Supporting Context-Aware Applications for Eldercare

Dario Bottazzi, Tarik Taleb, Mohsen Guizani, and Abdelhakim Senhaji Hafid

Abstract: **Extended life expectancy and a decrease in fertility rates are dramatically increasing the number of seniors who will eventually need professional care assistance. Although there is no clear technical solution to this problem, recent advances in ubiquitous computing offer opportunities to assist elders in their residence, thus reducing the need for professional assistance in special facilities. There have been several studies, but current solutions tend to address specific issues and cannot be easily extended, updated, and customized to meet the complex and evolving requirements of eldercare assistance. This study aims to bridge this gap, and this paper presents a context-management framework, called awareness for pervasive environments (APE), that provides easily customizable support for the development of ubiquitous eldercare services and applications.**

Index Terms: **Context management, integrated care environments, pervasive computing applications, ubiquitous eldercare.**

I. INTRODUCTION

The number of seniors is dramatically increasing, and this poses challenges for western societies [1]. Various studies predict that, in the medium to long term, the stress placed on welfare systems by the ever-growing number of seniors in need of professional care could undermine the economic sustainability of healthcare institutions [2].

This raises several challenging questions. How can we provide high-quality healthcare services to an ever increasing number of individuals? How can we achieve this goal without increasing healthcare budgets? Purely technical approaches to the above issues are not feasible. However, recent advances in wireless technologies, sensors, and actuators offer opportunities for the development of ubiquitous eldercare services for providing assistance to elderly people anywhere, anytime, and at a modest cost [3]–[5]. The main goal is to improve elders' independence, and thus, to reduce the need for professional assistance, with a positive impact from both social and economic perspectives.

Supporting the development of ubiquitous eldercare applications is challenging. The literature broadly acknowledges the need to collect, process, and distribute context information on the situation of both the elder and her caregivers, in order to determine the most appropriate assistive strategies, and, in order to act accordingly. In the following, we define context to be all the information that is needed by an eldercare service/application to determine the situation of an entity such as the elder or his/her caregivers [6]. Current proposals tend to consider different pieces of context information, such as, the elder's physical location, gestures, and heart rate, making it necessary to place instruments in the environment where the elder lives and/or making it necessary for the elder to wear assistive devices such as wireless bio-sensors [7]–[9], portable devices [10], or radio frequency identification (RFID) tags [3]–[5], [11].

Despite the broad research interest, current proposals have several limitations. Most studies present proof-of-concept prototypes that aim to demonstrate the effectiveness of innovative techniques for specific (and challenging) aspects of eldercare, e.g., vision-based emergency detection [12]. In addition, existing solutions do not recognize that providing effective eldercare support requires designers to implement strong service customization according to individual's pathologies and needs [4]. Furthermore, eldercare solutions should be easy to extend, so that a gradual evolution of the service according to the physical and psychological condition of the elder could be possible [10].

The above considerations suggest that the success of ubiquitous eldercare primarily depends on the availability of middleware-level solutions for integrating heterogeneous sources of context information (e.g., sensors, medical-record repositories), to provide customizable mechanisms for processing the collected data and for distributing them to interested entities. Therefore, this paper presents the awareness for pervasive environments (APE) framework, a middleware-level solution that is based on these ideas. APE has evolved from our previous research on AGAPE and ANGELAH [10], [11], and it provides a set of context-management facilities that promote and support the rapid development, customization, and deployment of eldercare services and applications.

The paper is organized as follows. Section II discusses related research and proposes relevant design guidelines. Section III presents the APE framework and details its model and architecture, and Section IV demonstrates its application to the support of an in-house safety application. Finally, we present concluding remarks.

II. CONTEXT MANAGEMENT IN ELDERCARE: SOLUTIONS AND DESIGN GUIDELINES

Several studies have demonstrated the centrality of context management in eldercare [3]–[5]. Context-management solutions gather context information from distributed context sources (e.g., sensors), aggregate the information to a higher abstraction level to facilitate application development, and distribute the aggregated information to interested parties, i.e., to context-aware services and applications [6], [13].

The current research in eldercare and related fields (e.g., smart environments) recognizes the opportunity to adopt centralized approaches for context management [13]. By assigning a centralized server the task of managing context informa-

Manuscript received September 30, 2010.

D. Bottazzi is with the Laboratori Guglielmo Marconi, Italy, email: dario.bottazzi@labs.it.

T. Taleb is with the NEC Laboratories Europe, Germany, email: tarik.taleb@neclab.eu.

M. Guizani is with the Department of Information Science, Kuwait University, Kuwait, email: mguizani@ieee.org.

A. S. Hafid is with the Network Research Lab, University of Montreal, Canada, email: ahafid@iro.umontreal.ca.

1229-2370/11/$10.00

tion, we can leverage the development of context-aware applications even for constrained and mobile devices. In addition, the availability of a context-management server reduces the network overhead. Different context-aware services and applications can gather previously aggregated data from the server without coordinating with distributed context sources, e.g., sensors.

However, despite the convergence toward centralized context-management approaches, the question of how to model and aggregate context information is still an open question [13]. Some authors suggest the representation of context information in terms of attribute-value couples [14]. Others suggest the adoption of logic-based context representations [15]. Recently, researchers have proposed context models based on semantic web technology [16]. In addition, context-management architectures tend to display a strong degree of heterogeneity, ranging from event-based solutions [6], to message-oriented solutions [16], to shared data spaces [17].

To the best of our knowledge, none of the current proposals can suitably address the unique requirements of eldercare scenarios [18]. The sensor platforms must be regularly extended according to the elder's changing needs, and so, the context aggregation and distribution logic must be updated. In addition, systems that assume predetermined context-representation models or the availability of specific sensor infrastructure are not suitable for eldercare. The emerging need stemming from eldercare domains to implement strong customizations on the sensing platforms and therefore on the context gathering, aggregation, and distribution logic, requires the development of innovative models and solutions able of promoting and facilitating system integration. Finally, current proposals are often implemented from scratch in an ad-hoc manner on top of the network layer. However, eldercare requires continuous service operations. We must therefore implement eldercare solutions on the basis of mature, widespread, and production-level software components that guarantee the necessary reliability. In addition, we need to base the middleware design on broadly accepted industrial standards that facilitate the future integration of eldercare services into broader healthcare-management systems.

We suggest a workflow-based context-management approach. Workflow technologies are commonly used to address complex integration issues. They can provide a suitable basis for the collection of context information from heterogeneous sources, e.g., sensor networks and context-data repositories. Workflow technologies also provide the required support for custom context aggregation and distribution logic, without the need for assumptions on context representation, languages, techniques, and technologies that are appropriate in context processing. Finally, workflow-based context-management solutions can be implemented on top of production-level service components that can tolerate high computational loads, and that guarantee full conformance with industrial standards, such as BPEL [19].

III. THE APE FRAMEWORK

APE is a context-management support tailored for eldercare scenarios. APE provides a set of tools and mechanisms for gathering and aggregating context information from heterogeneous and distributed sources, e.g., sensors and medical-record repositories. APE also permits the distribution of this information to interested entities, e.g., surveillance-center professionals and the elder's friends and family members. Finally, APE integrates a repository to store context information gathered over time, thus facilitating the early diagnosis of diseases that commonly affect elders [4].

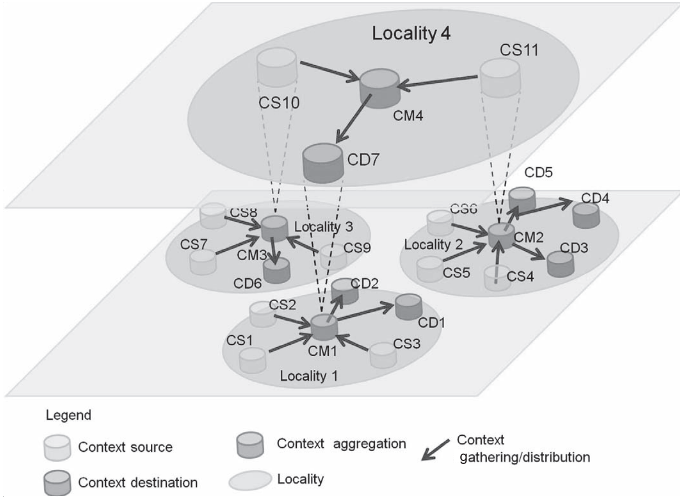


Fig. 1. The APE model.

A. The APE Context Management Model

APE identifies different roles in eldercare context management: The context source (CS), context destination (CD), and context manager (CM) management roles. CSs represent the sources of context information. CSs may be highly heterogeneous. Although sensors are perhaps the most common CS in eldercare domains, medical databases and eldercare applications can also provide relevant context information. CDs are entities that require context information to operate properly. CDs can be heterogeneous in nature and may range from the web services deployed at a medical center to the mobile phones of the elder's relatives to the actuators deployed in the elder's apartment. One entity can have both CS and CD context-management roles. Finally, CMs are central servers that gather the available context information from the deployed CSs, aggregate it, and then distribute it to the CDs.

Context gathering and distribution model. In APE, context management is based on locality, which lets us establish well-defined management boundaries. Each CM defines its own locality. Depending on the scenario, different mappings of the locality abstraction are possible. For example, a locality might define the set of CS/CD entities whose devices are currently connected to the same wireless cell or to the same mobile ad-hoc network (MANET). It is possible for different localities to overlap. However, CS and CD entities are statically associated with a single locality and provide context information to or obtain it from the locality CM. This is consistent with the requirements of eldercare domains, where most CSs are sensors deployed in apartments and the set of CSs is dedicated to a single elder in his/her locality. For integration flexibility, APE supports both push and pull communication between the CM and the CSs/CDs.

APE permits localities to be organized in a hierarchical fashion. It is thus possible for a CM to be placed within a locality managed by another CM entity and to play a CS and/or CD context-management role within that locality. For example, Fig. 1 depicts a situation where the CM entity of locality 3 plays a CS role in locality 4. Similarly, CM3 in locality 2 and CM1 in locality 1 play CS (i.e., CS10) and CD (i.e., CD7) roles, respectively, in locality 4.

Context aggregation model. In APE, context information is associated with meta-data, i.e., the meta-context which specifies the CS that originated the context data, the data type (e.g., the elder's location, temperature, heart rate), and optionally, the accuracy of the data and its validity time.

The context data type is the most relevant field in the meta-context. A CM associates each context data type with a context-management workflow specifying the sequence of aggregation activities that should be carried out. Different context data types are typically associated with different context-management workflows. It is possible to have transient phases where several context-management workflows are associated with the same data type. APE addresses this issue by associating the context data to be processed with the current workflow. The association between a context type and its workflow is maintained until the completion of the workflow execution.

A context-management workflow is defined as the set of activities needed to aggregate raw context data to a higher abstraction level. Context-management workflows are depicted as oriented graphs that establish the execution order of process activities and the synchronization policies required to accomplish a given piece of workflow. Each activity represents a single step in the workflow to be executed. Typically, activities concern the coordination of the workflow with the CSs or CDs. Each activity is characterized by several elements, namely, activity preconditions and actions to be executed. The preconditions of an activity are the logical conditions for APE to start the execution of the activity. An activity precondition typically represents the availability of a specific piece of context information that is needed to complete the execution of the aggregation process. Actions represent the set of operations to be executed on context information to increase the level of abstraction. The aggregation techniques depend on the application-specific context-representation model.

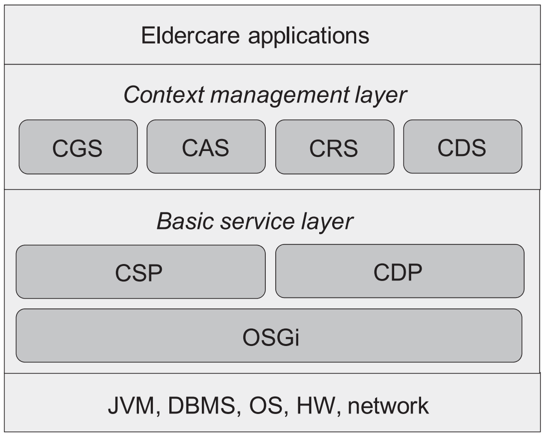
B. The APE Context Management Architecture

Fig. 2 depicts the APE middleware architecture implemented on top of the Java virtual machine. As Fig. 2 shows, the modular APE architecture is composed of two main layers, namely, basic service and context management layers.

Basic service layer. The basic-service layer integrates and manages all sources and destinations of context information, such as sensors and actuators deployed in the elder's home.

All basic-service layer support facilities are developed on top of the open service gateway initiative (OSGi) infrastructure [20]. OSGi provides a service-oriented, component-based support that significantly simplifies software life-cycle management, thus permitting frequent updates of the sensor/actuator infrastructure according to an elder's evolving needs.

In APE, each CS is statically associated with a context sensor



Legend
CGS: Context gathering service
CAS: Context aggregation service
CRS: Context repository service
CRS: Context distribution service
CSP: Context source proxy
GDP: Context destination proxy
OSGi: Open services gateway initiative

Fig. 2. The APE middleware architecture.

proxy (CSP). The CSP is placed between the CS and the APE context-management facilities, thus decoupling the context-gathering logic from the technical details of interaction with a CS, e.g., the protocol needed to query a sensor for its current reading and its technical specifications. Each CSP encapsulates properties that reflect not only the value of specific context data but also information such as the status of the CS (e.g., battery level of a sensor) and the meta-context data (e.g., type and accuracy of a sensor). CD entities are statically associated with proxies, namely, the context destination proxy (CDP), representing an entity to which context information is propagated. CDP maintains relevant information on CD entities, such as their names, addresses, and online availability. In addition, each CDP implements the support for interaction with CD entities, thus significantly simplifying the context distribution process. It is worth noticing that for both context gathering and context distribution, our approach permits CSPs/CDPs to optimize the interaction with CSs and CDs, e.g., by reducing the number of interactions among battery-powered entities.

Context management layer. The context-gathering service (CGS) receives data from the available CSP. Depending on the CS, it is possible for a CSP to supply context information either at regular times or when a specific event is detected. For example, an elder's home temperature can be sampled and communicated to the CGS at regular times, whereas a video-based fall detector may record a dangerous situation only when it actually occurs. The best suited approach to be adopted depends on the sensor characteristics and the application requirements.

Upon context data notification, the CGS forwards the context information to the context-repository service (CRS) and the context-aggregation service (CAS). The CGS maintains a list of available CSP instances, along with their IP addresses, unique identifiers, and properties. As Fig. 3 shows, it is necessary for a CSP to register with the CGS before taking part in context-gathering operations. All information about the CSP is supplied to the CGS during the CSP registration phase.

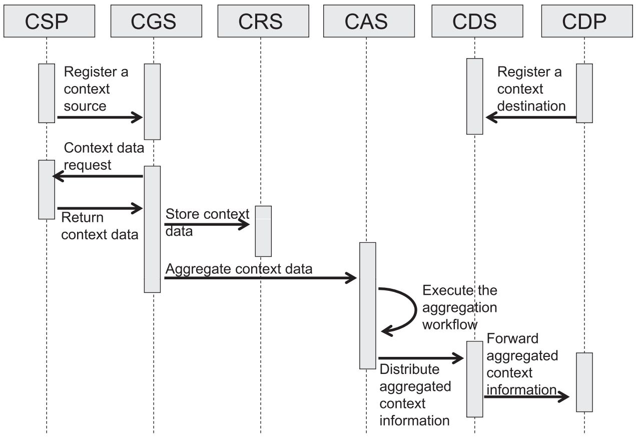


Fig. 3. The APE service interaction.

The CRS maintains the context information needed to build and update behavioral models that form the basis for detecting dangerous situations and facilitate the diagnosis of an elder's pathologies [4].

On the other hand, the CAS aggregates the context information obtained from different CSs, e.g., to detect possibly dangerous situations. The CAS is built on top of the JBoss JBpm workflow-management solution that provides production-level support to define, revise, or enact context-management workflows [19], [21]. Different activities are possible. For example, activities may access the context repository to obtain a piece of context information needed for data aggregation, or they may coordinate with the context-distribution service (CDS) to propagate context information to the interested parties. In addition, activities could be associated with scripts that implement the actual context-aggregation logic. Scripts can be implemented in different languages: Java, Python, TCL, or JavaScript. The business logic of a context-aggregation script depends on the available context information, its format and representation, and its accuracy. Context-aggregation logic also depends on application-specific considerations. It must supply the available information in the format and at the level of abstraction required for eldercare services.

Finally, the CDS distributes the aggregated context data to the interested entities. The CDS maintains the list of entities that should be included in the distribution, along with their names, their preferred communication protocols, and a reference to the associated CDP. Depending on application-specific considerations, it may be possible to use static lists of CD entities. However, in APE, it is also possible to identify an entity dynamically, e.g., by gathering the list of currently available co-located entities from a sensor proxy that coordinates with the AGAPE group management support [10].

IV. USE OF APE IN AN IN-HOUSE SAFETY APPLICATION

To demonstrate the suitability of APE for eldercare, we implemented an in-house safety application prototype that continuously monitors the situation of an elder in her living environment. Should a dangerous event occur, the application will detect the incident and communicate the emergency to both a metropolitan response center and the elder's family members via email and SMS.

Our scenario is the apartment of an elder affected by osteoporosis, so we focused on the development of a platform to detect accidental falls. We set up a sensor platform composed of a variety of devices: Video cameras and RFID readers. The cameras are ordinary digital video cameras operating at thirty frames per second; the cameras are ceiling-mounted, with vertically oriented optical axes and wide-angle lenses. Each camera has full visibility of one room. RFID readers are also carefully placed to provide complete coverage of the apartment. The system requires the elder to constantly wear an active RFID card.

We placed a PC that serves as the CM in the apartment. The PC runs Gentoo Linux, J2SE 1.6, JBoss jBPM, and the APE basic and context-management layers. In particular, the system was appropriately configured to gather context information from the sources in the apartment, i.e., in the locality. Each source was then statically associated with a particular CSP, within the CM, in charge of sampling and processing the data.

In particular, the CSP for an RFID can detect whether the elder is located near the associated reader, and the CSP for a camera can process the feedback from the associated camera and make decisions on whether a behavior anomaly occurred. [11].

In addition, various CDPs are deployed. In particular, SMS-enabled CDPs facilitate the distribution of emergency alerts to a predefined set of family members. Finally, a CDP instance that integrates the client-side components that communicate emergency situations to a metropolitan surveillance center is implemented. This CDP coordinates with a remote web server emulating the surveillance center, which is in charge of triggering response activities in the event of dangerous situations [11]. We emulated the surveillance center on a Linux-based server running Apache, Tomcat, and J2EE.

Context management. In our scenario, the CGS regularly obtains the current position of the elder. Every few seconds, the CGS sends a request to all CSPs associated with RFID readers to check the elder's location. We set the time between two consecutive requests to ten seconds. This allows us to track the elder's movements without imposing excessive overhead on the server involved in the context management or the networking environment. Then, the CGS forwards the information to the CAS. By analyzing the meta-context data, the CAS recognizes that the received context relates to location tracking and activates the corresponding context-management workflow.

The context-aggregation workflow for location tracking allows the system to determine the current position of the elder and to forward it to interested parties, i.e., the CDEs for the video sensors. Upon receipt, each CDE can determine whether to enable/disable video streaming from its sensor. Only the CDE associated with the room where the elder is currently located is in charge of detecting behavior anomalies, e.g., accidental falls. All the other CDEs disable their video sensors, reducing the computational load on the servers involved in the context management.

If an emergency is detected, the relevant CSP promptly forwards the emergency notification to the CGS. The CGS forwards the notification to the CAS that activates the emergency-response workflow. The activated workflow aggregates the information needed to arrange an emergency response. The CAS first attempts to understand what assistance is needed. An alert is then forwarded both to the surveillance center (via the SOAP protocol) and to the family members (via the SMS gateway).

As a final remark, the above scenario shows that workflow-based context management permits the simple implementation of a context-aware application for eldercare. The ability to easily revise context-management workflows makes it possible to extend the eldercare sensing platform and to integrate various technologies, to provide assistance according to an elder's needs.

A. *Prototype Evaluation*

The overhead of APE depends on the management functions involved: Context gathering, context aggregation, and context distribution. We will evaluate the APE network overhead. In addition, we will discuss the suitability of APE for monitoring a large number of elders. The following discussion assumes that each setting has the characteristics discussed above for the assistance of the elder affected by osteoporosis. Our scenario stresses the network overhead and requires different distributed entities to interact.

In the prototype, the network costs mainly arise from the regular gathering of context information from CSs and the distribution of context information to CDs. Our test-bed considers a small apartment composed of six rooms. For each room, we deployed two CSs: An RFID reader and a camera. The interaction between the server running the APE facilities and the CSs depends on the sensor characteristics. Every ten seconds, the server interacts with the RFID readers to find the room in which the elder is currently located. This information is used to enable video streaming only for the camera associated with the elder's current location. Every ten seconds, the server advises the cameras as to whether or not they should stream video. Clearly, the network cost in our prototype is mainly determined by the bandwidth consumed by the video streaming. The cost of gathering the location information and the overhead for the control of the video streaming are negligible. The context distribution is another factor that increases the network overhead. However, because the CD entities are provided with previously aggregated information of potential interest, no network-intensive coordination is needed between the server running the context-management facilities and its clients. In fact, both the SMS gateway and the surveillance center's web service require a single emergency notification to begin response operations. In addition, the emergency notification is sent only when prompt eldercare support is needed. When no elder assistance is required, the overall network overhead imposed by the context-information distribution is negligible.

We also considered the APE performance for the continuous monitoring of a large number of elders. In eldercare scenarios, it is necessary to integrate a number of sensors/actuators that should interoperate with different eldercare services. Our tests focus on the evaluation of the CAS, because it is the central component in the APE middleware architecture. The APE system architecture is designed to simplify the context-gathering and distribution logic, and CSPs and CDSs can be deployed over distributed servers.

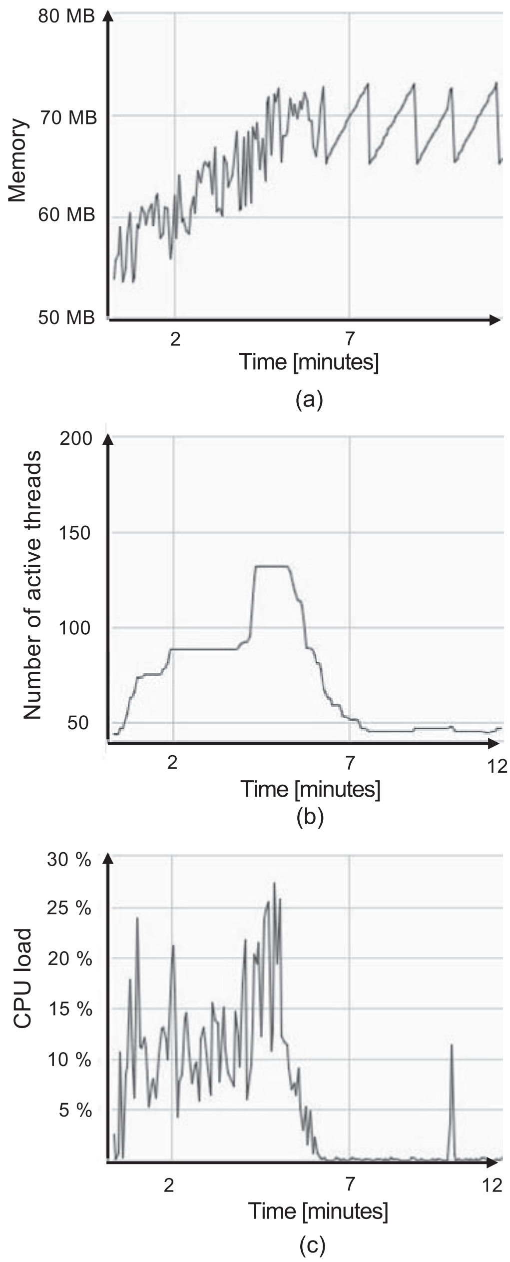


Fig. 4. (a) Memory use, (b) number of active threads, and (c) the CAS CPU load.

To exert stress on the CAS, we conducted several tests via emulation. In particular, we implemented an application to emulate an elder's mobility and possible emergency situations. The application uses the same interfaces as those described above, namely, RFID readers and video-based anomaly-detection interfaces. In our tests, we emulated the behavior of 200 elders living in their own apartments. As in our previous analysis, we assume that all apartments are composed of six rooms and that elders move from one room to another every 10 s. In addition, we also emulated the detection of emergency events. By considering 200 elders, we verified that the centralized CAS architecture can tolerate the computational load induced by context management. We considered 200 elders to intentionally exert stress on the APE middleware. In fact, we suggest that only a dozen elders be monitored at a time.

Fig. 4 shows the results. Our tests focus on the analysis of the memory use and CPU load of the CAS service. We also evaluate the number of active threads instantiated by the CAS over time. Our figures were obtained using the JConsole tool provided by the Java platform. As Fig. 4 shows, in CAS, we identify two distinct operating phases, namely, the bootstrap and service-provisioning phases. The bootstrap phase lasts about 5 min and permits the proper set-up and configuration of the CAS. This phase requires certain resources, and APE induces a significant CPU load. In addition, during this phase, about 130 threads are created and memory use grows linearly with time to about 70 MB. Following the bootstrap phase, the service-provisioning phase enables APE to aggregate and distribute the context information. During this phase, the CPU load imposed by the CAS significantly decreases and only 47 threads are active. The memory use varies between 65 MB and 72 MB. Fig. 4 depicts the response of the CAS in emergency situations. During emergencies, the CPU load reaches a value of 12%, but memory use and the number of live threads do not increase significantly. It is worth stressing that only a few hundred milliseconds are needed for the CAS to process emergency-notification events and to notify the CDs.

V. CONCLUDING REMARKS

The design, development, and deployment of eldercare services require novel context-management solutions. APE is a context-management framework for integrating heterogeneous sources of context information, for providing customizable mechanisms to process data, and for distributing the data to interested entities. We are currently testing the applicability of APE in a wide variety of eldercare scenarios ranging from home automation to behavior monitoring to the coordination of care. We have obtained encouraging results that could lead to further research for improving the APE prototype. In particular, we are currently investigating security concerns stemming from eldercare scenarios. Security is a particularly challenging aspect of eldercare. The development of eldercare applications requires designers to identify a tradeoff between the need to guarantee privacy and the need to disclose sensitive health-related information to caregivers. It seems necessary to adopt approaches to context-information disclosure that propagate information according to the specific situation and on the basis of the attributes and characteristics of the CD entities.

REFERENCES

[1] A. Lymperis, and A. Dittmar, "Advanced wearable health systems and applications, research, and development efforts in the European union," *IEEE Eng. Med. Biol. Mag.*, vol. 26, no. 3, pp. 29–33, May–June 2007.

[2] United Nations Population Division, World population prospects: The 2000 Revision. [Online]. Available: http://www.un.org/esa/population/publications/wpp2000

[3] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 1, pp. 1–12, Jan. 2010.

[4] P. Bellavista, D. Bottazzi, A. Corradi, and R. Montanari, *Challenges, Opportunities, and Solutions for Ubiquitous Eldercare*, CRC Press, Jan. 2006.

[5] A. Horgas and G. Abowd, *The Impact of Technology on Living Environments for Older Adults*, National Academy Press, 2004.

[6] A. K. Dey, "Understanding and using context," *Pers. Ubiquitous Comput. J.*, vol. 5, no. 1, pp. 4–7, 2001.

[7] J. Habetha, "The MyHeart project-fighting cardiovascular diseases by prevention and early diagnosis," in *Proc. 28th Ann. Int. IEEE EMBS Conf.*, 2006, pp. 6746–6749.

[8] A. Lymperis and R. Paradiso, "Smart and interactive textile enabling wearable personal applications: R&D state of the art and future challenges," in *Proc. 30th Ann. Int. IEEE EMBS Conf.*, 2008, pp. 5270–5273.

[9] M. Di Rienzo, F. Rizzo, G. Parati, G. Brambilla, M. Ferratini, and P. Castiglioni, "MagIC system: A new textile-based wearable device for biological signalmonitoring applicability in daily life and clinical setting," in *Proc. 27th Ann. Int. IEEE EMBS Conf.*, 2005, pp. 7167–7169.

[10] D. Bottazzi, A. Corradi, and R. Montanari, "Context-aware middleware solutions for anytime and anywhere emergency assistance to elderly people," *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 82–90, Apr. 2006.

[11] T. Taleb, D. Bottazzi, M. Guizani, and H. Nait-Charif, "ANGELAH: A framework for assisting elders at home," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 4, pp. 480–494, May 2009.

[12] S. J. McKenna and H. Nait-Charif, "Summarising contextual activity and detecting unusual inactivity in a supportive home environment," *Pattern Anal. Appl.*, vol. 7, no. 4, pp. 386–401, Dec. 2004.

[13] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *Int. J. Ad Hoc and Ubiquitous Comput.*, vol. 2, no. 4, pp. 263–277, June 2007.

[14] K. Henricksen, J. Indulska, and A. Rakotonirainy, "Modeling context information in pervasive computing systems," in *Proc. 1st Int. Conf. Pervasive Comput., LNCS 2414*, 2002, pp. 167–180.

[15] P. Fahy and S. Clarke, "CASS–middleware for mobile context-aware applications," in *Proc. Mobisys Workshop on Context Awareness*, 2004.

[16] D. Bottazzi, R. Montanari, and A. Toninelli, "A semantic context-aware middleware-level solution to support anytime and anywhere social networks," *IEEE Intell. Syst. Mag.*, vol. 22, no. 5, pp. 23–32, Sept–Oct. 2007.

[17] P. Korpipää, J. Mantyjarvi, J. Kela, H. Keranen, and E.-J. Malm, "Managing context information in mobile devices," *IEEE Pervasive Comput. Mag.*, vol. 2, no. 3, pp.42–51, July–Sept. 2003.

[18] A. Mihailidis, B. Carmichael, and J. Boger, "The use of computer vision in an intelligent environment to support aging-in-place, safety, and independence in the home," *IEEE Trans. Inf. Technol. Biomed.*, vol. 8, no. 3, pp. 238–247, Sept. 2004.

[19] Business Process Execution Language for Web Services Version 1.1. [Online]. http://www.ibm.com/developerworks/library/specification/ws-bpel

[20] C. Lee, D. Nordstedt, and S. Helal, "Enabling smart spaces with OSGi," *IEEE Pervasive Comput. Mag.*, vol. 2, no. 3pp. 89–94, Oct. 2003.

[21] JBoss jBPM Project. [Online]. Available: http://www.jboss.org/jbpm

Dario Bottazzi is Chief Researcher at Laboratori Guglielmo Marconi, and research fellow in the Department of Electronic and Information Systems at the University of Bologna. His research interests include middleware solutions for ubiquitous eldercare assistance, collaborative applications in mobile environments, social computing, and pervasive computing technologies. He received his Laurea degree in Computer Engineering, and his Ph.D. in Electronics, Computer, and Communications Engineering from the University of Bologna.

Tarik Taleb is currently working as Senior Researcher and 3GPP Standardization Engineer at NEC Europe Ltd, Heidelberg, Germany. Prior to his current position, he worked as assistant professor at the Graduate School of Information Sciences, Tohoku University, Japan. He received his B.E. degree in Information Engineering with distinction, M.Sc., and Ph.D. degrees in Information Sciences from GSIS, Tohoku Univ., in 2001, 2003, and 2005, respectively. His research interests lie in the field of architectural enhancements to 3GPP networks, mobile multimedia streaming, wireless networking, inter-vehicular communications, satellite and space communications, congestion control protocols, handoff and mobility management, and network security. He is/was on the editorial board of the IEEE Transactions on Vehicular Technology, IEEE Communications Surveys & Tutorials, IEEE Wireless Communications Magazine, and a number of Wiley journals. He is serving as secretary of the Wireless Communications Technical Committee. He also served as Secretary and then as Vice Chair of the Satellite and Space Communications Technical Committee of IEEE ComSoc (2006–2010).

Mohsen Guizani is currently a Professor and the Associate Dean of Academic Affairs at Kuwait University. He was the Chair of the CS Department at Western Michigan University from 2002 to 2006 and Chair of the CS Department at University of West Florida from 1999 to 2002. He also served in academic positions at the University of Missouri-Kansas City, University of Colorado-Denver and Syracuse University. He received his B.S. (with distinction) and M.S. degrees in Electrical Engineering; M.S. and Ph.D. degrees in Computer Engineering in 1984, 1986, 1987, and 1990, respectively, from Syracuse University, Syracuse, New York. His research interests include Computer Networks, Wireless Communications and Mobile Computing, and Optical Networking. He currently serves on the editorial boards of six technical Journals and the Founder and EIC of "Wireless Communications and Mobile Computing" Journal published by John Wiley and the Journal of Computer Systems, Networks and Communications, published by Hindawi. He is also the Founder and the Steering Committee Chair of the Annual International Conference of Wireless Communications and Mobile Computing (IWCMC). He is the author of seven books and more than 270 publications in refereed journals and conferences. He served as the Chair of IEEE ComSoc WTC and Chair of TAOS ComSoc Technical Committees. He was an IEEE Computer Society Distinguished Lecturer from 2003 to 2005. He is an IEEE Fellow and a Senior Member of ACM.

Abdelhakim Hafid is Professor at département d'Informatique et de recherche opérationnelle de l'Université de Montréal, where he founded the Network Research Lab (NRL) in 2005. Prior to joining U. of Montreal, he was with Telcordia Technologies (formerly Bell Communication Research), NJ, US, faculty at University of Western Ontario, research director at Advance Communication Engineering Center, Canada, researcher at Computer Research Institute of Montreal, Canada, and visiting scientist at GMD-Fokus, Germany. He has extensive academic and industrial research experience in the area of the management of next generation networks including wireless and optical networks, QoS management, distributed multimedia systems, and communication protocols.