

HL7 v3 Implementation Guide

PatientRegistry, PersonRegistry, CareManager, EncounterManager and DocumentManager

Publikasjonens tittel: HL7 v3 Implementation Guide
PatientRegistry, PersonRegistry, CareManager,
EncounterManager and DocumentManager

Teknisk standard nr.: HIS 1038:2011

Utgitt: 08/2011

Bestillingsnummer:

Utgitt av: Helsedirektoratet
Kontakt: Seksjon standardisering
Postadresse: Pb. 7000 St Olavs plass, 0130 Oslo
Besøksadresse: Universitetsgata 2, Oslo

Tlf.: 810 20 050
Faks: 24 16 30 01
www.helsedirektoratet.no

TITTEL**HL7 v3 Implementation Guide**

PatientRegistry, PersonRegistry, CareManager, EncounterManager and DocumentManager

Forfatter

Anita Lorck og Annebeth Askevold

Oppdragsgiver

NIKT

Dokumenttype og nummer	Dato	Antall sider	Gradering
HIS 1038:2011	24.08.2011	170	Åpen
KITH 1038 : 2011			

Godkjent av

Tom Christensen

**Sammendrag**

This national implementation guide is based on:

- the implementation guide developed by Helse Vest IKT:
http://hl7.ihelse.net/Dokument/HL7v3_ImplementationGuide_3.0c.doc
- the further work done by the HL7 Norway Patient Administrative work group:
<http://www.hl7.no/hl7wiki>

Regular procedures for this type of document have been followed. Interested parties have been commenting on a pre-version, and the comments have led to changes in the document.

All services in this document are based on the Normative Edition 2008 of HL7 version 3.

The intent of this document is to describe the key aspects of the services and the HL7 v3 documents related to them.

Content

1.	Introduction.....	7
1.1.	Identity Management in Norwegian Healthcare	7
1.1.1.	Introduction	7
1.1.2.	Registry Functionality	8
1.1.3.	Use of the Registries.....	9
1.2.	Identification of Patients and Persons	9
1.2.1.	Identity Domain.....	9
1.2.2.	Types of Identifier	10
1.2.3.	Method of linking identifiers.....	10
1.2.4.	HL7 version 3 model	11
1.2.5.	Guidance.....	12
2.	Definitions and relationships	14
2.1.	Assumptions	15
2.2.	Architecture of Identity and Demographics data management	17
3.	Services	18
3.1.	PatientRegistry	18
3.1.1.	PatientRegistry.AddPatient	18
3.1.2.	PatientRegistry.GetDemographics	28
3.1.3.	PatientRegistry.FindCandidates	37
3.1.4.	PatientRegistry.RecordRevised	49
3.2.	PersonRegistry.....	54
3.2.1.	PersonRegistry.AddPerson	54
3.2.2.	PersonRegistry.GetDemographics	61
3.2.3.	PersonRegistry.FindCandidates	68
3.2.4.	PersonRegistry.RecordRevised	81
3.3.	CareRecordManager.....	86
3.3.1.	Method: CareRecordManager.GetCareRecordProfile	86
3.4.	EncounterManager	93
3.4.1.	EncounterManager.FindEncounters	93
3.4.2.	EncounterManager.FindScheduledEncounters	100
3.5.	DocumentManager	107
3.5.1.	Method: DocumentManager.ProcessNewDocument	107
3.5.2.	Method: DocumentManager.ProcessDocumentReplacement.....	112
4.	Model Elements	115
4.1.	Object Identifiers (OIDs).....	115
4.1.1.	Coding Systems.....	116
4.1.2.	Identification Schemes	118

5.	Data types and coding systems	120
5.1.	Data Types	120
5.1.1.	PN - Person Name.....	120
5.1.2.	AD - Address	122
5.1.3.	Telephone and mail.....	124
5.1.4.	II – Instance Identifier.....	124
5.1.5.	NullFlavor usage in Data Types	125
5.1.6.	Administrative Entity associated with a Patient or Person	130
5.2.	Coding systems.....	132
5.2.1.	Maritalstatus (Sivilstand).....	132
5.2.2.	Administrative Gender.....	133
5.2.3.	Healthcare organization types.....	133
5.2.4.	Healthcare provider types	133
5.2.5.	Job code	133
5.2.6.	Encounter Types	133
5.2.7.	Administrative Observation types	134
6.	Wrappers	135
6.1.	Transmission Wrapper.....	135
6.1.1.	Information Model.....	135
6.1.2.	Examples.....	138
6.2.	ControlAct Wrapper	141
6.2.1.	Information Model.....	141
6.2.2.	Examples.....	145
6.3.	Registration Act	146
6.3.1.	Information Model.....	147
7.	Reusable Models	151
7.1.1.	Generic Person Model	151
7.1.2.	Generic Patient payload model	154
8.	Acknowledgements and Error Handling.....	157
8.1.	Accept Acknowledgement.....	157
8.1.1.	Introduction.....	157
8.1.2.	Examples.....	158
8.2.	Error Handling	159
8.2.1.	Overview.....	159
8.3.	WSDL and Schemas	165
9.	Mapping to other models	167
9.1.	Folkeregisteret	168
10.	References	170

1. Introduction

This national implementation guide is based on:

- the implementation guide developed by Helse Vest IKT:
http://hl7.ihelse.net/Dokument/HL7v3_ImplementationGuide_3.0c.doc
- the further work done by the HL7 Norway Patient Administrative work group:
<http://www.hl7.no/hl7wiki>

Regular procedures for this type of document have been followed. Interested parties have been commenting on a pre-version, and the comments have led to changes in the document.

All services in this document are based on the Normative Edition 2008 of HL7 version 3.

The intent of this document is to describe the key aspects of the services and the HL7 v3 documents related to them.

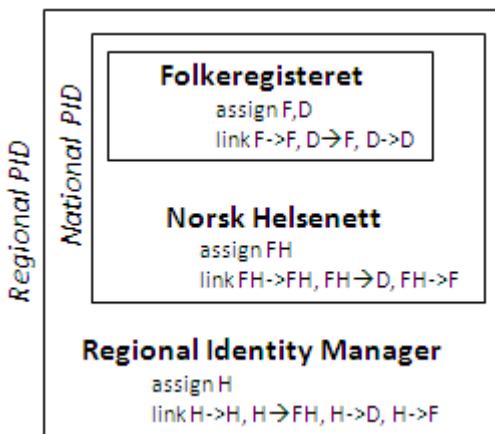
For additional information about HL7 and the HL7 version 3 standards, see
<http://en.wikipedia.org/wiki/HL7> and http://hl7book.net/index.php?title=HL7_version_3

1.1. Identity Management in Norwegian Healthcare

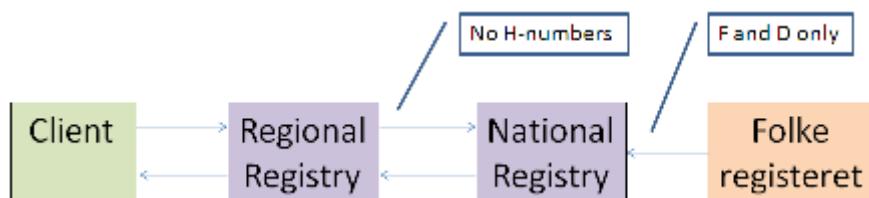
1.1.1. Introduction

The identification of persons (and patients played by persons) in Norwegian healthcare is based on a hierarchy of registries that assign/manage identifiers. The hierarchy consists of three levels:

1. **National person registry.** Folkeregisteret assigns F- and D-numbers in order to identify (temporary) Norwegian citizens. F- and D-numbers are assigned irrespective of any patient roles that these persons may play.
2. **National emergency number registry.** Norsk Helsenett assigns FH-numbers in order to identify persons (who are patients in this context) for whom an F- or D-number isn't known, or to identify persons that don't have an F- or D-number (e.g. foreigners, newborn babies).
3. **Regional patient registry.** The health regions may assign H-numbers to identify persons (who are patients in this context) if and when no FH-number can be requested from the National emergency number registry. Examples: the National emergency number registry is unavailable or can't be reached, or emergencies where large numbers of identifiers need to be assigned within a short timeframe.

**Figur 1 Hierarchy of registries**

The figure above depicts the relationship between the various registries. The outermost (regional) registry builds on top of the functionality offered by the (inner) national registries.

**Figur 2 F-, D-, FH- and H-number communication**

Using a software application schematic: the National emergency number registry is informed by Folkeregisteret about F- and D-numbers. A Client communicates with a Regional Registry, which manages H-numbers itself, and has the capacity to communicate with the National Registry about FH-, D- and F-numbers.

The concept of *Person/Patient Identity Domain (PID)* is introduced on [this page](#). In the context of this introduction there are two PIDs of importance:

1. National PID: the Identity Domain that encompasses the F-, D- and FH-numbers.
2. Regional PID: the Identity Domain that encompasses the identifiers used in the National PID as well as the H-number.

A person may have been assigned multiple identifiers (e.g. an F- and a temporary FH-number). The registries have the capability to associate/link the various identifiers that are all identifying one and the same real world person.

1.1.2. Registry Functionality

The National Registry as well as the Regional Registry are able to fulfill the following use cases:

1. Receive requests to create/assign a new H- or FH-number

2. Receive requests to update a set of demographics data associated with an existing H- or FH-number
3. Return full demographics data when queried based on a F-/D-/FH- or H-number
4. Return a list of potential candidates when queried based on partial demographics data (e.g. name, birth date).
5. Link (and unlink) identifiers in case one retrospectively discovers that two identifiers are in fact (not) identifying one and the same real world person. This is limited to the (un)linking of either an H- or FH-identifier with a F-/D-/FH- or H-number.

1.1.3. Use of the Registries

The distributed/hierarchical nature of the registries does pose a requirement on all client applications that depend on the functionality of these registries:

- All applications SHALL query the Regional/National Registry for the latest set of demographics information as well as the full set of associated/linked identifiers when starting to (again) actively use a particular patient record.

1.2. Identification of Patients and Persons

1.2.1. Identity Domain

An Identity Domain is defined as a set of software applications that share the same person/patient identifier, and use them under a common policy. Each and every Identity Domain contains (at least) one Identity Source (an application that assigns new Identities).

A Person Identity Domain, or a Patient Identity Domain will be referred to as a PID in this document.

Examples:

- All healthcare applications in Norway are part of the Norwegian PID - they all have the ability to use F-, D- and FH-numbers, assigned by a central registry application. The NNPR (Folkeregisteret) acts as the Identity Source for F- and D-numbers.
- The PAS system used in Bergen shares the Patient Identities it assigns with all other software applications used in Helse Vest. All software applications used in Helse Vest are part of one and the same Regional PID.
 - Note that the PAS system and all other software applications used in Helse Vest are part of both the Regional PID as well as the Norwegian PID.
- The Laboratory application in Stavanger assigns its own patient identifiers to all patients. This identifier is internal to the laboratory application only, it is never shared with any other application. This application forms its own PID.
 - Note that the Laboratory application is part of its own PID, the Regional PID as well as the Norwegian PID.
- The Folkeregister, which is itself part of the Norwegian PID, has no knowledge of any identifiers assigned by the PAS in Bergen, because the Folkeregister isn't part of the Regional PID.

A software application in a Regional PID SHALL not send any of its locally assigned identifiers to any applications outside of the Regional PID (e.g. software applications in other regions, the Folkeregister).

1.2.2. Types of Identifier

Most of the patients treated in Norwegian health care are registered in the Norwegian Person registry (Folkeregisteret; which manages the national Norwegian PID) and have a unique eleven digit number. These person identifiers are also used to identify the patient. These identifiers are meaningful to the user of a software application and hence are shown to the user. Three types of identifier exist:

1. The F-number: this identifier is also used to identify the patient. The F-number is assigned to persons living in Norway on a permanent basis. The (preferred) OID for the F-number identification system is 2.16.578.1.12.4.1.4.1.
2. The D-number: Persons that are not living in Norway on a permanent basis can in some circumstance be registered in NNPR with a unique eleven digit number (a D-number). The (preferred) OID for the D-number identification system is 2.16.578.1.12.4.1.4.2.
3. The FH-number: Persons can be assigned a temporary identifier, e.g. persons not living in Norway, newborn babies, or emergency patients. An FH-number may be linked with an F-number or D-number at some later point in time. The (preferred) OID for the FH-number identification system is 2.16.578.1.12.4.1.4.3.

A Patient may have one or more other patient identifier assigned within a regional healthcare organization (regional PID). These identifiers shall not be communicated outside of the organization (i.e. outside of the Identity Domain in which they were created).

Identifiers assigned and used within a Regional PID may include:

- Temporary identifier: An application within the organization may assign an emergency/temporary identifier (H-number) to a patient.
- Application internal identifier: the PAS application within the regional healthcare organization will assign a unique internal identifier.

These identification schemes each have their own unique OID associated with them.

Most of these identifiers are linked to either an F-number, a D-number, or an FH-number. The linking may occur at some later point in time.

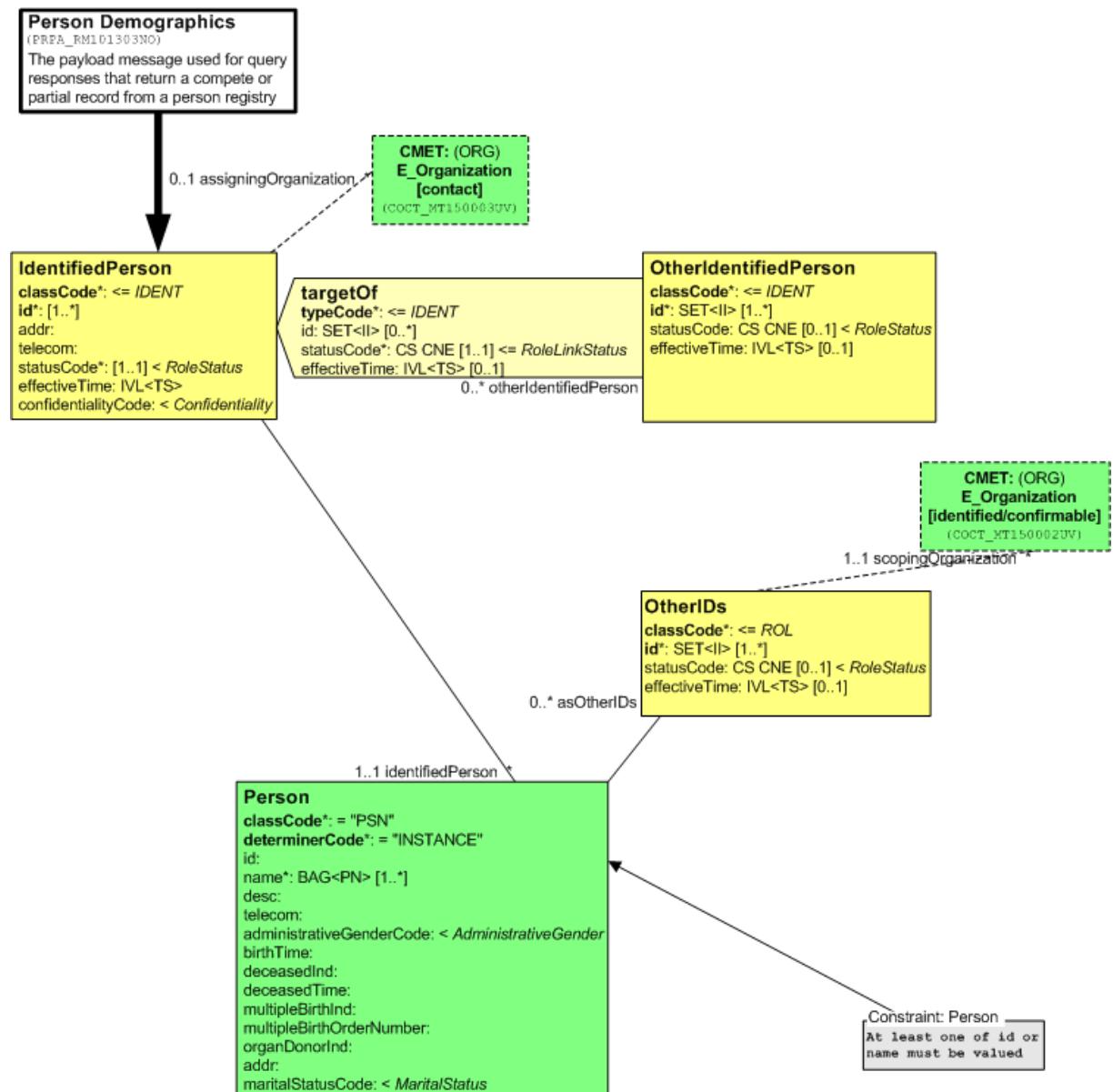
1.2.3. Method of linking identifiers

The various Person/Patient registries described here use a linking mechanism with the following key characteristics:

- Links are *unidirectional* in nature, from a less-preferred (less reliable) to a more-preferred (more reliable) identifier. For example: a link from an FH-number to an F-number.
 - Any queries for information related to identifiers that are linked to a more-reliable identifier are processed as if the query was sent using that more-reliable identifier.
 - For example: if one queries for the demographics data of a person with FH-number A, and A is known to be linked to the F-number B, then the service will return the demographics details associated with B.

- The link structure will be maintained as a flat structure with a maximum depth of 1. Any structures with a depth greater than 1 will be collapsed (based on the transitive nature of the links) into a structure with depth 1.
 - For example: if A is linked to B, and B to C, then the resulting link structure will consist of A-to-C and B-to-C.
 - For example: if we have one structure "A-to-C and B-to-C", and another structure "E-to-G and H-to-G and K-to-G" and we link E (which is G) to A (which is C), then the resulting structure will be "A-to-C and B-to-C and E-to-C and G-to-C and H-to-C and K-to-C". C has 6 child nodes; the overall structure has a depth of 1.

1.2.4. HL7 version 3 model



Figur 3 Generic Role/Person model

The above model shows the generic structure used in HL7 for Patient/Person registries.

Note: the above model is for the Person registry. The same principles apply to Patient registries.

- The focal class of the model is the IdentifiedPerson role with a mandatory ID attribute. This attribute contains the primary identifier(s) of the Person as known and used by the sending software application.
- The role is played by a Person entity, which has an optional ID attribute.
- There are zero or more OtherIdentifiedPerson classes, each of which has an identifiedBy (erroneously shown in the image as "TargetOf") relationship with the IdentifiedPerson class. The existence of an identifiedBy relationship indicates that the identifier of the OtherIdentifiedPerson is also an identifier of the associated IdentifiedPerson. Thus a link between the identifiers can be said to be in place.
- The OtherIDs class is intended to convey current identifiers for the person in other contexts (other registries). The person's identifiers in other health care organizations are modeled there. The organization that issued and uses the identifier is modeled as the scoping organization of OtherIDs. Examples: passport number, driver's license number, H-number as assigned by another PID region. This class is not used in Norway.

1.2.5. Guidance

When it comes to identifiers the following rules apply:

1. The Patient/IdentifiedPerson Role class shall have 1 identifier:
 - Either the F-number, the D-number, the FH-number, or the H-number (in that order of preference; if multiple of these identifiers are known, only the most-preferred identifier shall be sent here).
 - The H-number (a locally assigned temporary identifier) may only be sent if the service is used within a Regional PID.
2. The Person class shall have at most 1 identifier: Either the F-number, the D-number or the FH-number (in that order of preference; if multiple of these identifiers are known, only the most-preferred identifier shall be sent).
3. There shall be one instance of the OtherIdentifiedPerson/OtherPatient class (and identifiedBy association) for each other known identifiers:
 - The D-number if the F-number is also known; the FH-number(s) if the D-number or F-number is also known; non-preferential F-number if the F-number of the person changed/merged; non-preferential FH-numbers if the person has multiple FH-numbers.
 - (Only if the service is used within a Regional PID): old locally assigned temporary identifiers (H-numbers).

The following XML-snippet is an example of how these identifiers are conveyed as part of the HL7 version 3 model:

```
<patient>
  <!-- Response contains one of F/D/FH number -->
  (see RULE#1)
  <id root="2.16.578.1.12.4.1.4.1" extension="15076500565" assigningAuthorityName="F-nummer"/>
```

```
... patient demographics ..  
<patientPerson>  
    <!-- ONE Person identifier from the Folkeregister: here: F number -->  
    (see RULE #2)  
    <id root="2.16.578.1.12.4.1.4.1" extension="15076500565"/>  
    ... person demographics ..  
</patientPerson>  
... (RULE #3) zero or more other person identities that identify the same patient ...  
<identifiedBy typeCode="IDENT">  
    <statusCode code="active"/>  
    <otherIdentifiedPerson classCode="IDENT">  
        <id root="2.16.578.1.12.4.1.4.2" extension="70019950032"  
            assigningAuthorityName="D-nummer"/>  
    </otherIdentifiedPerson>  
</identifiedBy>  
</patient>
```

2. Definitions and relationships

Definitions used in this document:

OID	Object Identifier
NNPR	Norwegian National Population Register (<i>Folkeregister</i>)
F-number	Norwegian 11-digit personal identification number given to all who resides in Norway
D-number	11-digit identity number given to people intending to stay in Norway for a short period only. D-numbers are used for identification purposes in the same way as personal identification numbers.
FH-number	National temporary/emergency number given by a national service (planned, not in work yet)
H-number	Local temporary/emergency number as defined by KITH ¹ : http://www.kith.no/templates/kith_WebPage_609.aspx

¹ KITH was transferred to Helsedirektoratet effective 1.1.2012

2.1. Assumptions

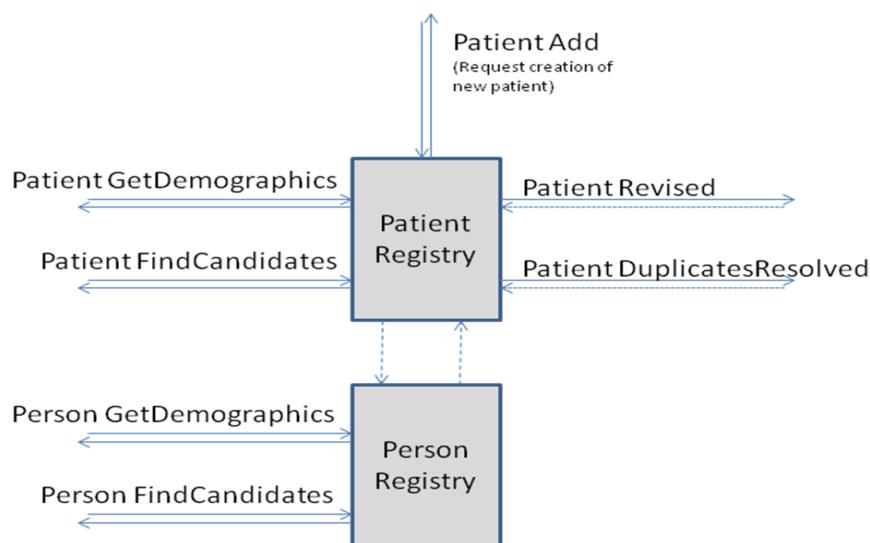
- A1: For privacy reasons the author (a person) of a query that is related to the demographics data of a patient/person should be sent. Medical data is subject to additional privacy/auditing regulations.
 - Audit logging should be compliant to:
http://www.shdir.no/samspill/informasjonssikkerhet/norm_for_informasjonssikkerhet_i_helsesektoren_232354.
 - Based on the requirements above, a decision has been made to include (in those messages not automatically generated by a software application) the “user identifier” of the person sending a message in all messages.
- A2: In all communication the patient will be identified using the F-number (if available); the D-number (if the F-number isn't available), the FH-number (when available and if the F-number or the D-number isn't available), or local H-number (if F-, D- or FH-number are not available).
 - H-numbers as defined by KITH should be used.
- A3: The standard used will be Normative Edition 2008 of the HL7 version 3 standard. The IHE HL7 v3 profiles aren't used – the Norwegian requirements go beyond the restricted model used in that profile.
- A4: The F-number as well as the D-number is used to both identify a person as well as a person-in-the-context-of-healthcare (a patient in the context of the services described in this document). The F- and D-numbers are not used to identify healthcare persons.
- A5: There will be one patient master (PAS) and there is one person master (Folkeregister).
 - There will be a maximum of one person.id (either the F-number or the D-number –in that order of preference-), and a maximum of one Patient.id (either the F-number, the D-number, the FH-number or the H-number –in that order of preference-). All other known identifiers for the Patient (older, previously used, temporary identifiers, non-preferential identifiers) should be sent in the OtherIds.id attribute.
- A6: There is a tightly controlled process when it comes to the merging of patient identifiers. The un-merging of previously merged patient identifiers rarely happens.
 - There will be a notification interaction from the Patient Registry to other applications. This to inform these applications that it should merge all data associated with a set of patient identifiers.
 - There won't be a notification from the Patient Registry in case identifiers are un-merged: this remains a manual process.

- An interaction from an application to the Patient Registry to request that identifiers be merged (as well as un-merged) won't be supported. Merges are exclusively made in the Patient Registry itself.

2.2. Architecture of Identity and Demographics data management

The architectural model related to the management of person/patient identifiers and demographics data is based on one core principle: there will be one patient master (Patient registry, PAS) and there is one person master (Person Registry, Folkeregister).

- A new patient can only be created within the Patient Registry. This could either be a manual process using the Patient Registry application, or another software application could use a service to request that the Patient Registry create a new patient.
- All updates related to patient demographics (including the merging of patient identifiers) are made available by the Patient Registry to other software applications in the form of notifications.
- To get hold of the latest demographic information (and the best quality in terms of identifiers) software applications can query the Patient/Person Registry using an identifier of the Patient/Person.
- In order to determine whether a patient/person is already present in the registry software applications have the option to search for potential matches using partial demographics person/patient information.



Figur 4 Architecture of identity and demographics data management

3. Services

This chapter contains the description of a number of services. The semantic interchange model used by the services is based on the international HL7 version 3 standards.

3.1. PatientRegistry

3.1.1. *PatientRegistry.AddPatient*

This is a new use-case not (yet) covered by the international HL7 version 3 standard.

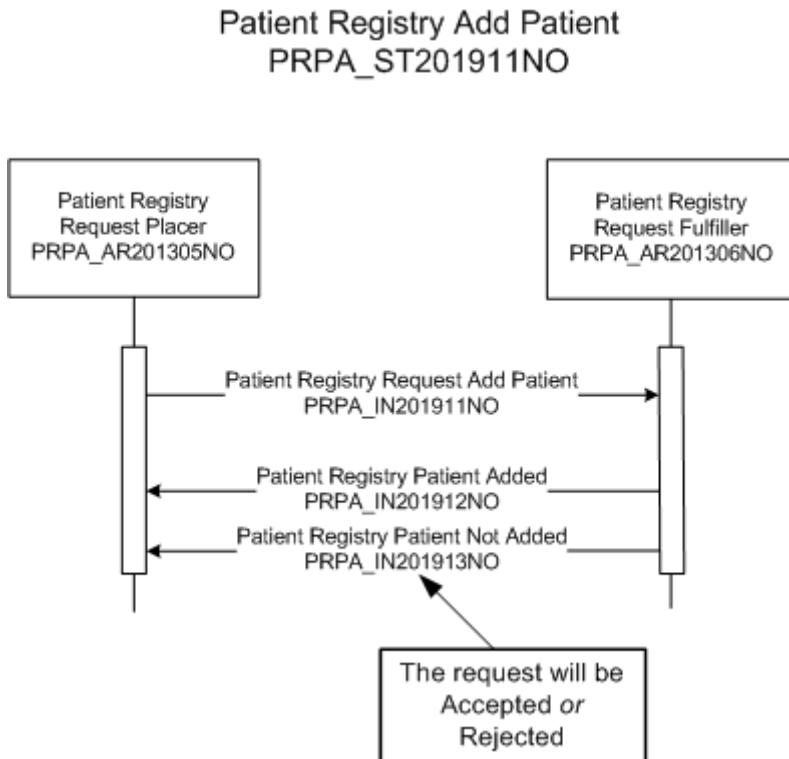
3.1.1.1. Operation-level Profile

Operation Name	PatientRegistry.AddPatient
Purpose Description	The aim of the PatientRegistry.AddPatient service is to request the receiving application to assign a unique identifier for a patient. This service is mostly used to request an H-number or FH-number (when available).
Logic Description	The service creates unconditionally a new identity with the attributes from the request.
Input/Output	Input: Patient Registry Add Patient Request (IN201911NO) Output (Added): Patient Registry Add Patient Query Response(PRPA_IN201912NO) Output (Not Added): Patient Registry Add Patient Query Response (PRPA_IN201913NO) See below for details of the information models as well as examples.
Composition Role	Business level service
Composition Member Capabilities	None
Keywords	Add patient, patient, Patientidentifier, H-number, FH-number
Version	4.0
Finalization Status within HL7 Norway	For review
Custodian	HL7 Norway

3.1.1.2. Purpose of the operation

The aim of the PatientRegistry.AddPatient service is to request the adding of a new patient into the patient registry. The response should contain a patient identifier as well as the request parameters.

Storyboard Diagram



Textual Storyboard

1. There is a new born baby. This person is not known in the Folkeregister. The clinical system which deals with the information about the birth sends a request to the Patient Registry for an H-number (emergency number). The request contains: Birth date, sex, name, address, kommunenr and maybe bydel. The Patient Registry responds with an ID (H-number or FH-number) and the set of demographics detail as provided in the request.
2. The clinical system for emergency patients sends a request to the Patient Registry to register the new patient. The clinical system has verified with the Folkeregister: the person is known in that registry. The request to the Patient Registry contains: F- or D-number, Birth date, sex, marital status, names, address, kommunenr and maybe bydel that has been found in the Folkeregister. The Patient Registry responds with a confirmation that the patient was successfully registered, as well as with a full set of demographics details. The request to add the patient will fail if the patient already exists (the PatientRegistry.getDemographics method should have been used instead).

3.1.1.3. Input/Output

The query interaction has an immediate response. The response interaction as set by the Patient Registry will contain all known identifiers as well as demographics details of the patient.

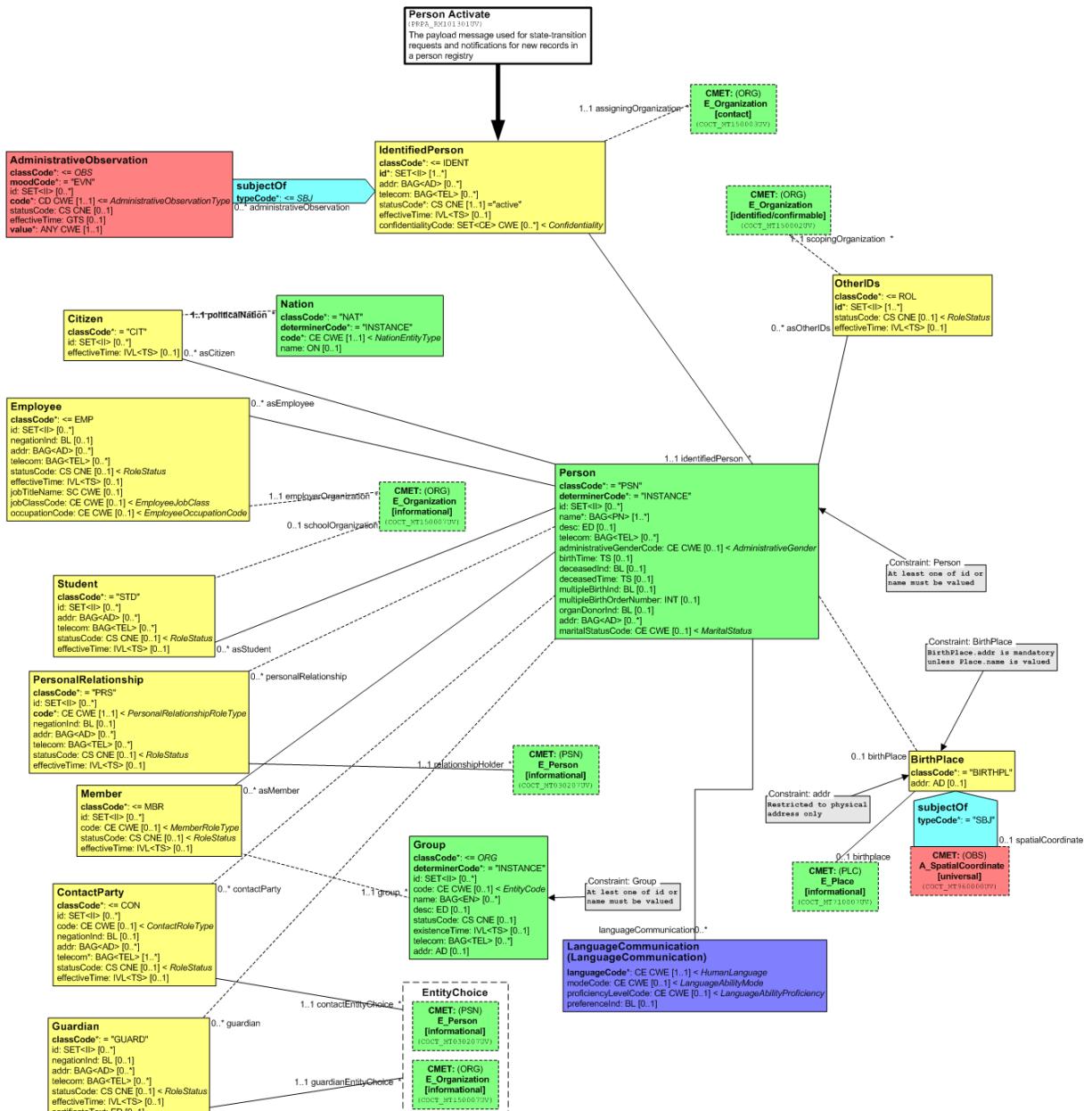
Interaction List

Patient Registry Request Add Patient	PRPA_IN201911NO
Patient Registry Request Added	PRPA_IN201912NO
Patient Registry Patient Not Added	PRPA_IN201913NO

Note that the above interactions use the “NO” realm code. The models used are (currently) specific for this project and will be brought forward for inclusion in the international standard.

Information Model

This model is for illustration. The model is for PatientRegistry.Activate



Figur 5 Source: Normative Edition 2008

3.1.1.4. XML Examples

Input Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!- Example Request to assign new Patient ID / request to create new patient record -->
<PRPA_IN201911NO ITSVersion="XML_1.0" xmlns="urn:hl7-org:v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v3
  ./schemas/PRPA_IN201911NO.xsd">

  <!- Unique identification of this message instance. Root derived from that of the sending application -->
```

```
<id extension="080623131707123" root="2.16.578.1.34.1.408.7"/>
<!- Time message was sent -->
<creationTime value="20080623131707"/>
<versionCode code="NE2008"/>
<!- Fixed values for Patient.Add message -->
<interactionId extension="PRPA_IN201911NO" root="2.16.840.1.113883.1.6"/>
<processingCode code="P"/>
<processingModeCode code="T"/>
<acceptAckCode code="AL"/>
<receiver typeCode="RCV">
  <device classCode="DEV" determinerCode="INSTANCE">
    <!- Receiving software application. Helse Vest assigned application identifier -->
    <id extension="871" root="2.16.578.1.34.1"/>
  </device>
</receiver>
<sender typeCode="SND">
  <device classCode="DEV" determinerCode="INSTANCE">
    <!- Sending software application. Helse Vest assigned application identifier -->
    <id extension="408" root="2.16.578.1.34.1"/>
  </device>
</sender>
<controlActProcess classCode="CACT" moodCode="EVN">
  <authorOrPerformer typeCode="AUT">
    <!- Identifies the person responsible for sending this message -->
    <assignedPerson classCode="ASSIGNED">
      <!- User Number assigned by Helse Vest IKT to identify the person -->
      <id extension="12345" root="2.16.578.1.34.3.1"/>

    </assignedPerson>
  </authorOrPerformer>
  <subject typeCode="SUBJ">
    <registrationEvent classCode="REG" moodCode="RQO">
      <id nullFlavor="UNK"/>
      <statusCode code="active"/>
      <subject1 typeCode="SBJ">
        <patient classCode="PAT">
          <!- NO Patient ID - that is the thing that will be assigned as a result of this request -->
          <statusCode code="active"/>
          <patientPerson>
            <!- Zero or One person identifier from Folkeregister (if known for this new patient): here: F
number -->
            <id root="2.16.578.1.12.4.1.4.1" extension="15076500565"
              assigningAuthorityName="F-nummer"/>
            <name use="L">
              <given>Roland</given>
```

```
<family>Gundersen</family>
</name>
<!-- Gendercode: 0, 1, 2 or 9 -->
<administrativeGenderCode code="1" codeSystem="2.16.578.1.12.4.1.1.3101"/>
<birthTime value="19650715"/>
<addr use="H">
  <streetAddressLine>Asker vei 34</streetAddressLine>
  <postalCode>1234</postalCode>
  <city>Oslo</city>
  <precinct>Bydel</precinct>
</addr>
<asMember classCode="MBR">
  <group classCode="PUB">
    <id extension="120106" root="2.16.578.1.12.4.1.1.3403"
assigningAuthorityName="Bydelsnummer"/>
      <code code="ADMENT" codeSystem="2.16.578.1.34.5 .1"/>
    </group>
  </asMember>
</patientPerson>
<providerOrganization classCode="ORG" determinerCode="INSTANCE">
  <id root="2.16.578.1.12.4.1.4.101" extension="983658725"/>
  <contactParty classCode="CON" nullFlavor="NA"/>
</providerOrganization>
</patient>
</subject1>
<!-- Author of the registration request, mostly the same as the author of the message -->
<author typeCode="AUT">
  <assignedEntity classCode="ASSIGNED">
    <!-- User Number assigned by Helse Vest IKT to identify the person -->
    <id extension="12345" root="2.16.578.1.34.3.1"/>
  </assignedEntity>
</author>
</registrationEvent>
</subject>
</controlActProcess>
</PRPA_IN201911NO>
```

Output Example

Added

```
<?xml version="1.0" encoding="UTF-8"?>
<PRPA_IN201912NO ITSType="XML_1.0" xmlns="urn:hl7-org:v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v3
  ./schemas/PRPA_IN201912NO.xsd">
```

```
<!-- Unique identification of this message instance. Root derived from that of the sending application -->
<id extension="938236661" root="2.16.578.1.34.1.871.3"/>
<!-- Time message was sent -->
<creationTime value="20080623131710"/>
<versionCode code="NE2008"/>
<!-- Fixed values for Response to PatientAdd request (in case request was successful) -->
<interactionId extension="PRPA_IN201912NO" root="2.16.840.1.113883.1.6"/>
<processingCode code="P"/>
<processingModeCode code="T"/>
<acceptAckCode code="NE"/>
<receiver typeCode="RCV">
  <device classCode="DEV" determinerCode="INSTANCE">
    <!-- Receiving software application. Helse Vest assigned application identifier -->
    <id extension="408" root="2.16.578.1.34.1"/>
  </device>
</receiver>
<sender typeCode="SND">
  <device classCode="DEV" determinerCode="INSTANCE">
    <!-- Sending software application. Helse Vest assigned application identifier -->
    <id extension="871" root="2.16.578.1.34.1"/>
  </device>
</sender>
<acknowledgement>
  <!-- Affirmative acknowledgement -->
  <typeCode code="AA"/>
<targetMessage>
  <!-- .. related to this original message ID -->
  <id extension="080623131707123" root="2.16.578.1.34.1.408.7"/>
</targetMessage>
</acknowledgement>
<controlActProcess classCode="CACT" moodCode="EVN">
  <authorOrPerformer typeCode="AUT">
    <!-- Identifies the software application responsible for sending this message -->
    <assignedDevice classCode="ASSIGNED">
      <!-- Sending software application. Helse Vest assigned application identifier -->
      <id extension="871" root="2.16.578.1.34.1"/>
    </assignedDevice>
  </authorOrPerformer>
  <subject typeCode="SUBJ">
    <!-- The registration activity done as a result of the request -->
    <registrationEvent classCode="REG" moodCode="EVN">
      <!-- The ID of the registration process, a.k.a. the DIPS Internal patient ID. This ID isn't published. -->
      <id nullFlavor="MSK"/>
      <statusCode code="active"/>
      <subject1 typeCode="SBJ">
```

```
<patient classCode="PAT">
    <!-- Response contains one of F/D/H number -->
    <id root="2.16.578.1.12.4.1.4.1" extension="15076500565" assigningAuthorityName="F-
number"/>
    <statusCode code="active"/>
    <patientPerson determinerCode="INSTANCE" classCode="PSN">
        <!-- ONE Person identifier from the Folkeregister: here: F number -->
        <id root="2.16.578.1.12.4.1.4.1" extension="15076500565"/>
        <name use="L">
            <given>Roland</given>
            <family>Gundersen</family>
        </name>
        <!-- Gendercode: 0, 1, 2 or 9 -->
        <administrativeGenderCode code="1" codeSystem="2.16.578.1.12.4.1.1.3101"/>
        <birthTime value="19650715"/>
        <addr use="HP">
            <streetAddressLine>Rod vei 123</streetAddressLine>
            <postalCode>2345</postalCode>
            <city>Rodby</city>
        </addr>
        <!-- Maritalstatus according to Folkeregister -->
        <maritalStatusCode code="1" codeSystem="2.16.578.1.12.4.1.1.3103" displayName="Ugift"/>
    </patientPerson>
    <providerOrganization classCode="ORG" determinerCode="INSTANCE">
        <!-- ID of Helse Vest RHF from the Enhetsregister; organization that is aware of the patient-role
of this person -->
        <id root="2.16.578.1.12.4.1.4.101" extension="983658725"/>
        <contactParty classCode="CON" nullFlavor="NA"/>
    </providerOrganization>
    <asMember classCode="MBR">
        <group classCode="PUB">
            <id extension="120106" root="2.16.578.1.12.4.1.1.3403"
assigningAuthorityName="Bydelsnummer"/>
            <code code="ADMENT" codeSystem="2.16.578.1.34.5 .1"/>
        </group>
        </asMember>
    </patient>
</subject1>
<custodian typeCode="CST">
    <assignedEntity classCode="ASSIGNED">
        <!-- ID of Helse Vest RHF from the Enhetsregister; organization that is aware of the patient-role of
this person -->
        <id root="2.16.578.1.12.4.1.4.101" extension="983658725"/>
    </assignedEntity>
</custodian>
```

```
</registrationEvent>
</subject>
</controlActProcess>
</PRPA_IN201912NO>
```

Not added

```
<?xml version="1.0" encoding="UTF-8"?>
<PRPA_IN201913NO ITSType="XML_1.0" xmlns="urn:hl7-org:v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v3
  ./schemas/PRPA_IN201913NO.xsd">
  <id extension="ANF291" root="2.16.578.1.34.1.871.3"/>
  <creationTime value="20080623131710"/>
  <versionCode code="NE2008"/>
  <interactionId extension="PRPA_IN201913NO" root="2.16.840.1.113883.1.6"/>
  <processingCode code="P"/>
  <processingModeCode code="T"/>
  <acceptAckCode code="AL"/>
  <receiver typeCode="RCV">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="408" root="2.16.578.1.34.1"/>
    </device>
  </receiver>
  <sender typeCode="SND">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="871" root="2.16.578.1.34.1"/>
    </device>
  </sender>
  <acknowledgement>
    <typeCode code="AE"/>
    <targetMessage>
      <id extension="080623131707123" root="2.16.578.1.34.1.408.7"/>
    </targetMessage>
  </acknowledgement>
  <controlActProcess classCode="CACT" moodCode="EVN">
    <authorOrPerformer typeCode="AUT">
      <assignedPerson classCode="ASSIGNED">
        <id extension="12345" root="2.16.578.1.34.3.1"/>
      </assignedPerson>
    </authorOrPerformer>
    <subject typeCode="SUBJ">
      <registrationEvent classCode="REG" moodCode="RQO">
        <id nullFlavor="UNK"/>
        <statusCode code="active"/>
        <subject1 typeCode="SBJ">
```

```
<patient classCode="PAT">
  <statusCode code="active"/>
  <patientPerson>
    <id root="2.16.578.1.12.4.1.4.1" extension="15076500565" assigningAuthorityName="F-
nummer"/>
    <name use="L">
      <given>Roland</given>
      <family>Gundersen</family>
    </name>
    <administrativeGenderCode code="1" codeSystem="2.16.578.1.12.4.1.1.3101"/>
    <birthTime value="19650715"/>
  </patientPerson>
  <providerOrganization classCode="ORG" determinerCode="INSTANCE">
    <id root="2.16.578.1.12.4.1.4.101" extension="983658725"/>
    <contactParty classCode="CON" nullFlavor="NA"/>
  </providerOrganization>
  <asMember classCode="MBR">
    <group classCode="PUB">
      <id extension="120106" root="2.16.578.1.12.4.1.1.3403"
assigningAuthorityName="Bydelsnummer"/>
        <code code="ADMENT" codeSystem="2.16.578.1.34.5 .1"/>
    </group>
  </asMember>
  </patient>
</subject1>
<author typeCode="AUT">
  <assignedEntity classCode="ASSIGNED">
    <id extension="12345" root="2.16.578.1.34.3.1"/>
  </assignedEntity>
</author>
</registrationEvent>
</subject>
<reasonOf typeCode="RSON">
  <detectedIssueEvent moodCode="EVN" classCode="ALRT" >
    <code code="KNOWNPAT" codeSystem="2.16.578.1.34.5.3" displayName="Patient already known.
This service requires a new patient."/>
  </detectedIssueEvent>
</reasonOf>
</controlActProcess>
</PRPA_IN201913NO>
```

3.1.2. *PatientRegistry.GetDemographics*

3.1.2.1. Operation-level Profile

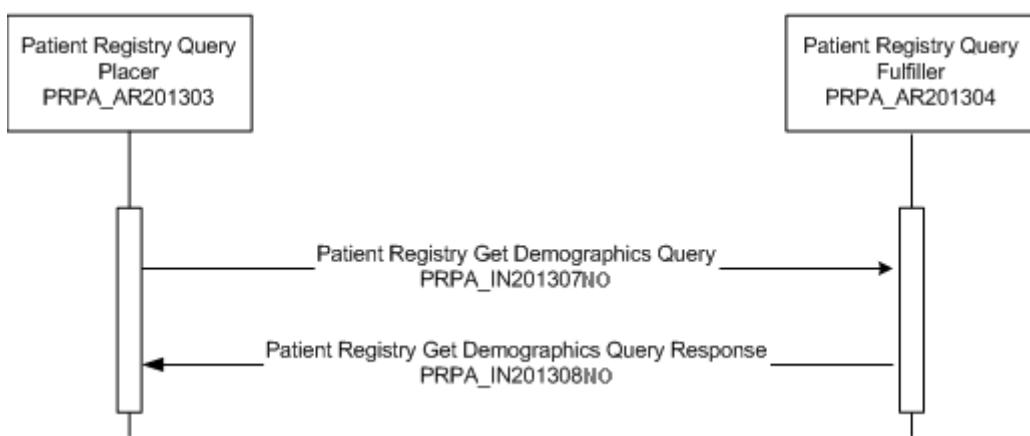
Operation Name	PatientRegistry.GetDemographics
Purpose Description	<p>To query (using a known identifier, e.g. F/D/FH/H-number) the patient registry for the demographic details of a single person.</p> <p>Note that it is the expectation that any application that (re-)activates a patient record should use the GetDemographics service to get hold of the latest data. Part of the data returned may be an indication that (for example) the D-number used has now been merged with an F-number.</p> <p>See below for storyboards that illustrate the purpose of this operation.</p>
Logic Description	The operation will match the supplied identifier, using an exact match, with the identifiers as known for patient it holds the demographics data of, and will return either 0 (in case there wasn't a match) or 1 matching records.
Input/Output	<p>Input: Patient Registry Get Demographics Query (PRPA_IN201307NO). The core of this model is the query parameter that holds the Person Identifier.</p> <p>Output: Patient Registry Get Demographics Query Response (PRPA_IN201308NO). The core of this model is formed by the Generic Person Model.</p> <p>See below for details of the information models as well as examples.</p>
Composition Role	Business level service
Composition Member Capabilities	None
Keywords	Demographics data, patient, search, patient identifier, Patientregister
Version	4.0
Finalization Status within HL7 Norway	For review
Custodian	HL7 Norway

3.1.2.2. Purpose of the operation

The storyboard below demonstrates retrieving the demographic data associated with a specified patient identifier. The query will be either based on the F-number, the D-number, the FH-number or a local H-number.

Storyboard diagram

The storyboard shown below demonstrates querying a patient registry to get demographic information for a registered patient. The query will be either based on the F-number, the D-number, the emergency patient identifier (the FH-number) or a local H-number.



Textual Storyboard

Connor Comrade has a list of patients who need to be invited for follow up to Dr Patricia Primary's clinic. He sets up a list of F-numbers and requests a *Patient Registry Get Demographics Query* for each number. Connor checks the details returned in the *Patient Registry Get Demographics Query Response* to make sure the patients are still alive. Where a date of death and a death status has been returned, Connor makes a note to follow these up later. For all the other patients, Connor uses the details returned in the *Patient Registry Get Demographics Response* messages to set up a mail merge file. Connor then uses this mail merge file to issue an invitation letter to each patient for the clinic.

3.1.2.3. Input/Output

The query interaction has an immediate response. The response interaction as sent by the Patient Registry will contain all known identifiers of the patient as well as the demographics details of the patient.

Interaction List

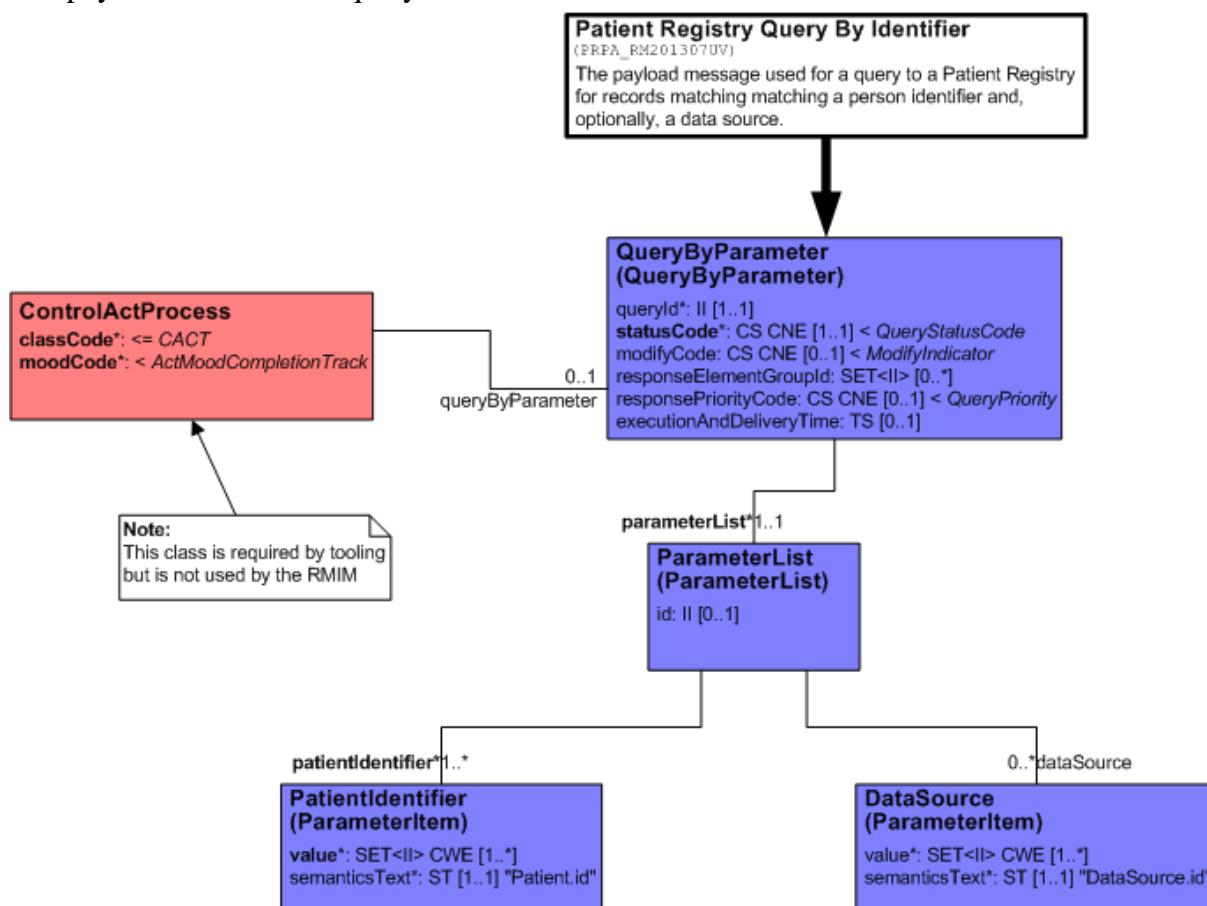
Patient Registry Get Demographics Query	PRPA_IN201307NO
Patient Registry Get Demographics Query Response	PRPA_IN201308NO

Note that the above interactions use the “NO” realm code, and not the original “UV”. The models used are (currently) 100% equivalent to the international models. By using the NO realm the artefact identifier can be used for versioning purposes at some future point in time.

Input Information Model

All interactions are defined in the form of two wrappers (which contain meta-data related to the information exchange; the [Transmission Wrapper](#) and the [ControlAct Wrapper](#)), and a so-called payload model.

The payload model of the query interaction is shown here:



Figur 6 Source: Normative Edition 2008

Class	Component	Documentation
QueryByParameter	queryId	<p>The queryId contains the unique identification of this query instance. See II for a description of the II data type used. The id contains the unique identification of this query instance.</p> <p>@root contains an identification of the ‘unique query instance numbering mechanism as used by this particular sending software application’, and @extension contains the identifier created according to that numbering mechanism. @root contains the ‘namespace’ of the identifier as contained in @extension</p> <p>Implementation note: re-use same @extension/@root as used to identify the message instance in the Transmission Wrapper, see the documentation of the wrappers.</p>
	statusCode	@code contains the fixed value ‘new’
	<i>Other</i>	<p>Any other elements not listed above, but shown in the diagram, are reserved for future use in Norway.</p> <p>Sending applications SHOULD not use these elements; receiving applications SHALL not produce an error if these elements are present, and SHALL ignore these elements if present.</p>

Class	Component	Documentation
PatientIdentifier	value	<p>Contains the patient identifier. This is a unique identification of a patient.</p> <p>See Identification of Patients and Persons for guidance related to the identification of Patients, and the II Data Type for documentation of the II data type.</p> <p>@root contains an identification of the ‘unique patient identification mechanism’ (i.e. the OID for F-number, D-number, etc.), and @extension contains the identifier created according to that identification mechanism.</p>
	semanticsText	semanticsText contains the fixed value ‘Patient.id’

Note: Any other classes not listed above, but shown in the diagram, are reserved for future use in Norway. Sending applications SHOULD not use these model elements; receiving applications SHALL not produce an error if these model elements are present, and SHALL

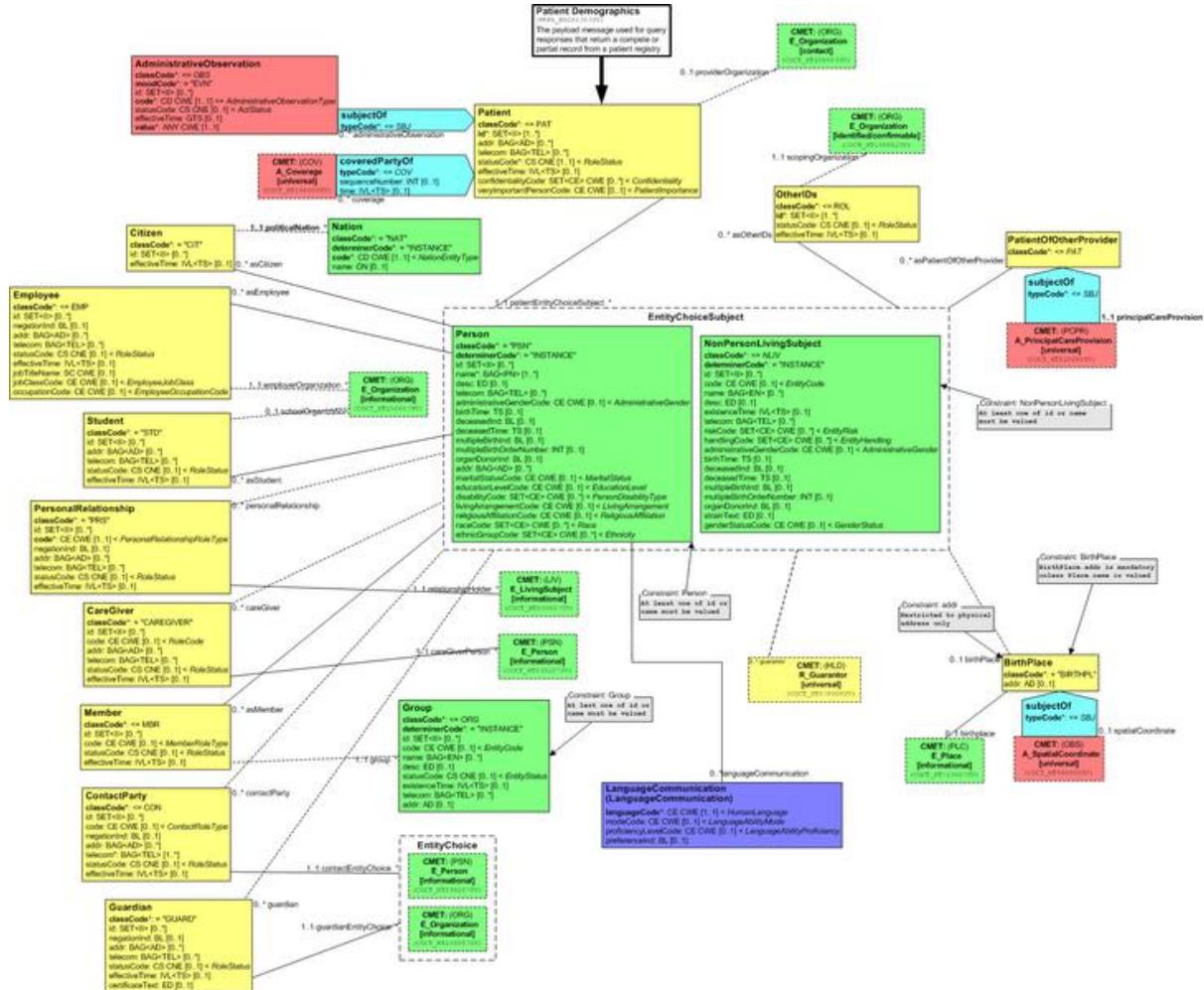
ignore these model elements if present.

```
<queryByParameter>
<queryId extension="3948375" root="2.16.578.1.34.1.145.1"/>
<statusCode code="new"/>
<parameterList>
  <patientIdentifier>
    <value root="2.16.578.1.12.4.1.4.2" extension="64109642356"/>
    <semanticsText>Patient.id</semanticsText>
  </patientIdentifier>
</parameterList>
</queryByParameter>
```

Output Response Model

The payload model of the response interaction is comprised of two parts:

1. a registry related set of metadata, see the [RegistrationEvent](#) section of the Registration Act description for details
2. the role information which is the result set of the query:



Figur 7 Source: Normative Edition 2008

This model is used in multiple services and is documented on the [Generic Patient Model](#) page.

3.1.2.4. XML examples

Input example

```
<?xml version="1.0" encoding="UTF-8"?>
<PRPA_IN201307NO ITSVersion="XML_1.0" xmlns="urn:hl7-org:v3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v3
./schemas/PRPA_IN201307NO.xsd">
    <!-- Unique identification of this message instance. Root derived from that of the sending application -->
    <id extension="080527104501_37" root="2.16.578.1.34.1.222.1"/>
    <!-- Time message was sent -->
    <creationTime value="20080527104501"/>
    <versionCode code="NE2008"/>
    <!-- Fixed values for GetPatientDemographics query -->
    <interactionId extension="PRPA_IN201307NO" root="2.16.840.1.113883.1.6"/>
    <processingCode code="P"/>
    <processingModeCode code="T"/>
```

```

<acceptAckCode code="NE"/>
<receiver typeCode="RCV">
  <device classCode="DEV" determinerCode="INSTANCE">
    <!-- Receiving software application. Helse Vest assigned application identifier -->
    <id extension="922" root="2.16.578.1.34.1"/>
  </device>
</receiver>
<sender typeCode="SND">
  <device classCode="DEV" determinerCode="INSTANCE">
    <!-- Sending software application. Helse Vest assigned application identifier -->
    <id extension="222" root="2.16.578.1.34.1"/>
  </device>
</sender>
<controlActProcess classCode="CACT" moodCode="EVN">
  <authorOrPerformer typeCode="AUT">
    <assignedPerson classCode="ASSIGNED">
      <id extension="12345" root="2.16.578.1.34.3.1"/>
    </assignedPerson>
  </authorOrPerformer>
  <queryByParameter>
    <queryId extension="080527104501_37" root="2.16.578.1.34.1.222.1"/>
    <statusCode code="new"/>
    <parameterList>
      <patientIdentifier>
        <value root="2.16.578.1.12.4.1.4.1" extension="15076500565"/>
        <semanticsText>Patient.id</semanticsText>
      </patientIdentifier>
    </parameterList>
  </queryByParameter>
</controlActProcess>
</PRPA_IN201307NO>

```

Output example

```

<?xml version="1.0" encoding="UTF-8"?>
<PRPA_IN201308NO ITSVersion="XML_1.0" xmlns="urn:hl7-org:v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v3
  ..schemas/PRPA_IN201308NO.xsd">
  <!-- Unique identification of this message instance. Root derived from that of the sending application -->
  <id extension="938211803" root="2.16.578.1.34.1.922.3"/>
  <!-- Time message was sent -->
  <creationTime value="20080527104501"/>
  <versionCode code="NE2008"/>
  <!-- Fixed values for Response to GetPatientDemographics query -->
  <interactionId extension="PRPA_IN201308NO" root="2.16.840.1.113883.1.6"/>

```

```
<processingCode code="P"/>
<processingModeCode code="T"/>
<acceptAckCode code="NE"/>
<receiver typeCode="RCV">
  <device classCode="DEV" determinerCode="INSTANCE">
    <!-- Receiving software application. Helse Vest assigned application identifier -->
    <id extension="222" root="2.16.578.1.34.1"/>
  </device>
</receiver>
<sender typeCode="SND">
  <device classCode="DEV" determinerCode="INSTANCE">
    <!-- Sending software application. Helse Vest assigned application identifier -->
    <id extension="922" root="2.16.578.1.34.1"/>
  </device>
</sender>
<acknowledgement>
  <!-- Affirmative acknowledgement -->
  <typeCode code="AA"/>
  <targetMessage>
    <!-- .. related to this original message ID -->
    <id extension="080527104501_37" root="2.16.578.1.34.1.222.1"/>
  </targetMessage>
</acknowledgement>
<controlActProcess classCode="CACT" moodCode="EVN">
  <authorOrPerformer typeCode="AUT">
    <!-- Identifies the software application responsible for sending this message -->
    <assignedDevice classCode="ASSIGNED">
      <!-- Sending software application. Helse Vest assigned application identifier -->
      <id extension="922" root="2.16.578.1.34.1"/>
    </assignedDevice>
  </authorOrPerformer>
  <subject typeCode="SUBJ">
    <registrationEvent classCode="REG" moodCode="EVN">
      <!-- The ID of the registration process, a.k.a. the DIPS Internal patient ID. This ID isn't published. -->
      <id nullFlavor="MSK"/>
      <statusCode code="active"/>
      <subject1 typeCode="SBJ">
        <patient classCode="PAT">
          <!-- Response contains one of F/D/H number.. All others in OtherId.id -->
          <id root="2.16.578.1.12.4.1.4.1" extension="15076500565" assigningAuthorityName="F-nummer"/>
          <statusCode code="active"/>
          <patientPerson>
            <!-- ONE Person identifier from the Folkeregister: here: F number -->
```

```
<id root="2.16.578.1.12.4.1.4.1" extension="15076500565" assigningAuthorityName="F-
nummer"/>
<name use="L">
  <given>Roland</given>
  <family>Gundersen</family>
</name>
<!-- Gendercode: 0, 1, 2 or 9 -->
<administrativeGenderCode code="1" codeSystem="2.16.578.1.12.4.1.1.3101"/>
<birthTime value="19650715"/>
<addr use="HP">
  <streetAddressLine>Flåklypa 31</streetAddressLine>
  <postalCode>2560</postalCode>
  <city>Alvdal</city>
</addr>
<!-- Maritalstatus according to Folkeregister -->
<maritalStatusCode code="2" codeSystem="2.16.578.1.12.4.1.1.3103" displayName="Gift"/>
<birthPlace>
  <addr>
    <city>Oslo</city>
  </addr>
</birthPlace>
</patientPerson>
<providerOrganization classCode="ORG" determinerCode="INSTANCE">
  <!-- ID of Helse Vest RHF from the Enhetsregister; organization that is aware of the patient-role
of this person -->
  <id root="2.16.578.1.12.4.1.4.101" extension="983658725"/>
  <contactParty classCode="CON" nullFlavor="NA"/>
</providerOrganization>
<asMember classCode="MBR">
  <group classCode="PUB">
    <id extension="030104" root="2.16.578.1.12.4.1.1.3403"
assigningAuthorityName="Bydelsnummer"/>
    <code code="ADMENT" codeSystem="2.16.578.1.34.5 .1"/>
  </group>
</asMember>
</patient>
</subject1>
<custodian typeCode="CST">
  <assignedEntity classCode="ASSIGNED">
    <!-- ID of Helse Vest RHF from the Enhetsregister; organization that is aware of the patient-role of
this person -->
    <id root="2.16.578.1.12.4.1.4.101" extension="983658725"/>
    </assignedEntity>
  </custodian>
</registrationEvent>
```

```
</subject>
<queryAck>
  <!-- Unique identification of this query conversation. Copied from the query interaction -->
  <queryId extension="080527104501_37" root="2.16.578.1.34.1.222.1"/>
  <queryResponseCode code="OK"/>
  <resultCurrentQuantity value="1"/>
  <resultRemainingQuantity value="0"/>
</queryAck>
</controlActProcess>
</PRPA_IN201308NO>
```

3.1.3. *PatientRegistry.FindCandidates*

3.1.3.1. Operation-level Profile

Operation Name	PatientRegistry.FindCandidates
Purpose Description	To query, using a set of search criteria, the patient registry for the patient record (an identifier and associated demographics data) of a patient matching the search criteria. Note that each registry entry will have one identifier (only), alternative/secondary identifiers (e.g. an FH-number for a patient with a known F-number) won't be returned. The PatientRegistry.GetDemographics operation can be used to retrieve a full set of known alternative/secondary identifiers.
Logic Description	The operation will match the supplied search criteria, using an exact or 'fuzzy' match, with the properties as known for patients it holds the demographics data of, and will return between 0 (in case there wasn't a match) and 50 matching records.
Input/Output	Input: PatientRegistry.FindCandidates Query (PRPA_IN201305NO) Output: PatientRegistry.FindCandidates Query Response(PRPA_IN201306NO) See below for details of the information models as well as examples.
Composition Role	Business level service
Composition Member Capabilities	None
Keywords	Demographics data, patient, search, criteria, Folkeregister, Patientidentifier
Version	4.0

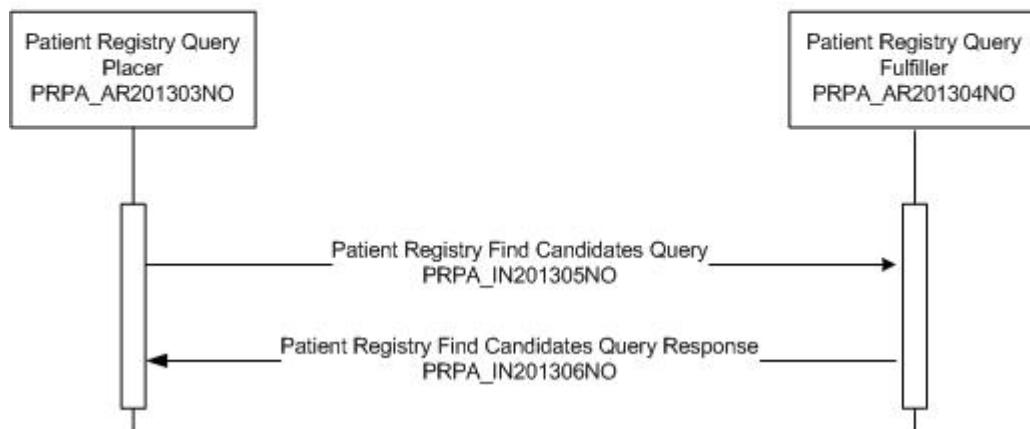
Finalization Status within HL7 Norway	Final
Custodian	HL7 Norway

3.1.3.2. Purpose of the operation

Storyboard diagram

This storyboard demonstrates querying a patient registry to retrieve potential matches based on partial patient demographic information.

Patient Registry Find Candidates Query PRPA_ST201305NO



Textual Storyboard

Dr. Patricia Primary is driving to work and witnesses a road traffic accident. One casualty is unconscious but Dr. Primary recognizes him as Mr. Everyman who came to see her last month. Dr. Primary assists the paramedics and provides as many demographic details for Mr. Everyman as she can remember.

At A&E Connor Comrade the receptionist initiates a Patient Registry Find Candidates Query for Mr. Everyman using sex, date of birth range and family name and initial and GP code provided by Dr. Primary.

Connor Comrade receives back a Patient Registry Find Candidates Query Response containing two matches to his Patient Registry Find Candidates Query for Mr. Everyman. Both records contain Registry ID number, name, sex, date of birth and current GP but one record also contains a date of death. Connor is looking at the record for the live patient when one of the paramedics comes past with the news that Mr. Everyman has come around

and given his date of birth. Connor checks that the record he is viewing matches Mr. Everyman's date of birth and selects it.

Patient Registry Find Single Candidate Query

Adam Everyman has a consultation with Dr. Fay Family today to discuss the likelihood that he has inherited Parkinson's disease. Adam's father died last year so Adam gives his permission for Dr. Family to look at his father John's records. Dr. Family enters John's name, sex, date of birth, date of death and postal code onto the system and performs a Patient Registry Find Candidates Query.

Dr. Family's search for Adam's father, John, is successful and only one match is found. A Patient Registry Find Candidates Query Response returns the Registry ID number, usual address, telephone numbers, current usual name, sex, date of birth, date of death, status of death notification and GP. She then uses this information to access John's records to see if any there is any clinical information relating to Parkinson's disease.

3.1.3.3. Input/Output

The querying system sends a query with a combination of the following query parameters: Date-of-birth, administrativeGender (sex), family name, middle name, given name and address.

- Family name
- Middle name
- Given name
- Sex
- Birth date (or an interval thereof)
- Address (notably street address)

Restrictions as to what combinations are applicable are up to the application that uses the service.

The query interaction has an immediate response. The response interaction as sent by the Patient Registry will contain a set of patient demographics data; the demographics data of patients that are as close a match to the supplied parameters as possible.

The HL7 interaction has been constrained: the service will return a maximum of 50 candidates.

Interaction List

Patient Registry Find Candidates Query	PRPA_IN201305NO
Patient Registry Find Candidates Query Response	PRPA_IN201306NO

Note that the above interactions use the "NO" realm code, and not the original "UV". The

models used are (currently) 100% equivalent to the international models. By using the NO realm the artefact identifier can be used for versioning purposes at some future point in time.

Input Information Model

The model is defined in the form of two wrappers (which contain meta-data related to the information exchange; the [Transmission Wrapper](#) and the [ControlAct Wrapper](#) and a so-called payload model). The payload model of the query interaction is shown here:

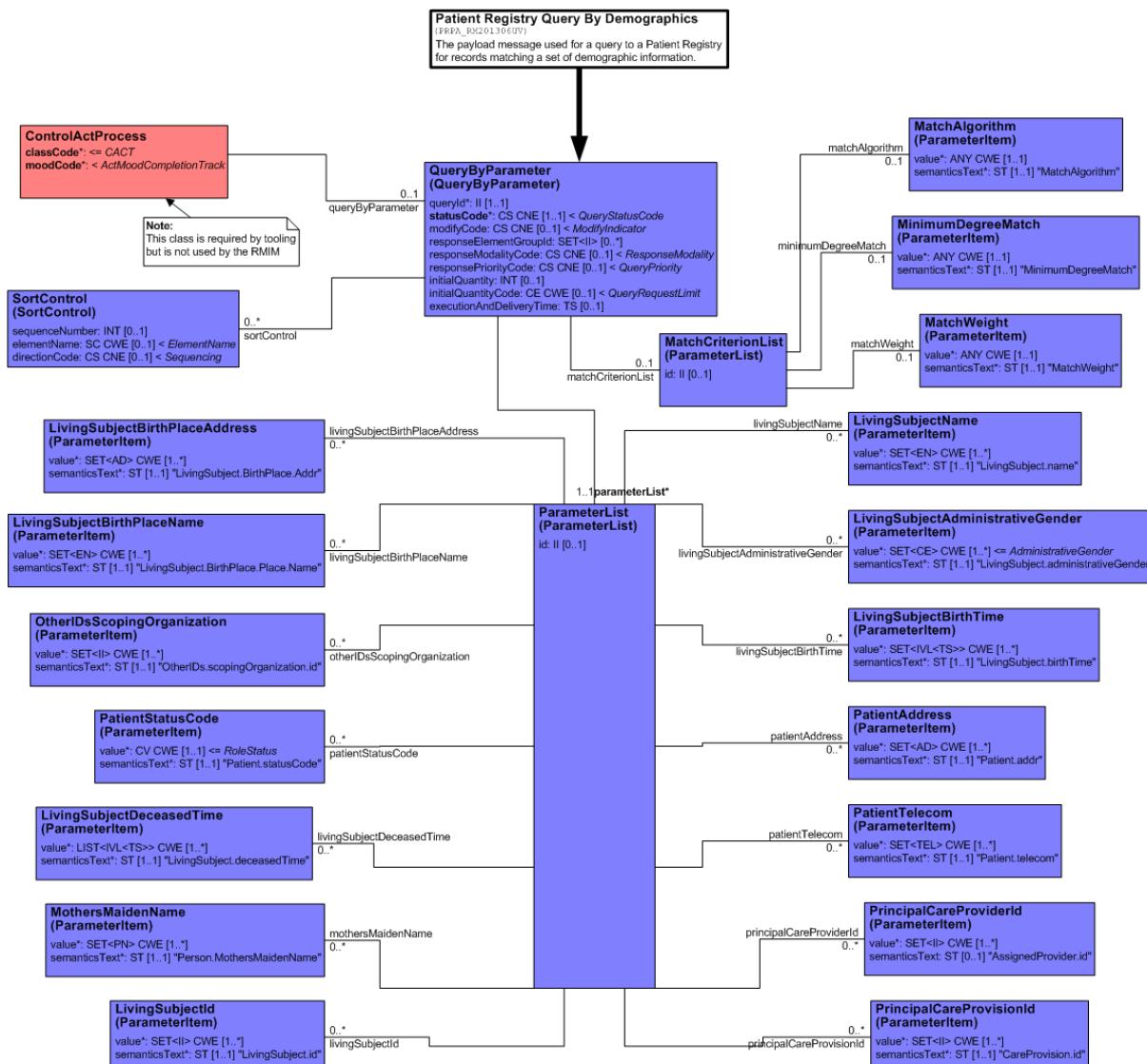


Figure 8 Source: Ballot May 2008

Class	Component	Documentation
QueryByParameter	queryId	The queryId contains the unique identification of this

		<p>query instance. See the description of the II data type used. The id contains the unique identification of this query instance.</p> <p>@root contains an identification of the ‘unique query instance numbering mechanism as used by this particular sending software application’, and @extension contains the identifier created according to that numbering mechanism. @root contains the ‘namespace’ of the identifier as contained in @extension.</p> <p>Implementation note: re-use same @extension/@root as used to identify the message instance in the Transmission Wrapper, see the documentation of the wrappers.</p>
	statusCode	@code contains the fixed value ‘new’.
	<i>Other</i>	<p>Any other elements not listed above, but shown in the diagram, are reserved for future use in Norway.</p> <p>Sending applications SHOULD not use these elements; receiving applications SHALL nor produce an error if these elements are present, and SHALL ignore these elements if present.</p>

Class	Component	Documentation
PatientAddress	value	Contains a (partial) address of the Patient whose details are being queried for. See AD Data Type specification for details.
	semanticsText	The element contains the fixed value ‘Patient.addr’.

Class	Component	Documentation
LivingSubjectAdministrativeGender	value	<p>Contains a code for the administrative gender of the patient being queried for.</p> <p>@codeSystem is fixed to 2.16.578.1.12.4.1.1.3101, @code should be either 0 (not known), 1 (male), 2 (female), or 9 (not applicable).</p> <p>Note that the Folkeregister (a Person registry) does only contain M and F, persons with an undifferentiated gender are encoded as F.</p>
	semanticsText	The element contains the fixed value ‘LivingSubject.administrativeGender’.

Class	Component	Documentation
LivingSubject BirthTime	value	<p>Contains a (partial) date of birth, or a range of dates, that should match the date of birth of the patient being queried for.</p> <p>Design Comments: A date or date range. This parameter can convey an exact moment (e.g., January 1, 1960 @ 03:00:00 EST), an approximate date (e.g., January 1960), or even a range of dates (e.g., December 1, 1959 through March 31, 1960). Multiple instances within a single parameter are combined with OR logic.</p> <p>For dates use YYYYMMDD.</p> <pre><value value="20100910"/></pre> <p>Example for daterange:</p> <pre><value> <low value="20100901"/> <high value="20100910"/> </value></pre>
	semanticsText	The element contains the fixed value 'LivingSubject.birthTime'.

Class	Component	Documentation
LivingSubjectName	value	Contains (part of) the name of the Patient whose details are being queried. See PN Data Type specification , inclusive of details on how to specify fuzzy names (using wildcards).
	semanticsText	The element contains the fixed value 'LivingSubject.name'.

Output Information Model

The model is defined in the form of two wrappers (which contain meta-data related to the information exchange; the [Transmission Wrapper](#) and the [ControlAct Wrapper](#) and a so-called payload model).

The payload model of the response interaction is documented on the [Generic Patient Model](#) page. Additionally the **QueryMatchObservation** class SHALL occur exactly once

for each matching person record. The QueryMatchObservation class can be used by the server to indicate the degree of match between the query parameters and the resulting record. The higher the degree of match, the more likely it is that the record matches the person record being searched for. It can be used by receiving applications to decide how to display (in what particular order) the matching records.

- Example: a query is done with parameters: (family name = juhanson), (gender = male), (birthdate = 1961).
 - The server finds a number of 'partially' matching records, and assigns a percentage to the degree of match:
 1. Match = 95%: Juhanson, Female, 1961
 2. Match = 90%: Johanson, Male, 1961
 3. Match = 70%: Junahson, Male, 1960

The algorithm according to which the server determines the 'degree of match' isn't specified by this implementation guide. The server has to assign a numerically higher degree of match, the more likely it thinks that a record matches the person record being searched for.

Note: if the server doesn't support the concept of a 'degree of match', one fixed numeric value should be used for all records in the response information model. This, in effect, states that all returned records share one and the same 'degree of match'.

Class	Component	Documentation
QueryMatchObservation	code	Indicates the kind of observation being done. @codeSystem contains the fixed value 2.16.578.1.34.5.2, @code contains the fixed value "PERC" (percentage).
	value	Contains the 'degree of match' expressed as a percentage (a Real number). Note: in XML instances an xsi:type="REAL" type override has to be used for this attribute.

```
<subjectOf1>
<queryMatchObservation moodCode="EVN" classCode="OBS">
  <code code="PERC" codeSystem="2.16.578.1.34.5.2"/>
  <value xsi:type="REAL" value="80"/>
</queryMatchObservation>
</subjectOf1>
```

3.1.3.4. Error Handling

The response may contain an indication of errors during the processing of the request. This operation only responds with an error if the query can't be processed. If a query leads to zero matching results this is not considered to be an error. See [Error Handling](#) for a general discussion of how errors can be identified in the response.

The error codes are taken from the PersonRegistryErrors coding system (managed/maintained by NHN), with OID 2.16.578.1.12.4.5.2.1.1.

In case of processing errors the value of Acknowledgement.typeCode will be set to "AE", and the value of the DetectedIssueEvent.code attribute will be set to one of the codes shown below.

Lvl	Code	Description	Documentation
1	INVALPID	Invalid person identifier	(One of) the person identifier(s) is invalid (null, empty, or not a valid F-, D-, or FH-number [a checksum failure])
2	PARAMERR	Generic Query Parameter	One of the query parameters doesn't conform to the specified data type (e.g. illegal characters in a timestamp), or the code specified is not contained in the required coding system (e.g. "Z" as an administrative gender).
1	SERVEROR	Generic error	Uncaught exception, either from code or database - will only occur if there is a bug in the link/unlink code.

3.1.3.5. XML Examples

Input Example

```
<PRPA_IN201305NO ITSVersion="XML_1.0" xmlns="urn:hl7-org:v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:hl7-org:v3 ..../schemas/PRPA_IN201305NO.xsd">
  <id extension="90204193108_33" root="2.16.578.1.34.1.805.1"/>
  <creationTime value="20090204193108"/>
  <versionCode code="NE2010NO"/>
  <interactionId extension="PRPA_IN201305NO" root="2.16.840.1.113883.1.6"/>
  <processingCode code="P"/>
  <processingModeCode code="T"/>
  <acceptAckCode code="NE"/>
  <receiver typeCode="RCV">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="922" root="2.16.578.1.34.1"/>
    </device>
  </receiver>
  <sender typeCode="SND">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="805" root="2.16.578.1.34.1"/>
    </device>
```

```
</sender>
<controlActProcess classCode="CACT" moodCode="EVN">
  <authorOrPerformer typeCode="AUT">
    <assignedPerson classCode="ASSIGNED">
      <id extension="987654" root="2.16.578.1.34.3.1"/>
    </assignedPerson>
  </authorOrPerformer>
  <queryByParameter>
    <queryId extension="90204193108_33" root="2.16.578.1.34.1.805.1"/>
    <statusCode code="new"/>
    <parameterList>
      <livingSubjectAdministrativeGender>
        <value codeSystem="2.16.578.1.12.4.1.1.3101" code="1"/>
        <semanticsText>LivingSubject.administrativeGender</semanticsText>
      </livingSubjectAdministrativeGender>
      <livingSubjectBirthTime>
        <value value="19650715"/>
        <semanticsText>LivingSubject.birthTime</semanticsText>
      </livingSubjectBirthTime>
    </parameterList>
  </queryByParameter>
</controlActProcess>
</PRPA_IN201305NO>
```

Output Example

```
<?xml version="1.0" encoding="UTF-8"?>
<PRPA_IN201306NO01 ITSPVersion="XML_1.0" xmlns="urn:hl7-org:v3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v3
..schemas/PRPA_IN201306NO.xsd">
    <id extension="9382399564" root="2.16.578.1.34.1.922.3"/>
    <creationTime value="20080622193112"/>
    <versionCode code="NE2010NO"/>
    <interactionId extension="PRPA_IN201306NO" root="2.16.840.1.113883.1.6"/>
    <processingCode code="P"/>
    <processingModeCode code="T"/>
    <acceptAckCode code="NE"/>
    <receiver typeCode="RCV">
        <device classCode="DEV" determinerCode="INSTANCE">
            <id extension="805" root="2.16.578.1.34.1"/>
        </device>
    </receiver>
    <sender typeCode="SND">
        <device classCode="DEV" determinerCode="INSTANCE">
            <id extension="922" root="2.16.578.1.34.1"/>
        </device>
    </sender>
    <acknowledgement typeCode="AA"/>
    <targetMessage>
        <id extension="80622193108_11" root="2.16.578.1.34.1.805.1"/>
    </targetMessage>
</acknowledgement>
<controlActProcess classCode="CACT" moodCode="EVN">
    <authorOrPerformer typeCode="AUT">
        <assignedDevice classCode="ASSIGNED">
            <id extension="922" root="2.16.578.1.34.1"/>
        </assignedDevice>
    </authorOrPerformer>
    <subject typeCode="SUBJ">
        <registrationEvent classCode="REG" moodCode="EVN">
            <id nullFlavor="UNK"/>
            <statusCode code="active"/>
            <subject1 typeCode="SBJ">
                <identifiedPerson classCode="IDENT">
                    <id root="2.16.578.1.12.4.1.4.1" extension="24109642356" assigningAuthorityName="F-
nummer"/>
```

```
<statusCode code="active"/>

<person>
  <name>
    <given>Ola</given>
    <given>Johan</given>
    <family>Hansen</family>
  </name>
  <administrativeGenderCode code="1" codeSystem="2.16.578.1.12.4.1.1.3101"/>
  <birthTime value="19961024"/>
  <addr use="H">
    <streetAddressLine>Parkveien 43</streetAddressLine>
    <postalCode>0258</postalCode>
    <city>Oslo</city>
  </addr>
  <asMember classCode="MBR">
    <group classCode="PUB">
      <id extension="030104" root="2.16.578.1.12.4.1.1.3403"
assigningAuthorityName="Bydelsnummer"/>
        <code code="ADMENT" codeSystem="2.16.578.1.34.5.1"/>
      </group>
    </asMember>
  </person>
  <subjectOf1>
    <queryMatchObservation moodCode="EVN" classCode="OBS">
      <code code="PERC" codeSystem="2.16.578.1.34.5.2"/>
      <value xsi:type="REAL" value="80"/>
    </queryMatchObservation>
  </subjectOf1>
</identifiedPerson>
</subject1>
<custodian typeCode="CST">
  <assignedEntity classCode="ASSIGNED">
    <id root="2.16.578.1.12.4.1.4.101" extension="983658725"/>
  </assignedEntity>
</custodian>
</registrationEvent>
</subject>
<subject typeCode="SUBJ">
  <registrationEvent classCode="REG" moodCode="EVN">
    <id nullFlavor="UNK"/>
    <statusCode code="active"/>
    <subject1 typeCode="SBJ">
      <identifiedPerson classCode="IDENT">
        <id root="2.16.578.1.12.4.1.4.1" extension="24109641234" assigningAuthorityName="F-
nummer"/>
      </identifiedPerson>
    </subject1>
  </registrationEvent>
</subject>
```

```
<statusCode code="active"/>

<person>
  <name>
    <given>Oluf</given>
    <given>Ronny</given>
    <family>Olsen</family>
  </name>
  <administrativeGenderCode code="1" codeSystem="2.16.578.1.12.4.1.1.3101"/>
  <birthTime value="19961024"/>
  <addr use="H">
    <streetAddressLine>Oppiveien 12</streetAddressLine>
    <postalCode>5254</postalCode>
    <city>Sandsli</city>
  </addr>
  <asMember classCode="MBR">
    <group classCode="PUB">
      <id extension="120106" root="2.16.578.1.12.4.1.1.3403"
assigningAuthorityName="Bydelsnummer"/>
        <code code="ADMENT" codeSystem="2.16.578.1.34.5.1"/>
    </group>
  </asMember>
</person>
<subjectOf1>
  <queryMatchObservation moodCode="EVN" classCode="OBS">
    <code code="PERC" codeSystem="2.16.578.1.34.5.2"/>
    <value xsi:type="REAL" value="80"/>
  </queryMatchObservation>
</subjectOf1>
</identifiedPerson>
</subject1>
<custodian typeCode="CST">
  <assignedEntity classCode="ASSIGNED">
    <id root="2.16.578.1.12.4.1.4.101" extension="983658725"/>
  </assignedEntity>
</custodian>
</registrationEvent>
</subject>
<queryAck>
  <queryId extension="80622193108_11" root="2.16.578.1.34.1.805.1"/>
  <queryResponseCode code="OK"/>
  <resultCurrentQuantity value="2"/>
  <resultRemainingQuantity value="0"/>
</queryAck>
</controlActProcess>
</PRPA_IN201306NO>
```

3.1.4. PatientRegistry.RecordRevised

3.1.4.1. Operation-level Profile

Operation Name	PatientRegistry.RecordRevised
Purpose Description	To update the demographical data of a single patient. See below for storyboards that illustrate the purpose of this operation.
Logic Description	The operation updates the demographical data of a patient, as long as it is known to the receiving system, and returns an accept acknowledgement.
Input/Output	Input: PatientRegistry.RecordRevised Notification (PRPA_IN201302NO) Output: Accept Acknowledgement (MCCI_IN0000002UV01) See below for details of the information models as well as examples.
Composition Role	Business level service
Composition Member Capabilities	None
Keywords	demographical data, patient, revised
Version	4.0
Finalization Status within HL7 Norway	Final
Custodian	HL7 Norway

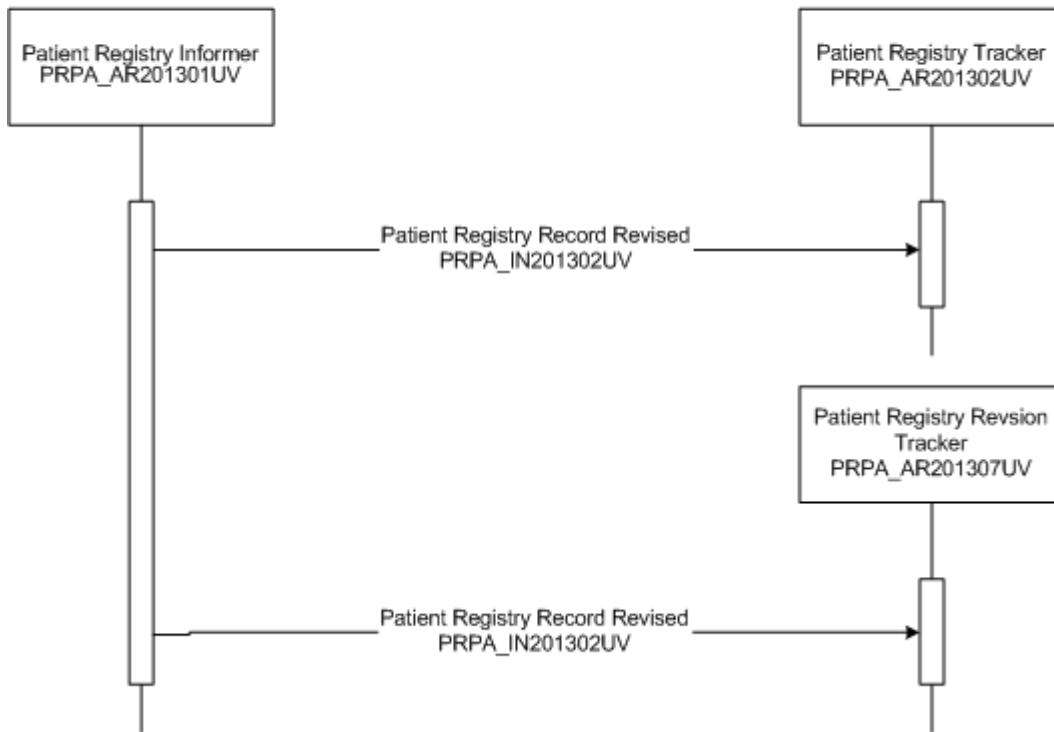
3.1.4.2. Purpose of the operation

The aim of the PatientRegistry.RecordRevised service is to update a patient registry with the demographic details of a given patient, based on the official identifier of the patient. In the current implementation, this identifier SHALL be the FH-number of the patient.

Storyboard Diagram

This storyboard demonstrates notifying tracking systems after a record is revised in a patient registry (note: The identifiers should end with “NO” instead of “UV”).

Patient Registry Record Revised PRPA_ST201302UV



Textual Storyboard

Dr. Patricia Primary scheduled an inpatient admission to the Good Health Hospital for her patient Mr. Adam Everyman and the registration clerk added Mr. Everyman to the GHH Patient Registry. The next day the Good Health Hospital's registration clerk, Christopher, called Mr. Everyman to gather additional demographic information and Mr. Everyman's emergency contact and guarantor information.

3.1.4.3. Input/Output

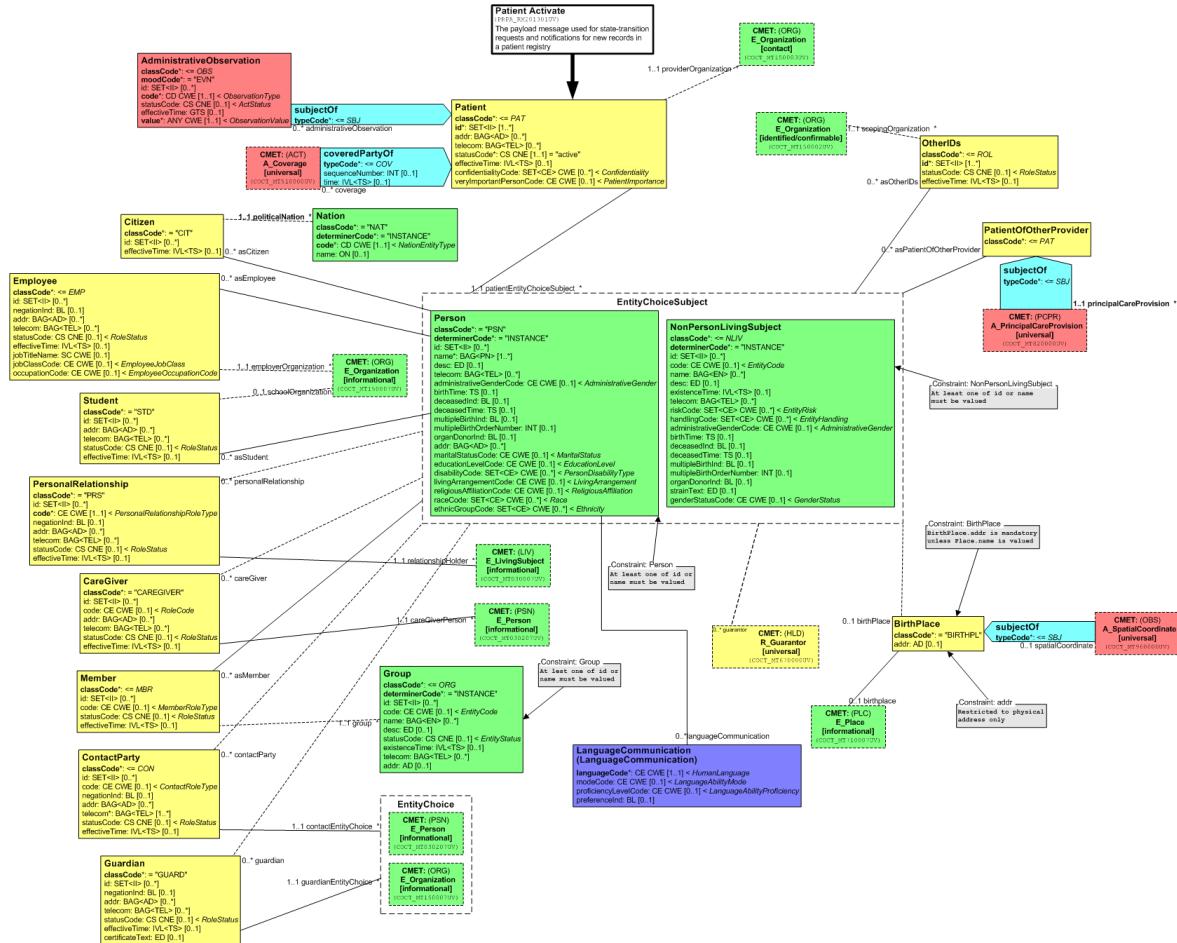
The notification interaction has an immediate response in the form of an [Accept Acknowledgement](#).

Interaction List

Patient Registry Record Revised Notification	PRPA_IN201302NO
Accept acknowledgement	MCCI_IN000002UV01

Input Information Model

The model is defined in the form of two wrappers (which contain meta-data related to the information exchange; the [Transmission Wrapper](#) and the [ControlAct Wrapper](#) and a so-called payload model). The payload model of the query interaction is shown here:



Figur 9 Source: Normative Edition 2008

See [Generic Patient Model](#) for more details.

Output Information Model

The output has the form of an [Accept Acknowledgement](#).

3.1.4.4. XML Examples

Input Example

```
<?xml version="1.0" encoding="UTF-8"?>
<PRPA_IN201302NO ITSVersion="XML_1.0" xmlns="urn:hl7-org:v3" xmlns:mif="urn:hl7-org:v3/mif"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <id extension="1109201010342045" root="2.16.578.1.34.1.408.7"/>
  <creationTime value="20100910103900"/>
  <versionCode code="NE2008"/>
  <interactionId root="2.16.840.1.113883.1.6" extension="PRPA_IN201302NO"/>
  <processingCode code="P"/>
  <processingModeCode code="T"/>
  <acceptAckCode code="AL"/>
```

```
<receiver typeCode="RCV">
  <device determinerCode="INSTANCE" classCode="DEV">
    <id extension="1" root="2.16.578.1.34.1.999"/>
  </device>
</receiver>
<sender typeCode="RCV">
  <device determinerCode="INSTANCE" classCode="DEV">
    <id extension="2" root="2.16.578.1.34.1.999"/>
  </device>
</sender>
<controlActProcess moodCode="EVN" classCode="CACT">
  <authorOrPerformer typeCode="AUT">
    <assignedPerson classCode="ASSIGNED">
      <id extension="1234" root="2.16.578.1.34.2.9999"/>
    </assignedPerson>
  </authorOrPerformer>
  <subject typeCode="SUBJ">
    <registrationEvent moodCode="EVN" classCode="REG">
      <id nullFlavor="UNK"/>
      <statusCode code="active"/>
      <subject1 typeCode="SBJ">
        <patient classCode="IDENT">
          <id extension="838912349234" root="2.16.578.1.12.4.1.4.3" assigningAuthorityName="FH-nummer"/>
          <statusCode code="active"/>
          <patientPerson>
            <name>
              <given>Ola</given>
              <given>Johan</given>
              <family>Hansen</family>
            </name>
            <administrativeGenderCode code="1" codeSystem="2.16.578.1.12.4.1.1.3101"/>
            <birthTime value="19750305"/>
            <addr use="H">
              <streetAddressLine>Parkveien 43</streetAddressLine>
              <postalCode>0258</postalCode>
              <city>Oslo</city>
            </addr>
            <asMember classCode="MBR">
              <group classCode="PUB">
                <id extension="030104" root="2.16.578.1.12.4.1.1.3403" assigningAuthorityName="Bydelsnummer"/>
                <code code="ADMENT" codeSystem="2.16.578.1.34.5 .1"/>
              </group>
            </asMember>
          </patientPerson>
        </patient>
      </registrationEvent>
    </subject>
  </controlActProcess>
```

```
</patientPerson>
</patient>
</subject1>
<custodian typeCode="CST">
  <assignedEntity classCode="ASSIGNED">
    <id extension="2" root="2.16.578.1.34.1.999"/>
  </assignedEntity>
</custodian>
</registrationEvent>
</subject>
</controlActProcess>
</PRPA_IN101302NO>
```

Output Example

See [Accept Acknowledgement](#).

3.2. PersonRegistry

3.2.1. PersonRegistry.AddPerson

This is a new use-case not (yet) covered by the international HL7 version 3 standard.

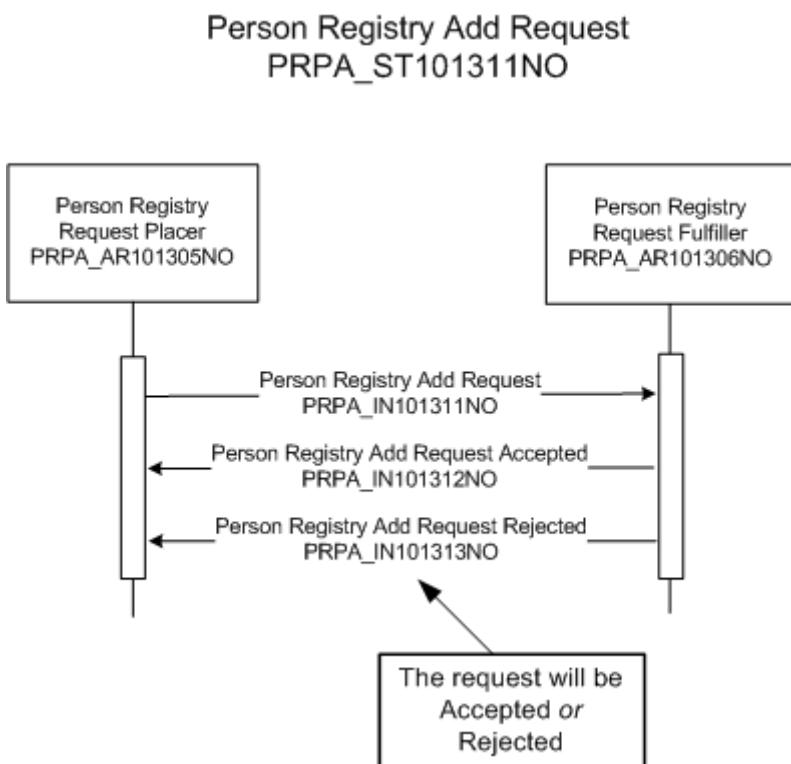
3.2.1.1. Operation-level Profile

Operation Name	PersonRegistry.AddPerson
Purpose Description	The aim of the PersonRegistry.AddPerson service is to request the receiving application to assign a unique identifier for a person. This service is used to request an FH-number.
Logic Description	The service creates unconditionally a new identity with the attributes from the request.
Input/Output	<p>Input: Person Registry Add Person Request (PRPA_IN101911NO). The core component is an instance of the Generic Person Model with the exception of the IdentifiedPerson.id attribute, which is not present in the context of the request to this service. It is allowed to have all attributes empty, confer with use case Emergency Alert described below.</p> <p>Output (OK): Person Registry Add Person Response (PRPA_IN101912NO). The core component is an instance of the Generic Person Model. The new person is identified by the IdentifiedPerson.id attribute.</p> <p>Output (Error): Person Registry Add Person Response (PRPA_IN101913NO).</p> <p>See below for details of the information models as well as examples.</p>
Composition Role	Business level service
Composition Member Capabilities	None
Keywords	FH-number, Demographics data, person, create, add, temporary person identifier, Folkeregister
Version	4.0
Finalization Status within HL7 Norway	Final
Custodian	HL7 Norway

3.2.1.2. Purpose of the operation

The storyboard demonstrates a local person registry sending a request to an enterprise (or national or regional) person registry to add a new record to the enterprise person registry. The enterprise person registry system will respond with a confirmation and the enterprise identifier for the new person record, or it will respond with a rejection and the reason for the rejection.

Storyboard Diagram



Textual Storyboard

Newborn

Nelda Nuclear delivers her baby son, Ned Nuclear, in Good Health Hospital. The hospital adds the newborn to the source system, assigning a medical record number for the baby, records the mother's unique identifier, mother's name, and newborn demographic data. An add client request is sent to the jurisdictional client registry system (Folkeregisteret). The Folkeregister sends an FH-number (Felles Hjelppenummer) in return to the hospital, to be used until the final social security number has been generated.

The Folkeregister adds the new person information to the registry and generates a unique client identifier which is returned in the add client confirmation message sent to the source system.

Fetus

The doctor Tom Expert will have to perform surgery on a fetus immediately after birth. The surgery has to be planned, and the baby needs its own identifier. The doctor's secretary Anna adds the baby along with the mother's demographics into the system. The system sends an add client request, and receives an FH-number to be used until the baby is born and has received its own social security number.

Dead born baby

Doctor Tom Begood wants to order laboratory tests and x-ray tests to investigate the cause of a dead born baby. He needs to identify the baby with a unique identifier. He asks for an FH-number from the Folkeregister. This baby will not receive a social security number later.

Tourists and other people without social security number

Fred Tourist has broken his arm and visits the Emergency Room. Fred Tourist does not have a Norwegian social security number, as he is from the Netherlands. Secretary Anna registrates his demographic information in the patient registry and asks for an FH-number from the Folkeregister service. This identifier is used for Fred Tourist as he is sent to the hospital for surgery of the complicated fracture.

John Doe is unconscious

The alarm central receives a call from Hans Hansen. He has found an unconscious person in the street. He doesn't know the person, but he can describe the patient as a male in his 50's, and he is located in Storgata. The alarm central orders an ambulance, alerts the hospital and registers the limited information about the man. The alarm central operator asks the Folkeregister for an identifier to be used to identify this John Doe. The Folkeregister returns an FH-number to be used until the person can be identified.

Emergency Alert

In case of larger catastrophes the Good Health Hospital orders several FH-numbers to be assigned to patients if necessary. This could be avalanches, car crashes, airplane accidents etc. The Good Health Hospital also needs a backup plan in case of down-time in the computer system.

Test patients

Marius Computerman needs to test his new system. To be able to test it properly, he needs to test it on persons in the Folkeregister. The Folkeregister flags the test person, and returns a test person with an FH-number.

3.2.1.3. Input/Output

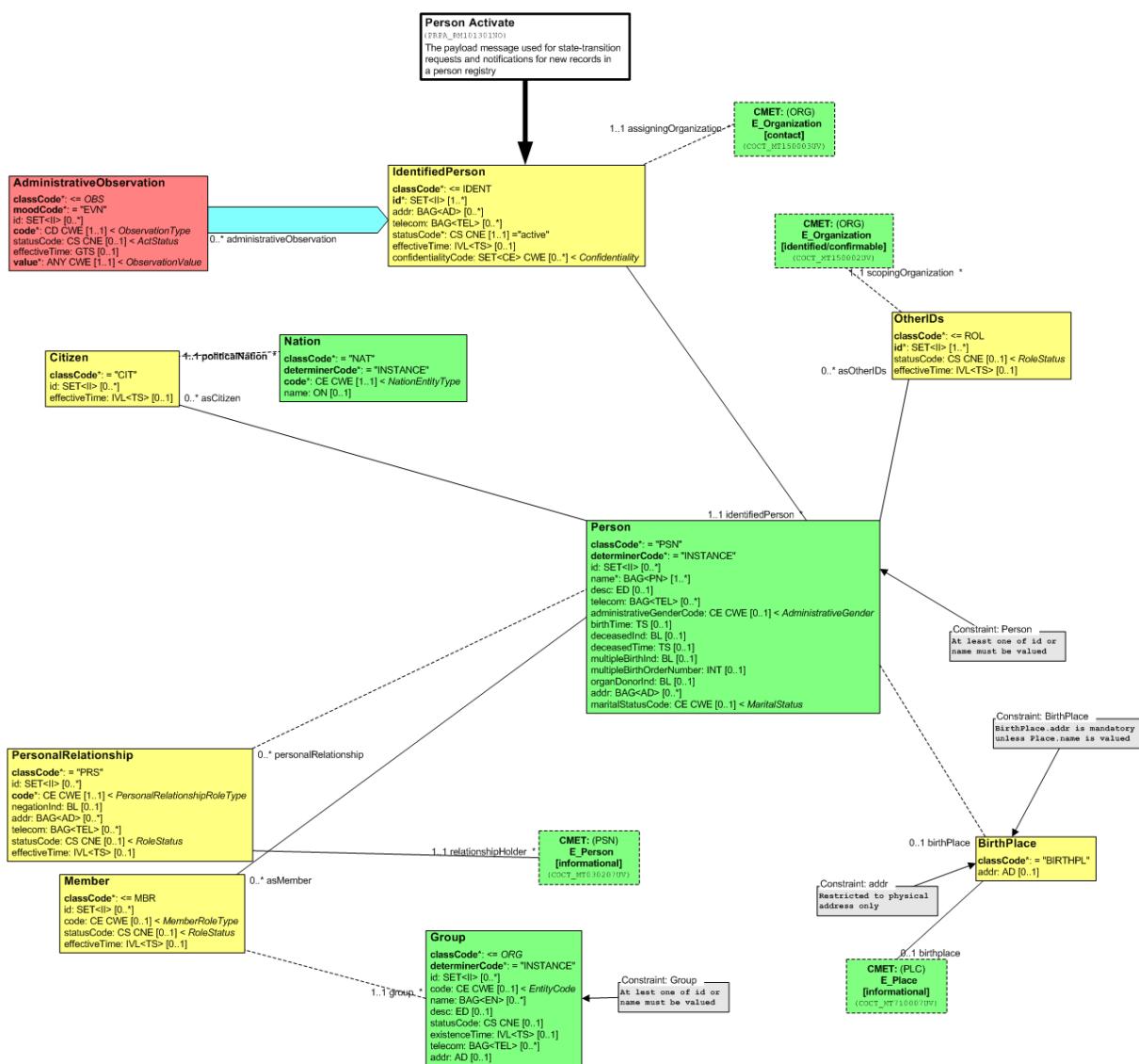
Person Registry Request Add Person	PRPA_IN101911NO
Person Registry Person Added	PRPA_IN101912NO
Person Registry Person Not Added	PRPA_IN101913NO

Note that the above interactions use the “NO” realm code. The models used are (currently) specific for the project developing this implementation guide and will be brought forward for inclusion in the international standard.

Input Information Model

The model is defined in the form of two wrappers (which contain meta-data related to the information exchange); the [Transmission Wrapper](#), the [ControlAct Wrapper](#), the [Registration Act](#) (a RegistrationRequest) and a so-called payload model.

This model is for illustration purposes only. The model is equal to the model documented as [Generic Person Model](#), with the exception of the IdentifiedPerson.id attribute, which is not present in this context.



Figur 10 Illustration

Output Information Model

The query interaction has an immediate response. The response interaction as sent by the Person Registry will contain a new unique identifier for the person.

The model is defined in the form of two wrappers (which contain meta-data related to the information exchange); the [Transmission Wrapper](#), the [ControlAct Wrapper](#), the [Registration Act](#) (a RegistrationEvent) and a so-called payload model. Since the operation is supposed to be performed unconditionally, possible errors are only related to the transmission and interpretation of the message, e.g. syntax conformance and code usage, confer [Error Handling](#).

The payload model is equal to the model documented above.

Error Handling

One of the errors is "no more FH-numbers available".

3.2.1.4. XML Examples

Input Example

```
<?xml version="1.0" encoding="UTF-8"?>
<PRPA_IN101911NO ITSPVersion="XML_1.0" xmlns="urn:hl7-org:v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v3
  ./schemas/PRPA_IN101911NO.xsd">
  <id extension="90204193108_33" root="2.16.578.1.34.1.805.1"/>
  <creationTime value="20090204193108"/>
  <versionCode code="NE2008"/>
  <interactionId extension="PRPA_IN101911NO" root="2.16.840.1.113883.1.6"/>
  <processingCode code="P"/>
  <processingModeCode code="T"/>
  <acceptAckCode code="NE"/>
  <receiver typeCode="RCV">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="922" root="2.16.578.1.34.1"/>
    </device>
  </receiver>
  <sender typeCode="SND">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="805" root="2.16.578.1.34.1"/>
    </device>
  </sender>
```

```
<controlActProcess classCode="CACT" moodCode="EVN">
  <authorOrPerformer typeCode="AUT">
    <assignedPerson classCode="ASSIGNED">
      <id extension="987654" root="2.16.578.1.34.3.1"/>
    </assignedPerson>
  </authorOrPerformer>
  <queryByParameter>
    <queryId extension="90204193108_33" root="2.16.578.1.34.1.805.1"/>
    <statusCode code="new"/>
    <parameterList>
      <personAdministrativeGender>
        <value codeSystem="2.16.578.1.12.4.1.1.3101" code="1"/>
        <semanticsText>LivingSubject.administrativeGender</semanticsText>
      </personAdministrativeGender>
      <personBirthTime>
        <value value="19961024"/>
        <semanticsText>LivingSubject.birthTime</semanticsText>
      </personBirthTime>
    </parameterList>
  </queryByParameter>
</controlActProcess>
</PRPA_IN101911NO>
```

Output Example

```
<?xml version="1.0" encoding="UTF-8"?>
<PRPA_IN101912NO ITSType="XML_1.0" xmlns="urn:hl7-org:v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v3
  ./schemas/PRPA_IN101912NO.xsd">
  <id extension="9382923825" root="2.16.578.1.34.1.922.3"/>
  <creationTime value="20080618105503"/>
  <versionCode code="NE2008"/>
  <interactionId extension="PRPA_IN101912NO" root="2.16.840.1.113883.1.6"/>
  <processingCode code="P"/>
  <processingModeCode code="T"/>
  <acceptAckCode code="NE"/>
  <receiver typeCode="RCV">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="805" root="2.16.578.1.34.1"/>
    </device>
  </receiver>
  <sender typeCode="SND">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="922" root="2.16.578.1.34.1"/>
    </device>
```

```
</sender>
<acknowledgement>
  <typeCode code="AA"/>
  <targetMessage>
    <id extension="080618105502_8" root="2.16.578.1.34.1.805.1"/>
  </targetMessage>
</acknowledgement>
<controlActProcess classCode="CACT" moodCode="EVN">
  <authorOrPerformer typeCode="AUT">
    <assignedDevice classCode="ASSIGNED">
      <id extension="922" root="2.16.578.1.34.1"/>
    </assignedDevice>
  </authorOrPerformer>
  <subject typeCode="SUBJ">
    <registrationEvent classCode="REG" moodCode="EVN">
      <id nullFlavor="UNK"/>
      <statusCode code="active"/>
      <subject1 typeCode="SBJ">
        <identifiedPerson classCode="IDENT">
          <id root="2.16.578.1.12.4.1.4.3" extension="88888888843">
            assigningAuthorityName="felles-hjelppenummer"/>
          <statusCode code="active"/>
          <identifiedPerson>
            <id root="2.16.578.1.12.4.1.4.3" extension="88888888843"/>
            <administrativeGenderCode code="1" codeSystem="2.16.578.1.12.4.1.1.3101"/>
            <birthTime value="19961024"/>
          </identifiedPerson>
        <assigningOrganization determinerCode="INSTANCE" classCode="ORG">
          <id root="2.16.578.1.12.4.1.4.101" extension="983658725"/>
          <contactParty classCode="CON" nullFlavor="NA"/>
        </assigningOrganization>
      </identifiedPerson>
    </subject1>
    <custodian typeCode="CST">
      <assignedEntity classCode="ASSIGNED">
        <id root="2.16.578.1.12.4.1.4.101" extension="983658725"/>
      </assignedEntity>
    </custodian>
  </registrationEvent>
</subject>
<queryAck>
  <queryId extension="080618105502_8" root="2.16.578.1.34.1.805.1"/>
  <queryResponseCode code="OK"/>
  <resultCurrentQuantity value="1"/>
  <resultRemainingQuantity value="0"/>
```

```
</queryAck>  
</controlActProcess>  
</PRPA_IN101912NO>
```

Remarks

If using Normative Edition 2008 schemas (NE2008), there is a bug there stating that "ContactParty" is mandatory. E.g. it has to be present, but with a nullFlavor.

3.2.2. PersonRegistry.GetDemographics

3.2.2.1. Operation-level Profile

Operation Name	PersonRegistry.GetDemographics
Purpose Description	<p>To query (using a known identifier, e.g. F/D/FH/H-number) the person registry for the demographic details of a single person.</p> <p>Note that it is the expectation that any application that (re-)activates a person record should use the GetDemographics service to get hold of the latest data. Part of the data returned may be an indication that (for example) the D-number used has now been merged with an F-number.</p> <p>See below for storyboards that illustrate the purpose of this operation.</p>
Logic Description	The operation will match the supplied identifier, using an exact match, with the identifiers as known for persons it holds the demographics data of, and will return either 0 (in case there wasn't a match) or 1 matching records. Alternate (less preferable, less reliable) identifiers of the matching person will be included in the matching record.
Input/Output	<p>Input: Person Registry Get Demographics Query (PRPA_IN101307NO). The core of this model is the query parameter that holds the Person Identifier.</p> <p>Output: Person Registry Get Demographics Query Response (PRPA_IN101308NO). The core of this model is formed by the Generic Person Model, inclusive of the linkages from other person identifiers to the main person class.</p> <p>See below for details of the information models as well as examples.</p>
Composition Role	Business level service
Composition Member	None

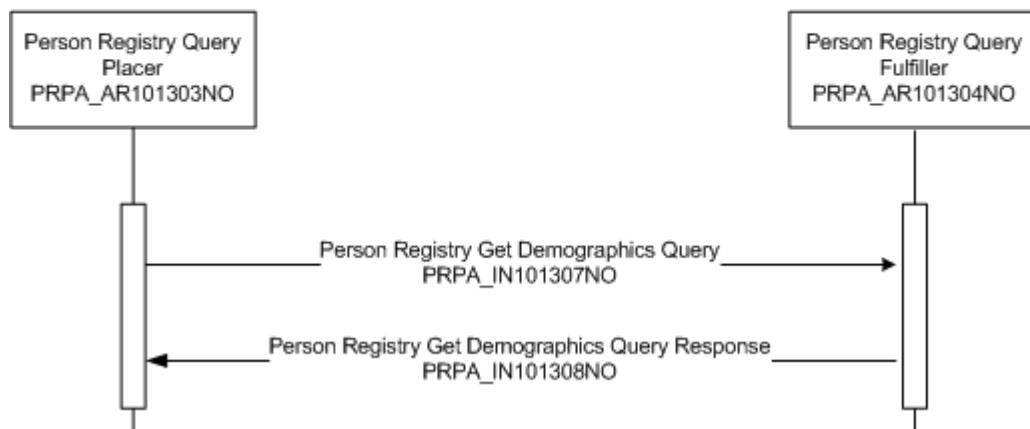
Capabilities	
Keywords	Demographics data, person, search, person identifier, Folkeregister
Version	5.0
Finalization Status within HL7 Norway	Final
Custodian	HL7 Norway

3.2.2.2. Purpose of the operation

The storyboard below demonstrates retrieving the demographic data associated with a specified person identifier. The query will be either based on the F-number, the D-number, the H-number or the FH-number.

Storyboard Diagram

Person Registry Get Demographics Query
PRPA_ST101307NO



Note that the service is represented by the actor on the right hand side of this diagram.

Textual Storyboard

Dr. Alan Admit, who has admitting privileges at Good Health Hospital, notified the GHH Registration Office that his patient, Adam Everyman, would arrive at the hospital at a scheduled time, and that his office had performed a pre-admit on Adam Everyman.

As part of the admitting process, the GHH Registration clerk Alice Admitter entered Mr. Everyman's person identifier and initiated a PersonRegistry.GetDemographics operation. The operation returned Mr. Everyman's current demographic information and Alice confirmed this information with Mr. Everyman as she completed the admission process.

3.2.2.3. Input/Output

The query (input) model contains the person identifier. The output format will contain the F-number (and/or D-number) of the person as well as the demographics details of the person.

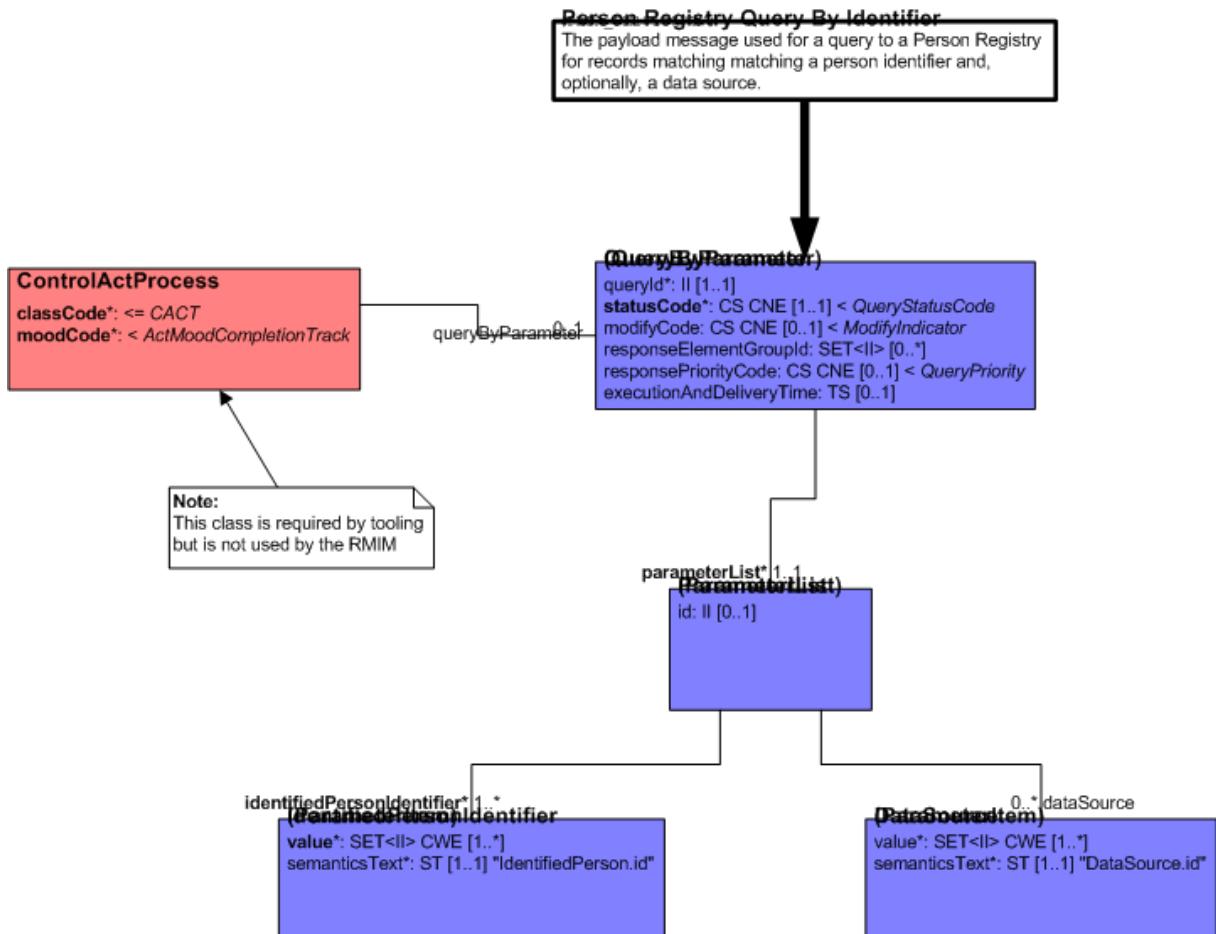
Input/Output List

Person Registry Get Demographics Query	PRPA_IN101307NO01
Person Registry Get Demographics Query Response	PRPA_IN101308NO01

Note that the above interactions use the “NO” realm code, and not the original “UV”. The models used are (currently) 100% equivalent to the international models. By using the NO realm the artefact identifier can be used for versioning purposes at some future point in time.

Input Information Model

The model is defined in the form of two wrappers (which contain meta-data related to the information exchange; the [Transmission Wrapper](#) and the [ControlAct Wrapper](#)) and a so-called payload model. The payload model of the query interaction is shown here:



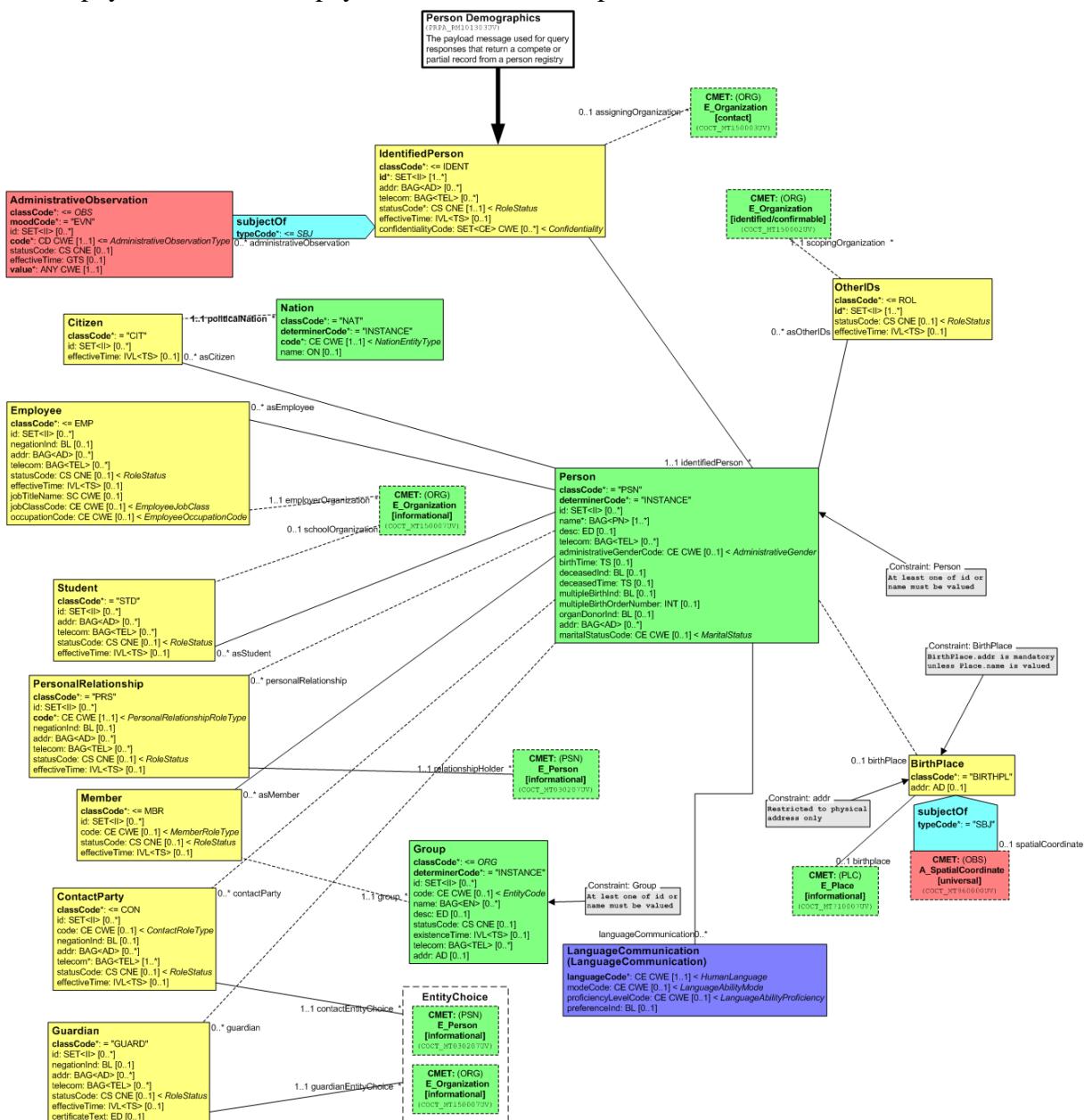
Figur 11 Source: Normative Edition 2008

Class	Component	Documentation
QueryByParameter	queryId	<p>The queryId contains the unique identification of this query instance. See the description of the II data type used. The id contains the unique identification of this query instance.</p> <p>@root contains an identification of the ‘unique query instance numbering mechanism as used by this particular sending software application’, and @extension contains the identifier created according to that numbering mechanism. @root contains the ‘namespace’ of the identifier as contained in @extension.</p> <p>Implementation note: re-use same @extension/@root as used to identify the message instance in the Transmission Wrapper, see the documentation of the wrappers.</p>
	statusCode	@code contains the fixed value ‘new’.
	Other	Any other elements not listed above, but shown in the diagram, are reserved for future use in Norway. Sending applications SHOULD not use these elements; receiving applications SHALL nor produce an error if these elements are present, and SHALL ignore these elements if present.

Class	Component	Documentation
PersonIdentifier	value	<p>Contains the person identifier. This is a unique identification of a person.</p> <p>See Identification of Patients and Persons for guidance related to the identification of persons, and the documentation of the II data type.</p> <p>@root contains an identification of the ‘unique personidentification mechanism’ (i.e. the OID for F-number, D-number, H-number or FH-number), and @extension contains the identifier created according to that identification mechanism.</p>
	semanticsText	The element contains the fixed value ‘Person.id’.

Output Information Model

The model is defined in the form of two wrappers (which contain meta-data related to the information exchange; the [Transmission Wrapper](#) and the [ControlAct Wrapper](#) and a so-called payload model. The payload model of the response interaction is shown here:



Figur 12 Source: Normative Edition 2008

See [Generic Person Model](#) for more details.

3.2.2.4. XML Examples

Input Example

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

```
<PRPA_IN101307NO01 ITSPVersion="XML_1.0" xmlns="urn:hl7-org:v3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v3
..schemas/PRPA_IN101307NO.xsd">
<id extension="080618105502_8" root="2.16.578.1.34.1.805.1"/>
<creationTime value="20080618105502"/>
<versionCode code="NE2010NE"/>
<interactionId extension="PRPA_IN101307NO01" root="2.16.840.1.113883.1.6"/>
<processingCode code="P"/>
<processingModeCode code="T"/>
<acceptAckCode code="NE"/>
<receiver typeCode="RCV">
<device classCode="DEV" determinerCode="INSTANCE">
<id extension="922" root="2.16.578.1.34.1"/>
</device>
</receiver>
<sender typeCode="SND">
<device classCode="DEV" determinerCode="INSTANCE">
<id extension="805" root="2.16.578.1.34.1"/>
</device>
</sender>
<controlActProcess classCode="CACT" moodCode="EVN">
<authorOrPerformer typeCode="AUT">
<assignedPerson classCode="ASSIGNED">
<id extension="987654" root="2.16.578.1.34.3.1"/>
</assignedPerson>
</authorOrPerformer>
<queryByParameter>
<queryId extension="080618105502_8" root="2.16.578.1.34.1.805.1"/>
<statusCode code="new"/>
<parameterList>
<identifiedPersonIdentifier>
<value root="2.16.578.1.12.4.1.4.1" extension="17109012343" assigningAuthorityName="F-
nummer"/>
<semanticsText>IdentifiedPerson.id</semanticsText>
</identifiedPersonIdentifier>
</parameterList>
</queryByParameter>
</controlActProcess>
</PRPA_IN101307NO01>
```

Output Example

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

```
<PRPA_IN101308NO01 ITSType="XML_1.0" xmlns="urn:hl7-org:v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:hl7-org:v3 ..schemas/PRPA_IN101308NO.xsd">
  <id extension="9382923825" root="2.16.578.1.34.1.922.3"/>
  <creationTime value="20080618105503"/>
  <versionCode code="NE2010NO"/>
  <interactionId extension="PRPA_IN101308NO01" root="2.16.840.1.113883.1.6"/>
  <processingCode code="P"/>
  <processingModeCode code="T"/>
  <acceptAckCode code="NE"/>
  <receiver typeCode="RCV">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="805" root="2.16.578.1.34.1"/>
    </device>
  </receiver>
  <sender typeCode="SND">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="922" root="2.16.578.1.34.1"/>
    </device>
  </sender>
  <acknowledgement typeCode="AA"/>
  <targetMessage>
    <id extension="080618105502_8" root="2.16.578.1.34.1.805.1"/>
  </targetMessage>
</acknowledgement>
<controlActProcess classCode="CACT" moodCode="EVN">
  <authorOrPerformer typeCode="AUT">
    <assignedDevice classCode="ASSIGNED">
      <id extension="922" root="2.16.578.1.34.1"/>
    </assignedDevice>
  </authorOrPerformer>
  <subject typeCode="SUBJ">
    <registrationEvent classCode="REG" moodCode="EVN">
      <id nullFlavor="UNK"/>
      <statusCode code="active"/>
      <subject1 typeCode="SBJ">
        <identifiedPerson classCode="IDENT">
          <id root="2.16.578.1.12.4.1.4.1" extension="17109012343" assigningAuthorityName="F-
nummer"/>
          <statusCode code="active"/>
          <identifiedPerson>
            <id root="2.16.578.1.12.4.1.4.1" extension="17109012343"/>
            <name use="L">
              <given>Ole</given>
              <given>Helge</given>
```

```
<family>Duck</family>
</name>
<administrativeGenderCode code="1" codeSystem="2.16.578.1.12.4.1.1.3101"
displayName="Mann"/>
<birthTime value="19901017"/>
<addr use="HP">
<streetAddressLine>Apalveien 13</streetAddressLine>
<postalCode>3162</postalCode>
<city>Andebu</city>
</addr>
<maritalStatusCode code="1" codeSystem="2.16.578.1.12.4.1.1.3103" displayName="Ugift"/>
</identifiedPerson>
<assigningOrganization determinerCode="INSTANCE" classCode="ORG">
<id root="2.16.578.1.12.4.1.4.101" extension="983658725"/>
<contactParty classCode="CON" nullFlavor="NA"/>
</assigningOrganization>
<identifiedBy typeCode="IDENT">
<statusCode code="active"/>
<otherIdentifiedPerson classCode="IDENT">
<id root="2.16.578.1.12.4.1.4.2" extension="64109642356"
assigningAuthorityName="D-nummer"/>
</otherIdentifiedPerson>
</identifiedBy>
</identifiedPerson>
</subject1>
<custodian typeCode="CST">
<assignedEntity classCode="ASSIGNED">
<id root="2.16.578.1.12.4.1.4.101" extension="983658725"/>
</assignedEntity>
</custodian>
</registrationEvent>
</subject>
<queryAck>
<queryId extension="080618105502_8" root="2.16.578.1.34.1.805.1"/>
<queryResponseCode code="OK"/>
<resultCurrentQuantity value="1"/>
<resultRemainingQuantity value="0"/>
</queryAck>
</controlActProcess>
</PRPA_IN101308NO01>
```

3.2.3. PersonRegistry.FindCandidates

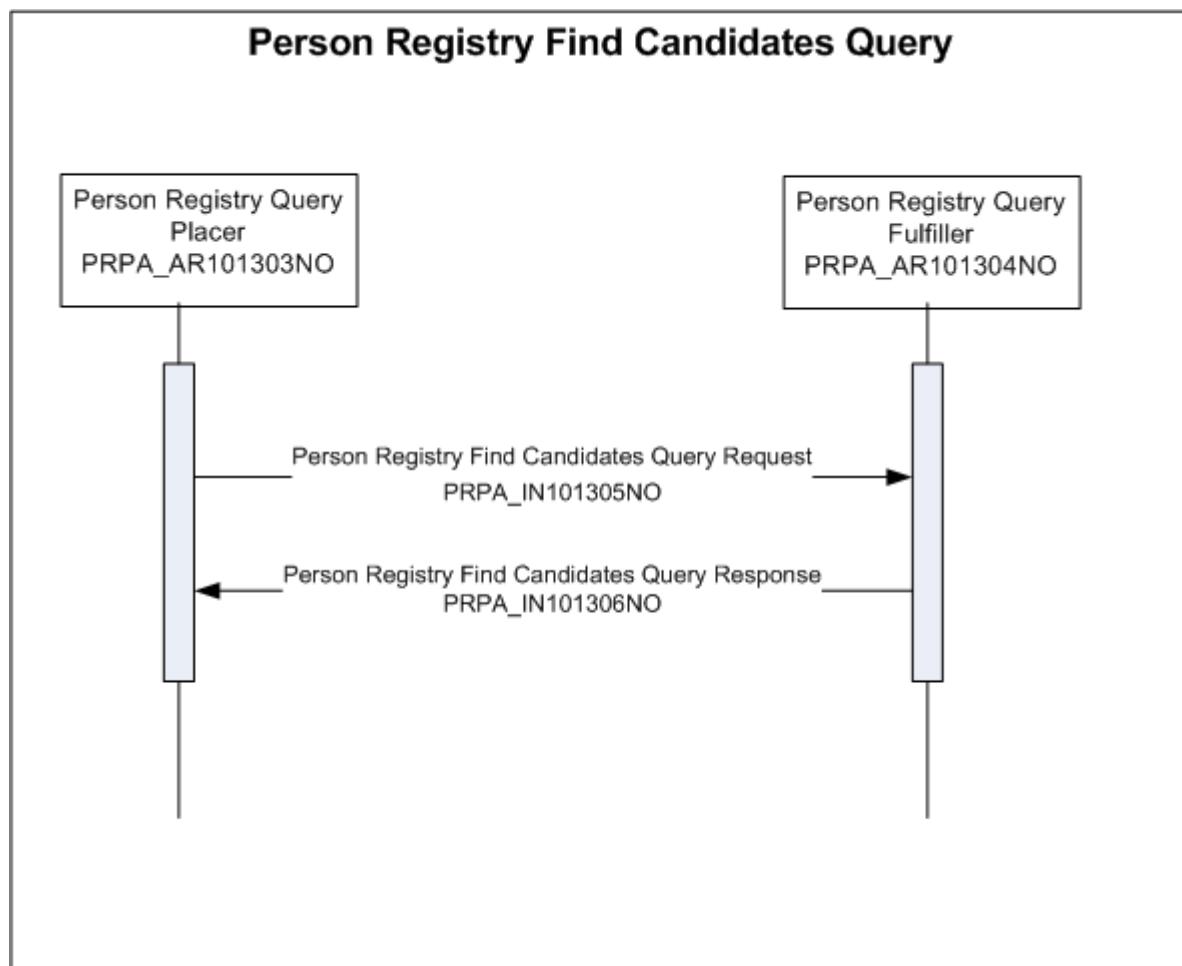
3.2.3.1. Operation-level Profile

Operation Name	PersonRegistry.FindCandidates
Purpose Description	To query, using a set of search criteria, the person registry for the person record (an identifier and associated demographics data) of a person matching the search criteria. Note that each registry entry will have one identifier (only), alternative/secondary identifiers (e.g. an FH-number for a person with a known F-number) won't be returned. The PersonRegistry.GetDemographics operation can be used to retrieve a full set of known alternative/secondary identifiers.
Logic Description	The operation will match the supplied search criteria, using an exact or 'fuzzy' match, with the properties as known for persons it holds the demographics data of, and will return between 0 (in case there wasn't a match) and 50 matching records.
Input/Output	<p>Input: PersonRegistry.FindCandidates Query (PRPA_IN101305NO01)</p> <p>Output: PersonRegistry.FindCandidates Query Response (PRPA_IN101306NO01)</p> <p>See below for details of the information models as well as examples.</p>
Composition Role	Business level service
Composition Member Capabilities	None
Keywords	Demographics data, person, search, criteria, Folkeregister, Personidentifier
Version	5.0
Finalization Status within HL7 Norway	Final
Custodian	HL7 Norway

3.2.3.2. Purpose of the operation

Storyboard Diagram

This storyboard demonstrates querying a person registry to retrieve potential matches based on partial person demographic information. (PRPA_ST101305UV02).



Textual Storyboard

Dr. Harold Hippocrates, who has admitting privileges at Good Health Hospital, requested person demographics for his patient, Adam Everyman. Adam Everyman did not remember his social security number (Fødselsnummer).

The GHH medical record clerk, Christopher Clerk, first determined that Mr. Everyman was not already registered at the Good Health Hospital. Christopher then went online and entered Mr. Everyman's last name, sex and birth date and initiated a Person Registry Find Candidates Query query to the Norwegian National Population Registry (Folkeregisteret). The person registry returned a Person Registry Find Candidates Query Response with the collection of candidate records allowing Christopher Clerk to refine the query for Adam Everyman, if desired.

3.2.3.3. Input/Output

The querying system sends a query with a combination of the following query parameters: Date-of-birth, administrativeGender (sex), family name, middle name, given name, address, deceased and administrative entity.

- Family name
- Middle name
- Given name
- Sex
- Birth date (or an interval thereof)
- Address (notably street address)
- Deceased indicator (true or false)
- Administrative Entity (Fylke/Kommune/Bydel)

Restrictions as to what combinations are applicable are up to the application that uses the service.

The query interaction has an immediate response. The response interaction as sent by the Person Registry will contain a set of patient demographics data; the demographics data of patients that are as close a match to the supplied parameters as possible.

The HL7 interaction has been constrained: the service will return a maximum of 50 candidates.

Interaction List

Person Registry Find Candidates Query	PRPA_IN101305NO01
Person Registry Find Candidates Query Response	PRPA_IN101306NO01

Note that the above interactions use the “NO” realm code, and not the original “UV”. The models used are (currently) 100% equivalent to the international models. By using the NO realm the artefact identifier can be used for versioning purposes at some future point in time.

Input Information Model

The model is defined in the form of two wrappers (which contain meta-data related to the information exchange; the [Transmission Wrapper](#) and the [ControlAct Wrapper](#) and a so-called payload model). The payload model of the query interaction is shown here:

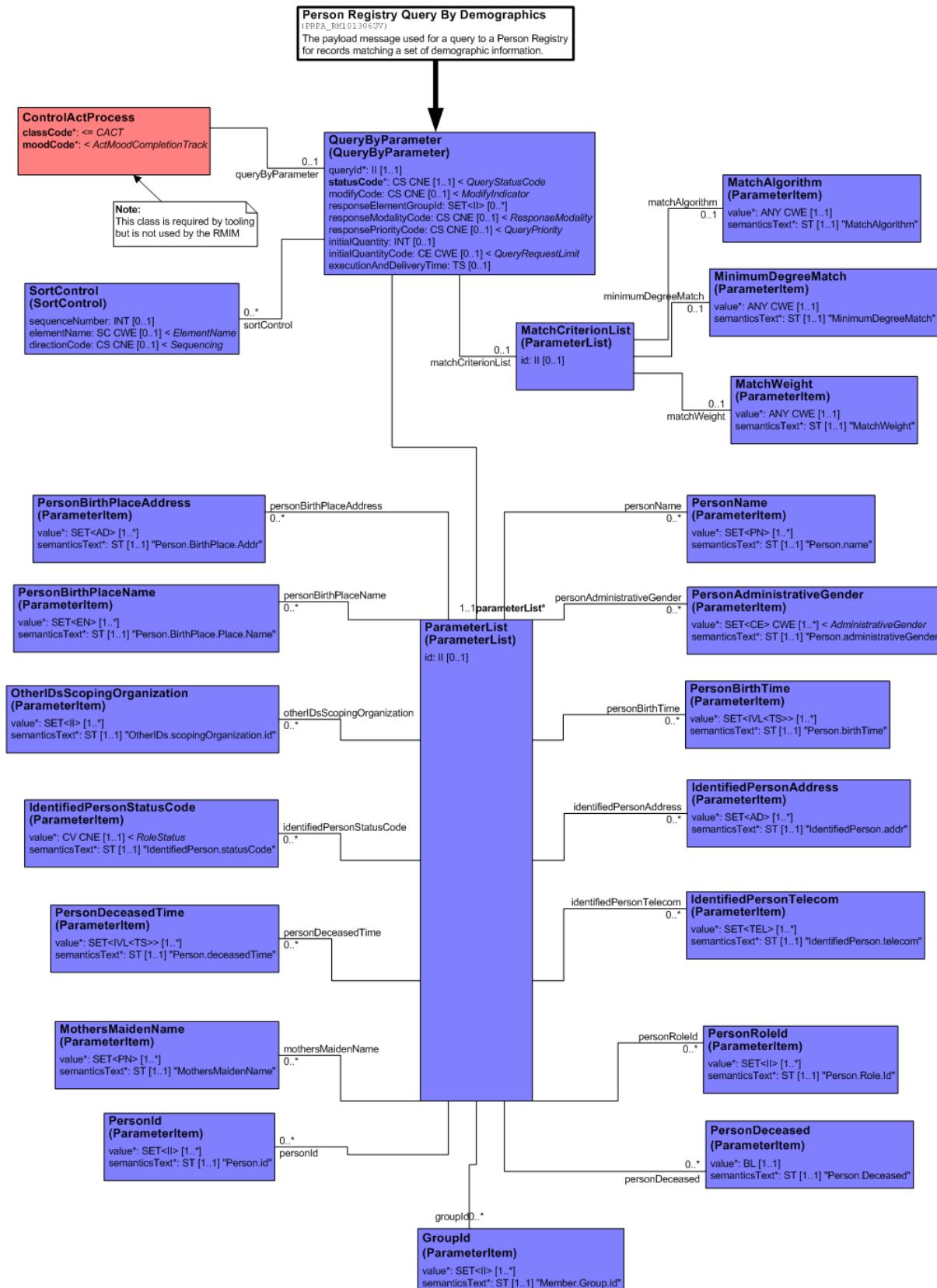


Figure 13 Source: Normative Edition 2008

Class	Component	Documentation
-------	-----------	---------------

QueryByParameter	queryId	<p>The queryId contains the unique identification of this query instance. See the description of the II data type used. The id contains the unique identification of this query instance.</p> <p>@root contains an identification of the ‘unique query instance numbering mechanism as used by this particular sending software application’, and @extension contains the identifier created according to that numbering mechanism. @root contains the ‘namespace’ of the identifier as contained in @extension.</p> <p>Implementation note: re-use same @extension/@root as used to identify the message instance in the Transmission Wrapper, see the documentation of the wrappers.</p>
	statusCode	@code contains the fixed value ‘new’.
	<i>Other</i>	<p>Any other elements not listed above, but shown in the diagram, are reserved for future use in Norway. Sending applications SHOULD not use these elements; receiving applications SHALL nor produce an error if these elements are present, and SHALL ignore these elements if present.</p>

Class	Component	Documentation
IdentifiedPersonAddress	value	Contains a (partial) address of the Person whose details are being queried for. See AD Data Type specification for details.
	semanticsText	The element contains the fixed value ‘IdentifiedPerson.addr’.

Class	Component	Documentation
PersonAdministrativeGender	value	<p>Contains a code for the administrative gender of the person being queried for.</p> <p>@codeSystem is fixed to 2.16.578.1.12.4.1.1.3101, @code should be either 0 (not known), 1 (male), 2 (female), or 9 (not applicable).</p> <p>Note that the Folkeregister (a Person registry) does only contain M and F, persons with an undifferentiated gender are encoded as F.</p>
	semanticsText	The element contains the fixed value ‘Person.administrativeGender’.

Class	Component	Documentation
PersonBirthTime	value	<p>Contains a (partial) date of birth, or a range of dates, that should match the date of birth of the person being queried for.</p> <p>Design Comments: A date or date range. This parameter can convey an exact moment (e.g., January 1, 1960 @ 03:00:00 EST), an approximate date (e.g., January 1960), or even a range of dates (e.g., December 1, 1959 through March 31, 1960). Multiple instances within a single parameter are combined with OR logic.</p> <p>For dates use YYYYMMDD.</p> <pre><value value="20100910"/></pre> <p>Example for daterange:</p> <pre><value> <low value="20100901"/> <high value="20100910"/> </value></pre>
	semanticsText	The element contains the fixed value ‘Person.birthTime’.

Class	Component	Documentation
PersonDeceased	value	<p>Contains an indicator whether or not the person being queried for is deceased or not.</p> <p>PersonDeceased is one of the optional parameters of this query interaction. If the element is omitted, both living and deceased persons will be returned. @code should contain either ‘true’ or ‘false’.</p>
	semanticsText	The element contains the fixed value ‘PersonDeceased’.

Class	Component	Documentation
PersonName	value	Contains (part of) the name of the Person whose details are being queried. See PN Data Type specification , inclusive of details on how to specify fuzzy names (using wildcards).
	semanticsText	The element contains the fixed value ‘Person.name’.

Class	Component	Documentation
GroupId	Value	<p>Contains an identification of the administrative region responsible for the person being queried for. See Administrative Entity associated with a Patient or Person for a description.</p> <p>@root should contain either 2.16.578.1.12.4.1.1.9043 (ISO 3166-1:2007 Country codes), 2.16.578.1.12.4.1.1.3402 (SSB Kommunenummer (Fylke/Kommune)) or 2.16.578.1.12.4.1.1.3403 (SSB Bydelsnummer (Fylke/Kommune/Bydel)). @extension contains a code from the coding system identified in @root.</p>
	semanticsText	The element contains the fixed value 'Member.Group.id'.

Output Information Model

The model is defined in the form of two wrappers (which contain meta-data related to the information exchange; the [Transmission Wrapper](#) and the [ControlAct Wrapper](#) and a so-called payload model).

The payload model of the response interaction is documented on the [Generic Person Model](#) page. Additionally the **QueryMatchObservation** class SHALL occur exactly once for each matching person record. The QueryMatchObservation class can be used by the server to indicate the degree of match between the query parameters and the resulting record. The higher the degree of match, the more likely it is that the record matches the person record being searched for. It can be used by receiving applications to decide how to display (in what particular order) the matching records.

- Example: a query is done with parameters: (family name = juhanson), (gender = male), (birthdate = 1961).
 - The server finds a number of 'partially' matching records, and assigns a percentage to the degree of match:
 4. Match = 95%: Juhanson, Female, 1961
 5. Match = 90%: Johanson, Male, 1961
 6. Match = 70%: Junahson, Male, 1960

The algorithm according to which the server determines the 'degree of match' isn't specified by this implementation guide. The server has to assign a numerically higher degree of match, the more likely it thinks that a record matches the person record being searched for.

Note: if the server doesn't support the concept of a 'degree of match', one fixed numeric value should be used for all records in the response information model. This, in effect, states that all returned records share one and the same 'degree of match'.

Class	Component	Documentation
QueryMatchObservation	code	Indicates the kind of observation being done. @codeSystem contains the fixed value 2.16.578.1.34.5.2, @code contains the fixed value "PERC" (percentage).
	value	Contains the 'degree of match' expressed as a percentage (a Real number). Note: in XML instances an xsi:type="REAL" type override has to be used for this attribute.

```
<subjectOf1>
<queryMatchObservation moodCode="EVN" classCode="OBS">
  <code code="PERC" codeSystem="2.16.578.1.34.5.2"/>
  <value xsi:type="REAL" value="80"/>
</queryMatchObservation>
</subjectOf1>
```

3.2.3.4. Error Handling

The response may contain an indication of errors during the processing of the request. This operation only responds with an error if the query can't be processed. If a query leads to zero matching results this is not considered to be an error. See [Error Handling](#) for a general discussion of how errors can be identified in the response.

The error codes are taken from the PersonRegistryErrors coding system (managed/maintained by NHN), with OID 2.16.578.1.12.4.5.2.1.1.

In case of processing errors the value of Acknowledgement.typeCode will be set to "AE", and the value of the DetectedIssueEvent.code attribute will be set to one of the codes shown below.

Lvl	Code	Description	Documentation
1	INVALPID	Invalid person identifier	(One of) the person identifier(s) is invalid (null, empty, or not a valid F-, D-, or FH-number [a checksum failure])
2	PARAMERR	Generic Query Parameter	One of the query parameters doesn't conform to the specified data type (e.g. illegal characters in a timestamp), or the code specified is not contained in the required coding system (e.g. "Z" as an administrative gender).
1	SERVERROR	Generic error	Uncaught exception, either from code or database - will only occur if there is a bug in the link/unlink code.

3.2.3.5. XML Examples

Input Example

```
<PRPA_IN101305NO01 ITSVersion="XML_1.0" xmlns="urn:hl7-org:v3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:hl7-org:v3 ../schemas/PRPA_IN101305NO01.xsd">
<id extension="90204193108_33" root="2.16.578.1.34.1.805.1"/>
<creationTime value="20090204193108"/>
<versionCode code="NE2010NO"/>
<interactionId extension="PRPA_IN101305NO01" root="2.16.840.1.113883.1.6"/>
<processingCode code="P"/>
<processingModeCode code="T"/>
<acceptAckCode code="NE"/>
<receiver typeCode="RCV">
  <device classCode="DEV" determinerCode="INSTANCE">
    <id extension="922" root="2.16.578.1.34.1"/>
  </device>
</receiver>
<sender typeCode="SND">
  <device classCode="DEV" determinerCode="INSTANCE">
    <id extension="805" root="2.16.578.1.34.1"/>
  </device>
</sender>
<controlActProcess classCode="CACT" moodCode="EVN">
  <authorOrPerformer typeCode="AUT">
    <assignedPerson classCode="ASSIGNED">
      <id extension="987654" root="2.16.578.1.34.3.1"/>
    </assignedPerson>
  </authorOrPerformer>
  <queryByParameter>
    <queryId extension="90204193108_33" root="2.16.578.1.34.1.805.1"/>
    <statusCode code="new"/>
    <parameterList>
      <personAdministrativeGender>
        <value codeSystem="2.16.578.1.12.4.1.1.3101" code="1"/>
        <semanticsText>LivingSubject.administrativeGender</semanticsText>
      </personAdministrativeGender>
      <personBirthTime>
        <value value="19650715"/>
        <semanticsText>LivingSubject.birthTime</semanticsText>
      </personBirthTime>
      <personDeceased>
        <value value="false"/>
        <semanticsText>PersonDeceased</semanticsText>
      </personDeceased>
    </parameterList>
  </queryByParameter>
</controlActProcess>

```

```
</personDeceased>
</parameterList>
</queryByParameter>
</controlActProcess>
</PRPA_IN101305NO01>
```

Output Example

```
<?xml version="1.0" encoding="UTF-8"?>
<PRPA_IN101306NO01 ITSVersion="XML_1.0" xmlns="urn:hl7-org:v3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v3
./schemas/PRPA_IN101306NO.xsd">
  <id extension="9382399564" root="2.16.578.1.34.1.922.3"/>
  <creationTime value="20080622193112"/>
  <versionCode code="NE2010NO"/>
  <interactionId extension="PRPA_IN101306NO01" root="2.16.840.1.113883.1.6"/>
  <processingCode code="P"/>
  <processingModeCode code="T"/>
  <acceptAckCode code="NE"/>
  <receiver typeCode="RCV">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="805" root="2.16.578.1.34.1"/>
    </device>
  </receiver>
  <sender typeCode="SND">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="922" root="2.16.578.1.34.1"/>
    </device>
  </sender>
  <acknowledgement typeCode="AA"/>
    <targetMessage>
      <id extension="80622193108_11" root="2.16.578.1.34.1.805.1"/>
    </targetMessage>
  </acknowledgement>
  <controlActProcess classCode="CACT" moodCode="EVN">
    <authorOrPerformer typeCode="AUT">
      <assignedDevice classCode="ASSIGNED">
        <id extension="922" root="2.16.578.1.34.1"/>
      </assignedDevice>
    </authorOrPerformer>
    <subject typeCode="SUBJ">
      <registrationEvent classCode="REG" moodCode="EVN">
        <id nullFlavor="UNK"/>
        <statusCode code="active"/>
        <subject1 typeCode="SBJ">
          <identifiedPerson classCode="IDENT">
            <id root="2.16.578.1.12.4.1.4.1" extension="15076500565" assigningAuthorityName="F-
nummer"/>
```

```
<statusCode code="active"/>
<identifiedPerson>
  <name>
    <given>Roland</given>
    <family qualifier="MID">Mellomnavn</family>
    <family>Gundersen</family>
  </name>
  <administrativeGenderCode code="1" codeSystem="2.16.578.1.12.4.1.1.3101"/>
  <birthTime value="19650715"/>
  <addr use="H">
    <streetAddressLine>Flåklypa 31</streetAddressLine>
    <postalCode>2560</postalCode>
    <city>Alvdal</city>
  </addr>
  <asMember classCode="MBR">
    <group classCode="PUB">
      <id extension="0438" root="2.16.578.1.12.4.1.1.3402"
assigningAuthorityName="Kommunenummer"/>
      <code code="ADMENT" codeSystem="2.16.578.1.34.5.1"/>
    </group>
  </asMember>
</identifiedPerson>
<subjectOf1>
  <queryMatchObservation moodCode="EVN" classCode="OBS">
    <code code="PERC" codeSystem="2.16.578.1.34.5.2"/>
    <value xsi:type="REAL" value="80"/>
  </queryMatchObservation>
</subjectOf1>
</identifiedPerson>
</subject1>
<custodian typeCode="CST">
  <assignedEntity classCode="ASSIGNED">
    <id root="2.16.578.1.12.4.1.4.101" extension="983658725"/>
  </assignedEntity>
</custodian>
</registrationEvent>
</subject>
<queryAck>
  <queryId extension="80622193108_11" root="2.16.578.1.34.1.805.1"/>
  <queryResponseCode code="OK"/>
  <resultCurrentQuantity value="1"/>
  <resultRemainingQuantity value="0"/>
</queryAck>
</controlActProcess>
</PRPA_IN101306NO01>
```

3.2.4. PersonRegistry.RecordRevised

3.2.4.1. Operation-level Profile

Operation Name	PersonRegistry.RecordRevised
Purpose Description	To update the demographical data of a single person. See below for storyboards that illustrate the purpose of this operation.
Logic Description	The operation updates the demographical data of a person, as long as it is known to the receiving system, and returns an accept acknowledgement.
Input/Output	Input: PersonRegistry.RecordRevised Notification (PRPA_IN101302NO) Output: Accept Acknowledgement (MCCI_IN0000002UV01) See below for details of the information models as well as examples.
Composition Role	Business level service
Composition Member Capabilities	None
Keywords	demographical data, person, revised
Version	4.0
Finalization Status within HL7 Norway	Final
Custodian	HL7 Norway

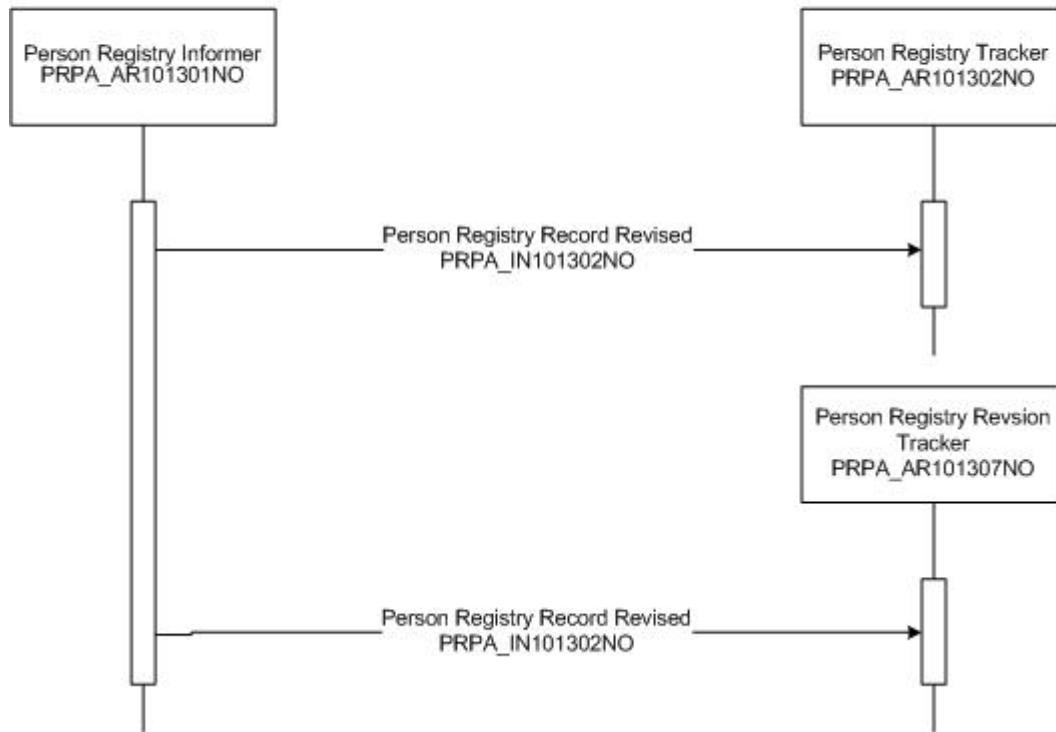
3.2.4.2. Purpose of the operation

The aim of the PersonRegistry.RecordRevised service is to update a person registry with the demographic details of a given person, based on the official identifier of the persons. In the current implementation, this identifier SHALL be the FH-number of the person.

Storyboard Diagram

This storyboard demonstrates notifying tracking systems after a record is revised in a person registry.

Person Registry Record Revised PRPA_ST101302NO



Textual Storyboard

An unconscious patient was admitted at Good Health Hospital and received an FH-number. When he regained consciousness, he informed the staff that his name was John Doe. As his demographical data was updated in the HIS, a person revised notification was sent to the national FH-number registry.

3.2.4.3. Input/Output

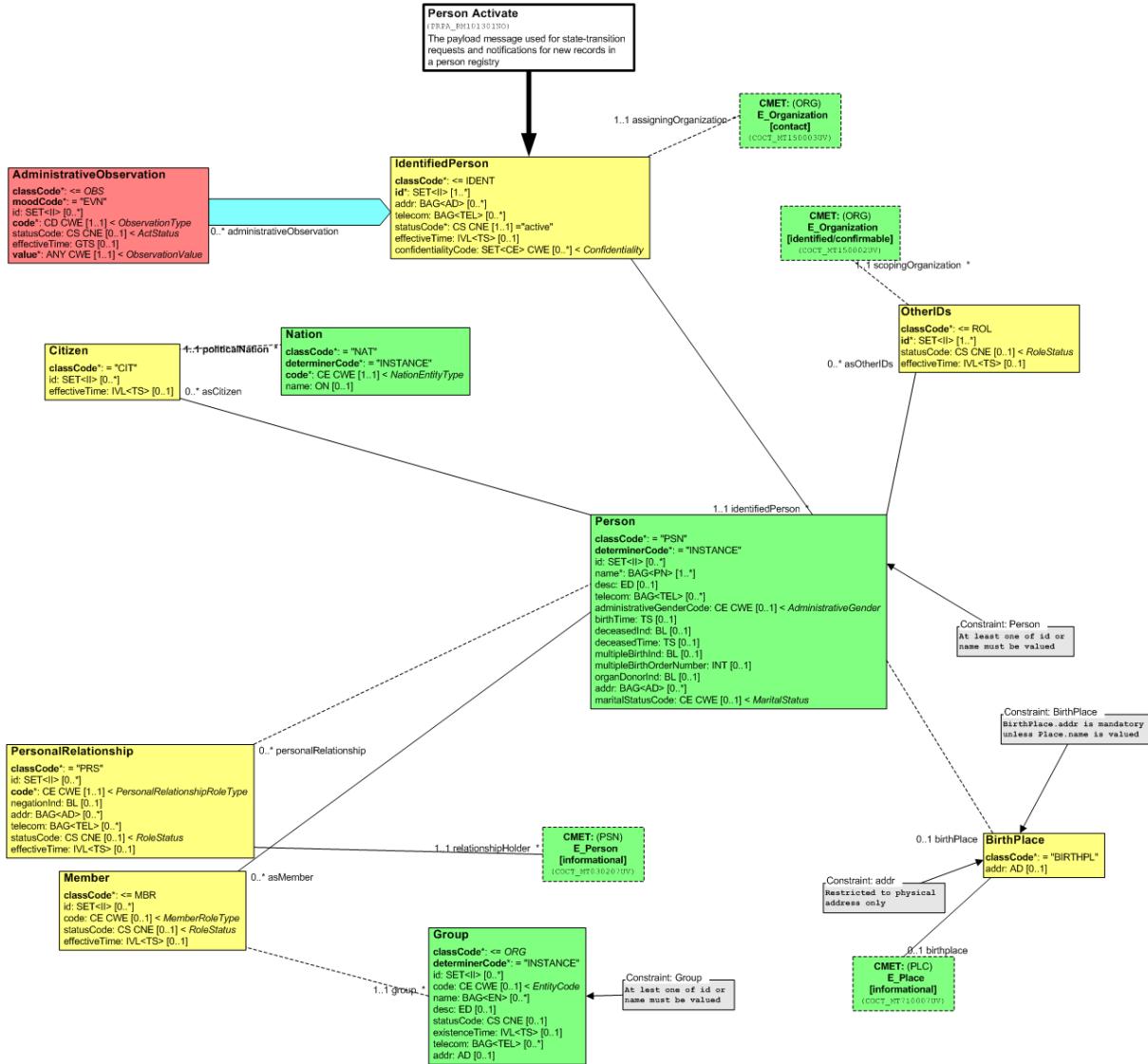
The notification interaction has an immediate response in the form of an [Accept Acknowledgement](#).

Interaction List

Person Registry Record Revised Notification	PRPA_IN101302NO
Accept acknowledgement	MCCI_IN000002UV01

Input Information Model

The model is defined in the form of two wrappers (which contain meta-data related to the information exchange; the [Transmission Wrapper](#) and the [ControlAct Wrapper](#)) and a so-called payload model. The payload model of the query interaction is shown here:



Figur 14 Source: Normative Edition 2008

See [Generic Person Model](#) for more details.

Output Information Model

The output has the form of an [Accept Acknowledgement](#).

3.2.4.4. XML Examples

Input Example

```
<?xml version="1.0" encoding="UTF-8"?>
<PRPA_IN101302NO ITSVersion="XML_1.0" xmlns="urn:hl7-org:v3" xmlns:mif="urn:hl7-org:v3/mif"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id extension="1109201010342045" root="2.16.578.1.34.1.408.7"/>
  <creationTime value="20100910103900"/>
```

```
<versionCode code="NE2008"/>
<interactionId root="2.16.840.1.113883.1.6" extension="PRPA_IN101302NO"/>
<processingCode code="P"/>
<processingModeCode code="T"/>
<acceptAckCode code="AL"/>
<receiver typeCode="RCV">
  <device determinerCode="INSTANCE" classCode="DEV">
    <id extension="1" root="2.16.578.1.34.1.999"/>
  </device>
</receiver>
<sender typeCode="RCV">
  <device determinerCode="INSTANCE" classCode="DEV">
    <id extension="2" root="2.16.578.1.34.1.999"/>
  </device>
</sender>
<controlActProcess moodCode="EVN" classCode="CACT">
  <authorOrPerformer typeCode="AUT">
    <assignedPerson classCode="ASSIGNED">
      <id extension="1234" root="2.16.578.1.34.2.9999"/>
    </assignedPerson>
  </authorOrPerformer>
  <subject typeCode="SUBJ">
    <registrationEvent moodCode="EVN" classCode="REG">
      <id nullFlavor="UNK"/>
      <statusCode code="active"/>
      <subject1 typeCode="SBJ">
        <identifiedPerson classCode="IDENT">
          <id extension="838912349234" root="2.16.578.1.12.4.1.4.3" assigningAuthorityName="FH-nummer"/>
          <statusCode code="active"/>
          <identifiedPerson>
            <name>
              <given>Ola</given>
              <given>Johan</given>
              <family>Hansen</family>
            </name>
            <administrativeGenderCode code="1" codeSystem="2.16.578.1.12.4.1.1.3101"/>
            <birthTime value="19750305"/>
            <addr use="H">
              <streetAddressLine>Parkveien 43</streetAddressLine>
              <postalCode>0258</postalCode>
              <city>Oslo</city>
            </addr>
            <asMember classCode="MBR">
              <group classCode="PUB">
```

```
<id extension="030104" root="2.16.578.1.12.4.1.1.3403"
assigningAuthorityName="Bydelsnummer"/>
<code code="ADMENT" codeSystem="2.16.578.1.34.5 .1"/>
</group>
</asMember>
</identifiedPerson>
</identifiedPerson>
</subject1>
<custodian typeCode="CST">
<assignedEntity classCode="ASSIGNED">
<id extension="2" root="2.16.578.1.34.1.999"/>
</assignedEntity>
</custodian>
</registrationEvent>
</subject>
</controlActProcess>
</PRPA_IN101302NO>
```

Output Example

See [Accept Acknowledgement](#).

3.3. CareRecordManager

3.3.1. Method: CareRecordManager.GetCareRecordProfile

This storyboard illustrates the use of this method. It involves querying a Care Record Manager application for a list of encounters that are ongoing or have taken place in the past, including the discharge diagnoses associated with those encounters.

Find Last 20 Encounters

Adam Everyman is at home and suffers a sudden onset of chest pain. The pain continues during the next half an hour. Eve Everyman, his wife, calls the emergency phone number. Nurse Nightingale at the emergency unit identifies the patient (using the patient name, birth date and gender: Patient Registry Find Multiple Candidates). Using the patient identifier (F-number 24035412356) she initiates a Get Care Record Profile Query to get hold of the last 20 (inpatient/outpatient) encounters of Adam Everyman, including discharge diagnoses related to those encounters. The returned information provides her with contextual information for the clinical history. She notices that Adam has had frequent encounters with the Cardiology department, associated with diagnoses codes ‘minor stroke’ and ‘high blood pressure’.

She orders an ambulance and sends a message related to a planned encounter to the Good Health Hospital emergency room application.

Nurse Theresa receives the message from the emergency room application about Adam Everyman and she initiates a Get Care Record Profile Query to get hold of the last 20 encounters of Adam everyman, including discharge diagnoses related to those encounters. She can see that Adam Everyman has been at the Cardiology department on May 4, 2008, and based on this information she requests additional data from that departmental system in preparation of the arrival of the patient.

The querying system sends a query with a person identifier to a Care Record Manager application. The query will be either based on the F-number, the D-number, or an Emergency patient identifier.

The query interaction has an immediate response. The response interaction will contain the details of all known historic and ongoing encounters. Both the querying application (if it explicitly requests so) as well as the responding system may introduce an upper limit in the number of encounters returned.

Interaction List

Get Care Record Profile Query

 QUPC_IN043100NO

Get Care Record Profile Response

 QUPC_IN043200NO

Note that the above interactions use the “NO” realm code, and not the original “UV”. The models used are equivalent to the international models. By using the NO realm the artefact identifier can be used for versioning purposes at some future point in time.

Each interaction is defined in the form of two wrappers (which contain meta-data related to the information exchange) and a so-called payload model. The payload model of the query interaction is shown here:

http://www.hl7.org/v3ballot2008MAY/html/domains/uvpc/uvpc_CareRecordQuery.htm#QUPC_RM040300UV-rmi

The payload model of the response interaction is shown here:

http://www.hl7.org/v3ballot2008MAY/html/domains/uvpc/uvpc_CareRecord.htm#REPC_RM004000UV-rmi

3.3.1.1. Structure of the query interaction

The query interaction (Get Care Record Profile Query) is defined as a Transmission Wrapper (see section 6.1) and a ControlAct wrapper (see section 6.2). The final part of the interaction is the QueryByParameter element, and child elements.

Nesting Level	Element	Attributes
0	queryByParameter	@moodCode, @classCode
	The QueryByParameter class contains the specification of the query.	
1	queryId	@extension, @root
	<p>The id contains the unique identification of this query instance. @root contains an identification of the ‘unique query instance numbering mechanism as used by this particular sending software application’, and @extension contains the identifier created according to that numbering mechanism.</p> <p>@root contains the ‘namespace’ of the identifier as contained in @extension</p> <p>Note: re-use same @extension/@root as used to identify the message instance in the Transmission Wrapper, see section 6.1.</p>	
1	statusCode	@code
	@code contains the fixed value ‘new’	
1	initialQuantity	@value
	Optionally contains a numeric value that specifies the maximum number of matches (encounters) the responding application should include in the response.	
1	initialQuantityCode	@code, @codeSystem
	Use only if the initialQuantity element is used. If initialQuantity has a value @code should contain RD, and @codeSystem should contain	

	2.16.840.1.113883.5.1112	
1	parameterList	
	The parameterList contains the list of parameters.	
2	patientId	
	The patientIdentifier forms the parameter of this query interaction.	
3	value	@root, @extension.
	Contains the patient identifier. This is a unique identification of a patient. @root contains an identification of the ‘unique patient identification mechanism’ (i.e. the OID for F-number, D-number, H-number or FH-number), and @extension contains the identifier created according to that identification mechanism.	

3.3.1.2. Structure of the response interaction

The response interaction (Get Care Record Profile Response) is defined as a Transmission Wrapper (see section 6.1) and a ControlAct wrapper (see section 6.2). The final part of the interaction is the Subject/registrationEvent element, and child elements.

Nesting Level	Element	Attributes
0	Subject	
	The subject of this revision interaction is an updated registration of a patient.	
1	registrationEvent	@classCode, @moodCode
	The registration activity of this set of demographics data. Note that in this project this is used in a minimalistic fashion. @classCode contains the fixed value ‘REG’, @moodCode contains the fixed value ‘EVN’	
2	Id	@nullFlavor
	Registration activities aren’t identified when there is only one Patient Registry. @nullFlavor contains the fixed value ‘UNK’.	
2	statusCode	@code
	@statusCode contains the fixed value ‘active’	
2	Custodian	@typeCode
	Identifies the custodian of the registration, the organization that maintains the registry. @typeCode contains the fixed value ‘CST’.	
3	assignedEntity	@classCode
	@classCode contains the fixed value ‘ASSIGNED’.	
4	Id	@extension. @root
	@root contains the OID of the Norwegian organization numbers: 2.16.578.1.12.4.1.4.101. @extension contains the number of the organization.	
2	subject2	@typeCode
	@typeCode contains the fixed value SUBJ.	
3	careProvisionEvent	@classCode, @moodCode
	<i>This is the root element of the care provision model. The model itself is described elsewhere in this document.</i>	

Nesting Level	Element	Attributes
---------------	---------	------------

0	careProvisionEvent	@classCode, @moodCode
	Contains information related to an encounter. @classCode contains the fixed value ENC. @moodCode contains the fixed value EVN.	
1	Id	@root, @extension
	Contains the unique identifier of the encounter. @extension contains the identifier itself, and @root contains the OID which identifies the identification mechanism used to create the identifier.	
1	Code	@code, @codeSystem
	Identifies the kind of encounter (e.g. ambulatory, inpatient). @codeSystem contains the OID associated with the coding system listed in section 5.2.6. @code contains a code from that coding system.	
1	statusCode	@code
	Contains the status of the encounter. For historic encounters, the value of @code is fixed to 'completed'.	
1	effectiveTime	
	Identifies the time (frame) during which the encounter took place.	
2	Low	@value
	For (inpatient) encounters longer than one day, and for those encounters where admit and discharge time are of importance, the low and high elements must both be populated. @value contains the start date of the encounter in YYYYMMDDHHMMSS format. The time part of the value is optional. <i>Low may not be combined with its sibling element center.</i>	
2	High	@value
	For (inpatient) encounters longer than one day, and for those encounters where admit and discharge time are of importance, the low and high elements must both be populated. @value contains the end date of the encounter in YYYYMMDDHHMMSS format. The time part of the value is optional. <i>High may not be combined with its sibling element center.</i>	
2	Center	@value
	For (ambulatory) encounters that occurred on one particular day, and where the admit/discharge time are unknown: @value contains the date the encounter took place in YYYYMMDD format. <i>If admit/discharge time is known, elements low and high must be used instead.</i> <i>Center may not be combined with either its sibling elements low or high.</i>	
1	Subject	@typeCode
	Identifies the subject of the encounter. @typeCode contains the fixed value SBJ.	
2	Patient	@classCode
	The subject is played by a patient role. @classCode contains the fixed value	

	PAT.	
3	Id	@root, @extension
	<p>Contains exactly one Patient.id (either the F-number, the D-number, the FH-number or the H-number –in that order of preference).</p> <p>@root contains the OID of the identification scheme, @extension contains the identification number according to that identification scheme.</p>	
	statusCode	@code
3	Contains the fixed value ‘active’.	
3	patientPerson	@determinerCode, @classCode
	Identifies the person playing the role of patient. @determinerCode contains the fixed value INSTANCE. @classCode contains the fixed value PSN.	
4	Name	@use
	<p>Occurs one or more times. Contains the name(s) of the person.</p> <p>See section 5.1.1 for a description of the sub elements and the usage of the sub elements.</p>	
1	Performer	@typeCode
	Identifies the performer of the encounter activity in the form of a department. @typeCode contains the fixed value PRF.	
2	responsibleParty	@classCode
	@classCode contains the fixed value ‘ASSIGNED’	
3	Code	@code, @codeSystem
	<p>Identifies the type of performer, i.e. the type of department.</p> <p>@code contains a code identifying the department type,</p> <p>@codeSystem contains the OID which identifies the coding scheme.</p>	
3	agentOrganization	@classCode, @determinerCode
	<p>Identifies the department.</p> <p>@classCode contains the fixed value ORG.</p> <p>@determinerCode contains the fixed value INSTANCE.</p>	
4	Id	@root, @extension
	<p>Contains the unique identifier of the department/organization according to the RESH Registry (Register for Enheter i SpesialistHelsetjenesten).</p> <p>@root contains the fixed value 2.16.578.1.12.4.1.4.102,</p> <p>@extension contains the identifier from the RESH registry.</p>	
4	Name	
	The element contains an unstructured name of the department identified by the id element (the RESH identifier).	

1	pertinentInformation3	@typeCode, @contextConductionInd
	Contains clinical information that is of some relevance to the encounter. @typeCode contains the fixed value PERT. @contextConductionInd contains the fixed value ‘true’.	
2	Observation	@classCode, @moodCode
	Contains the discharge diagnoses (primary and –optionally- secondary).	
3	Code	@code, @codeSystem
	Identifies that this is an observation of type ‘discharge diagnoses’. @codeSystem contains the fixed value 2.16.840.1.113883.5.4. @code contains the fixed value DISDX.	
3	Value	@code, @codeSystem, @xsi:type @displayName
	Contains the primary discharge diagnoses, coded using the Norwegian version of ICD-10. @xsi:type contains the fixed value ‘CE’. @codeSystem contains the fixed value 2.16.578.1.12.4.1.1.7110. @code contains the ICD-10 code. @displayName contains a human readable description of the diagnosis code. The description should be taken from the Norwegian version of ICD-10.	
3	targetOf	@typeCode
	Contains clinical information that is of some relevance to the primary discharge diagnoses, i.e. this part identifies the secondary diagnoses. @typeCode contains the fixed value PERT. <i>Note that this element and child elements are optional if there is no secondary diagnosis.</i>	
4	Observation	@classCode, @moodCode
	Contains the secondary discharge diagnosis.	
5	Code	@code, @codeSystem
	Identifies that this is an observation of type ‘discharge diagnoses’. @codeSystem contains the fixed value 2.16.840.1.113883.5.4. @code contains the fixed value DISDX.	
5	Value	@code, @codeSystem, @xsi:type @displayName
	Contains the primary discharge diagnoses, coded using the Norwegian version of ICD-10. @xsi:type contains the fixed value ‘CE’. @codeSystem contains the fixed value 2.16.578.1.12.4.1.1.7110. @code contains the ICD-10 code. @displayName contains a human readable description of the diagnosis code. The description should be taken from the Norwegian version of ICD-10.	

3.4. EncounterManager

3.4.1. *EncounterManager.FindEncounters*

This is a new use-case not (yet) covered by the international HL7 version 3 standard.

3.4.1.1. Operation-level Profile

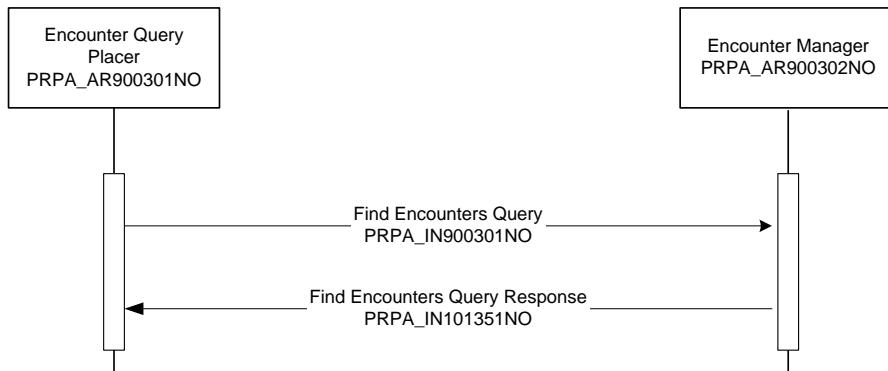
Operation Name	EncounterManager.FindEncounters
Purpose Description	To query, using a set of search criteria, an Encounter Manager for active and completed encounters. See below for storyboards that illustrate the purpose of this operation.
Logic Description	The operation will match the supplied search criteria and return between 0 (in case there wasn't a match) and 250 matching records (encounters).
Input/Output	Input: EncounterManager.FindEncounters Query (PRPA_IN900301NO) Output: EncounterManager.FindEncounters Query Response (PRPA_IN900351NO) See below for details of the information models as well as examples.
Composition Role	Business level service
Composition Member Capabilities	None
Keywords	Encounter, Search criteria, patient, outpatient, inpatient
Version	4.0
Finalization Status within HL7 Norway	Draft
Custodian	HL7 Norway

3.4.1.2. Purpose of the operation

The storyboard demonstrates querying an Encounter Manager (an application which contains information about encounters) for both active and completed encounters. This use case is also known as “who is here”, or “who is sitting in the waiting room”.

Storyboard Diagram

Find Encounters Query PRPA_ST900301NO



Textual Storyboard

- Who is sitting in the waiting room

Patient Eva Everywoman enters the waiting room of the Ear, Eye and Mouth Unit (EAM) and reports to the desk of the Simone Support, the secretary of the EAM Unit. Simone registers Eva as being “present”, and requests Eva to take a seat. Dr. Mike Molar, a dental surgeon who works for the EAM unit, opens a module in his software application and requests to see a list of all those persons that are available in the waiting room of the EAM Unit and waiting for the start of an outpatient (ambulatory) encounter (Find Encounters Query, PRPA_IN900301NO, and Find Encounters Query Response, PRPA_IN101351NO). Dr. Molar sees that Eva is present and has been waiting the longest time. He goes to the waiting room and asks Eva to join him in his office for a consultation.

- Who is here

Dr. Eric Eye, an eye specialist who works for the EAM unit, opens a module in his software application and requests to see a list of all those persons that have been admitted for an inpatient stay -and that are still present in the hospital (Find Encounters Query, PRPA_IN900301NO, and Find Encounters Query Response, PRPA_IN101351NO). Dr. Eye sees that one of the patients he’s been consulted for is still in the hospital and decides to call his colleague to see if further consultation will be necessary.

- Where is the patient

Simon Support, a secretary at the clinical biochemistry laboratory, requests the sample list for the 8 o’ clock round at the children’s clinic. The sample list software application queries the encounter manager for the location, down to bed-level) of each patient on the list (Find Encounters Query, PRPA_IN900301NO, and Find Encounters Query Response, PRPA_IN900351NO). The complete sample list is printed out and given to the staff responsible for the sample collection round.

- Is patient admitted

Clerk Kent, a secretary at the radiology department has a list of inpatients scheduled for examination during the day. As the appointment for the patient Adam Everyman is getting close, Kent opens a module in his software application and request to see at which ward

Everyman is admitted. (Find Encounters Query, PRPA_IN900301NO, and Find Encounters Query Response, PRPA_IN900351NO). Kent then makes a phone call to the ward and asks them to have the patient delivered to the examination room in 30 minutes.

- Inpatient or outpatient

Bill Costly, the secretary responsible for preparing the invoices at the radiation therapy department, has a list of all patients treated the last day. For each patient on the list, Costly opens a module in his software application and request to see if the patient was an inpatient or outpatient at the time of the treatment. (Find Encounters Query, PRPA_IN900301NO, and Find Encounters Query Response, PRPA_IN900351NO). If the patient was an outpatient, Costly prepares an invoice.

3.4.1.3. Input/Output

EncounterManager.FindEncounters Query	PRPA_IN900301NO
EncounterManager.FindEncounters Query Response	PRPA_IN900351NO

Note that the above interactions use the “NO” realm code. The models used are (currently) specific for HL7 Norway and will be brought forward for inclusion in the international standard.

Input Information Model

The query interaction (Find Encounters Query) is defined as a [Transmission Wrapper](#) and a [ControlAct Wrapper](#). The final part of the interaction is the QueryByParameter element, and child elements.

Model of QueryByParameter (Find Encounters Query):

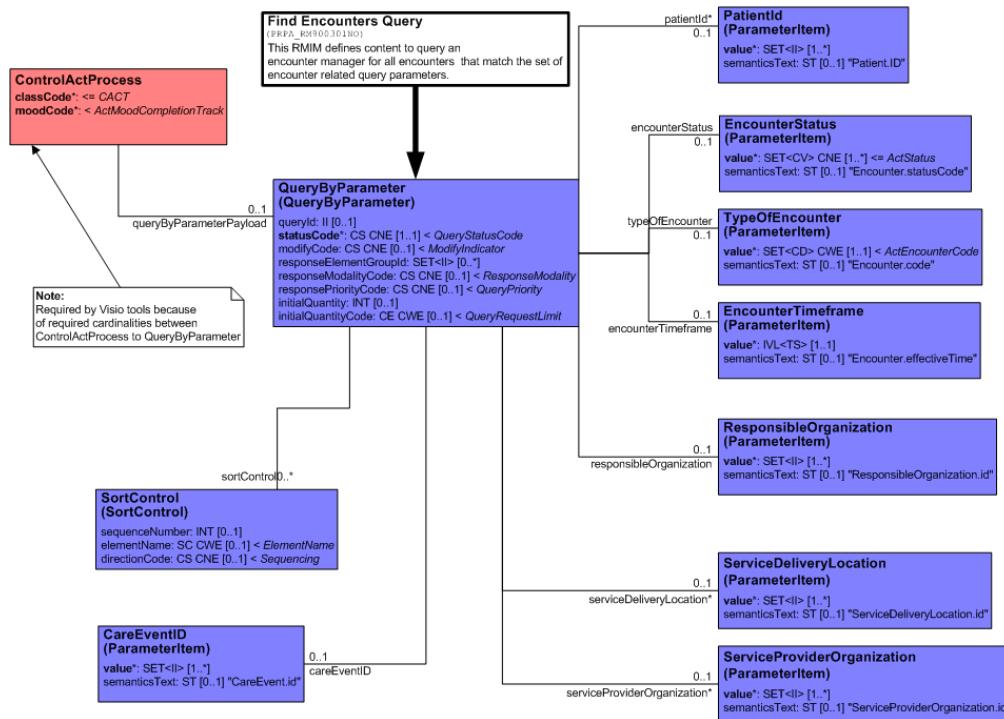


Figure 15 Model of QueryByParameter

Output Information Model

The response interaction (Find Encounters Query Response) is defined as a [Transmission Wrapper](#) and a [ControlAct Wrapper](#). The final part of the interaction is the Subject element, and child elements.

Model of EncounterEvent (FindEncounters Query Response):

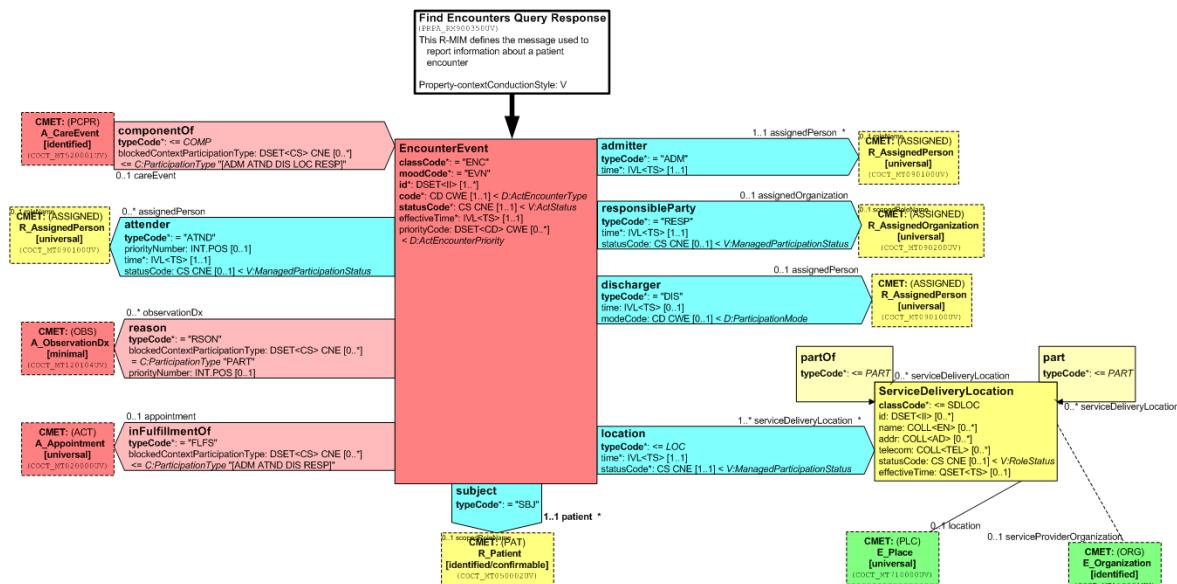


Figure 16 Model of EncounterEvent

3.4.1.4. XML Examples

Input Example

```
<?xml version="1.0" encoding="UTF-8"?>
<PRPA_IN900301NO ITSType="XML_1.0" xmlns="urn:hl7-org:v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:hl7-org:v3 ..schemas/PRPA_IN900301NO.xsd">

  <id extension="090305172501_917" root="2.16.578.1.34.1.805.1"/>

  <creationTime value="20090305172501"/>
  <versionCode code="NE2008"/>

  <interactionId extension="PRPA_IN900301NO" root="2.16.840.1.113883.1.6"/>
  <processingCode code="P"/>
  <processingModeCode code="T"/>
  <acceptAckCode code="NE"/>
  <receiver typeCode="RCV">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="922" root="2.16.578.1.34.1"/>
    </device>
  </receiver>
  <sender typeCode="SND">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="805" root="2.16.578.1.34.1"/>
    </device>
  </sender>
  <controlActProcess classCode="CACT" moodCode="EVN">
    <authorOrPerformer typeCode="AUT">
      <assignedPerson classCode="ASSIGNED">
        <id extension="987654" root="2.16.578.1.34.3.1"/>
      </assignedPerson>
    </authorOrPerformer>
    <queryByParameter>

      <queryId extension="090305171908_917" root="2.16.578.1.34.1.805.1"/>
      <statusCode code="new"/>
      <encounterTimeframe>
        <value>
          <center value="20080625120000"/>
        </value>
      </encounterTimeframe>
    </queryByParameter>
  </controlActProcess>
</PRPA_IN900301NO>
```

```

        </value>
    </encounterTimeframe>

    <patientId>
        <value root="2.16.578.1.12.4.1.4.1" extension="15076500565"
              assigningAuthorityName="F-nummer"/>
    </patientId>
    <typeOfEncounter>

        <value codeSystem="2.16.840.1.113883.5.4" code="IMP"/>
    </typeOfEncounter>

</queryByParameter>
</controlActProcess>
</PRPA_IN900301NO>

```

Output Example

```

<?xml version="1.0" encoding="UTF-8"?>
<PRPA_IN900301NO_OperationResponse>
<PRPA_IN900301NO-Response>
    <PRPA_IN900351NO ITSVersion="XML_1.0">
        <id root="2.16.578.1.34.922.3" extension="8f8ba502-7cef-4b33-b71c-e503fe03f14b"/>
        <creationTime value="20110331141502"/>
        <versionCode code="NE2008"/>
        <interactionId root="2.16.840.1.113883.1.6" extension="PRPA_IN900351NO"/>
        <processingCode code="T"/>
        <processingModeCode code="T"/>
        <acceptAckCode code="NE"/>
        <receiver typeCode="RCV">
            <device determinerCode="INSTANCE" classCode="DEV">
                <id root="2.16.578.1.34.1" extension="805"/>
            </device>
        </receiver>
        <sender typeCode="SND">
            <device determinerCode="INSTANCE" classCode="DEV">
                <id root="2.16.578.1.34.1" extension="2.16.578.1.34.20.1"/>
            </device>
        </sender>
        <acknowledgement>
            <typeCode code="AA"/>
        <targetMessage>

```

```
<id root="2.16.578.1.34.1.805.1" extension="090305172501_917"/>
</targetMessage>
</acknowledgement>
<controlActProcess moodCode="EVN" classCode="CACT">
  <authorOrPerformer typeCode="AUT">
    <assignedDevice classCode="ASSIGNED">
      <id root="2.16.578.1.34.1" extension="2.16.578.1.34.20.1"/>
    </assignedDevice>
  </authorOrPerformer>
  <subject typeCode="SUBJ">
    <encounterEvent moodCode="EVN" classCode="ENC">
      <id root="2.16.578.1.34.111.1.1" extension="1"/>
      <code code="IMP" codeSystem="2.16.840.1.113883.5.4"/>
      <statusCode code="completed"/>
      <effectiveTime>
        <low value="20080621000000"/>
        <high value="20080701000000"/>
      </effectiveTime>
      <subject typeCode="SBJ">
        <patient classCode="PAT">
          <id root="2.16.578.1.12.4.1.4.1" extension="15076500565" assigningAuthorityName="F-
nummer"/>
          <patientPerson determinerCode="INSTANCE" classCode="PSN">
            <name xsi:type="PN" use="L">
              <given>Roland</given>
              <family>Gundersen</family>
            </name>
          </patientPerson>
        </patient>
      </subject>
      <location typeCode="LOC">
        <serviceDeliveryLocation classCode="SDLOC">
          <id root="2.16.578.1.34.2.18" extension="1"/>
          <code code="BED" codeSystem="2.16.840.1.113883.5.1060"/>
          <name>Fødeposten, Rom1, S1</name>
          <statusCode code="active"/>
          <serviceProviderOrganization determinerCode="INSTANCE" classCode="ORG">
            <id root="2.16.578.1.34.1000.4" extension="100021"/>
          </serviceProviderOrganization>
        <partOf typeCode="PART">
          <serviceDeliveryLocation classCode="SDLOC">
            <id root="2.16.578.1.34.2.20.2" extension="24"/>
            <code code="WARD" codeSystem="2.16.840.1.113883.5.1060"/>
            <name>Fødeposten</name>
            <telecom value="tel:75592000"/>
          </serviceDeliveryLocation>
        </partOf>
      </location>
    </encounterEvent>
  </subject>
</controlActProcess>
```

```

<statusCode code="active"/>
<serviceProviderOrganization determinerCode="INSTANCE" classCode="ORG">
  <id root="2.16.578.1.34.1000.4" extension="100021"/>
</serviceProviderOrganization>
</serviceDeliveryLocation>
</partOf>
</serviceDeliveryLocation>
</location>
<inFulfillmentOf typeCode="FLFS">
  <appointment moodCode="APT" classCode="ENC">
    <id root="2.16.578.1.34.8240.1" extension="123456789"/>
  </appointment>
</inFulfillmentOf>
</encounterEvent>
</subject>
<queryAck>
  <queryId root="2.16.578.1.34.1.805.1" extension="090305172501_917"/>
  <queryResponseCode code="OK"/>
  <resultCurrentQuantity value="1"/>
  <resultRemainingQuantity value="0"/>
</queryAck>
</controlActProcess>
</PRPA_IN900351NO>
</PRPA_IN900301NO-Response>
</PRPA_IN900301NO_OperationResponse>

```

Remarks

If using Normative Edition 2008 schemas (NE2008), there is a bug there stating that "ContactParty" is mandatory. E.g. it has to be present, but with a nullFlavor.

3.4.2. *EncounterManager.FindScheduledEncounters*

This is a new use-case not (yet) covered by the international HL7 version 3 standard.

Note: Description of interactions and message content is known to be incomplete (notably in the area of the identification of the equipment). A new version of this implementation guide will contain a finalized description.

3.4.2.1. Operation-level Profile

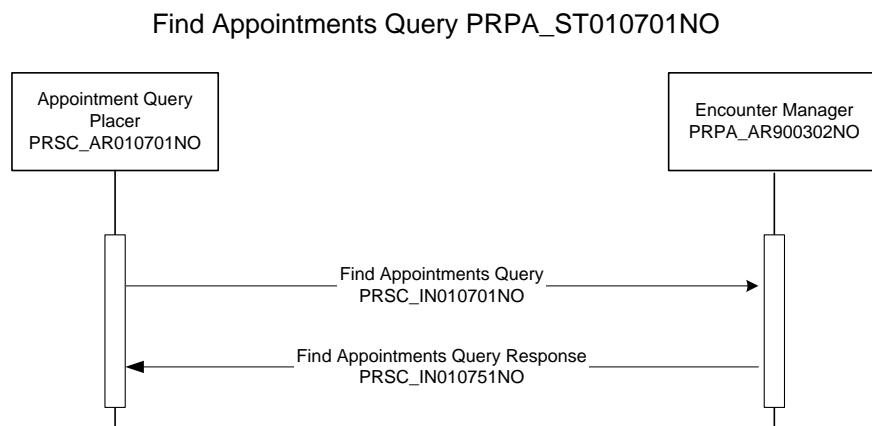
Operation Name	EncounterManager.FindScheduledEncounters
Purpose Description	To query, using a set of search criteria, a Schedule Manager for

	<p>planned encounters.</p> <p>See below for storyboards that illustrate the purpose of this operation.</p>
Logic Description	The operation will match the supplied search criteria and return all known scheduled encounters that match the search criteria as listed in the query interaction.
Input/Output	<p>Input: Find Appointments Query (PRSC_IN010701NO)</p> <p>Output: Find Appointments Query Response (PRSC_IN010751NO) See below for details of the information models as well as examples.</p>
Composition Role	Business level service
Composition Member Capabilities	None
Keywords	Encounter, Appointment, Schedule Manager, Search criteria, patient, outpatient, inpatient
Version	4.0
Finalization Status within HL7 Norway	Final
Custodian	HL7 Norway

3.4.2.2. Purpose of the operation

The storyboard demonstrates querying Schedule Manager (an application which contains information about scheduled encounters) for planned encounters.

Storyboard Diagram



Textual Storyboard

- What imaging equipment has been scheduled for the patient

Dr. Rudolf Röntgen, a Radiologist, has been informed that his patient, Eve Everywoman, has arrived in the waiting room of the Radiology department. Dr. Röntgen opens a module in his software application and requests to see what imaging equipment has been scheduled for Eve. (Find Appointments Query, PRSC_IN010701NO, and Find Appointments Query Response, PRSC_IN010751NO). He sees that her examination is scheduled to take place using imaging equipment known as ‘CT number 1’.

- Find all patients scheduled for an examination today, involving a given equipment

Dr. Rudolf Röntgen, a Radiologist, opens a module in his software application and requests to see a list of all patients that are scheduled to have an examination (today) involving the imaging equipment known as ‘CT number 1’. (Find Appointments Query, PRSC_IN010701NO, and Find Appointments Query Response, PRSC_IN010751NO). Dr. Röntgen sees that the next appointment is in 10 minutes, for patient Eve Everywoman..

3.4.2.3. Input/Output

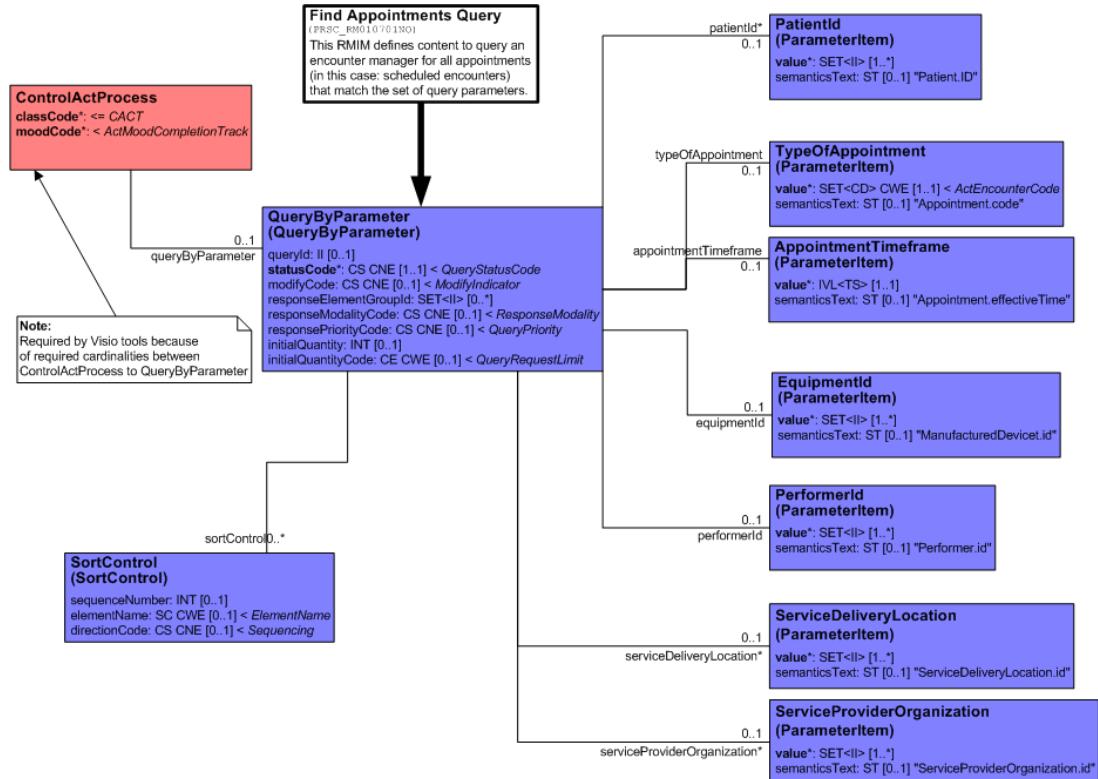
EncounterManager.FindScheduledEncounters Query (Find Appointments Query)	PRSC_IN010701NO
EncounterManager.FindScheduledEncounters Query Response (Find Appointments Query Response)	PRSC_IN010751NO

Note that the above interactions use the “NO” realm code. The models used are (currently) specific for HL7 Norway and will be brought forward for inclusion in the international standard.

Input Information Model

The query interaction (Find Appointments Query) is defined as a [Transmission Wrapper](#) and a [ControlAct Wrapper](#). The final part of the interaction is the QueryByParameter element, and child elements.

Model of QueryByParameter (Find Appointments Query):

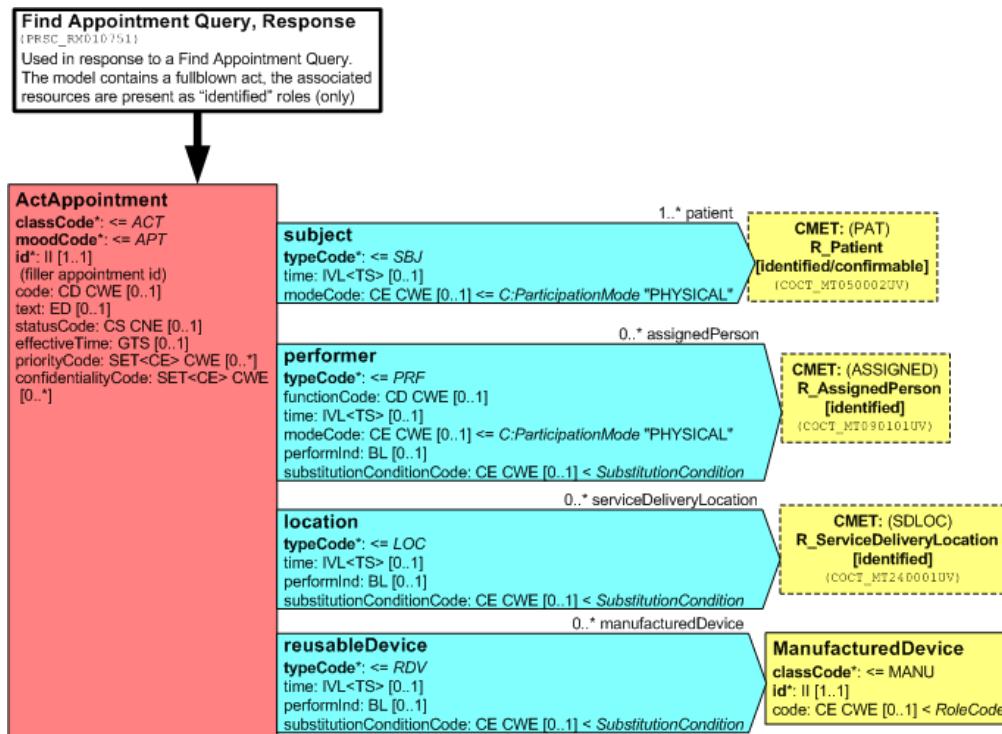


Figur 17 Model of QueryByParameter

Output Information Model

The response interaction (Find Appointments Query Response) is defined as a [Transmission Wrapper](#) and a [ControlAct Wrapper](#). The final part of the interaction is the Appointment element, and child elements.

Model of Appointment element (Find Appointments Query Response):



Figur 18 Model of Appointment element

3.4.2.4. XML Examples

Input Example

```
<?xml version="1.0" encoding="UTF-8"?>
<PRSC_IN010701NO ITSVersion="XML_1.0" xmlns="urn:hl7-org:v3">
  <id root="MessageConversationApplicationOwner" extension="ConversationId"/>
  <sender typeCode="SND">
    <device determinerCode="INSTANCE" classCode="DEV">
      <id root="rootValue" extension="SenderSystemId"/>
    </device>
  </sender>
  <controlActProcess moodCode="EVN" classCode="CACT">
    <queryByParameter>
      <appointmentTimeframe>
        <value>
          <low value="20110201000000"/>
          <high value="20111201235959"/>
        </value>
      </appointmentTimeframe>
      <performerId>
        <value root="2.16.578.1.34.1000.3"/>
      </performerId>
      <serviceDeliveryLocation>
        <value extension="10000"/>
      </serviceDeliveryLocation>
      <typeOfAppointment>
        <value code="IMP" codeSystem=""/>
      </typeOfAppointment>
    </queryByParameter>
  </controlActProcess>
</PRSC_IN010701NO>
```

```
</typeOfAppointment>
</queryByParameter>
</controlActProcess>
</PRSC_IN010701NO>
```

Output Example

```
<?xml version="1.0" encoding="UTF-8"?>
<PRSC_IN010701NO-Response xmlns="urn:hl7-org:v3">
<PRSC_IN010751NO ITSVersion="XML_1.0">
<id root="2.16.578.1.34.922.3" extension="e68788b6-a352-4863-ae95-b6bb0389acc9"/>
<creationTime value="20110520095310"/>
<versionCode code="NE2008"/>
<interactionId root="2.16.840.1.113883.1.6" extension="PRSC_IN010751NO"/>
<processingCode code="T"/>
<processingModeCode code="T"/>
<acceptAckCode code="NE"/>
<receiver typeCode="RCV">
<device determinerCode="INSTANCE" classCode="DEV">
<id root="2.16.578.1.34.1" extension="SenderSystemId"/>
</device>
</receiver>
<sender typeCode="SND">
<device determinerCode="INSTANCE" classCode="DEV">
<id root="2.16.578.1.34.1" extension="2.16.578.1.34.20.1"/>
</device>
</sender>
<acknowledgement>
<typeCode code="AA"/>
<targetMessage>
<id root="MessageConversationApplicationOwner" extension="ConversationId"/>
</targetMessage>
</acknowledgement>
<controlActProcess moodCode="EVN" classCode="CACT">
<authorOrPerformer typeCode="AUT">
<assignedDevice classCode="ASSIGNED">
<id root="2.16.578.1.34.1" extension="2.16.578.1.34.20.1"/>
</assignedDevice>
</authorOrPerformer>
<subject typeCode="SUBJ">
<actAppointment moodCode="APT" classCode="ENC">
<id root="2.16.578.1.34.111.1.1" extension="edbe67b3-62dd-4a84-a5ba-6ba5e74a5967"/>
<code code="IMP" codeSystem="2.16.840.1.113883.5.4"/>
<effectiveTime xmlns:q1="urn:hl7-org:v3" xsi:type="q1:IVL_TS">
<q1:low value="20100101080000"/>
</effectiveTime>
<subject typeCode="SBJ">
<patient classCode="PAT">
<id root="2.16.578.1.12.4.1.4.1" extension="15076500565" assigningAuthorityName="F-
Number"/>
<patientPerson determinerCode="INSTANCE" classCode="PSN">
<name xmlns:q2="urn:hl7-org:v3" xsi:type="q2:PN" use="L">
```

```
<q2:given>Roland</q2:given>
<q2:family>Gundersen</q2:family>
</name>
</patientPerson>
</patient>
</subject>
<performer typeCode="PRF">
  <assignedPerson classCode="ASSIGNED">
    <id root="2.16.578.1.34.1000.3" extension="5023410"/>
  </assignedPerson>
</performer>
<location typeCode="LOC">
  <serviceDeliveryLocation classCode="SDLOC">
    <id root="2.16.578.1.12.4.1.4.102" extension="10000"/>
    <statusCode code="active"/>
    <serviceProviderOrganization determinerCode="INSTANCE" classCode="ORG">
      <id root="2.16.578.1.12.4.1.4.102" extension="20000"/>
    </serviceProviderOrganization>
  </serviceDeliveryLocation>
</location>
</actAppointment>
</subject>
<queryAck>
  <queryId root="MessageConversationApplicationOwner" extension="ConversationId"/>
  <queryResponseCode code="OK"/>
  <resultCurrentQuantity value="1"/>
  <resultRemainingQuantity value="0"/>
</queryAck>
</controlActProcess>
</PRSC_IN010751NO>
</PRSC_IN010701NO-Response>
```

Remarks

If using Normative Edition 2008 schemas (NE2008), there is a bug there stating that "ContactParty" is mandatory. E.g. it has to be present, but with a nullFlavor.

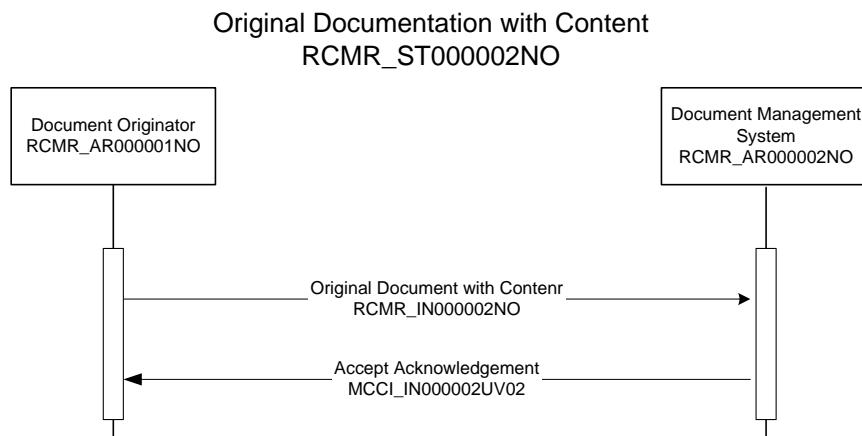
3.5. DocumentManager

The DocumentManager supports a series of services related to the management of documents in a document archive.

A document is a *persistent* (unchangeable) object. Whenever a document is transmitted (e.g. a document in a PDF format, a TIFF image, or a CDA document) one has to send document management metadata alongside the actual document object.

3.5.1. Method: DocumentManager.ProcessNewDocument

This storyboard demonstrates the sending of a new document to a document management archive.



Textual Storyboard #1: Dr. Simon Surgeon has performed an operation earlier today and decides to write a ‘Surgery Note’ document. Once he has finalized the document he decides to share it with others in his hospital by uploading it to the central document archive (*Original Document with Content*, RCMR_IN000002NO.)

The notification interaction has an immediate response in the form of an Accept Acknowledgement. Note that an Accept Acknowledgement serves both as an assurance that the interaction was received, and as a statement that the interaction was –at a glance– syntactically correct.

Interaction List

Original Document with Content

RCMR_IN000002NO

Accept Acknowledgement

[MCCI_IN000002UV01](#)

Note that one of the above interactions use the “NO” realm code, and not the original “UV”. The models used are (currently) 100% equivalent to the international models. By

using the NO realm the artefact identifier can be used for versioning purposes at some future point in time.

Each interaction is defined in the form of two wrappers (which contain meta-data related to the information exchange) and a so-called payload model. The payload model of the Document Notification interaction is shown here:

http://www.hl7.org/v3ballot2009jan/html/domains/uvmr/uvmr_DocumentManagement.htm#RCMR_RM000050UV02-rmi

The [Accept Acknowledgement](#) response interaction has no payload.

3.5.1.1. Structure of the document notification interaction

The document notification interaction (Original Document with Content) is defined as a Transmission Wrapper (see section 6.1) and a ControlAct wrapper (see section 6.2). The final part of the interaction is the subject element, and child elements.

Nesting Level	Element	Attributes
0	Subject	
	Contains the payload of the interaction.	
1	ClinicalDocument	@classCode, @moodCode
	Contains metadata of the document, e.g. the ID of the document, the document type, the document title and the document language. @classCode contains the fixed value ‘DOCCLIN’; @moodCode contains the fixed value ‘EVN’.	
2	Id	@root, @extension
	Contains a unique identifier of this document. @root identifies (using an OID) the identification mechanism itself, and @extension the identification assigned within that identification mechanism. Note: if the document that’s being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/id element.	
2	Code	@code, @codeSystem, @displayName
	This attribute identifies the document type. @code contains a code from the LOINC coding system; @codeSystem contains the fixed value ‘2.16.840.1.113883.6.1’, @displayName the name of the document type, e.g. ‘Discharge Summary’. Note: if the document that’s being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/code element.	
2	Text	@mediaType,

		@representation
	<p>The use of this element is mandatory in the DocumentManager.ProcessNewDocument (RCMR_IN000002NO) service, and disallowed in the DocumentManager.ProcessDocumentReplacement (RCMR_IN000015NO) service.</p> <p>The element contains the document. Note that non-plain-text files must be Base64 encoded, this requirement is also true for XML files.</p> <p>For CDA documents:</p> <ul style="list-style-type: none"> • The @mediaType attribute contains the fixed value ‘application/hl7-sda+xml’. The @representation attribute contains the fixed value ‘B64’. • The element itself contains the Base64 representation of the CDA document. 	
2	statusCode	@code
	<p>Contains the status of the document within the overall document management lifecycle.</p> <ul style="list-style-type: none"> • @code contains ‘active’ for new documents in the context of the DocumentManager.ProcessNewDocument (RCMR_IN000002NO) service • @code contains ‘obsolete’ for documents that have been replaced – in the context of the DocumentManager.ProcessDocumentReplacement (RCMR_IN000015NO) service. 	
2	effectiveTime	@value
	<p>@value contains the date and time the document was created, in YYYYMMDDHHMM format.</p> <p>Note: if the document that’s being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/effectiveTime element.</p>	
2	confidentialityCode	@code, @codeSystem
	<p>@code contains the fixed value ‘N’; @codeSystem contains the fixed value ‘2.16.840.1.113883.5.25’.</p> <p>Note: if the document that’s being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/confidentialityCode element.</p>	
2	recordTarget	@typeCode
	<p>Identifies the patient (and subject) of this document. @typeCode contains the fixed value ‘RCT’.</p> <p>Note: if the document that’s being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/recordTarget element.</p>	
3	Patient	@classCode
	<p>Identifies the Patient using an F- or D-number. @classCode contains the fixed value ‘PAT’.</p> <p>Note: if the document that’s being referred to is a CDA document, the value of this element can be copied from the</p>	

	ClinicalDocument/recordTarget/patientRole element.	
4	Id	@root, @extension
	<p>Contains exactly one Patient.id (the F-number, the D-number, the FH-number or the H-number –in that order of preference). @root contains the OID of the identification scheme; @extension contains the identification number according to that identification scheme.</p> <p>See section 1.2 for details.</p> <p>Note: if the document that's being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/recordTarget/PatientRole/id element.</p>	
4	statusCode	@code
	@code contains the fixed value 'active'.	
4	patientPerson	@classCode, @determinerCode
	<p>Identifies the person that plays the role of patient. @classCode contains the fixed value 'PSN'; @determinerCode contains the fixed value 'INSTANCE'.</p> <p>Note: if the document that's being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/recordTarget/PatientPerson/Patient element.</p>	
5	Name	@use
	<p>Occurs one or more times. Contains the name(s) of the person.</p> <p>See section 5.1.1 for details.</p> <p>Note: if the document that's being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/recordTarget/PatientPerson/Patient/name element.</p>	
2	Author	@typeCode
	<p>Identifies the author of the Document. @typeCode contains the fixed value 'AUT'.</p> <p>Note: if the document that's being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/author element.</p>	
3	Time	@value
	<p>The time that the document was created; @value contains the same value as the clinicalDocument/effectiveTime/@value attribute.</p> <p>Note: if the document that's being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/author/time element.</p>	
3	assignedAuthor	@classCode
	<p>@classCode contains the fixed value 'ASSIGNED'.</p> <p>Note: if the document that's being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/author/assignedAuthor element.</p>	
4	Id	@root, @extension

	Contains the HPR number of the person who created the document. Note: if the document that's being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/author/assignedAuthor/id element.	
2	Custodian	@typeCode
	Identifies the organization responsible for archiving the 'master version' of this electronic document. @typeCode contains the fixed value 'CST'. Note: if the document that's being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/custodian element.	
3	assignedCustodian	@classCode
	Identifies the role of custodian as played by an organization. @classCode contains the fixed value 'ASSIGNED'. Note: if the document that's being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/custodian/assignedCustodian element.	
4	representedOrganization	@classCode, @determinerCode
	Identifies the organization that acts as the custodian. @classCode contains the fixed value 'ORG'; @determinerCode the fixed value 'INSTANCE'. Note: if the document that's being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/custodian/assignedCustodian/representedCustodian element.	
5	Id	@root, @extension
	Identification of the custodian organization. Note: if the document that's being referred to is a CDA document, the value of this element can be copied from the ClinicalDocument/recordTarget element.	
2	componentOf	@typeCode
	Identifies the encounter this document is associated with. @typeCode contains the fixed value 'COMP'.	
3	encompassingEncounter	@moodCode, @classCode
	Contains the details of the encounter. @classCode contains the fixed value 'ENC'; @moodCode contains the fixed value 'EVN'.	
4	Id	@root, @extension
	Contains the unique identifier of the encounter. @extension contains the identifier itself, and @root contains the OID which identifies the identification mechanism used to create the identifier.	
4	Code	@code, @codeSystem
	Identifies the kind of encounter (e.g. ambulatory, inpatient). @codeSystem contains the OID associated with the coding system listed in section 5.2.6. @code contains a code from that coding system.	

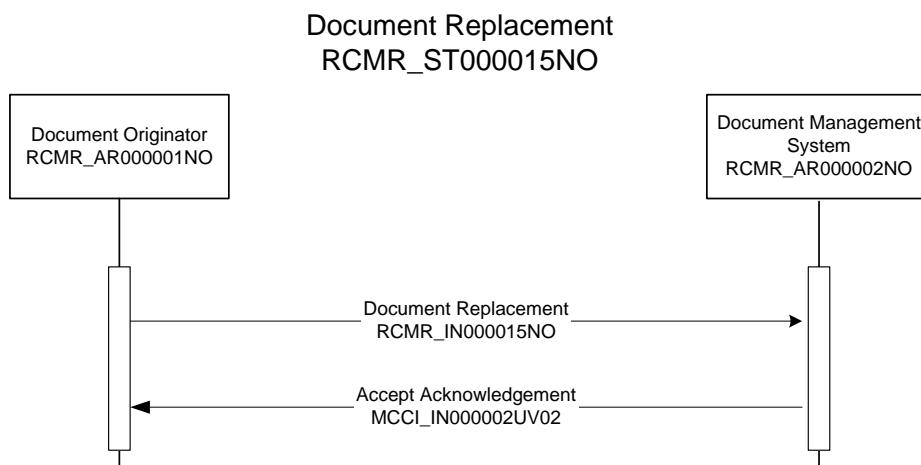
4	effectiveTime	High, low
	Contains the time(frame) during which the encounter took place / is taking place.	
4	Location	@typeCode
	Identifies the location/the responsible clinic for the encounter. @typeCode contains the fixed value ‘LOC’.	
5	healthcareFacility	@classCode
	Identifies the location of the encounter. @classCode contains the fixed value ‘SDLOC’	
6	serviceProviderOrganization	@classCode
	Identifies the responsible clinic for the encounter. @classCode contains the fixed value ‘ORG’	
7	Id	@root, @extension
	Clinic responsible for the ward/room/bed; RESH identifier. @root contains the fixed value ‘2.16.578.1.12.4.1.4.102’; @extension the RESH identifier.	

3.5.1.2. Structure of the response interaction

The response interaction (Accept Acknowledgement) is defined as a Transmission Wrapper (see section 6.1). The interaction doesn't contain a ControlAct wrapper nor a payload.

3.5.2. Method: *DocumentManager.ProcessDocumentReplacement*

This storyboard demonstrates the notification to a document management archive by the author of a document that the document has been/will be replaced by a newer version.



Textual storyboard #1: Dr. Simon Surgeon has made a ‘Surgery Note’ available to the centralized document archive. He finds out that that document needs to be corrected, so he notifies the centralized document archive (*Document Replacement*, RCMR_IN000015NO) that the document has been, or will be, replaced by a corrected version of the document.

The notification interaction has an immediate response in the form of an Accept Acknowledgement. Note that an Accept Acknowledgement serves both as an assurance that the interaction was received, and as a statement that the interaction was –at a glance– syntactically correct.

Interaction List

Document Replacement

 RCMR_IN000015NO

Accept Acknowledgement

 [MCCI_IN00002UV01](#)

Note that one of the above interactions use the “NO” realm code, and not the original “UV”. The models used are (currently) 100% equivalent to the international models. By using the NO realm the artefact identifier can be used for versioning purposes at some future point in time.

Each interaction is defined in the form of two wrappers (which contain meta-data related to the information exchange) and a so-called payload model. The payload model of the Document Notification interaction is shown here:

http://www.hl7.org/v3ballot2009jan/html/domains/uvmr/uvmr_DocumentManagement.htm#RCMR_RM000050UV02-rmi

The [Accept Acknowledgement](#) response interaction has no payload.

3.5.2.1. Structure of the document replacement interaction

The document replacement interaction (Document Replacement) is defined as a Transmission Wrapper (see section 6.1) and a ControlAct wrapper (see section 6.2).

The final part of the interaction is the subject element, and child elements. The model is the same as the model documented in section 3.5.1.1, with the additional requirement that:

- The ClinicalDocument/statusCode/@code attribute be valued with ‘obsolete’ to identify that the document has been replaced by another document.
- The ClinicalDocument/text attribute may not contain the document.

3.5.2.2. Structure of the response interaction

The response interaction (Accept Acknowledgement) is defined as a Transmission Wrapper (see section 6.1). The interaction doesn't contain a ControlAct wrapper nor a payload.

4. Model Elements

4.1. Object Identifiers (OIDs)

OIDs are an ISO mechanism for the unique identification of objects. Within the context of HL7 interactions OIDs are quite often used to uniquely identify an identification scheme (a particular methodology for the identification of a class of objects, e.g. HPR-number) or a coding system (a terminology; a table of codes; e.g. ICD-10, Sivilstand). For an introduction to OIDs, see http://www.ringholm.de/docs/00900_en.htm

Commonly used OIDs (i.e. used for a purpose that is wider in scope than the regional healthcare organization; used in communications between healthcare regions) can be found in OIDs registries such as www.oid-info.com, www.hl7.org, HL7 OID registry in the UK (<http://www.hl7.org.uk/version3group/oids.asp>) and in Norway www.volven.no.

Note: www.volven.no does only present the last part of the OID. The branch (2.16.578.1.12.4.1.1) must prefix all OIDs before they are referenced.

Note: All OIDs with an element of 999 or 9999 are for documentation purposes only. They will be replaced by permanent OIDs once these have been assigned/found/applied for.

Helse Vest is used as an example representing a regional healthcare organization throughout this document.

The Helse Vest OID is: 2.16.578.1.34. Helse Vest will use subbranches of that OID to identify objects or coding systems. Updated information: <http://www.oid-info.com/get/2.16.578.1.34>

New OIDs within the regional healthcare organizations shall be assigned by Helsedirektoratet.

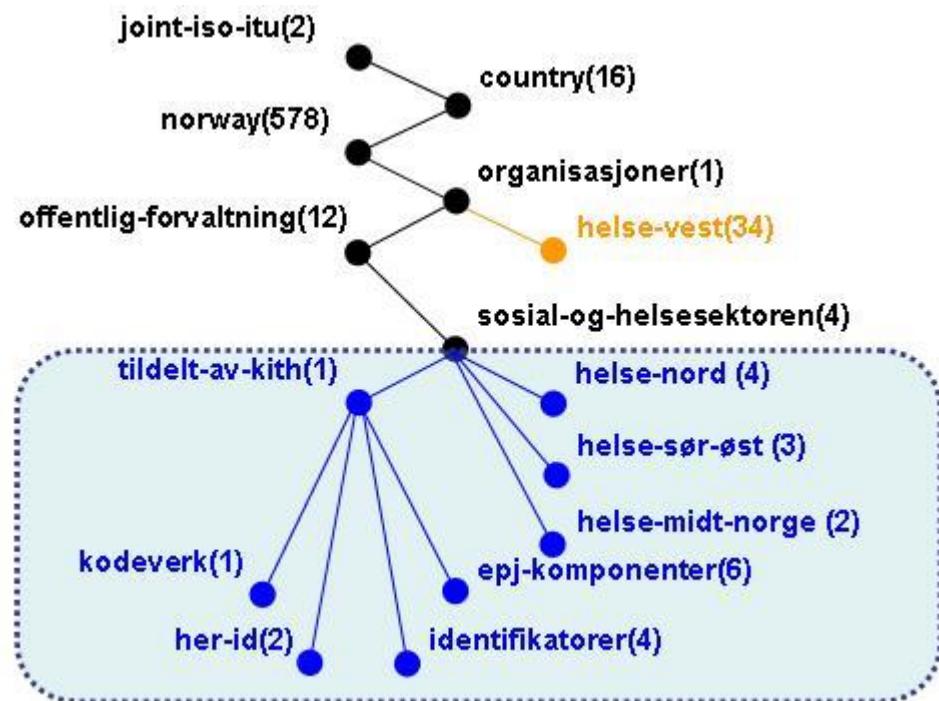
4.1.1.Coding Systems

The following are OIDs for coding systems:

OID	CodingSystemName	Description	Issuer
2.16.578.1.12.4.1.1.7220	Norwegian NCPM	NCPM Procedure Codes	
2.16.578.1.12.4.1.1.7210	Norwegian NCSP	NCSP Codes	
2.16.578.1.12.4.1.1.7110	Norwegian ICD-10	Norwegian version of the International Classification of Diseases 10th Edition	World Health Organisation
2.16.578.1.12.4.1.1.1302	Norwegian Job classification code	Job classification code (<i>Yrkeskode ;tre-tegnskode</i>)	SSB - Statistisk sentralbyrå
2.16.840.1.113883.6.5	SNOMED	SNOMED codes	College of American Pathologists
2.16.578.1.12.4.1.1.9043	ISO 3166-1 (second edition)	Country codes as per ISO 3166-1:2007	ISO
2.16.578.1.12.4.1.1.3403	Bydelsnummer	Country subdivisions (for county / kommune/ bydel)	SSB – Statistisk sentralbyrå
2.16.578.1.12.4.1.1.3402	Kommunenummer	Country subdivisions (for county / kommune)	SSB – Statistisk sentralbyrå
2.16.578.1.12.4.1.1.3103	Maritalstatus (Sivilstand)	Norwegian marital statuscode according to Folkeregister	Folkeregister
2.16.578.1.12.4.1.1.3101	Administrative gender (Kjønn)	The administrative gender of a person	ISO
2.16.578.1.34.5	Helse Vest code tables	(not used; serves as an intermediate node only)	Helse Vest
2.16.578.1.34.5.1	Administrative Observation types related to patients	Coding system.	Helse Vest
2.16.578.1.34.5.2	Type of observation as to the query match in responses to a “Find Candidates” query	Coding system. One current value: PERC: the percentage of match with the query parameters, as determined by the registry. A higher percentage expresses a better match.	Helse Vest
2.16.578.1.34.5.3	Business-level errors	Coding system. Current values: KNOWNPAT: The identified patient is known to the receiving system. The patient can't be created twice. VALIDATION: The query parameters are not valid or insufficient.	Helse Vest

OID	CodingSystemName	Description	Issuer
		AUTHENTICATION: The user could not be authenticated. AUTHORIZATION: The user is not authorized. OTHER: Other errors	
2.16.578.1.12.4.5.2.1.1	PersonRegistryErrors	INVALPID: Invalid person identifier PARAMERR: Generic Query Parameter SERVEROR: Generic error	

4.1.2. Identification Schemes



Figur 19 OID structure

The following are OIDs for identification schemes:

OID	Identification SystemName	Description	Issuer
2.16.578.1.12.4.1	KITH	Root OID for Helsedirektoratet.	Norwegian Post and telecommunications authority
2.16.578.1.12.4.1.1	Kodeverk	Identification scheme for coding systems published by Helsedirektoratet on http://www.volven.no	Helsedirektoratet
2.16.578.1.12.4.1.4.1	Fødselsnummer	Person identifier (for Norwegians and permanent residents) as contained in the Norwegian person register; length=11 digits (<i>Fødelsenummer, F-nummer</i>)	Skatteetaten
2.16.578.1.12.4.1.4.2	D-nummer	Person identifier (for non-permanent residents) as contained in the Norwegian person register; length=11 digits (<i>D-nummer</i>)	Skatteetaten
2.16.578.1.12.4.1.4.3	Felles-hjelppenummer	Person identifier (for temporary/emergency use) assigned by Norsk Helsenett	Norsk Helsenett
2.16.578.1.12.4.1.4.4	HPR-nummer	Identification system for healthcare practitioners in Norway	Statens autorisasjonskontor for helsepersonell (SAFH)
2.16.578.1.12.4.1.4.101	Organisasjonsnummer	Identification scheme for Norwegian	Brønnøysund

OID	Identification SystemName	Description	Issuer
		organizations (<i>Enhetsregister - Brønnøysund</i>) See www.brreg.no	
2.16.578.1.12.4.1.4.102	RESH-id (Register for Enheter i SpesialistHelsetjenesten)	Identification of departments and organizational units http://www.shdir.no/norsk_pasientregister/resh/	SHDIR

5. Data types and coding systems

5.1. Data Types

5.1.1. PN - Person Name

The Person Name (PN) data type is used to convey a person names. The PN data type is defined as a sequence of name parts. In XML terms the PN data type is a mixed content data type, which would allow for a mixture between textual and structural parts. Textual parts SHALL not be used in Norway.

The structure of the name follows the structure as used by the Folkeregister and divides the name into two (or three) parts:

```
<name>
  <given>Ola</given>
  <given>Petter</given>
  <family>Nordmann</family>
</name>
```

Note that the name components SHALL be sent in the usual Norwegian sequence (given names before the family name).

Names from the Folkeregister should be marked with use=OR (Official Registry).

```
<name use="OR">
  <given>Ola</given>
  <given>Petter</given>
  <family>Nordmann</family>
</name>
```

Note that the name attribute is repeating, i.e. multiple (types of) names could be sent. It is up to the receiver (based on the value of the use attribute, and validTime) to decide what name it wants to use or import in its database.

5.1.1.1. Previous Name

Names that have been valid at some point in time, but are no longer in use can be identified using the OLD use code. Examples include previous, birth and maiden names.

```
<name use="OLD">
  <given>Roland</given>
  <family>Gundersen</family>
</name>
```

Note that a name may have multiple use codes. The example below shows an old name as it was known in the Folkeregister at some point in time, as well as the current name known in the Folkeregister.

```
<name use="OR OLD">
  <given>Rolandus</given>
  <family>Gundersen</family>
</name>
<name use="OR">
```

```
<given>Roland</given>
<family>Gundersen</family>
</name>
```

Note: the concept of an old (no longer valid) name is not supported in Data Types Release 1 (currently used by HL7 Norway). It is however present in Data Types Release 2. On 20100910 HL7 Norway decided (by a 6-0-0 vote) to pre-adopt the OLD use code and to modify the XML Schema accordingly.

5.1.1.2. Mellomnavn

Mellomnavn is the equivalent of a 'middle family name'. If a family name has the qualifier MID it is to be interpreted as the mellomnavn. If it does not have the MID qualifier, it is the "normal" family name. The following example specifies Petterson to be the mellomnavn:

```
<name use="OR">
  <given>Ola</given>
  <family qualifier="MID">Petterson</family>
  <family>Nordmann</family>
</name>
```

Note: the Mellomnavn concept is not supported in Data Types Release 1 (currently used by HL7 Norway). It is however present in Data Types Release 2. On 20100910 HL7 Norway decided (by a 6-0-0 vote) to pre-adopt the MID qualifier and to modify the XML Schema accordingly.

5.1.1.3. Fuzzy Name (for searches)

Most names (instances of the PN data type) are intended not just for exchange, but also for archival purposes.

In those cases where one wishes to designate a name as being suitable only for the purpose of searching (i.e. a phonetic version of a name as entered by a registration clerk), and where the name itself should not be stored one should use the SRCH use code.

Use of the SRCH code indicates two things:

1. The name is known to be 'wrong' or 'approximate', and therefore the receiver is explicitly requested to apply fuzzy matching logic (for example: based on phonetic principles, based on the assumption that given and family name may have been swapped upon data entry, based on common typing mistakes)
2. The name is not to be archived or persisted, its sole purpose is for searching.

The SRCH use code SHALL not be combined with any other use code.

The following example designates the name to be a name for searching purposes. The receiving application may find a match with a record for "Muhammad Incirlik".

```
<name use="SRCH">
  <given>Mohammed</given>
  <family>Insjrlek</family>
```

```
</name>
```

Note: instead of a "real" value the data type could carry a NullFlavor to indicate a) that no "real" value is available, and b) why that value isn't available.

Requirement: All software applications that process data types SHALL have the ability to process NullFlavors instead of a "real" value.

5.1.2.AD - Address

The address (AD) data type is used to convey an address. The AD data type is defined as a sequence of address parts. In XML terms the AD data type is a mixed content data type, which would allow for a mixture between textual and structural parts. Textual parts SHALL not be used in Norway.

The following address parts will be used (and SHALL be sent in the order as shown):

- careOf: contains the name of the person or the organization that's being used for c/o addresses. Should only be used for c/o addresses.
- streetAddressLine: for street name and house number. streetAddressLine may occur multiple times.
- postalCode: for ZIP/postal code
- city: for city (poststed)
- country: defaults to Norway, and should be used for addresses outside of Norway.

Each address may have zero or more use codes to identify the address type. When it comes to the address of the patient, the address from the Folkeregister shall be sent as part of the Person (entity) class, using the HP use code.

```
<patientPerson>
<addr use="HP ">
  <streetAddressLine>Hallskaret 72</streetAddressLine>
  <postalCode>5117</postalCode>
  <city>ULSET</city>
</addr>
</patientPerson>
```

In addition to the Folkeregister address the Patient may have other (temporary) addresses. These shall be sent as part of the Patient (Role) class. Only H (Home), WP (Workplace), PST (Postal address) and TMP (temporary) shall be used. Note: The code HP is reserved for the Folkeregister address.

```
<patient>
  <addr use="H PST">
    <streetAddressLine>Thormøhlens gate 12</streetAddressLine>
    <postalCode>5006</postalCode>
    <city>BERGEN</city>
  </addr>
</patient>
```

Example of a foreign address, followed by a temporary Norwegian Address:

```
<addr use="H PST">
  <streetAddressLine>Storgatan 123</streetAddressLine>
  <postalCode>464 05</postalCode>
  <city>MELLERUD</city>
  <country>Sverige</country>
</addr>
<addr use="TMP PST">
  <streetAddressLine>Thormøhlens gate 12</streetAddressLine>
  <postalCode>5006</postalCode>
  <city>BERGEN</city>
</addr>
```

A C/O address can be used when the contact party is visiting someone, living somewhere temporary or having mail sent to their place of work, and her or his name is not on the mailbox. Example of a C/O (care of) address:

```
<!-- The postal address is manually entered -->
<addr use="TMP PST">
  <careOf>c/o Jane Doe</careOf>
  <streetAddressLine>Somewhere Road 1</streetAddressLine>
  <postalCode>1234</postalCode>
  <city>A Place</city>
</addr>
```

Note: instead of a "real" value the data type could carry a NullFlavor to indicate a) that no "real" value is available, and b) why that value isn't available.

Requirement: All software applications that process data types SHALL have the ability to process NullFlavors instead of a "real" value.

5.1.3. Telephone and mail

The Telecommunication (TEL) data type is used to convey data related to all telecommunication devices used by a person or patient.

The underlying model is based on URLs (see [1] URLs for Telephone Calls). The voice telephone URLs begin with tel:, fax URLs begin with "fax:" and mail address with "mailto".

```
<telecom use="H" value="tel:55354257" />
<telecom use="H" value="fax:56542558" />
<telecom use="WP MC" value="tel:97555786" />
<telecom use="WP" value="mailto:tor.nordmann@helse-vest-ikt.no" />
<telecom use="H" value="mailto:tor.nordmann@broadpark.no" />
```

Note: RFC 2806 doesn't allow for any spaces nor braces in the telephone number.

The codes shown below will be used in Norwegian implementations of the "use" attribute. Note that multiple codes can be specified in the form of a space-delimited list.

- H – Home
- WP – Work
- MC can be used (in addition to H or WP) to denote that this is a mobile phone.

Country codes shall be used if the telephone/fax number is located outside of Norway. The default country code (if not explicitly listed) is 47.

Note: instead of a "real" value the data type could carry a NullFlavor to indicate a) that no "real" value is available, and b) why that value isn't available.

Requirement: All software applications that process data types SHALL have the ability to process NullFlavors instead of a "real" value.

5.1.4. II – Instance Identifier

The Instance Identifier (II) data type is used to identify one single object instance.

The II data type has three parts:

- **root:** identifies the identification mechanism used to assign the extension (e.g. the OID for the *F-number identification mechanism*, or an OID for *Spanish passport numbers*, an OID for *X-Ray machine identifiers assigned by the Stavanger RIS*, an OID for *Electronic Document identifiers created in software application XYZ in OUS*).
 - The root is mandatory in all instances of the II data type.
 - See [OID page](#) and [Helsedirektoratet OID page](#) for applicable OID values.
- **extension:** identifier created according to the identifier mechanism identified by root (e.g. the F-number 12059815671, a passport number A2830373, or X-Ray machine ID CT0003).
 - In Norway the use of extension is mandatory in all instances of the II data type.

- **assigningAuthorityName:** contains a human readable description of the organization that assigned the identifier. This data SHALL not be used for software processing other than storage or display; it solely exists for human display purposes.
 - The use of assigningAuthorityName is optional.
- For the use of II to identify Persons and Patients, see [Identification of Patients and Persons](#).

Examples

F-number:

```
<id root="2.16.578.1.12.4.1.4.1" extension="24109642356" assigningAuthorityName="Folkeregister"/>
```

Message-id: an ever increasing number (@extension)

```
<id root="2.16.578.1.12.4.1.4.1" extension="24109642356" assigningAuthorityName="Folkeregister"/>
```

Note: instead of a "real" value the data type could carry a NullFlavor to indicate a) that no "real" value is available, and b) why that value isn't available.

Requirement: All software applications that process data types SHALL have the ability to process NullFlavors instead of a "real" value.

5.1.5. NullFlavor usage in Data Types

5.1.5.1. An introduction to NullFlavors

A NullFlavor can be used instead of a "real" value in a data type to indicate a) that no "real" value is available, and b) why that value isn't available.

The NullFlavor property is available on all attributes (i.e. it's part of all data type definitions). It is especially useful if a sending application doesn't have any data for an attribute that has a minimum cardinality of 1 in the HL7 model. NullFlavors may not be used in certain attributes, see section below for details.

Examples:

1. No value for a required "pregnant yes/no observation" (we didn't ask)

```
<value nullFlavor="NASK"/>
```

2. A name (although known) that's not provided for privacy reasons (name is masked)

```
<name nullFlavor="MSK"/>
```

Note: HL7 allows for the use of NullFlavors in attributes, and for associations. In Norwegian implementations the use of NullFlavors SHALL be limited to the use in attributes.

Note: if the nullFlavor property is used for an attribute, then (in Norwegian implementations) other attributes (allowed by the datatype) SHALL NOT be valued as well. The example below is non-conformant (the name can not be both masked, as well as present):

```
<name nullFlavor="MSK"><given>Per</given><family>Petersen</family></name>
```

5.1.5.2. NullFlavor Hierarchy

A NullFlavor may have one of the following predefined values:

Lvl	Type, Domain name and/or Mnemonic code	Print Name	Definition/ description
1	S: NoInformation (NI)	NoInformation	The value is exceptional (missing, omitted, incomplete, improper). No information as to the reason for being an exceptional value is provided. This is the most general exceptional value. It is also the default exceptional value.
2	L: (NA)	not applicable	Known to have no proper value (e.g., last menstrual period for a male).
2	S: Unknown (UNK)	Unknown	A proper value is applicable, but not known.
3	L: (NASK)	not asked	This information has not been sought (e.g., patient was not asked)
3	S: AskedButUnknown (ASKU)	AskedButUnknown	This information has not been sought (e.g., patient was not asked)
4	L: (NAV)	Temporarily unavailable	Information is not available at this time but it is expected that it will be available later.
3	L: (TRC)	Trace	The content is greater than zero, but too small to be quantified. This nullFlavor is only applicable in the PQ data type (Physical Quantity).
2	S: Other (OTH)	Other	The actual value is not a member of the set of permitted data values in the constrained value domain of a variable. (e.g., concept not provided by required code system).
3	L: (PINF)	positive infinity	Positive infinity of numbers.
3	L: (NINF)	negative infinity	Negative infinity of numbers.

2	L: (MSK)	Masked	<p>There is information on this item available but it has not been provided by the sender due to security, privacy or other reasons. There may be an alternate mechanism for gaining access to this information.</p> <p>Note: using this null flavor does provide information that may be a breach of confidentiality, even though no detail data is provided. Its primary purpose is for those circumstances where it is necessary to inform the receiver that the information does exist without providing any detail.</p>
---	----------	--------	--

Note that nullFlavors cover multiple scenarios:

- I don't know/have a value (NI, NA, UNK, NASK, ASKU, NAV)
- I do know there's a value, but I'm unable to provide it (TRC, PINF, NINF, OTH, MSK)

5.1.5.3. When are NullFlavors allowed?

The use of nullFlavors is not allowed for certain attributes; this is specified by the conformance rules (explained below in the conformance section) associated with a HL7 version3 model. For optional attributes (with a cardinality [0..n]) the use of nullFlavors isn't a requirement (see section below on cardinality).

Note that an empty XML-element SHALL NOT be used in any part of an HL7 v3 XML-instance. This may satisfy the requirements of the XML-schema, but it doesn't constitute a valid HL7 v3 instance. Instead of an empty XML-element one should provide a nullFlavor.

Conformance .. and NullFlavors

Conformance is a term used by HL7 to specify if, and how, sending and receiving applications should have the capability to process certain parts of a model.

Note that this differs from the cardinality of an attribute, which is only about the presence of an attribute in an XML-instance. An attribute could have a conformance requirement which states that "all receiving applications SHALL have the capability to process this". Such conformance requirements are mostly associated with attributes that HL7 has determined to be essential/key parts of a model; i.e. without having the capability to process certain attributes one can't claim that one really supports the underlying full model.

Each and every attribute (and association) has one out of three conformance requirements:

1. **Optional:** the presence of this attribute is optional. Senders don't have to populate the attribute; receivers don't have to have the ability to process the attribute.
 - In HL7 version 3 models, one can recognize these attributes, because their names are shown in a regular font.
2. **Required:** Senders SHALL put a value in this attribute (and may use a nullFlavor). All receiving applications SHALL have the capability to process the attribute.

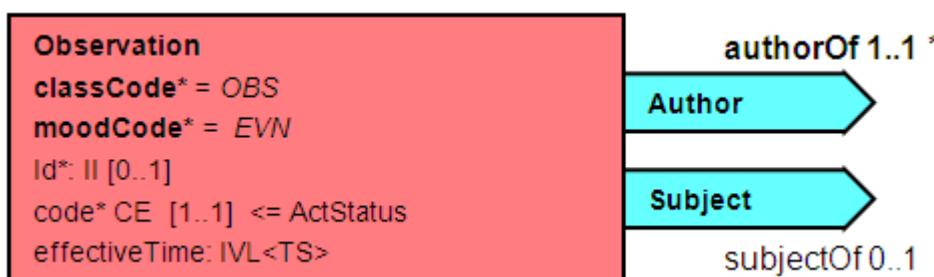
Senders may not put a dummy value in the attribute; the attribute has to be truly supported within the application.

- In HL7 version 3 models, one can recognize these attributes, because their names have an asterisk (*) appended to them.
3. **Mandatory:** Senders SHALL put a 'real' value in this attribute (and nullFlavors SHALL NOT be used). All receiving applications SHALL have the capability to process the attribute. Senders may not put a dummy value in the attribute; the attribute has to be truly supported within the application.
- In HL7 version 3 models, one can recognize these attributes, because their names are shown in a **bold** font, with an added asterisk (*) to the name.

Cardinality .. and NullFlavors

If an attribute has a [0..n] cardinality (i.e. in conformance terms: it's optional or required) one could also simply omit the entire attribute in the XML instance. This is also interpreted as "unknown".

Example Model



Figur 20 Example model: Observation

In the above model:

- **classCode** and **moodCode** are designated as **mandatory** attributes. The **authorOf** association is also designated as mandatory.
 - nullFlavors may not be used.
- **id** and **code** is designated as a *required* attribute.
 - nullFlavors have to be used if no real value is known/applicable.
- **effectiveTime** and the **subjectOf** association are designated as *optional*.
 - nullFlavors may be used.

5.1.5.4. Additional Examples

1. The time at which a prescription was authored is required in a HL7 v3 model. The sending application is however not aware of the time when it was authored.

```

<author>
  <time nullFlavor="UNK"/>
</author>
  
```

2. When sending a list of ingredients of a pharmaceutical product the sending application wishes to identify that it contains 'traces of iron' (i.e. very small amounts, not significant enough to measure).

```
<ingredient>
<code code="Iron" codeSystem="1.2.3.4.5">
<quantity nullFlavor="TRC"/>
</ingredient>
```

5.1.6. Administrative Entity associated with a Patient or Person

5.1.6.1. Administrative Entity

The patient is part of (an inhabitant of) an administrative entity within Norway. The identification of the administrative entity is of importance for a multitude of organizational/administrative purposes, e.g. it has an impact on financial arrangements between the administrative entities, as well as on the choice of preferential healthcare provider organizations where care is delivered. The administrative entity can't be (easily) derived from the patient's address.

The administrative entity could be specified as being:

- A Fylke: the Fylke represents the administrative entity (region) at its highest level of abstraction.
 - If the patient resides outside of Norway a "special" (virtual) Fylke is being used. In addition to the virtual Fylke the country of residence is specified.
- A Kommune: if the Kommune is specified, then this implies the Fylke. A Kommune is part of exactly one Fylke.
- A Bydel: if the Bydel is identified, then this implies the Kommune and Fylke. A Bydel is part of exactly 1 Kommune.

5.1.6.2. Coding Systems

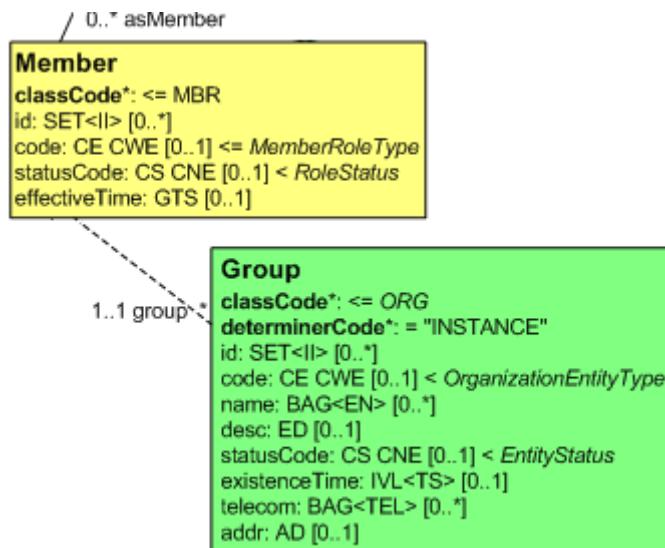
The following coding systems are used in order to identify the administrative entity and the country of residence of the person:

- ISO codes for Countries - should the person reside outside of Norway. This coding system (using the 2-character country codes) has the OID 2.16.578.1.12.4.1.1.9043.
- ISO codes for Country Subdivisions (Fylke/Kommune/Bydel). This coding system has the OID 2.16.578.1.12.4.1.1.3403.
 - For Norway, the code is constructed as follows: "FFKKBB", where FF is the code for the Fylke, KK for Kommune, and BB for Bydel. Example: 120108 is Åsane (Bergen, Hordaland).
- ISO codes for Country Subdivisions (Fylke/Kommune). This coding system has the OID 2.16.578.1.12.4.1.1.3402.
 - The code is constructed as follows: "FFKK" where KK is an optional part of the code. Examples: 1201 is Bergen (Hordaland) and 12 is Hordaland.

Note: as defined in ISO 704, the codes in these coding systems are appellations: they identify one single object. As such the codes can also be used as identifiers.

5.1.6.3. HL7 version 3 Model aspects

The model below documents the main classes used to identify the administrative region. The Person class (not shown) has 0..* asMember associations with the member class.



Figur 21 Model for Administrative region

Class	Component	Documentation
Member	<i>all attributes</i>	None of the attributes in this class are used.
Group	id	The identification of the administrative region, using a code system as identified above.
	code	Contains a code for the "kind of group". Contains the fixed value 'ADMENT' from the coding system with OID "2.16.578.1.34.5.1".

Examples

1. Kommunenummer: The person is a member of the administrative region Bergen/Hordaland

```

<Patient>
  ... other elements ...
  <Person>
    ... other elements ...
    <asMember classCode="MBR">
      <group classCode="PUB">
        <id root="2.16.578.1.12.4.1.1.3402" extension="1201">
          assigningAuthorityName="Kommunenummer"/>
        <code code="ADMENT" codeSystem="2.16.578.1.34.5.1"/>
      </group>
    </asMember>
  </Person>
</Patient>
  
```

2. Bydelsnummer: The person is a member of the administrative region Strinda/Trondheim/Sør-Trøndelag

```

<Patient>
  
```

```

... other elements ...

<Person>
  ... other elements ...
  <asMember classCode="MBR">
    <group classCode="PUB">
      <id root="2.16.578.1.12.4.1.1.3403" extension="160102"
           assigningAuthorityName="Bydelsnummer"/>
      <code code="ADMENT" codeSystem="2.16.578.1.34.5.1"/>
    </group>
  </asMember>
</Person>
</Patient>
```

3. Country: The person is a member of the administrative body which is the country of Sweden.

```

<Person>
  ... other elements ...
  <asMember classCode="MBR">
    <group classCode="PUB">
      <id root="2.16.578.1.12.4.1.1.9043" extension="SE"
           assigningAuthorityName="Landkoder"/>
      <code code="ADMENT" codeSystem="2.16.578.1.34.5.1"/>
    </group>
  </asMember>
</Person>
```

5.2. Coding systems

5.2.1. Maritalstatus (Sivilstand)

The Folkeregisteret uses the following coding system (with OID 2.16.578.1.12.4.1.1.3103) for Maritalstatus (Sivilstand):

- 1 = Ugift
- 2 = Gift
- 3 = Enke/-mann
- 4 = Skilt
- 5 = Separert
- 6 = Registrert partner - Partnership between persons of same sex
- 7 = Separert partner
- 8 = Skilt partner
- 9 = Gjenlevende partner

Example:

```
<maritalStatusCode code="2" codeSystem="2.16.578.1.12.4.1.1.3103"/>
```

5.2.2. Administrative Gender

The administrative gender indicates the gender of a person. Values are: 0 (Not known), 1 (Male), 2 (Female) and 9 (Not applicable). The codes are taken from ISO 5218 with OID 2.16.578.1.12.4.1.1.3101.

Example:

```
<administrativeGenderCode code="1" codeSystem="2.16.578.1.12.4.1.1.3101"/>
```

5.2.3. Healthcare organization types

As coding system for organization types in healthcare (e.g. hospital, GP practice, etc.), the identification scheme for Norwegian organizations will be used. (Enhetsregisteret – Brønnøysund) The OID of the coding system is 2.16.578.1.12.4.1.4.101.

5.2.4. Healthcare provider types

The coding system for healthcare provider types (person roles within healthcare organizations, e.g. physician internal medicine, head nurse, radiologist) has the OID 2.16.578.1.12.4.1.1.9060.

5.2.5. Job code

The coding system for job types is taken from the international standard Standard Classification of Occupations. The responsible organization is SSB. The OID of the coding system is 2.16.578.1.12.4.1.1.1302.

5.2.6. Encounter Types

The table below shows the ActEncounterCode (2.16.840.1.113883.5.4) coding system, which identifies the type of encounters.

. . AMB	ambulatory	A comprehensive term for health care provided in a healthcare facility (e.g. a practitioner's office, clinic setting, or hospital) on a nonresident basis. The term ambulatory usually implies that the patient has come to the location and is not assigned to a bed. Sometimes referred to as an outpatient encounter.
---------	------------	--

. . EMER	emergency	A patient encounter that takes place at a dedicated healthcare service delivery location where the patient receives immediate evaluation and treatment, provided until the patient can be discharged or responsibility for the patient's care is transferred elsewhere (for example, the patient could be admitted as an inpatient or transferred to another facility.)
. . FLD	Field	A patient encounter that takes place both outside a dedicated service delivery location and outside a patient's residence. Example locations might include an accident site and at a supermarket.
. . HH	home health	Healthcare encounter that takes place in the residence of the patient or a designee
. . IMP	inpatient encounter	A patient encounter where a patient is admitted by a hospital or equivalent facility, assigned to a location where patients generally stay at least overnight and provided with room, board, and continuous nursing service.
. . . ACUTE	inpatient acute	An acute inpatient encounter.
. . . NONAC	inpatient non-acute	Any category of inpatient encounter except 'acute'
. . SS	short stay	An encounter where the patient is admitted to a health care facility for a predetermined length of time, usually less than 24 hours.
. . VR	virtual	A patient encounter where the patient and the practitioner(s) are not in the same physical location. Examples include telephone conference, email exchange, robotic surgery, and televideo conference.

5.2.7. Administrative Observation types

A coding system for Administrative Observations related to Persons and Patients. The coding system has OID 2.16.578.1.34.5.1.

The coding system contains the following coded concepts:

Code	Short description	Notes
ADMREG	Fylke, Kommune, Bydel	Contains either Fylke, or Fylke and Kommune, or Fylke, Kommune and Bydel
ADMFYLYL	Fylke	Contains Fylke (only)
ADMKOM	(Fylke and) Kommune	Contains Fylke and Kommune
ADMBYD	(Fylke, Kommune and) Bydel	Contains Fylke, Kommune and Bydel
ADMSTT	Stat (Country)	Contains an identifier of the Country

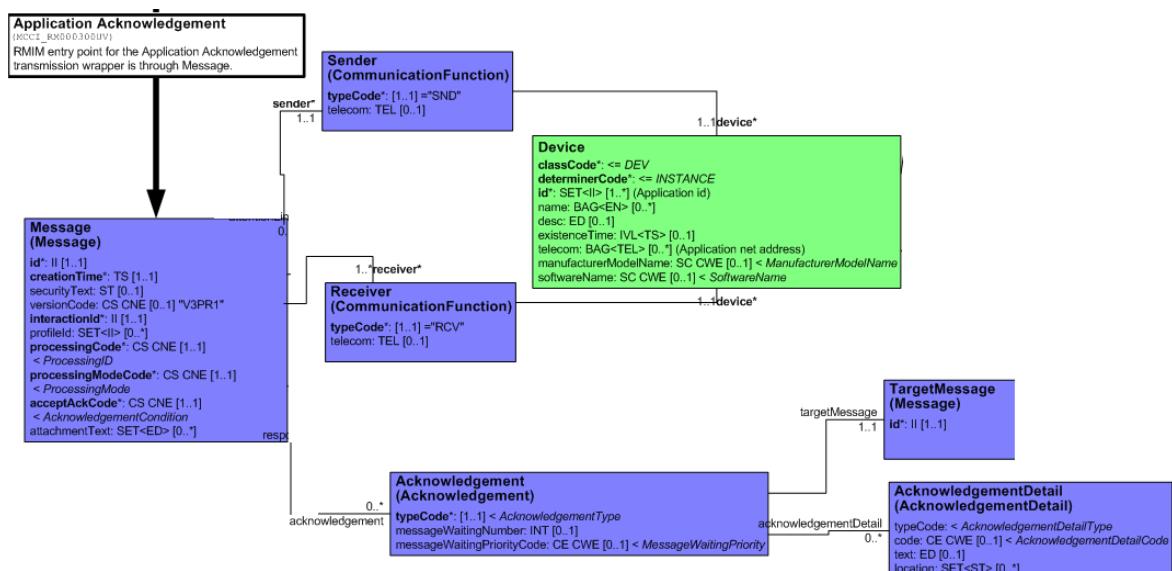
6. Wrappers

6.1. Transmission Wrapper

All HL7 version 3 interactions contain a Transmission Wrapper, an optional [Trigger Event Control Act Wrapper](#) (mostly referred to as the ControlAct Wrapper, present in 99% of all interactions except on: the [Accept Acknowledgement](#)) and a so-called Payload (in 90% of interactions there is exactly 1 payload, some interactions –notably responses to queries– may contain zero or multiple payloads).

The Transmission Wrapper has the aim to identify the sending and receiving applications, the time that the interaction was sent, as well as some other meta data related to the interaction.

6.1.1. Information Model



Figur 22 Application acknowledgement transmission wrapper

The entry class in this model is the Message class. Note that the root element of the XML instance is equal to the interaction type, e.g. PRPA_IN201103NO. It has the same value as the Message/interactionId/@extension XML attribute. The Message class has the following attributes:

Class	Component	Documentation
Message	id	The id contains the unique identification of this instance of the interaction. This attribute corresponds to the II data type .
	creationTime	Contains the date/time the interaction was sent. Format: YYYYMMDDHHMMSS.
	versionCode	Contains the base version of HL7 version 3 used in this interaction. In Norway this should either contain the fixed

		value 'NE2008' (if you use the specifications prior to 2011) or the fixed value 'NE2010NO' (if you use these HL7 specifications which were created in or after 2011).
	interactionId	<p>Identifies the interaction ID (the type of service). @root contains the fixed value '2.16.840.1.113883.1.6'. @extension contains the identifier as specified for the interaction, e.g. PRPA_IN201103NO. The value of @extension has to be equal to the name of the root element in the XML instance. This attribute corresponds to the II data type.</p>
	processingCode	<p>ProcessingCode is a mandatory attribute within all interactions. @code shall either have the value T (Test) or the value P (Production).</p> <ul style="list-style-type: none"> • P = Production. The receiver of the interaction shall process the contents of the interaction in a production environment and the production database. If an interaction with a processingCode equal to P is received by a test application (based on a test database) the receiving application shall not process the interaction and shall create an error message. • T = Test. The receiver of the interaction shall process the contents of the interaction in a test environment and the test database. If an interaction with a processingCode equal to T is received by a production application (based on a production database) the receiving application shall not process the interaction and shall create an error message. <p>Note that all responses related to an interaction have to use the same ProcessingCode as the original interaction. If the original interaction had ProcessingCode 'T', then all response interactions that it results in shall also have processingCode 'T'.</p>
	processingModeCode	No further description. Fixed value 'T'.
	acceptAckCode	Specifies whether or not the receiver should send an Accept Acknowledgement (MCCI_IN000002UV02 interaction) in response to this interaction. In the context of the Norwegian infrastructure the value of Message.acceptAckCode will always be either AL (Always) or NE (Never), and the value will be implied by the WSDL used for the service. Therefore, the receiving application will always be aware if an Accept Acknowledgement has to be sent simply because of the WSDL definition. The value of the acceptAckCode may

		not conflict with the expectations as documented in the WSDL.
--	--	---

A Message has exactly one Sender and one Receiver associated with it; these are modeled by means of the **Sender**, **Receiver** and **Device** classes. These classes have the following attributes:

Class	Component	Documentation
Sender/Receiver	typeCode	Identifies the sending/receiving software application by means of an abstract identifier. @typeCode has the fixed value 'RCV' (for receiver) or 'SND' (for sender).
Device	classCode, determinerCode	Identifies a software application. @classCode has the fixed value 'DEV', @determinerCode has the fixed value 'INSTANCE'
	id	The id contains the unique identification of a software application. This attribute corresponds to the II data type .

If the Transmission Wrapper is part of a response interaction (i.e. an interaction sent as a response to a prior interaction) instead of the 'initial interaction' the **Acknowledgement** and **targetMessage** classes are mandatory in the Transmission Wrapper. Their purpose is to indicate if the original interaction could be processed; and to identify the original interaction that this interaction forms a response to. The Acknowledgement and targetMessage classes have the following attributes:

Class	Component	Documentation
Acknowledgement	typeCode	<p>The acknowledgement part is mandatory in response interactions, and may not be used in 'initiating interactions'. The purpose of the Acknowledgement class is to identify if the interaction has been successfully processed, and to identify the interaction to which this is a response. @typeCode identifies if the interaction has been successfully processed.</p> <ul style="list-style-type: none"> • If the response interaction is an Accept Acknowledgement (MCCI_IN000002UV02) then the allowable values for @typeCode are either 'CE' (error, content not processed), or 'CA' (accepted, contents processed). • For any other interaction the allowable values for @typeCode are either 'AE' (error, content not processed), or 'AA' (accepted, contents processed). <p>See Error Handling for details about identifying the errors themselves.</p> <p>Warning: in NE2008 the HL7 typeCode attribute was mapped to an XML element, see example below:</p>

		<pre><acknowledgement> <typeCode code="AE"/></pre> <p>In NE2010 the HL7 typecode attribute is mapped to an XML attribute, see example below:</p> <pre><acknowledgement typeCode="AE"/></pre>
targetMessage	id	<p>Identifies the original interaction to which this is a response. This is equal to the values of /id/@root and /id/@extension as contained in the original interaction. This attribute corresponds to the II data type.</p>

If the Transmission Wrapper is part of a response interaction (i.e. an interaction sent as a response to a prior interaction) instead of the 'initial interaction' the **AcknowledgementDetail** class can be used to identify zero or more errors/warnings/informational issue detected during the processing of the original interaction. The AcknowledgementDetail class contains the following attributes:

Class	Component	Documentation
AcknowledgementDetail	typeCode	<p>Indicates if the class documents an error, a warning or informational data. Contains either E (Error), W (Warning), or I (Informational).</p> <p>If the typeCode equals 'E', Acknowledgement.TypeCode (see above for description) SHALL be set to either CE or AE.</p>
	code	<p>Contains a code that uniquely identifies the error/warning/information item. See Error Handling for details of the coding system used.</p> <p>If the typeCode equals 'E' a code SHALL be specified in Norwegian implementations. A code SHOULD be specified if the typeCode equals W or I.</p> <p>This code is used by the receiving software application for processing logic (e.g. decide if the problem can be taken care of automatically, or to display the issue to a user or system operator)</p>
	text	<p>Contains (optional) additional textual information about the error. The text should preferably be written in such a way to be understandable by an end user.</p>

Note: Any other classes/attributes not listed above, but shown in the diagram, are reserved for future use in Norway. Sending applications SHOULD not use these model elements; receiving applications SHALL nor produce an error if these model elements are present, and SHALL ignore these model elements if present.

6.1.2. Examples

1. The interaction below is of type PRPA_IN201307NO. It was sent on 20080719140010, from application 145 to application 922.

```
<PRPA_IN201307NO01 ITSType="XML_1.0" xmlns="urn:hl7-org:v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"      >
  <id extension="3948375" root="2.16.578.1.34.1.145.1"/>
  <!-- Time message was sent -->
  <creationTime value="20080719140010"/>
  <versionCode code="NE2010NO"/>
  <!-- Fixed values for GetPatientDemographics query -->
  <interactionId extension="PRPA_IN201307NO01" root="2.16.840.1.113883.1.6"/>
  <processingCode code="P"/>
  <processingModeCode code="T"/>
  <acceptAckCode code="NE"/>
  <receiver>
    <device>
      <!-- Receiving software application. Helse Vest assigned application identifier -->
      <id extension="922" root="2.16.578.1.34.1"/>
    </device>
  </receiver>
  <sender>
    <device>
      <!-- Sending software application. Helse Vest assigned application identifier -->
      <id extension="145" root="2.16.578.1.34.1"/>
    </device>
  </sender>
  <controlActProcess>
```

See [ControlAct Wrapper](#) for details of the secondary wrapper and the payload.

```
</controlActProcess>
</PRPA_IN201307NO01>
```

2. Application Acknowledgement of type PRPA_IN090002 - interaction 0806071541103 is rejected (CE) with errors (E217 code)

```
<PRPA_IN090002 ITSType="XML_1.0" xmlns="urn:hl7-org:v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v3">
  <id extension="34236" root="2.16.578.1.34.1.871.3"/>
  <creationTime value="20080607154128"/>
  <versionCode code="NE2010NO"/>
  <interactionId extension="PRPA_IN090002" root="2.16.840.1.113883.1.6"/>
  <processingCode code="P"/>
  <processingModeCode code="T"/>
  <acceptAckCode code="NE"/>
  <receiver typeCode="RCV">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="408" root="2.16.578.1.34.1"/>
```

```
</device>
</receiver>
<sender typeCode="SND">
  <device classCode="DEV" determinerCode="INSTANCE">
    <id extension="871" root="2.16.578.1.34.1"/>
  </device>
</sender>
<controlActProcess>
```

See [ControlAct Wrapper](#) for details of the secondary wrapper and the payload.

```
</controlActProcess>
<acknowledgement typeCode="AE">
  <acknowledgementDetail typeCode="E">
    <code code="E217" displayName="Invalid F-Number"
      codeSystem="2.16.840.1.113883.5.1100"/>
  </acknowledgementDetail>
  <targetMessage>
    <id extension="0806071541103" root="2.16.578.1.34.1.408.7"/>
  </targetMessage>
</acknowledgement>
</PRPA_IN090002>
```

6.2. ControlAct Wrapper

All HL7 version 3 interactions contain a [Transmission Wrapper](#), an optional Trigger Event Control Act Wrapper (mostly referred to as the ControlAct Wrapper, present in all interactions except one: the [Accept Acknowledgement](#)) and a so-called Payload (in 90% of interactions there is exactly 1 payload, some interactions –notably responses to queries– may contain zero or multiple payloads).

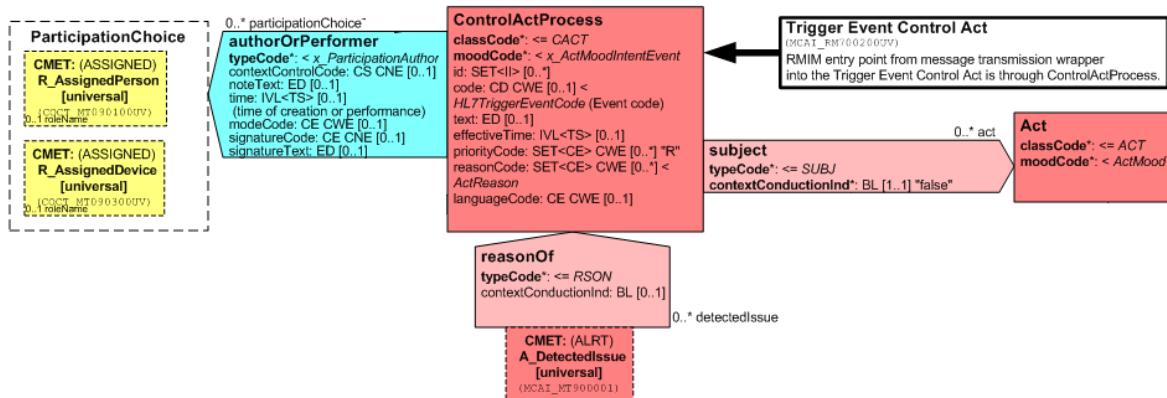
The ControlAct wrapper has the aim to identify the Trigger Event (the underlying activity that caused the necessity for the information exchange) and the responsible party who caused the trigger event to happen.

Example: If Dr. Smith decides to change the status of a radiology observation to 'final' (in HL7 v3 terms: from 'active' to 'completed') then this status change is the trigger event for sending a 'radiology result update' message. If Nurse Jones decides to send a query to the Folkeregistret then her decision to do so is the trigger event.

The ControlAct wrapper, if used in query interactions or responses thereto, also supports the specification of query parameters and the sending of query responses.

Example: If one queries Folkeregisteret using a Person-Name, a Birthdate and a Postalcode, these three query parameters are modeled as part of the ControlAct wrapper. If the response to that query contains 5 matching records, that quantity (5) is modeled as part of the ControlAct wrapper as well. The matching records are Payload models and as such aren't part of the ControlAct wrapper.

6.2.1. Information Model



Figur 23 The controlActProcess

The **controlActProcess** class forms the entry class into the ControlAct wrapper model. The class represents the Trigger Event the underlying activity that caused the necessity for the information exchange). It contains the following elements:

Class	Component	Documentation
controlActProcess	moodCode, classCode	The controlActProcess identifies the person or software application that is responsible for triggering the interaction, i.e. the person/application that has sent the interaction.

		@moodCode has the fixed value 'EVN'. @classCode has the fixed value 'CACT'.
--	--	---

The **authorOrPerformer**, **assignedDevice** and **assignedPerson** classes jointly identify the party responsible for causing the Trigger Event. These classes have the following elements:

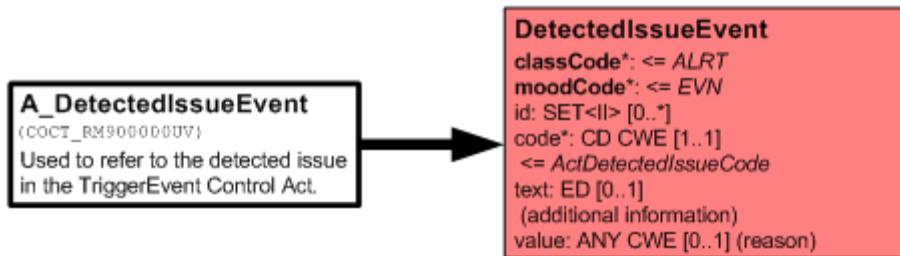
Class	Component	Documentation
authorOrPerformer	typeCode	<p>Identifies the sender (the author) of this interaction. For auditing purposes all interactions SHALL contain the identification of the author. The author can be either a person, or a software application.</p> <p>@typeCode contains the fixed value 'AUT'. The authorOrPerformer is identified as either an assignedDevice or an assignedPerson.</p>
assignedPerson	classCode	<p>Identifies a person. @classCode has the fixed value 'ASSIGNED' (either assignedPerson or assignedDevice must be present)</p>
	id	<p>This class SHALL contain an identifier. The id contains a unique identification of a person. The attribute uses the II data type.</p> <p>As long as there is no national identifier for all employees in healthcare the sender</p> <ul style="list-style-type: none"> • SHALL send the 'software user identifier / login name' as assigned by the sending organization. Each software application that assigns 'user identifiers' will need to assign an OID to their own identification scheme. • SHOULD (additionally) send the HPR number.
assignedDevice	classCode	<p>Identifies a software application. Software applications can be the author of an interaction if the contents of the interaction (the payload) was generated automatically, e.g. whenever a software application generates a response to a query. @classCode contains the fixed value 'ASSIGNED'</p> <p>(either assignedDevice or assignedPerson must be present)</p>
	id	<p>The id contains the unique identification of a software application. The attribute uses the II data type.</p>

The **subject** act relationship associates zero or more Payload models with the ControlAct. The subject act relationship has no attributes associated with it.

Note: Any other classes/attributes not listed above, but shown in the diagram, are reserved for future use in Norway. Sending applications SHOULD not use these model elements;

receiving applications SHALL nor produce an error if these model elements are present, and SHALL ignore these model elements if present.

6.2.1.1. Detected issues



Figur 24 *DetectedIssueEvent*

The **reasonOf** and **DetectedIssueEvent** classes can be used to identify "the reason for a trigger event". These classes are most often used (in responses) to identify errors/issues found in the message that is being responded to. The issues identified in the **DetectedIssueEvent** class have a business-processing characteristic (i.e. issues detected during the processing of the contents of a message, not during its syntax check). See [Error Handling](#) for additional details.

Example: A laboratory receives an order to perform a "pregnancy test for a male patient". The laboratory system responds with a "reject order" interaction; the **DetectedIssueEvent** class is used to identify the underlying reason (males can't be pregnant).

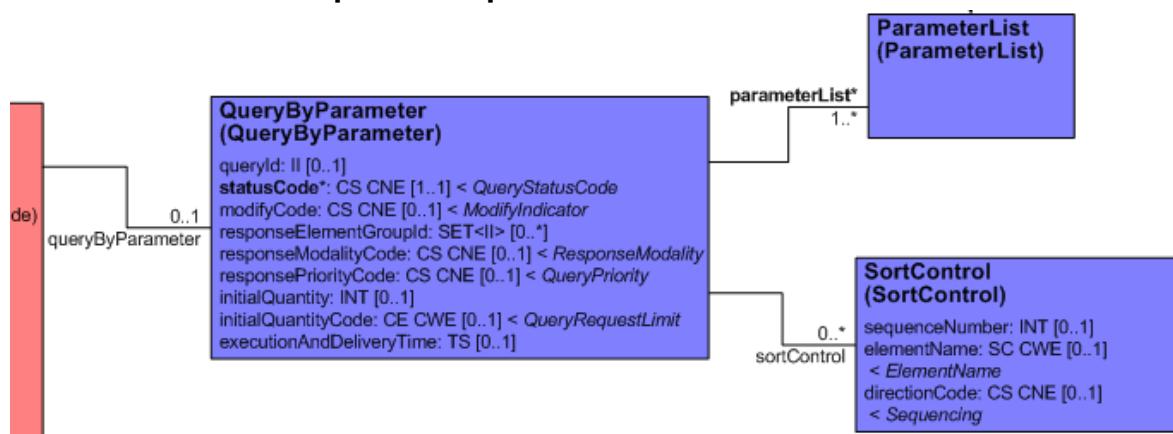
Example: A pharmacy system receives a prescription for medication X. The pharmacy system is aware that the patient is currently taking medication Y. For that reason, the pharmacy system sends a response message to reject the prescription; the **DetectedIssueEvent** class is used to identify the underlying reason (X has a contra-indication Y).

Class	Component	Documentation
reasonOf	id	This element occurs only in response interactions. It contains the identification of a business/process-level error. In the current project it is used to identify why a PatientRegistry.Add request could not be fulfilled. @typeCode has the fixed value 'RSON'.
DetectedIssueEvent		Identifies one single issue/error.
	code	<p>This attribute identifies the kind of detected issue. For example: "medication contra-indication", or "Duplicate Patient ID".</p> <p>@codeSystem contains the OID of the error code table, @code contains a code from that table, and @displayName contains a human readable description of the code. See Error Handling for additional details and the coding system.</p> <p>Example:</p> <pre><code code="KNONPAT" codeSystem="2.16.578.1.34.5.3"</pre>

		displayName="Patient already known. This service requires a new patient."/>
	value	<p>This attribute contains additional information about the specific issue. For example: "Medication X causes a conflict with medication Y", or "Patient ID 1234 was already created on 2009-09-01".</p> <p>The value attribute has data type ANY, which is mostly overridden to be ST (string).</p> <p>Example:</p> <pre><value xsi:type="ST">Patient ID 1234 was already created on 2009-09-01</value></pre>

Note: Any other classes/attributes not listed above, but shown in the diagram, are reserved for future use in Norway. Sending applications SHOULD not use these model elements; receiving applications SHALL nor produce an error if these model elements are present, and SHALL ignore these model elements if present.

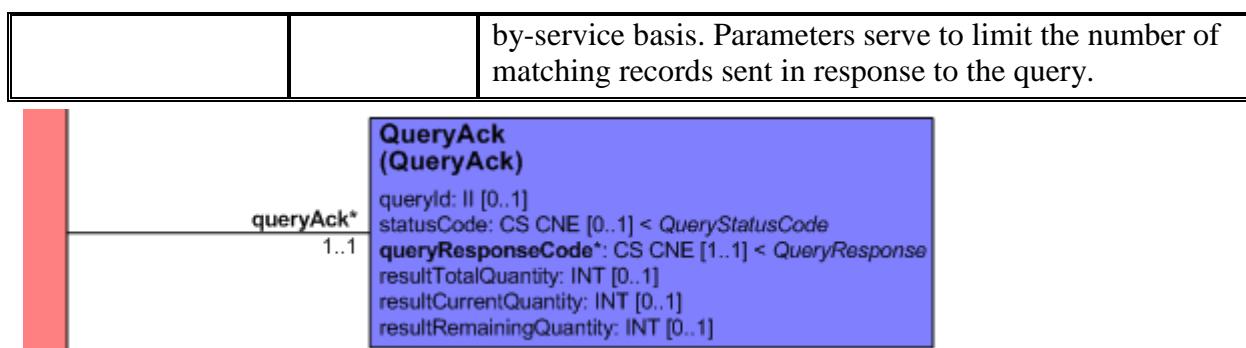
6.2.1.2. Queries/Response to queries



Figur 25 QueryByParameter

The **queryByParameter** class (and related parameter classes) is used to specify the details of a query. The class is only used in query messages. The **queryByParameter** class contains the following elements:

Class	Component	Documentation
queryByParameter		The QueryByParameter class contains the specification of the query.
	queryId	The id contains the unique identification of this query instance. Note: re-use same @extension/@root as present in the Message.id attribute of the Transmission Wrapper .
	statusCode	@code contains the fixed value ‘new’
parameters	value	The individual parameters are documented on a service-



Figur 26 QueryAck

The **queryAck** class is used to convey the number of matching records returned in the response. This class is only used in messages that are a response to a query. The **queryAck** class contains the following elements:

Class	Component	Documentation
queryAck		This element only occurs in responses to queries. It contains information about the number of responses in the subject payload.
	queryId	The id contains the unique identification of the original query instance (the value of the QueryByParameter.queryId attribute).
	queryResponseCode	@code contains either 'OK' (if one or more records were found), 'NF' (if zero matching records were found), or 'QE' (if the query couldn't be processed because of errors related to the query parameters – see Error Handling for details).
	resultCurrentQuantity	Contains the number of responses/records in this response interaction. @value may be 0 if there were zero matching records.
	resultRemainingQuantity	Contains the fixed value '0'.

Note: Any other classes/attributes not listed above, but shown in the diagram, are reserved for future use in Norway. Sending applications **SHOULD** not use these model elements; receiving applications **SHALL** nor produce an error if these model elements are present, and **SHALL** ignore these model elements if present.

6.2.2. Examples

Note that in the examples below neither the [Transmission Wrapper](#) nor the payload model is shown.

1. The author of the interaction is a person (e.g. in queries, or request interactions):

```

<controlActProcess moodCode="EVN">
    <authorOrPerformer typeCode="AUT">

```

```

<assignedPerson>
  <id extension="teneur" root="2.16.578.1.34.3.1"/>
  <id extension="1234567" root="2.16.578.1.12.4.1.4.4"/>
</assignedPerson>
</authorOrPerformer>
<subject>
  ...
</subject>
</controlActProcess>

```

2. The author of the interaction is a software application (e.g. in an automated response to a query):

```

<controlActProcess moodCode="EVN">
  <authorOrPerformer typeCode="AUT">
    <assignedDevice>
      <id extension="922" root="2.16.578.1.34.1"/>
    </assignedDevice>
  </authorOrPerformer>
  <subject>
    ...
  </subject>
</controlActProcess>

```

3. Parameter specification in a query with QueryId 80622193108_11. The parameters themselves aren't shown.

```

<queryByParameter>
  <queryId extension="80622193108_11" root="2.16.578.1.34.1.805.1"/>
  <statusCode code="new"/>
  <parameterList>
    ...
  </parameterList>
</queryByParameter>

```

4. Response to the query in example 3. The query could be successfully processed (OK), with 2 matching records.

```

<queryAck>
  <queryId extension="80622193108_11" root="2.16.578.1.34.1.805.1"/>
  <queryResponseCode code="OK"/>
  <resultCurrentQuantity value="2"/>
  <resultRemainingQuantity value="0"/>
</queryAck>

```

6.3. Registration Act

The Registration Act is a modeling construct used in Registries. It, and its associated classes, captures meta information about a registration of a role (e.g. patient, person) in a

registry. The meta information includes such things as: who registered the role, when was the role registered and whether or not the registration is still an 'active' one (i.e. whether the Role can be used and queried for).

From a modeling perspective it forms a predefined part of the Payload of an interaction. From an implementation perspective it appears to be part of the [ControlAct Wrapper](#) (the registration act is included in the XML schema for a particular type of ControlAct wrapper).

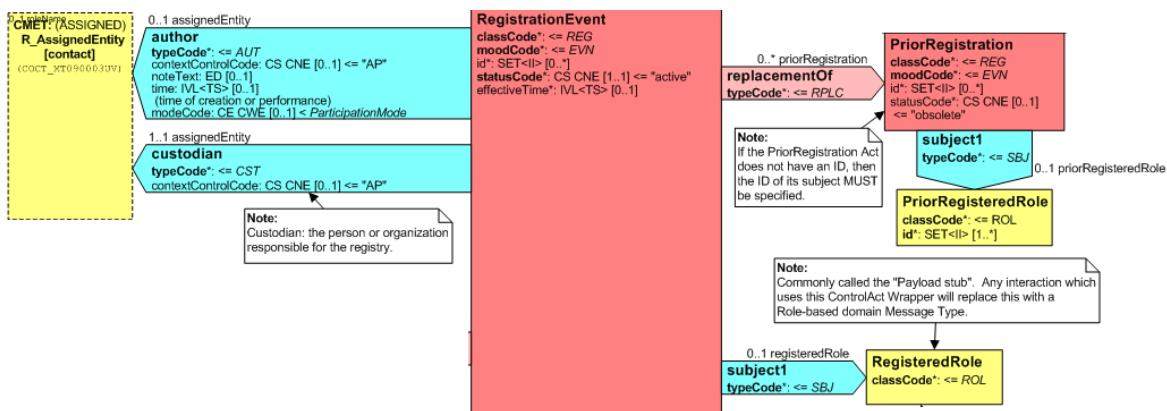
6.3.1. Information Model

The Registration Act occurs in multiple flavors. Each flavor is documented in a separate section below.

- RegistrationRequest: this flavor is used in a request message sent to a Registry with the intent to request that a new registration be added to that registry.
- RegistrationEvent: this flavor is used to send (responses to requests/notifications of changes) from a Registry to a receiving software application.

6.3.1.1. RegistrationEvent

The RegistrationEvent flavor is most often used. It is used to send (responses to requests/notifications of changes) from a Registry to a receiving software application. The model is shown below:



Figur 27 RegistrationEvent

The entry class in this model is the **RegistrationEvent** class. A registry assigns a unique identification to each and every registration (a different Id than the Id of the Role being registered). The registrationEvent class has the following attributes:

Class	Component	Documentation
RegistrationEvent	classCode, moodCode	The registration activity of this set of Role-related data. @classCode contains the fixed value 'REG', @moodCode contains the fixed value 'EVN'
	id	Identifies the registration process within the Registry of a set of Role-related data. If this identifier can't be valued because the registry doesn't support it, the nullFlavor

		'UNK' should be used.
	statusCode	@statusCode contains the registration status of the registration. The value will be either 'active' (for registrations that can be actively used and queried for) or 'obsolete' (for registrations that have been replaced by other registrations, e.g. after a merge).
	subject1	Identifies the object that is in the registry. @typeCode contains the fixed value 'SBJ'.
PayloadRole		This is a placeholder for the Role and its associated details. For example: the Generic Patient Model .

The **custodian** and **assignedEntity** classes identify the organization responsible for the Registry. These classes contain the following attributes:

Class	Component	Documentation
custodian	typeCode	Identifies the custodian of the registration, the organization that maintains the registry. @typeCode contains the fixed value 'CST'.
assignedEntity	id	Contains the identity of the organization that has the responsibility for the Registry (and this registration). This attribute uses the II datatype .

The **replacementOf**, **priorRegistration** and the **priorRegisteredRole** classes are only applicable if an earlier registration is being replaced by a new one. These classes may not be used in Norwegian implementations.

Examples:

1. Registration information sent by the Role-registry when a new Role has been registered or an update has been made to an existing registration.

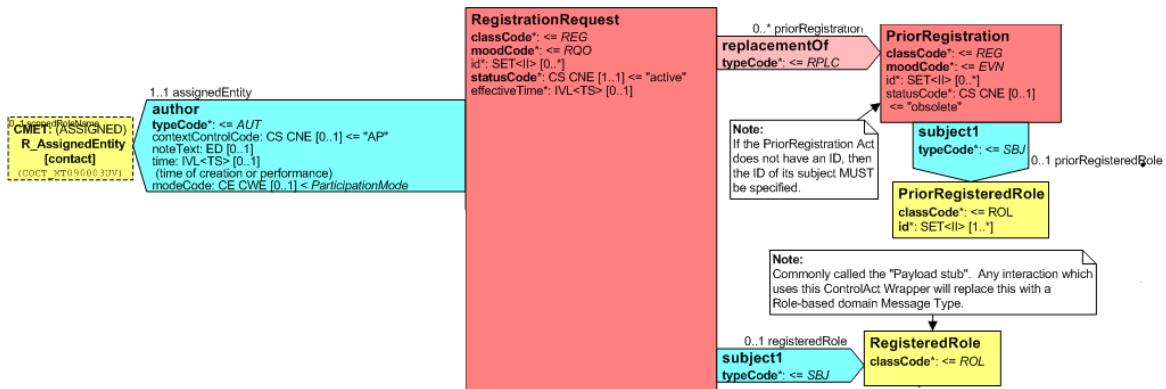
```
<registrationEvent classCode="REG" moodCode="EVN">
  <id nullFlavor="UNK"/>
  <statusCode code="active"/>
  <subject1 typeCode="SBJ">
    <patient classCode="PAT">
      ... details not shown ..
    </patient>
  </subject1>
  <custodian typeCode="CST">
    <assignedEntity classCode="ASSIGNED">
      <id extension="408" root="2.16.578.1.34.1"/>
    </assignedEntity>
  </custodian>
</registrationEvent>
```

Note: Any other classes/attributes not listed above, but shown in the diagram, are reserved

for future use in Norway. Sending applications SHOULD not use these model elements; receiving applications SHALL nor produce an error if these model elements are present, and SHALL ignore these model elements if present.

6.3.1.2. RegistrationRequest

This Registration Act flavor is used in a request message sent to a Registry with the intent to request that a new registration be added to that registry. The model is shown below:



Figur 28 RegistrationRequest

The entry class in this model is the **RegistrationRequest** class. The registrationRequest class has the following attributes:

Class	Component	Documentation
RegistrationRequest	classCode, moodCode	The registration activity of this set of Role-related data. @classCode contains the fixed value ‘REG’, @moodCode contains the fixed value ‘RQO’
	id	@nullFlavor contains the fixed value ‘UNK’.
	statusCode	@statusCode contains the fixed value ‘active’
	subject1	Identifies the object that is to be added to the registry. @typeCode contains the fixed value ‘SBJ’.
PayloadRole		This is a placeholder for the Role and its associated details. For example: the Generic Patient Model .

The **author** and **assignedEntity** classes identify the organization responsible for the Registry. These classes contain the following attributes:

Class	Component	Documentation
author	typeCode	Identifies the author of the registration request. @typeCode contains the fixed value ‘AUT’.
assignedEntity	id	Contains an identifier for the person who requested the

		<p>registration. This attribute uses the II datatype.</p> <p>As long as there is no national identifier for all employees in healthcare the sender</p> <ul style="list-style-type: none"> • SHALL send the 'software user identifier / login name' as assigned by the sending organization. Each software application that assigns 'user identifiers' will need to assign an OID to their own identification scheme. • SHOULD (additionally) send the HPR Number.
--	--	---

Example:

1. Request to add a patient (PAT) role to a registry. The request is authored by the person with Id 12345. In this example that Id identifies a software application user within Helse Vest.

```
<registrationRequest classCode="REG" moodCode="RQO">
  <id nullFlavor="UNK"/>
  <statusCode code="active"/>
  <subject1 typeCode="SBJ">
    <patient classCode="PAT">
      .. details of the patient role not shown ..
    </patient>
  </subject1>
  <author typeCode="AUT">
    <assignedEntity classCode="ASSIGNED">
      <id extension="12345" root="2.16.578.1.34.3.1"/>
    </assignedEntity>
  </author>
</registrationEvent>
```

Note: Any other classes/attributes not listed above, but shown in the diagram, are reserved for future use in Norway. Sending applications SHOULD not use these model elements; receiving applications SHALL nor produce an error if these model elements are present, and SHALL ignore these model elements if present.

7. Reusable Models

This section documents some of the payload models used in multiple HL7 version 3 interactions.

7.1.1. Generic Person Model

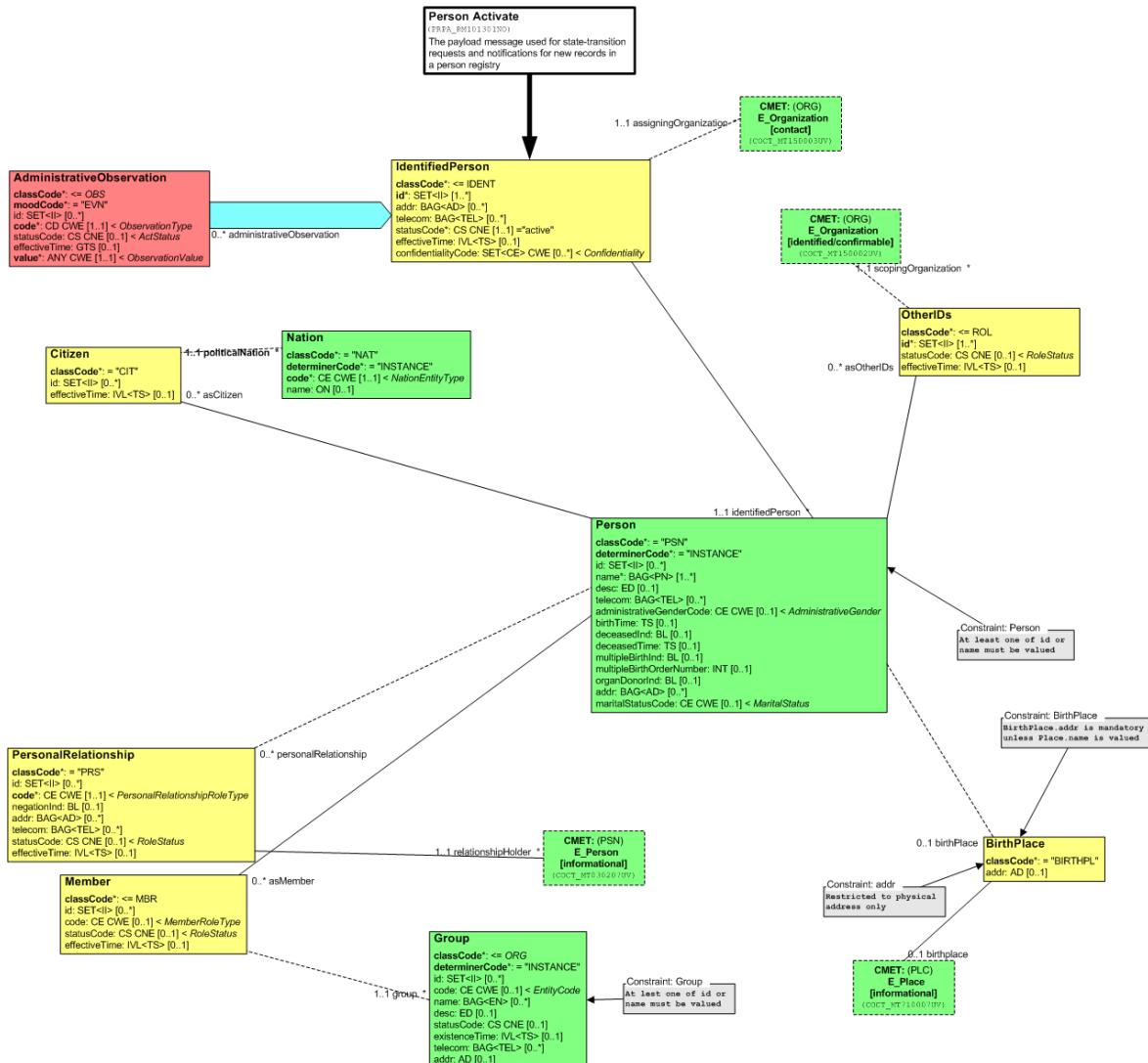
This page documents the core of the Person (payload) model as used by multiple operations. This model has the aim to convey identifying and demographics data related to a person entity. Note: in most circumstances (if the Person has either an F- or D-number) the values for these elements and attributes are populated by Folkeregisteret.

Note: The Norwegian Person Registry project is based on the assumption that the core of all Person models is one and the same (i.e. the core part comprised of IdentifiedPerson, assigningOrganization/E_Organization, Person and classes related to Person).

The PersonRegistry.AddPerson operation is the sole exception: the model used by this operation doesn't contain the IdentifiedPerson.id attribute nor a Person.id attribute.

7.1.1.1. Information Model

The image below is shown for reference purposes only.



Figur 29 The Person model

The following parts of the above model are known to be useful in the Norwegian context and are documented individually:

- IdentifiedPerson and Person
- PersonalRelationship and E_Person (not yet documented)
- Member and Group
- Birthplace
- Citizen and Nation
- Guardian (not yet documented)

The following part (comprised of the IdentifiedPerson and the Person classes) is always present:

Class	Component	Documentation
-------	-----------	---------------

Person	id	Contains a maximum of one unique person identifier. @root contains an identification of the ‘unique person identification mechanism’ (the OID for F-number: 2.16.578.1.12.4.1.4.1, or the OID for D-number: 2.16.578.1.12.4.1.4.2), and @extension contains the identifier created according to that identification mechanism. If both F-number and D-number are known, only the (current) F-number should be sent using this element.
	Name	Occurs one or more times. Contains the name(s) of the person. See section Person Name (PN) data type for a description of the sub elements and the usage of the sub elements.
	administrativeGenderCode	Contains a code for the administrative gender of the person. @codeSystem is fixed to 2.16.578.1.12.4.1.1.3101, @code should be either 0 (Not known), 1 (Male), 2 (Female), or 9 (Not applicable).
	birthTime	Contains the date of birth of the person. At least the year has to be specified. The format of @value is YYYYMMDD
	deceasedInd	Optional. Indicates whether or not the person is deceased. @value is set to ‘true’ in that case. <i>Note: The element is optional if the person is alive, but mandatory if the person is deceased.</i>
	deceasedTime	Optional. Contains the date the person deceased. At least the year has to be specified. The format of @value is YYYYMMDD. <i>Note: The element is mandatory if element deceasedInd.value equals ‘true’.</i>
	Addr	Occurs one or more times. Contains the address(es) of the person. See section Address(AD) data type for a description of the sub elements and the usage of the sub elements.
	maritalStatusCode	Contains a code for the marital status of the person. @codeSystem is fixed to 2.16.578.1.12.4.1.1.3103, @code should be taken from the coding system with OID=3103.

A Person may have one Birthplace class associated with it:

Class	Component	Documentation
BirthPlace	Addr	Identifies the birth place of the person. See section Address (AD)

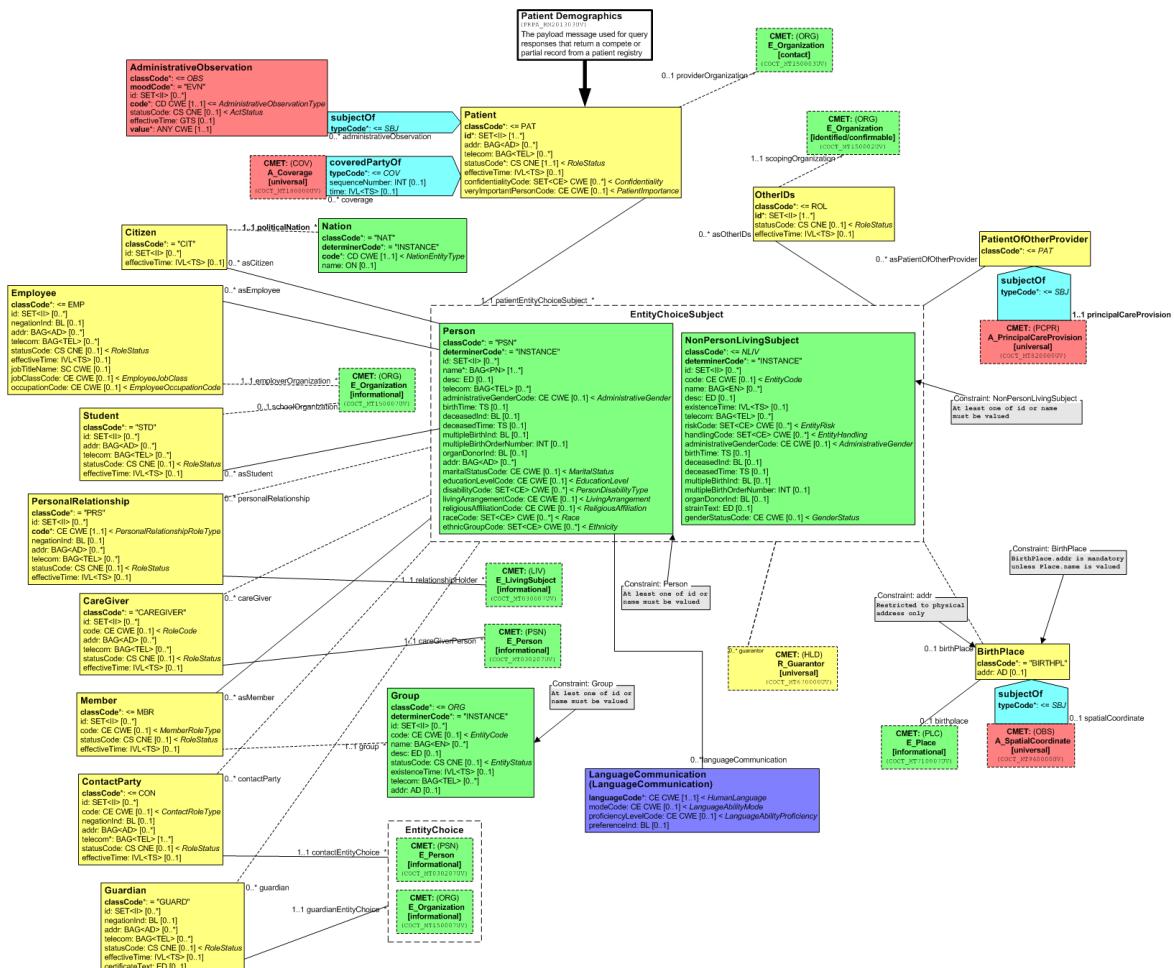
		data type for a description of the sub elements. <i>Note: For birth place Country and possibly City are the relevant attributes, it is not necessary to populate Streetname etc.</i>
--	--	---

Remarks

If using Normative Edition 2008 schemas (NE2008), there is a bug there stating that "ContactParty" is mandatory. E.g. it has to be present, but with a nullFlavor.

7.1.2. Generic Patient payload model

7.1.2.1. Information Model



Figur 30 The Patient payload model

The Patient payload model has the aim to convey identifying and demographics data related to a person entity. The description below is limited to those elements used in this project.

Nesting	Element	Attributes
---------	---------	------------

Level		
0	<i>Root element</i>	
	The name of the root element of this model is determined by its context.	
1	id	@root, @extension
	<p>Contains one Patient.id (either the F-number, the D-number, FH-number or the H-number –in that order of preference-).</p> <p>@root contains the OID of the identification scheme, @extension contains the identification number according to that identification scheme. Note: all other known Ids for the Patient (older, previously used, temporary Ids, non-preferential IDs) should be sent in the OtherIds.id attribute.</p> <p><i>Note: this element is not allowed/present in a PatientRegistry.Add request. The identity of the patient can't be assigned by the requesting party.</i></p>	
1	addr	@use
	One or more addresses of the patient. Contains all known addresses, except for the address as registered in the Folkeregister (that address is conveyed as part of the Person class). See section 5.1.2 for a description of the sub-elements of addr.	
1	telecom	@use, @value
	One or more telecom details (e.g. phone, fax, e-mail) of the patient. See section 5.1.3 for a description of the sub-elements of telecom.	
1	statusCode	@code
	<p>@code contains the fixed value ‘active’. This element is not present in all variations of this model: it is not present/allowed in PatientRegistry.UpdateRecord (PRPA_IN201302NO).</p>	
1	patientPerson	
	<i>Root element of the generic person model described in section 7.1.1.</i>	
1	providerOrganization	@classCode, @determinerCode
	Identifies the organization that is aware of the patient identifiers as contained in the Registry, e.g. Helse Vest.	
2	Id	@root, @extension
	<p>@root contains 2.16.578.1.12.4.1.4.101 when organization number is used to identify the organization.</p> <p>@extension contains the actual organization number.</p>	
3	contactParty	@classCode, @nullFlavor
	@classCode contains the fixed value ‘CON’. @nullFlavor contains the fixed value ‘NA’.	
1	subjectOf1 (in Patient Registry Find Candidates Query Response interactions ONLY)	

	Identifies the degree-of-match between this response records and the parameters as sent in the query.	
2	queryMatchObservation	@classCode, @moodCode
	@classCode contains the fixed value ‘OBS’. @moodCode contains the fixed value ‘EVN’.	
3	Code	@code, @codeSystem
	@codeSystem contains the fixed value ‘2.16.578.1.34.5.2’. @code contains the fixed value ‘PERC’. This indicates that the percentage of match is sent.	
3	Value	@xsi:type, @value
	The Person registry currently uses a fixed value of 80% for all records. @xsi:type contains the fixed value ‘REAL’. @value contains the fixed value ‘80’.	
1	subjectOf2 (in Patient Registry Find Candidates Query Response interactions) subjectOf (in all other interactions)	
	Root element of an observation that identifies the kommune/bydel associated with the patient. See 5.1.6 for details of the sub-elements and attributes.	

8. Acknowledgements and Error Handling

8.1. Accept Acknowledgement

8.1.1. Introduction

An Accept Acknowledgement (the MCCL_IN000002 interaction) is sent in response to the receipt of an interaction to communicate the fact that the interaction has been received; that its contents successfully passed an initial validation step (see below), and that its contents were safely stored for further processing. The Accept Acknowledgement doesn't make a statement about the ability of the receiver to process the contents of the message.

The *initial validation process* is a fully automated process executed by the receiving software (mostly: the software component responsible for message processing) to validate the syntax of the interaction that was received. This may be limited to checking the interaction type; it may also include validation against a scheme, and it could extend to validation of codes used.

8.1.1.1. Structure of the Interaction

In terms of structure the Accept Acknowledgement interaction is defined as a [Transmission Wrapper](#). The interaction doesn't contain a ControlAct wrapper nor a payload.

Given the purpose of the Accept Acknowledgement interaction the following attributes have a key role in the interaction:

- Sender/device.id: should be equal to the Receiver/device.id of the interaction that this Accept Acknowledgement is in response of.
- Receiver/device.id: should be equal to the Sender/device.id of the interaction that this Accept Acknowledgement is in response of.
- AcceptAckCode: fixed to NE (Never) - Accept Acknowledgement interactions themselves are never to be responded to.
- Acknowledgement.typeCode (mandatory): should be either CA (accept) or CE (error).
- TargetMessage.id (mandatory): contains the ID of the interaction that this Accept Acknowledgement is in response of.
- The AcknowledgementDetail class may occur zero or more times:
 - AcknowledgementDetail.typeCode: contains either E (Error), W (Warning), or I (Informational). See [Error Handling](#) for additional details.
 - AcknowledgementDetail.code: contains a code that uniquely identifies the error/warning/information item. See [Error Handling](#) for additional details.

8.1.1.2. When should an Accept Acknowledgement be sent?

A receiver should send an Accept Acknowledgement interaction if the sender requested one to be sent in the Message.acceptAckCode of the initial interaction. If the value of that

attribute is unequal to NE (Never) an Accept Acknowledgement may have to be sent. Other values for Message.acceptAckCode are: AL (Always), and ER (only in case of errors).

In the context of the Norwegian infrastructure the value of Message.acceptAckCode will always be either AL (Always) or NE (Never), and the value will be implied by the WSDL used for the service. Therefore, the receiving application will always be aware if an Accept Acknowledgement has to be sent simply because of the [WSDL](#) definition.

8.1.2. Examples

- Accept Acknowledgement - interaction 0806071541103 is accepted (CA) without errors or warnings

```
<MCCI_IN000002UV01 ITSType="XML_1.0" xmlns="urn:hl7-org:v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v3">
  <id extension="34236" root="2.16.578.1.34.1.871.3"/>
  <creationTime value="20080607154128"/>
  <versionCode code="NE2008"/>
  <interactionId extension="MCCI_IN000002UV01" root="2.16.840.1.113883.1.6"/>
  <processingCode code="P"/>
  <processingModeCode code="T"/>
  <acceptAckCode code="NE"/>
  <receiver typeCode="RCV">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="408" root="2.16.578.1.34.1"/>
    </device>
  </receiver>
  <sender typeCode="SND">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id extension="871" root="2.16.578.1.34.1"/>
    </device>
  </sender>
  <acknowledgement>
    <typeCode code="CA"/>
    <targetMessage>
      <id extension="0806071541103" root="2.16.578.1.34.1.408.7"/>
    </targetMessage>
  </acknowledgement>
</MCCI_IN000002UV01>
```

- Accept Acknowledgement - interaction 0806071541103 is rejected (CE) with errors (NS200 code)

```
<MCCI_IN000002UV01 ITSType="XML_1.0" xmlns="urn:hl7-org:v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v3">
  <id extension="34236" root="2.16.578.1.34.1.871.3"/>
  <creationTime value="20080607154128"/>
```

```
<versionCode code="NE2008"/>
<interactionId extension="MCCI_IN000002UV01" root="2.16.840.1.113883.1.6"/>
<processingCode code="P"/>
<processingModeCode code="T"/>
<acceptAckCode code="NE"/>
<receiver typeCode="RCV">
  <device classCode="DEV" determinerCode="INSTANCE">
    <id extension="408" root="2.16.578.1.34.1"/>
  </device>
</receiver>
<sender typeCode="SND">
  <device classCode="DEV" determinerCode="INSTANCE">
    <id extension="871" root="2.16.578.1.34.1"/>
  </device>
</sender>
<acknowledgement>
  <typeCode code="CE"/>
  <acknowledgementDetail typeCode="E">
    <code code="NS200" displayName="Unsupported InteractionID"
      codeSystem="2.16.840.1.113883.5.1100"/>
  </acknowledgementDetail>
  <targetMessage>
    <id extension="0806071541103" root="2.16.578.1.34.1.408.7"/>
  </targetMessage>
</acknowledgement>
</MCCI_IN000002UV01>
```

8.2. Error Handling

8.2.1. Overview

Errors, warnings and informative messages can be identified in multiple places; the use of the various options mainly depends on the type of issue one wants to identify.

1. **HTTP errors/SOAP faults:** optional; errors related to the erroneous use of the HTTP and SOAP protocols
2. **Acknowledgement.TypeCode:** mandatory indication in all responses if the message being responded to could be processed successfully or not.
3. **AcknowledgementDetail.code:** optional; to identify errors related to syntactical issues upon receipt of the HL7 v3 message.
4. **DetectedIssueEvent.code:** optional; to identify errors related to the violation of business-rules during the processing of the contents of the HL7 v3 message.
5. **QueryAck.queryResponseCode:** conditional (mandatory in query responses); to identify errors related to the processing of queries or the interpretation of query parameters.

Note that all errors/issues are identified using a code (with additional text). The ability to use a code to identify an issue allows the receiver of the message to take automatic action to alleviate the issue. A textual error message (without a code) is useless when it comes to software processing logic.

8.2.1.1. HTTP/SOAP

- Error category: errors related to the use of HTTP or SOAP protocols.
- Error location: HTTP response, or SOAP faults

This implementation guide is silent about the use of HTTP errors. The use of the recommendations provided in the WS-I Basic Profile may be advisable.

The SOAP Body contains a fault element that can be used to convey errors at the SOAP level. This implementation guide is silent about the use of SOAP faults. The use of the recommendations provided in the WS-I Basic Profile may be advisable.

8.2.1.2. Acknowledgement.TypeCode

- Error category: ability to process HL7 version 3 message
- Error location: the TypeCode attribute in the Acknowledgement class of the [Transmission Wrapper](#). TypeCode must be specified in all responses.

The most important attribute when it comes to the identification of errors in a response interaction is the Acknowledgement.typeCode (see [Transmission Wrapper](#) for a description of the attribute). The value of this attribute indicates whether or not the original interaction (the one being responded to) could be successfully processed or not.

typeCode indicates either success (the message could be processed), or failure (unable to process the message). Mainly for historic reasons there are four codes, two of which may ONLY be used in the [Accept Acknowledgement](#) interaction; the other two codes (AA, AE) are used in all other response interactions. AA and AE may NOT be used in [Accept Acknowledgements](#).

Error/Success	MCII_IN000002UV02 Accept Acknowledgement	Other response interactions Application Acknowledgement
Success	CA	AA
Error	CE	AE

Notes:

1. Regardless of the value of the HL7 v3 typeCode the HTTP response SHALL be '200 OK'.
2. In addition to the typeCode one may provide one or more AcknowledgementDetail and/or DetectedIssue classes (see below).
3. In query responses one has to provide an additional queryResponseCode next to typeCode.

Examples

1. Accept Acknowledgement, with a CA typeCode

```
<MCCI_IN000002UV01 ITSType="XML_1.0" xmlns="urn:hl7-org:v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hl7-org:v3">
  <id extension="34236" root="2.16.578.1.34.1.871.3"/>
  ...
  <acknowledgement>
    <typeCode code="CA"/>
    <targetMessage>
      <id extension="0806071541103" root="2.16.578.1.34.1.408.7"/>
    </targetMessage>
  </acknowledgement>
```

2. Application Acknowledgement, with an AE typeCode

```
<acknowledgement>
  <typeCode code="AE"/>
  <targetMessage>
    <id extension="0806071541103" root="2.16.578.1.34.1.408.7"/>
  </targetMessage>
</acknowledgement>
```

8.2.1.3. AcknowledgementDetail.code

- Error category: errors and warnings related to syntactical issues upon receipt of the HL7 v3 message.
- Error location: the AcknowledgementDetail.code attribute in the [Transmission Wrapper](#). This attribute is present in all response messages.

The AcknowledgementDetail class is a repeating construct in the [Transmission Wrapper](#). Each class contains multiple attributes (typeCode, code, text) to identify the details of the error/warning. See [Transmission Wrapper](#) for documentation of the class structure. The key attribute is code. The standard coding system used is AcknowledgementDetailCode (shown below).

The *AcknowledgementDetailCode* coding system (with OID 2.16.840.1.113883.5.1100) contains (amongst other things) the following values:

Code	Description	Comments
SYN100	Required class missing	Error: Required class missing in message; or the sequence of the classes is different than required by the standard or one of the conformance profiles identified in the message.
Error	CE	AE
SYN101	Required attribute missing	Error: A required attribute is missing in a class

SYN102	Data type error	Error: The attribute contained data of the wrong data type, e.g. a numeric attribute contained "FOO".
SYN103	Value not found in code system	Error: An attribute value was compared against the corresponding code system, and no match was found. This error code is also used if a Realm-specific vocabulary has been selected by means of the RealmCode attribute.
SYN104	Invalid code system in CNE	An attribute value referenced a code system that is not valid for an attribute constrained to CNE.
SYN110	Number of class repetitions exceeds limit	Error: the number of repetitions of a (group of) class(es) exceeds the limits of the standard or one of the conformance profiles identified in the message.
SYN112	Number of attribute repetitions exceeds limit	Error: the number of repetitions of an attribute exceeds the limits of the standard or one of the conformance profiles identified in the message.
NS200	Unsupported interaction	Rejection: The interaction (or: this version of the interaction) is not supported.
NS202	Unsupported processing id	Rejection: The Processing ID is not supported.
NS250	Unsupported processing Mode	Rejection: The processing mode is not supported.
NS203	Unsupported version id	Rejection: The Version ID is not supported.
NS260	Unknown sender	Rejection: the Device.id of the sender is unknown.
RTUDEST	Message routing error, unknown destination.	The destination of this message is unknown to the receiving application. The receiving application in the message does not match the application which received the message. The message was neither routed, processed nor stored by the receiving application.
NOSTORE	No storage space for message.	Rejection: The message can't be stored by the receiver due to an unspecified internal application issue. The message was neither processed nor stored by the receiving application.

Examples

1. Error (AE), SYN102, with error text "Fødselsdato skal kun inneholde numre".

```
<acknowledgement>
  <typeCode code="AE"/>
  <acknowledgementDetail typeCode="E">
    <code code="SYN102" displayName="Datatype Error"
      codeSystem="2.16.840.1.113883.5.1100"/>
    <text xsi:type="ST">Fødselsdato skal kun inneholde numre</text>
```

```

</acknowledgementDetail>
<targetMessage>
  <id extension="0806071541103" root="2.16.578.1.34.1.408.7"/>
</targetMessage>
</acknowledgement>

```

2. Error (AE), Error (E) SYN101 (required attribute missing, see table) and Warning (W) SYN103 (unknown code).

```

<acknowledgement>
  <typeCode code="AE"/>
  <acknowledgementDetail typeCode="E">
    <code code="SYN101" displayName="Datatype Error"
      codeSystem="2.16.840.1.113883.5.1100"/>
  </acknowledgementDetail>
  <acknowledgementDetail typeCode="W">
    <code code="SYN103" displayName="Datatype Error"
      codeSystem="2.16.840.1.113883.5.1100"/>
  </acknowledgementDetail>
  <targetMessage>
    <id extension="0806071541103" root="2.16.578.1.34.1.408.7"/>
  </targetMessage>
</acknowledgement>

```

8.2.1.4. DetectedIssueEvent.code

- Error category: errors related to the violation of business-rules during the processing of the contents of the HL7 v3 message
- Error location: the DetectedIssueEvent.code attribute in the [ControlAct Wrapper](#). This attribute is present in all responses except the [Accept Acknowledgement](#).

The reasonOf element can be used in the response interactions to give a reason why a request is rejected.

The OID of the (Helse Vest defined) coding system is 2.16.578.1.34.5.3. So far, the following error codes have been defined:

Code	Description
KNONPAT	The patient is already known in the patient registry.
VALIDATION	The query parameters are wrong or insufficient.
AUTHENTICATION	The user cannot be authenticated.
AUTHORIZATION	The user is not authorized.

Notes:

1. HL7 also has a coding system for errors in this category; the coding system isn't used at the moment.

2. Helse vest also included a code called "OTHER" for "other errors. The attribute displayName may contain a useful error message.". This is not a proper use of the data type and is therefore discouraged. The proper way would be as follows:

```
<code nullFlavor="OTH"><originalText>error description</originalText></code>
```

Examples

1. AddNewPatient service. Reason for the reject of the message is that the patient already exists.

```
<reasonOf typeCode="RSON">
  <detectedIssueEvent moodCode="EVN" classCode="ALRT">
    <code code="KNOWNPAT" codeSystem="2.16.578.1.34.5.3"
      displayName="Patient already known. This service requires a new patient." />
  </detectedIssueEvent>
</reasonOf>
```

2. Wrong query parameters; unable to execute the query

```
<reasonOf typeCode="RSON">
  <detectedIssueEvent moodCode="EVN" classCode="ALRT">
    <code code="VALIDATION" codeSystem="2.16.578.1.34.5.3"
      displayName="insufficient parameters in query" />
  </detectedIssueEvent>
</reasonOf>
```

8.2.1.5. QueryAck.queryResponseCode

- Error category: errors related to the processing of queries or the interpretation of query parameters
- Error location: queryResponseCode attribute of the QueryAck class in the [ControlAct Wrapper](#). The use of this attribute is limited to responses to Queries.

Examples of error in this category include missing parameters in a query, wrong parameter format (e.g. have a letter where there should be a number), or the use of an illegal F-/D-number format. In case of this category of parameter related errors:

- QueryAck.queryResponseCode shall be set to QE (see [ControlAct Wrapper](#) for a description of this attribute), and AcknowledgementDetail.typeCode shall be set to AE (see [Transmission Wrapper](#) for a description of this attribute).

Examples

1. The query could be successfully processed (OK).

```
<queryAck>
  <queryId extension="80622193108_11" root="2.16.578.1.34.1.805.1"/>
  <queryResponseCode code="OK"/>
  <resultCurrentQuantity value="2"/>
  <resultRemainingQuantity value="0"/>
</queryAck>
```

2. The query could not be successfully processed (QE - Query Error).

```
<queryAck>
  <queryId extension="80622193108_11" root="2.16.578.1.34.1.805.1"/>
```

```
<queryResponseCode code="QE"/>
<resultCurrentQuantity value="0"/>
<resultRemainingQuantity value="0"/>
</queryAck>
```

8.3. WSDL and Schemas

WSDL documents are XML documents that describe services as a set of message-enabled or procedure-oriented abstract endpoints. These operations and/or messages and their associated data types are described conceptually, and then bound concretely to a network protocol, message format, and programming language as needed.

A WSDL document defines the following elements for describing services:

- **Types:** data type definitions
- **Message:** an abstract definition of the data being transferred
- **Operation:** an abstract description of a service procedure
- **Port Type:** an abstract set of operations supported by one or more endpoints
- **Binding:** a concrete protocol and data format for a given port type
- **Port:** a single endpoint defined as a binding and a network address
- **Service:** a collection of related endpoints or ports

This document is published jointly with a set of normative WSDL and Schemas. Together they are an expression of the contract on the 'wire level'. The WSDL provided in NE2008 v2 are strongly typed. That term refers to a service contract that contains a complete definition of its input and output messages in XML Schema, a schema that is either included in the WSDL definition or referred to by that WSDL definition.

The actual WSDL is specific as to the HL7v3 payload message contained in the <soap:Body> definition, with references to the actual schemas involved shown in listing below.

```
<types>
  <xsd:schema targetNamespace="urn:hl7-org:v3">
    <xsd:include schemaLocation="../schemas/QUPC_IN043100NO.xsd" />
    <xsd:include schemaLocation="../schemas/QUPC_IN043200NO.xsd" />
    <xsd:element name="QUPC_IN043100NO-Response">
      <xsd:complexType>
        <xsd:choice>
          <xsd:element ref="hl7:QUPC_IN043200NO" />
        </xsd:choice>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
</types>
<message name="QUPC_IN043100NO">
  <part name="body" element="hl7:QUPC_IN043100NO" />
```

```
</message>
<message name="QUPC_IN043100NO-Response">
    <part name="body" element="hl7:QUPC_IN043100NO-Response" />
</message>
<portType name="CareRecordQueryFulfiller_PortType">
    <operation name="QUPC_IN043100NO_Operation">
        <input message="hl7:QUPC_IN043100NO" />
        <output message="hl7:QUPC_IN043100NO-Response" />
    </operation>
</portType>
```

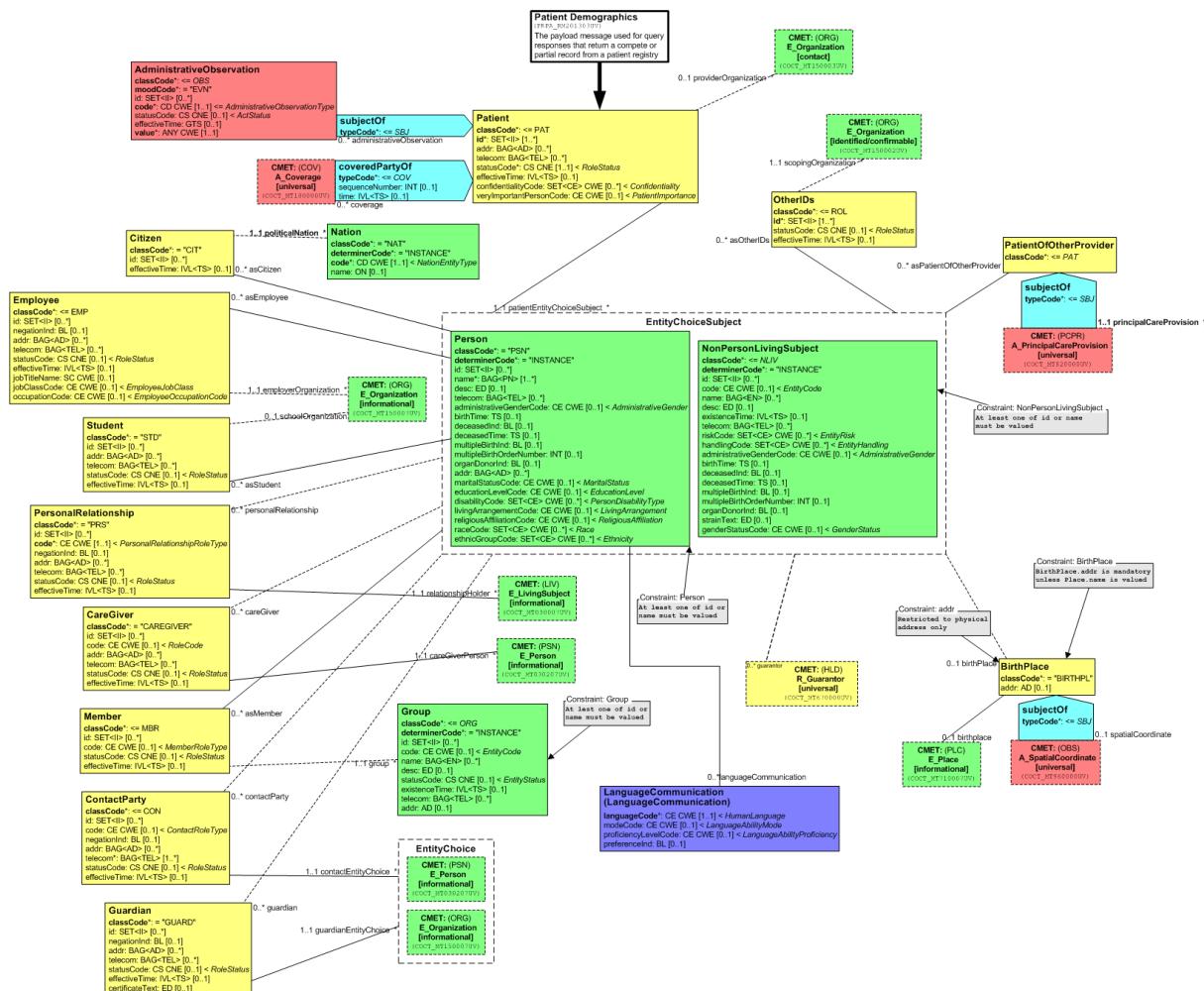
The published style of WSDL (“strongly typed”) describes the actual XML instances going in and out as well as the HL7v3 Schema's used. The WSDL can therefore easily be used to validate a service or guide in the design of a service. This generally increases the level of automation, code generation, tool support, and use of standardized middleware. It also produces more stable code and relieves the developer from having to create infrastructure level code. And even a generically implemented service can provide a strongly typed interface definition.

This 'contractual' flavour of WSDL and Schema is the only flavour that will be published, they are normative and the web services they describe should be implemented. Whether they are used 'as is' for code generation is an independent, often environment-dependent question, and vendors are free to use their own **derived** WSDL and (modified) schemas as long as the contractual WSDL is warranted.

9. Mapping to other models

This section contains a mapping of the data provided by Folkeregisteret and HL7 version 3 models.

The model below is the payload model used (with minor variations) in most interactions related to patient demographics.



Figur 31 Patient demographics

9.1. Folkeregisteret

The Norwegian Folkeregisteret (see

<http://www.skatteetaten.no/Templates/Artikel.aspx?id=6640&epslanguage=NO> for additional information) is a person registry.

Folkeregistervariablet knyttet til personer	HL7 version 3 equivalent	
fødselsnummer	Patient.id and Person.id (root OID indicates F-number)	OID=2.16.578.1.12.4.1.4.1
status (bosatt, utvandret, død osv.)	If status = 5: Person.deceasedInd = true	
Statusdato	If status = 5: Person.deceasedTime	
	Person.birthDate	6 first digits of F-number
	Person.administrativeGender	If digit 9 of F-number is even: Female, else male
fødested	Birthplace.addr / city	
etternavn	Person.name / family ; use = OR	
Fornavn	Person.name / given ; use = OR	
Mellomnavn	Person.name / middle ; use = OR	
etternavn som ugift	Person.name / family ; use = birthname	
forkortet navn (brukes ved adressering av post)		
Statsborgerskap	Not used - (Citizen) Nation.code	
Familienummer		
personkode (referanseperson, ektefelle, barn),		
spesifisert registreringstype (diplomat, klient, adresesperring o.l.)		
sivilstand	Person.maritalStatus	
ektefelles fødselsnummer, navn og statsborgerskap		
kommunenummer		

flyttedato	Person.addr / usablePeriod?	Date of last move of address
nummerisk bostedsadresse	Person.addr; use = P; usablePeriod after <i>flyttedato</i>	
postadresse for forsendelse av post	Person.addr; use = HP; usablePeriod after <i>flyttedato</i>	
land innvandret fra	Person.addr / country ; usablePeriod before <i>flyttedato</i>	
land utvandret til	Person.addr / country ; usablePeriod after <i>flyttedato</i>	
stemmerett		
vergemål		
foreldreansvar		
arbeidstillatelse		
DUF-nummer (nummer i utlendingsmyndighetens register)		
mors fødselsnummer, navn og statsborgerskap	Not used - personalRelationship.E_LivingSubject	
fars fødselsnummer, navn og statsborgerskap	Not used - personalRelationship.E_LivingSubject	
fødselsnummer og navn for hvert barn	Not used - personalRelationship.E_LivingSubject	
medlemskap i Den norske kirke	Not used - Person.religiousAffiliationCode	
referanser til utgåtte fødselsnummer/D-nummer.	Person.id /Patient.id (note: previous/old ID)	

10. References

- [1] RFC 2806: <http://tools.ietf.org/html/rfc2806> and <http://www.ietf.org/rfc/rfc2806.txt>
URLs for Telephone Calls
- [2] The implementation guide developed by Helse Vest IKT:
http://hl7.ihelse.net/Dokument/HL7v3_ImplementationGuide_3.0c.doc
- [3] The HL7 Norway wiki: <http://www.hl7.no/hl7wiki>