iCARDEA

"An Intelligent Platform for Personalized Remote Monitoring of the Cardiac Patients with Electronic Implant Devices"

SPECIFIC TARGETED RESEARCH PROJECT

PRIORITY Objective ICT-2009.5.1: Personal Health Systems - a) Minimally invasive systems and ICT-enabled artificial organs: a1) Cardiovascular diseases

iCARDEA D6.3.1 Interoperability Infrastructure for Personal Health Records

Due Date:	November 30, 2011
Actual Submission Date:	December 02, 2011
Project Dates:	Project Start Date : February 01, 2010 Project End Date : January 31, 2013 Project Duration : 36 months
Leading Contractor Organization:	SRFG

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination Level		
PU	Public	x
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document History:

Version	Date	Changes	From	Review
V0.1	November 7, 2011	Initial Document	SRFG	All partners
V0.2	November 29, 2011	Draft deliverable	SRFG	SRDC
V0.3	November 30, 2011	SRDC Comments	SRDC	SRFG
V0.4	November 30, 2011	SRFG updates	SRFG	SRDC
V0.5	December 01, 2011	SRDC Comments	SRDC	SRFG
V0.6	December 01, 2011	Update in for figure 1 (EHR block text), accept all the changes and correction for typos.	SRFG	All partners
V0.7	December 02, 2011	Small corrections.	SRFG	OFFIS

Table of contents

1 Introduction 4
1.1 Purpose 4
1.2 Definitions, acronyms and abbreviations 4
1.3 Basic idea about Semantic Interoperability 4
2 Technological Approach 5
2.1 Overview 5
2.2 Introduction to the PHRS Interoperability Layer 6
2.2.1 RDF and interoperability 7
2.2.2 RDF Serialization 7
2.2.3 More Support for RDF by RDFS 8
2.2.4 Why not OWL? 8
2.2.5 Why not NoSQL? 8
2.2.6 Design Decisions 9
2.3 Communications and Protocols 9
2.3.1 Interoperability Standards - Message Profile 9
2.3.2 Medical Taxonomies 10
2.3.3 Service Layer and Message Exchange Pattern 11
3 Architectural Overview 11
3.1 The PCC09/PCC10 SOAP based Web Services Tier 11
3.2 The Persistence Driver Clients 12
3.3 The Persistence Driver tier 13
3.4 RDF Store tier 13
3.5 Functional overview 13
4 PHR System role in the iCARDEA 14
5 PHRS Interoperability (drone based) Demonstration 15
5.1 Vital Signs 15
5.2 Medications 16
5.3 Problems and Symptoms 17
5.4 Risk Factors 18
5.5 Observation of Daily Living 19
6 Annex – PHRS Data Model Specification 21
6.1 Vital Signs 21
6.2 Observation 26
6.3 Medications 30

1 Introduction

1.1 Purpose

This document describes the main principles and concepts related with the interoperability layer from the PHR System point of view. The interoperability layer allows and enables the PHR System to get involved into meaningful information exchange with the other components in the iCARDEA project and basically potentially also with other Personal Health Applications.

1.2 Definitions, acronyms and abbreviations

Abbreviation/Acronym	DEFINITION
HIS	Hospital information System
PHR	Personal Health System
RDF	Resource Definition Framework
RDFS	Resource Description Framework Scheme
OWL	Web Ontology Language
XML	Extensible Markup Language
IHE	Integrating the Healthcare Enterprise
PCC	Patient Care Coordination
RIM	Reference Information Model
SOAP	Simple Object Access Protocol
WSDL	Web Services Description Language
MEP	Message Exchange Pattern
SKOS	Simple Knowledge Organization System

Table 1 List of Abbreviations and Acronyms

1.3 Basic idea about Semantic Interoperability

One of the most intriguing problems related with the communication between one or more parts (named in the literature communicating entities) is the exchange from meaningful information. The semantic interoperability *"discipline"* tries to address this kind of problems and tries to provide suitable solution/answer for these problems.

There are (at least) three approaches for semantic interoperability:

- On an ontological level – each part involved in the communication will use the same ontology. This kind of communication increases the degree of accuracy.
- On a vocabulary level – each part uses the same (controlled) vocabulary. This kind of communication is less accurate than the "ontological level") because the

relations between the (controlled vocabulary) concepts cannot be meaningfully described by using the vocabulary related tools.

- On a model level – each part uses the same model. This is the least performant type of communication. It can introduce the greater ambiguity because there is no agreement about the vocabulary and/or the relation between the vocabulary concepts.

The PHR System uses a combination between the first and the second cases (ontological and vocabulary level). The next section will provide more details focused on different perspectives (e.g. technical perspectives).

2 Technological Approach

2.1 Overview

The PHR System consists of two main parts, each of this part can coexist independent of the other one. Both can communicate with in different protocols (e.g. plain API, web services, etc.). This architecture makes any of the part interchangeable; this aspect allows other health care provider to be integrated without to follow the IHE-CM profiles. The actual implementation does not try to replace the IHE-CM profiles it tries to make them easy available for different providers with less effort as possible. For example: a custom application (named *"Health Care Provider"* in diagram below) that is running on a portable device specialized only in measuring the body weight (for a given patient) can be integrated even if it can not produce SOAP messages required by the IHE-CM.

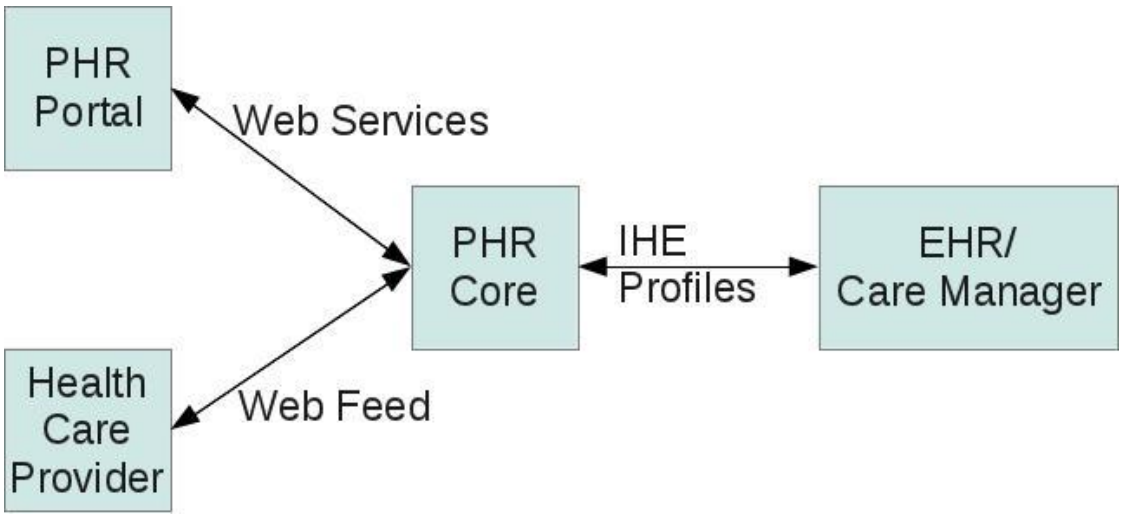

Figure 1 - PHR System Overview

2.2 Introduction to the PHRS Interoperability Layer

The PHR System uses the RDF and RDFS mechanism as the core of its persistence mechanism to define to address interoperability related problems. Below is a list with RDF related advantages:

- RDF (and RFDS) is a flexible (meta) model that allows defining different eHealth aspects with an increase degree of accuracy.
- RDF provides the *"Open close principle"* – this allows keeping the model (or the code) open for any extension but closed for modification.
- A lot of already defined ontologies can be used. These ontologies can be related with eHealth taxonomies (e.g. SNOMED[1]) or with general concepts (e.g. SKOS[2]).
- RDF allows combining information from different sources. This feature emphasizes modularity because individual modules can be built and tested and combined later in to composite (and unified) model.
- RDF builds graphs; graph can be involved in more complex and accurate queries than the one build for the relational model. The relational model is still an efficient way to express and work with (data) models but it tends to increase the system complexity when using complex queries (e.g. with a lot of joins statements). With the RDF graph complex queries can easily formulated.
- There are several products/framework able to manage and manipulate RDF information. Those products/frameworks are known as *"RDF Stores"* or *"Triplestores"*. Most of these products reach a certain level of maturity and they provide an accurate documentation community support.

With all of this upper-listed advantages RDF brings also some disadvantages as well. Below are some of them:

- RDF tends to increase the level of complexity. If the modes (and or the ontologies) are increasing in complexity the RDF files tend to be unreadable for human reader. From this perspective changes and maintenance become a very delicate task.
- Compared with the relation model – the performance may be lower, especially for simple operations.
- Scalability, code of practice demonstrates that the relational model tends to scale better.
- The RDF model is only able to define graphs without features like class or class properties. The typing (classes) feature can be very useful during data modeling. This disadvantage was eliminating by introducing the RDFS; more about this theme in the next sections.
- Increasing number of existing models. One of the major problems for the semantic web (and as well for RDF) is that everyone who publishes Semantic Web data is free to create its own RDF schemas for storing data. In this way there are a lot

[1] http://en.wikipedia.org/wiki/SNOMED_CT
[2] http://www.w3.org/2004/02/skos/

of already existing models that serve the same purpose but in the different way. This aspect may make the decisional process very difficult and error prone.

Note about performance: This does not mean that the triple stores are slow and unscalable; there are a lot of projects (products) able to prove the contrary (e.g. AllegroGraph[3]) but most of these products provide proprietary solution and in this case the portability is sacrificed for performance.

2.2.1 RDF and interoperability

The RDF meta-model follows very easy rules and it does not constrain the user to any other specific rules (or standards). The RDF meta-model are generic and they can be use to express various information(s), the RDF information produced in this way can be consumed by any RDF complaint consumer. This specific feature solves the model level interoperability (defined in the section 1.3). For this reason as the data model of the persistence store of the PHRS Interoperability system RDF is chosen. In other words, the updated clinical data shared by the PHR Portal will be persisted in an RDF based persistence store in PHRS Interoperability System. On top of this, to share clinical data with other entities in the iCARDEA architecture such as EHR system or the Care Plan Engine which acts as a care manager, we need to define a common vocabulary and to apply it to our information; the vocabulary can be local one or can be based on an already existing international standard. At this level the already existing interoperability gets more accurate because of the common vocabulary. If interoperability degree is still not enough we can define relation and constrains between the information(s) and apply them to our RDF model. The ontologies can exist separate and they can be applied to any existent RDF model. At this level we achieve the highest degree of interoperability because: we use the same model, with the same vocabulary and same ontologies.

The PHR System can represent information in RDF(S) form to other external entities in iCARDEA Architecture, this RDF(S) information can be materialized in any suitable form; one of this form is the content models defined by IHE CM profiles. In the iCARDEA project the PHR System uses IHE CM profiles to exchange meaningful information with all the other involved parts. The reason why the PHR System does not directly represents its information only like IHE CM content models is because in this way it will not be bound to certain set of rules and constrains specific to a profile (or standard), and it will be possible to share the clinical information later based on future interoperability specifications. For this reason, the PHR System is able to consume and produce IHE CM profile conformant clinical content but it internally represents it as an RDF graph.

2.2.2 RDF Serialization

RDF is just a specification an RDF model. It does not have a default manifestation form. There are several ways to manifest RDF data. Some of the most popular are N3 and XML.

[3] http://www.franz.com/agraph/allegrograph/

For this implementation XML form was chosen. The reason for this is beside the XML popularity the validation feature (it allows to detect syntactical errors) followed by the ability to create specific meta-languages that may help in the data modelling process. Most of the existing RDF stores are able to consume RDF data in XML form.

2.2.3 More Support for RDF by RDFS

In the upper section the RDF was briefly introduced. Additionally, it was mentioned that RDF requires classes and properties to increase the data modeling expressivity. These missing features are provided by RDFS. RDFS supports the definition of classes and properties based on set inclusion. In RDFS classes and properties are orthogonal. RDFS is encoded as RDF by using the same syntax – this means that the RDFS solution can be integrated without any additional effort.

2.2.4 Why not OWL?

OWL provides more and better modeling features than RDFS and the similar to the RDFS it uses the same syntax like RDF – no extra effort for integration. The RDFS modeling features are sufficient for the current implementation. If future requirements will require elaborate (data) modeling abilities the OWL can be adopted without any additional effort.

2.2.5 Why not NoSQL[4]?

The NoSQL initiative gains more and more popularity and together with the RDF(s) stores, it provides a professional alternative for the classical relational (SQL) storages. The main NoSQL premise is to get rid of the static models provided in the classical relational (SQL) word and to replace this with dynamic models. The NoSQL solutions are quite preformat and tend to scale pretty well.
From the point of view of the storage strategy the NoSQL world can be divided in two main categories:

1. Document-based systems (e.g. CouchDB[5] and MongoDB[6])
2. Key/value systems (e.g., Redis[7], MemcacheDB[8], SimpleDB[9], Voldemort[10], Dynamo1[11], Big Table[12], and Linda [13]style tuple stores)

[4] http://en.wikipedia.org/wiki/NoSQL
[5] http://couchdb.apache.org/
[6] http://www.mongodb.org/
[7] http://redis.io/
[8] http://memcachedb.org/
[9] http://en.wikipedia.org/wiki/Amazon_SimpleDB
[10] http://project-voldemort.com/
[11] http://en.wikipedia.org/wiki/Dynamo
[12] http://en.wikipedia.org/wiki/BigTable
[13] http://en.wikipedia.org/wiki/Linda_%28coordination_language%29

The main problem here and the reason why the RDF(S) was favored as the "dynamical" aspect of the model, each product is using a property solution for the data model, choosing one of the upper listed solution will bound the PHR System to a singular technology and will reduce the system portability.

2.2.6 Design Decisions

The PHR System uses RDFS together with a RDF store (triplestore). Because the actual RDF store scene provides various products (able to store and manipulate RDF information) and because each product uses its own proprietary interface SRFG chooses to unify the access by using a wrapper. The wrapper uses the Strategy pattern and encapsulates the all the specific (proprietary) behavior and makes it interchangeable. Because the user must be able to select/change the underplayed RDF storage the actual implementation uses the Abstract Factory/ Factory patters pair to encapsulate all the constructional related details. In this way each RDF store has his own Factory (encapsulating all the specific constructional) and user can choose the needed wrapper with a given configuration via the Abstract Factory.

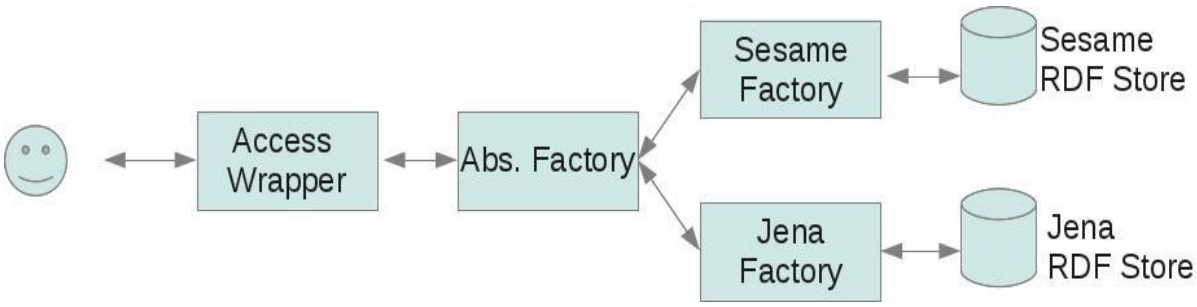

Figure 2 - Persistence Driver

The PHR System uses RDFS technology (to model data) and it can uses different RDF storages. The next section will describe how the data models can be transmitted and received (consumed).

2.3 Communications and Protocols

2.3.1 Interoperability Standards - Message Profile

The previous section describes how the models are built and stored but it does not reveal how the model can be sent or received (consumed).
According to the official IHE web site: *"**IHE** is an initiative by healthcare professionals and industry to improve the way computer systems in healthcare share information. IHE promotes the coordinated use of established standards such as DICOM and HL7 to address specific clinical need in support of optimal patient care. Systems developed in ac-*

cordance with IHE communicate with one another better, are easier to implement, and enable care providers to use information more effectively."
One of the IHE initiative are the profiles (and standards): *"**IHE Profiles** organize and leverage the integration capabilities that can be achieved by coordinated implementation of communication standards, such as DICOM, HL7 W3C and security standards. They provide precise definitions of how standards can be implemented to meet specific clinical needs."* This definition is originates from the official IHE web site.
The actual implementation of PHRS uses IHE [14]profiles PCC9 [15]and PCC10 [16]to send and to receive health related information.
According with the *"Technical Framework volume 2 revision 6.0"*[17] a **Patient Care Coordination PCC profile** is *"a set of functional components of healthcare enterprise and healthcare information networks, called IHE actors and specifies their interactions in terms of a set of coordinated, standards based transactions"*.[18]
The IHE provides several transactions (e.g. PCC1, PCC2, etc.). PCC1 message is the main/principal message for the entire PCC message set all the other messages are derived from the PCC1 message. iCARDEA and hence, the PHRS support PCC9 and PCC10 profiles. More precisely iCARDEA can produce and consume PCC9 and PCC10 messages in a meaningful way.
The PCC9/PCC10 tandem uses a publish-subscribe message pattern. More precisely the PCC9 registers a request and the PCC10 is the response every time when changes of interest are occurring. The content of the response (PCC10) can be customized with specific codes – named *core provision codes* - this code(s) are provided in the PCC9 request. If the PCC9 does not contain any core provision code than all the available information (e.g. medication, vital signs, etc.) will be involved. Beside the *"provision codes"* upper described the PCC9 request may contain also patient(s) related information.
One of the biggest advantages for the IHE profiles (especially for the PCC9 and PCC10) is the underlying meta-model named **RIM** (HL7 Reference Information Model). This meta-model was specially designed for eHealth applications and contains all the required aspects necessary to describe accurate the health related aspects required in the iCARDEA project.

2.3.2 Medical Taxonomies

The previously mentioned profiles describe how information must be structured and which actors are involved in the information exchange (transaction) but it does not specify the coding systems for annotating the clinical entries. Several different clinical terminology systems can be used for this purpose. The PHRS System uses the following terminology systems:

- LOINC for vital signs.
- SNOMED for problems, risk factors, observation, activities of daily living, etc.

[14] http://www.ihe.net/
[15] http://wiki.ihe.net/index.php?title=PCC-9
[16] http://wiki.ihe.net/index.php?title=PCC-10
[17] *www.ihe.net/.../IHE_PCC_TF_Rev7-0_**Vol_2**_2011-09-09.pdf*
[18] *IHE Patient Care Coordination (PCC) - Technical Framework volume 2 revision 6.0 page 9*

- UMLS for most of the concepts. This allows PHRS to map different taxonomies to a central uniform accepted meta-taxonomy.

2.3.3 Service Layer and Message Exchange Pattern

IHE suggests SOAP based web services, and provides a set of WSDL artifacts for building these services. Reasons for choosing SOAP are described as follows:

- SOAP defines protocol able to sustain exchange of structural information between two or more web services (parts).
- The SOAP based web services can be described (in a platform independent) way by using the WSDL language. This makes the web service declaration independent programming language.
- SOAP uses MEP to describe how two different parts of a message passing system are connected and can communicate with each other (e.g. no replay expected, way for answer, etc).
- SOAP is transport neural can use any underlying transport protocol (e.g. HTTP, SMTP, etc).
- The JDK provides out of the box libraries and is able to publish and consume SOAP based web services and hence, no other tools apart from JDK are needed. Detailed information can be found in the Metro project[19].

By using the WSDL files based on IHE, the PHRS System is able to produce standard web services able to produce and consume PCC messages. A lot of tools / frameworks are able to generate web services for a given set of artifacts. The PHRS system uses Apache Axis2 to produce and consume SOAP based services. More precisely, the PHRS System provides end points able to get involved in PCC09 / PCC10 transactions. The PHRS System provides also a set of test purposed tools able generate PCC09/PCC10 messages based on a given context on a generated one. These tools are also able to render the messages in human readable form. The section named *"PHRS Interoperability (drone based) Demonstration"* will present a suite of scenarios where the test tools are involved.

3 Architectural Overview

The PHRS core follows a four-tier architecture.

1. The PCC09/PCC10 SOAP based web services tier
2. The persistence driver clients
3. The persistence driver
4. RDF Store

3.1 The PCC09/PCC10 SOAP based Web Services Tier

This layer provides implemented SOAP based services. The web service structure (exposed methods, methods arguments, method exception, etc.) is generated based on the WSDL archetypes provided by the IHE. The IHE provides WSDL archetypes for most of

[19] http://metro.java.net/discover/

the PCC messages; this implementation is using only the PCC09 and PPC10. This layer cannot manage and manipulate data; this layer is only offered in a specific way (SOAP messages) functionality from the underlying layers. This layer also serializes the SOAP messages into files (or string) and can also consume serialized messages. This feature allows building and testing each service separately.

The section, PHRS Interoperability (drone based) Demonstration, describes some real life scenarios. These scenarios are executed together with some specialized programs named drones. The test is using two kinds of drones PCC9 and PCC10, each drone is specialized in a PCC message.

3.2 The Persistence Driver Clients

This layer defines the domain objects and uses them to define the PHR System business. In the same way as the web service tier this tier cannot store any data but it relays on the persistence abilities from its underlying layers. This layer contains:

1. Vocabulary Client – provides manage and manipulation abilities for vocabulary concepts. This client allows us also to do the conversion between one or more code systems for individual vocabulary items. The supported vocabularies SNOMED, LOINC, our own vocabulary. The vocabulary client allows us to organize vocabulary concepts using SKOS rules. The vocabulary client allows also tagging for the vocabulary concepts; a vocabulary concept can be applied like "tag" to another vocabulary concept – this feature allows us to enrich the SKOS order related features. The vocabularies and tags can be loaded from one or more RDF files.
2. Observation Client contains all the logic related to the IHE Observation concept. The Observation model can be used to manage and manipulate: Activity of daily living, Risks and Problems.
3. Vital Sign Client contains all the logic related to the IHE Vital Sign concept. This client uses LOINC code system. Because of the flexible model the Vital sign client can use also other code systems.
4. Medication Client contains all the logic related with the IHE Medication concept.

Healthcare applications, such as the PHR Portal, use these clients to exchange information with the interoperability layer. The information gathered from the Observation Client, Vital Sign Client and Medication Client is persisted and used to generate PCC10 responses. The PCC10 message is send every time when a change of interest occurs.

Chapter: *"Annex – PHRS Data Model Specification"* contains the model definition for the upper mentioned methods together with some examples.

3.3 The Persistence Driver tier

The Persistence Driver provides unified access to different RDF stores. The tier interacts directly with the real RDF Store. It does not contain models. All the models were already defined in the "client layer". This tier can be configured via an XML file.
The persistence driver can notify any changes in its state.

3.4 RDF Store tier

This is the RDF store and it is vendor specific. It interacts physically with the resources (hard disk, databases, databases files, etc). The actual implementation provides supports only two RDF stores (openRDF) Sesame [20]and Jena[21].

3.5 Functional overview

The persistence driver tier can publish changes in its state to any registered subscriber (publish-subscriber design pattern). Every time when the web services tier get a request (PCC9) from a client it registers a subscriber, the subscriber is customized based on the information from the request (care provision code), see Figure 3 - PCC9 Request for details. Changes in the data are notified back to the web services tier, here the SOAP message is build (based on the notified data) and send back to the client. The web services tier uses an Abstract Factory to build a meaningful SOAP messages (PCC10). This abstract factory is delegating the request to more specialized Factories that encapsulate the creational details for each particular message part (e.g. medication factory, vital sign factory, etc.). Figure 4 - PCC10 Response displays the workflow in a graphical way (the number on the left side represents the workflow step).
The client address (end point) is obtained from the SOAP message header.

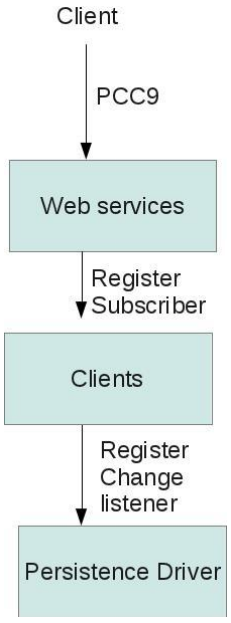

Figure 3 - PCC9 Request

[20] http://www.openrdf.org/
[21] http://jena.sourceforge.net/

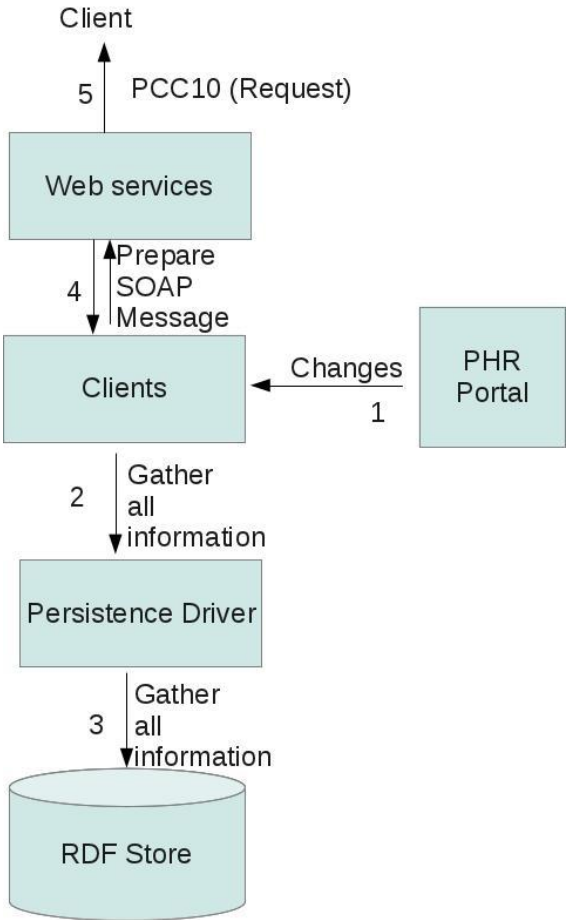

Figure 4 - PCC10 Response

4 PHR System role in the iCARDEA

The PHR System plays two roles in the iCARDEA project:

- "Clinical Information Data Source" for the communication with the *"Care Planner"* component.
- Data Consumer for the communication with the *"EHR"* system.

For the communication with the *"Care Planner"* system the PHR System is using the IHE CM profiles.

For the communication with the EHR system the PHR System is using also IHE CM profiles. More precisely, the PHR System sends a PCC9 message to the "EHR" system and it becomes all the need information like PCC10. The information from the PCC10 response can be customized via the care provision code. This can be used for example to retrieve the discharge medications of a patient from the EHR System, so that the patient can present his/her medication compliance easily from the PHR Portal.

In summary, the PHR System can play two roles – data source and data consumer.

5 PHRS Interoperability (drone based) Demonstration

The PHRS is able to register PCC09 queries and it is able to inform (publish) changes in PCC10 transactions.

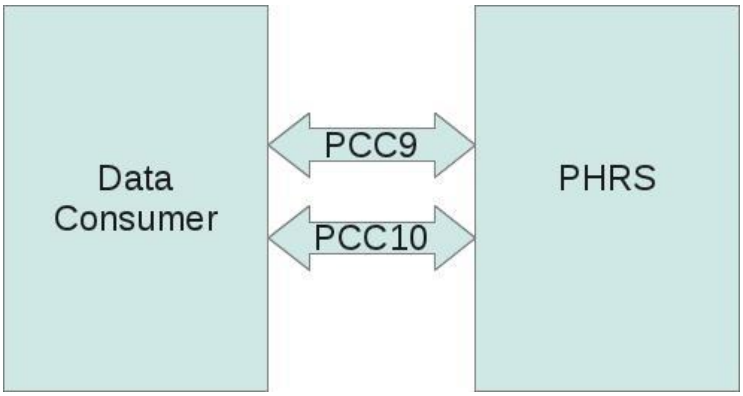

Figure 5 - General Data Flow

The following screenshots show how the PHRS interact due the PCC09 / PCC10 transactions for different scenarios. The scenario below is based on specialized programs named drone, each drone is specialized on a singular message type (PCC9 or PCC10). A drone can build and send a message (to a given end point) and it can read and render a message in to a readable form.

5.1 Vital Signs

In this scenario the PHRS is queried for "*Vital Signs*" (care provision code: "*COBSCAT*"). The PCC09 Drone intercepts this query and presents it in a human readable form.

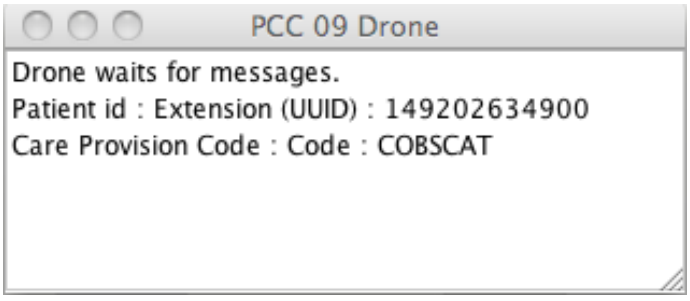

Figure 6 - PCC09 (Vital Signs) request

After the query is registered, and if the required information is available, a PCC10 transaction is started. The information about (the other) transaction end point is extract from the previous PCC09 message and contains all the vital signs for a given patient

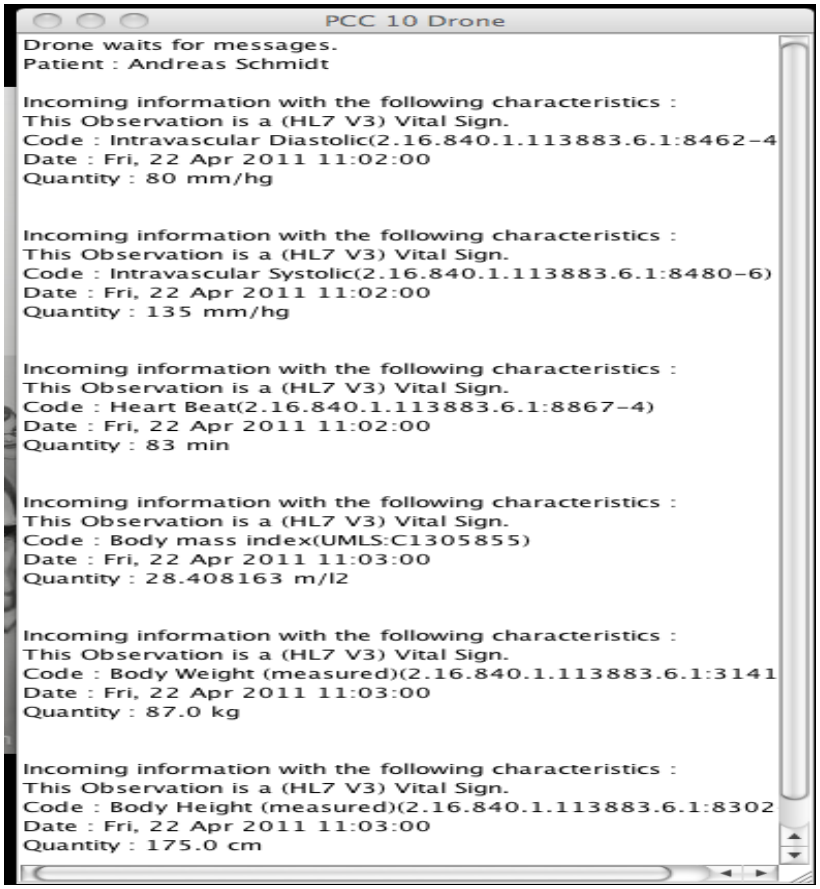

Figure 7 - PCC10 response with vital signs

5.2 Medications

In this scenario, the PHRS is queried for "Medications" (care provision code: "*MEDLIST*"). The PCC09 Drone intercepts this query and presents it in a human readable form.

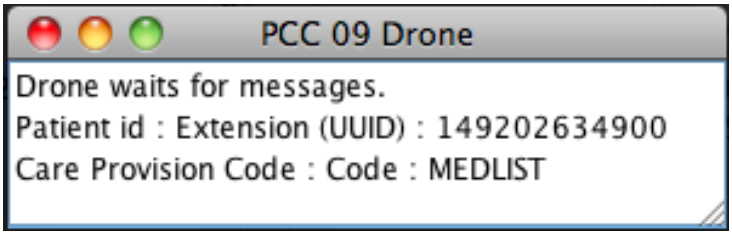

Figure 8 - PCC09 Medications request

After the query is registered, and if the required information is available, a PCC10 transaction is started. The information about the transaction end point is extract from the previous PCC09 message and contains all the Medications for a given patient.

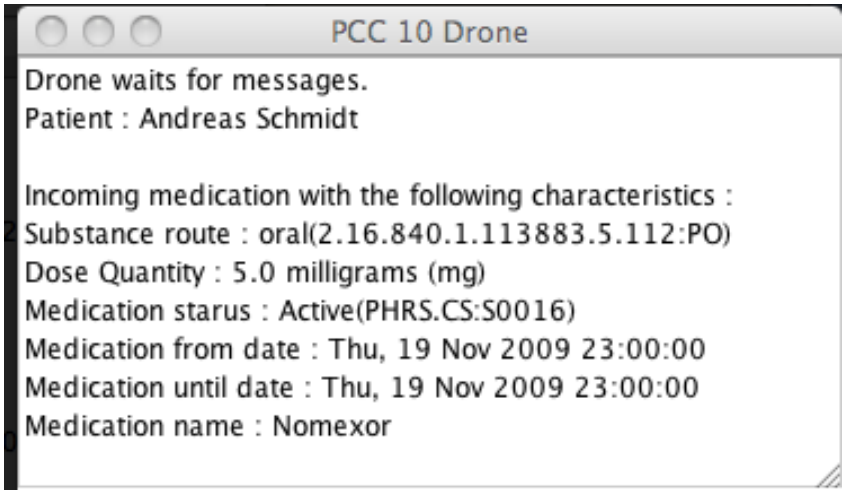

Figure 9 - PCC10 response for medication

5.3 Problems and Symptoms

In this scenario, the PHRS is queried for "Problems and Symptoms" (care provision code: "*MEDCCAT*").
The PCC09 Drone intercepts this query and presents it in a human readable form.

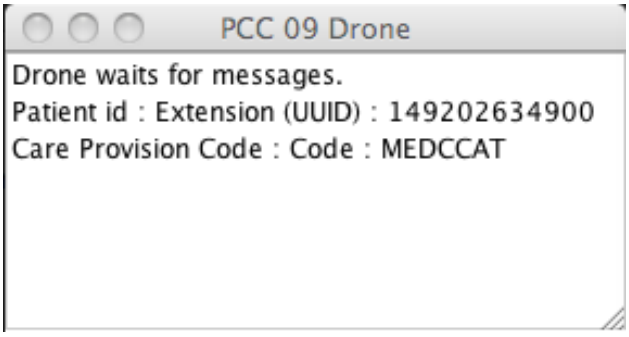

Figure 10- PHRS PCC09 problems (and symptoms) request

After the query is registered and if the required information is available a PCC10 transaction is started. The information about the transaction end point is extract from the previous PCC09 message and contains all the Problems and Symptoms for a given patient.

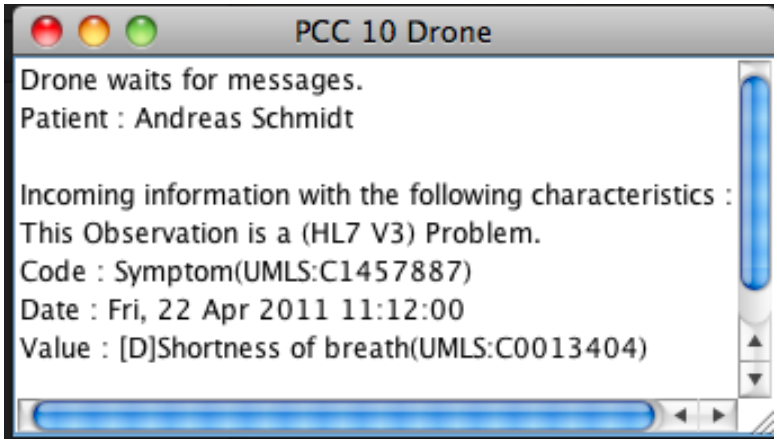

Figure 11 - PHRS PCC10 response for problems (and symptoms)

5.4 Risk Factors

In this scenario, the PHRS is queried for "Risk Factors" (care provision code: "*RISK-LIST*").
The PCC09 Drone intercepts this query and presents it in to a human readable form.

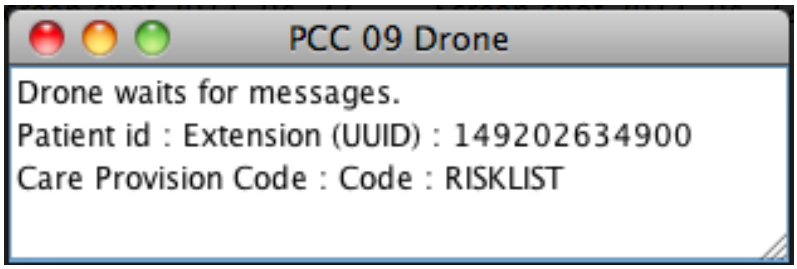

Figure 12- PHRS PCC09 Risk Factors request

After the query is registered, and if the required information is available, a PCC10 transaction is started. The information about transaction end point is extract from the previous PCC09 message.

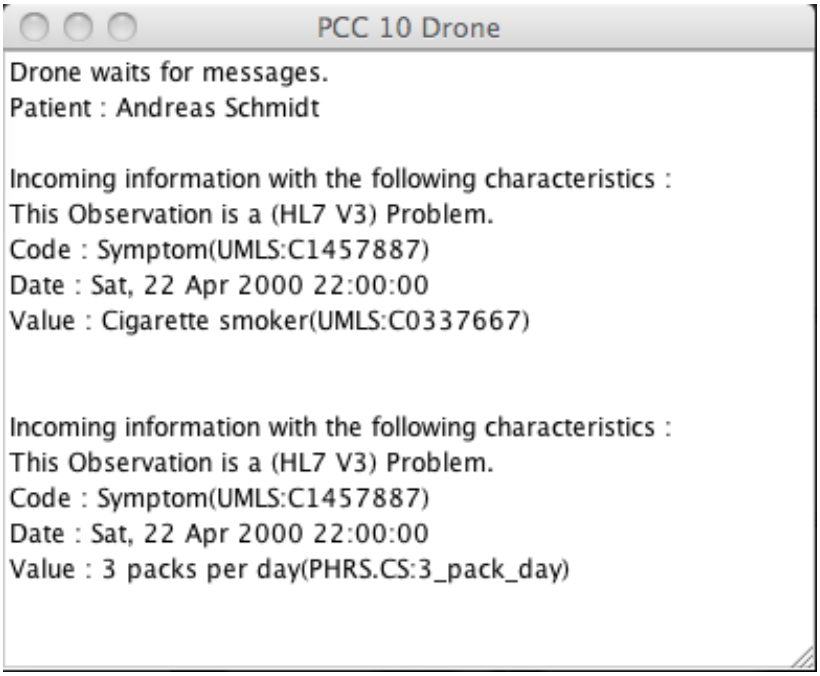

Figure 13 - PHRS PCC10 response for risk factors

5.5 Observation of Daily Living

In this scenario the PHRS is queried for "Observation of Daily Living" (care provision code: "*ODLS*").
The PCC09 Drone intercepts this query and presents it in to a human readable form.

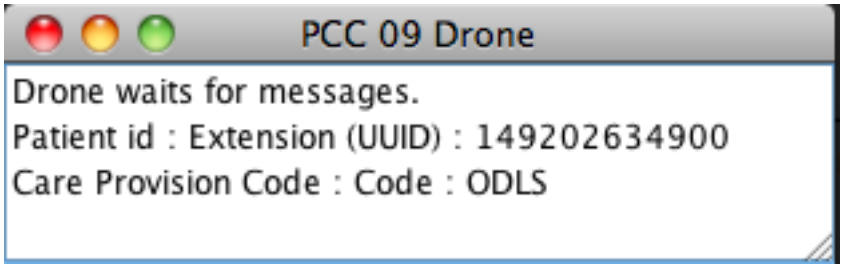

Figure 14 - PHRS PCC09 (Observation of Daily Living) request

After the query, is registered and if the required information is available, a PCC10 transaction is started. The information about the transaction end point is extracted from the previous PCC09 message and contains all the Observation of Daily Living information for a given patient. The relevant PCC properties are shown in Table 2**Error! Reference source not found.**.

Table 2 - PCC Properties Observations of Daily Living

name-space	local name	type	value	cardinality	manda-tory	Description	Introduced By
http://www.icardea.at/phrs/hl7V3#	tem-plIdRoot	literal	1.3.6.1.4.1.19376.1.5.3.1.4.13	many values	yes	All the tem-plIdRoot tem-pleate id allows me to identify the (involved) resource like a “Vital Sign”	Back-end
http://www.icardea.at/phrs/hl7V3#	tem-plIdRoot	literal	1.3.6.1.4.1.19376.1.5.3.1.4.13.2	many values	yes	5.5.1.1	Back-end
http://www.icardea.at/phrs/hl7V3#	tem-plIdRoot	literal	2.16.840.1.113883.10.20.1.31	many values	yes	5.5.1.2	Back-end
http://www.icardea.at/phrs/hl7V3#	code	re-source	NA there are already prede-fined instances for status e.g. Sys-tolicBlood Pressure) 5.5.1.3	only one	yes	describes the vital sign (e.g. the Systolic Blood Pressure)	Back-end
http://www.w3.org/2004/02/skos/core#	note	literal	NA (the value is a string - try to keep it short)	only one	no	attaches a note to the involved vital sign	Back-end
http://www.icardea.at/phrs/hl7V3#	status	re-source	NA (there are already prede-fined instances for status e.g. com-plete)	only one	yes	describes the vital sign status.	Back-end
http://www.icardea.at/phrs/hl7V3#	effec-tiveTime	literal it must follow the	NA	only one	yes	describes the time when the date when the vital sign was	Back-end

		format yyy-ymmddhhss (e.g. 201006010000)				registered.	
http://www.icardea.at/phrs/hl7V3#	value	literal (e.g. 100)	NA	only one	yes	describes the quantity for involved vital sign	Back-end
http://www.icardea.at/phrs/hl7V3#	unit	re-source	NA (there are already prede-fined instances for unit e.g. mmHG)	only one	yes	describes the measure unit for the vital sign value	Back-end

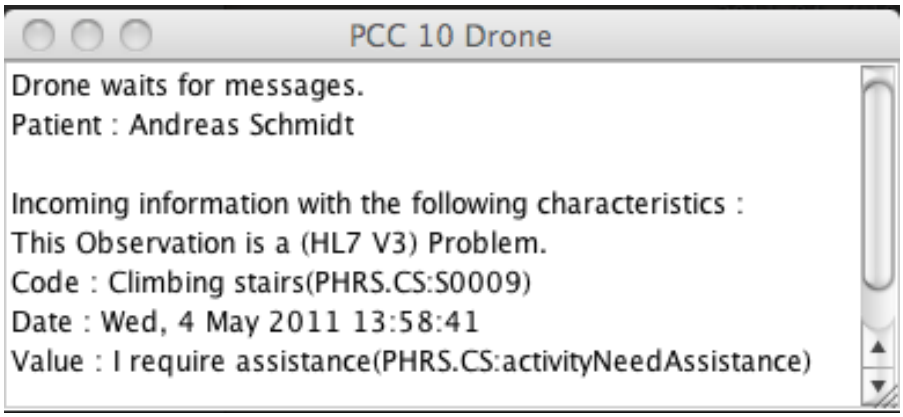

Figure 15 - PHRS PCC10 response for Observation of Daily Living.

6 Annex – PHRS Data Model Specification

The iCARDEA PHRS can generate PCC09/10 messages, in order to generate meaning-ful PCC10 messages every resource must contain at least the following properties.

6.1 Vital Signs

For a mean-fully Vital Sign PCC10 message the involved resource must contains at least the following properties.

The following is a XML/RDF snippet for a vital sign:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
       xmlns:skos="http://www.w3.org/2004/02/skos/core#"
       xmlns:icady="http://www.icardea.at/phrs/"
       xmlns:icadyCode="http://www.icardea.at/phrs/code#"
       xmlns:icadyCodeSystem="http://www.icardea.at/phrs/codeSystem#"
       xmlns:icadyStatus="http://www.icardea.at/phrs/status#"
       xmlns:icadyBloodPreasure="http://www.icardea.at/phrs/status#"
       xmlns:icadyHL7V3="http://www.icardea.at/phrs/hl7V3#">

       <!--UMLS code system-->
       <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/codeSystem/Ulms">
       <icadyCodeSys-
tem:codeSystemCode>2.16.840.1.113883.6.86</icadyCodeSystem:codeSystemCode>
       <icadyCodeSystem:codeSystemName>UMLS</icadyCodeSystem:codeSystemName>
       </rdf:Description>

       <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/MeasureSystem/MmHg">
       <skos:prefLabel>Milimeter Hg</skos:prefLabel>
       <skos:notation>mm[Hg]</skos:notation>
       </rdf:Description>

       <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/status/Complete">
       <skos:prefLabel>complete</skos:prefLabel>
       </rdf:Description>


       <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/example/MyVitalSignEntry">

       <icadyHL7V3:templIdRoot>1.3.6.1.4.1.19376.1.5.3.1.4.13</icadyHL7V3:templI
dRoot>

       <icadyHL7V3:templIdRoot>1.3.6.1.4.1.19376.1.5.3.1.4.13.2</icadyHL7V3:temp
lIdRoot>

       <icadyHL7V3:templIdRoot>2.16.840.1.113883.10.20.1.31</icadyHL7V3:templIdR
oot>

       <icadyHL7V3:code>
              <!--This is an instance-->
              <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/SystolicBloodPressure"/>
       </icadyHL7V3:code>

       <skos:note>
              This is a vital sign example. This is free text.
       </skos:note>
```

```
       <icadyHL7V3:status>
              <!--This is an instance-->
              <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/status/Complete"/>
       </icadyHL7V3:status>

       <icadyHL7V3:effectiveTime>
              201006010000
       </icadyHL7V3:effectiveTime>

       <icadyHL7V3:value>
              100
       </icadyHL7V3:value>

       <icadyHL7V3:unit>
              <!--This is an instance-->
              <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/MeasureSystem/MmHg"/>
       </icadyHL7V3:unit>

       </rdf:Description>

       <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/SystolicBloodPressure">
       <skos:prefLabel>Systolic blood pressure</skos:prefLabel>

       <skos:related>http://www.icardea.at/phrs/instances/VitalSigns</skos:relat
ed>
       <icadyCode:code>
              <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/SystolicBloodPressureULMSCode">
              <icadyCode:codeValue>C0871470</icadyCode:codeValue>
              <icadyCode:codeSystem>
                     <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/codeSystem/Ulms"/>
              </icadyCode:codeSystem>
              </rdf:Description>
       </icadyCode:code>
       </rdf:Description>
</rdf:RDF>
```

The following is an RDFS scheme able to validate a vital sign:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:skos="http://www.w3.org/2004/02/skos/core#">

    <rdfs:Class rdf:about="http://www.icardea.at/phrs/types/1.0/VitalSign">
        <rdfs:isDefinedBy
rdf:resource="http://www.icardea.at/phrs/types/1.0/"/>
        <rdfs:label xml:lang="en">HL7 V3 Vital Sign</rdfs:label>
```

```
        <rdfs:comment xml:lang="en">Encapsulate all the information for a sin-
gular HL7 V3 vital sign.</rdfs:comment>
    </rdfs:Class>

    <rdf:Property rdf:about="http://www.icardea.at/phrs/hl7V3#templIdRoot">
        <rdfs:isDefinedBy rdf:resource="http://www.icardea.at/phrs/hl7V3"/>
        <rdfs:label xml:lang="en">HL7 V3 Root Template ID</rdfs:label>
        <rdfs:domain
rdf:resource="http://www.icardea.at/phrs/types/1.0/VitalSign"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Literal"/>
        <rdfs:comment xml:lang="en">The template id used to indetify the mes-
sage like a certain one (e.g. IHE Survey Observation )</rdfs:comment>
    </rdf:Property>

    <rdf:Property rdf:about="http://www.icardea.at/phrs/hl7V3#code">
        <rdfs:isDefinedBy rdf:resource="http://www.icardea.at/phrs/hl7V3"/>
        <rdfs:label xml:lang="en">HL7 Code</rdfs:label>
        <rdfs:domain
rdf:resource="http://www.icardea.at/phrs/types/1.0/VitalSign"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Resource"/>
        <rdfs:comment xml:lang="en">The Code and Code System infromation used
to unique identify an item</rdfs:comment>
    </rdf:Property>

    <rdf:Property rdf:about="http://www.icardea.at/phrs/hl7V3#note">
        <rdfs:isDefinedBy rdf:resource="http://www.icardea.at/phrs/hl7V3"/>
        <rdfs:label xml:lang="en">Short Note</rdfs:label>
        <rdfs:domain
rdf:resource="http://www.icardea.at/phrs/types/1.0/VitalSign"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Literal"/>
        <rdfs:comment xml:lang="en">a short note</rdfs:comment>
    </rdf:Property>

    <rdf:Property rdf:about="http://www.icardea.at/phrs/hl7V3#status">
        <rdfs:isDefinedBy rdf:resource="http://www.icardea.at/phrs/hl7V3"/>
        <rdfs:label xml:lang="en">HL7 V3 Status</rdfs:label>
        <rdfs:domain
rdf:resource="http://www.icardea.at/phrs/types/1.0/VitalSign"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Resource"/>
        <rdfs:comment xml:lang="en">Indicates the HL7 V3 status</rdfs:comment>
    </rdf:Property>

    <rdf:Property rdf:about="http://www.icardea.at/phrs/hl7V3#effectiveTime">
        <rdfs:isDefinedBy rdf:resource="http://www.icardea.at/phrs/hl7V3"/>
        <rdfs:label xml:lang="en">Effective time</rdfs:label>
        <rdfs:domain
rdf:resource="http://www.icardea.at/phrs/types/1.0/VitalSign"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Literal"/>
        <rdfs:comment xml:lang="en">The effective time must follow the format
yyyymmddhhss (e.g. 201006010000)</rdfs:comment>
```

```
    </rdf:Property>

    <rdf:Property rdf:about="http://www.icardea.at/phrs/hl7V3#value">
        <rdfs:isDefinedBy rdf:resource="http://www.icardea.at/phrs/hl7V3"/>
        <rdfs:label xml:lang="en">The Value</rdfs:label>
        <rdfs:domain
rdf:resource="http://www.icardea.at/phrs/types/1.0/VitalSign"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Literal"/>
        <rdfs:comment xml:lang="en">The value for a measurement</rdfs:comment>
    </rdf:Property>

    <rdf:Property rdf:about="http://www.icardea.at/phrs/hl7V3#unit">
        <rdfs:isDefinedBy rdf:resource="http://www.icardea.at/phrs/hl7V3"/>
        <rdfs:label xml:lang="en">The Unit</rdfs:label>
        <rdfs:domain
rdf:resource="http://www.icardea.at/phrs/types/1.0/VitalSign"/>
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Resource"/>
        <rdfs:comment xml:lang="en">The unit for the involved value (from a
measurement)</rdfs:comment>
    </rdf:Property>

</rdf:RDF>
```

Extras: The previous table and example lists only the properties strictly related with the IHE PCC message; unfortunately in practice, additional properties may be useful as shown in Table 3.

namespace	local name	type	cardinality	mandatory	Description	Intro-duced By
http://www.icardea.at/phrs#	owner	literal	one or more	no	It contains information about the vital sign owner-ship	back end
http://www.w3.org/1999/02/22-rdf-syntax-ns#	type	resource	one or more	no	It describes the type for a vital sign	back end
http://www.icardea.at/phrs#	createDate	literal	none or one	no	the date in time when the vital sign item was created.	back end
http://www.icardea.at/phrs#	update-Date	literal	none or one	no	the date in time when the vital sign item was updated.	back end
http://www.icardea.at/phrs#	creator	literal	none or one	no	the instance (or entity) that	back end

					creates this vital sign	

Table 3 - PCC Properties Vital Signs

6.2 Observation

For a meaningful Observation PCC10 message the involved resource must contain at least the following properties shown in Table 4.

name-space	local name	type	value	cardi-nality	manda-tory	Description	Intro-duced By
http://www.icardea.at/phrs/hl7V3#	tem-plIdRoot	literal	2.16.840.1.113883.10.20.1.28	many	yes	All the templIdRoot templeate id allows me to identify the (involved) resource like a “Obser-vation”	Back-end
http://www.icardea.at/phrs/hl7V3#	tem-plIdRoot	literal	1.3.6.1.4.1.19376.1.5.3.1.4.5	many	yes	All the templIdRoot templeate id allows me to identify the (involved) resource like a “Obser-vation”	Back-end
http://www.icardea.at/phrs/hl7V3#	code	re-source	NA (there are already pre-defined in-stances for code e.g. Symptom)	only one	yes	Allows to identify the observation type (e.g. this observation is a Symptom)	Back-end
http://www.icardea.at/phrs/hl7V3#	status	re-source	NA (there are already pre-defined in-stances for status e.g. complete)	only one	yes	allows to identify the observation status.	6.2.1.1
http://www.icardea.at/phrs/hl7V3#	time-Start	literal it must follow the format yyy-ymmd dhhss (e.g. 20100 6010000)	NA	only one	yes	indicates the time when the observation starts.	Back-end
http://www.icardea.at	timeEnd	literal it must	NA	only one	no	indicates the time when the observation ends.	Back-end

/phrs/hl7V3#		follow the format yyy-ymmd dhhss (e.g. 20100 60100 00)				If this property id absent the observation is still running.	
http://www.w3.org/2004/02/skos/core#	note	literal	NA (the value is a string - try to keep it short)	only one	no	attaches a note to the involved observation	Back-end
http://www.icardea.at/phrs/hl7V3#	value-Code	re-source	NA (there are already pre-defined in-stances for code e.g. Bleeding-Gums)	only one	yes	allows to describe the observation in detail	Back-end

Table 4 - PCC Observation Properties

The Observation can be used for :

- Activity of daily living
- Risks
- Problems
- Symptoms

Nota:

- The Observation type can be manipulated via one or more specific **templIdRoot** e.g. if an *Observation* will be a HL7 V3 conform *Problem Entry* then the templIdRoot property must carry the values **1.3.6.1.4.1.19376.1.5.3.1.4.5** and **2.16.840.1.113883.10.20.1.28**.
- The Observation establishing process can be manipulated one specific **code**. This **code** allows us to say something like: This *Problem Entry* is a *Finding*, in this special case the Observation will contains two **templIdRoot** properties with values (**1.3.6.1.4.1.19376.1.5.3.1.4.5** and **2.16.840.1.113883.10.20.1.28**) and a single **code** property that points to a (controlled vocabulary instance named) *Finding*.

In order to relate Observation to each of this terms you can use the skos:related predicate. From the PCC10 point of view this information is irrelevant.

Here is a XML/RDF snippet for an Observation:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
```

```
       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
       xmlns:skos="http://www.w3.org/2004/02/skos/core#"
       xmlns:icady="http://www.icardea.at/phrs/"
       xmlns:icadyCode="http://www.icardea.at/phrs/code#"
       xmlns:icadyCodeSystem="http://www.icardea.at/phrs/codeSystem#"
       xmlns:icadyStatus="http://www.icardea.at/phrs/status#"
       xmlns:icadyBloodPreasure="http://www.icardea.at/phrs/status#"
       xmlns:icadyHL7V3="http://www.icardea.at/phrs/hl7V3#">


       <!--UMLS code system-->
       <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/codeSystem/Ulms">
       <icadyCodeSys-
tem:codeSystemCode>2.16.840.1.113883.6.86</icadyCodeSystem:codeSystemCode>
       <icadyCodeSystem:codeSystemName>UMLS</icadyCodeSystem:codeSystemName>
       </rdf:Description>


       <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/status/Complete">
       <skos:prefLabel>complete</skos:prefLabel>
       </rdf:Description>

       <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/Problem/Symptom">
       <skos:prefLabel>Symptom</skos:prefLabel>
       <icadyCode:code>
              <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/SymptomULMSCode">
              <icadyCode:codeValue>C1457887</icadyCode:codeValue>
              <icadyCode:codeSystem>
                     <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/codeSystem/Ulms"/>
              </icadyCode:codeSystem>
              </rdf:Description>
       </icadyCode:code>

       </rdf:Description>

       <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/BleedingGums">
       <skos:prefLabel>Bleeding gums</skos:prefLabel>

       <skos:related>http://www.icardea.at/phrs/instances/Problem/Symptom</skos:
related>
       <icadyCode:code>
              <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/BleedingGumsULMSCode">
              <icadyCode:codeValue>C0017565</icadyCode:codeValue>
              <icadyCode:codeSystem>
                     <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/codeSystem/Ulms"/>
              </icadyCode:codeSystem>
```

```
            </rdf:Description>
      </icadyCode:code>
      </rdf:Description>


      <!--
      Here starts the observation.
      -->
      <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/example/MyProblemEntry">

      <icadyHL7V3:templIdRoot>2.16.840.1.113883.10.20.1.28</icadyHL7V3:templIdR
oot>

      <icadyHL7V3:templIdRoot>1.3.6.1.4.1.19376.1.5.3.1.4.5</icadyHL7V3:templId
Root>

      <icadyHL7V3:code>
            <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/Problem/Symptom"/>
      </icadyHL7V3:code>

      <skos:note>
            This is a problem entry example. This is free text.
      </skos:note>

      <icadyHL7V3:status>
            <!--This is an instance-->
            <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/status/Complete"/>
      </icadyHL7V3:status>

      <icadyHL7V3:timeStart>
            201006010000
      </icadyHL7V3:timeStart>

      <icadyHL7V3:timeEnd>
            <!-- The end node can be absent-->
            201106010000
      </icadyHL7V3:timeEnd>

      <icadyHL7V3:value>
            <!--This is an instance-->
            <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/BleedingGums"/>
      </icadyHL7V3:value>
      </rdf:Description>

</rdf:RDF>
```

Extras: The previous table and example lists only the properties strictly related with the IHE PCC message; unfortunately in practice, additional properties may be useful. These are listed in Table 5.

name-space	local name	type	cardinality	manda-tory	Description	Introduced By
http://www.icardea.at/phrs#	owner	literal	one or more	no	It contains information about the observation ownership	back end
http://www.w3.org/1999/02/22-rdf-syntax-ns#	type	resource	one or more	no	It describes the type for a vobservation	back end
http://www.icardea.at/phrs#	create-Date	literal	none or one	no	the date in time when the observation item was created.	back end
http://www.icardea.at/phrs#	update-Date	literal	none or one	no	the date in time when the observation item was updated.	back end
http://www.icardea.at/phrs#	creator	literal	none or one	no	the instance (or entity) that creates this observation	back end

Table 5 – Extra Properties for Observation

6.3 Medications

For a meaningful Medication PCC10 message the resource must contains at least the following properties show in Table 6 and Table 7.

namespace	local name	type	value	cardinal-ity	manda-tory	Description	Intro-duced By
http://www.icardea.at/phrs/hl7V3#	tem-plIdRoot	literal	2.16.840.1.113883.10.20.1.24	many	yes	allows to identify the (involved) resource like a “Mediaction”	Back-end
http://www.icardea.at/phrs/hl7V3#	tem-plIdRoot	literal	1.3.6.1.4.1.19376.1.5.3.1.4.7	many	yes	allows to identify the (involved) resource like a “Mediaction”	Back-end
http://www.w3.org/2004/02/skos/core#	note	literal	NA (the value is a string - try to keep it short)only one	only one	no	attach a comment to this medication	Back-end
http://www.icar	status	re-	NA	only one	yes	indicate the	Back-

dea.at/phrs/hl7V3#		source	(there are already pre-defined in-stances for status e.g. complete)			medication status	end
http://www.icardea.at/phrs/hl7V3#	time-Start	literal it must follow the format yyy-ymmddhhss (e.g. 201006010000)	NA	only one	yes	indicates the time when the medication starts	Back-end
http://www.icardea.at/phrs/hl7V3#	timeEnd	literal it must follow the format yyy-ymmddhhss (e.g. 201006010000)	NA	none or only one	no	indicates the time when the medication end. If this value is missing then the medication is still running.	Back-end
http://www.icardea.at/phrs/hl7V3#	fre-quency	re-source	NA (there are already pre-defined in-stances for status e.g. twice in day)	only one	yes	indicates the medication frequency for the given time interval.	Back-end
http://www.icardea.at/phrs/hl7V3#	admin-Route	re-source	NA (there are already pre-defined in-stances for status e.g. oral rute)	only one	yes	indicates the medication administration mode (rute).	Back-end
http://www.icardea.at/phrs/hl7V3#	dosage	re-source	see the ex-ample from the snippet It contains the quantity and the unit of measure	only one	yes	describe the dosage for the medication	Back-end
6.3.1.1	6.3.1.2	6.3.1.3	6.3.1.4	6.3.1.5	6.3.1.6	6.3.1.7	6.3.1.8
6.3.1.9	6.3.1.10	6.3.1.11	6.3.1.12	6.3.1.13	6.3.1.14	6.3.1.15	6.3.1.16

Table 6 - PCC Properties Medications

Medication Dosage

namespace	local name	type	value	cardinality	manda-tory	Descrip-tion	6.3.1.17
http://www.icardea.at/phrs/hl7V3#	value	literal	NA	only one	yes	the dosage value	Back-end
http://www.icardea.at/phrs/hl7V3#	unit	resource	NA (there are already prede-fined instances for the unit e.g. Pill)	only one	yes	describe the unit for this dos-age value.	Back-end

Table 7 - PCC properties Medication Dosage

Here is a XML/RDF snippet for a Medication:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
       xmlns:skos="http://www.w3.org/2004/02/skos/core#"
       xmlns:icady="http://www.icardea.at/phrs/"
       xmlns:icadyCode="http://www.icardea.at/phrs/code#"
       xmlns:icadyCodeSystem="http://www.icardea.at/phrs/codeSystem#"
       xmlns:icadyStatus="http://www.icardea.at/phrs/status#"
       xmlns:icadyBloodPreasure="http://www.icardea.at/phrs/status#"
       xmlns:icadyHL7V3="http://www.icardea.at/phrs/hl7V3#">


       <!--UMLS code system-->
       <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/codeSystem/Ulms">
       <icadyCodeSys-
tem:codeSystemCode>2.16.840.1.113883.6.86</icadyCodeSystem:codeSystemCode>
       <icadyCodeSystem:codeSystemName>UMLS</icadyCodeSystem:codeSystemName>
       </rdf:Description>



       <rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/status/Complete">
       <skos:prefLabel>complete</skos:prefLabel>
       </rdf:Description>
```

```
	<rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/MeasureSystem/Tablet">
	<skos:prefLabel>tablet</skos:prefLabel>
	<skos:altLabel>pill</skos:altLabel>
	</rdf:Description>

<!--The next code is generated -->

	<rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/example/MyMedicationEntry">

	<icadyHL7V3:templIdRoot>2.16.840.1.113883.10.20.1.24</icadyHL7V3:templIdR
oot>

	<icadyHL7V3:templIdRoot>1.3.6.1.4.1.19376.1.5.3.1.4.7</icadyHL7V3:templId
Root>

	<icadyHL7V3:code>
		<!--This is an instance-->
		<rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/SystolicBloodPressure"/>
	</icadyHL7V3:code>

	<skos:note>
		This is a vital sign example. This is free text.
	</skos:note>

	<icadyHL7V3:status>
		<!--This is an instance-->
		<rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/status/Complete"/>
	</icadyHL7V3:status>

	<icadyHL7V3:timeStart>
		201006010000
	</icadyHL7V3:timeStart>

	<icadyHL7V3:timeEnd>
		<!-- The end node can be absent-->
		201106010000
	</icadyHL7V3:timeEnd>

	<icadyHL7V3:frequecy>
		<!--This is an instance-->
	</icadyHL7V3:frequecy>

	<icadyHL7V3:adminRoute>
		<rdf:Description rdf:about="http://www.icardea.at/phrs/OralRute"/>
	</icadyHL7V3:adminRoute>

	<icadyHL7V3:dosage>
		<rdf:Description
rdf:about="http://www.icardea.at/phrs/MyDossage"/>
	</icadyHL7V3:dosage>
```

```
	</rdf:Description>

	<rdf:Description rdf:about="http://www.icardea.at/phrs/OralRute">
	<icadyCode:code>
		<rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/OralRuteCode">
		<icadyCode:codeValue>C0017565</icadyCode:codeValue>
		<icadyCode:codeSystem>
			<rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/codeSystem/Ulms"/>
		</icadyCode:codeSystem>
		</rdf:Description>
	</icadyCode:code>



	</rdf:Description>


	<rdf:Description rdf:about="http://www.icardea.at/phrs/MyDossage">
	<icadyHL7V3:value>
		1.2
	</icadyHL7V3:value>
	<icadyHL7V3:unit>
		<rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/MeasureSystem/Tablet"/>
	</icadyHL7V3:unit>
	</rdf:Description>

	<rdf:Description rdf:about="http://www.icardea.at/phrs/MyDroug">
	<icadyCode:code>
		<rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/DrugCode">
		<icadyCode:codeValue>C0017565</icadyCode:codeValue>
		<icadyCode:codeSystem>
			<rdf:Description
rdf:about="http://www.icardea.at/phrs/instances/codeSystem/Ulms"/>
		</icadyCode:codeSystem>
		</rdf:Description>
	</icadyCode:code>
	<skos:prefLabel>complete</skos:prefLabel>
	<skos:note>
		This is a problem entry example. This is free text.
	</skos:note>


	</rdf:Description>

</rdf:RDF>
```

Extras : The previous table and example lists only the properties strictly related with the IHE PCC message; unfortunately in practice, additional properties may be useful. Those are listed in Table 8.

namespace	local name	type	cardinality	manda-tory	Description	Introduced By
http://www.icardea.at/phrs#	owner	literal	one or more	no	It contains information about the medication ownership	back end
http://www.w3.org/1999/02/22-rdf-syntax-ns#	type	resource	one or more	no	It describes the type for a medication	back end
http://www.icardea.at/phrs#	createDate	literal	none or one	no	the date in time when the medication item was created.	back end
http://www.icardea.at/phrs#	update-Date	literal	none or one	no	the date in time when the medication item was updated.	back end
http://www.icardea.at/phrs#	creator	literal	none or one	no	the instance (or entity) that creates this medica-tion	back end

Table 8 - Extra Properties for Medications