Mobile Health and Wellness Application Framework

T. Laakko, J. Leppänen, J. Lähteenmäki, A. Nummiahko
VTT Technical Research Centre of Finland, Espoo, Finland

1. Introduction

Telemedicine is increasingly gaining popularity due to its high potential for cost savings and increased efficiency in healthcare. Although, there are currently only a few examples of operational telemedicine services, rapid growth in both business-to-business (B2B) and business-to-consumer (B2C) application areas is expected [1]. Telemedicine applications are currently largely based on using mobile phones as user terminals [2, 3]. In a typical case, a mobile application is used to connect medical measurement devices with back-end information systems [4]. Due to the limited processing power and user interface functionality, usability is a remarkable challenge for mobile health applications. Context-sensitivity has been identified as a means for improving the user experience by adapting the user interface and providing relevant information and services according to the user’s situation [5]. Additionally, it is important that the scope of the application is limited and clearly focused to the targeted use case. Application frameworks can be used to speed up the development process and to keep the development costs of tailored software at reasonable level. Koch [6] describes a generic framework, which defines the elements and interactions required to implement intelligent mobile applications. Broens et. al. [7] present requirements for application framework for context-aware mobile health applications and illustrate the framework usage in an epilepsy patient monitoring use case.

Both [6, 7] take a system-level approach and provide support for architectural design covering the mobile application and the back-end system. In the present paper, we intend to provide more detailed support for the definition and implementation of the actual software application in the mobile unit. We propose an enhancement to the state-of-the-art by providing an application framework specifically addressing document-based data management, local storage and connectivity. Based on the framework we have integrated the UPHIAC platform (Ubiquitous Personal Health Information Access), which can be used for providing information exchange between healthcare professionals, patients and measurement devices (Fig. 1). The platform can be used as a basis for developing mobile applications for specific purposes.

The UPHIAC platform connects with a server component which provides conversion between the data structures handled by the mobile application and standard medical document formats such as the CDA (HL7 R2) and Annotated ECG (aECG) [9] by HL7 [10]. Operational prototype for tele-ECG domain has been implemented as an example use case.

2. The UPHIAC Mobile Application Framework

The UPHIAC application framework provides a set of core classes constituting basic functionalities needed in mobile health applications. Additionally, open interfaces for measurement, storage, network connectivity and user interface implementations are provided (see Fig. 2).

A new UPHIAC application is created by providing implementations for the core interfaces. We have also provided reference implementations for the interfaces, which together with the core classes constitute the UPHIAC platform. New applications may be generated based on the platform so that implementation of all the interfaces is not required.

Summary

Objectives: There is an increasing need for user-friendly and interoperable mobile applications in health and wellness domain. The objective of this work has been to provide support for rapid and cost-effective development of such applications.

Methods: We have introduced an application framework which provides a generic tool for mobile application designers. We have demonstrated the usage of the application framework by providing an example implementation and demonstrating its usage in a Tele-ECG use case. In order to support interoperability we propose a solution compatible with clinical document standards such as the HL7 CDA.

Results: A new mobile platform applicable for a wide range of telemedicine and wellness applications is introduced. The platform provides connectivity between healthcare professionals, patients and measurement devices. It is based on an open application framework that provides interfaces for measurement, user interface, database and network connectivity implementations.

Conclusions: Mobile application development based on the application framework was demonstrated successfully. The developed UPHIAC platform adopts a new technical approach using a local database solution for caching of information at the mobile terminal. The platform applies a document-based approach providing a versatile and reliable way of sharing and collaboratively complementing of health information including standard information model documents (HL7/CDA), and integration with health information systems and personal storages. The implemented mobile tele-ECG case demonstrates the overall function of the platform. The platform can be used as a basis for developing new applications for specific purposes.

Keywords

eHealth, mobile healthcare, mobile platform, telemedicine applications, application framework

Methods Inf Med 2008; 47: 217–222
doi: 10.3414/ME9113
Documents are the main data entities handled by the platform. This complies with the document-based approach adopted by HL7 in the CDA standard [8]. A document may include measurement data augmented with additional metadata such as information about the patient (e.g., identity number). The raw data from measurement device is completed into the structured document form. Further, the document provides the means for sharing, analyzing and collaboratively completing the health information content.

The core interfaces (cf. the sub-interfaces of the abstract base interface UP-HIACInterface in Fig. 3a) include:

- **Measurement.** This interface communicates with measurement devices. Each device provides a measurement service that constitutes an entry point for data transfer and measurement.

- **Network.** The data are shared and synchronized with a virtual storage space located in the server side (e.g., an external health information system or a personal storage). It is also possible to send and receive alerts within the virtual space.

- **UI.** The interface takes care of user interaction and visualizing defined data, e.g., as tables or graphs. It can be adapted suitable for different roles of authenticated users.

- **Storage.** The storage interface takes care of the local storing of the measurement data and other information handled by the platform.

The core classes (cf. the subclasses of the abstract base class UPHIAC in Fig. 3b) are summarized in the following:

- **The Document class represents the contents and the meta-information of a document.**

- **The Service class represents a service, which acts as the remote source or receptor for documents. A service may be a health information system or a personal storage.**

- **The User class represents user and user role-specific information.**

- **ServiceType provides additional information related to a service, including information about which document types can be applied by the service.**

- **UserService specifies how a user can access and use a service.**

- **DocumentType provides additional information about the document types.**

- **DocumentService provides information about services applied to a document.**

### 3. Retrieving Measurement Data

A measurement service provided by a measurement device is used to start and stop measuring, and to retrieve and push measurement data.

The retrieving of data can be accomplished in several ways (cf. Fig. 4). The control of starting the data transfer can be in either side, and the measurement can be initiated either by the UPHIAC application or by the measurement device. In the case a), first, the application makes a request for measuring. In the case b), the application indicates that it is ready for receiving...
measurement data. In the case c), the control is in the measurement device side.

Typically, the communication with a measurement device occurs wirelessly, e.g., using Bluetooth or Near Field Communication (NFC). However, UPHIAC is not restricted to certain communication techniques.

The Measurement interface implementations are specific to the measuring devices and the supported ways of communication. The implementations (cf. Fig. 5) should provide methods for:
- sending and receiving raw data (bytes) to/from measurement device,
- querying measurement devices and setting device to be used, and
- getting name and address of the measurement service provided by the measurement device.

The querying measurement devices and selecting the device is required only when the measurement device is changed (presetting phase). The selected device is set as the current measurement device and stored for later use.

Reference implementations of the Measurement interface have been done for each of the cases of Figure 4. In the implementations, the Java Bluetooth API (JSR 82) was used. Also, the Java MIDP push registry facility has been applied to enable applications to be launched automatically, without user initiation. On one hand, the push registry is used to wake up the measurement device while the UPHIAC application is requesting or waiting for measurement, and, on the other hand, to start the application while the measurement device is directly pushing data.

Also, user-specific information such as user profile, role, and access rights to services are stored in the database.

Typically, the storage and processing capabilities of a mobile device are limited compared to that of desktop systems. Therefore, it should also be possible to transfer the data to external storages if required. Further, the mobile storage system needs to be as light as possible and, thus, not necessarily has to provide large general database functionality (such as full SQL database functionality), but more likely the operations that are required by the application suite. Also, the portability demands in providing implementations to a variety of different mobile devices should be noticed.

The database is defined so that it provides general local storage for the information handled by the diverse UPHIAC applications. Figure 6 outlines the database schema, which can be implemented as a relational database (e.g., SQL database). The data fields of the tables have a close relationship to the contents (attributes) of the respective core classes.

The database is defined so that it provides general local storage for the information handled by the diverse UPHIAC applications. Figure 6 outlines the database schema, which can be implemented as a relational database (e.g., SQL database). The data fields of the tables have a close relationship to the contents (attributes) of the respective core classes.

The storage interface (Fig. 7) provides an application interface to the local database, and all the database operations by UPHIAC applications are performed through it. There

4. Local Data Management

The local database implements the Storage interface of the framework and manages locally the measurement data enriched with additional information (metadata etc.) contained in the input data model. The information includes the services provided by external health information systems (and personal storages) and measurement services.
are 34 different interface methods. These include, for instance, basic methods to store instances of each of the classes (addDocument, addUser, …) and to retrieve instances from the database (getDocument, …). The reference implementation of the Storage interface provides a light implementation of the database schema with relational database functionality for UPHIAC applications. It is implemented using the Java MIDP RMS (Record Management System). The database is shared within the application suite, which may include several UPHIAC applications. In the database implementation, the payload data is stored with IDs and other metadata for finding and managing the data. A document service specifies the services applied to the document and may include references to external storages. Therefore, the local database is also aware of the real placement of each data element, even if located in an external storage.

5. Utilization of Clinical Standards

UPHIAH is typically used in telemedicine applications where connection between health sensors at the patient and a back-end system is needed. This includes B2B services where medical professionals exchange information related to the patient’s care process and B2C services where monitoring information is sent from the patient to the care unit.

Free form text messages between the care process contributors are supported. However, in order to share health information between systems, a common understanding of the document structures, semantics and handling rules is needed. Interoperability can be enabled based on specifications and standards – such as those prepared by HL7 [10] – enabling methods for exchanging, storing and accessing the patients’ health information. The standard-based information can be stored into hospital systems like Electronic Health Records (EHR).

UPHIAH relies on a server component for conversion of information between the compact data structures handled by the mobile application and the standard document structures. Two mechanisms are available: collecting manually entered form data and forwarding of measurement data. The HL7 CDA R2 [8] document format has been used to convey information entered in user-filled forms. In order to speed up communication only a subset of the CDA document content is transmitted over the wireless link. The document subset includes the relevant form fields to be exchanged and the instructions for their appearance in the mobile user interface. The document subset is defined by...
W3C/XForms definitions embedded in the CDA content. Embedding is based on a local mark-up extension as allowed by the CDA R2 standard.

In the case of measurement data handling, the data is forwarded to the server in the original form used by the measurement device. Conversion into standard format is carried out at the server. For example, the aECG format is used for ECG data. The server module supports completion of the measurement metadata along the care process. A document can be completed step-by-step (cf. Fig. 8):

- **Step 1.** Empty XML-document for ECG data is available at the server.
- **Step 2.** When the user logs in, the user-specific data, such as role and personal preferences, is sent to the server, which can then start filling in the XML document (Provider Info).
- **Step 3.** Each measurement instance produces telemedicine data, which is sent to the server. The measured data is used for updating the structured document (Payload Data).
- **Step 4.** The document is updated with the measurement target’s overhead data (Patient Info).

Incremental completion of the telemedicine document is especially useful in urgent telemedicine applications – such as Tele-ECG – where the patient id may not be available when the process starts. The process can then directly begin with the measurement. When the patient’s personal information (name, personal identity number, etc.) is added afterwards, the measurement data can be linked with the patient’s medical history, which is in key role in the patient’s health care process.

### 6. Case Example: Tele-ECG Measuring

Tele-ECG refers to a use case, where the electrocardiogram (ECG) of a heart patient is sent from the ECG measurement device in the field unit (e.g. ambulance) to another location for the analysis carried out by a cardiologist.

The use case is supported by the UPHIAC platform and involves the following steps (cf. Fig. 9):

- When started the UPHIAC application in the field unit is automatically initialized and set-up to receive data (cf. case b in Fig. 4).
- The actual measurement is started by the measurement device user interface functions. The measured data is received by UPHIAC through Bluetooth connection and stored locally. Subsequently, the data is delivered to the remote server and shaped into the structured form, namely into the aECG format based on HL7 version 3 messaging standard [9].
- The cardiologist receives the structured data and, then, analyzes the graphs, prepares the first conclusions, and sends the results back to the field unit.
- After or during the treatment phase in the field unit the measured data will be complemented with the patient’s personal information such as the personal identity number.
- The cardiologist gets the complemented data and will be able to search more information about the patient from EHR and prepare the patient’s reception at the hospital.

In this use case, PIR™ document server [12] acts as the mediator of documents to the back-end system, which provides the user interface for ECG graph viewing and analysis. The ECG measurement device is emulated by another mobile phone connected with Bluetooth to the UPHIAC application. The measurement device emulator allows to test applications before the implementation of the actual interface functions for a particular measurement device.

The above use case is based on reference implementations of the UI, Storage, Measurement and Network interfaces. The reference implementations can be used as templates for other use cases. For example, we have modified the UI and Network reference interfaces to support a home monitoring use case for collecting blood pressure information.

### 7. Discussion and Conclusions

As indicated in the introduction our aim has been to provide an application framework, which helps rapid development of mobile applications. As a demonstration of using
the application framework we have implemented the multipurpose UPHIAC platform, which can be used as a basis for new applications for specific purposes. The design targets of the platform have been modularity, generality, reusability and extensibility. The open interfaces of the mobile platform facilitate implementations of different kinds of applications and connectivity with a variety of measurement devices and health information systems. The local database interface provides versatile and efficient methods for managing the core data entities. Further, the document-based service architecture approach enables the sharing and collaboratively complementing of health information and integrating the platform within organizational and personal storages.

Using standards and open interfaces bring remarkable advantages as interoperability between systems can be achieved. UPHIAC includes server-side modules which take care of converting documents between the measurement device outputs and the formats used by the back-end systems. Currently, modules for CDA R2 and aECG have been implemented. New converter modules can be implemented as needed.

The CDA R2 format could also be used for building a portable personal health record (PHR) as proposed in [11]. The PHR may provide the needed mobility of health information between healthcare units in the case when direct connection between the units is not possible. The UPHIAC platform provides all the relevant functionalities, such as CDA R2 and local storage support, for the implementation of a mobile phone-based PHR.

International standards for describing health information are now widely available [8] and large companies are aiming at interoperable solutions through industry consortia, such as the Continua Health Alliance [13]. Despite of the trend towards open systems, the fact is that most of the clinical devices are using proprietary data formats and open APIs (Application Programming Interfaces) are typically not available. The proposed application framework is targeted to be flexible. It is not bound to any particular network or device interface and it can be used with both open and proprietary document formats.

References

Correspondence to:
Timo Laakko, D. Sc. (Tech.)
VTT Technical Research Centre of Finland
Voormiestehntie 3, Espoo
P.O. Box 1000
02044 VTT
Finland
E-mail: timo.laakko@vtt.fi