

A FRAMEWORK FOR SMART DISTRIBUTION OF BIO-SIGNAL PROCESSING UNITS IN M-HEALTH

Hailiang Mei¹, Ing Widya¹, Tom Broens¹, Pravin Pawar¹, Aart van Halteren²,
Boris Shishkov¹, Marten van Sinderen¹

¹*Department of Computer Science, University of Twente, Enschede, The Netherlands
{meih; widya; t.h.f.broens; p.pawar; b.b.shishkov; m.j.vansinderen}@ewi.utwente.nl*

²*Philips Research, Eindhoven, The Netherlands
aart.van.halteren@philips.com*

Keywords: Bio-Signal Processing Unit; Context; Distribution Framework; Session Transition; m-Health.

Abstract: This paper introduces the Bio-Signal Processing Unit (BSPU) as a functional component that hosts (part of) the bio-signal information processing algorithms that are needed for an m-health application. With our approach, the BSPUs can be dynamically assigned to available nodes between the bio-signal source and the application to optimize the use of computation and communication resources. The main contributions of this paper are: (1) it presents the supporting architecture (e.g. components and interfaces) and the mechanism (sequence of interactions) for BSPU distribution; (2) it proposes a coordination mechanism to ensure the correctness of the BSPU distribution; (3) it elaborates the design of smooth transition during BSPU distribution in order to minimize the disturbance to the m-health streaming application.

1 INTRODUCTION

Information and Communication Technologies (ICT) provide increasingly powerful tools improving the delivery of services, including health-care services; this concerns efficiency improvements, error reduction, and cost savings (Shishkov & Van Sinderen, 2007). A particular type of such services are the mobile health-care services relying on improved device capabilities, miniaturization, and connectivity (AWARENESS, 2007). Thus, mobile applications become feasible in daily health-care practice. All this gives way to the emerging discipline of mobile health-care or m-health (for short) that closely relates to the notion of telemedicine (Pattichis et al., 2002).

As a kind of telemedicine, m-health is concerned with the provision of health-care services and sharing of medical knowledge over distance, using telecommunication means. A common example of telemedicine is the use of assistive technology for aiding the elderly (Miskelly, 2001).

This work addresses especially telemedicine systems where bio-signals of patients are collected and sent to care professionals for diagnosis. Optionally, these systems could also support care professionals in their returning feedback that can be used for the patients' treatment.

Currently we observe a trend towards continuous (24/7) monitoring/treatment of patients which leads to challenges that have to be taken into account when developing telemedicine/m-health applications:

- ? Overcoming resource scarceness on the devices used for 24/7 monitoring and treatment (e.g., avoiding insufficient battery, storage, CPU);
- ? Ensuring usability of the devices for the patients and care professionals (e.g., avoiding feedback in unwanted situations).

A consideration of the notion of context-awareness appears to be promising in tackling these challenges (Shishkov & Van Sinderen, 2007). Context-aware applications adapt to changes in their environment, which could concern, for instance, the

location of a patient or his activity. An example of a possible use of context-aware applications is in a situation that concerns unwanted feedback: an m-health context-aware application can take activity of a user (patient) (e.g., determined on basis of the user's calendar) and decide whether feedback that is sent by the care professional is wanted or should be delayed until a more convenient time for the user.

Aiming at overcoming the first of the above mentioned challenges and inspired by the notion of context-awareness, we outline in this paper a (proposed) context-aware infrastructure. It concerns a stream of bio-signals flowing from the location of the patient to the location where the health-care service takes place, as envisioned from the perspective of an m-health system that supports the (remote) health-care professionals (Figure 1). With regard to such a system, in order to enable optimal use of the computational, storage, and communication resources (of the system or the supporting infrastructure), we might distribute parts of the processing task along the bio-signal path. For example, depending on these contexts, we can bring some parts of the processing task to the data source, instead of only bringing (raw) data to a data processing cruncher at the final destination of the bio-signal path. Hence, the current paper introduces the Bio-Signal Processing Unit (BSPU) as a functional component that hosts part of bio-signal information processing algorithms and further proposes the architecture and the mechanisms to enable the distribution of such a unit.

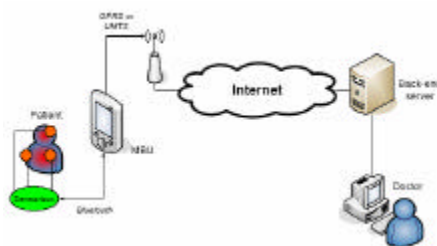


Figure 1: M-health system, supporting the (remote) healthcare professional to take appropriate treatment actions for the patient upon processed bio-signals

The paper is organized as follows: Section 2 elicits the requirements towards the BSPU distribution system and related infrastructure. Section 3 briefly outlines the proposed architectural components concerning the BSPU distribution framework. On basis of this, Section 4 elaborates the framework, in the directions of coordination,

distribution, and session transition. These results are matched to an analysis of related work, in Section 5. Finally, Section 6 contains the conclusions.

2 REQUIREMENTS ELICITATION

Capturing systems requirements can be based on the analysis of scenarios and the identification of use cases from these scenarios (Arlow & Neustadt, 2002; Shishkov & Dietz, 2005). In the current section, we derive in this way the requirements concerning the software framework and mechanisms that support the smart distribution of BSPUs. More extended discussion on scenarios and requirements can be found in (Mei et al., 2005).

2.1 Epilepsy Scenario

Epilepsy is a serious chronic neurological condition that is characterized by recurrent unprovoked seizures. According to WHO (World Health Organization, 2006), there are about 50 million people worldwide who suffer from this disease. Because of the random and unpredictable nature of seizures, epilepsy patients are experiencing less life quality. In order to provide a better medical support for epilepsy patients, the possibility of applying M-health technologies into the daily monitoring of epilepsy has been studied in the IST MobiHealth (MobiHealth, 2003) and AWARENESS (AWARENESS, 2007; Van Sinderen et al., 2005) projects. The targeted scenario is described below.

Mr. Janssen is a 46-year-old man suffering from epilepsy. Recently, he has been wearing a 24-hour seizure-monitoring system, consisting of a sensor-box for collecting bio-signals and a Mobile Base Unit – MBU, for local data processing. A related seizure monitoring program can be implemented with six BSPUs (Veld et al., 2005) as shown in Fig. 2. Measuring on heart rate variability and physical activity, this program can predict future seizures and is able to automatically contact relatives or health-care professionals. The aim of using this system is to provide Mr. Janssen with higher levels of safety and independence in order to function more normally in society despite his disease.

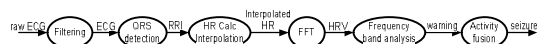


Figure 2: Epileptic seizure detection program consisting of six BSPUs

When Mr. Janssen is at home, a broadband network is available to transfer the raw ECG and activity information about Mr. Janssen to the remote monitoring centre, e.g. the back-end server in Figure 1. In this case, all six BSPUs are deployed onto the back-end server and the doctor can be warned if a seizure is likely to occur.

Despite his epilepsy, Mr. Janssen is a fanatical runner. One afternoon, he runs his usual circuit in the forest. Since there is no broadband network available in the forest, the bio-signals of Mr. Janssen cannot be sent due to insufficient network bandwidth. Therefore, all six BSPUs are loaded into his MBU and his bio-signals are processed locally. During his running, a possible imminent epileptic seizure is detected and, very likely, is going to occur within a few minutes. Mr. Janssen is immediately warned, and stops running. At the same time, an alarm and the position of Mr. Janssen are sent to the monitoring centre via a narrow band connection, e.g. GSM. An ambulance with a health team is sent to Mr. Janssen, and while driving, the team is constantly informed about the position of Mr. Janssen. When the team arrives, they find Mr. Janssen in a *status epilepticus*, meaning that seizures keep following each other while Mr. Janssen is unconscious. Medical action is necessary and Mr. Janssen is transported to the nearest hospital by ambulance.

2.2 Identification of Key Components and Processes

Based on the scenario described above and related use cases, whose derivation and specification is left beyond the scope of this paper, we identify a list of key components and processes that should be supported by the framework. The identified components and related processes are the following:

1. A controlling entity detects context changes that could justify a re-distribution of BSPUs involved in a particular m-health application. Based on such context events, the controlling entity must decide whether or not to re-distribute the BSPUs involved in a particular m-health application;
2. The controlling entity identifies the devices that can support the m-health application;
3. BSPUs can be uploaded to the devices, if necessary, BSPUs are installed and linked with peer BSPUs on other devices;
4. An m-health application session starts by starting the newly distributed BSPUs;

5. The parameters of BSPUs can be remotely controlled, if necessary;

6. Security-related concerns, such as authentication and encryption, must be appropriately addressed during the above-mentioned processes.

3 FRAMEWORK OVERVIEW

Being inspired by the Component-Based Development paradigm (Shishkov & Dietz, 2005), we envision software components as units whose execution location may be distributed. This applies in the current case for the BSPU that is considered as a software component. The choice for executing a BSPU on different devices depends on resource constraints like processing power and available bandwidth. Based on the derived requirements, a high-level architecture of the BSPU distribution framework is identified consisting of a Coordinator, a BSPU Repository and a set of Facilitators (Figure 3).

The Coordinator coordinates the Facilitators. It can monitor its environment for context changes or be notified by other entities, called context monitors, about context events. Some context monitors may be co-located with a Facilitator and monitor the Facilitator node's context, e.g. local resources and user preferences. Based on these observations, the Coordinator can make decisions to (re)distribute a certain BSPU to a certain device in the network. At the conceptual level, we treat the Coordinator as a single entity. However, in a deployment, the Coordinator is not necessarily centralized, but can be divided in parts which are distributed across multiple nodes. In such a case, the Coordinator parts should work together to consistently coordinate the Facilitators. Furthermore, there is no restriction of having only one coordinator: multiple coordinators might exist in the network with different start levels, to avoid the risk of single point failure.

Each device has a Facilitator to manage and facilitate the resident BSPUs. A Facilitator is the representative of its device in the distribution framework: it can report the presence/absence of the device, it can receive the control commands on BSPU life-cycle management from the Coordinator and it can report errors on distribution, back to the Coordinator. A Facilitator can also receive as input bio-signal data packets, feed the decoded bio-signal data into one of its managed BSPUs, encode the output bio-signal data from a BSPU into packets and send these packets to the next Facilitator on the path.

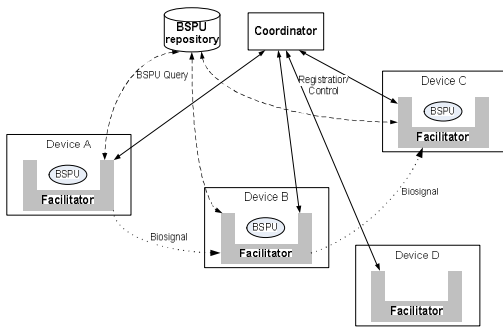


Figure 3: Architectural components of BSPU distribution framework

The BSPU Repository stores BSPU implementations that can be provided on request. In case that the BSPUs are preloaded onto each device, the BSPU repository becomes a distributed component, which eliminates the overhead of downloading new BSPU at runtime.

4 FUNCTIONALITIES AND INTERACTIONS

In aiming at elaborating the framework (introduced in Section 3), we will thoroughly discuss in this section not only the BSPU distribution mechanism but also related session transition issues.

4.1 BSPU Distribution Coordination

The behaviour of the Coordinator is explained with the state chart shown in Figure 4. The initial state is “waiting”, in which the Coordinator is capable of distinguishing the registration request and (re)distribution request. Once a registration request is received (from a Facilitator), the Coordinator will update a registry table containing the status of all the available Facilitators: if the received request is a presence message, the Coordinator will add a new Facilitator to its registry; if the received request is an absence message or if no keep-alive message is received from a particular Facilitator, the Coordinator will delete the corresponding Facilitator from the registry.

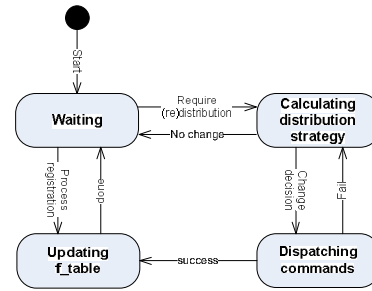


Figure 4: The state chart of Coordinator

The Coordinator will move to the “calculating” state, if a new BSPU distribution calculation is required. If the calculated BSPU distribution differs from the current BSPU distribution, the Coordinator will move to the “dispatching commands” state to notify certain Facilitators to reconfigure through the control interface. The normal BSPU distribution mechanism can be divided into three phases in a “start-run-stop” manner: *bootstrap*, *dispatching*, and *termination*.

- ? **Bootstrap.** Once a Facilitator is on line, it needs to report its presence to the Coordinator by requesting a new BSPU distribution service, which is necessary for the Coordinator to be able to locate the Facilitator later. If this request is granted, the Coordinator needs to return a confirmation message including a keep-alive timer interval.
- ? **Dispatching.** In this phase, the Facilitator sends regular keep-alive messages to the Coordinator based on the agreed timer interval and is ready to act upon the received unsolicited notification messages from the Coordinator, as it is shown in Figure 5. The asynchronous notification message may either trigger the Facilitator to install or remove a BSPU. The “request to install” message contains the ID of the required BSPU and a configuration file (BSPU descriptor) indicating how this required BSPU can be connected with the peer BSPUs on other Facilitators. The installation operation is defined as an atomic action. The “request to remove” message only contains the ID of the BSPU that must be removed. After receiving this message, the Facilitator should deactivate this BSPU and may as well physically remove it from the local disk, if BSPU caching is disabled.

- ? **Termination.** The Facilitator sends a signal to terminate the distribution service when its host application or node is going to be switched off or move out of the network coverage. This signal is used to update the Coordinator's knowledge on which Facilitators are available for making further distribution decisions.

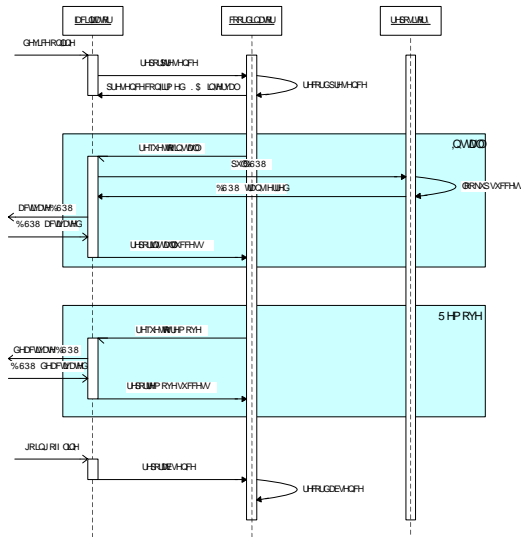


Figure 5: Message sequence diagram indicating how Facilitator, Coordinator and (BSPU) Repository interact with each other for the BSPU distribution

4.2 Session Transition

One of the challenges of the Facilitator design is the smooth replacement of a current BSPU by a new one during the BSPU activation operation. We discuss our solution based on a simple transition example involving different BSPUs.

As shown in Figure 6, an original activity and the target activity are described as “pre-distribution” stage and “post-distribution” stage respectively, where “BSPU 3>>2>>1” and “BSPU 3’>>2’>>1’” form two coherent bio-signal processing activities. To make the transition from “pre-distribution” to “post-distribution”, we introduce an intermediate stage, referred to as the “transition” stage. During the transition stage, the Facilitator runs both the old BSPU and the new one and it exchanges bio-signal data packets with other Facilitators. A bio-signal data packet is a “unit-of-interpretation” of bio-signal

data and it follows the defined bio-signal data model (Mei et al., 2006). Once a facilitator receives a new vital sign packet, it can tell who pre-processed this data by checking the packet’s header part, e.g. BSPU2 is the previous processor of packet “~2” and packet “~3” is from BSPU3’. Therefore the Facilitator, with the knowledge of the BSPU descriptor, is capable of directing the vital sign data flow to the appropriate BSPU during the transition stage. Once no more received data packets are processed by the “pre-distribution” stage BSPU, the Facilitator can disable the BSPU and release its resource. Using keep-alive signals between Facilitators is one solution to tell when to disable the outdated BSPU. This approach is still valid for handling more complicate transition scenarios, e.g. introducing a new device into the “post-distribution” activity

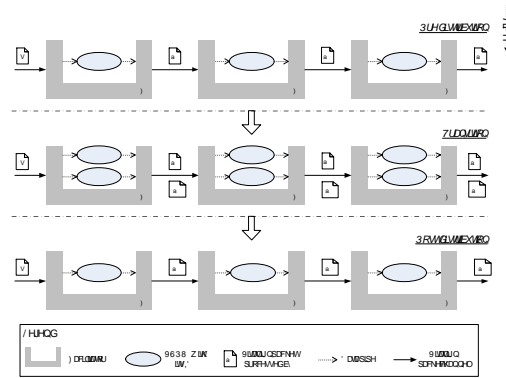


Figure 6: Transition between activities supported by Facilitator

5 RELATED WORK

Adapting transmitted (multimedia) content is a well-recognized technique to cope with the dynamism of network conditions and client characteristics (Wai Yip & Lau, 2002; Han et al., 1998; Fox et al., 1998). Content adaptation can be lossy or lossless with respect to the transmitted content. Lossy adaptation techniques may apply dropping some unimportant content, lossy data compression, downgrading the data resolution or conversion to a different data format. Lossless adaptation is mainly supported by lossless data compression techniques, e.g. Huffman coding. In order to reduce the computation load for client devices, content adaptation may be pushed into the network (Yarvis et al., 2000). For example, adaptation agents can be either deployed centrally on

a single intermediate proxy server in the network (Wai Yip & Lau, 2002) or distributed among multiple nodes along the content stream (Yarvis et al., 1999). A conceptual framework to capture the essence of the adaptation mechanism was summarized in (Badrinath et al., 2000). The goal of content adaptation is to make sure that clients can receive an as complete as possible original content and in a suitable format. Our approach is different in the sense that we can benefit from the special property of m-health applications: the ultimate goal in bio-signal processing is to have the bio-signal processing result rather than the original data. Therefore, the processing tasks (BSPUs) can be deployed closer to the data source (the patient) to reduce to the amount of transferred data. In this way, the m-health application response time and other system performance aspects can be improved.

In this paper we address one of the key challenges in building dynamic reconfiguration systems for streaming applications, namely to achieve seamless session handoff. Generally, session handoff consists of three steps (Bagrodia et al., 2003): (1) identifying a subset of the application state that concisely captures the semantics of a partially completed computation; (2) transferring that state subset to a different client device; and (3) resuming work on that target platform – all with very low latency while satisfying appropriate security, scalability and heterogeneity constraints. Executing session handoff among heterogeneous devices is very challenging, because the content of a new session has to fit with the new target device, therefore, content adaptation is required to be included. In our approach, the transfer of application state information is not necessary. This is because the state information is already captured in the bio-signal data packet. Further, through the introduction of Facilitator components, we hide the device heterogeneity and no content adaptation is needed.

6 CONCLUSIONS

In this paper, we have proposed improvements that concern the design of m-health (tele-monitoring) distributed ICT applications. In particular, we have introduced the BSPU (Bio-Signal Processing Unit) as a software component having bio-signal processing capabilities that concern bio-signal information processing algorithms (hosted on the component).

Based on the consideration of a real-life health-care-related scenario and the elicitation of

requirements (in terms of desirable key components and processes), we propose a high-level architecture for the BSPU distribution. We have elaborated some aspects concerning this architecture: firstly, we have addressed in detail the BSPU distribution mechanism; secondly, we have thoroughly considered the transition between activities in case of re-distribution, i.e., when a BSPU is replaced by another BSPU.

Hence, the paper's contribution is considered to be three-fold:

1. A high-level architecture for BSPU distribution is proposed;
2. A related coordination mechanism is presented, to ensure the correctness of the BSPU distribution;
3. A solution for the smooth transition between activities during BSPU replacement is described.

A prototype, based on the OSGi technology, has recently been built, to validate the design of the proposed framework. By proposing performance metrics on this reconfiguration infrastructure and mechanism, the interrupt to the streaming application can be measured and analyzed. This concerns the main direction of further related work.

Next to that, we aim at strengthening the requirements elicitation for the context-aware infrastructure under study.

ACKNOWLEDGEMENTS

This work is part of the Freeband AWARENESS project (<http://awareness.freeband.nl>). Freeband is sponsored by the Dutch government under contract BSIK 03025.

REFERENCES

- Arlow, J. and I. Neustadt, UML and the Unified Process: Practical Object-Oriented Analysis and Design. 2002: Addison-Wesley.
- AWARENESS 2007: Dutch project: context AWARE mobile NETworks and ServiceS, Gvrnm. supported.
- Badrinath, B., et al., A Conceptual Framework for Network and Client Adaptation. Mobile Networks and Applications (MONET), 2000.
- Bagrodia, R., et al., 2003. iMASH: Interactive Mobile Application Session Handoff. In Int. Conf. on Mobile Systems, Applications, and Services (MobiSys 2003).
- Fox, A., et al., Adapting to network and client variation using infrastructural proxies: lessons and perspectives. Personal Communications, IEEE, 1998. 5(4): p. 10-19.

- Han, R., et al., Dynamic adaptation in an image transcoding proxy for mobile Web browsing. *Personal Communications, IEEE* [see also *IEEE Wireless Communications*], 1998. 5(6): p. 8-17.
- Mei, H., et al., An architecture for smart distribution of health signal processing, in *Freeband AWARENESS D4.14*. 2005.
- Mei, H., et al. A Flexible Vital Sign Representation Framework for Mobile Healthcare. in *1st International Conference on Pervasive Computing Technologies for Healthcare*. 2006. Innsbruck, Austria.
- Miskelly, F., Assitive technology in elderly care. *Age and Ageing*, 2001. 30(6): p. 455-458.
- MobiHealth. MobiHealth project webpage. 2003 [cited 2006 November]; Available from: <http://www.mobihealth.org>.
- Pattichis, C., et al., Wireless Telemedicine Systems: An overview. *IEEE Antenna 's and Propagation Magazine*, 2002. 44(2): p. 143-153.
- Shishkov, B., Dietz, J.L.G., 2005. Applying component-based UML-driven conceptual modeling in SDBC. In *ICEIS'05, 7th International Conference on Enterprise Information Systems*. INSTICC Press.
- Shishkov, B., Van Sinderen, M.J., 2007. Model-driven design of context-aware applications. In *ICEIS'07, 9th International Conference on Enterprise Information Systems*. INSTICC Press.
- Van Sinderen, M.J., H. Batteram, and E. Meeuwissen, *Freeband AWARENESS D1.1 "Scope and scenarios"*. 2005.
- Veld, M.H.A.H.i.t., et al., Context aware algorithm for epileptic seizure detection. 2005, *Awareness deliverables*.
- Wai Yip, L. and F.C.M. Lau, A context-aware decision engine for content adaptation. *Pervasive Computing, IEEE*, 2002. 1(3): p. 41-49.
- World_Health_Organization. Atlas: Epilepsy care in the world. 2005 [cited 2006 November]; Available from: http://www.who.int/entity/mental_health/neurology/Epilepsy_atlas_r1.pdf.
- Yarvis, M., P. Reiher, and G.J. Popek. Conductor: a framework for distributed adaptation. in *Hot Topics in Operating Systems*, 1999. Proceedings of the Seventh Workshop on. 1999.
- Yarvis, M., P. Reiher, and G.J. Popek. A Reliability Model for Distributed Adaptation. in *3rd IEEE Conference on Openarchitectures and Network Programming*. 2000.