuous modes of information retrieval, consumer applications show their needs and the system configures the services to deliver the requested data. For example, if an application needs to watch out if a given user is moving into a specific zone, and this 'location monitor service' is not available, CASanDRA automatically activates it. Afterwards, if other consumer application also needs it, the Subscription Manager will deliver the information to both applications. This process of incremental service composition is represented in Figure 2a, which shows a view of an operational moment in CASanDRA. In the next section, a brief description of some of the available acquisition and fusion services is done.

### B. *Basic context information services*

CASanDRA system bases on an infrastructure which enables retrieving signal strength, beacon events, accelerometers data or biometric signals. On this information, the acquisition layer provides context estimators and the fusion layer is able to enhance them and, additionally, use them to dynamically configure application-specific inference services. Following there is a review of the basic context estimation services integrated in CASanDRA:

1. Location estimation. Indoor location data are provided through algorithms built on top of WiFi, Bluetooth, ZigBee networks and RFID (Near Field Communications) beacon based systems. Depending on the available technologies and the needed accuracy, the 'Location Manager' decides which of them to use. When outdoors, GPS and cell-id are the location technologies. By now, CASanDRA provides signal strength location algorithms (based on real and simulated fingerprinting and propagation models), offering symbolic location (zone-based) and physical location (coordinates). The average error is 2.5 meters for physical location. In Figure 2b, the standard output for the ZigBee location service is shown.

2. Proximity detection. Based on the availability of RFID (NFC) tags which only allow short distance readings, a proximity detection algorithm is offered in CASanDRA. This type of event detection facilitates the use of calibration methods for the location algorithms above. For example, a Least Mean Squares method which aims at adapting a general propagation model to the particularities of the environment may use these fixed tags as beacons.

3. Kinetic information. CASanDRA Lite have a set of functions to access the accelerometers in the mobile device. Those acceleration data are afterwards sent to CASanDRA server. Speed and acceleration are available from this service.

4. Activity inference. CASanDRA aims at having a set of activity inference algorithms, capable of determining if a person is standing, sit, walking, going upstairs or downstairs, in the elevator, etc. CASanDRA server hosts a service which integrates acceleration and positioning data in order to estimate activity.

5. RFID readings detection. NFC tags may contain information embedded in. This service manages readings events and forwards them to upper inference levels in order to manage the content in the tags.

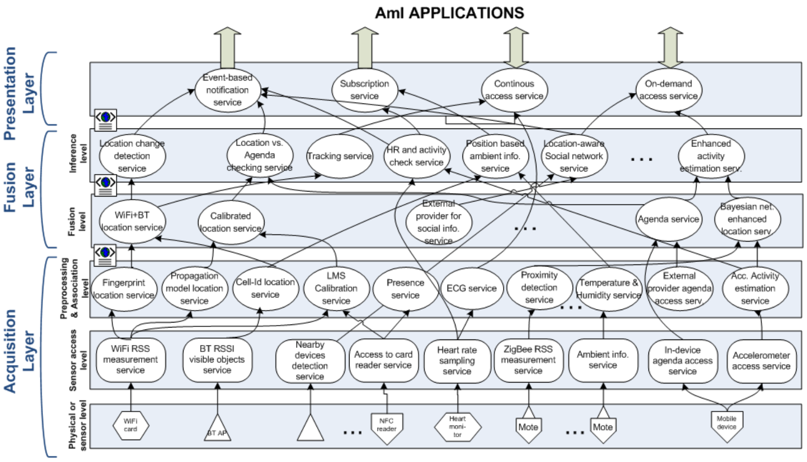
6. Ambient information. Ambient information is acquired from a ZigBee (IEEE 802.15.4) wireless network, which is composed by a number of nodes capable of managing multihop routing.

7. Biometric data access. For some ambient intelligence applications, biometric data from a given user are requested. For this reason, CASanDRA allow to get heart beat rate and oxymetry information from two different Bluetooth sensors. In these cases, the mobile device acts as a gateway, forwarding the data streams into CASanDRA server. When an application is willing to build the electrocardiogram signal, it is necessary to continuously retrieve data. In this case, the mobile device enables a server in one of its ports and CASanDRA server is able to get data from it.

8. Access to virtual sensors. There are many external providers which may store context-related information, which is retrievable through semantic interfaces. It is the case of weather information or services enabling to connect and share real time sensor data from objects. CASanDRA offers an API to register and query those services to opportunistically use the information they provide in upper inference layers.

9. Calendar service. Information about the user agenda may be available both in-device and through virtual sensors (web calendars). This service merges the agenda data coming from both sources.

Each service is described through an HTTP XML API which contain the connection data which to retrieve the context features (Figure 2).


AmI APPLICATIONS
Presentation Layer
Fusion Layer
Acquisition Layer
Inference level
Fusion level
Preprocessing & Association level
Sensor access level
Physical or sensor level
Event-based notification service
Subscription service
Continous access service
On-demand access service
Location change detection service
Location vs. Agenda checking service
Tracking service
HR and activity check service
Position based ambient info. service
Location-aware Social network service
Enhanced activity estimation serv.
WiFi+BT location service
Calibrated location service
External provider for social info. service
Agenda service
Bayesian net enhanced location serv.
Fingerprint location service
Propagation model location service
Cell-Id location service
LMS Calibration service
Presence service
ECG service
Proximity detection service
Temperature & Humidity service
External provider agenda access serv.
Acc. Activity estimation service
WiFi RSS measurement service
BT RSSI visible objects service
Nearby devices detection service
Access to card reader service
Heart rate sampling service
ZigBee RSS measurement service
Ambient info. service
In-device agenda access service
Accelerometer access service
WiFi card
BT AP
NFC reader
Heart moni-tor
Mote
Mote
Mobile device

```
<?xml version="1.0" encoding="ISO-
8859-1"?>
<!--id, zone & error probability-->
   <proxiProbabilities>
   <id>"5"</id>
   <objectZone>
     <Zone>"living room"</Zone>
     <idZone>"6"</idZone>
<probability>"97.6744%"</probability>
   </objectZone>
   <objectZone>
     <Zone>"bathroom"</Zone>
     <idZone>"4"</idZone>
<probability>"1.5503%"</probability>
   </objectZone>
   <objectZone>
     <Zone>"corridor"</Zone>
     <idZone>"5"</idZone>
<probability>"0.7752%"</probability>
   </objectZone>
   </proxiProbabilities>
```

Figure 2. a) Service composition. b) Example of the ZigBee location service XML interface. It is the response to the HTML query http://server_ip:8180/locationZigbee/proximityProb?id=5.

### C. *Implementation technologies*

CASanDRA Server is implemented in Java, into an application container (Apache Tomcat) running on a Linux server. Databases are built on MySQL. CASanDRA Lite runs in a mobile device, and clusters different type of acquisition software, dealing with low level sensor access (such as RSS measurement, accelerometer access, RFID and ZigBee SD cards access) and acting as a server for special cases. CASanDRA Lite prototype is built for Windows Mobile devices, using different programming languages for sensor access (C#, C++) and the J9 as Java Virtual Machine.

## V. Case of Use: An Ambient Home Care Monitor based on CASanDRA

By now, CASanDRA has been used to develop a number of context applications on top of it. For example, CASanDRA has served as a basis for an Ambient Home Care Monitor (AHCM). An AHCM is a system designed to check and give up-to-date information about an elderly to his caregivers. The AHCM service contains the logic to monitor the elderly state, and takes the decision on when it is necessary to deliver an alarm. When no-emergency status is registered, the AHCM periodically generates different messages to be sent and published in the elderly's microblog website (eg. Twitter style), which is available to authorized readers through common mobile widgets and web interfaces.

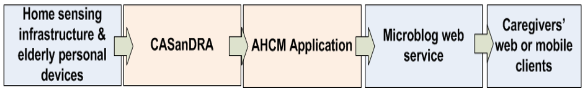

Home sensing infrastructure & elderly personal devices
CASanDRA
AHCM Application
Microblog web service
Caregivers' web or mobile clients

Figure 3. Components flow for the AHCM service.

The AHCM service has been deployed in our laboratory (Fig. 4), under the assumption that a realistic ambient intelligence home network will count on ZigBee wireless sensors (suitable for location and ambient data acquisition), ZigBee equipped pressure mats (placed at the house entrance and on seats and beds), biometric devices (such as an oxymeter or a heart rate monitor) and RFID tags (in a near future, capable of enabling automatic activation of appliances and services).

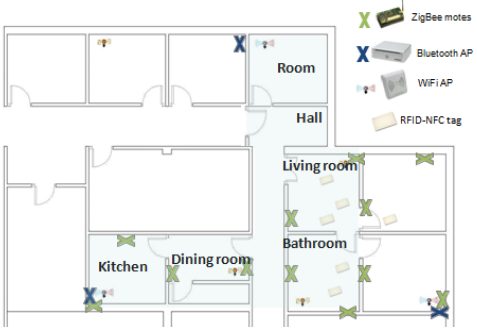


Figure 4. Map of an elderly's simulated home.

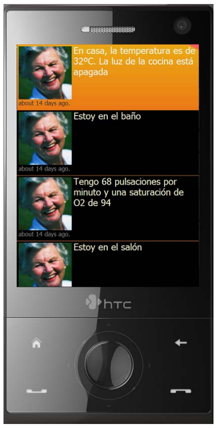
Table 1 shows some messages that the AHCM posts to the microblog site on the information from CASanDRA. In this case, the AHCM application, demands CASanDRA to configure different services to detect events (see the E legend in Table 2) or receive periodic updates on different sensors (legend S). Figure 5a shows the XML which defines the AHCM needs for information. Afterwards, the AHCM logic decides which messages to post in the microblogging site (Figure 5b).

TABLE I. Example of AHCM logic for message posting. Query mode: (E) Event service, (S) Subscription service.

| Event | Action | Microblog message |
|---|---|---|
| (E) Stepping on the pressure mat at the entrance of the house (outdoors). | On the event detection, the service retrieves the weather forecast from an external provider to complete the post. | 'I am leaving. 34º outside!'<br>'I'm back home' |

| | | |
|---|---|---|
| (S) Detecting sustained pressure (over 1hour) in a mat on an armchair. | Generate a message in order to suggest the caregiver to ask the user to move. | 'I have been sit for more than an hour' |
| (E) Detecting that the user's location has changed. | (1) Posting a message<br>(2) If no location change has been detected in an hour, the system posts a special message. | (1) 'I'm in the kitchen'<br>(2) 'I am still in the living room' |
| (S) Normal data from an oxymeter. | Post biometric data three times a day if the values. | 'I have a heartbeat of 70 (bpm) and my oxygen blood sat. level is 98' |
| (S) Data from ambient sensors. | Location-aware ambient info posted each 3h. | '24ºC at the living room' |
| (E) Lights on unneededly or irregularly. | If any light in the house is switched on when the user is not there, a special post is generated. | 'The kitchen light is on'. |
| (E) RFID tag reading detection. | Post a message identifying the appliance which has been activated. | 'I'm in front of my computer'<br>'Rehab time!' |
| (S) The time for an appointment or medication intake is approaching. | Each hour, the system will publish the agenda for the next hour. | 'Aspirin at 19:00h'<br>'Doctor at 17:30' |

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<session>
<applicationId>AHCM</applicationId>
<user>00:13:02:5F:9A:08</user>
<info>
  <location>
    <event>zoneChange</event>
  </location>
  <pressureMats>
    <matID>entrance</matID>
    <event>pressed</event>
    <subsSampling>1</subsSampling>
  </pressureMats>
  <oxymeter>
    <subsSampling>8</subsSampling>
  </oxymeter>
  <ambient>
    <subsSampling>3</subsSampling>
  </ambient>
  <light>
    <event>onUserLocation</event>
  </light>
  <NFCTags>
    <event>on</event>
  </NFCTags>
  <calendar>
    <calendarID>Google</calendarID>
    <calendarUser>[user]</calendarUser>
    <calendarPwd>[pwd]</calendarPwd>
    <event>1</event>
  </calendar>
</info>
</session>
```



En casa, la temperatura es de 32ºC. La luz de la cocina está apagada

about 14 days ago.

Estoy en el baño

about 14 days ago.

Tengo 68 pulsaciones por minuto y una saturación de O2 de 94

about 14 days ago.

Estoy en el salón

Figure 5. a) AHCM request for information and b) Messages in a standard off-the-shelf mobile interface.

## VI. Discussion and Further Work

CASanDRA is still evolving and enhancing its capabilities. In particular, quality of context management and sensors' feedback control mechanisms are currently in the research agenda.

CASanDRA's architecture is applicable both to 'network-based' and 'device-based' implementation approaches. A 'device-based' design to make possible autonomous mobile middleware context acquisition and reasoning is currently being developed under the concept of CASanDRA Mobile. Light inference strategies are being addressed, and different options to face the general architecture are being evaluated. In particular, mobile OSGi is being considered as an option to facilitate the maintenance of the services life cycle. Moreover, CASanDRA Mobile will be oriented to facilitate ad-hoc context-sharing among mobile devices and direct interaction with augmented objects. The objective is to facilitate the exploration of the Internet of Things paradigm.

## Acknowledgments

This work has been supported by the Government of Madrid under grant S-0505/TIC-0255 and by the Spanish Ministry of Science and Innovation under grant TIN2008-06742-C02-01.

## References

[1] A.K. Dey, G.D. Abowd, D. Salber. "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications", *Human-Computer Interaction (HCI) Journal*, Vol. 16 (2-4), pp. 97-166, 2001.

[2] H. Wu. "Sensor Data Fusion for Context-Aware Computing Using Dempster-Shafer Theory". *PhD Dissertation*, Tech. report CMU-RI-TR-03-52, Robotics Institute, Carnegie Mellon University, 2003.

[3] M. Román, C. Hess, R. Cerqueira, A. Ranganat, R.H. Campbell, K. Nahrstedt. "Gaia: A Middleware Infrastructure to Enable Active Spaces". *IEEE Pervasive*, Vol. 1, pp. 74-82, 2002.

[4] G. Chen, D. Kotz. "Solar: A pervasive-computing infrastructure for context-aware mobile applications". *Dartmouth Computer Science Technical Report TR2002-421*, Dartmouth College, 2002.

[5] H. Chen, T. Finin, A. Joshi. "An Intelligent Broker for Context-Aware Systems"**.** *Adjunct Proceedings of Ubicomp 2003*, pp. 183-184, UbiComp, 2003.

[6] P. Korpipää. "Blackboard-based software framework and tool for mobile device context awareness". PhD Dissertation, 225p. VTT Publications 579. Espoo, Finland, 2005.

[7] T. Gu, H.K. Pung, D.Q. Zhang. "A Middleware for Building Context-Aware Mobile Services". *IEEE Vehicular Technology Conference*, Vol. 5, pp. 2656- 2660, IEEE Computer Society, 2004.

[8] C. Sørensen, M. Wu, T. Sivaharan, G.S. Blair, P. Okanda, A. Friday, H. Duran-Limón. "A Context-Aware Middleware for Applications in Mobile Ad Hoc Environments". *Proc. 2nd Workshop on Middleware for Pervasive and* Ad hoc *Computing*, pp. 107-110, NY: ACM, 2004.

[9] D. Ejigu, M. Scuturici, L. Brunie, "Hybrid Approach to Collaborative Context-Aware Service Platform for Pervasive Computing", *Journal of Computers*, Vol. 3, No. 1, 2008.

[10] M. Sharmin, S. Ahmed, S.I. Ahamed, "MARKS for Mobile Devices of Pervasive Computing Environments". *Proc. of the Third Int. Conf. on Information Technology*, pp. 306 – 313, 2006.

[11] A.T.S. Chan, .S. Chuang (2003). MobiPADS: A Reflective Middleware for Context-Aware Mobile Computing. *IEEE Transactions on Software Engineering*, Vol. 29, No. 12.

[12] T. Yamabe, A. Takagi, T. Nakajima (2005). Citron: A Context Information Acquisition Framework for Personal Devices. *Proc. of the 11th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05),* pp. 489-495, IEEE Computer Society.

[13] OASIS Standard. "Reference Model for Service Oriented Architecture 1.0." 2006.

[14] A.M. Bernardos, P. Tarrío, J.R. Casar, "A data fusion framework for context-aware mobile services". *Proceedings of the IEEE Int. Conf. in Multisensor Fusion and Integration for Intelligent Systems*, pp. 606-613, Seoul, August 20-22, 2008.