

The Personal Medical Unit – A Ubiquitous Computing Infrastructure for Personal Pervasive Healthcare

Jakob E. Bardram

Centre for Pervasive Healthcare
Department of Computer Science, University of Aarhus
Aabogade 34, DK-8200 Aarhus N., Denmark
bardram@daimi.au.dk

Abstract. In this short paper we present an initial outline of the design of a *Personal Medical Unit* (PMU) for storing and synchronizing personal medical data with other clinical computer systems. The paper primarily discusses the motivation for such a pervasive healthcare device focusing on the potential in having the patient him or her self as the data integrator. The paper also outlines the preliminary design of the PMU Architecture.

1 Introduction

This paper describes an outline of a software architecture for a Personal Medical Unit (PMU), which can be used for storing and integrating personal medical data. With the advancement of processing power and memory capacity, contemporary technology is able to store and process quite extensive amount of data in small devices. For example, USB memory sticks can now store up to 1 GB of data, and most cellular phones have extensive processing power and a similar amount of storage capacity. Hence, it is possible to store most medical data, including images, for a patient on a personal token – like an USB memory stick. Furthermore, the development of short-range wireless connectivity, like IrDA, Bluetooth or ZigBee, enables easy communication between such a PMU device and its surrounding data sources and receivers.

The PMU, as described in this paper, is intended for storing personal medical data for a patient. It basically has the following features:

- It can *synchronize medical data* between a PMU and another medical or clinical computer system, like a Hospital Information System (HIS) or an Electronic Patient Record (EPR). These systems can be found in various places, including hospitals, out-patient clinics, nursing homes, specialized medical clinics, or at the general practitioners. Hence, by carrying a PMU device from one clinical setting to another, the patient him or her self ensure data integration between these two clinical systems.
- It can synchronize and store personal medical data from personal *medical sensing and monitoring devices*. For example, the PMU can store time series of weight, temperature, blood pressure, pulse, etc., which have been monitored at home.

- It can store *personal annotations*. For example, annotations about the intake of medicine, and why some prescription was not taken. Or annotations on how well some medication is working, or general notes about well-being related to some prescribed treatment.
- It can be used in an *emergency situation*. Paramedics, who find an unconscious person can read this person's PMU and immediately get crucial information about any chronic disease, or allergies to medication besides basic information about name, social security number, address, and who to contact.
- It contains a *private key* in a digital signature / public-private key infrastructure (PKI) and can hence be used to authorize the use and transfer of personal medical data. In this way, the patient has full control over medical data concerning him or her.

The exact form factor of a PMU has not yet been designed. It might be a USB memory stick, a necklace or some other piece of jewelry, an application running on a mobile phone, or a something completely different. In principle, the PMU could be a web-application that could be accessed by the patient via the Internet. However, the original concept of a PMU was some kind of physical artifact that the patient would carry with him or her, and use it for data authentication and synchronization in close proximity to the data source or data receiver. We call this principle for *Proximity-Based Computing*.

2 Motivation

The use of a personal and physical token for data storage and integration clearly has some disadvantages. Such a PMU token can be lost, stolen, misplaced, or the patient might not always carry it with him or her – it might be left at home when rushing to the hospital. Furthermore, there are many instances of medical interventions which happens while the patient and a doctor is not co-located. For example, when the doctor prescribes some medicine for the patient using the phone¹. Furthermore, there is a large group of patients, especially elderly and cognitive disabled patients, which would not be able to manage a personal physical device as the PMU, let alone the complexity of using such a small computer.

Despite these disadvantages, we still believe that the whole principle behind the PMU is worth pursuing for a number of reasons. We believe that it is possible to craft the design of the PMU in order to accommodate for the disadvantages mentioned above. Furthermore, we do not believe that the PMU is 'one size fits all', but is more to be viewed as an opportunity for people and situations where it makes sense. Hence, equipping an elderly citizen with a PMU might, in the general case, not be such a good idea.

¹ The whole idea behind 'telemedicine' and 'homecare' is to provide remote medical assistance and services between a patient and the clinicians. Hence, in such telemedicine and homecare scenarios, the PMU might not seem directly applicable. However, the PMU is actually designed to support various homecare scenarios, as discussed below.

2.1 The Patient as the Data Integrator

The primary motivation behind the PMU architecture is to address the ‘Integration Challenge’ in healthcare. The amount of different computer systems in use in healthcare today is extraordinary. Numerous systems exist, which each store and manage medical information for a patient. For example, clinical computer systems at the General Practitioners, in the different pharmacies, in nursing homes, in special medical clinics, and not to mention the large amount of different clinical systems used in a hospital. Hence, creating a coherent and appropriate collection of medical data for a patient is an enormous data integration task. A common strategy is to negotiate (or dictate) one common data format, which all data systems must support. However, already several such ‘standards’ exist. For example, the US-based ‘Health-Level 7’ (HL7) standard, the EU-based ‘Health Information System Architecture’ (HISA) standard, and the Danish National standard called ‘G-EPJ’. The consequence – in Denmark at least – is that in addition to all the systems that have proprietary data formats, we have different systems that support one of the standards mentioned above. Creating a centralized data integration where all systems basically can exchange data with each other is simply a gigantic task, and has so far also proven little success. Only a few central systems in the Danish healthcare sector work together in a rudimentary fashion.

In contrast, the basic data integration principle in the PMU design is to have ‘the patient as the data integrator’. In the centralized approach to data integration, the principle is to have all data in all systems integrated at all time – in case some of it is needed in the treatment of a patient. In the de-centralized approach used in the PMU design, the patient simply carries with him or her the relevant medical data, and data is only integrated when needed – i.e. when a patient with a PMU meets a clinician with a clinical system. Instead of data integration, the PMU uses *ad hoc data synchronization* as needed. As we will discuss below in section 3, this approach also needs some kind of standardization. However, the PMU Architecture relies on standards for the process of data integration rather than the data format as such.

2.2 Handling Heterogeneous and Non-standard Data

The centralized data integration approach, which relies on common data standards, obviously runs into problems when dealing with heterogeneous data that does not comply to the standard. Examples of such heterogeneous, non-standard medical data is personal annotations by a patient, small medical devices for monitoring, devices using some proprietary data format, etc. In many cases, it is unfeasible to ensure that a medical system complies to some standard, because changing the data format in a system often implies fundamental changes in all parts of the system.

The PMU architecture is designed to be able to handle and store arbitrary kind of data. The key principle is to have the PMU token accept, store, and hand over ‘raw’ data and let the receiver do the semantic interpretation of it.

2.3 Ubiquitous Coverage

The core disadvantage of the centralized integration approach is that access to medical data requires network connection to the central data storage. Thus, in an emergency

situation, the paramedics would need Internet access to be able to access a patient's medical data. Because the PMU stores medical data locally, only near-field connectivity is required. Hence, the paramedics can read the patient's PMU using his handheld computer via Bluetooth. In a traveling scenario, the tourist would also be able to share his or her medical data with a local physician without the need for network access to some central database.

2.4 Patient Empowerment and Data Privacy

From a societal and patient's rights perspective the PMU also creates the technological platform for literally 'putting the medical data (back) in the hand of the patient'. Data synchronization between the PMU and a medical system is controlled by the patient, who approves this transfer each time. Furthermore, the proximity-based nature of the architecture ensures that the patient (or at least the PMU) is co-located with the systems requesting medical data. Hence, the patient is in physical control over the exchange of data. This approach is an attempt to make the integration, use, and exchange of medical data physical and *tangible* for the patient.

3 Architecture Outline

The design of the PMU Architecture is in a very preliminary stage. In this section we will just touch upon the core challenges in this architecture. Figure 1 shows an outline of the PMU Architecture. The architecture basically contains three technical components: (i) the *PMU Token* itself, including its data storage formats and communication with its surroundings; (ii) the *PMU Infrastructure*, which enables data integration between the PMU and large healthcare related computer systems, like Hospital Information Systems (HIS) or Electronic Patient Records (EPR); and (iii) the *PMU Communication Protocol* between a PMU device and a data source or data receiver. Let us consider these three components.

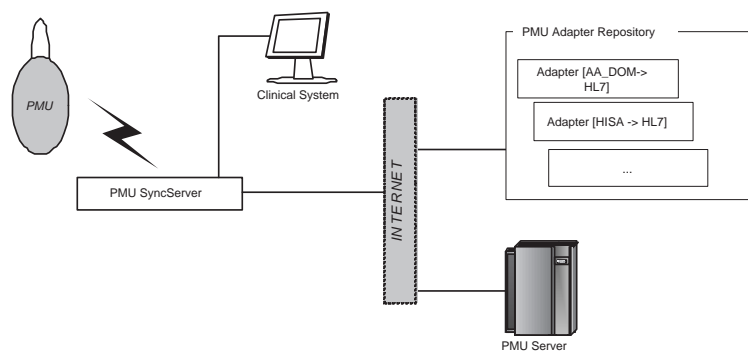


Fig. 1. An outline of the PMU Architecture.

3.1 The PMU Token

The PMU Token holds the medical data in an XML format. There is very little requirements to the format of the XML – it is basically a wrapper for heterogeneous XML data of all kind. Hence, all medical data, which can be serialized into XML can be stored on the PMU. In addition, the PMU can store byte objects, which can be references from the XML data, such as medical images.

The PMU should be able to communicate with systems in its proximity using e.g. Bluetooth. Furthermore, it might contains some minimal user-interface in terms of an on/off button and a button for acceptance of data transfer to and from the PMU. This acceptance ‘button’ might be a finger-print reader for user authentication. A PMU might have a display, but we do not consider this a requirement.

3.2 The PMU Infrastructure

Beside the PMU Token, the PMU Infrastructure consists of the following components:

- **PMU Synchronization Server.** The PMU SyncServer resides on the clinical computer system and manages the communication and synchronization to and from the PMU device, using the PMU protocol over some network stack (e.g. TCP/IP over Bluetooth). The PMU SyncServer is also responsible for PMU device discovery using e.g. the Service Discovery Protocol (SDP) in Bluetooth.
- **PMU Adapters.** A PMU Adapter is used to adapt or transform medical data in a system into data, which can be store on the PMU. These adapters are simple Java programs, which are written by the vendor of a medical system. The PMU Synchronization Server uses these adapters in the synchronization process. Simple XSLT stylesheet transformation can be used, or more sophisticated adapters can be written.
- **PMU Adapter Repository** – this is a network-based registry containing a range of PMU Adapters. PMU SyncServers can query for appropriate adapters and adapters can cooperate by creating e.g. a pipeline of adapters.
- **PMU Server.** The PMU Server contains a synchronized version of a patient’s PMU device. Hence, the PMU Server can be viewed as a backup service. However, it is intended to be used in situations where the patient and the clinicians (or the PMU and the clinical system) is not in close proximity of each other – e.g. in telemedicine cases. Voice recognition for user authentication could be used to ensure data privacy and user control.

3.3 The PMU Protocol

The PMU protocol consists of two parts. The one part is an implementation of the standard HTTP protocol, enabling the PMU to work as a web server. Hence, data stored on the PMU can be browsed using a web browser on a computer nearby. The PMU needs to be able to communicate using TCP/IP, which can be achieved in the Bluetooth stack. Furthermore, the content of the PMU – the medical data – should be formatted in HTML. Therefore, each data XML element in the PMU has an associated XSLT

stylesheet for transformation between the XML format on the PMU and HTML. This stylesheet is required to be available from the PMU Registry as part of an adapter. The PMU might also cache these stylesheets for use in non-connected cases. However, this is not a requirement of the PMU device. The transformation is done on the client side (using the XML/XSLT transformation in the browser), because we expect the client computer to hold more processing power than the PMU. Browsing medical data might be protected by user authentication, like a normal web server supports.

The second part of the PMU protocol is a synchronization protocol for data authentication and synchronization, which is used in the case where a PMU and some data source or data receiver needs to exchange medical data. We have not yet designed this protocol but we expect to be able to use SyncML as our basic standard.

4 Discussion

We have already conducted some future workshops where the use of a PMU in the Danish healthcare sector was discussed and role-played by physicians, nurses, home care nurses, and patients. A preliminary prototype for emergency cases has been implemented and one paramedic has tried it out in a lab setting. Based on such experience from these and other workshops to come, the next step is to finish the design of the PMU architecture and to initiate an iterative implementing of it. We are currently starting two home care projects at the Centre for Pervasive Healthcare, where the PMU is to play a role. Hence, much more evidence on the PMU as a platform for pervasive healthcare is to be gained in the coming two years.

Short Bio for Jakob Bardram

Jakob E. Bardram is an Associate Professor at the Computer Science Department at the University of Aarhus, Denmark, and he is the manager of the Danish Centre for Pervasive Healthcare [www.cfph.dk]. His research interests include; Pervasive Computing, Pervasive Healthcare, Distributed Object-Oriented Systems, and Human-Computer Interaction. The Danish Competence Centre 'ISIS Katrinebjerg' is funding his research.