SALUS

"Scalable, Standard based Interoperability Framework for Sustainable Proactive Post Market Safety Studies"

SPECIFIC TARGETED RESEARCH PROJECT

PRIORITY Objective ICT-2011.5.3 b) Tools and environments enabling the re-use of electronic health records

SALUS D5.2.2 Query Based Interoperability Profiles and Open Source Toolsets –R2

Due Date:	January 31, 2014
Actual Submission Date:	January 31, 2014
Project Dates:	Project Start Date : February 01, 2012 Project End Date : January 31, 2015 Project Duration : 36 months
Deliverable Leader:	OFFIS

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	2013-01-31	Initial Document	OFFIS	All Consortium
V0.2	2014-01-20	First version	OFFIS	SRDC
V0.3	2014-01-24	Review result	SRDC	OFFIS
V0.4	2014-01-28	Second version	OFFIS	All Consortium
V1.0	2013-01-31	Final version	SRDC	EU-commission

Contributors (Benef.)	Frerk Müller (OFFIS), Gokce Banu Laleci Erturkmen (SRDC) Ali Anil Sinaci (SRDC) Gerard Freriks (ERS) Tobias Krahn (OFFIS) Marco Eichelberg (OFFIS)			
Responsible Author	Frerk Müller		**Email**	frerk.mueller@offis.de
	Beneficiary	OFFIS	**Phone**	+49-441-9722-146

SALUS Consortium Contacts:

Beneficiary	Name	Phone	Fax	E-Mail
SRDC	Gokce Banu Laleci Erturkmen	+90-312-2101763	+90(312)2101837	gokce@srdc.com.tr
EUROREC	Georges De Moor	+32-9-2101161	+32-9-3313350	georges.demoor@ugent.be
UMC	Niklas Norén	+4618656060	+46 18 65 60 80	niklas.noren@who-umc.org
OFFIS	Wilfried Thoben	+49-441-9722131	+49-441-9722111	thoben@offis.de
AGFA	Dirk Colaert	+32-3-4448408	+32 3 444 8401	dirk.colaert@agfa.com
ERS	Gerard Freriks	+31 620347088	+31 847371789	g.freriks@e-Recordservices.eu
LISPA	Davide Rovera	+3902393311	+39 02 39331207	davide.rovera@lispa.it
INSERM	Marie-Christine Jaulent	+33142346983	+33153109201	marie-christine.jaulent@crc.jussieu.fr
TUD	Peter Schwarz	+49 351 458 2715	+49 351 458 7319	Peter.Schwarz@uniklinikum-dresden.de
ROCHE	Jamie Robinson	+41-61-687 9433	+41 61 68 88412	jamie.robinson@roche.com

EXECUTIVE SUMMARY

This document describes the Technical Interoperability Layer (TIL) of SALUS. It sets up the communication layer between external parties and SALUS semantic layer as well as SALUS semantic layer and EHR systems by using existing standards. This documents describes the data to be exchanged, the profiles developed for communication, the extension of existing profiles to be made to fulfil SALUS requirements and the Open Source Toolset including the software to be implemented. This is the second version of TIL specification for query based communication protocols; the first version was delivered by Month 12 in D5.2.1.

Within the second year, the components have been implemented and adapted if needed. Some slight changes of the profiles have been made due to issues found during implementation. The components have been tested on LISPA data-warehouse and a fake database running locally at the developer site. Next to the profiles also the LISPA-connector has been implemented. This component extracts the required information from the LISPA data-warehouse to be transported through IHE QED profile. In case of this deliverable it represents the EHR-connector, which is needed to map database-tables to IHE profile fields.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Purpose

The goal of the SALUS is to establish an interoperability layer between post market safety analysis and reporting tools and the underlying EHR systems. There are two kinds of interoperability challenges we need to address: technical and syntactic interoperability and semantic interoperability. SALUS workpackage 5 addresses technical and syntactic interoperability challenge, by enabling the post market safety analysis and reporting tools seamlessly exchange information with the EHR systems. On the other hand, Workpackage 4 addresses the semantic interoperability challenge, i.e. ensures that the exchanged information can be meaningfully interpreted by the communicating parties.

When we have analysed the selected SALUS pilot use cases (described in detail in SALUS Deliverable D8.1.1), we have seen that there is a need for standard based interfaces:

- (1) to seamlessly query heterogeneous EHR systems for analysing and detecting possible ADEs, pre-filling case safety reports and for enabling signal follow-up studies to trace the safety reports back to the related EHRs
- (2) to seamlessly specify the target eligible patient group for enabling set up of continuous safety studies that screen EHRs
- (3) to specify the requested clinical data by intelligent data analysis tools for the selected group of patients
- (4) to transfer the specified de-identified clinical data to the clinical data registries for the selected patients for safety analysis

A further analysis showed that, two kinds of interactions shall be supported between post market safety analysis and reporting tools and the underlying EHR systems: query based and subscription based. This deliverable focuses on query based communication protocol, while Deliverable 5.1.2 focuses on a subscription based protocol.

The purpose of the document is to explain which existing standards have been analyzed to perform the standard based communication between SALUS system and existing EHR systems as well as between external parties (such as safety analysis tools) and the SALUS system for a query based interaction. Next to this it will be explained how the chosen existing standards have been adapted to transport all information needed from SALUS. Therefore the standards chosen (in this case IHE Query for Existing Data -QED) were adapted to also carry population based queries which is not supported by IHE QED yet and also EN13606 related content. To fulfil this task the HL7 messages of the PCC-1 transaction of the IHE QED profile was changed. This document will show how this was done. Secondly the Open Source Toolset is described.

As presented, this document is closely related to D5.1.2. As the query based interoperability profiles also the subscription based profiles will need to transport SALUS specific queries and EN13606 related content. As both profiles depend on the same standards this document refers to D5.1.2 in case of duplicated information to avoid any unnecessary redundancies within these two documents.

1.2 Achievements in the 2nd release

Whereas the first project year was focusing on the selection of the profile and the changes to be made, the second project year focused on the implementation, testing and debugging of the profile. Within the second year, the web services for the transactions, the message generation and parsing as well as

the components connecting the profiles to the EHR system have been integrated to SALUS system. During implementation also some minor changes of the IHE QEDext module have been made. Those issues have been found during implementation of the profile and were necessary for satisfying the SALUS use cases. Next to the implementation also the integration and deployment tests to the target system at LISPA site have been performed. Due to data protection issues the code has been mostly developed and debugged on developer systems using fake data and afterwards transported to LISPA system. On LISPA systems the technical crew performed a checkout and tested the implementation against a given deployment guide. The results have been sent back to the developer to check if the test results were satisfying. Once the test on the LISPA IT system runs smooth on the fake data, the same tests have been applied to the real database (including millions of patients), to figure out if the system is still stable against big sets of data. A meeting in February 2014 will make the integration of TIL to SALUS system complete. As the remote testing was satisfying so far, it is expected to have TIL ready for evaluation during the meeting.

This Deliverable is an update of the D5.2.1 delivered on month 12. Sections that have major revisions are “Query for Existing Data Integration Profile extended (QEDext)”, “Structure of IHE QEDext Module”, “Parsing of QEDext query objects (parameter list)” and Appendix.

1.3 Reference documents

The following documents were used or referenced in the development of this document:

- SALUS Deliverable 5.1.2 - Subscription Based Interoperability Profiles and Open Source Toolsets – R2
- SALUS Deliverable 5.2.1 - Query Based Interoperability Profiles and Open Source Toolsets – R1
- SALUS Deliverable 4.1.1 - SALUS Content models for the Functional Interoperability Profiles for Post Market Safety Studies - R1
- SALUS Description of Work (SALUSPartB_20110118.pdf)
- SALUS Deliverable 8.1.1- Pilot Application Scenario and Requirement Specifications of the Pilot Application
- SALUS Deliverable 3.3.1- Requirement Specification of the SALUS Architecture
- SALUS Deliverable 3.2.1 – Survey of the state of the art

1.3.1 Abbreviations and Acronyms

Table 1 List of Abbreviations and Acronyms

Abbreviation/ Acronym	DEFINITION
ADE	Adverse Drug Event
ANT	ADE Notification Tool
CCD	Continuity of Care Document
CDA	Clinical Document Architecture
EHR	Electronic Health Record
EN13606	Health informatics - Electronic Health Record Communication
HL7	Health Level 7
HQMF	Health Quality Measurement Format
IHE	Integrating the Healthcare Enterprise
IHE CM	IHE Care Management profile
IHE QED	IHE Query for Existing Data profile
IHE RFD	IHE Request Form for Data capture profile
IHE CRD	IHE Clinical Research Document profile
IHE DSC	IHE Drug Safty Content profile
PCC	Patient Care Coordination
QRDA	Quality Reporting Document Architecture
RDF	Resource Description Framework
RIM	Reference Information Model
SIL	Semantic Interoperability Layer
SPARQL	SPARQL Protocol And RDF Query Language
TIL	Technical Interoperability Layer
TIQSDS	Technical Interoperability Query Source Data Service
TILDS	Technical Interoperability Layer Data Service
UML	Unified Modelling Language
XMI	XML Metadata Interchange
XML	eXtensible Markup Language
XSD	XML Schema Definition

2 CONCEPTUAL DESIGN OF QUERY BASED PROFILES

This section focuses on the design of a profile to query data from an existing EHR system. The profile needed should explicitly be used for querying data not for subscribing to data. A query requires a direct feedback. The query call leads to a query result directly on method call synchronously.

As the system should be connected at the end to an existing EHR system which already may follow the standards used in health care it was decided not to design a new protocol but to follow existing protocols and extend them if needed. Therefore Integrating the Healthcare Enterprise (IHE) profiles are used. IHE profiles are not standards themselves but using several existing standards and constraining them. They aim to facilitate selected use cases. This also includes combination and restriction of standards within one profile.

Already within D3.2.1 several of the IHE profiles have been analyzed with respect to the SALUS needs. There, IHE RFD (Request Form for Data capture), IHE DSC (Drug Safety Content profile), IHE CRD (Clinical Research Document), IHE QED (Query for Existing Data profile) and also IHE CM (Care Management) have been taken into account. IHE RFD is focusing on transportation of data with the goal of filling or pre-filling forms. Also the forms itself can be transported by IHE RFD. As SALUS needs to pre-fill forms for the case safety reports, this profile was analyzed. IHE DSC and IHE CRD are extensions of this profile, providing conversion guidelines between selected content models. After examining the use cases described in D8.1.1, it was found that in case of the SALUS architecture an additional profile is needed which does not focus on the forms but on the queries. Therefore IHE RFD was eliminated from the option list. As IHE CM is doing a subscription to a Data Repository and IHE QED is querying for data, IHE QED was chosen to perform the query.

For SALUS, the IHE QED profile had still to be extended to fulfil all features of SALUS. During the state of the art analysis it was found that there is no profile fulfilling all needs of SALUS therefore IHE QED was chosen as it is an industry led initiative in healthcare domain and only needed slight adaptations. Basically two major issues had to be addressed. IHE QED is a patient centred profile which does mean that it is focusing on queries for a single patient. SALUS also needs to query for population based information not related to a single patient. An example for such a query may be: "Bring me the de-identified medical summaries of patients got a diagnosis A after taking a drug B". Such a query and also the result are not covered by IHE QED profile. We have analysed the HL7 Health Quality Measures Format (HQMF)[1] as an option to express population based queries, and agreed that a subset of HQMF might be integrated to the extended IHE QED profile to express population based queries. Therefore new fields had to be added to the IHE QED profile as explained in the following sections, and the new profile is called IHE QEDext. When it comes to result sets, in IHE QED itself, the results of the queries are shared in terms of HL7 CCD based entry templates. As we are also querying de-identified medical summaries of eligible patient populations, we again aim to use CCD templates. For addressing the requirements of SALUS pilot applications, in D4.1.2 the required CCD templates have already been defined, these are used for representing the result sets of QEDext response messages. A brief overview of these content models is presented in D5.1.2, while these templates are presented in detail in D4.1.2.

In SALUS we do not want to stick only to one standard to carry patient data, such as HL7 CCD based templates. We would like to also cover a second standard which is EN13606. The respective EN13606 based artefacts are already described in D4.1.2, and briefly summarized in D5.1.2. The data transported by IHE QED EN13606 is represented in an XML structure. This allows easy integration of EN13606 into the transport process within QED transactions. EN13606 itself does not specify any transportation or communication layer. It just describes the medical data to be transported. Next to this, EN13606 will also add a query format to the project which will allow defining a query for all data needed in EN13606.

[1] Representation of the Health Quality Measures Format (eMeasure), http://www.hl7.org/implement/standards/product_brief.cfm?product_id=97

As these query and result definitions are identical to the ones used in subscription based profile already described in D5.1.2 (the corresponding deliverable to this one focusing on subscription based queries) these will be not repeated within this deliverable to avoid further redundancies. The message templates to be exchange are described more in detail in D4.1.2.

3 COMMUNICATION PROTOCOL FOR DATA EXCHANGE BETWEEN SALUS SYSTEM AND AN EXTERNAL DATA SOURCE BASED ON IHE PROFILES

As explained in Section 2, several existing industry led profiles and standards have been analysed in advance to figure out which profile fits best to the SALUS use cases. For query based interaction with an EHR system, we have selected the IHE QED profile. This is an already existing profile, which was extended for some points described within the following to fulfil all required features.

3.1.1 Query for Existing Data Integration Profile extended (QEDext)

IHE QED is part of the IHE Patient Care Coordination (PCC) domain and, therefore, part of the PCC technical framework. It supports dynamic data query for clinical data. It covers common observations, problems and allergies, medications, immunizations, professional services and diagnostic results. Therefore, QED supports clinical care, quality reporting, financial transactions, public health reporting, clinical trials, drug interaction checking, and patient qualification. This is most of the information needed for SALUS scenarios, but a few things had to be adapted. One thing that is missing is the transportation of EN13606 based content. As the content currently transported by IHE QED profile is CDA/CCD the EN13606 document type had to be added to the profile and therefore also to be transported by HL7v3. Next to the challenge of transporting EN13606, it was also necessary to extend the query language for IHE QED as it should be possible to express population based queries through this extension. For this, HL7 HQMF was a solution as it is already defined in terms of the HL7 RIM. As the aim of the profile to be designed is to be standard conformant in case of a HL7 RIM based message format, it is necessary that HL7 messages to be extended can be derived from HL7 RIM. As HQMF already is derived from RIM it could be easily integrated to the new messages to be introduced to IHE QED profile. Next to this HQMF was already designed to represent population based queries. Therefore a subset of HQMF was a good solution to query for population based datasets. Within this deliverable HQMF will be introduced as the query format to be transported. Representing the queries with HQMF and carrying the resulting data sets with the available CCD templates and 13606 templates (defined in D4.1.2) fulfills our needs.

IHE defines two actors related to QED profile. These are the *Clinical Data Consumer* actor querying data and the *Clinical Data Source* actor providing data. The query itself is defined by the QED profile specification (see Figure 1). For QEDext no changes within this structure were made as only the content to be delivered, but not the participating actors, had to be adapted.

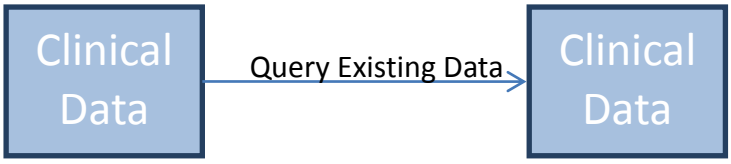

Figure 1: QED actors and transactions

Regarding the data exchanged, the content is related to the HL7 Continuity of Care Document (CCD) that it is a constrained version of the Clinical Document Architecture (CDA). [2] This had to be adapted as EN13606 is not based on CDA/CCD and also the results of a HL7 HQMF might not fit to a patient centered document[3].

To describe how the currently available IHE QED profile was adapted to be EN13606 aware in the following the profile is explained in more detail and possible solutions carrying EN13606 is presented as well as for transporting population based queries which are presented in HQMF and the corresponding data sets in CDA or EN13606.

Figure 2 describes the high level actions between the actors. The transaction defined within the IHE profile is the PCC-1 transaction. This transaction was modified as described within the following. Therefore the PCC-1 transaction is split in several sub-transactions. The first sub-transaction to be used for querying data is the "Get Care Record Profile Query". This query can be further split into several HL7 messages which will be described in more detail below. This message carries the query to be asked for and is sent from the *Clinical Data Consumer* to the *Data Repository*. This query can contain several options which define the data to be asked for. Next to the type of data to be returned also the quantity of data and the parameters to select of the patient of interest will be send. There are several other parameters which will not be discussed here as they are not important to describe the principals of this approach. Below the options to be chosen are described. If none of them is present in the query information about all of these options shall be returned.

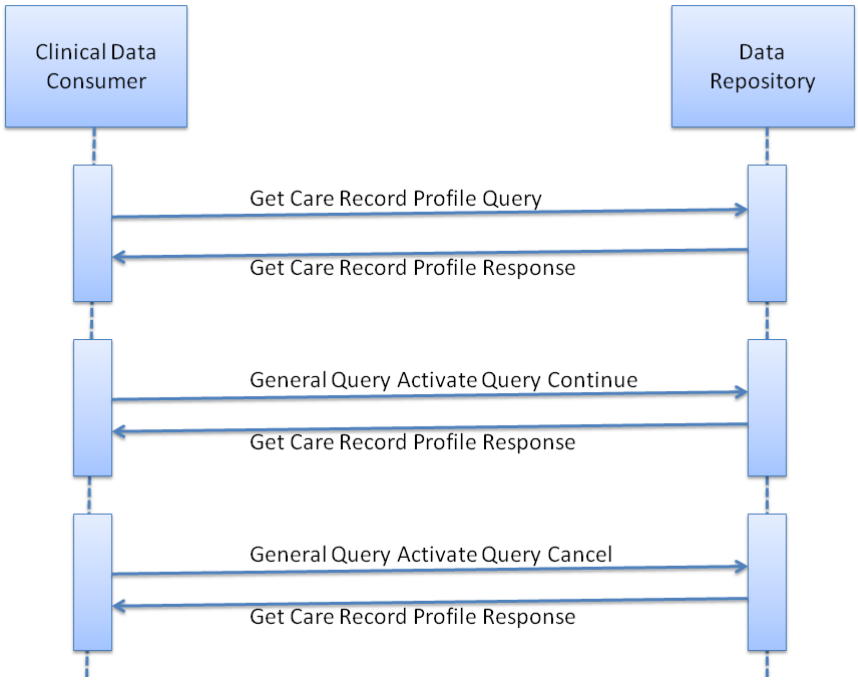

Figure 2: Interaction diagram of IHE QED profile

[2] IHE Query for Existing Data Integration Profile (PCC TF-1/QED) specification: http://www.ihe.net/Technical_Framework/upload/IHE_PCC_Query_for_Existing_Data_QED_Supplement_TI_2008-08-22.pdf

[3] It should be noted that HQMF is designed for collecting statistics as quality measures from the selected population. In SALUS on the other hand we would like to collect de-identified medical summaries of the selected population to be analyzed further by post market safety analysis tools. Hence, for the time being, we will stick with HL7 CCD based templates and 13606 EHRExtract templates as the query results.

Options:

The QED profile divides information into six main classes to distinguish where the information might be found.

- **Vital Signs Option**: A *Clinical Data Consumer* implementing the query for *Vital Sign Option* has to use the vocabulary for *Vital Signs* as defined in *PCC-1*.
- **Diagnostic Data Option**: This is a collection of information gathered from e.g. laboratories, testing equipments or imaging procedures. A *Clinical Data Consumer* implementing the query for the *Diagnostic Data Option* has to use the vocabulary for *Diagnostic Data* as defined PCC-1.
- **Problems and Allergies Option**: This is a collection of diagnoses, clinical findings, allergies or other risk factors that are recorded for a patient.
    - Conditions
    - Intolerances
    - Risk Factors

  A *Clinical Data Consumer* implementing the query for the *Problems and Allergies Option* has to use the vocabulary for *Problems and Allergies* as defined in *PCC-1*.
- **Medications Option**: A *Clinical Data Consumer* implementing the query for the *Medication Option* has to use the vocabulary for *Medications* as defined in *PCC-1*.
- **Immunizations Option***:* A *Clinical Data Consumer* implementing the query for the *Immunization Option* has to use the vocabulary for *Immunization* as defined in *PCC-1*.
- **Professional Services Option***:* This is a collection of procedures and/or encounters which the patient has to participate in. A *Clinical Data Consumer* implementing the query for the *Professional Services Option* has to use the vocabulary for *Professional Services* as defined in *PCC-1*.

Once the query has been sent to the *Data Repository* it will return the data available within the "Get Care Record Profile Response". This response may include some kind of error message due to several expected issues. This might be a simple template problem within the query, missing information or also technical issues as network problems. If no issue could be found the resulting data will be encapsulated into the response message and sent back to the *Clinical Data Consumer*. This information does not need to be complete as by the query more than one repository may return the results. The medical content returned by the query is based on extracts of the CCD/CDA templates defined for the IHE QED profile. The data consumer can collect the data and recreate a complete CCD/CDA document out of the results. A similar approach is done for the EN13606 messages. It is possible to transport the relevant parts of EN13606 messages and reconstruct a complete EN13606 document derived from several parties at the end. Therefore the HL7 messages performing the "Get Care Record Profile Response" had to be extended. Beside the transport of patient centered data independent from EN13606 or CDA also population based queries can be performed. Therefore a subset of HQMF is selected to be adopted as a solution. It allows querying for population based datasets which can be used for example in statistics. The result of such a query is a CCD based document.

Once the *Data Consumer* got a "Get Care Record Profile Response" it can decide to query for more information by sending a "General Query Activate Query Continue". This message does not contain any medical information and can be therefore left unchanged for the profile extension. The response on this message is again a "Get Care Record Profile Response" which will need to have the same extension as being a result of the "Get Care Record Profile Query".

Once the *Data Consumer* got all information needed or all *Data Repositories* stated that no more information is available, the query can be closed by the "General Query Activate Query Cancel" message which will just have an empty "Get Care Record Profile Response" as result to acknowledge the cancelation message. Within the following the transactions are described in more detail and also the changes to be made for transportation of the different contents are explained.

Transactions
This list of transaction as part of the IHE profile is divided into the following set of simple transactions as described before (see Figure 2). These transactions are based on the IHE QED profile and consist of a set of HL7 version 3 messages to fulfill the transport. These messages have been adapted to transport on the one hand complex population based queries and the results as also EN13606 content. How this was solved will be described within the following.

Get Care Record Profile Query: In the original IHE QED profile, this message is executed when a *Clinical Data Consumer* queries information about a patient. It contains a *Transmission Wrapper* holding information about transmission and routing, a *Control Act Wrapper* holding information about the participating actors and the payload itself composed of a set of parameters that can be collected by a *Clinical Data Source* and includes information about the patient.

An example of the *Transmission Wrapper* can be found below. All colored lines are remaining to be explicitly affected by the IHE QED profile. Furthermore the green marked lines will have to be adapted for the IHE QEDext profile. It may be necessary to introduce a new HL7 message called "Get Care Record Profile Query extension" which will lead to a new "QUPC_IN043100UVext" name which has to be changed within the *Transmission Wrapper*. The transmission wrapper to be adapted is the based on the HL7 message "MCCI_MT000100UV01".

```
<QUPC_IN043100UVext xmlns="urn:hl7-org:v3" ITSVersion="XML_1.0"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <id root=' ' extension=' '/>
 <creationTime value=' '/>
 <interactionId extension='QUPC_IN043100UVext' root='2.16.840.1.113883.5'/>
 <processingCode code='D|P|T'/>
 <processingModeCode code='T'/>
 <acceptAckCode code='AL'/>
 <receiver typeCode="RCV">
  <device determinerCode="INSTANCE">
   <id/>
   <name/>
   <telecom value=' ' />
   <manufacturerModelName/>
   <softwareName/>
  </device>
 </receiver>
 <sender typeCode="SND">
  <device determinerCode="INSTANCE">
   <id/>
   <name/>
   <telecom value=' ' />
   <manufacturerModelName/>
   <softwareName/>
  </device>
 </sender>
 <controlActProcess>
  See Control Act Wrapper below
 </controlActProcess>
</QUPC_IN043100UV>
```

Within validation schema of the "MCCI_MT000100UV01" HL7 message it is shown that this message will be not affected by the changes made for the IHE QEDext profile ("QUPC_IN043100UVext"). Therefore the Transmission Wrapper will be not described in more detail. To distinguish clearly between these two messages: "QUPC_IN043100UV" describes the usage of the *Transmission Wrapper* and the *Control Act Wrapper*. This needed adaption within the Transmission Wrapper part as the message got a new name and root-ID. The *Transmission Wrapper* definition itself was left unchanged as the exchange of name and root-ID is still a valid "MCCI_MT000100UV01" message.

As the *Transmission Wrapper* of "QUPC_IN043100UV" needs no further adaption the next message to analyze is the Control Act Wrapper "QUQI_MT020001UV01". This Control Act Wrapper transports the payload defined in HL7 message "QUQI_MT020001UV01". This will be described later in the document. Below it is shown an example of the *Control Act Wrapper*. As for the *Transmission Wrapper* it is also colored to label the adoptions.

```
<controlActProcess moodCode="RQO">
  <id root=' ' extension=' '/>
  <code code='QUPC_TE043100UV'/>
  <effectiveTime value=' '/>
  <languageCode code=' '/>
  <authorOrPerformer typeCode=' '></authorOrPerformer>

  <queryByParameter>
    <id root=' ' extension=' '/>
    <statusCode code='new'/>
    <responseModalityCode code='R'/>
    <responsePriorityCode code='I'/>
    <initialQuantity value=/>
    <initialQuantityCode code='REPC_RM000100UV' codeSystem='2.16.840.1.113883'>
    <parameterList>
      see Query Parameter List below
    </parameterList>
  </queryByParameter>
</controlActProcess>
```

Blue marked sequences are affecting the HL7 message "QUQI_MT020001UV01" directly within the IHE QED profile and do not belong to any of the extensions to be made for SALUS.

```
<parameterList>
    <queryByEn13606/>
    <populationBasedQuery>
      </QualityMeasureDocument>
      …
      </QualityMeasureDocument>
    </populationBasedQuery>
  <careProvisionCode>
    <value code=' ' displayName=' ' codeSystem=' ' codeSystemName=/>
  </careProvisionCode>
  <careProvisionReason>
    <value code=' ' displayName=' ' codeSystem=' ' codeSystemName=/>
  </careProvisionReason>
  <careRecordTimePeriod>
    <value><low value=' '/><high value=' '/></value>
  </careRecordTimePeriod>
  <clinicalStatementTimePeriod>
    <value><low value=' '/><high value=' '/></value>
  </clinicalStatementTimePeriod>
  <includeCarePlanAttachment><value value='true|false'/></includeCarePlanAttachment>
  <maximumHistoryStatements><value value=' '/></maximumHistoryStatements>
  <patientAdministrativeGender>
    <value code=' ' displayName=' '
      codeSystem='2.16.840.1.113883.5.1' codeSystemName='AdministrativeGender'/>
  </patientAdministrativeGender>
  <patientBirthTime><value value=' '/></patientBirthTime>
  <patientId><value root=' ' extension=' '/></patientId>
  <patientName><value></value></patientName>
</parameterList>
```

Depending on the type of query the *Parameter List* will hold the information which queries to perform. Therefore the *Parameter List* will have to be adapted within the payload field. The payload is defined in HL7 message "QUPC_MT040100UV". An example is shown above.

The payload defines the type of data to be queried for. Within the IHE QED profile this part contains the medical information and is therefore the most interesting part of the whole message. As shown in the example this payload does not only define the data to query for by the options, but also the patientID or some other patient related data. As this new extension also allows querying for population based data, the *Parameter List* has to be adapted. The most obvious thing to change is that the *Parameter List* has to carry exactly one *Patient-ID* in the original profile. As this query will also be used for population based queries which may have as a result a measured value which does not necessarily belong to a single patient, but may contain statistical data of a cohort of patients. This field has been changed in cardinality from [1..1] to [0..n]. In case of a "*"-operator, information for all patients is queried. If the patient Id is left empty the tag for population based queries within the payload has to be used.

The 2 green marked tags labeled "<queryByEn13606/>" and "<populationBasedQuery />" are added to the message to transport this new information. Only one of these tags should be present depending on the information to be carried. Both tags should extend the ACT-definition of HL7-RIM to be HL7v3 compatible.

In case of a query defined in EN13606 content model the "<queryByEn13606/>" has to be used. In case of a HQMF query the "<populationBasedQuery/>" tag will carry the information. Within this tag a complete HQMF document encapsulated into <QualityMeasureDocument> section has to be placed. HQMF was already described in section 2 in D5.1.2. The same information coded in HQMF can also be expressed in EN13606. As HL7 message format was not designed to carry EN13606 content the "<queryByEn13606/>" was introduced to the *Payload.* EN13606 is also based on XML and can therefore be fitted to the structure of the message: In the simplest way as structured XML. The validation of this private tag will just stick to XML and not analyze the EN13606 content as this is not part of the HL7 RIM or R-MIM objects.

This adaption of this two new fields lead to new modified *Parameter List* as defined below:

Parameter Name	Cardinality	Data Type	Vocabulary Domain	Consumer	Source
careProvisionCode	0..1	CD		O	R
careProvisionReason	0..*	CD		O	O
careRecordTimePeriod	0..1	IVL<TS>		O	R
clinicalStatementTimePeriod	0..1	IVL<TS>		O	R
includeCarePlanAttachment	0..1	BL		R	R
maximumHistoryStatements	0..1	INT		O	R
patientAdministrativeGender	0..1	CE	AdministrativeGender	O	R
patientBirthTime	0..1	TS		O	R
patientId	0..n	II		R	R
patientName	0..1	PN		O	R
populationBasedQuery	0..1	ED	HQMF	O	R
queryByEn13606	0..1	ED	EN13606	O	R

Colored entries within the table mark modifications to the original IHE QED profile. The *Patient ID* is changed in cardinality and can be now left out in case of an HQMF or EN13606 based query and also set to "*"-operator, if the query should belong to all patients of the data-warehouse. The two green marked entries are totally new to the IHE QED extension. Both entries can be left out as it is in this case compatible to IHE QED. The data type is set to encapsulate data (ED) as not further constrained. This would allow also transportation of other content then XML based data. The

vocabulary domains are mapped to EN13606 or HQMF depending on the tag. Both fields must be supported by a *Data Source* if the extension should be used, but can be left out by the *Data Consumer*.

Below the possible result sets of IHE QEDext are listed. In SALUS Project, the IHE QEDext profile and its implementation will be demonstrated at Lombardy region (LISPA site). The LISPA data warehouse as a clinical data source does not include all clinical data needed in a coded way, for example lab results are missing. The green marked Options in the following table are the options that can be tested on LISPA dataset. Other cannot be tested due to the fact that the information is not available in LISPA Datawarehouse. However, the implementation of the toolset was made to transport all of these options.

Actor	**Option**
Clinical Data Source	Vital Signs Option
	Problems and Allergies Option
	Diagnostic Data Option
	Medications Option
	Immunizations Option
	Professional Services Option
Clinical Data Consumer	Vital Signs Option
	Problems and Allergies Option
	Diagnostic Data Option
	Medications Option
	Immunizations Option
	Professional Services Option

Get Care Record Profile Response: This message is executed as answer on *Get Care Record Profile Query, General Query Activate Query Continue* or *General Query Activate Query Cancel* message. It contains a *Transmission Wrapper*, a *Control Act Wrapper* and a *Query Response*. The response contains information about the owner of the returned document as well as the content of the document. This message carries in IHE QED the result on a query and was adapted in case of HQMF results or EN13606 content. As the "Get Care Record Profile Query" also the "Get Care Record Profile Response" sends a *Transmission Wrapper* and a *Control Act Wrapper*.

The first message to look into is the *Transmission Wrapper*. An example can be seen below. All colored lines are remaining to be explicitly affected by the IHE QED profile. Furthermore the green marked lines were adapted for the IHE QEDext profile. It was necessary to introduce a new HL7 message called "Get Care Record Profile Response extension" which leads to a new "QUPC_IN043200UVext" name which has to be changed within the *Transmission Wrapper*. The transmission wrapper to be adapted is the based on the HL7 message "MCCI_MT000300UV01".

Within validation schema of the "MCCI_MT000300UV01" HL7 message it is shown that this message was not affected by the changes made for the IHE QEDext profile ("QUPC_IN043200UVext"). Therefore the Transmission Wrapper will be not described in more detail. To distinguish clearly between these two messages: "QUPC_IN043200UV" describes the usage of the *Transmission Wrapper* and the *Control Act Wrapper*. This needed adaption within the *Transmission Wrapper* part as the message got a new name and root-ID. The *Transmission Wrapper* definition itself was left unchanged as the exchange of name and root-ID is still a valid "MCCI_MT000300UV01" message.

```
<QUPC_IN043200UVext xmlns="urn:hl7-org:v3" ITSVersion="XML_1.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <id root=' ' extension=' '/>
```

```
<creationTime value=' '/>
<interactionId extension='QUPC_IN043200UVext' root='2.16.840.1.113883.5'/>
<processingCode code='D|P|T'/>
<processingModeCode code='T'/>
<acceptAckCode code='NE'/>
<receiver typeCode="RCV">
  <device determinerCode="INSTANCE">
    <id/>
    <name/>
    <telecom value=' ' />
    <manufacturerModelName/>
    <softwareName/>
  </device>
</receiver>
<sender typeCode="SND">
  <device determinerCode="INSTANCE">
    <id/>
    <name/>
    <telecom value=' ' />
    <manufacturerModelName/>
    <softwareName/>
  </device>
</sender>
<controlActProcess>
 See Control Act Wrapper below
</controlActProcess>
</QUPC_IN043200UV>
```

Next to the *Transmission Wrapper* the *Control Act Wrapper* "MFMI_MT700712UV01" is sent and was adapted. The *Control Act Wrapper* carries the payload. An example of the *Control Act Wrapper* is shown below.

```
<controlActProcess moodCode="EVN">
  <id root=' ' extension=' '/>
  <code code='QUPC_TE043200UV'/>
  <effectiveTime value=' '/>
  <languageCode code=' '/>
  <authorOrPerformer typeCode=' '></authorOrPerformer>
  <subject>
    See Query Response below
  </subject>
  <queryAck>
    <queryId root=' ' extension=' '/>
    <statusCode code=' '/>
    <queryResponseCode code=' '/>
    <resultTotalQuantity value=' '/>
    <resultCurrentQuantity value=' '/>
    <resultRemainingQuantity value=' '/>
  </queryAck>
</controlActProcess>
```

Everything marked in blue is affected by the IHE QED profile and has to be used the same way specified in IHE QED. The green labeled tag is the subject tag. It carries the payload which was adapted due to the restriction that the result is in IHE QED a patient centered CDA/CCD document which does not allow carrying EN13606 content and also leads to problems regarding transportation of population based results. The following example is an example for a payload "REPC_MT004000UV" already adapted for IHE QEDext.

```
<subject>
  <registrationEvent>
    <statusCode code='active'/>
    <custodian>
      <assignedEntity>
        <id root=" extension="/>
```

```
      <addr></addr>
      <telecom></telecom>
      <assignedOrganization>
        <name></name>
      </assignedOrganization>
    </assignedEntity>
  </custodian>
  <subject2>
    <careProvisionEvent>
      <recordTarget>
        <patient>
          <id root=" extension="/>
          <addr></addr>
          <telecom value=" use="/>
          <statusCode code='active'/>
          <patientPerson>
            <name></name>
            <administrativeGenderCode code=" displayName="
              codeSystem='2.16.840.1.113883.5.1' codeSystemName='AdministrativeGender'/>
            <birthTime value="/>
          </patientPerson>
        </patient>
      </recordTarget>
      <pertinentInformation3>
        <!-- Domain Content -->
      </pertinentInformation3>
    </careProvisionEvent>
    <parameterList>
    </parameterList>
  </subject2>
 </registrationEvent>
</subject>
```

The fields marked green are the fields to be adapted by this profile. The field "<pertinentInformation3>" carries the CDA based payload. This message had to be adapted to carry also EN13606 based content as well as population based results. The results are coded in CDA format or EN13606. As the "<parameterList>" field should be the same as in the query it has also to follow the same structure. Therefore also the query based on EN13606, the population based query or the adapted option list in the *Care Provision Code* have to be respected. The "<patient>" tag must be switched to optional as it will be possible that the query is not patient centered and therefore does not contain a single patient. This modification will lead to an adaption of the *Care Provision Event* ("REPC_MT004000UV01") defined in HL7v3. The *Pertinent Information 3* field is switched to a new defined ACT derived from HL7 RIM to ether respect the *Pertinent Information 3* as it was defined for IHE QED, but also a result set of a population based query which may be a CDA document or also an EN13606 based content.

General Query Activate Query Continue: This message is sent when a *Clinical Data Consumer* needs more information on a query. This message also contains *Transmission Wrapper* "MCCI_MT000300UV01" and *Control Act Wrapper* "QUQI_MT000001UV01" as well as a continuation request holding information about further queries.

```
<QUQI_IN000003UV01ext xmlns="urn:hl7-org:v3" ITSVersion="XML_1.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <id root=' ' extension=' '/>
 <creationTime value=' '/>
 <interactionId extension='QUQI_IN000003UV01' root='2.16.840.1.113883.5'/>
 <processingCode code='D|P|T'/>
 <processingModeCode code='T'/>
 <acceptAckCode code='AL'/>
 <receiver typeCode="RCV">
  <device determinerCode="INSTANCE">
```

```
    <id/>
    <name/>
    <telecom value=' ' />
    <manufacturerModelName/>
    <softwareName/>
  </device>
 </receiver>
 <sender typeCode="SND">
  <device determinerCode="INSTANCE">
    <id/>
    <name/>
    <telecom value=' ' />
    <manufacturerModelName/>
    <softwareName/>
  </device>
 </sender>
 <controlActProcess>
  See Control Act Wrapper below
 </controlActProcess>
</QUQI_IN000003UV01>
```

All colored lines are remaining to be explicitly affected by the IHE QED profile. It was necessary to introduce a new HL7 message called "General Query Activate Query Continue extension" which leads to a new "QUQI_IN000003UV01ext" name which had to be changed within the *Transmission Wrapper*. The transmission wrapper adapted is based on the HL7 message "MCCI_MT000300UV01". Within validation schema of the "MCCI_MT000300UV01" HL7 message it is shown that this message is not affected by the changes made for the IHE QEDext profile ("QUQI_IN000003UV01ext"). Therefore the *Transmission Wrapper* is not described in more detail. To distinguish clearly between these two messages: "QUQI_IN000003UV01" describes the usage of the *Transmission Wrapper* and the *Control Act Wrapper*. This needs adaption within the *Transmission Wrapper* part as the message gets a new name and root-ID. The *Transmission Wrapper* definition itself is left unchanged as the exchange of name and root-ID is still a valid "MCCI_MT000300UV01" message. The Transmission Wrapper does not carry any medical information. The control act process is explained within the following.

```
<controlActProcess moodCode="RQO">
  <id root=' ' extension=' '/>
  <code code='QUQI_TE000003UV01'/>
  <effectiveTime value=' '/>
  <languageCode code=' '/>
  <authorOrPerformer typeCode=' '></authorOrPerformer>
  <queryContinuation>
    <queryId root=' ' extension=' '/>
    <statusCode code='waitContinuedQueryResponse'/>
    <startResultNumber value=' '/>
    <continuationQuantity value=' '/>
  </queryContinuation>
</controlActProcess>
```

Also the *Control Act Wrapper* ("QUQI_MT000001UV01") just refers to the current query and states about continuing result delivery. It does not contain medical or direct query related data. Therefore the *Control Act* is left unchanged for the IHE QEDext profile and does not need to be further discussed in detail here.

General Query Activate Query Cancel: This message is sent when the query of a *Clinical Data Consumer* needs no further information. This message also contains *Transmission Wrapper* "MCCI_MT000300UV01" and *Control Act Wrapper* "QUQI_MT000001UV01" as well as a cancellation request including information on cancellation of the connection.

```
<QUQI_IN000003UV01 xmlns="urn:hl7-org:v3" ITSVersion="XML_1.0"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<id root=' ' extension=' '/>
<creationTime value=' '/>
<interactionId extension='QUQI_IN000003UV01' root='2.16.840.1.113883.5'/>
<processingCode code='D|P|T'/>
<processingModeCode code='T'/>
<acceptAckCode code='AL'/>
<receiver typeCode="RCV">
  <device determinerCode="INSTANCE">
    <id/>
    <name/>
    <telecom value=' ' />
    <manufacturerModelName/>
    <softwareName/>
  </device>
</receiver>
<sender typeCode="SND">
  <device determinerCode="INSTANCE">
    <id/>
    <name/>
    <telecom value=' ' />
    <manufacturerModelName/>
    <softwareName/>
  </device>
</sender>
<controlActProcess>
 See Control Act Wrapper below
</controlActProcess>
</QUQI_IN000003UV01>
```

All colored lines are remaining to be explicitly affected by the IHE QED profile. It was necessary to introduce a new HL7 message called “General Query Activate Query Cancel extension” which leads to a new “QUQI_IN000003UV01ext” name which has to be changed within the *Transmission Wrapper*. The *Transmission Wrapper* to be adapted is based on the HL7 message “MCCI_MT000300UV01”. Within validation schema of the “MCCI_MT000300UV01” HL7 message it is shown that this message is not affected by the changes made for the IHE QEDext profile (“QUQI_IN000003UV01ext”). Therefore the *Transmission Wrapper* is not described in more detail. The *Transmission Wrapper* does not carry any medical information. The control act process is explained within the following.

```
<controlActProcess moodCode="RQO">
  <id root=' ' extension=' '/>
  <code code='QUQI_TE000003UV01'/>
  <effectiveTime value=' '/>
  <languageCode code=' '/>
  <authorOrPerformer typeCode=' '></authorOrPerformer>
  <queryContinuation>
    <queryId root=' ' extension=' '/>
    <statusCode code='aborted'/>
    <startResultNumber value=' '/>
    <continuationQuantity value='0'/>
  </queryContinuation>
</controlActProcess>
```

Also the *Control Act Wrapper* (“QUQI_MT000001UV01”) just refers to the current query and states about canceling of result delivery. It does not contain medical or direct query related data. Therefore the *Control Act* is left unchanged for the IHE QEDext profile and does not need to be further discussed here.

4 OPEN SOURCE TOOLSET

This section shows how the Technical Interoperability Layer is placed within the SALUS system and how the components are technically implemented as Open Source Toolset. For technical

interoperability two major components are placed within the core. The first component is the Technical Interoperability Layer Data Service (TILDS). It accepts several data formats based on already know standards or extensions developed within this project. These standards focus on IHE profiles. TILDS will for example be triggered by a tool as the Adverse Drug Event Notification Tool (ANT) or also by the ICSR reporting tool. If one of those tools needs medical information it can trigger TILDS by using IHE QEDext, IHE CMext or next to TILDS also a SPARQL endpoint to query (as described in D4.4.2).

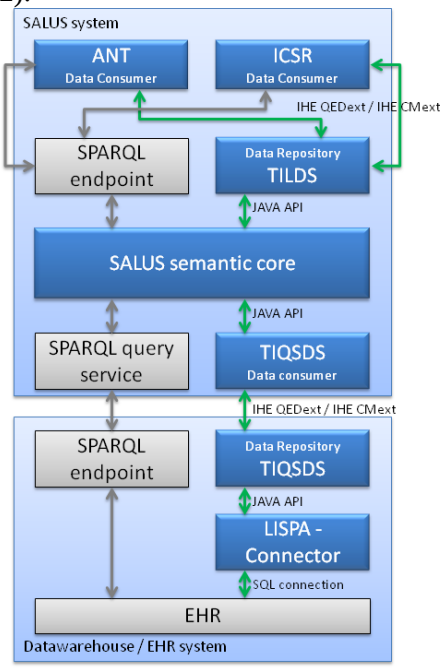

Figure 3: Overview of TILDS and TIQSDS in SALUS system

Figure 3 gives a simplified overview of the SALUS system focusing on TILDS and Technical Interoperability Query Source Data Service (TIQSDS) in case of an IHE QEDext transaction. Gray boxes or arrows within the graph are only placed for orientation and completeness, but do not affect the IHE profile part or TIL. A QEDext query or IHE CMext subscriptions is always initiated by a Data Consumer. In case of this example this might be the ANT or ICSR reporting tool. Both tools would also be allowed to use SPARQL, but this will be not explained here in more detail. Please find more information about semantic queries in D4.4.1. Regarding standard based queries the Data Consumer queries the Data Repository of TILDS. This extracts the payload of the IHE QEDext transaction including the parameterList and sends it to the SALUS semantic interoperability layer. The semantic layer decides what kind of data source to query in the end. This could be for example a SPARQL endpoint, an EHR system connected by IHE QEDext or an EHR connected by IHE CMext. In all cases the SALUS semantic layer generates the query in the standard needed. In case of IHE QEDext / IHE CMext it would be the same parameter list as already accepted from TILDS. If the query was routed from SALUS through TIQSDS the result will be generated by the LISPA-Connector and returned from the Data Repository Connector to the Data Consumer of SIL. The LISPA-connector is the component implemented by SALUS project on top of LISPA data warehouse so that it can receive and queries from SALUS Technical Interoperability Layer, and produce HL7 CCD results. It is described in detail in Deliverable 5.1.2.

SIL checks if the querying party was also using IHE QEDext / CMext or SPARQL. In case of IHE profiles the result can be untranslated forwarded to the ANT/ISCR tool. Otherwise the result has to be translated to SPARQL result before returning the result.

The components described within the following are parts of TIL and therefore are focusing on the IHE profile parts of SALUS. These parts are TILDS and TIQSDS and its subcomponents. As some of the components are also related to D5.1.2 for subscription based queries these sections will refer to this document and not described here.

4.1.1 Workflow Management

It is the main router for workflow messages in between TIL components. It was designed to route workflow messages through TILDS or TIQDS to external sources, but can also be used to route workflow messages from one internal source to an internal target. The workflow management is further described in detail in D5.1.2.

4.1.2 Technical Interoperability Layer

As already described TIL consist of two components which are the Technical Interoperability Layer Data Service (TILDS) and the Technical Interoperability Query Source Data Service (TIDQS). However these components are mostly identically from implementation point of view as both components represent a Clinical Data Consumer and a Clinical Data Repository and both components implement the SALUS IHE profiles IHE QEDext and IHE CMext. The biggest difference between these components is the way they are used in SALUS. TILDS offers a third party application a way to query information from SALUS. Therefore the Clinical Data Consumer from IHE profile point of view is an external party. These external parties can also be systems using IHE CM and IHE QED (the original unmodified profiles) to request information. This allows SALUS to be compatible to external systems already using these profiles in real environments. SALUS takes role of a *Clinical Data Repository* in case of TILDS. In case of TIQSDS it is the other way round. If SALUS wants to get information from an external EHR system, it takes the role of a *Clinical Data Consumer* to query an IHE QEDext, IHE QED, IHE CMext, IHE CM supporting system. Therefore it was necessary to implement in SALUS the *Clinical Data Consumer* as well as the *Clinical Data Repository* for both IHE QEDext and IHE CMext.
Within the following the structure of the IHE QEDext module is described in detail as this is the profile supporting querying for information. IHE CMext is described more in detail in D5.1.2 which is focusing on subscribing to data.

4.1.2.1 Structure of IHE QEDext Module

The QED Profile enables two actors in the health care, named as Clinical Data Consumer and Clinical Data Source, to exchange patient or medical information through the usage of standardized transaction messages (see Figure 4). Therefore it is the purpose of the QED Module to implement the IHE QED Profile and extend it by features missing, like the ability to specify inclusion and exclusion criteria for listing patient groups in HQMF (in case of a population based query) or EN13606 related queries.

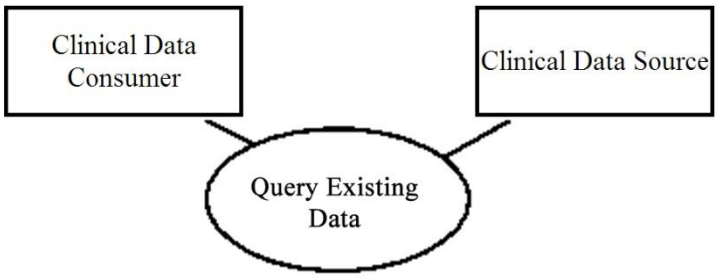

Figure 4: Overview of the actors participating in the QED Profile

The transaction messages being made are based upon HL7 standards and composed of two parts: Firstly, a transaction wrapper providing metadata of the message to be sent (like the message type) and information about the sender and receiver, and secondly, the payload of the message itself, containing for example the actual query. It is not the task of the module to generate this payload, but to construct the necessary transaction wrapper and to implement the communication.

1) An external tool as the ANT should not directly query the Datawarehouse/ EHR system. This work is done by the SALUS system. The ANT just has to query what it needs to know by sending the query (as a parameter list) to the IHE QEDext module in TILDS (see Figure 3:

Overview of TILDS and TIQSDS in SALUS system), such as the small example below. It indicates whose patient data is requested and what is being looked for, in this case all recorded vital signs.

```
<parameterList>
    <careProvisionCode>
        <value code="COBSCAT" codeSystem="ActCode"/>
    </careProvisionCode>
    <patientId>
        <value extension="somePatientId" root="someDirectory"/>
    </patientId>
</parameterList>
```

Figure 5: Example of a parameter list

Although the module has received the parameter list it cannot perform the query on the EHR itself. The parameter list has to be forwarded to the SALUS semantic layer. Once the Semantic Interoperability Layer has got the query it has to decide which underlying service (SPARQL query service or TIQSDS) should be called based on the type of the EHR System.

2) The TIQSDS has been chosen to perform the query, but it does not generate and send the query message itself. It delegates these tasks to the QED Module (see Figure 3: Overview of TILDS and TIQSDS in SALUS system). To transmit the message the module uses its web service interface since the receiver is located outside of the system.

3) A valid query transaction message has been received and forwarded to the LISPA connector which will query the EHR system itself by means of SQL statements (see Figure 3: Overview of TILDS and TIQSDS in SALUS system). As soon as the results arrived at the QED Module they are resend back.

The way back from the Datawarehouse/ EHR system to the user is pretty the same, but this time the transmitted payload is now the response in the form of an HL7 CareEntry or even an EN13606 document.

In summary these following requirements have to be fulfilled by the module:

- R0) It is possible to use the QED Module on different locations within the SALUS system
- R1) The QED Module has to generate transaction messages for the Clinical Data Consumer
- R2) The QED Module has to generate transaction messages for the Clinical Data Source
- R3) The QED Module has to offer a web service interface to be addressed from the outside of the SALUS system
- R4) The QED Module has to be implemented as a workflow service to be addressed from the inside of the SALUS system
- R5) The QED Module has to be able to initiate a new workflow to communicate with other components within the SALUS system (like SALUS Semantic Interoperability Layer)
- R6) The QED Module has to transmit messages over its web service interface if the receiver is located outside of the SALUS system.

Implementation

IHE profiles are based on webservices therefore the communication between two actors is done in XML format. The HL7 has defined a huge set of XSD schema files derived from HL7 RIM representing syntax and semantic of HL7 messages to be sent between actors. To ensure stable communication between *Clincal Data Consumer* and *Clinical Data Repository* a tool was used to transform the HL7 schema files to corresponding JAVA classes. These JAVA classes allow creation of the XML documents out of the payload to be sent and also to parse the XML documents to get the payload transported through XML in JAVA classes. By using the same JAVA classes on both sides (sender and receiver) it is possible to ensure that data is translated to XML without harming the HL7 XSD schemas and the interpretation of the payload on both sides is the same.
In the beginning of this document it was already described how IHE QED profile was changed to fit the SALUS use cases. Therefore the XSD files of the corresponding QED HL7 messages have been adapted. Afterwards the translation between HL7 schema files and JAVA classes have been made. Therefore the JAVA classes do automatically support the IHE QED extension as described in the document. Within the following the important components the QEDext profile implementation will be described more in detail.

HL7 Transaction message generation

For extending the IHE QED profile in general two of the HL7 messages have been adapted. These are:

- **Get Care Record Profile Query EXT (QUPC_TE043100UVext):** This message is transporting the query from the *Clinical Data Consumer* to the *Clinical Data Repository*. This message is holding the parameter list with the query to be answered. As the query was adapted to support also population based queries, also the message needs to be extended.
- **Get Care Record Profile Response EXT (QUPC_TE043200UVext):** This message is holding the medical data to be returned from Clinal Data Repository to Clinical Data Consumer. Next to transmission wrapper and control act wrapper it is carrying the result set ether filled to CCD templates or EN13606 document type.

Within the following these two messages and the way how they have to be used will be described more in detail within the following.

Get Care Record Profile Query EXT (QUPC_TE043100UVext)
This message generation can be evoked by instantiating the the QedQueryBuilder-Object and is described exemplary.

```
// assume someone gives you the parameterList as string
II senderId = Helper.createII("offis", "some.sender");
II receiverId = Helper.createII("offis", "some.receiver");
StringBuilder str = new StringBuilder();
str.append("<parameterList xmlns=\"urn:hl7-org:v3\">");
str.append("<patientId><value                                    root=\"01234\"
extension=\"56789\"/></patientId>");
str.append("<careProvisionCode><value                              code=\"1+2\"
codeSystem=\"3\"/>3</careProvisionCode>");
str.append("</parameterList>");

builder = new QedQueryBuilder(senderId,
      receiverId,
      Helper.createXmlElement(QUPCMT040300UV01ParameterList.class,
            str.toString()));

QUPCIN043100UV01 message = builder.build();
```

The constructor to create a QedQeueryMessage expects a QedQueryTemplate. The constructor demands three parameter of the user: The identification of the sending entity (senderId), the identification of the receiving entity (receiverId) and the query itself (parameterList). The parameterList refines the search area of the query, e.g. by specifying that only results in a particular point in time are relevant. These parameters are necessary in order to build a minimized message. If the user wants to add more information he/she just has to call the appropriate setter methods. The string variable xmlQueryParameter holds the parameterList, but an instance of Document-class is needed. Therefore the string has to be transformed by the createDocument method. The returned object implements the QedQueryDocument interface which is a subtype of Document. The generated XML document itself looks like following:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<QUPC_IN043100UV01 xmlns="urn:hl7-org:v3" ITSVersion="XML_1.0">
    <id                    extension="854817d6-2ff4-400a-9e3b-36df11f929f8"
root="1.2.276.0.7230010"/>
    <creationTime value="20121208142707"/>
    <interactionId extension="QUPC_IN043100UV" root="2.16.840.1.113883.5"/>
    <processingModeCode    code="T"    codeSystem="2.16.840.1.113883.5.101"
codeSystemName="ProcessingMode"/>
    <acceptAckCode      code="AL"      codeSystem="2.16.840.1.113883.5.1050"
codeSystemName="AcknowledgementCondition"/>
    <receiver typeCode="RCV">
        <device classCode="DEV" determinerCode="INSTANCE">
            <id extension="some.receiver" root="offis"/>
        </device>
    </receiver>
    <sender typeCode="SND">
        <device classCode="DEV" determinerCode="INSTANCE">
            <id extension="some.receiver" root="offis"/>
        </device>
    </sender>
    <controlActProcess moodCode="RQO">
        <code code="QUPC_TE043100UV"/>
        <queryByParameter>
            <queryId       extension="5b5da969-854b-439d-ab33-4f21cfec20f3"
root="1.2.276.0.7230010"/>
            <statusCode code="new"/>
            <responseModalityCode code="R"/>
            <responsePriorityCode code="I"/>
            <initialQuantity value="1"/>
            <initialQuantityCode                    code="REPC_RM000100UV"
codeSystem="2.16.840.1.113883.5"/>
            <parameterList>
                …
            </parameterList>
        </queryByParameter>
    </controlActProcess>
</QUPC_IN043100UV01>
```

Get Care Record Profile Response EXT (QUPC_TE043200UVext)

This message generation can be evoked by instantiating the QedResponseBuilder Object and is described exemplary.

```
// assume someone gives you the parameterList as string
II senderId = Helper.createII("offis", "some.sender");
II receiverId = Helper.createII("offis", "some.receiver");
II queryId = Helper.createII("offis", "some.receiver");
```

```
QUPCIN043200UV01MFMIMT700712UV01RegistrationEvent observation =
      Helper.createXMLElement(QUPCIN043200UV01MFMIMT700712UV01RegistrationE
vent.class,
      QedResponseBuilderTest.class.getResourceAsStream("/templates/registra
tionEvent.xml"));

builder = new QedResponseBuilder(senderId,
      receiverId,
      queryId,
      Arrays.asList(observation));

QUPCIN043200UV01 message = builder.build();
```

The constructor to create a QedResponseMessage expects a QedResponseTemplate. The constructor demands three parameter of the user: The identification of the sending entity (senderId), the identification of the receiving entity (receiverId) and a set of responses. This can be seen in the code above as ArrayList of observations. These parameters are necessary in order to build a minimized message. If the user wants to add more information he/she just has to call the appropriate setter methods. The following is showing an exemplary XML document as created by the response builder.

```
//responseMessage
<QUPC_IN043200UV01 xmlns="urn:hl7-org:v3" ITSVersion="XML_1.0">
  <id root="6cc85f95-db46-4bc9-b3a8-b29bed8c4b76"/>
  <interactionId root="2.16.840.1.113883.5" extension="QUPC_IN043200UV"/>
  <processingModeCode     code="T"     codeSystem="2.16.840.1.113883.5.101"
codeSystemName="ProcessingMode"/>
  <acceptAckCode       code="NE"       codeSystem="2.16.840.1.113883.5.1050"
codeSystemName="AcknowledgementCondition"/>
  <receiver typeCode="RCV">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id root="SALUS-Project" extension="QEDExt module -- sender"/>
    </device>
  </receiver>
  <sender typeCode="SND">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id root="SALUS-Project" extension="QEDExt module -- receiver"/>
    </device>
  </sender>
  <controlActProcess moodCode="EVN">
    <code code="QUPC_TE043200UV"/>
    <subject>
      <registrationEvent>
        ...
      </registrationEvent>
      <registrationEvent>
        ...
      </registrationEvent>
    </subject>
    <queryAck>
      <queryId root="c1b5b1b9-d3bf-47e0-99db-5d23626c0042"/>
      <statusCode code="deliveredResponse"/>
      <queryResponseCode code="OK"/>
      <resultTotalQuantity value="2"/>
      <resultCurrentQuantity value="2"/>
      <resultRemainingQuantity value="0"/>
    </queryAck>
  </controlActProcess>
</QUPC_IN043200UV01>
```

The code snipped shows two “registration events”. These sections are representing the list elements of the ArrayList holding the observation. So each registration event is one of the array list observations, even though a registration event can contain more than one observation. Within SALUS a registration event can contain up to all observations regarding on single patient.

4.1.2.2 Parsing of QEDext query objects (parameter list)

The parser component of the QED has to understand the query send by the *Data Consumer* semantically and to advice the EHR to return the resulting set of data. By the XSD schema files it was already checked that the query is syntactically valid. In the next step the parser figures out if the query is defined as population based query or just query directly for patient data. Therefore the query must not contain a patient ID and also needs to have a filled HQMF tag for defining population based query.
In case of a population based query the parser starts to analyze the temporal patterns which are be defined as shown in Figure 6. Therefore two different temporal queries are defined which were extracted from HQMF. The first is called “Condition Starts after of Start of Medication” and searches for patients having a condition/medication temporal relation like:

1. *In case of positive time window:* The condition starts after the medication starts with a temporal delay bigger than the time window specified.
2. *In case of negative time window:* The condition starts before the medication in range of the time window specified.

Both can be seen in the Figure 6 (top). The second temporal relation defined is called “Condition Starts before of Start of Medication” and searches for a condition/medication temporal relation in type of:

1. *In case of positive time window:* The condition starts within the range of the time window after the medication was taken.
2. *In case of negative time window:* The medication starts with a delay of time window size after condition ends.

Both can be seen in the Figure 6 (bottom).

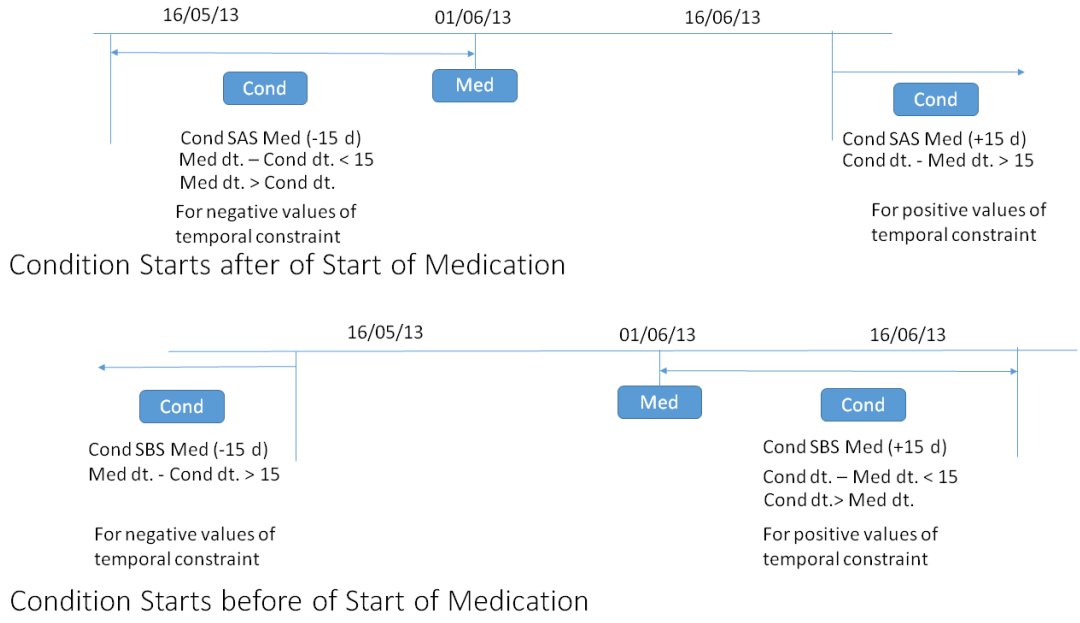

Figure 6: The figure gives an overview of the temporal relation queries

In the end these temporal relation queries will end up in 1..n patient ids. These list is used by the QED parser as if it is a list of patients which was directly queried for by QED. Within the parameter list it was already stated which options of these patient is searched for. These can be:

- Intolerances
- Problem list
- Medication list
- Immunizations
- Professional Services Category (ambulatory and hospitalization information)
- Pregnancy

If the list has been left empty, all of the information will be returned in one registration event per patient. The mapping of the data ware house fields to the CCD templates can be found in D5.2.2 in section “LISPA connector”.

4.1.3 LISPA Connector

The LISPA connector is a module to execute queries send to TIQSDS and return the results. Queries are checked by the connector and in case of an exception rejected with an appropriate HL7 message. In case of a valid query, the connector queries the EHR system of LISPA and puts the results to valid templates based on CDA or EN13606. The LISPA connector is described in detail in D5.1.2.

5 CONCLUSION

Within the first 2 years of the project, first of all a lot of state of the art analysis has been made. One of the results was that using IHE profiles for communication between Data Consumer and Data Repositories is the most common way in especially if the communication should be established between systems of different vendors. Among these, the IHE QED profile is one of the best fitting. For addressing a query based communication protocol between safety analysis systems and EHR systems SALUS is using the PCC-1 transaction of the PCC technical framework. As this transaction fulfils already all needs regarding the queries of patient centred data, this one was chosen. Next to patient centred information also population based queries have to be performed. Therefore a subset of HQMF is used for querying temporal relationships between conditions and medications of certain patients.

Next to the extension of the IHE QED profile by population based queries also EN13606 will be transported. As EN13606 does currently not define its own transport profiles and no other are available, it was decided to add also an EN13606 related data field to the PCC-1 transaction of the IHE QED profile.

At the end only one common query based profile was adapted by adding population based queries and transportation of EN13606 based data to fulfil SALUS requirements in case of querying data in a synchronous way.

6 APPENDIX

Below the modified HL7 XSD schema files can be found which needed to be adapted to support either population based querying as well as EN13606 transportation. To extend IHE CM and IHE QED three messages have been adapted.

Message schema: MCCI_MT000300UV01

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ex="urn:hl7-org/v3-
example"
           xmlns="urn:hl7-org:v3"
           targetNamespace="urn:hl7-org:v3"
           elementFormDefault="qualified"><!--
********************************************************************************
*****************************
* XML schema for message type MCCI_MT000300UV01.
* Source information:
*     Rendered by: RoseTree 4.2.7
*     Rendered on:
* HMD was rendered into XML using software provided to HL7 by Beeler Consulting
LLC.
 HMD to MIF Transform: $Id: RoseTreeHmdToMIFStaticModel.xsl 49 2009-03-25 15:09:56Z
Woody $
  Base transform: $Id: ConvertBase.xsl 8 2009-01-26 19:10:36Z Woody $
  Package Id Conversion: $Id: TransformPackageIds.xsl 8 2009-01-26 19:10:36Z Woody
$
  HTML To MIF markup: $Id: HtmlToMIFMarkup.xsl 8 2009-01-26 19:10:36Z Woody $
 Flat to Serialization Transform: $Id: MIFStaticModelFlatToSerialization.xsl 8
2009-01-26 19:10:36Z Woody $
 Fix Names Transform: $Id: FixMifNames.xsl 47 2009-03-23 00:05:58Z Woody $
  Base transform: $Id: ConvertBase.xsl 8 2009-01-26 19:10:36Z Woody $
  Package Id Conversion: $Id: TransformPackageIds.xsl 8 2009-01-26 19:10:36Z Woody
$
*
* Generated by XMLITS version 3.2.5
*   MIF to XSD Transform $Id: StaticMifToXsd.xsl 98 2009-08-08 15:53:37Z Woody $
*     Package Id Conversion: $Id: TransformPackageIds.xsl 8 2009-01-26 19:10:36Z
Woody $
*
* Copyright (c) 2002, 2003, 2004, 2005, 2006, 2007 Health Level Seven. All rights
reserved.
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. All advertising materials mentioning features or use of this software
*     must display the following acknowledgement:
*       This product includes software developed by Health Level Seven.
*  THIS SOFTWARE IS PROVIDED BY HEALTH LEVEL SEVEN, INC. AND CONTRIBUTORS "AS IS"
AND
*  ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
*  IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
*  ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
*  FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
*  DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
*  OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
*  HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
*  LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
*  OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
*  SUCH DAMAGE.
```

```
*
*********************************************************************************
*******************************
        --><xs:annotation>
      <xs:documentation>Generated using schema builder version 3.2.5. Stylesheets:
HMD was rendered into XML using software provided to HL7 by Beeler Consulting LLC.
 HMD to MIF Transform: $Id: RoseTreeHmdToMIFStaticModel.xsl 49 2009-03-25 15:09:56Z
Woody $
  Base transform: $Id: ConvertBase.xsl 8 2009-01-26 19:10:36Z Woody $
  Package Id Conversion: $Id: TransformPackageIds.xsl 8 2009-01-26 19:10:36Z Woody
$
  HTML To MIF markup: $Id: HtmlToMIFMarkup.xsl 8 2009-01-26 19:10:36Z Woody $
 Flat to Serialization Transform: $Id: MIFStaticModelFlatToSerialization.xsl 8
2009-01-26 19:10:36Z Woody $
 Fix Names Transform: $Id: FixMifNames.xsl 47 2009-03-23 00:05:58Z Woody $
  Base transform: $Id: ConvertBase.xsl 8 2009-01-26 19:10:36Z Woody $
  Package Id Conversion: $Id: TransformPackageIds.xsl 8 2009-01-26 19:10:36Z Woody
$
StaticMifToXsd.xsl version 2.0</xs:documentation>
   </xs:annotation>
   <xs:include schemaLocation="../coreschemas/infrastructureRoot.xsd"/>
   <xs:include schemaLocation="COCT_MT040203UV01.xsd"/>
   <xs:include schemaLocation="QUQI_MT020001UV01.xsd"/>
   <xs:complexType name="MCCI_MT000300UV01.Acknowledgement">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="messageWaitingNumber" type="INT" minOccurs="0"
maxOccurs="1"/>
         <xs:element name="messageWaitingPriorityCode" type="CE" minOccurs="0"
maxOccurs="1"/>
         <xs:element name="targetMessage" type="MCCI_MT000300UV01.TargetMessage"
nillable="true"
                     minOccurs="1"
                     maxOccurs="1"/>
         <xs:element name="acknowledgementDetail"
type="MCCI_MT000300UV01.AcknowledgementDetail"
                     nillable="true"
                     minOccurs="0"
                     maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
      <xs:attribute name="typeCode" type="AcknowledgementType" use="required"/>
   </xs:complexType>
   <xs:complexType name="MCCI_MT000300UV01.AcknowledgementDetail">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="code" type="CE" minOccurs="0" maxOccurs="1"/>
         <xs:element name="text" type="ED" minOccurs="0" maxOccurs="1"/>
         <xs:element name="location" type="ST" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
      <xs:attribute name="typeCode" type="AcknowledgementDetailType"
use="optional"/>
   </xs:complexType>
   <xs:complexType name="MCCI_MT000300UV01.Agent">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="representedOrganization"
type="MCCI_MT000300UV01.Organization"
                     nillable="true"
                     minOccurs="0"
                     maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
```

```
      <xs:attribute name="classCode" type="RoleClassAgent" use="required"/>
   </xs:complexType>
   <xs:complexType name="MCCI_MT000300UV01.AttentionLine">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="keyWordText" type="SC" minOccurs="0" maxOccurs="1"/>
         <xs:element name="value" type="ANY" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
   </xs:complexType>
   <xs:complexType name="MCCI_MT000300UV01.Device">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="id" type="II" minOccurs="1" maxOccurs="unbounded"/>
         <xs:element name="name" type="EN" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element name="desc" type="ED" minOccurs="0" maxOccurs="1"/>
         <xs:element name="existenceTime" type="IVL_TS" minOccurs="0"
maxOccurs="1"/>
         <xs:element name="telecom" type="TEL" minOccurs="0"
maxOccurs="unbounded"/>
         <xs:element name="manufacturerModelName" type="SC" minOccurs="0"
maxOccurs="1"/>
         <xs:element name="softwareName" type="SC" minOccurs="0" maxOccurs="1"/>
         <xs:element name="asAgent" type="MCCI_MT000300UV01.Agent" nillable="true"
minOccurs="0"
                     maxOccurs="1"/>
         <xs:element name="asLocatedEntity" type="MCCI_MT000300UV01.LocatedEntity"
nillable="true"
                     minOccurs="0"
                     maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="classCode" type="EntityClassDevice" use="required"/>
      <xs:attribute name="determinerCode" type="EntityDeterminerSpecific"
use="required"/>
   </xs:complexType>
   <xs:complexType name="MCCI_MT000300UV01.EntityRsp">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="id" type="II" minOccurs="1" maxOccurs="1"/>
         <xs:element name="name" type="EN" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element name="telecom" type="TEL" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
      <xs:attribute name="classCode" type="EntityClassRoot" use="required"/>
      <xs:attribute name="determinerCode" type="EntityDeterminerSpecific"
use="required"/>
   </xs:complexType>
   <xs:complexType name="MCCI_MT000300UV01.LocatedEntity">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="location" type="MCCI_MT000300UV01.Place" nillable="true"
minOccurs="0"
                     maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
      <xs:attribute name="classCode" type="RoleClassLocatedEntity" use="required"/>
   </xs:complexType>
   <xs:complexType name="MCCI_MT000300UV01.Message">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="id" type="II" minOccurs="1" maxOccurs="1"/>
         <xs:element name="creationTime" type="TS" minOccurs="1" maxOccurs="1"/>
         <xs:element name="securityText" type="ST" minOccurs="0" maxOccurs="1"/>
```

```
         <xs:element name="versionCode" type="CS" minOccurs="0" maxOccurs="1"/>
         <xs:element name="interactionId" type="II" minOccurs="1" maxOccurs="1"/>
         <xs:element name="profileId" type="II" minOccurs="0"
maxOccurs="unbounded"/>
         <xs:element name="processingCode" type="CS" minOccurs="1" maxOccurs="1"/>
         <xs:element name="processingModeCode" type="CS" minOccurs="1"
maxOccurs="1"/>
         <xs:element name="acceptAckCode" type="CS" minOccurs="1" maxOccurs="1"/>
         <xs:element name="attachmentText" type="ED" minOccurs="0"
maxOccurs="unbounded"/>
         <xs:element name="receiver" type="MCCI_MT000300UV01.Receiver"
minOccurs="1"
                     maxOccurs="unbounded"/>
         <xs:element name="respondTo" type="MCCI_MT000300UV01.RespondTo"
nillable="true"
                     minOccurs="0"
                     maxOccurs="unbounded"/>
         <xs:element name="sender" type="MCCI_MT000300UV01.Sender" minOccurs="1"
maxOccurs="1"/>
         <xs:element name="attentionLine" type="MCCI_MT000300UV01.AttentionLine"
nillable="true"
                     minOccurs="0"
                     maxOccurs="unbounded"/>
         <xs:element name="acknowledgement"
type="MCCI_MT000300UV01.Acknowledgement" nillable="true"
                     minOccurs="0"
                     maxOccurs="unbounded"/>
         <!-- ControlActProcess -->
         <xs:element name="controlActProcess"
                     type="QUQI_MT020001UV01.ControlActProcess"
                     minOccurs="1"
                     maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
   </xs:complexType>
   <xs:complexType name="MCCI_MT000300UV01.Organization">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="id" type="II" minOccurs="1" maxOccurs="unbounded"/>
         <xs:element name="name" type="EN" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element name="telecom" type="TEL" minOccurs="0"
maxOccurs="unbounded"/>
         <xs:element name="notificationParty"
type="COCT_MT040203UV01.NotificationParty"
                     nillable="true"
                     minOccurs="0"
                     maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
      <xs:attribute name="classCode" type="EntityClassOrganization"
use="required"/>
      <xs:attribute name="determinerCode" type="EntityDeterminerSpecific"
use="required"/>
   </xs:complexType>
   <xs:complexType name="MCCI_MT000300UV01.Place">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="id" type="II" minOccurs="1" maxOccurs="unbounded"/>
         <xs:element name="name" type="EN" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element name="telecom" type="TEL" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
      <xs:attribute name="classCode" type="EntityClassPlace" use="required"/>
      <xs:attribute name="determinerCode" type="EntityDeterminerSpecific"
use="required"/>
```

```
   </xs:complexType>
   <xs:complexType name="MCCI_MT000300UV01.Receiver">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="telecom" type="TEL" minOccurs="0" maxOccurs="1"/>
         <xs:element name="device" type="MCCI_MT000300UV01.Device" minOccurs="1"
maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="typeCode" type="CommunicationFunctionType"
use="required"/>
   </xs:complexType>
   <xs:complexType name="MCCI_MT000300UV01.RespondTo">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="telecom" type="TEL" minOccurs="0" maxOccurs="1"/>
         <xs:element name="entityRsp" type="MCCI_MT000300UV01.EntityRsp"
nillable="true"
                     minOccurs="1"
                     maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
      <xs:attribute name="typeCode" type="CommunicationFunctionType"
use="required"/>
   </xs:complexType>
   <xs:complexType name="MCCI_MT000300UV01.Sender">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="telecom" type="TEL" minOccurs="0" maxOccurs="1"/>
         <xs:element name="device" type="MCCI_MT000300UV01.Device" minOccurs="1"
maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="typeCode" type="CommunicationFunctionType"
use="required"/>
   </xs:complexType>
   <xs:complexType name="MCCI_MT000300UV01.TargetMessage">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="id" type="II" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
   </xs:complexType>
</xs:schema>
```

Message schema: QUPC_MT040300UV01

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ex="urn:hl7-org/v3-
example"
           xmlns="urn:hl7-org:v3"
           targetNamespace="urn:hl7-org:v3"
           elementFormDefault="qualified"><!--
********************************************************************************
*****************************
* XML schema for message type QUPC_MT040300UV01.
* Source information:
*     Rendered by: RoseTree 4.2.7
*     Rendered on:
* HMD was rendered into XML using software provided to HL7 by Beeler Consulting
LLC.
 HMD to MIF Transform: $Id: RoseTreeHmdToMIFStaticModel.xsl 49 2009-03-25 15:09:56Z
Woody $
  Base transform: $Id: ConvertBase.xsl 8 2009-01-26 19:10:36Z Woody $
```

```
  Package Id Conversion: $Id: TransformPackageIds.xsl 8 2009-01-26 19:10:36Z Woody
$
  HTML To MIF markup: $Id: HtmlToMIFMarkup.xsl 8 2009-01-26 19:10:36Z Woody $
 Flat to Serialization Transform: $Id: MIFStaticModelFlatToSerialization.xsl 8
2009-01-26 19:10:36Z Woody $
 Fix Names Transform: $Id: FixMifNames.xsl 47 2009-03-23 00:05:58Z Woody $
  Base transform: $Id: ConvertBase.xsl 8 2009-01-26 19:10:36Z Woody $
  Package Id Conversion: $Id: TransformPackageIds.xsl 8 2009-01-26 19:10:36Z Woody
$
*
* Generated by XMLITS version 3.2.5
*   MIF to XSD Transform $Id: StaticMifToXsd.xsl 98 2009-08-08 15:53:37Z Woody $
*     Package Id Conversion: $Id: TransformPackageIds.xsl 8 2009-01-26 19:10:36Z
Woody $
*
* Copyright (c) 2002, 2003, 2004, 2005, 2006, 2007 Health Level Seven. All rights
reserved.
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. All advertising materials mentioning features or use of this software
*     must display the following acknowledgement:
*       This product includes software developed by Health Level Seven.
*  THIS SOFTWARE IS PROVIDED BY HEALTH LEVEL SEVEN, INC. AND CONTRIBUTORS "AS IS"
AND
*  ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
*  IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
*  ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
*  FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
*  DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
*  OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
*  HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
*  LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
*  OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
*  SUCH DAMAGE.
*
********************************************************************************
*******************************
        --><xs:annotation>
      <xs:documentation>Generated using schema builder version 3.2.5. Stylesheets:
HMD was rendered into XML using software provided to HL7 by Beeler Consulting LLC.
 HMD to MIF Transform: $Id: RoseTreeHmdToMIFStaticModel.xsl 49 2009-03-25 15:09:56Z
Woody $
  Base transform: $Id: ConvertBase.xsl 8 2009-01-26 19:10:36Z Woody $
  Package Id Conversion: $Id: TransformPackageIds.xsl 8 2009-01-26 19:10:36Z Woody
$
  HTML To MIF markup: $Id: HtmlToMIFMarkup.xsl 8 2009-01-26 19:10:36Z Woody $
 Flat to Serialization Transform: $Id: MIFStaticModelFlatToSerialization.xsl 8
2009-01-26 19:10:36Z Woody $
 Fix Names Transform: $Id: FixMifNames.xsl 47 2009-03-23 00:05:58Z Woody $
  Base transform: $Id: ConvertBase.xsl 8 2009-01-26 19:10:36Z Woody $
  Package Id Conversion: $Id: TransformPackageIds.xsl 8 2009-01-26 19:10:36Z Woody
$
StaticMifToXsd.xsl version 2.0</xs:documentation>
   </xs:annotation>
   <xs:include schemaLocation="../coreschemas/infrastructureRoot.xsd"/>
   <xs:include schemaLocation="../hqmf/POQM_MT000001UVNew.xsd"/>
   <xs:complexType name="PopulationBasedQueryType">
      <xs:sequence>
         <xs:element name="qualityMeasureDocument"
type="POQM_MT000001UV.QualityMeasureDocument"/>
      </xs:sequence>
   </xs:complexType>
```

```
   <xs:complexType name="QUPC_MT040300UV01.CareProvisionCode">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="value" type="CD" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
   </xs:complexType>
   <xs:complexType name="QUPC_MT040300UV01.CareProvisionReason">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="value" type="CD" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
   </xs:complexType>
   <xs:complexType name="QUPC_MT040300UV01.CareRecordTimePeriod">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="value" type="IVL_TS" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
   </xs:complexType>
   <xs:complexType name="QUPC_MT040300UV01.ClinicalStatementTimePeriod">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="value" type="IVL_TS" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
   </xs:complexType>
   <xs:complexType name="QUPC_MT040300UV01.IncludeCarePlanAttachment">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="value" type="BL" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
   </xs:complexType>
   <xs:complexType name="QUPC_MT040300UV01.MaximumHistoryStatements">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="value" type="INT" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
   </xs:complexType>
   <xs:complexType name="QUPC_MT040300UV01.ParameterList">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="careProvisionCode"
type="QUPC_MT040300UV01.CareProvisionCode"
                     nillable="true"
                     minOccurs="0"
                     maxOccurs="1"/>
         <xs:element name="careProvisionReason"
type="QUPC_MT040300UV01.CareProvisionReason"
                     nillable="true"
                     minOccurs="0"
                     maxOccurs="unbounded"/>
         <xs:element name="careRecordTimePeriod"
type="QUPC_MT040300UV01.CareRecordTimePeriod"
                     nillable="true"
                     minOccurs="0"
                     maxOccurs="1"/>
         <xs:element name="clinicalStatementTimePeriod"
                     type="QUPC_MT040300UV01.ClinicalStatementTimePeriod"
```

```
                     nillable="true"
                     minOccurs="0"
                     maxOccurs="1"/>
         <xs:element name="includeCarePlanAttachment"
                     type="QUPC_MT040300UV01.IncludeCarePlanAttachment"
                     nillable="true"
                     minOccurs="0"
                     maxOccurs="1"/>
         <xs:element name="maximumHistoryStatements"
                     type="QUPC_MT040300UV01.MaximumHistoryStatements"
                     nillable="true"
                     minOccurs="0"
                     maxOccurs="1"/>
         <xs:element name="patientAdministrativeGender"
                     type="QUPC_MT040300UV01.PatientAdministrativeGender"
                     nillable="true"
                     minOccurs="0"
                     maxOccurs="1"/>
         <xs:element name="patientBirthTime"
type="QUPC_MT040300UV01.PatientBirthTime"
                     nillable="true"
                     minOccurs="0"
                     maxOccurs="1"/>
         <xs:choice>
            <xs:element name="patientId" type="QUPC_MT040300UV01.PatientId"
minOccurs="0" maxOccurs="1"/>
             <xs:element name="populationBasedQuery"
type="PopulationBasedQueryType"/>
             <xs:element name="queryByEn13606" type="xs:anyType"/>
         </xs:choice>
         <xs:element name="patientName" type="QUPC_MT040300UV01.PatientName"
nillable="true"
                     minOccurs="0"
                     maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
   </xs:complexType>
   <xs:complexType name="QUPC_MT040300UV01.PatientAdministrativeGender">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="value" type="CE" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
   </xs:complexType>
   <xs:complexType name="QUPC_MT040300UV01.PatientBirthTime">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="value" type="TS" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
   </xs:complexType>
   <xs:complexType name="QUPC_MT040300UV01.PatientId">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="value" type="II" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
   </xs:complexType>
   <xs:complexType name="QUPC_MT040300UV01.PatientName">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="value" type="PN" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
   </xs:complexType>
```

```
</xs:schema>
```

Message schema: QUQI_MT020001UV01

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ex="urn:hl7-org/v3-
example"
           xmlns="urn:hl7-org:v3"
           targetNamespace="urn:hl7-org:v3"
           elementFormDefault="qualified"><!--
********************************************************************************
*****************************
* XML schema for message type QUQI_MT020001UV01.
* Source information:
*     Rendered by: RoseTree 4.2.7
*     Rendered on:
* HMD was rendered into XML using software provided to HL7 by Beeler Consulting
LLC.
 HMD to MIF Transform: $Id: RoseTreeHmdToMIFStaticModel.xsl 49 2009-03-25 15:09:56Z
Woody $
  Base transform: $Id: ConvertBase.xsl 8 2009-01-26 19:10:36Z Woody $
  Package Id Conversion: $Id: TransformPackageIds.xsl 8 2009-01-26 19:10:36Z Woody
$
  HTML To MIF markup: $Id: HtmlToMIFMarkup.xsl 8 2009-01-26 19:10:36Z Woody $
 Flat to Serialization Transform: $Id: MIFStaticModelFlatToSerialization.xsl 8
2009-01-26 19:10:36Z Woody $
 Fix Names Transform: $Id: FixMifNames.xsl 47 2009-03-23 00:05:58Z Woody $
  Base transform: $Id: ConvertBase.xsl 8 2009-01-26 19:10:36Z Woody $
  Package Id Conversion: $Id: TransformPackageIds.xsl 8 2009-01-26 19:10:36Z Woody
$
*
* Generated by XMLITS version 3.2.5
*   MIF to XSD Transform $Id: StaticMifToXsd.xsl 98 2009-08-08 15:53:37Z Woody $
*     Package Id Conversion: $Id: TransformPackageIds.xsl 8 2009-01-26 19:10:36Z
Woody $
*
* Copyright (c) 2002, 2003, 2004, 2005, 2006, 2007 Health Level Seven. All rights
reserved.
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*  1. Redistributions of source code must retain the above copyright
*     notice, this list of conditions and the following disclaimer.
*  2. Redistributions in binary form must reproduce the above copyright
*     notice, this list of conditions and the following disclaimer in the
*     documentation and/or other materials provided with the distribution.
*  3. All advertising materials mentioning features or use of this software
*     must display the following acknowledgement:
*       This product includes software developed by Health Level Seven.
*  THIS SOFTWARE IS PROVIDED BY HEALTH LEVEL SEVEN, INC. AND CONTRIBUTORS "AS IS"
AND
*  ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
*  IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
*  ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
*  FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
*  DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
*  OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
*  HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
*  LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
*  OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
*  SUCH DAMAGE.
*
********************************************************************************
*******************************
        --><xs:annotation>
     <xs:documentation>Generated using schema builder version 3.2.5. Stylesheets:
```

```
HMD was rendered into XML using software provided to HL7 by Beeler Consulting LLC.
 HMD to MIF Transform: $Id: RoseTreeHmdToMIFStaticModel.xsl 49 2009-03-25 15:09:56Z
Woody $
  Base transform: $Id: ConvertBase.xsl 8 2009-01-26 19:10:36Z Woody $
  Package Id Conversion: $Id: TransformPackageIds.xsl 8 2009-01-26 19:10:36Z Woody
$
  HTML To MIF markup: $Id: HtmlToMIFMarkup.xsl 8 2009-01-26 19:10:36Z Woody $
 Flat to Serialization Transform: $Id: MIFStaticModelFlatToSerialization.xsl 8
2009-01-26 19:10:36Z Woody $
 Fix Names Transform: $Id: FixMifNames.xsl 47 2009-03-23 00:05:58Z Woody $
  Base transform: $Id: ConvertBase.xsl 8 2009-01-26 19:10:36Z Woody $
  Package Id Conversion: $Id: TransformPackageIds.xsl 8 2009-01-26 19:10:36Z Woody
$
StaticMifToXsd.xsl version 2.0</xs:documentation>
   </xs:annotation>
   <xs:include schemaLocation="../coreschemas/infrastructureRoot.xsd"/>
   <xs:include schemaLocation="COCT_MT090300UV01.xsd"/>
   <xs:include schemaLocation="COCT_MT090100UV01.xsd"/>
   <xs:include schemaLocation="MCAI_MT900001UV01.xsd"/>
   <xs:complexType name="QUQI_MT020001UV01.AuthorOrPerformer">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="noteText" type="ED" minOccurs="0" maxOccurs="1"/>
         <xs:element name="time" type="IVL_TS" minOccurs="0" maxOccurs="1"/>
         <xs:element name="modeCode" type="CE" minOccurs="0" maxOccurs="1"/>
         <xs:element name="signatureCode" type="CE" minOccurs="0" maxOccurs="1"/>
         <xs:element name="signatureText" type="ED" minOccurs="0" maxOccurs="1"/>
         <xs:choice>
            <xs:element name="assignedDevice"
type="COCT_MT090300UV01.AssignedDevice" nillable="true"
                        minOccurs="1"
                        maxOccurs="1"/>
            <xs:element name="assignedPerson"
type="COCT_MT090100UV01.AssignedPerson" nillable="true"
                        minOccurs="1"
                        maxOccurs="1"/>
         </xs:choice>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
      <xs:attribute name="typeCode" type="x_ParticipationAuthorPerformer"
use="required"/>
      <xs:attribute name="contextControlCode" type="ContextControl" use="optional"
default="AP"/>
   </xs:complexType>

<!-- added by mustafa -->
   <xs:complexType name="QUQI_MT020001UV01.ControlActProcess">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="id" type="II" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element name="code" type="CD" minOccurs="0" maxOccurs="1"/>
         <xs:element name="text" type="ED" minOccurs="0" maxOccurs="1"/>
         <xs:element name="effectiveTime" type="IVL_TS" minOccurs="0"
maxOccurs="1"/>
         <xs:element name="priorityCode" type="CE" minOccurs="0"
maxOccurs="unbounded"/>
         <xs:element name="reasonCode" type="CE" minOccurs="0"
maxOccurs="unbounded"/>
         <xs:element name="languageCode" type="CE" minOccurs="0" maxOccurs="1"/>
         <xs:element name="overseer" type="QUQI_MT020001UV01.Overseer"
nillable="true" minOccurs="0"
                     maxOccurs="unbounded"/>
         <xs:element name="authorOrPerformer"
type="QUQI_MT020001UV01.AuthorOrPerformer"
                     nillable="true"
                     minOccurs="0"
                     maxOccurs="unbounded"/>
```

```
         <xs:element name="dataEnterer" type="QUQI_MT020001UV01.DataEnterer"
nillable="true"
                     minOccurs="0"
                     maxOccurs="unbounded"/>
         <xs:element name="informationRecipient"
type="QUQI_MT020001UV01.InformationRecipient"
                     nillable="true"
                     minOccurs="0"
                     maxOccurs="unbounded"/>
         <xs:element name="reasonOf" type="QUQI_MT020001UV01.Reason"
nillable="true" minOccurs="0"
                     maxOccurs="unbounded"/>
         <xs:element name="queryByParameter"
                     type="QUQI_MT020001UV01.QueryByParameter"
                     nillable="true"
                     minOccurs="0"
                     maxOccurs="1"/>
         <xs:element name="queryContinuation"
            type="QUQI_MT020001UV01.QueryContinuation"
            nillable="true"
            minOccurs="0"
            maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="classCode" type="ActClassControlAct" use="required"/>
      <xs:attribute name="moodCode" type="x_ActMoodIntentEvent" use="required"/>
   </xs:complexType>
   <xs:complexType name="QUQI_MT020001UV01.QueryContinuation">
      <xs:sequence>
         <xs:element name="queryId" type="II" minOccurs="0" maxOccurs="1"/>
         <xs:element name="statusCode" type="CS" minOccurs="1" maxOccurs="1"/>
         <xs:element name="startResultNumber" type="INT" minOccurs="0"
maxOccurs="1"/>
         <xs:element name="continuationQuantity" type="INT" minOccurs="0"
maxOccurs="1"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="QUQI_MT020001UV01.QueryByParameter">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="queryId" type="II" minOccurs="0" maxOccurs="1"/>
         <xs:element name="statusCode" type="CS" minOccurs="1" maxOccurs="1"/>
         <xs:element name="modifyCode" type="CS" minOccurs="0" maxOccurs="1"/>
         <xs:element name="responseElementGroupId" type="II" minOccurs="0"
maxOccurs="unbounded"/>
         <xs:element name="responseModalityCode" type="CS" minOccurs="0"
maxOccurs="1"/>
         <xs:element name="responsePriorityCode" type="CS" minOccurs="0"
maxOccurs="1"/>
         <xs:element name="initialQuantity" type="INT" minOccurs="0"
maxOccurs="1"/>
         <xs:element name="initialQuantityCode" type="CE" minOccurs="0"
maxOccurs="1"/>
         <xs:element name="executionAndDeliveryTime" type="TS" minOccurs="0"
maxOccurs="1"/>
         <xs:element name="parameterList" type="QUPC_MT040300UV01.ParameterList"
minOccurs="1"
                     maxOccurs="unbounded"/>
         <xs:element name="sortControl" type="QUQI_MT020001UV01.SortControl"
nillable="true"
                     minOccurs="0"
                     maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
   </xs:complexType>
<!-- /end -->
```

```
   <xs:complexType name="QUQI_MT020001UV01.DataEnterer">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="time" type="IVL_TS" minOccurs="0" maxOccurs="1"/>
         <xs:element name="assignedPerson" type="COCT_MT090100UV01.AssignedPerson"
nillable="true"
                     minOccurs="1"
                     maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
      <xs:attribute name="typeCode" type="ParticipationDataEntryPerson"
use="required"/>
      <xs:attribute name="contextControlCode" type="ContextControl" use="optional"
default="AP"/>
   </xs:complexType>
   <xs:complexType name="QUQI_MT020001UV01.InformationRecipient">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="time" type="IVL_TS" minOccurs="0" maxOccurs="1"/>
         <xs:element name="assignedPerson" type="COCT_MT090100UV01.AssignedPerson"
nillable="true"
                     minOccurs="1"
                     maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
      <xs:attribute name="typeCode" type="ParticipationInformationRecipient"
use="required"/>
      <xs:attribute name="contextControlCode" type="ContextControl" use="optional"
default="AP"/>
   </xs:complexType>
   <xs:complexType name="QUQI_MT020001UV01.Overseer">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="noteText" type="ED" minOccurs="0" maxOccurs="1"/>
         <xs:element name="time" type="IVL_TS" minOccurs="0" maxOccurs="1"/>
         <xs:element name="modeCode" type="CE" minOccurs="0" maxOccurs="1"/>
         <xs:element name="signatureCode" type="CE" minOccurs="0" maxOccurs="1"/>
         <xs:element name="signatureText" type="ED" minOccurs="0" maxOccurs="1"/>
         <xs:element name="assignedPerson" type="COCT_MT090100UV01.AssignedPerson"
nillable="true"
                     minOccurs="1"
                     maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
      <xs:attribute name="typeCode" type="x_ParticipationVrfRespSprfWit"
use="required"/>
      <xs:attribute name="contextControlCode" type="ContextControl" use="optional"
default="AP"/>
   </xs:complexType>
   <xs:complexType name="QUQI_MT020001UV01.Reason">
      <xs:sequence>
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="detectedIssueEvent"
type="MCAI_MT900001UV01.DetectedIssueEvent"
                     nillable="true"
                     minOccurs="1"
                     maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
      <xs:attribute name="typeCode" type="ActRelationshipReason" use="required"/>
      <xs:attribute name="contextConductionInd" type="bl" use="optional"/>
   </xs:complexType>
   <xs:complexType name="QUQI_MT020001UV01.SortControl">
      <xs:sequence>
```

```
         <xs:group ref="InfrastructureRootElements"/>
         <xs:element name="sequenceNumber" type="INT" minOccurs="0" maxOccurs="1"/>
         <xs:element name="elementName" type="SC" minOccurs="0" maxOccurs="1"/>
         <xs:element name="directionCode" type="CS" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
      <xs:attributeGroup ref="InfrastructureRootAttributes"/>
      <xs:attribute name="nullFlavor" type="NullFlavor" use="optional"/>
   </xs:complexType>
</xs:schema>
```