Implementation of an end-to-end standard-based patient monitoring solution

l. Martínez, J. Fernández, M. Galarraga, L. Serrano, P. de Toledo, S. Jiménez-Fernández, S. Led, M. Martínez-Espronceda and J. García

Abstract: A proof-of-concept design of a patient monitoring solution for intensive care unit environments has been presented. It is end-to-end standard-based, using ISO/IEEE 11073 (X73) in the bedside environment and EN13606 to communicate the information to an electronic health-care record (EHR) server. At the bedside end, the system is a plug-and-play sensor network communicating with a gateway that collects medical information and sends the data to a monitoring server. The monitoring server transforms this information into an EN13606 extract to be stored on the EHR server. The system has been implemented to comply with the last X73 and EN13606 available versions and tested in a laboratory environment to demonstrate the feasibility of an end-to-end standard-based solution.

1 Introduction

Advances in technology have implemented many changes in the bedside environments of hospitals. Medical devices (MDs), computers and sensors at the point of care (PoC) end are now part of the intensive care unit (ICU) [1−3]. Moreover, technology possibilities continue growing as sizes and costs shrink [4−6]. These devices and sensors acquire huge amounts of very valuable information, without the need for manually writing down each measurement, contributing to the generation of the electronic health-care record (EHR) [7−9].

The communications and interfaces among components of patient monitoring systems, and between these systems become now very important in exploiting all the possibilities offered by the information gathered [10−12]. However, different manufacturers use their own software and communication protocols; building proprietary solutions that can only work alone or inside a single-vendor system. As each device speaks a different language, an important interoperability problem emerges, leading to difficulties when a part of a system must be replaced as well as high costs [13]. Furthermore, the information acquired cannot easily be integrated into and exchanged with EHRs.

There is a need for developing open sensors and middleware components that allow transparent integration and plug-and-play interoperability of monitoring devices and systems (see Fig.1). The use of communications standards seems to be an efficient way to solve these problems [14, 15]. Most stakeholders in hospital information systems, medical devices and telemedicine scenarios agree that standard-based solutions are needed to foster the development of these markets and improve the service provided to patients. However, it is frequently claimed that it is not feasible to develop such systems because of either costs or lack of development of the standards.

Medical information and communication standards define information representation and exchange formats allowing interoperability. Some of these standards are: POCT-1A2 (communication protocols between the device and an access point [16]), health level 7 (HL7, for the exchange, management and integration of electronic health-care information [17]), ISO/IEEE11073 PoC-MDC (for PoC-medical device communications [18]) and EN13606 (for EHR communication [19]). The ISO/IEEE11073 PoC-MDC (also known as X73) is a family of standards for MDs interoperability that are considered European standards via the TC251 of the European Committee for Standardization (CEN). X73 was formed by absorbing three previous standards: (1) a standard for vital signs information representation (ENV13734 or VITAL) for the upper layers [20], (2) a standard for interoperability of patient connected MD (ENV13735 or INTERMED) for the intermediate layers [21] and (3) a standard for the medical information bus (MIB/IEEE1073) for the lower layers [22 − 24]. EN13606 is the European standard for EHR. The standard defines the architecture of an interoperable EHR, as opposed to other standards in the domain (i.e. HL7) that only define standard messages to implement the most common data exchanges among information systems in hospital environments. The main advantage of EN13606 is its transparency: it is possible to add new medical information to the standard without modifying the model. This goal is achieved by the so-called dual model: a reference model (that gathers information features that are not subject to change) and an archetype model (that gathers domain concepts, which may vary as medical

I. Martínez, J. Fernández and J. García are with the Communications Technologies Group (GTC), Aragon Institute for Engineering Research (I3A), University of Zaragoza (UZ), Spain

M. Galarraga, L. Serrano, S. Led and M. Martínez-Espronceda are with the Electrical and Electronics Engineering Dep., Public University of Navarra, Spain

P. de Toledo and S.Jiménez-Fernández are with the Bioengineering and Telemedicine Research Centre (GBT) of the Technical University of Madrid, Spain

P. de Toledo is also Visiting Professor at the Informatics Dept. of the Carlos III University of Madrid, Spain
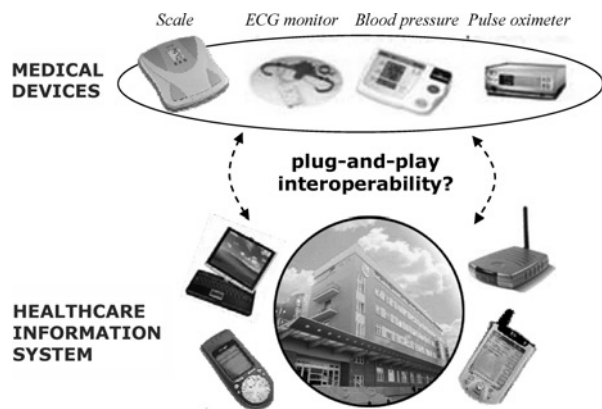
E-mail: imr@unizar.es

Fig. 1 *Interoperability among medical devices and healthcare information systems*

knowledge evolves). A key feature in the standard is the communication model, which specifies how to exchange information among clinical information systems. The element exchanged is the extract, a document that may contain any type of clinical information, generated on the basis of the different archetypes previously defined. To interpret an extract, the receiving system may look up these archetypes in a common repository.

In the field of patient monitoring MDs, there have been several implementation experiences based on standards, even commercial devices compliant –to different degrees– with standards, and demo experiences where the interoperability was tested [25–27]. Most of these implementations were done using the pre-11073 standards (VITAL or VSIR [25, 28–31], INTERMED or MIB [32–35]). The X73 family of standards is changing very fast, because of both the market and industry demand to adapt it to new functional requirements (especially those arising from its use in telemedicine and home monitoring scenarios), and to the wireless communication technologies zoo (i.e. Bluetooth, Zigbee, Wibree, WiFi, RFID etc.) [36, 37]. The implementation of the X73 standard is not trivial, and this complexity is one of the reasons why some manufacturers are still reluctant to adopt it.

The field of EHRs has also brought several implementations of standard-based systems allowing the sharing of EHR between hospitals and addressing challenges for all healthcare organisations, such as accessibility and relevance. HL7 is a widely adopted standard, with implementations in most hospitals around the world [38, 39]. However, HL7 integration is based on the exchange of messages, and has several limitations for the implementation of a higher level interoperable EHR and, unlike HL7, there are a few implementations of EN13606 [40].

To the best of our knowledge, there is no experience where the full chain from bedside monitoring devices to EHR has been implemented using standards. Even though there have been some initiatives to combine different standards, the vision of an entire end-to-end standard system is not yet a fact.

The system presented is a proof-of-concept design to show that it is feasible to create a patient monitoring solution for ICU environments that is end-to-end standard-based. We have used ISO/IEEE 11073 (X73) in the bedside environment and EN13606 to communicate the information to an EHR server. The components of this system (sensors, gateway, EHR server etc.) can be replaced individually by equivalent standard-based devices with no need of system reconfiguration. In the case of bedside monitoring devices, this interoperability is plug-and-play.

This work aims to generate know-how and implementation guidelines to be used in future developments and detect requirements that have not been addressed up to now. The feed-back results will be used to analyse how these implementations must be adapted to follow the standard and how the standard can be modified to better fit real-world needs, especially keeping in mind telemedicine and home monitoring scenarios.

The paper is organised as follows. First, an overview of the architecture and design of the proposed system is given in Section 2. The implementation experience according to X73 and EN13606 standards is detailed in Sections 3 and 4, respectively. Results obtained are discussed in Section 5.

2 System architecture and design

The proposed system architecture, including the fully standard-based prototype composed of independent and interoperable modules, is presented in Fig. 2. The generic design consists of a PoC (in the ICU) based on a X73 gateway that receives and integrates the information collected with several X73 MDs. These X73 gateways from each PoC (even from others X73-sensors networks) interconnect with the monitoring server to manage the e-Health service. Moreover, this monitoring server can send the acquired patient information to the EHR server according to the EN13606 standard. Fig. 3 shows the mapping of this proposal and our implementation. The specific technical characteristics of each architecture element are:

2.1 X73 MDs and adapters

The existence of X73 MDs is strongly conditioned to the vendors and manufacturers. Currently there are no 100% X73 compliant MDs, but the proposed architecture will allow to integrate them as soon as they are available. Moreover, it is difficult to find MDs with an
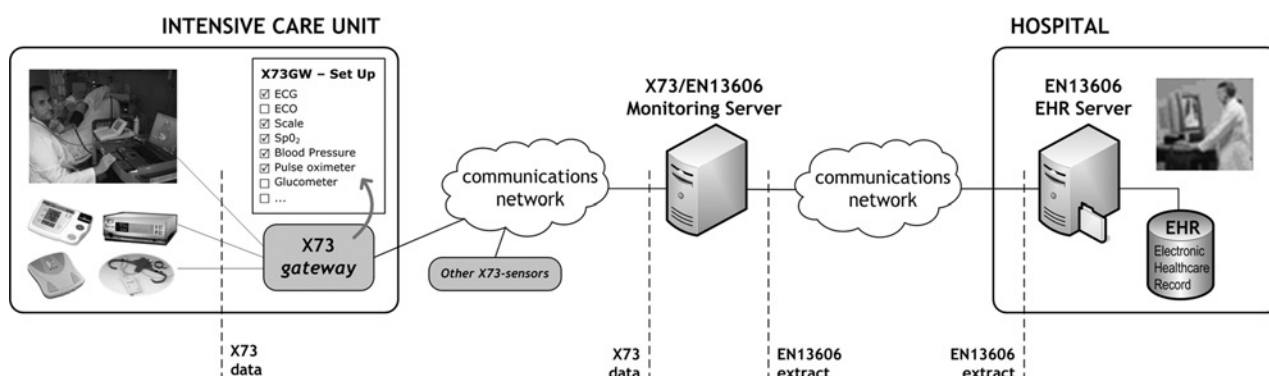

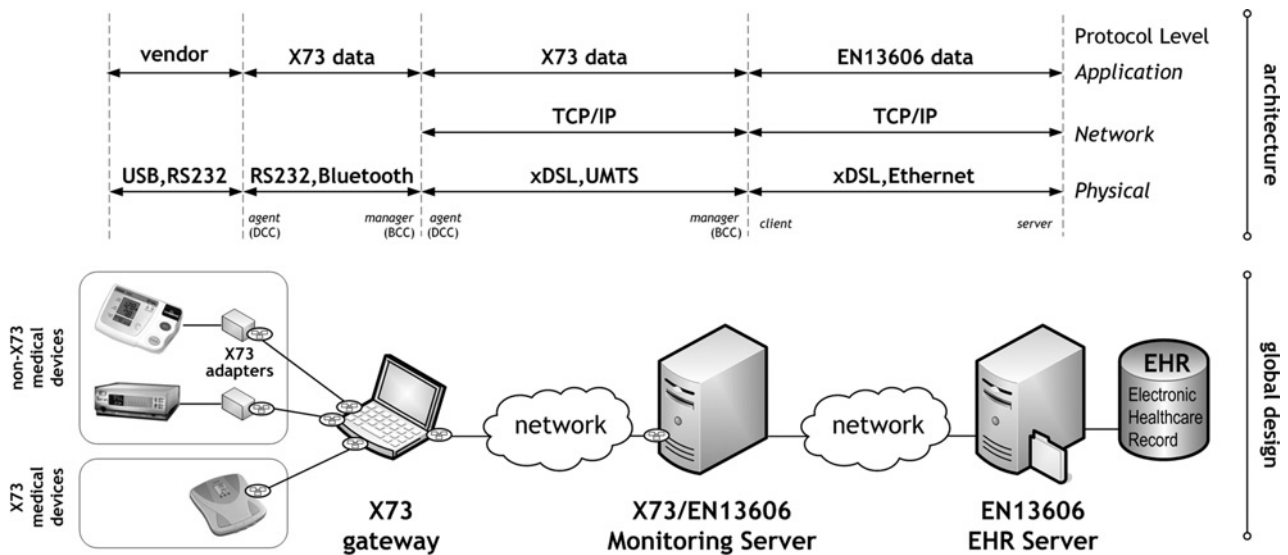**Fig. 2** *End-to-end standard-based generic scenario*

Fig. 3 *Proposed end-to-end X73/EN13606 design scheme*

X73-compatible physical output (i.e. MIB or Bluetooth); the most extended interfaces (i.e. USB or serial line) are not included at this moment in the X73 standard (only RS-232 and IrDA). Thus, in this development, proprietary MDs with X73-adapters have been used. These adapters, on one side, implement the standard X73 and, on the other, proprietary interfaces that can only be modified by knowing the vendor's programming codes. These MDs are OMRON 705IT (blood-pressure and pulse measurements, with a 28 acquisitions memory) and DATEX-Ohmeda 3900 (pulse oximeter with a serial-port output each 2 s that includes blood oxygen saturation (SpO$_2$), heart rate values and alarms).

2.2 X73 gateway

The designed gateway is an X73 access point. It supports a network of X73-compatible MDs measuring vital signs from different patients in different locations. The data acquired from the different sensors are transmitted to the X73 gateway, which can be personalised by configuring its working mode (by setting the buffering frequency, the synchronisation and transmission techniques used, or the type of X73-sensors, as it is shown in X73GW-SetUp, see Fig. 2). The X73 gateway provides a logic module for medical data information base (MDIB) that allows monitoring and obtaining alerts and measurements. Thus, following the X73 nomenclature, this gateway is represented like a medical device system (MDS): when each MDs is connected or disconnected (plug-and-play) the MDIB is automatically adjusted, a MDS is created or destroyed in the object hierarchy and the changes are propagated to the clients. In this implementation, for example, the MDS has two channel objects: pulse-oximeter channel (with two metrics: oxygen saturation information (SpO$_2$) and the plethysmograph waveform), and pulse-rate channel (with a unique metric: pulse rate information).

2.3 X73/EN13606 monitoring server

The monitoring server provides a double role: manager of the communications with the X73 gateways, and client of the connections to the EHR server. Thus, it may belong to the ICU, the reference hospital or even it can be an external management server. The plug-and-play functionality is remarkable in this element because it receives the measurements from different X73 gateways (associated to diverse patients

and sensors) and through different technologies. The most usual transmission lines in ICU environments are RS-232 RJ-45, or USB, but the use of cables is actually being substituted by other wireless solutions: infrared (IrDA), RF WLAN (802.11x), wPAN (Bluetooth) or Zigbee (802.15.4). Some of these technologies are currently not included in X73, but their future development is foreseen (see Table 1).

As a manager of the X73 gateways, the monitoring server manages the decision making in an intelligent way through several functions: automation of the data acquisition process, technological transparency (independent of the transmission modes of the X73 gateways), pattern's definition to identify, detect and manage anomalies (alarms, failures), patient specifications (decision thresholds, operation limits), updates and so on.

As a client of the EHR server, the monitoring server creates an EN13606 extract (from X73 data of the MDs) which is transmitted to the EN13606 EHR server in an intermediate archetype file format.

2.4 EN13606 EHR server

The EHR server is a clinical information storage system where the data associated with each patient are added. It adapts the incoming information from the MDs to an EN13606 compliant format. Thus, it receives the archetype files together with the EN13606 extract, validates this extract according to EN13606 standard, stores it in the appropriate EHR database and sends an acknowledgment back to the client. This process has been traditionally studied in an isolated way as an EHR communications interoperability problem, but in this implementation experience it is integrated with the monitoring process to propose an end-to-end standard-based complete solution.

Table 1: Available X73 connectivity technologies

Technology	Standard	Status
RJ-45 (IRDA-based)	11 073−30 200	available
infrared (IRDA)	11 073−30 300	available
IP-based	11 073−30 400	future work
RF WLAN (802.11x)	11 073−30 501	future work
wPAN (Bluetooth)	11 073−30 502	future work
Zigbee (802.15.4)	11 073−30 505	future work

3 X73 Implementation. Gateway and monitoring server

3.1 Implementation methodology

Currently, the X73 standard is in the development phase; indeed, many of its parts are still in draft status. It includes a family of standards that can be jointly used at different levels providing connectivity to the involved devices and giving a complete solution from the lowest levels (the wire itself and the connector) to the highest levels (abstract representation of information and services). These standards are divided into the following main groups: device data (object-oriented model and terminology for data representation), services of general application (events or polling), transport (wired or wireless), network communication (several networks interconnected) and gateway standards (DICOM or HL7 messages). Thus, any implementation experience should require a structured design following a top-down process within the stack layers. The most relevant X73 modules to be considered are:

– Medical device data language (MDDL): it is the X73 nomenclature that stands for the semantics and syntax used in the fields of the protocol data unit (PDU), usually with object-oriented attributes.
– Static model/domain information model (DIM): it is the object-oriented model to represent any device, element or metric that is going to be communicated. It is presented in the following packets: virtual medical or device objects, channel, metric or numeric objects, sample arrays, enumeration, complex metric and so on.
– Dynamic model: it provides a communication model based on the ISO concept of agent-manager with a 'device communication controller' (DCC) and a 'bedside communication controller' (BCC) for agent and manager, respectively (see Fig. 3). The X73 communication between two devices (DCC and BCC) follows four synchronised steps: 1-connection, 2-association, 3-configuration and 4-operation, using a finite states machine.

3.2 Implementation experience

Following this methodology, the complete implementation experience scheme is shown in Fig. 4, where each element has been considered individually to facilitate a progressive development directly led by X73. The programming languages used have been: Java and C/C++ (Cygwin and GNU GCC 3.4 environments). Various tools also were needed such as ANTLR 2.7, Java SDK 5.0 and ASN.1c 0.9.22 compiler. Abstract Syntax Notation One (ASN.1) is a language to define standards regardless of implementation (it defines what is a 'type', a 'module', how to 'label' a type so it can be correctly codified etc.). Basic encoding rules (BER) are a set of rules to codify ASN.1 data in a sequence of octets that can be transmitted through a communications' link (it defines the methods to codify ASN.1 values, rules to decide the use of each method, the data format etc.). Thus, with the ASN.1c 0.9.22 compiler, the X73 specifications for message exchange among layers are translated into BER codification and ASN.1 structures and functions (which are packed into a static ASNX library). Moreover, MDER are the proposed codification rules for X73 specialised MDs (they define methods to transform ASN.1 in a byte stream ready for the communication and they are optimised for managing X73 objects). Because the ASN.1c compiler does not support the MDER transfer syntax, an *ad hoc* translator (ANTLR-based) has been developed for working with MDER and BER. Thus, the translator generates a syntactic tree from an ASN.1 grammar that allows defining the ASN.1 structures (specified by X73) and writing the C++ source code and the Java utilities.

Following the top-down process, the protocol stack ASN.1 implementation is described in Table 2:

– Firstly, the application layer is defined by several protocols: ACSE (for association control), CMISE (for the basic services defined in VITAL) and ROSE (for the link between call requests and responses). In this proposal, CMISE and ROSE service elements have been grouped in the CMDISE interface defined by X73. MDER syntax uses a set of ASN.1 data types, and redefines the bit-pattern encoding of the types. The ASN.1 structures are automatically generated using the ANTLR parser and each ASN.1 data type is mapped to a C++ class.
– Secondly, the presentation layer is mainly a negotiation mechanism for the syntaxes used by higher layers: the abstract syntax (which set of messages are to be exchanged) and the transfer syntax (how the messages are encoded); therefore the MDER syntax has been designed in order to reduce complexity of the message processing implementation.
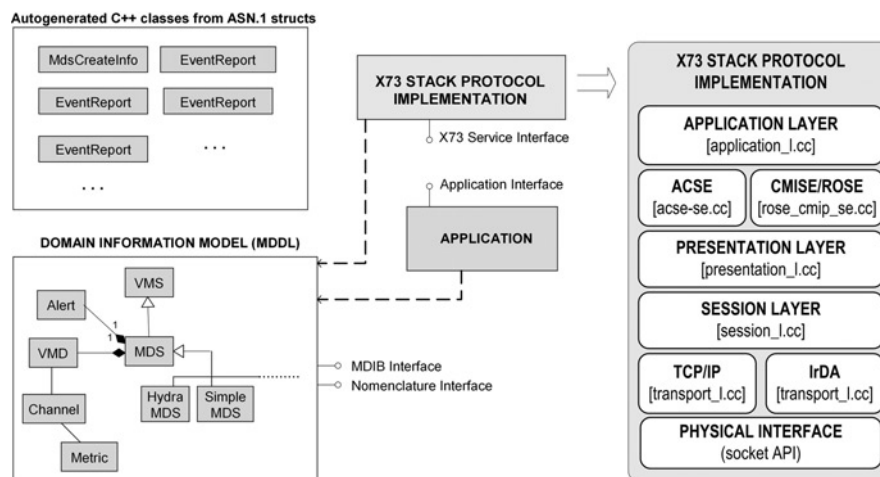

Fig. 4 *X73 gateway and monitoring server implementation scheme*

Table 2: X73 layered interface design

Protocol level	source	event
ACSE (X.227)		
`class acse_se.cc {`		
`public:`		
`void p_con_cnf (const st_buffer & buffer)`	AC-peer	AARE+
`void p_con_rf (const st_buffer & buffer)`	AC-peer	AARE-
`st_buffer p_con_ind (const st_buffer & buffer)`	AC-peer	AARQ
`st_buffer p_rel_ind (const st_buffer & buffer)`	AC-peer	RLRQ
`void s_rel_cnf (const st_buffer & buffer)`	AC-peer	RLRE+/-
`void p_abort ()`	AC-peer	ABRT
`void assoc_req (const st_buffer & buffer)`	AC-user	A-ASCreq
`void rel_assoc_req ()`	AC-user	A-RLSreq
`void abort_req ()`	AC-user	A-ABRreq
`};`		
ROSE (X.229) and CMISE (X.711)		
`class rose_cmip_se.cc {`		
`public:`		
`void assoc_ind ()`		
`void assoc_rel ()`		
`void p_data (const st_buffer & buffer)`		
`int rc_get_req (fcmise flag,const OID_Type&invoke_id,GetArgument&arg)`		
`int rc_get_resp (fcmise flag,const OID_Type&invoke_id,GetResult&res)`		
`int rc_event_req (fcmise flag,const OID_Type&invoke_id,EventReport&rep)`		
`int rc_event_resp (fcmise flag,const OID_Type&invoke_id,EventReport&res)`		
`int rc_create_req (fcmise flag,const OID_Type&invoke_id,CreatArgument&arg)`		
`int rc_create_resp (fcmise flag,const OID_Type&invoke_id,CreateResult&res)`		
`...`		
`}`		
Presentation (X.226)		
`class presentation_l.cc {`		
`public:`		
`void s_data_ind (onst st_buffer & buffer)`	PPDU	DT
`void s_mdap_ind (const st_buffer & buffer)`	PPDU	MDAP*
`void s_con_cnf (const st_buffer & buffer)`	PPDU	CPA
`void s_con_rf (const st_buffer & buffer)`	PPDU	CPR
`st_packet* s_con_ind(const st_buffer & buffer)`	PPDU	CP
`st_packet* s_rel_ind(const st_buffer & buffer)`	SSprimitive	SRELind
`void s_rel_cnf (const st_buffer & buffer)`	SSprimitive	SRELcnf
`void s_abort ()`	PPDU	ARP/ARU
`void p_data_req (const st_buffer & buffer)`	PSprimitive	P-DTreq
`void p_mdap_req (const st_buffer & buffer)`	PSprimitive	P-MDreq*
`void p_con_req (const st_buffer & buffer)`	PSprimitive	P-CONreq
`void p_rel_req (const st_buffer & buffer)`	PSprimitive	P-RELreq
`};`		
Session (X.225)		
`class session_l.cc {`		
`public:`		
`void t_data (const st_buffer & buffer)`	SPDU	DT
`void s_con_req (st_packet * packet)`	SSuser	SCONreq
`void s_data_req (st_packet * packet)`	SSuser	SDTreq
`void s_rel_req (st_packet * packet)`	SSuser	SRELreq
`};`		
Transport (X.224)		
`class transport_l.cc {`		
`public:`		
`void allow (int flags)`	-	-
`void t_send_req (st_packet * packet)`	TSuser	TDTreq
`void t_con_req (connector * conn)`	TSuser	TCONreq
`void t_dis_req ()`	TSuser	TDISreq
`void n_con_ind ()`	TPDU	CR
`void n_dis_ind (bool)`	TPDU/Nprovider	TDISind
`};`		

– Thirdly, the session layer provides support to the ACSE (a simplified version without synchronisation or dialogue control).

– Finally, the implementation of the transport layer varies depending on which protocol is used. The proposed system is based on TCP/IP over RS-232 transmission.

In the same way as described above, the implementation details are explained, step by step, according to the message exchange among the different layers and using implementation scheme (see Fig. 5):

Step 1. The DCC controller (agent) initiates a connection to the BCC controller (manager).

Step 2. The BCC receives a connection request. It begins accepting incoming events and passes the request up to the application layer. The BCC starts an association request to the DCC. The parameter is an encoded buffer containing the medical device service element structure, which is used for configuration and interoperability issues. Once the PDU has been generated, the request goes down through several layers, each adding its corresponding PDU header.

Step 3. The DCC receives an association request and sends a confirmation answer.

Step 4. The BCC receives the association confirmation.

Step 5. The connection procedure is ended. Now the entities exchange messages according to the ROSE/CMIP protocol. Under normal operation, the DCC sends an event report to the BCC. After that, data go down through the lower layers, similarly to how it was previously described.

```
Step 1.
stack->transport->t_con_req( conn );

Step 2.
→ transport_fsm::n_con_ind.
    allow  ( TP_READ );
    stack->application->t_con_ind();
→ application_l::t_con_ind ()
    stack->acse->assoc_req (context.buffer_encoded);
→ acse_se::assoc_req ( const st_buffer & buffer )
    make_AARQ ( buffer );
    stack->presentation->p_con_req ( tmp_buffer.made_buffer );
→ presentation_l::p_con_req ( const st_buffer & buffer )
    make_CP ( buffer );
    stack->session->s_con_req (
    make_packet_from_buffer (tmp_buffer.made_buffer) );
→ session_l::s_con_req ( st_packet * packet )
    create_CN   (packet)
    stack->transport->t_send_req( packet );
    stack->transport->allow ( TP_READ | TP_WRITE );
→ transport_fsm::t_send_req ( st_packet * packet )
    (load in buffer the sending packet to the TCP server).

Step 3.
→ transport_fsm::buffer_received ( const st_buffer & buffer )
    stack->session->t_data( buffer );
→ session_l::t_data ( const st_buffer & buffer )
    parse_pdu ();
→ session_l::s_CN ()
    st_packet*packet=stack->presentation-> s_con_ind (get_data_buffer());
    create pdu AC
    stack->transport->t_send_req( packet );
→ presentation_l::s_con_ind ( const st_buffer & buffer )
    parse_pdu (…)
    buffer = stack->acse-> p_con_ind ( user_data )
→ acse_se::p_con_ind ( const st_buffer & buffer )
    process association request
    bool do_assoc = stack->application->assoc_ind(user_data, out_buffer);
    if ( do_assoc ) {
    stack->rose_cmip->assoc_ind ();
    make_AARE ( … );
    return tmp_buffer.made_buffer;}
→ application_l::assoc_ind (const st_buffer & buffer, st_buffer &out_buffer)
    validate assotiation/negotiation data
    make response
    return true;

Step 4.
→ transport_fsm::buffer_received ( const st_buffer & buffer )
    stack->session->t_data( buffer );
→ session_l::t_data ( const st_buffer & buffer )
    parse_pdu ();
→ session_l::s_AC ()
    stack->presentation->s_con_cnf (get_data_buffer());
→ presentation_l::s_con_cnf (const st_buffer & buffer )
    stack->acse->p_con_cnf ( user_data );
→ acse_se::p_con_cnf ( const st_buffer & buffer )
    stack->application->assoc_cnf ( user_data );
→ application_l::assoc_cnf (const st_buffer & buffer)
```

Fig. 5 *X73 Implementation details*

4 EN13606 Implementation. EHR server

The proposed implementation experience is completed with the logging process of the medical device information in the EHR server, using the EN13606 standard. This information is received from the X73 gateway through the X73/EN13606 monitoring server (see Fig. 3). The client/server architecture has been implemented with middleware technologies, based on web services (WS), as it is shown in Fig. 6. The used tools have been apache axis (open source), the Servlets/JSPs container apache tomcat, extensible mark-up language (XML, WS-based framework consisting of an implementation of the SOAP server) and various utilities and APIs for generating and deploying WS applications. Data are transmitted using HTTPS protocol. In order to gain security the apache WSS4J framework (a Java library that can be used to sign and verify SOAP messages with WS-security information) was used.

4.1 Client

On the client side, a developed Java application is used to read the X73 data (received from the X73 gateway through an XML document) and generate an intermediate EN13606 (archetype file using extensible style sheet language transformations (XSLT). After that, the WS will store this information in the EHR server (see Fig. 7). A set of classes is organised in packages for giving greater modularity. A representative package (cardioweb.client) is detailed. It contains the classes to invoke the web service, obtained by means of a WSDL compiler:

– cardioweb.client.handler. This package, containing PWCallback.java class, allows setting the userkey password.
– cardioweb.client.utils. This packet contains classes with specific functionalities:

• XSLApply. It uses the Transform class and it informs if the process has been successful.
• StringUtils. It allows the client to make string operations such as filling or substitution.
• TextFile. It allows the client to write/read text files.

• Transform. It transforms XML documents using XSLT style sheets.

In order to obtain a XML document with EN13606 standard format, XSLT style sheets (used by the Xalan processor and XML Xerces parser) have been developed. The document object model (DOM) API is used to work with XML documents stored in memory and for Java API XML processing (JAXP) which makes possible XSLT transformations from the Java code. Fig. 8 shows a piece of XML code that contains an extract of EN13606 standard.

4.2 Server

On the server side, a WS has been implemented for receiving and validating the XML extract, in EN13606 format, sent by the client, using XML schemes (see Fig. 9). Once the XML extract is validated, an acknowledgment will be sent back to the client. If it was successful, the extract will be stored in the EHR server database. In the same way, as in the client implementation, classes are organised in packets achieving a greater modularity. A representative package (cardioweb.server.handler) is detailed:

– cardioweb. server.handler. It is used to verify the security requirements (usernames and passwords).
– cardioweb.server.services package. It contains the WSs available for the client:

• getExtractDevice: It creates an EN13606 extract from received data.
• WriteXMLExtract.java: It validates a XML file that contains an EN13606 extract using XML schemas. If the process is successful, it stores it in EHR database and sends a notification to the client.

– cardioweb.server.utils: It contains classes with specific functionalities:

• BDUtils: It allows the server to connect and disconnect the MySQL database.
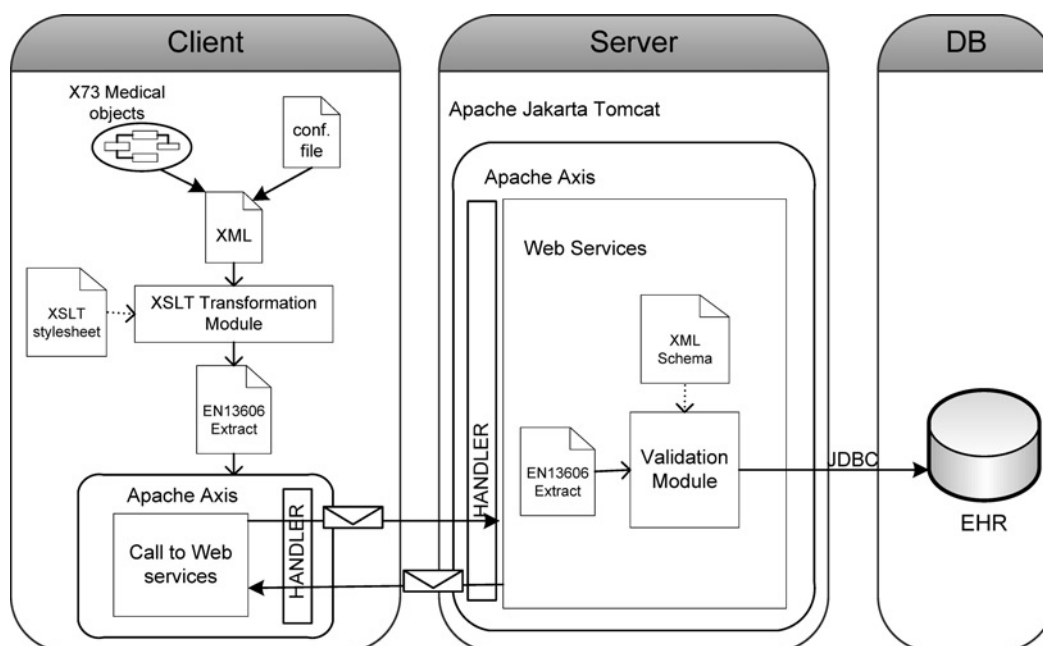• BDStoreExtract: It allows the server to record an EN13606 extract in the EHR database.

Fig. 6 *EN13606 EHR server implementation scheme*

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xalan="http://xml.apache.org/xslt"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.unizar.es/cardioweb/EN13606">
<xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes" xalan:indent-amount="4"/>
...
<ENTRY synthesised="false" sensitivity="3" uncertainty_expressed="false">
    <name originalText=" Taking arterial blood pressure ">
    <ELEMENT synthesised="false" sensitivity="3">
      <name originalText="Systolic Blood Pressure">
      <xsl:variable name="id_systolic">
        <xsl:value-of select="concat('HCE-',$hceversion,'-','2')"/>
      </xsl:variable>
      <rc_id assigningAuthorityName="UnizarCardioWeb">
        <xsl:attribute name="extension">
          <xsl:value-of select="$id_systolic"/>
        </xsl:attribute>
        <root oid="50018Z"/>
        <validTime literal="TS">
          <high time="1/1/3000"/>
          <low time="1/1/1990"/>                  </validTime>
      </rc_id>
      <meaning codeValue="271649006" codingSchemeName="SNOMED-CT"
          codingSchemeVersion="2005" displayName="Systolic Blood Pressure">
        <codingScheme oid="SNOMED-CT"/>              </meaning>
      <value>
        <PQ>
        <xsl:attribute name="value">
          <xsl:value-of select="/Results/value"/>
        </xsl:attribute>                        <property originalText="Blood pressure">
        <unit codingSchemeName="UCUM" codingSchemeVersion="1.4b" >
          <xsl:attribute name="codeValue">
            <xsl:value-of select="/Results/unit"/>
          </xsl:attribute>              <codingScheme oid="UCUM"/>
        </unit>
        </PQ>
      </value>
    </ELEMENT>
</ENTRY>
```

Fig. 7 *X73/EN13606 XSLT style sheet*

- StringUtils: It allows the server to make string operations such as filling or substitution.
- TextFile: It allows the server to write/read text files.
- ValidateXMLSchema: It allows the server to validate an XML file using XML schemas.

5 Discussion and conclusion

The adoption of a complete end-to-end standard-based solution for patient monitoring in bedside environments can be extremely useful for the integration and interoperability of the huge amounts of vital data collected every day, making a more efficient use of such information, allowing it to be shared among professionals, giving more facilities for the mobility of patients within the hospitals, reducing costs and increasing usability for patients, health care professionals and manufacturers.

The complexity and density of the documents that shape the bulk of the X73 family of standards are two of the keypoints that are restraining its adoption. From this implementation experience it was seen that X73 implementation is feasible and the standard documents are reasonably understandable; however, some guidelines could be of great help for developers that want to apply standards to a specific system. In this case, and without previous experience, the implementation of the two standards in the entire system took no more than 3 months and it was performed by a

group of five soft/hardware designers following the advice of the rest of the research group. Obviously, this is a prototype and there are still some important issues (i.e. size of the monitoring devices or battery consumption etc.) to address before a final implementation is done. Although the proposed design may appear to be complex this is completely transparent to the users (patients, doctors etc.), who only obtain its benefits (interoperability, plug-and-play functionalities, exchange of medical information etc.).

Some parts of the X73 standard are still under revision, and some layers are being adapted to improve its performance with different types of wireless transmission technologies. The X73 protocol is also being adapted to be lighter and more efficient in order to be appropriate for home and ambulatory telemonitoring devices. The EN13606 standard is also evolving and changes in the DIM are expected. These changes will lead to a redesign of the prototype presented as part of future research. The modular design adopted is expected to facilitate this task. The fact that HL7 is well established and used in many hospital information systems makes it advisable to modify the design proposed to consider an HL7 interface, co-existing with the EN13606 interoperability layer. In August 2006, the Integrating the Healthcare Enterprise initiative, with the collaboration of ISO/IEEE11073 and HL7 released the patient care device technical framework, which will be the guideline to follow in this future work.

```
<EHR_E rm_id="EN13606-1.0 v3e 2006-02-09"
  xmlns="http://www.unizar.es/cardioweb/EN13606"
  xmlns:xalan="http://xml.apache.org/xslt"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.unizar.es/cardioweb/EN13606 extract.xsd">
      <ENTRY sensitivity="3" synthesised="false" uncertainty_expressed="false">
        <name originalText=" Taking arterial blood pressure ">
          <language codeValue="us"
            codingSchemeName="ISO 639-1" codingSchemeVersion="2002">
            <codingScheme oid="ISO 639-1"/>
          </language>
          <charset codeValue="UTF-8"
            codingSchemeName="RFC 3629" codingSchemeVersion="2003">
            <codingScheme oid="RFC 3629"/>
          </charset>
        </name>
        <rc_id assigningAuthorityName="UnizarCardioWeb" extension="HCE-1-1">
          <root oid="50018Z"/>
          <validTime literal="TS">
            <high time="1/1/3000"/>
            <low time="1/1/1990"/>
          </validTime>
        </rc_id>
        <meaning codeValue="40594005"
          codingSchemeName="SNOMED-CT"
          codingSchemeVersion="2005" displayName="Taking arterial blood pressure">
          <codingScheme oid="SNOMED-CT"/>
        </meaning>
        <items>
          <ELEMENT sensitivity="3" synthesised="false">
            <name originalText=" Systolic Blood Pressure ">
            <rc_id assigningAuthorityName="UnizarCardioWeb" extension="HCE-1-2">
              <root oid="50018Z"/>
              <validTime literal="TS">
                <high time="1/1/3000"/>
                <low time="1/1/1990"/>
              </validTime>
            </rc_id>
            <meaning codeValue="271649006"
              codingSchemeName="SNOMED-CT"
              codingSchemeVersion="2005" displayName="Systolic Blood Pressure">
              <codingScheme oid="SNOMED-CT"/>
            </meaning>
            <item_category codeValue="IC01"
              codingSchemeName="EN13606-3 ITEM_CATEGORY" codingSchemeVersion="1.0">
              <codingScheme oid="000003"/>
            </item_category>
            <value>
              <PQ value="70">
                <property originalText="Blood pressure">
                  <language codeValue="us"
                  codingSchemeName="ISO 639-1" codingSchemeVersion="2002">
                  <codingScheme oid="ISO 639-1"/>
                  </language>
                  <charset codeValue="UTF-8"
                  codingSchemeName="RFC 3629" codingSchemeVersion="2003">
                  <codingScheme oid="RFC 3629"/>
                  </charset>
                </property>
                <unit codeValue="mmHg"
                  codingSchemeName="UCUM" codingSchemeVersion="1.4b">
                  <codingScheme oid="UCUM"/>
                </unit>
              </PQ>
            </value>
          </ELEMENT>
        </items>
      </ENTRY>
</EHR>
```

Fig. 8 *Extract of EN13606 standard – X73 compatible*

Home telemonitoring has experienced an important growth in the past few years, and nowadays it has proved its efficiency as a follow-up option in scenarios such as chronic disease management, home hospitalisation, follow-up after ambulatory surgery and elderly patients'

care. Mobile telehealth solutions are expanding the limits of telemonitoring, allowing patient follow-up while they carry out their daily living activities. The main barriers for a wider adoption of home telemonitoring are the difficulty of integrating telemonitoring in routine patient

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 U (http://www.xmlspy.com) by kk (kk) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.unizar.es/cardioweb/EN13606"
targetNamespace="http://www.unizar.es/cardioweb/EN13606" elementFormDefault="qualified"
attributeFormDefault="unqualified">
<xs:include schemaLocation="file:///d:/app/demographics.xsd"/>

<!-- RECORD_COMPONENT -->
  <xs:complexType name="RECORD_COMPONENT">
      <xs:sequence>
    <xs:element name="name" type="TEXT"/>
    <xs:element name="archetype_id" type="II" minOccurs="0"/>
    <xs:element name="rc_id" type="II"/>
    <xs:element name="meaning" type="CV" minOccurs="0"/>
    <xs:element name="policy_ids" type="II" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="orig_parent_ref" type="II" minOccurs="0"/>
    <xs:element name="links" type="LINK" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="feeder_audit" type="AUDIT_INFO" minOccurs="0"
        maxOccurs="unbounded"/>
      </xs:sequence>
    <xs:attribute name="synthesised" type="xs:boolean" fixed="false"/>
    <xs:attribute name="sensitivity" type="xs:integer"/>
  </xs:complexType>

<!-- ELEMENT -->
  <xs:complexType name="ELEMENT">
    <xs:complexContent>
      <xs:extension base="ITEM">
      <xs:sequence>
        <xs:element name="value">
          <xs:complexType>
            <xs:choice>
          <xs:element name="OID" type="OID" minOccurs="0"/>
          <xs:element name="CS" type="CS" minOccurs="0"/>
          <xs:element name="CV" type="CV" minOccurs="0"/>
          <xs:element name="TEXT" type="TEXT" minOccurs="0"/>
          <xs:element name="URI" type="URI" minOccurs="0"/>
          <xs:element name="TS" type="TS" minOccurs="0"/>
          <xs:element name="IVL_T" type="IVL_TS" minOccurs="0"/>
          <xs:element name="II" type="II" minOccurs="0"/>
          <xs:element name="ED" type="ED" minOccurs="0"/>
          <xs:element name="PQ" type="PQ" minOccurs="0"/>
          <xs:element name="boolean" type="xs:boolean" minOccurs="0"/>
          <xs:element name="date" type="xs:date" minOccurs="0"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
      ......
</xs:schema>
```

Fig. 9 *X73/EN13606 XML schema*

management workflow, usability and costs. The work presented can be adapted for the implementation and design of a standard end-to-end solution for medical device communications in home and ambulatory environments, having an important impact in this promising market. Standard-based telemonitoring devices are critical for the e-Health sector, as they foster competitiveness between manufacturers and help service providers in a definitive adoption of telemedicine.

6 Acknowledgment

7 References

1 Meyer, C.: 'Visions of tomorrow's ICU', *Am. J. Nurs.*, 1993, **93**, pp. 26–31

2 East, T.D.: 'Computers in the ICU: panacea or plague?', *Respir. Care*, 1992, **37**, pp. 170–180

3 Barie, P.S.: 'Advances in critical care monitoring', *Arch. Surg.*, 1997, **132**, pp. 734–739

4 Kohli-Seth, R., and Oropello, J.M.: 'The future of bedside monitoring', *Crit. Care Clin.*, 2000, **16**, pp. 557–578

5 Seiver, A.: 'Critical care computing: past, present, and future', *Crit. Care Clin.*, 2000, **16**, pp. 601–621

6 Clemmer, T.P.: 'Computers in the ICU: where we started and where we are now', *J. Crit. Care*, 2004, **19**, pp. 201–207

7 Uckert, F., Ataian, M., Görz, M., and Prokosch, H.U.: 'Functions of an electronic health record', *Int. J. Computerised Dentistry*, 2002, **5**, pp. 125–132

8 O'Toole, M.F., Kmetik, K.S., Bossley, H., Cahill, J.M., Kotsos, T.P., Schwamberger, P.A., and Bufalino, V.J.: 'Electronic health record systems: the vehicle for implementing performance measures', *Am. Heart Hospital J.*, 2005, **3**, pp. 88–93

9 Kalra, D.: 'Electronic health records: the European scene', *Br. Med. J.*, 1994, **309**, pp. 1358–1361

10 Stead, W.W., Miller, R.A., Musen, M.A., and Hersh, W.R.: 'Integration and beyond: linking information from disparate sources and into workflow', *J. Am. Med. Inf. Assoc.*, 2000, **7**, pp. 135–146

11 Pedersen, S., and Hasselbring, W.: 'Interoperability for information systems among the health service providers based on medical standards', *Informatik - Forschung Und Entwicklung*, 2004, **18**, pp. 174–188

12 Kennelly, R.J.: 'Improving acute care through use of medical device data', *Int. J. Med. Inf.*, 1998, **48**, pp. 145–149

13 Sengupta, S.: 'Heterogeneity in health care computing environments'. Proc. Annual Symp. on Computer Applications in Medical Care, 1989, pp. 355–359

14 Kling, R.: 'Learning about information technologies and social change: the contribution of social informatics', *Inf. Soc.*, 2000, **16**, pp. 217–232

15 Galarraga, M., Serrano, L., Martinez, I., and Toledo de, P.: 'Standards for medical device communication: X73 PoC-MDC', *Stud. Health Technol. Inf.*, 2006, **121**, pp. 242–256

16 'Point-of-Care Connectivity; Approved Standard – Second Edition Preview Sample Pages', 2006, available at: http://www.clsi.org/source/orders/

17 HL7 health level seven, IEEE interoperability JWG available at: http://www.ieee1073.org/related/hl7/jwg/hl7ieeeinterop.html

18 IEEE1073. Health informatics. Point-of-care medical device communication. Standard for Medical Device Communications – Overview and Framework,. available at: http://www.ieee1073.org

19 ENV13606 − CEN/TC251. Electronic Healthcare Record Communication. Parts 1, 2, 3 and 4, Pre-standard, 2000, available at: http://www.medicaltech.org

20 IEEE13734 – 'CEN/TC251. VITAL Health informatics – Vital signs information representation' available at: http://www.medicaltech.org

21 ENV13735 – 'CEN/TC251. INTERMED. Health informatics – interoperability of patient connected medical devices. [1999]'. Available at: http://www.medicaltech.orgh

22 Kennelly, R.J., and Wittenber, J.: 'New IEEE standard enables data collection for medical applications'. Proc. Annu. Symp. Comput. Appl. Med. Care, 1994, pp. 531–535

23 Kennelly, R.J.: 'IEEE standards for physical and data communications', *Biomed. Instrum. Technol.*, 1996, **30**, pp. 172–175

24 Kennelly, R.J., and Gardner, R.M.: 'Perspectives on development of IEEE 1073: the Medical Information Bus (MIB) standard', *Int. J. Clin. Monit. Comput.*, 1997, **14**, pp. 143–149

25 Susperregui, A., Paloc, C., Macia, I., and Garcia, I.: 'A standard-based mobile and wireless vital signs monitoring system', VICOMTech Association, 2005, available at: http://www.vicomtech.es/publicaciones/

26 Warren, S., Craft, R.L., Parks, R.C., Gallagher, L.K., Garcia, R.J., and Funkrouser, D.R.: 'Proposed information architecture for telehealth system interoperability'. Annual Int. Conf. IEEE Engineering in Medicine and Biology − Proc., 1999, vol. 2, p. 702

27 Yao, J., and Warren, S.: 'Applying the ISO/IEEE 11073 standards to wearable home health monitoring systems', *J. Clin. Monit. Comput.*, 2005, **19**, pp. 427–436

28 Anagnostaki, A.P., Pavlopoulos, S., Kyriakou, E., and Koutsouris, D.: 'A novel codification scheme based on the "VITAL" and "DICOM" standards for telemedicine applications', *Biomed. Eng., IEEE Trans.*, 2002, **49**, pp. 1399–1411

29 Weigand, C.: 'VITAL: use and implementation of a medical communication standard in practice', 2005, pp. 319 − 322

30 Pavlopoulos, S., Anagnostaki, A., Koutsouris, D., Lymberis, A., Levene, P., Reynolds, M., Georgiadis, N., Lambrinoudakis, C., and Gritzalis, D.: 'Vital signs monitoring from home with open systems', *Stud. Health Technol. Inf.*, 2000, **77**, pp. 1141–1145

31 Zywietz, C., Kraemer, M., Fischer, R., and Widiger, B.: 'Integrating the ECG enterprise-HES-EKG with the built-in vital signs information nomenclature', *Comput. Cardiol.*, 2004, **31**, pp. 41–44

32 Kampmann, J., Lau, G., Kropp, St., Schwarzer, E., and Hernandez, S.C.: 'ER-Connection of electronic medical devices in ICU according to the standard "MIB" ', *J. Clin. Monit. Comput.*, 1991, **V8**, pp. 163–166

33 Salvador, C.H., Pulido, N., Quiles, J.A., and Gonzalez, M.A.: 'An implementation of the IEEE medical information bus standard', *IEEE Eng. Med. Biol. Mag.*, 1993, **12**, pp. 81–88

34 Lebak, J.W., Yao, J., and Warren, S.: 'Implementation of a standards-based pulse oximeter on a wearable, embedded platform'. Annual Int. Conf. IEEE Engineering in Medicine and Biology – Proc., 2003, vol. 4, pp. 3196–3198

35 Warren, S., Yao, J., Schmitz, R., and Lebak, J.: 'Reconfigurable point-of-care systems designed with interoperability standards'. Annual Int. Conf. IEEE Engineering in Medicine and Biology – Proc., 2004, vol. 26 V, pp. 3270–3273

36 'Continua Alliance web site', 2006, available at: http://www.continuaalliance.org/home/

37 'Bluetooth SIG Medical Devices WG', 2006, available at: http://www.bluetooth.com/Bluetooth/SIG/

38 Schadow, G., Föhring, U., and Tolxdorff, T.: 'Implementing HL7: from the standard's specification to production application', *Methods Inf. Med.*, 1998, **37**, pp. 119–123

39 Ohe, K., and Kaihara, S.: 'Implementation of HL7 to client-server hospital information system (HIS) in the university of Tokyo hospital', *J. Med. Syst.*, 1996, **20**, pp. 197–205

40 Muñoz, A., Somolinos, R., Pascual, M., Fragua, J.A., Gonzalez, M.A., Monteagudo, J.L., and Salvador, C.H.: 'Proof-of-concept design and development of an EN13606-based electronic health care record service', *J. Am. Med. Inf. Assoc.*, 2007, **14**, pp. 118–129