

Towards an application framework for context-aware m-health applications

Tom Broens¹, Aart van Halteren¹, Marten van Sinderen¹, Katarzyna Wac^{1,2}

¹ Center for Telematics and Information Technology, ASNA group, University of Twente,
P.O. Box 217, 7500 AE Enschede, The Netherlands,
{k.e.wac, a.t.vanhalteren, t.h.f.broens}@utwente.nl,
<http://asna.ewi.utwente.nl>

² University of Geneva, CUI, OSG group,
24 rue General Dufour, 1211 Geneva 4, Switzerland,
Katarzyna.Wac@cui.unige.ch

Abstract— Several social issues, like aging, stimulate the use of mobile ICT applications for mobile healthcare (e.g., tele-monitoring). To support novel m-health applications, the consequences of developing these applications should be considered in the scope of a comprehensive architecture. This paper describes ongoing research that focuses on developing an application framework for supporting the development of context-aware m-health applications. It gives initial requirements for such a framework and it gives a first attempt for a functional decomposition. The use of the framework is illustrated by means of an epilepsy tele-monitoring scenario.

Index Terms— application framework, context awareness, tele-monitoring, m-health

I. INTRODUCTION

Recent improvements in healthcare have focused primarily on: efficient working practices for professionals, self management of patients, proactive disease prevention, and continuous care management. These improvements often imply the use of Information and Communication Technology (ICT) by both patients and healthcare professionals [1],[2],[3].

The following social-economical developments justify the need to further support healthcare professionals and patients with ICT [4],[5]:

- *Patient-centric healthcare*: For a long time, healthcare was government controlled. Now that patients become better informed, organized and educated, healthcare is shifting from offer- to demand-driven. Consider for instance the tendency that patients are treated longer in their own homes instead of at care centers [2]. This requires flexibility in the healthcare provisioning process.
- *Efficiency & cost saving*: Society is aging, which implies a rapidly growing number of elderly people that need healthcare. To manage the anticipated increase in

healthcare spending, cost saving and provisioning of more efficient healthcare is an important issue.

- *Cross-domain integration*: To provide more efficient and cost effective healthcare, it is recognized that healthcare must be organized as a value-chain that integrates multiple domains in healthcare (e.g., hospital – care center – general practitioner). This requires a form of transparency between domains and mechanisms to effectively integrate them.
- *Justification & Accountability*: Patients are becoming more demanding and want to know what happens to them and how their money is spent on healthcare. This requires a mechanism to effectively reflect on healthcare and communicate this to the patient.

Furthermore, the introduction of public wireless network infrastructures, such as the Universal Mobile Telecommunications System (UMTS), enables innovative mobile healthcare services. Promising examples of such m-health services are tele-monitoring and tele-treatment services. These services allow healthcare professionals to monitor the health condition of a patient and to act upon indications of an abnormal state at any time and at any place.

Several initiatives, such as Mobihealth [6], XMotion [7] and MyHeart [8] have investigated the feasibility and benefits of m-health services and applications. However, these initiatives do not provide a clear application framework that simplifies the development of m-health applications. The objective of this paper is to present an initial application framework and to show how this framework facilitates application development of m-health applications. Initially, we focus on the decomposition of the framework in architectural building blocks.

A second challenge that m-health application developers and users face, is the handling of immense volumes of health-related data. For instance, monitoring epileptic patients with currently available ambulant monitoring devices requires data rates up to 264kb/s. This means a production of raw vital sign data of approximately 2,9 GB per patient per day [9]. Other

relevant information, like patient records and insurance information should also be accessible. Raw processing of such high volumes of data becomes impossible and consequently inhibits high quality healthcare. We claim that by using contextual information for adapting application components this effect can be countered and user-tailored applications are enabled [10]. The application framework presented in this paper incorporates the use of contextual information.

The remainder of this paper is structured as follows: Section 2 presents a future m-health scenario to illustrate the problems involved in the provisioning of a typical m-health service and to derive general requirements for our application framework. Section 3 presents our initial application framework architecture and decomposes it into functional blocks. Section 4 describes the scenario from section 2 reflected in the framework architecture from section 3. Section 5 presents some conclusions and directions for future work.

II. TELE-MONITORING SCENARIO

We present a scenario that illustrates a patient-centric approach to healthcare:

Sophie is a young woman who suffers from epilepsy, a neurological disorder in which nerve cells of the brain occasionally release abnormal electrical pulses, leading to so-called seizures. Because the occurrence of a seizure is often sudden and unexpected, patients may feel limited in their daily life and insecure when they are alone [11]. Sophie has recently been provided with an ambulant 24-hours monitoring device, which is part of the Epilepsy Safety System (ESS). Thanks to the ESS, Sophie is no longer bound to traditional institutional care and can live a more normal life without losing her feeling of safety.

One day, while Sophie is driving her car to the shopping centre, the ESS detects the likely occurrence of a seizure. The following actions are then taken by the system:

- The ESS warns Sophie immediately, without mediation of a healthcare professional. Consequently, Sophie is able to stop the car at the side of the road before the seizure occurs. In this way preventing danger for her and possible bystanders.
- The ESS determines the availability of voluntary caregivers (e.g., family or friends of Sophie, aware of Sophie's condition and educated on how to provide first-aid in case of a seizure), in order to contact the most nearby person. Unfortunately, this time none of the volunteers is available and nearby, and the system takes no further action towards voluntary caregivers is taken by the system (such as sending a message to one of the volunteers, informing him about Sophie's possible seizure and her current location).
- The ESS sends an alarm to the care centre, where an available doctor starts looking at Sophie's full sensor data that the system forwards. The doctor concludes that the situation is serious. He therefore decides to send a nearby health team as soon as possible to the location that the system specifies. Even more so, since he is also informed

by the system that no voluntary caregiver is around to assist.

This scenario features the ESS (see figure 1), emphasizing the important role of ICT. The ESS can be implemented in many possible ways, but all implementations will involve a medical Body Area Network (BAN) [12],[13] to capture and to process raw patient data and possible contextual information. A possible structure of the ESS could comprise the following components (see figure 1):

- *Body Area network*: a local network that connects devices responsible for directly measuring, processing and possibly storing of patient data. Entities contained in the BAN are:
 - *Sensors*: devices that transform electro-physical signals to analog or digital signals. Possible sensor types are: an ECG sensor for measuring the patient's hearth activity; an activity sensor for measuring the patient's physical activity (motion); and a GPS to determine the location of the patient.
- *MBU (Mobile Base Unit)*: a device (e.g., PDA, Mobile phone or smart phone) that offers gateway functionality to connect the BAN with the m-health support system.
- *M-health support system*: connecting one or more BANs to m-health support services that can analyze and monitor the condition of patients. Entities in the M-health support system are for instance back-end systems and the caregiver application.

In this paper we focus on the tele-monitoring domain as a subset of the healthcare domain. This implies the occurrence of a *medical* BAN and the M-health support system. However, generally, the BAN and M-health support system can be replaced with arbitrary types of BANs and support systems for different applications domains (e.g. home entertainment, e-commerce).

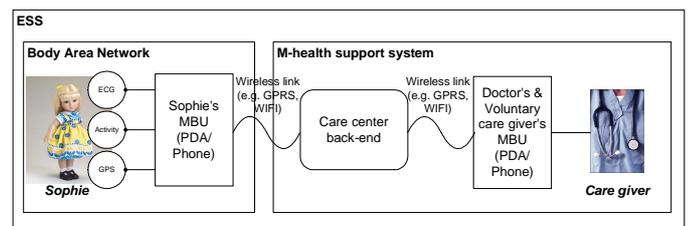


Fig. 1. ESS overview.

The scenario and the ESS structure are just examples of how patient-centric healthcare could be provided and what a supporting system could look like. There are many other possible healthcare applications for many other health conditions, and there are many variation points in both the presented scenario and the presented structure of the ESS. However, if we look at the application-level ICT support, we can also conclude that there are recurring problems to solve and issues to address:

- *Mobile nomadic service provisioning*: When offering patient-centric services, we see a tendency that the information producer-consumer roles are inverted. The

former consumer is increasingly becoming a producer of information. This paradigm is called nomadic mobile service provisioning [14], which entails that one or more mobile devices act as mobile service provider. Consider Sophie’s BAN, it acts as a service provider to the m-health support system. From another perspective, also her caregiver may act as a mobile nomadic service provider, offering healthcare services, like online consulting and responsiveness to alarms.

- *Context information processing and application adaptation:* When providing Sophie with the ESS, major amounts of information (vital data, treatment data, and location data) are produced that must be analyzed and acted upon by Sophie’s caregivers. Without a smart mechanism that tailors the available data into usable information, the m-health applications will become practically unusable. Contextual information, which is information about the environment of the user (e.g., location), can provide information to enable this tailoring. Furthermore, contextual information can be used to adapt the application. For instance, to contact different caregivers based on the availability and location of the caregivers and Sophie.

We argue that an application framework that provides generic support for solving these recurring problems will reduce the time and effort required to develop m-health applications. Such a framework may also foster the re-use of components between projects in the same application domain (e.g., addressing a specific disease) or even between different domains. From the previous, we can derive the following general requirements for the application framework:

- *Mobility support:* Sophie wants to live a normal life and does not want to be limited in her mobility. This implies the need for a mobile solution.
- *Context-awareness & Adaptation support:* Sophie, her doctor and her voluntary caregivers have different information needs. This implies a need for adaptation mechanism which tailors Sophie’s information to personalized units of information for different caregivers. We claim that context improves this adaptation. Therefore, the application framework should support adaptation mechanisms that can use context.
- *Execution environment:* Sophie’s BAN consists of sensors and mobile communication devices. Currently, these communication devices are limited in their capabilities (e.g., processing power, memory, OS capabilities). This implies certain additional requirements (e.g., performance and small footprint) for the application framework architecture.

III. APPLICATION FRAMEWORK ARCHITECTURE

Before presenting the initial functional decomposition of our application framework, we take the broader view of a complete ICT m-health system in which we position our application framework (see Figure 2). This broader overview

is based on the architecture proposed by the Freeband AWARENESS project [15].

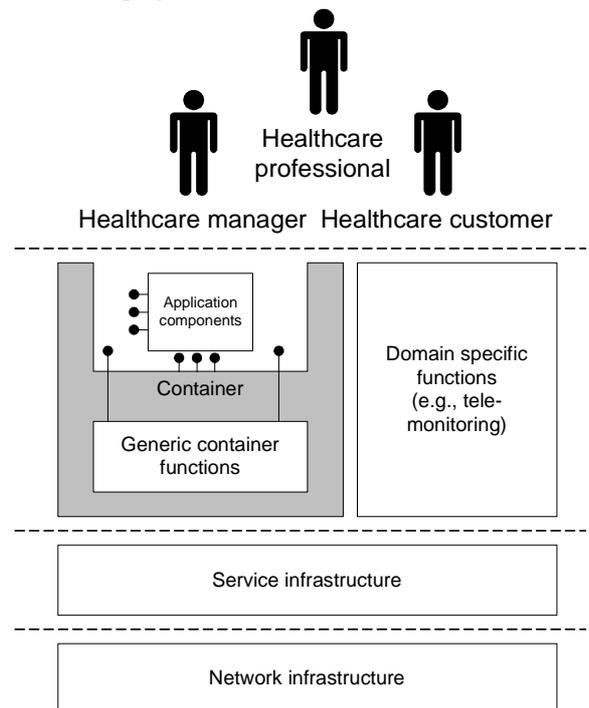


Fig. 2. Positioning of the application framework in the AWARENESS architecture.

AWARENESS considers a three-layered architecture. The bottom layer of the architecture is the network infrastructure layer, offering seamless mobile connectivity (e.g., GPRS, UMTS, WiFi). The middle layer is the service infrastructure layer that provides an execution environment for nomadic mobile services. It provides general functionality like service life-cycle management, service discovery and security mechanisms. The top layer is the application layer where we position our application framework. It offers an application container that provides an execution environment for application components and additionally provides access to generic container functions, domain specific functions and the service infrastructure in general. The generic container functions offer generic functions like context management, which apply to all application domains. Furthermore, there are domain specific functions. For instance in the case of the ESS: tele-monitoring services (e.g., BANmanagement). For a more detailed overview of the framework see Figure 3. All the functions in the application framework will be discussed in more detail later on in this section.

We assume that the network and service infrastructure supply the application framework with some common functionality:

- *Connectivity:* Applications should be able to communicate transparently with other applications without detailed knowledge of the underlying network technology. This requirement corresponds with the “connectivity” service access point (SAP (i.e. represented by ovals in figure 3)).

- *Context management*: Applications should be able to retrieve contextual information about the service/network infrastructure and other applications/services. Furthermore, it should be able to propagate its own contextual information to other context consumers via the service infrastructure. This requirements corresponds with the “context management” SAP.
- *Service management*: Application components should be able to register and de-register themselves to the service infrastructure such that they can be found by other services. This requirement corresponds with the “Service management” SAP.

The provided service access points are used by the generic container functions defined in our application framework. Generic container functions are common functions for context-aware applications which cannot be encapsulated by the service infrastructure. This is due to the fact that application specific knowledge cannot be dealt with by the service infrastructure (e.g., a detected seizure should trigger context-aware adaptation of the application. The service infrastructure cannot handle this application specific adaptation.). Furthermore, the function of the service infrastructure is different from that of the application framework. The application provides end-user oriented functionality while the service infrastructure provides supporting functionality (on service level, for instance accessing context sources [16]) for the applications. There must be a separation of concerns between the application and service infrastructure [17].

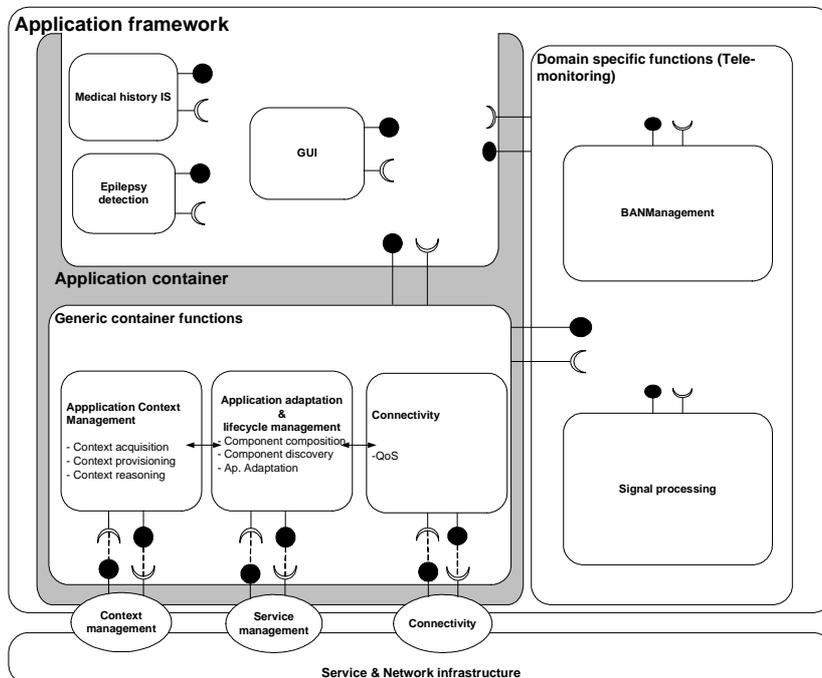


Fig. 3. Application framework

We identified the following components in the *generic container functions* in our framework:

- *Application Context Management*: acquires and provides application context from the application components to other components or the service infrastructure. Additionally, it may involve analysis of context on its quality (QoC) and inference of context from lower level context (e.g., $\text{highlevel.speed} = \text{infer}(\text{lowlevel.timeinterval}, \text{lowlevel.distance})$).
- *Application Adaptation & Lifecycle Management*: enables the registration and de-registration of application components and deals with their composition. It also provides adaptation mechanisms that can adapt the composition of application components based on contextual information (for further information see [16]). Additionally, it should provide mechanism to find (i.e. possibly with delegation to the service infrastructure) third party and infrastructure services. This enables dynamic binding between application components and third party/infrastructure services.
- *Connectivity*: provides transparent end-to-end connectivity by supporting uniform access to the service infrastructure. Additionally, it translates user QoS requirements into service infrastructure QoS requirements [18] with help of contextual information. Connectivity is provided based on the contextual information and service infrastructure QoS requirements. Connectivity affects choices of the network infrastructure supporting service delivery. Furthermore, the QoS offered by the chosen communication network infrastructure is provided as contextual information to the application adaptation functional block.

Domain specific functions are functions (i.e. facilitating toolboxes) that are specific to a certain application domain (in this case tele-monitoring) but common for different applications in this domain (e.g., epilepsy, cardiac-arrest, and diabetic monitoring applications or even personal sports management applications). They use the generic container functions and provide additional functions to the application components. In case of the tele-monitoring domain, they should provide, for instance, signal processing services and BANManagement services.

Signal processing functions are concerned with sensor data analysis and processing mechanisms. Examples are: high pass filters, signal computations and heartbeat detection and calculation. *BANManagement functions* entail a model of the available sensors and devices in a particular BAN. This model forms the abstraction from the physical entities and

provides access to these entities (see [17] for an initial design of this model).

These functionalities are provided to the application components using an application container (examples of application container models are CCM [19], J2EE [20]). A

container provides an execution environment and single point of access for *application components* (e.g., GUI in figure 3) to the service infrastructure and other application components. To simplify the development process of such application components, it provides API's for common functionality offered by the generic and domain specific services.

The ideas presented in this section are a first sketch of our application framework. Further work will focus on refining this framework, for instance, determining which (additional) components will be needed to support context-aware m-health applications.

IV. THE EPILEPSY MONITORING SCENARIO REVISITED

This section describes the scenario from section 2 reflected in the framework architecture from section 3. We present a possible application of the framework for the three major entities in the scenario (i.e., patient, care center, caregiver).

Figure 4 depicts a patient-side instance of the application framework (i.e. Sophie's application). We consider a situation in which the ESS is initialized (i.e. it is properly configured and all necessary services/components are registered).

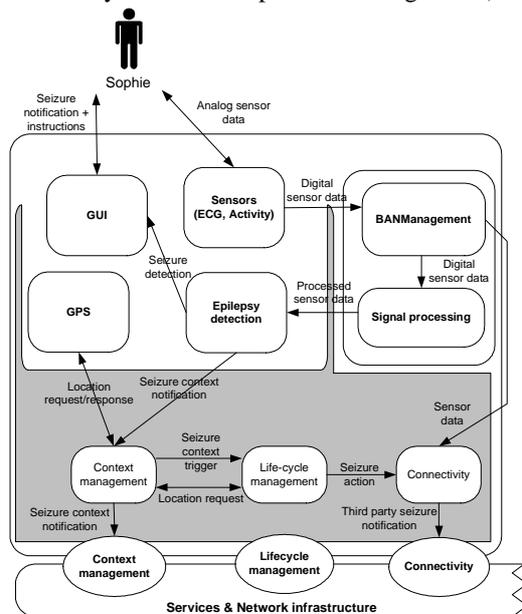


Fig. 4. Patient-side ESS

Sophie is monitored by sensors (Sensors component). The BANManagement component manages the composition and data flow of the sensors attached to Sophie. The raw data from the sensors is processed by the Signal processing component such that they can be used by the Epilepsy detection component [11]. This algorithm detects that Sophie's will get a major epileptic seizure. Sophie is advised to stop her car to prevent accidents and additionally she is informed that a caregiver will be notified (GUI component). The seizure detection is handled by the Context management component and propagated to the service infrastructure which reserves bandwidth needed to convey Sophie's vital signs (Connectivity component). Additionally it triggers adaptation mechanisms

(e.g., Event-Condition-Action rules [17],[21] see Figure 5).

```

if (major_epileptic_seizure) then
{notify_nearby(Doctor, Voluntary_aid) &
 notify(Sophie)}
else if (epileptic_seizure) then
{notify_nearby(Voluntary_aid) &
 notify(Sophie)}
else if (minor_epileptic_seizure) then
{notify(Sophie)}

```

Fig. 5. ECA rules

In case of a major and a normal seizure, this results in a service call to the care center system which can dispatch the correct caregiver. Figure 6 focuses on the Care center side and Figure 7 on the Caregiver side of the ESS.

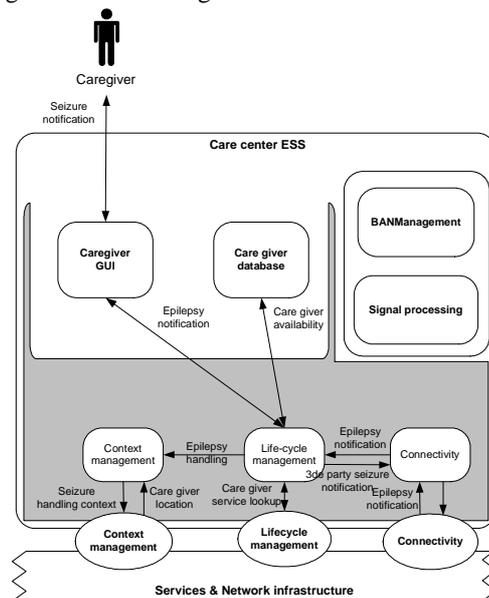


Fig.6. Care center -side ESS

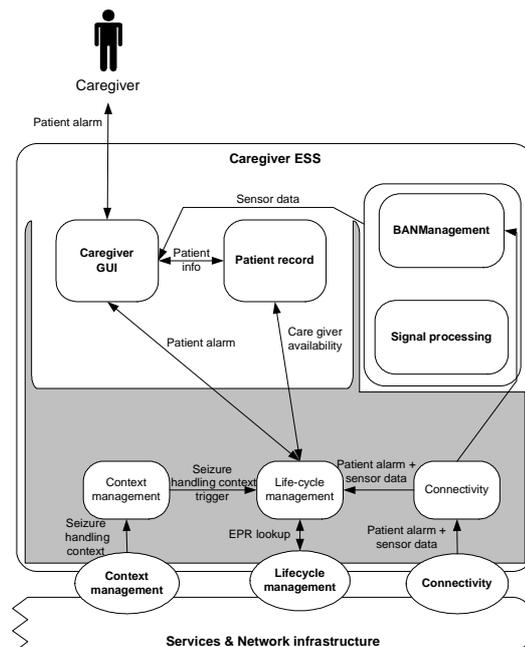


Fig.7. Caregiver -side ESS

The care center system receives the seizure notification (Connectivity component) and checks which doctors/voluntary caregivers are available to help Sophie (Life-cycle management component). It then requests the location of the available caregivers and matches it to the location of Sophie (Context management component). A service call is made to the available caregiver that is nearest to Sophie (Connectivity component). The caregiver gets a seizure notification (Care giver GUI component) and based on his status (doctor or voluntary caregiver) he gets respectively the full sensor output or a pre-processed set of signals (Signal processing and BANmanagement component).

V. CONCLUSIONS AND FUTURE RESEARCH

This paper discusses ongoing research on the development of an application framework for context-aware m-health applications. We see a growing attention for ICT in healthcare. However, this results in an enormous increase of production of electronic patient data and number of available services. Without a context-aware execution environment, development and usability of mobile m-health applications becomes infeasible. We claim that a context-aware application framework provides such an application environment.

There are several initiatives for creating m-health applications. However, a framework that supports application developers in creating context-aware m-health applications is missing. In this paper we present requirements for such a framework and an initial decomposition of our framework architecture in functional blocks. The framework consists of an application container providing an execution environment for application components. This container makes use of domain specific services (e.g., tele-monitoring services) and generic container functions (e.g., context management).

However, several aspects of this architecture require further research:

- *Identification of functional blocks*: The proposed framework is an initial attempt to identify functional blocks relevant for context-aware m-health applications. However, the identified blocks in this decomposition might not be orthogonal and/or certain blocks may be missing.
- *Local vs. remote processing*: Among the physical execution entities will be mobile devices (e.g., PDA) which have limited resources available. It is therefore possible that some of the functionality identified in the architecture will not be able to run on a single mobile device.
- *Context & adaptation mechanisms*: Context plays an important role in our framework. Therefore, context management and adaptation based on this context is very important. It needs to be investigated how context can be efficiently acquired and how higher level context can be inferred. Another issue is how applications can exchange their context with other applications and with the

service/communication network infrastructure. Finally, it needs to be investigated how to adapt application components and their composition to provide user-tailored services.

- *Component model*: The component container is the execution environment of the application components. The details of the API's to be offered by the container, as well as the embedding of the application components with current component containers (e.g., CCM, J2EE) is still unclear. Furthermore, most current container technologies are not suitable for mobile devices which indicate another research issue.

We intent to combine our further research on these issues with the development of proof-of-concept prototypes and the definition and execution of case studies together with stakeholders from the mobile healthcare domain.

ACKNOWLEDGMENT

We would like to thank Dick Quartel for his contribution to this paper. This work is part of the Freeband AWARENESS project. Freeband is sponsored by the Dutch government under contract BSIK 03025. (<http://awareness.freeband.nl>).

This work is partially supported by the E-NEXT Network of Excellence under contract FP6 IST 506869."

REFERENCES

- [1] Tachakra, S., Wang, X.H., Istepanian, S.H., Song, Y.H., "Mobile e-health: The Unwired Evolution of Telemedicine", Telemedicine Journal and E-health, volume 9, number 3, pp 247-257, 2003.
- [2] Ross, P., "Managing Care Through the Air", IEEE Spectrum, pp 26-31 December 2004.
- [3] Air2Web, "The Possibilities of Wireless Healthcare", Whitepaper, <http://www.air2web.com>, 2002
- [4] The Telemedicine Alliance, "Telemedicine 2010: Visions for a personal Medical Network", ISBN 92-9092-799-2, 2004.
- [5] Dean, K. (ed.), "Connected Health: essays from health innovators", Cisco Thought Leaders series, ISBN 0-9546445-0-6, 2004.
- [6] Mobihealth project webpage, <http://www.mobihealth.org/>.
- [7] X-motion project webpage, <http://www.ist-ipv6.org/modules.php?op=modload&name=News&file=article&sid=39&mode=thread&order=0&thold=0>.
- [8] MyHearth project webpage, <http://www.extra.research.philips.com/euprojects/myheart/>.
- [9] Broens, T., "Bandwidth optimization of the TMSI fiber protocol", Internship report, University of Twente, 2003, http://wwwhome.cs.utwente.nl/~broens/dloadcontent/Broens_TMSI_internship_report_v1.pdf.
- [10] Bardram, J., "Applications of Context-Aware Computing in Hospital work", Proc. of ACM Symposium on Applied Computing (SAC), pp1547-1579, Nicosia, Cyprus, 2004
- [11] Huis in 't Veld, R., Hermens, H., Vollenbroek-Hutten, M., "Tools and methods for reliable measurement of sensory input information: State of the art and first selection of parameters", Freeband AWARENESS D4.1, December 2004, https://doc.freeband.nl/ds.cgi/ds.py/Get/File-49231/Deliverable_4_1.pdf.
- [12] Halteren, A., Bults, R., Widya, I., Jones, V., Konstantas, D., "Mobihealth-Wireless Body Area Networks for Healthcare", Proc. New generation of wearable systems for e-health 2003, pp121-126, 2003
- [13] Dokovski, N., Halteren, A. van, Widya, I., "BANip: enabling remote healthcare monitoring with Body Area Networks", Proceedings of FIDJI'03, 2003.

- [14] Mascolo, C., Carpa, L., Emmerich, W., "Middleware for Communications" , "Principles of Mobile Computing Middleware", John Wiley & Sons, pp 261-280, ISBN 0-470-86206-8, 2004.
- [15] Wegdam, M., Sinderen, M. van, Brok, J., Benz, H., Kranenburg, H. van, Hermens, H., Lagerberg, K., "The AWARENESS project: Towards Context-Aware Mobile Networks and Services", Freeband AWARENESS D0.1, December 2004, https://doc.freeband.nl/dscgi/ds.py/Get/File-49232/D0.1v1.0_-_AWARENESS_Overview_Whitepape_-_2004_version.pdf.
- [16] Kranenburg H. van, Eertink, H., "Processing Heterogeneous Context Information", Proceedings of the Next Generation IP-based Service Platforms for Future Mobile Systems workshop at SAINT'05, Trento, Italy, 2005.
- [17] Broens, T.H.F., Dokovski, N., Halteren, A. van, Peet, B., Salden, A., Widya, I., "Context-aware application components", Freeband AWARENESS D4.2, December 2004, https://doc.freeband.nl/dscgi/ds.py/Get/File-49222/D4.2_-_Context-aware_application_components.pdf.
- [18] Wac, K., Halteren, A. van, Broens, T., "Context-aware QoS provisioning in m-health service platform", Proceedings of 11th Open European Summer School (EUNICE 2005), 6-8 July 2005, Colmenarejo, Spain.
- [19] OMG, "Corba Component Model", version 3.0, 2002, <http://www.omg.org/technology/documents/formal/components.htm>.
- [20] Sun microsystems, "J2EE Specification", version 1.4, 2003, http://java.sun.com/j2ee/j2ee-1_4-fr-spec.pdf.
- [21] Dockhorn Costa, P., Pires, L., Sinderen, M van., "Architectural patterns for context-aware services platforms", Proc. of the 7th International Conference on Enterprise Information Systems (IWUC 2005), 24-25 May 2005, Miami, USA, 2005