❖ InterCare ❖

The Interworking and Interoperability of networked services for Healthcare using Internet-based technology

HC 4011

An 'umbrella' project of the

European Commission - DG XIII
Healthcare Telematics Applications Programme

incorporating

Cardlink 2, Hector, *Star*☆, Synapses and TrustHealth

D2.2 Demonstrator Design Specifications

Workpackage:	WP2
Editor:	Mats Gustafsson, Lars-Åke Johansson (SITI/SISU)
Dissemination:	RP
Document Number:	INTERCARE/WP2/SISU005
Version:	1.0
Date:	1999-01-15
Web address:	http://www.imsgrp.com/intercare/ims001.html

Abstract:

This document contains the design specifications of InterCare demonstrator applications. For each demonstrator, the business objectives, purpose of application and metrics to be used for evaluation are presented. The Unified Modeling Language (UML) has been used for the specifications work, and the documentation for each demonstrator includes the Use case and Logical views. Implementation plans and architectural considerations are included as well.

CONTENTS

1. Introduction

One of the goals of InterCare is to create generalised systems components, called products, and show how they can be used for applications development.

Some partners of InterCare will build reusable generalised system components, and other partners have the role of developing usable applications.

In this document the InterCare demonstrator applications are presented and specified. These are going to be developed within the InterCare project and they will play several roles:

1. To correspond to valid information needs for important healthcare activities in different European countries especially connected to communication-oriented applications.
2. To demonstrate how the different generalised components, referred to as "products", developed in the InterCare project can be used for application development, reflecting important needs.

This means that the applications are aimed to be relevant at the different demonstrator sites and that the applications are specified to their external properties.

The second role means that the applications shall show how the different products can be used as generalised components and how they can be called from the applications.

In the following the specifications of the applications are presented. Extensive analysis work has been performed at the different demonstrator sites in order to produce the specifications.

To create valid specifications is a question of communication between people in the organisation. It is important that the business actors themselves can be involved to define the specifications that will lead to the applications. The business itself and the user actors have to have a common understanding of both how work is performed today and how the processes can be formed in a better future business.

To perform the communication around business goals and information needs in an effective way, one needs effective instruments to support the process. One of the important instruments is modelling. For this purpose the InterCare project has chosen UML (Unified Modelling Language) as the modelling language to be used for specification purposes. UML is a modern approach recognised all around the world and it is often referred to as a standard modelling language.

UML is a set of modelling tools. One has to add methods to get the tools to be used in an effective analysis and synthesis process.

The specifications are constituted of several parts:

- Business objectives
- Use case model
- Class model
- Implementation plan.

Business objects are important to express so that it can be a meaningful relationship between where the business is heading and what external properties there will be for the information system. In this process, it is important to express in what situation the business is standing now and where it wants to go. Business goals can focus a wide range of aspects from offering new services for patients to using particular resources in a new way. These statements are very important to get the right information systems support and to meet the implicit and explicit expectations on information systems.

Use cases are often used to make business actors and users talk about their business about what is happening in particular routines and what can be made better. Are there new goals and how must we work differently.

The object class model is important to express which information the information system has to be able to represent. The object class models have to be rich in the sense that all properties and relations have to be defined. This type of model can then be a base for how the system works and how the data are stored.

An implementation plan is added to the specifications in order to show how the development work is planned before testing and integrating to the different products.

2. Dutch Demonstrator Design Specification

2.1 Business Objectives and Purpose of the Application

2.1.1 Objectives of the Project

The overall objectives of the InterCare project are to increase the quality of care and to improve the information, available to citizens, on the healthcare facilities in their region. For improving the quality of care, a virtual patient record will be created, that can be accessed by all authorised professionals. For supplying citizens with health-related information, a multi-media information-point will be created, that can be accessed by everyone.

At the Dutch site, the virtual patient record will be restricted to medication data. There exists a lack of information about the use of medication by a patient. The GP is the gatekeeper in the Dutch healthcare system. This implies that he has an overall view of the medical treatment and medication of his patients. At the moment this is not the case. The medication-file is now stored in (at least) three places: the GP information system, the Hospital Pharmacy System, and the systems of the peripheral pharmacist(s). Pharmacists also need an overall view over the medication taken by a patient, in order to be able to localise possible contra-indications. This lack of information results in many time-consuming calls to and from the pharmacy, in the prescription of wrong or unnecessary medication, thus generating great costs and needless risks (drug interactions) for the patient. The quality of care will be enhanced by providing the health professionals access to the complete and up-to-date medication record. In the InterCare project we will facilitate this by creating a virtual patient record that consists of the combined medication records of the health care professionals involved. A major requirement is that access of medical record information is achieved in a secure and reliable way.

The Dutch demonstrator site has also the intention to develop the functionality of Extended Yellow Pages. These pages will contain information about the availability of care in the region, such as information about the waiting-lists of the hospital. This functionality is an extended part of the already existing Web-site of the hospital and should be seen as an Extranet. On this Extranet there will be also information available restricted only to Physicians, General Practitioners and other health professionals and institutions.

The Yellow pages will be accessible directly from the Hospital Web-site and also through a regional Yellow pages server, which is a product earlier developed by Zorg 2000 in the Delft region. Here the STAR-concept off a Regional Yellow Pages server is used. It will also be a medium, where medical professionals can exchange information in a secure environment, so a GP can communicate with a Physician in the hospital about certain patients and vice versa. Although this looks very much like the normal forum and chat functionality from the standard Internet, it is clearly that the aspect of security will have a very high priority.

2.1.2 Metrics

Because InterCare is a demonstration project, we will not have a great number of health professionals or patients that are going to participate. Therefore we do not expect great effects in the short run. However, we expect that prescriptions will be of better quality because more information is available when they are composed. They are also better readable in the pharmacy. In the pharmacy also more information is available. This reduces errors in dispensation. The efficiency of health professionals will increase, because less time has to be spent in gathering information.

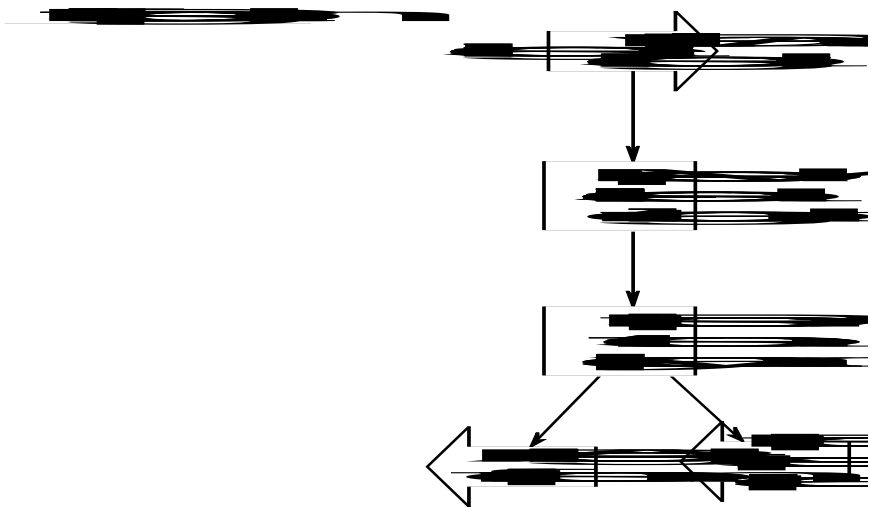
The yellow pages must add new services to the existing ones. Citizens can access information in a more flexible way, 24 hours a day. Information will be more up-to-date. We also expect that, for

institutions and health professionals, the effort for informing the public and updating the information will be reduced.

2.2 Use Cases and Scenarios

2.2.1 Business Processes and Scenarios

We start with the discharge of a patient from the hospital. This process is depicted in the following diagram.



First, the medical specialist (in our case the Cardiologist) prescribes the discharge medication (if any). Three main important parts of such a discharge medication are:

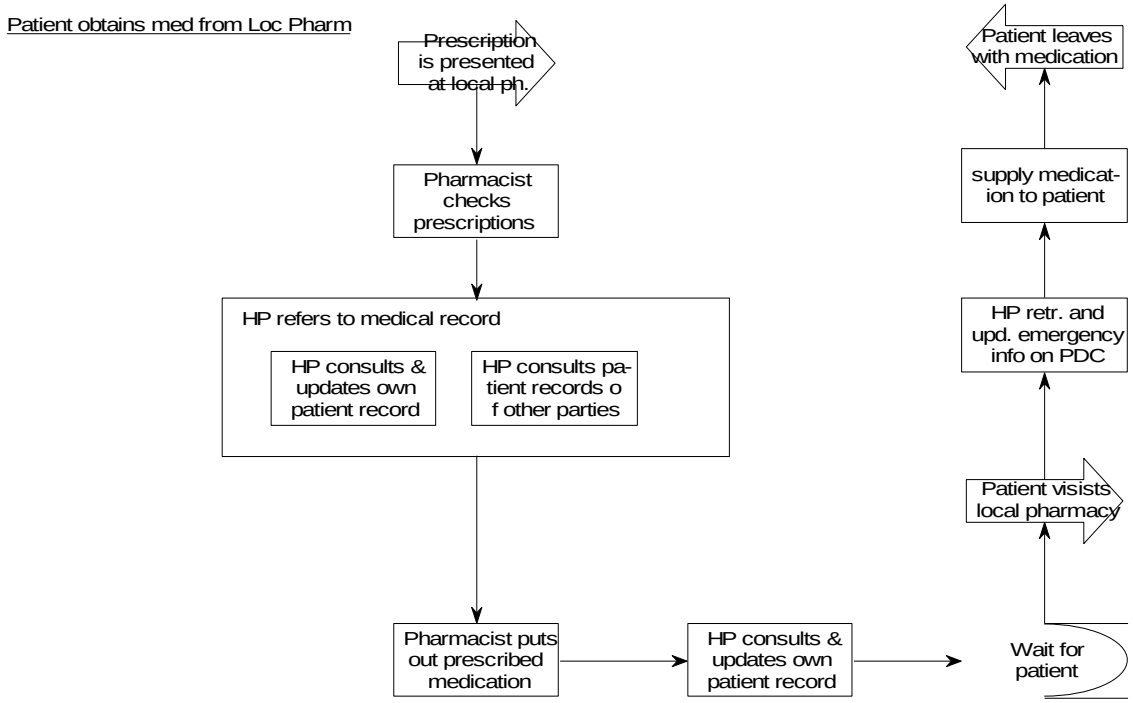
- The type of medication,
- Mode of use, and frequency, and
- Total quantity.

Then, the discharge prescription is sent to the local pharmacy. This either done by the outpatient clinic nurse, or by the ASP (Apothekers Service Point). Before the prescription can actually be sent, a number of things have to be done:

With the prescription it is indicated whether the medication must be delivered to the Patients home address, or whether the patient (or a relative) comes to pick the medication up at the pharmacy.

- The time of delivery is indicated.
- The validity of the prescription is checked (when it passes the ASP).
- The local pharmacy to send it to is selected.

When all this is handled the prescription is sent to the local pharmacy. The process taking place at the local pharmacy after sending is depicted in the figure below.

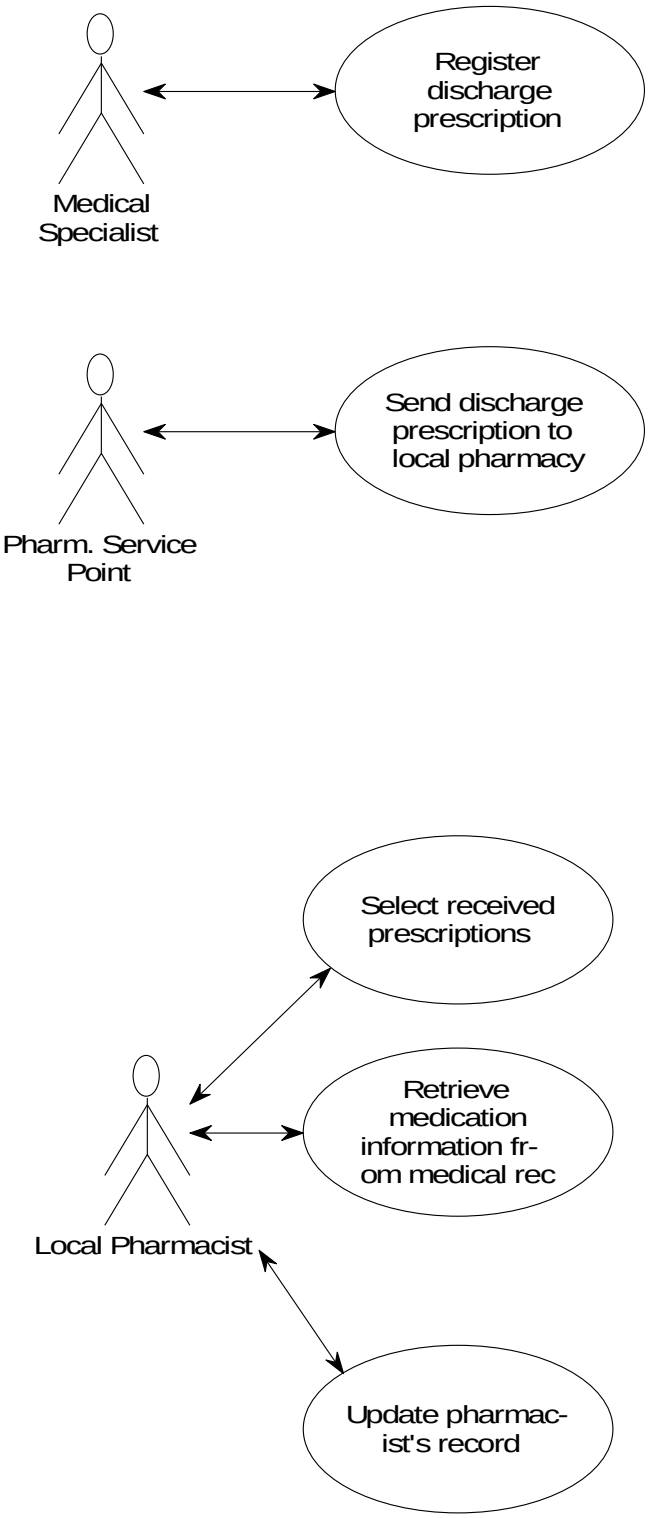


First, the pharmacist checks the prescriptions that have been sent. In order to decide whether certain medication can be supplied to a patient or not, the pharmacist can refer to the medical record of the patient (i.e. the current and historic medication as registered in the GP system, the Hospital Information System, and in his own systems or systems of other pharmacists. If everything is ok, the prescribed medication is put out for the patient. Then the pharmacist updates his own record system by adding the new prescription. Now, the medication is ready, and it can be sent to the patients home address (if required), or the visit of the patient is awaited. When the patient arrives, the medication is handed over to him.

2.2.2 Use Cases

Based on the presentation of the workflow descriptions in the previous section, we will distinguish five functions (use-cases) that must be delivered to the end-users in order to support their co-operative work. These five main functions are presented in this section by means of Use Case diagrams.

2.2.2.1 Use Case Diagram



2.2.2.2 Use Case Descriptions

Use Case: Register discharge prescription

Used by:

[Actors]

Medical Specialist

Description:

Registration and forwarding to Pharmaceutical Service Point.

Intent:

Registration of the discharge prescription by MS.

Patient identification.

MS identification (being the requester of the prescription).

Identification of the local pharmacy the prescription will be sent to.

Filling out one or more receipts (forms) and adding some free text remarks. Which together are the prescription.

Use Case: Send discharge prescription to local pharmacy

Used by:

[Actors]

Pharm. Service Point

Description:

Validation of the discharge prescription.

Adding dispensation info.

Sending prescription to local pharmacy.

Intent:

The prescription of the discharge medication of a patient must be send to a local pharmacy, where the patient will collect his medication.

Pre_Conditions:

HP has logged into the system.

Post_Conditions:

Patients prescription has been send to the local pharmacy.

Use Case: Select received prescriptions

Used by:

[Actors]

Local Pharmacist

Description:

TBD

Intent:

From a list of prescriptions a selection is made for further processing.

Pre_Conditions:

Prescriptions have been send to loc. pharmacy by third parties.

Post_Conditions:

Selected prescriptions.

Use Case: Retrieve medication information from medical record

Used by:

[Actors]

General Practitioner, Hospital Pharmacist, Local Pharmacist, Medical Specialist, Patient

Description:

Through InterCare, the information systems of local pharmacies and hospitals are accessed and current medication information is displayed.

- Login on InterCare-server with HPC and PIN-code.
- Identification of patient (if possible automatic on XIS)
- Notification on authorisation by security server
- Display FHCR of patient

Intent:

To inform HP about medication of the patient.

Pre_Conditions:

HP has question on medication of patient.

Post_Conditions:

HP's question about medication of patient is answered.

Use Case: Update pharmacist's record

Used by:

[Actors]

Local Pharmacist

Description:

After selection of description the pharmacist decides to load the prescription into his local database.

Intent:

Store obtained prescription in local pharmacist database.

Pre_Conditions:

Selected prescription.

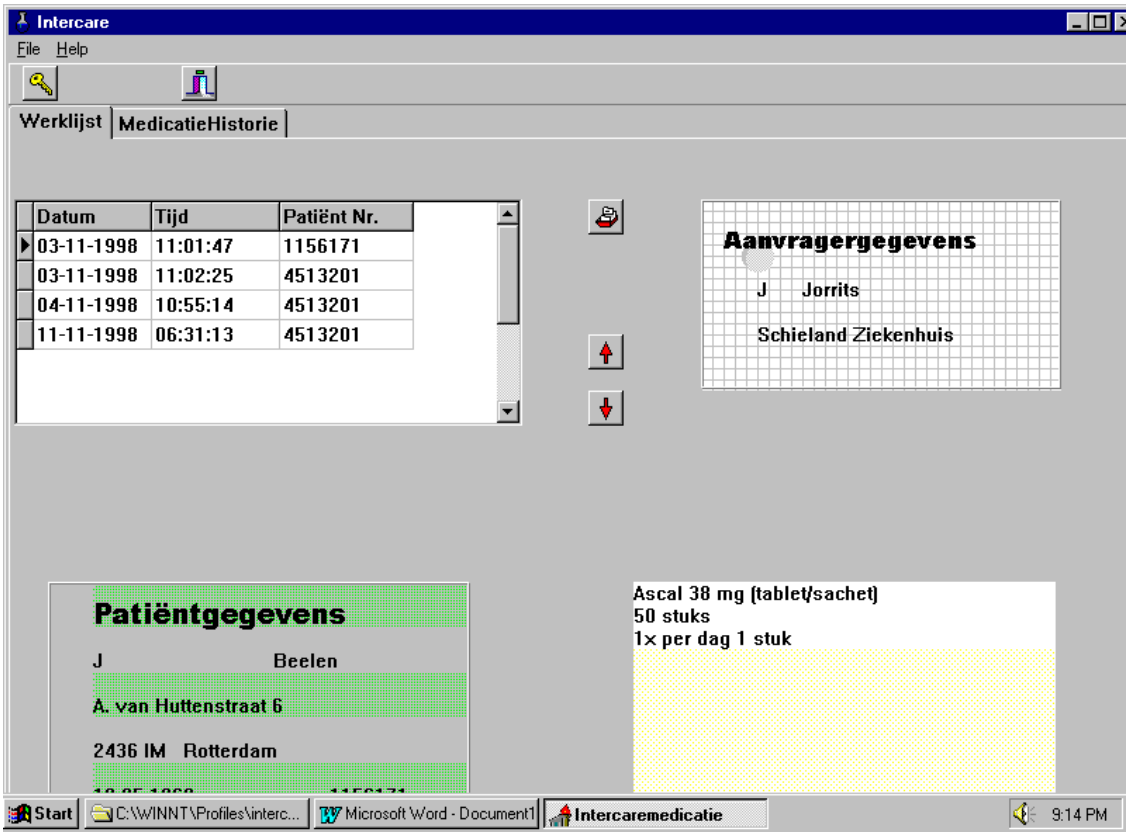
Post_Conditions:

Prescription in local database.

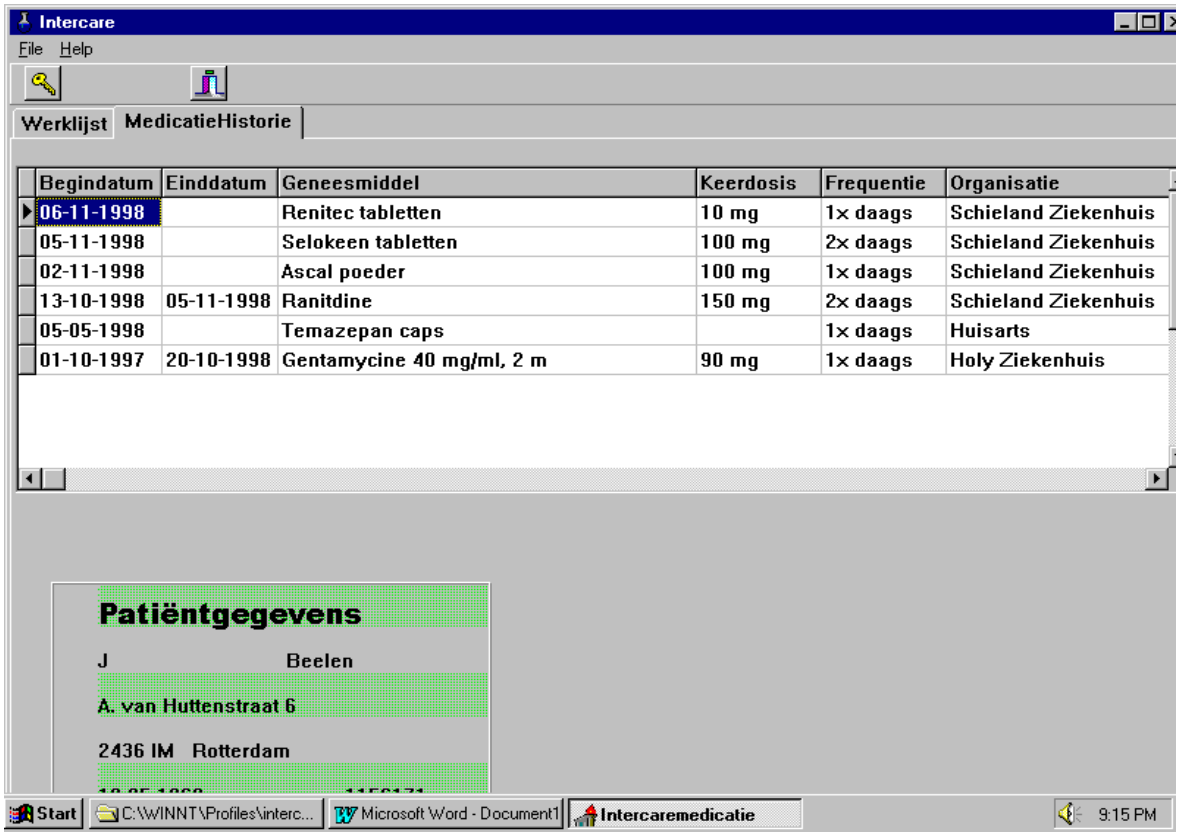
2.2.3 Prototypes of End-user Screens

Given these functions we will now provide two screen prototypes that illustrate how the end-users interact with the system to be developed.

The first screen shows how the list of prescriptions to be processed looks like for the Pharmacist. In the left upper corner the list is displayed. In the right upper corner some information on the prescribing health professional is displayed. In the left lower corner, some information on the patient belonging to the selected item is displayed. And finally, in the right lower corner, the prescription information of the selected item is shown.

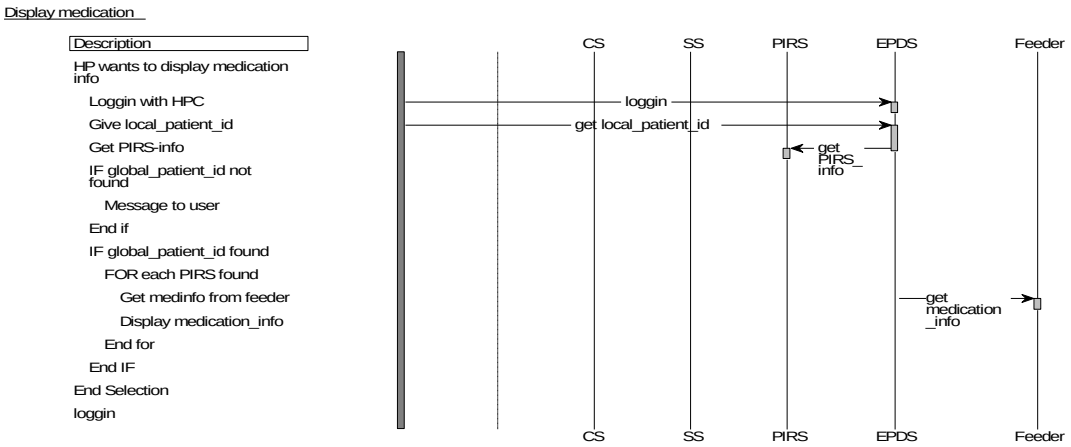


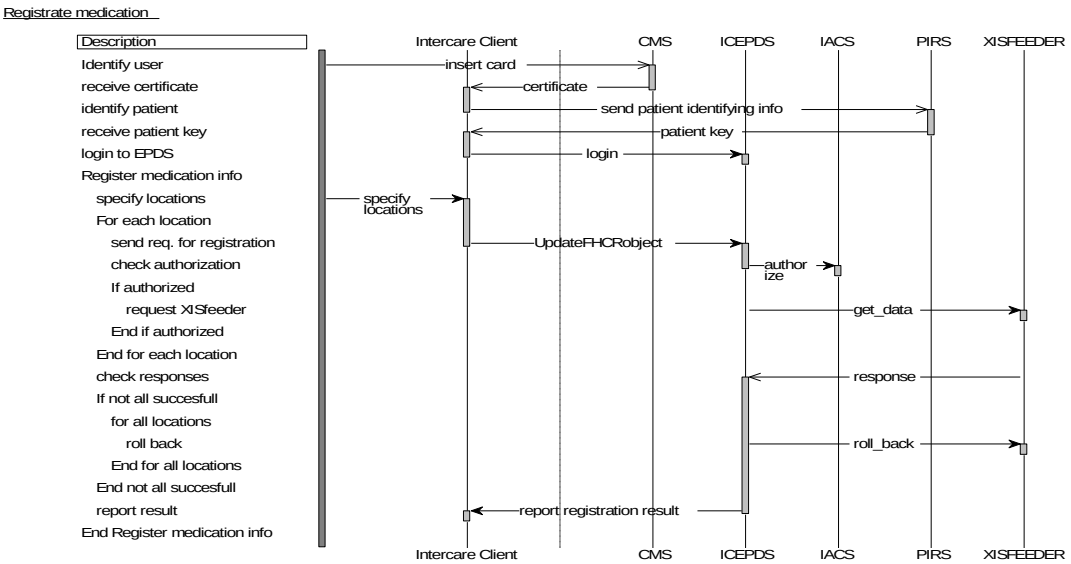
The second screen below shows how the Pharmacist will view the information retrieved from the medication record of a given patient, stored in the Hospital Information System. Each item in the list has a number of attributes relating to the medicinal product name, the way of use, and the frequency. Again, in the left lower corner some patient demographics are displayed.



2.2.4 Sequence Diagrams Showing Use of Common Products

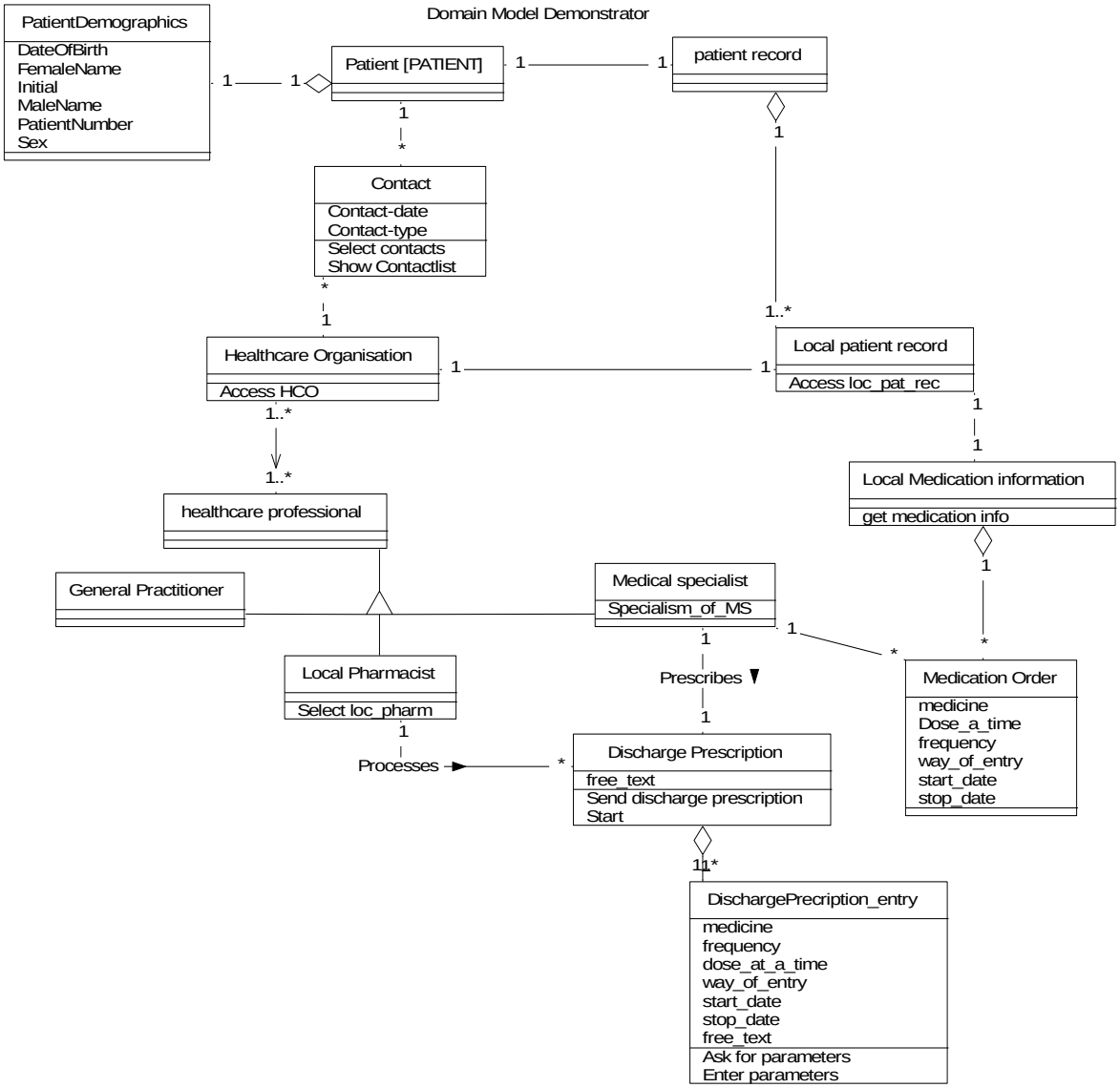
We now have seen the required functions, and a first impression of the screens with which the end-users interact with the InterCare System. In this section we will very briefly indicate how the InterCare Common Products are used in order to realise these functions. This is done by means of two Object Sequence Diagrams for two general functions that are part of the Use Cases discussed before: display medication, and register medication. It must be stressed that these Object Sequence Diagrams are not yet finalised, but they do illustrate the role of each common product.





2.3 Object Class Model

2.3.1 Class Diagram



2.3.2 Class Descriptions

Below we provide descriptions for each Class specified in the Class Diagram.

Class: Discharge Prescription

Description:

The discharge prescription that is prescribed by the medical specialist who has the patient under treatment in the hospital.

Attributes: [Public] ***free_text***

Operations: [Public] **Send discharge prescription** - void *Send discharge prescription ()*.

Start - void *Start ()*.

Association Statements

Each Discharge Prescription:

Is made up of one or more DischargePrecription_entry. Navigation bi-directional.

prescribed_to Is related to only one Patient. Navigation bi-directional.

Prescribes Is related to only one Medical specialist. Navigation bi-directional.

Send_to Is related to only one Local pharmacy. Navigation bi-directional.

Put_in Is related to only one Out-box. Navigation bi-directional.

Processes Is related to only one Local Pharmacist. Navigation bi-directional.

Is related to only one In-box. Navigation bi-directional.

Class: DischargePrecription_entry

Description:

A discharge prescription-entry is the prescription of exactly one medicine, together with frequency etc.

This will be related to medi-services already existing.

Attributes: [Public]

medicine

frequency

dose_at_a_time

way_of_entry

start_date

stop_date

free_text

Operations: [Public]

Ask for parameters - void *Ask for parameters ()*.

Enter parameters - void *Enter parameters ()*.

Association Statements

Each DischargePrecription_entry:

Is part of only one Discharge Prescription. Navigation bi-directional.

Class: Patient

Description:

Person related to healthcare organisations in a consumer way.

Association Statements

Each Patient:

Inherits from FolderRIC

Is made up of only one PatientDemographics. Navigation bi-directional.

Is made up of zero or more Insurance. Navigation bi-directional.

Is related to zero or more patient index. Navigation bi-directional.

precsribed_to Is related to only one Discharge Prescription. Navigation bi-directional.

Is related to only one patient record. Navigation bi-directional.

Is related to zero or more Contact. Navigation bi-directional.

Class: Medical specialist

Description:

Healthcare professional working as a medical specialist. E.g. cardiologist.

Attributes: [Public] ***Specialism_of_MS***

Association Statements

Each Medical specialist:

Inherits from healthcare professional

Prescribes Is related to only one Discharge Prescription. Navigation bi-directional.

has Is related to only one Out-box. Navigation bi-directional.

Is related to zero or more Medication Order. Navigation bi-directional.

Class: Local Pharmacist

Description: Pharmacist whose pharmacy is outside the hospital.

Operations: [Public] **Select loc_pharm** - void *Select loc_pharm ()*.

Association Statements

Each Local Pharmacist:

Inherits from healthcare professional

Is related to only one In-box. Navigation bi-directional.

Processes Is related to zero or more Discharge Prescription. Navigation bi-directional.

Class: Contact

Description:

A patient has none or more contacts with a particular healthcare organisation over time. Each has a definite startdate and a definite healthcare professional is involved.

Attributes: [Public]

Contact-date

Contact-type

Operations: [Public]

Select contacts - void *Select contacts ()*.

Show Contactlist - void *Show Contactlist ()*.

Association Statements

Each Contact:

Is part of only one patient index. Navigation bi-directional.

Is related to only one Healthcare Organisation. Navigation bi-directional.

Is related to only one Patient. Navigation bi-directional.

Class: PatientDemographics

Description:

Used for identification of a patient

Attributes: [Public]

DateOfBirth

FemaleName

Initial

MaleName

PatientNumber

Sex

Association Statements

Each PatientDemographics:

Is part of only one Patient. Navigation bi-directional.

Class: Healthcare Organisation

Description:

HCO. Referenceobject.

Constraints:

The HCO of a GP is not defined?

Operations: [Public] **Access HCO** - void *Access HCO ()*.

Association Statements

Each Healthcare Organisation:

Is related to zero or more HCO_of_HP. Navigation bi-directional.

Is related to zero or more Contact. Navigation bi-directional.

Is related to only one Local patient record. Navigation bi-directional.

Works at Is related to one or more healthcare professional. Navigated from Healthcare Organisation to healthcare professional.

Class: healthcare professional

Description:

Pragmatically defined as any person who holds a Health Professional Card. Medical Specialists, General Practitioner and Pharmacist are Healthcare Professionals.

Association Statements

Each healthcare professional:

has Is related to zero or more authorities. Navigation bi-directional.

Performes at Is related to zero or more HCO_of_HP. Navigation bi-directional.

Is related to only one health professional card. Navigation bi-directional.

Works at Is related to one or more Healthcare Organisation. Navigated from Healthcare Organisation to healthcare professional.

Is related to zero or more Medication Order. Navigation bi-directional.

Class: patient record

Description:

The virtual patient record as integrated from the various existing local patient records.

Association Statements

Each patient record:

Is part of only one Local Medication information. Navigation bi-directional.

Is made up of only one Local Medication information. Navigation bi-directional.

Is made up of only one emergency information. Navigation bi-directional.

Is related to only one Patient. Navigation bi-directional.

Is related to only one Patient. Navigation bi-directional.

Is made up of one or more Local patient record. Navigation bi-directional.

Class: Local patient record

Description:

Patient record as existing within a particular healthcare organisation.

Operations:

[Public] **Access loc_pat_rec** - void *Access loc_pat_rec ()*.

Association Statements

Each Local patient record:

Is related to only one Healthcare Organisation. Navigation bi-directional.

Is part of only one patient record. Navigation bi-directional.

Is related to only one Local Medication information. Navigation bi-directional.

Class: Local Pharmacist

Description:

Pharmacist whose pharmacy is outside the hospital.

Operations: [Public] **Select loc_pharm** - void *Select loc_pharm ()*.

Association Statements

Each Local Pharmacist:

Inherits from healthcare professional

Is related to only one In-box. Navigation bi-directional.

Processes Is related to zero or more Discharge Prescription. Navigation bi-directional.

Class: Local Medication information

Description:

Local medication information as existing in a particular organisation as part of the local patient record.

Operations: [Public] **get medication info** - void *get medication info ()*.

Association Statements

Each Local Medication information:

Is made up of only one patient record. Navigation bi-directional.

Is part of only one patient record. Navigation bi-directional.

Is made up of zero or more Medication Order. Navigation bi-directional.

Is related to only one Local patient record. Navigation bi-directional.

2.4 Implementation Plan

If we look at the two scenarios described in the previous section, we can derive the following set of required functions.

- ***Register discharge prescription*** – used by the **Cardiologist**
- ***Send discharge prescription to local pharmacist*** – used by the **ASP**
- ***Select received prescriptions*** - used by the **Local Pharmacist**
- ***Retrieve medication information from medical record*** – used by the **Local Pharmacist**
- ***Update pharmacist's record*** - used by the **Local Pharmacist**

The implementation will take place in an iterative way. We distinguish between three iterations.

This first iteration encompasses:

- ***Retrieve medication information from medical record*** – used by the **Local Pharmacist** (at least in this scenario)

The main objective of this iteration is to offer the local Pharmacist the opportunity to view medication information from the medical record of a given patient, as stored in the Hospital Information System of the Schieland Hospital.

The second iteration encompasses:

- ***Register discharge prescription*** – used by the **Cardiologist**
- ***Send discharge prescription to local pharmacist*** – used by the **ASP**
- ***Select received prescriptions*** - used by the **Local Pharmacist**

The main objective of this iteration is to offer the two end-users in the hospital the opportunity to register and send discharge prescriptions. Furthermore, the local pharmacist must be able to select one of the received prescriptions for further processing.

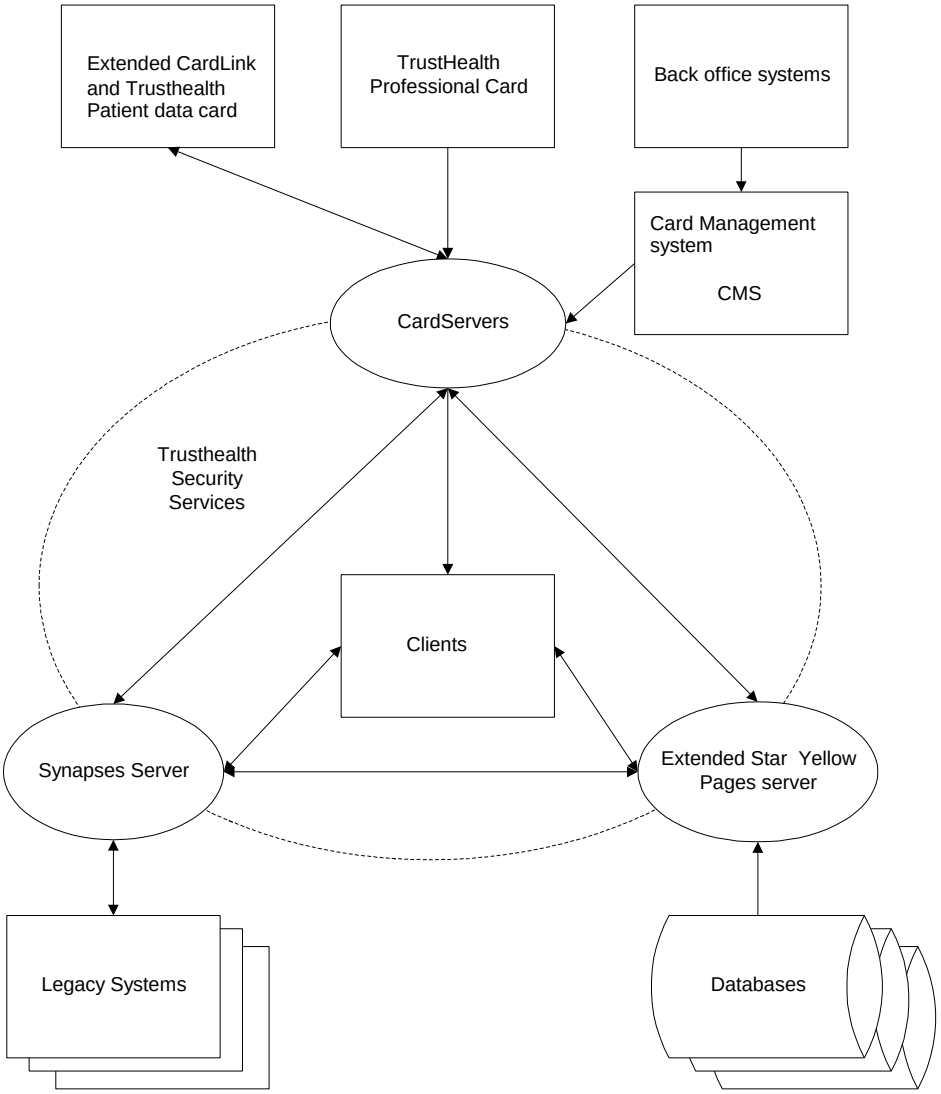
This iteration encompasses:

- ***Retrieve medication information from medical record*** – used by the **Local Pharmacist** (at least in this scenario)
- ***Update pharmacist's record*** - used by the **Local Pharmacist**

The main objective of this cycle is to extend the medical record information retrieve functionality. It must now also be possible to retrieve information from the GP system, and other Pharmacy systems. So, this must be a further step in the direction of the realisation of a virtual distributed record. Furthermore, in this iteration a connection is realised between the InterCare client receiving discharge prescriptions and the Pharmacy system, enabling insertion of the prescription information into the pharmacy system.

2.5 Architectural Aspects

The figure below shows the target system-architecture of the InterCare demonstrator in Schiedam. Migration towards this architecture will proceed according to the implementation plan outlined in the previous section.



3. Finnish Demonstrator Design Specification

3.1 Business objectives

The Finnish InterCare project will demonstrate a regional information systems architecture which is compatible with the current and near future regulations as defined in the Finnish data security legislation. The architecture will make possible the use of the smartcards as a means to improve systems security, patient and user identification and information storage and transferring. It will support regional seamless care health services using the star* services, Cardlink2 healthcards and TrustHealth security services.

Objectives of the the InterCare project applications are:

- To enable healthcare units to send electronic (edifact etc.) referrals to other units and to receive electronic referrals from other units
- To enable healthcare units to send epicrisis (feedback) in electronic format to other units/physicians
- To enable physicians to browse medical record information in other healthcare units based on patients permission
- To improve and enable dialog between healthcare professionals in electronic format
- To use strong identification of the patient and the professional (smartcards)
- To utilise secure web-technology
- To improve data security and access control
- To create regional patient reference database and management systems
- To create regional healthcare information services (yellow pages)
- To create possibilities to implement regional booking services
- To enable citizen interactively to participate in his/her care and services

3.2 Use Cases

3.2.1 Non-scheduled visit to a doctor

	Actor action		**System response**
1.	Patient A arrives into the healthcare unit's reception desk		
2.	Patient A is identified	3.	Patient's data is written into the legacy system
4.	Physician K decides to search for the patient's data stored in other healthcare units.	5.	Physician K is strongly identified. Physician K makes a query about the patient A for to the regional PRM and gets as an answer N references sorted according to the date. The references points in the healthcare actions (episodes of care, outpatient visits and booked appointments) done to patient in different healthcare units.
6.	Physician K decides to look in details the data from A's last episode of care in healthcare unit X		
7.	Physician K declares, what is her/his relationship (role) with the patient A	8.	The system checks the announced role against the permission profile (a part of the reference). The system sends a query into the server for the server/legacy system in the unit X.
		9.	The legacy system in unit X starts an user interface based on HTML, which provides the possibility to browse the detailed information of the specified data set.
10.	Physician K reads the data and closes the connection with the server in unit X. and with the regional reference database.	11.	The data from the use of reference database (physician K's identification data, patient id, time, which references were asked for and which were delivered) are written into the log-file of the reference database.
12.	The care of the patient A continues and the new data from the visit is written in.	13.	The data from the visit is stored into the legacy system.
14.	At the end of the visit the physician K asks the patient the permission profile concerning the data of the visit and the	15.	The permission profile is stored in the legacy system separately for the outpatient visit and the period of care

	period of care.		
		16.	The legacy system sends the reference data into the regional reference database (including the permission profile).

3.2.2 Scheduled appointment

	Actor action		**System response**
1.	Patient B visits the healthcare center and is decided to send into secondary care unit Z for care and examination.		
2.	A referral is generated for the patient B into the unit Z	3.	The referral is booked into the legacy system in the healthcare center and is sent into the unit Z.
4.	Unit Z receives the referral letter.	5.	The data of the referral is stored into the legacy system of the secondary care unit Z
6.	The chief physician decides to accept the patient to be taken in and the time for the first visit is scheduled.	7.	The visit is booked in the legacy system in unit Z.
		8.	Reference of the booked appointment is send into the reference database and into the permission profile a permission for the referring physician is marked.
9.	Patient A arrives into the healthcare unit's reception desk.		
10.	Patient A is identified	11.	Patient's data is written into the legacy system
12.	Physician M decides to search for the patient's data stored in other healthcare units.	13.	Physician M makes a query about the patient A for to the regional PRM and gets as an answer N references sorted according to the date. The references points in the healthcare actions (episodes of care, outpatient visits and booked appointments) done to patient in different healthcare units.
14.	Physician M decides to look in details the data from B's visit into healthcare center.		
15.	Physician M declares, what is her/his	16.	The system checks the announced role

	relationship (role) with the patient A		against the permission profile (a part of the reference). The system sends a query into the server for the server/legacy system in the unit X.
		17.	The legacy system in unit X starts an user interface based on HTML, which provides the possibility to browse the detailed information of the specified data set.
18.	Physician K reads the data and closes the connection with the server in unit X. and with the regional reference database.	19.	The data from the use of reference database (physician K's identification data, patient id, time, which references were asked for and which were delivered) are written into the log-file of the reference database.
20.	Patient is diagnosed and the physician M decides to make a hip operation (DRG-209). The new data from the visit is written in	21.	The data from the visit is stored into the legacy system.
22.	At the end of the visit the physician K asks the patient the permission profile concerning the data of the visit and the period of care.	23.	The permission profile is stored in the legacy system separately for the outpatient visit and the period of care
		24.	The legacy system sends the reference data into the regional reference database (including the permission profile).

3.2.3 Following the Patient B during the care

	Actor action		**System response**
1.	Patient B has visited the healthcare center Y. She has a referral for the further examinations into secondary care unit Z. The referral is based on ICPC-coding.		
2.	The referring physician L wants to follow the patient B in unit Z	3.	Physician L connects into the reference database and identifies him/herself.
4.	Physician L chooses the patient according to the name, city and social security number and announces his/her role and relationship to patient B	5.	Physician K makes a query about the patient A for to the regional PRM and gets as an answer N references sorted according to the date. The references points in the healthcare actions (episodes of care, outpatient visits and booked appointments) done to patient in different

			healthcare units.
6.	Physician L decides to search for the references related to the regional guidelines/programme of care of a hip patient, which is coded in primary healthcare using ICPC-code XXX.	7.	The system checks the announced role against the permission profile. System makes a query using as keys Patient Id, codes DRG209 and ICPC XXX and gets as an answer N references pointing into the different healthcare units.
8.	Physician L reads the references (actions of care) and finds that the patient have had a period of care in the unit X, which has operations related. The physician decides to look the data in more detail.	9.	The legacy system in unit Z starts an user interface based on HTML, which provides the possibility to browse the detailed information of the specified data set.
10.	Physician L wants to see the epicrise information of this period of care	11.	The server in unit Z interfaces the legacy system in the unit and provides the data in HTML-format.
12.	Physician K reads the data and closes the connection with the server in unit Z. and with the regional reference database.	13.	The data from the use of reference database (physician K's identification data, patient id, time, which references were asked for and which were delivered) are written into the log-file of the reference database.

3.2.4 Patient B wants to check who has searched for or used his/her medical records during the period of year 19XX

	Actor action		**System response**
1.	Patient B comes into the health care center Y and wants to know who has searched for or looked his/her data during the period of year 19xx.		
2.	The patient's "own" physician guides the patient to the nurse. The nurse logs in.	3.	System connects into the reference database and identifies the Nurse N.
4.	The nurse N chooses the patient according to the name, city and social security number. The nurse also announces her role.	5.	The system generates query for the reference database system logfile, which returns the information from the log file (physician K's identification data, time, which references were asked for and which were delivered) sorted according to the physician id and date).
6.	The Nurse N prints the log information on the patient's request.	7.	The system prints the report.
8.	The Nurse N closes the connection with the reference database.	9.	A reference is generated into the reference database (Nurse N's identification data,

			patient id, time, which kind of report was generated)

3.2.5 Viewing the possible specialised healthcare services in the area

	Actor action		**System response**
1.	Citizen A wants to know which specialised healthcare services are available in his living area		
2.	Citizen A search for the links and sites according his municipality by internet-search engine		
3.	Citizen A contacts home pages of his municipality Z		
4.	Citizen A is studying the primary healthcare web-pages of his municipality Z, which has links to the website of regional, specialised healthcare. Citizen A Decides to enter the regional specialised healthcare web-site	5.	Regional server for regional information services response for the query and loads the front page of web-pages to the user
6.	Citizen A is interested in the specialised healthcare services in the area of his municipality and he chooses map-interface to enter the information.	7.	Regional server for regional information services loads the map to the user
8.	Citizen A chooses his municipality from the MAP-interface.	9.	Regional server for regional information services loads the map of the municipality to the browser and shows the public sector service providers by red colour, private sector service providers by blue colour and emergency unit by red cross.
10.	Citizen A decides to study the services that local hospital provides	11.	Regional server for regional information services loads the font pages of the local hospital to the user
12.	Citizen A wants to have more detailed information of the internal diseases of the local hospital	13.	Regional server for regional information services loads the web-pages of the internal diseases of the local hospital to the user
14.	Citizen A gets the needed contact information and service information of the internal diseases specialty of the local hospital and ends the www-connection	15.	Regional server for regional information services registers the contact and writes the log for the statistics

3.2.6 Viewing the descriptions of wardlines

	Actor action		**System response**
1.	Citizen A wants to know what wardline called 'Hip surgery' means in the area of his municipality		
2.	Citizen A looks for the links of regional specialised healthcare by internet-search engine		
3.	Citizen A moves over to the web-pages of specialised healthcare by the answer of the query	4.	Regional server for regional information services response for the query and loads the front page of web-pages to the user
5.	Citizen A starts the internet-search engine and gives the word 'Hip surgery wardline' to the search word and also the name of his municipality	6.	Regional server for regional information services starts the query and loads the description of the wardline (described by local GP and specialised physician) to the browser of the user
7.	Citizen A reads the description and prints the instruction of the patients to the printer and ends the connection	8.	Regional server for regional information services registers the contact and writes the log for the statistics

3.2.7 Regional reference service

	Actor action		**System response**
1.	Patient A visits physician Y and physician Y decides to send the patient A to the specialised healthcare		
2.	Physician Y contacts the regional server and logs in as the physician 'the services of yellow pages of physicians'	3.	Regional server identifies the user by strong authentication (username, password)
4.	Physician Y asks the patient the best possible date to the first visit in the hospital, which patient answers that the weeks C and D are best ones (6-8 weeks ahead)		
5.	Physician moves over to the web-pages of the available times of first visits which the hospital has informed (6-8 week ahead)	6.	Regional server loads to the browser of the physician the times available for the first visit in weeks C and D

7.	Physician Y and Patient A decide to reserve provisionally the best available time in the week of D and day E and time F	8.	Regional server marks the reserved time to be booked provisionally referred by physician Y
9.	Physician Y fills the reference (running the regional server) form and signs the reference by his smartcards	10.	The server receives the referral information written by physician Y, combines the wanted date for the first visit as a wish to the hospital (hospital confirms the final data) and combines the electronic signature to the message, which is transferred to the hospital as edifact-message. The legacy systems of the hospital confirm the transfer by email to the physician.
11.	Physician Y ends the connection to the regional server	12.	Regional server registers the contact and writes all the functions to the log file.

3.2.8 Digital signature, creation and checking

Actor: application, e.g. referral system

System: X.500 directory, CMS

	Actor action		**System response**
1.	User asks the application to digitally sign the message.		
2.	Application calls API-library function (e.g. RSA_sign), which is installed into hard disk.		
3.	API-library function creates a hash from the message. Function encrypts the created bit sequence with user's private key.	4.	Function reads the private key either from the hard disk or from the smart card, where the x.509 certificate and private key are installed. If the private key is in the smart card, the function calls CMS to return the private key.
5.	The function returns encrypted bit sequency for the application.		
6.	The application adds the bit sequence with the message and sends it. The message may or may not be encrypted.	7.	The system gains the message, forwards it or stores into to system. It may or may not be encrypted.
8.	The receiver or another user wants to read the message and asks an application to fetch the message from the system.	9.	The system checks the rights (according what is described in use case Access Control) and sends it for

			the requesting application.
10.	The receiver wants to check the digital signature		
11.	The application asks the API-library function to fetch the sender's public key.	12.	The system returns the public key from the X.500 directory, where the certificate and the associated public key are stored.
14.	The API-library function decrypts the digital signature with the public key, recalculates the hash and compares it with the original hash. This way both the origin and the correctness of the message are both checked out.		

3.2.9 Use Case: Identification

Actor: user application (e.g. browser) or a system

System: Identification server, X.500 directory

	Actor action		**System response**
1.	User/system wants to access the service.		
2.	Application/system connects with the service.	2.	Identification server returns an authentication request (=bit sequency) and its public key in the x.509 certificate.
3.	Application/system digitally signs the request with its private key (as described in use case Digital Signature)		
4.	Application/system encrypts the digitally signed authentication request with the identification server's public key using an API-library function.	5.	System decrypts the authentication request with its private key and checks the digital signature.
		6.	System generates a session key and encrypts it with its private key and sends it back for the application
7.	Application/system decrypts the session key message with the public key of the identification server and starts using the session key to encrypt/decrypt the messages		
8.	User/system sends his/hers/its role and sends it encrypting it using the session key.	9.	Continue in use case Access Control

3.2.10 Use Case: Access Control

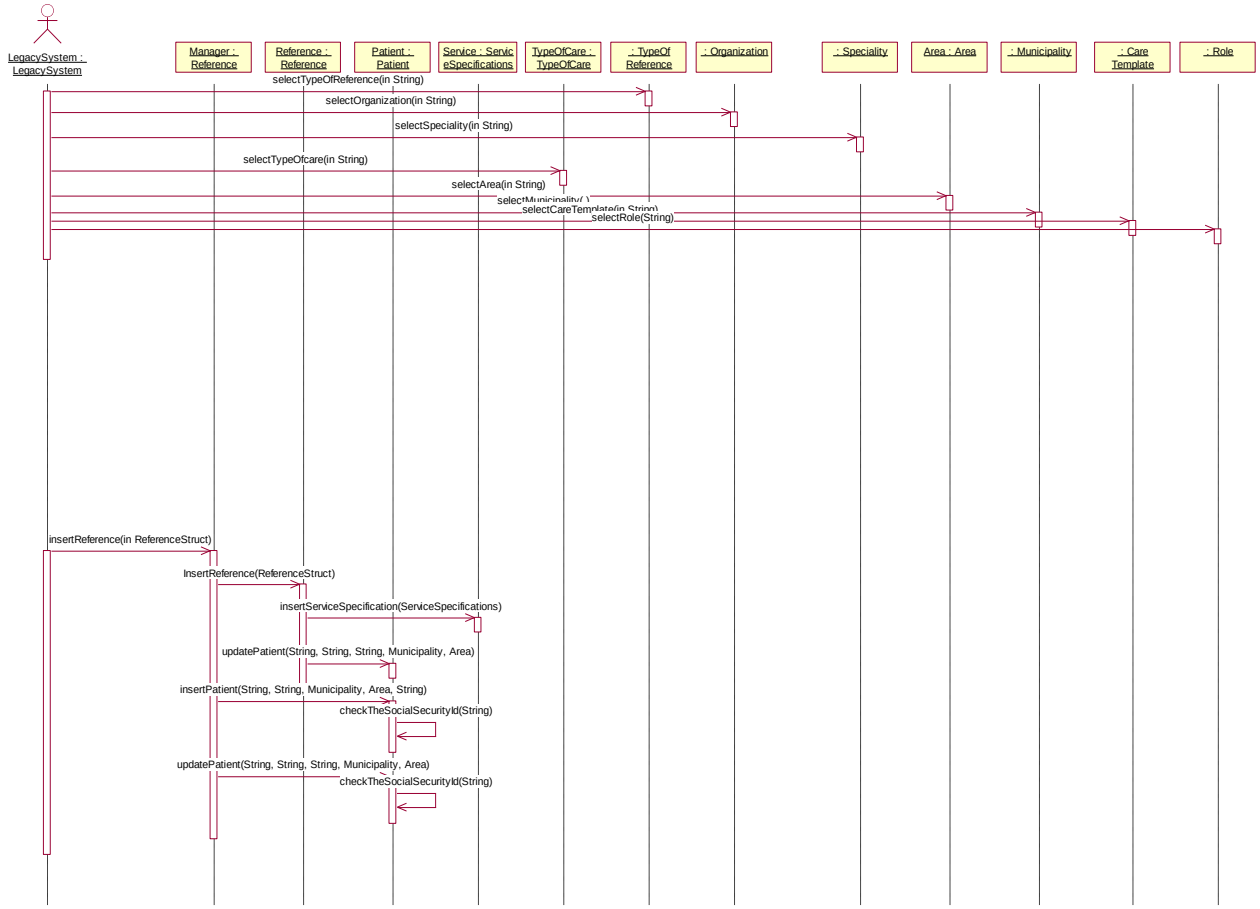
Actor: user application (e.g.. browser) or a system

System: Security Server

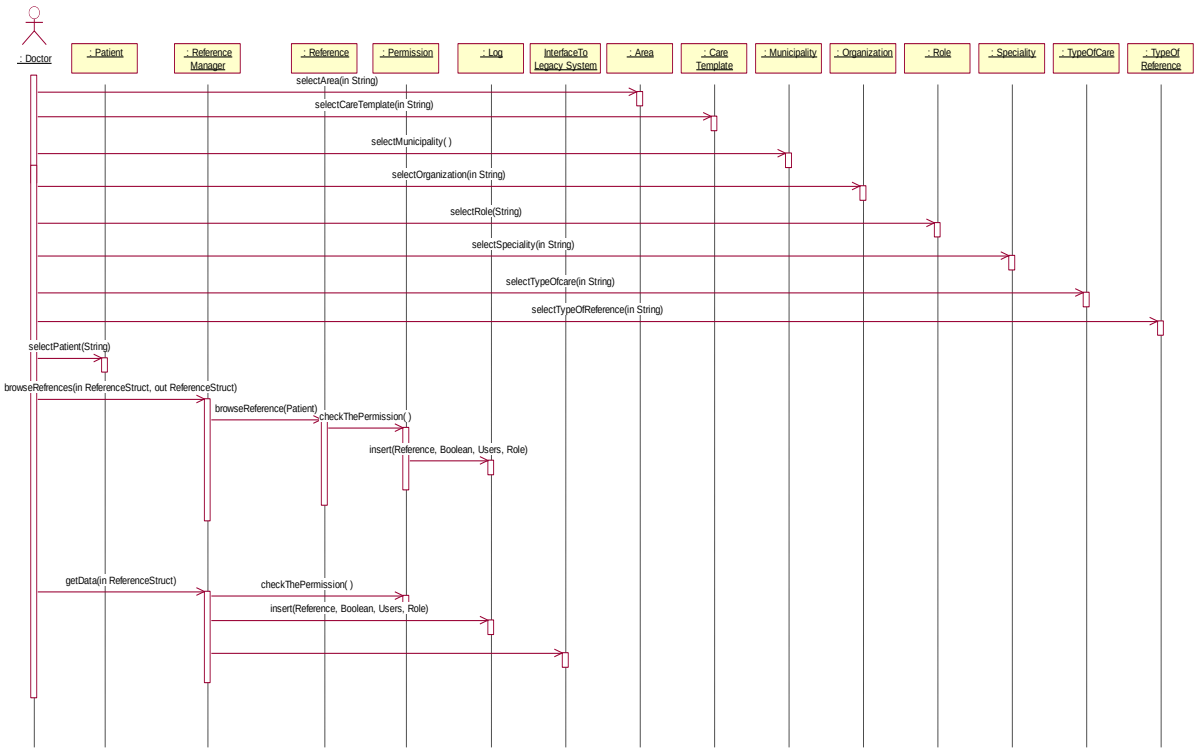
	Actor action		**System response**
1.	User/system wants to access the service.		
2.	Application/system connects with the service.	3.	System Identifies the user/system as described in use case Identification.
		4.	System creates an access ticket using announced role and sends it back to the requesting user/system using the key session to encrypt transfer. System makes a check based on identification and business rules, concludes the role, creates an access ticket and sends it back to the requesting user/system using the key session to encrypt transfer.
5.	Application/system decrypts the ticket using the session key.		
6.	Application/system calls the service needed and sends the ticket for the system. The transfer is encrypted using the session key.	7.	System checks the tickets, the accesses granted there and if it is valid and opens the service. A mark is made into the log file.
8.	Application/system asks the information. The request is encrypted using the session key.	9.	The system checks if the role and the permission profile correlates and returns the requested information. A mark is made into the log file.

3.3 Sequence diagrams

Reference update and browsing

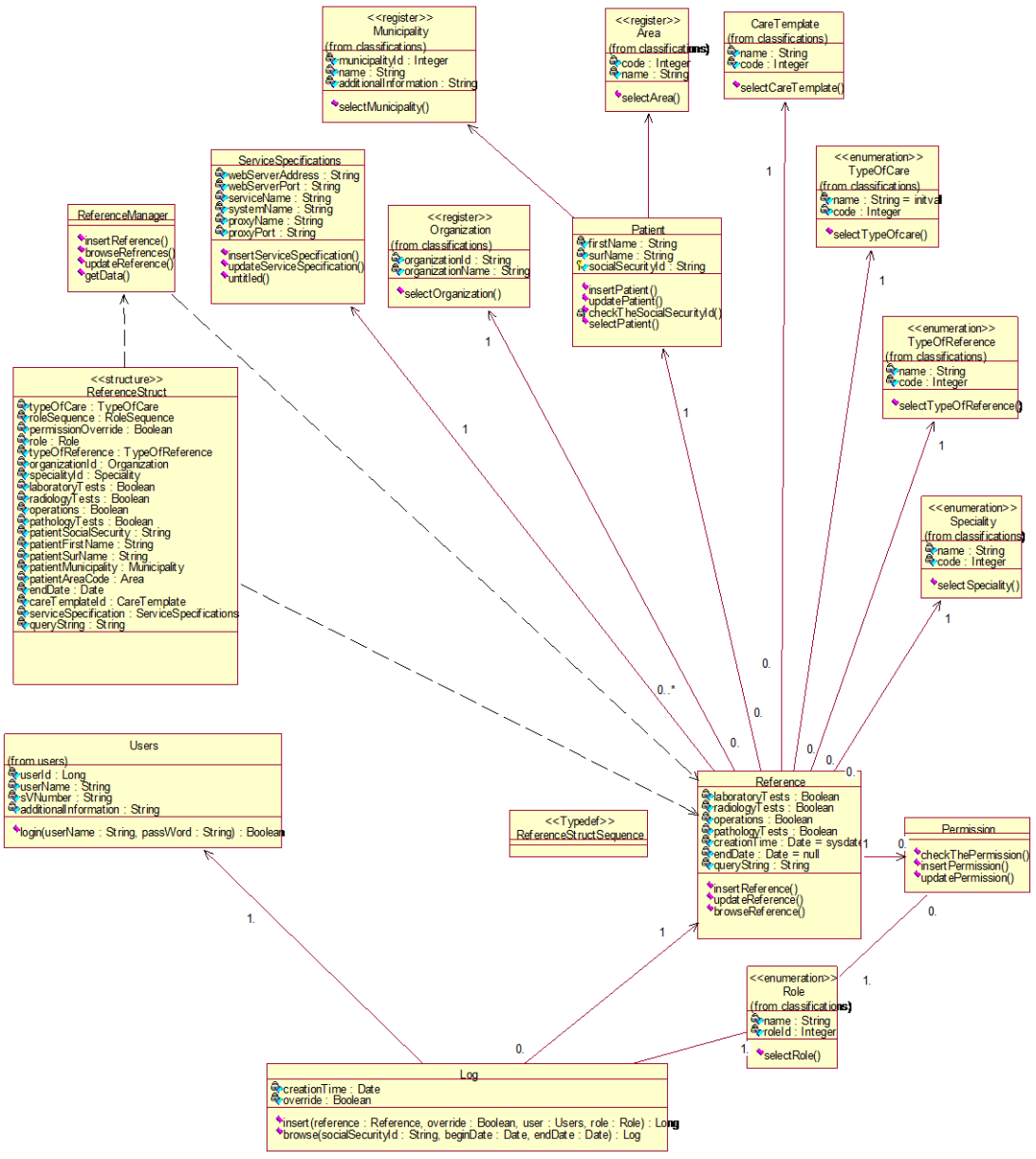


Doctor appointment



3.4 Object Class Model

3.4.1 Class Diagram



3.4.2 Class Descriptions

Regional Reference database

Regional Reference Database

Reference

Derived from IReference, ReferenceManager

ServiceSpecifications

Specifies the services or URLs, who gives the actual reference data from legacy systems according to reference.

Public Operations:

insertServiceSpecification (argname :) : Long

Inserts the new Service specification.

updateServiceSpecification () : Long

Updates an existing service specification.

Patient

Public Operations:

insertPatient (patient : Patient) : Long

Inserts a new patient

updatePatient () : Long

Updates an existing patient's data

selectPatient (socialsecurityId : String) : Patient

Permission

Permission holds the information who can read the reference. It contains 28 different roles that could be permitted or not. Each reference has it own permission object.

Public Operations:

checkThePermission () : Boolean

This operation checks if the reference information is permitted to the given role.

insertPermission () : Boolean

updatePermission () : Boolean

IReference

Log

Log holds information about every read of a reference.

Public Operations:

insert () : Long

Inserts a log row.

browse (socialSecurityId : String, beginDate : Date, endDate : Date) : Log

Browses the log entries.

ReferenceStruct

Reference structure describes the reference message between client and server. In real world applications the set and get methods for each attribute of a class is too heavy and that is why in some cases we have to send only the data. (Corba 3 should have an object by value -mechanism, which solves that kind of problems).

ReferenceManager

Reference manager is the control class of reference-system. It maintains reference inserts, updates, selections and the actual data fetching from the legacy systems.

Public Operations:

InsertReference (referenceRecord : ReferenceStruct) : Long

Inserts a patient's new reference. ReferenceStruct is the input argument.

browseRefrences (referenceRecord : in ReferenceStruct, referenceList : out ReferenceStruct) : Long

Gives a list of a patient's references with given criteria (in referenceRecord).

updateReference (referenceRecord : ReferenceStruct) : Long

Updates an existing reference

getData (referenceRecord : ReferenceStruct) : String

getData fetching the data from legacy system according to reference data, which is an input argument. It returns an HTML-document as a string.

classifications

Municipality

Public Operations:

checkTheExistence (municipalityId : Integer) : Boolean

checkTheExistency checks if a given id exists in Municipality object.

Area

Public Operations:

checkTheExistence (areaCode : Integer) : Boolean

TypeOfCare

Contains the type of care pairs: code and description

Public Operations:

checkTheExistence (typeOfCareId : String) : Boolean

TypeOfReference

Contains the type of references (e.g. booking, period of care, ...)

Public Operations:

checkTheExistence (typeOfReferenceId : String) : Boolean

Checks the existence of a given id in TypeOfReference object.

Organization

Public Operations:

checkTheExistence (organizationId : String) : Boolean

Speciality

Contains the speciality codes

Public Operations:

checkTheExistence (specialityId : String) : Boolean

CareTemplate

Care template describes the different phases of a care.

Public Operations:

checkTheExistence (careTemplateId : String) : Boolean

Users

Users

Public Operations:

login (userName : String, passWord : String) : Boolean

Login gets username and password and gives the boolean value indicates the success or fail.

3.5 Implementation Plan

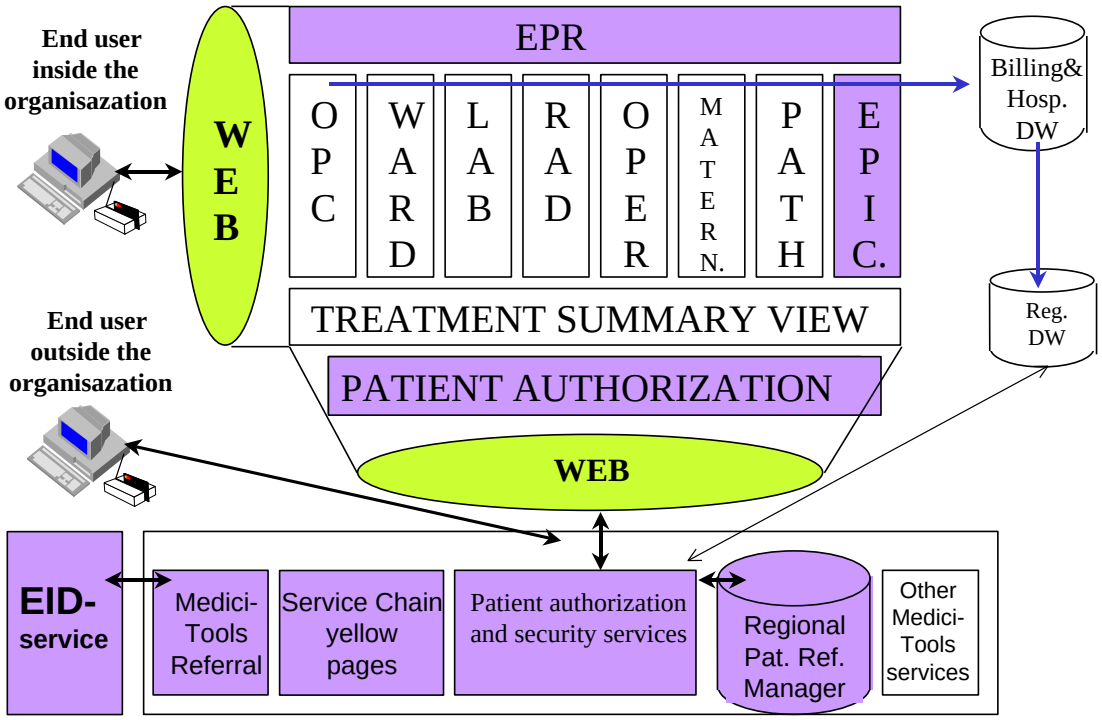
The users of the demonstration are citizens, health professionals in health care centres and hospitals, administrators and care planners. The applications will help to manage the whole care chain and thus shorten waiting lists and reduce costs of care in the region. They also enable the citizen's interactive participation in the organisation of his/her care.

Implementation will covers the following issues:

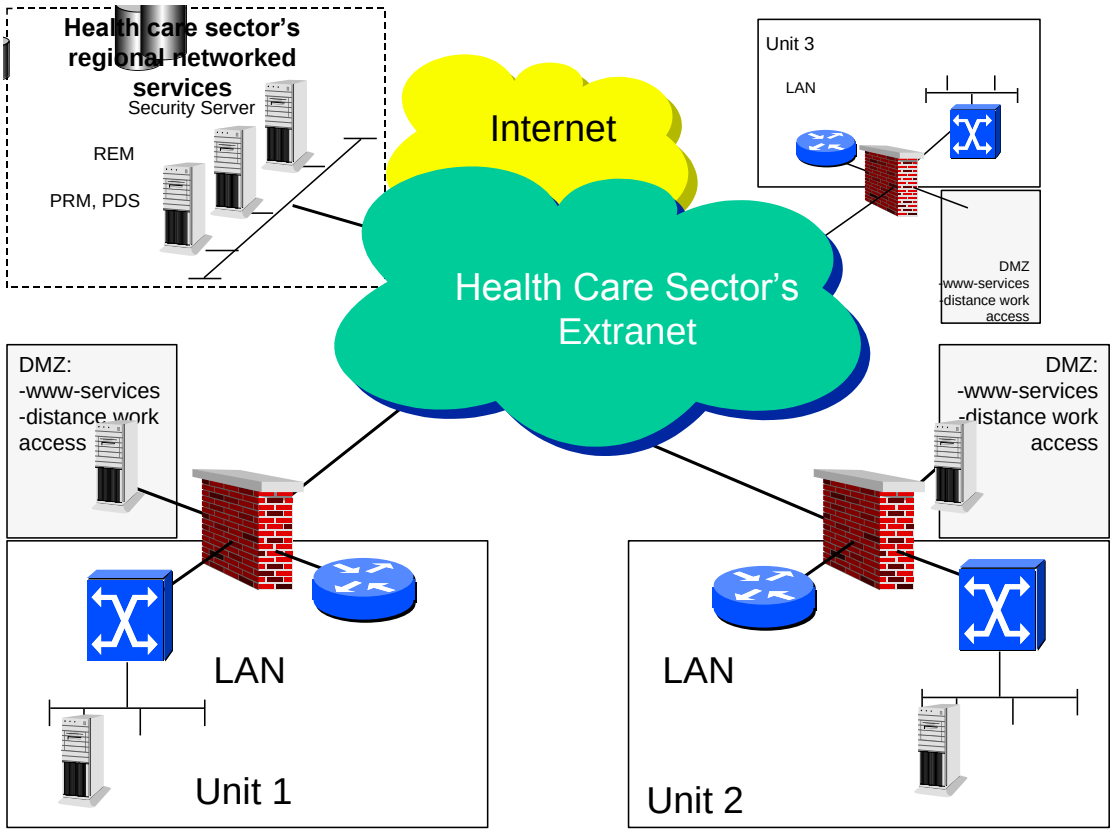
- 2,800 smartcards will be delivered in november 1998
- Smardcard applications and cardreaders will be implemented in november 1998
- Regional reference database, security server and management system will be implemented in february 1999
- Local enterprise reference managers will be implemented in april 1999
- Local patients's permission applications will be implemented in april 1999
- Security services will be implemented in may 1999
- Regional healthcare information systems will be implemented in may/june 1999

3.6 Architectural Aspects

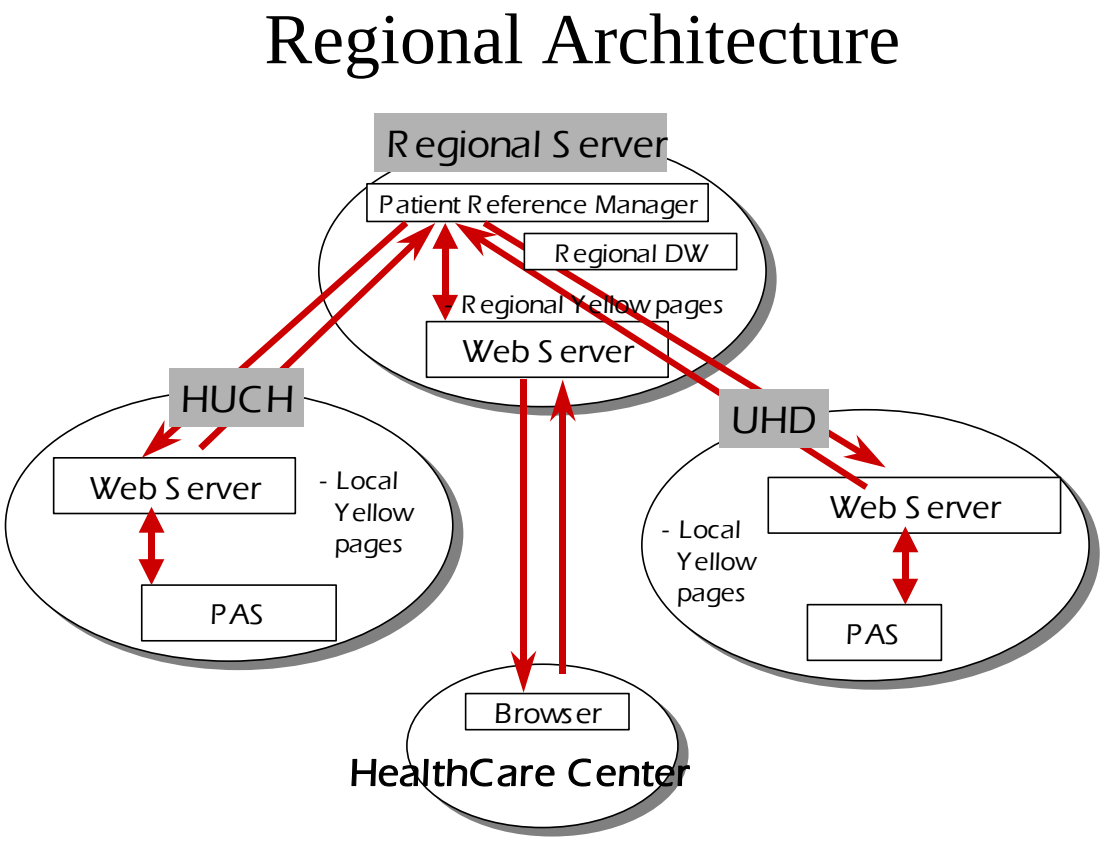
3.6.1 Overall Architecture of the Regional and Local Services and Techniques



3.6.2 Network Architecture



3.6.3 Technical Architecture



4. Crete Demonstrator Specification

4.1 Introduction

Recently, due to the greater mobility of the population as whole, national and international healthcare networks are increasingly used to facilitate the sharing of healthcare-related information among the various healthcare actors. This sharing of information resources is generally accepted as the key to substantial improvements in productivity and better quality of service. Hence, although each healthcare facility is autonomous and devoted to the delivery of a particular set of services, continuity of care requires that different healthcare facilities, offering complementary services or different levels of expertise, exchange relevant patient data, and operate in a co-operative working environment. In this environment, diverse user groups require secure customisable access and sharing of information residing at geographically distributed autonomous information systems. Thus, a Healthcare Information Infrastructure (HII) that enables the coherent integration of heterogeneous components is necessary to reduce the inherent complexity of the related tasks. The core of the HII is middleware services that facilitate interoperability through open standards and public or protected interfaces.

As an example, consider the Virtual Electronic HealthCare Record (EHCR), one of the key user-oriented services enabled by an HII. Heterogeneous autonomous information systems designed to support specific data types, manage EHCR segments. Hence, comprehensive medical information about a patient is difficult to obtain efficiently, unless the distributed and heterogeneous EHCR segments are integrated into a Virtual EHCR and viewed on-line through a unified user interface and visualisation environment. The seamless integration of distributed EHCR segments requires interoperability of heterogeneous autonomous information systems. As a result, standardisation efforts for middleware that facilitate interoperability and enable the communication of information through standard messages are very active. Health Level Seven (HL7) was founded in 1987 to develop standards for the electronic interchange of clinical, financial, and administrative information among independent healthcare-oriented computer systems. The ACR-NEMA Digital Imaging and Communications in Medicine (DICOM) standard has also been developed to meet the needs of manufacturers and users of medical imaging equipment for open interconnection of devices on standard networks. However the use of any single standard by all individual software components still could not satisfy the preferences of different customer groups. What was needed was to have increased site autonomy without loosing the benefit of using standards, and this is exactly the type of solution that has been used in electronic commerce. Here the standard used each time two components communicate has to be determined during their initial negotiation, and the only way to provide very high levels of autonomy to individual components, is to locate mediation services outside local systems. In this setting, general-purpose open integration standards like CORBA play a critical role, facilitating integration at different levels of abstraction. In addition, ***X.500/ LDAP*** directory services provide a robust universal infrastructure that enables the publishing of information on networked resources, in an attribute-value format. Through appropriate IDL interfaces, CORBA enables access to views of the information in a global directory, while LDAP provides tight integration and secure access to the full information model of the directory, under appropriate security constraints. This technological approach can support any consolidation efforts of today for offering regional healthcare resource services, and can function successfully as a global directory service in the future.

4.1.1 Virtual EHCR Services

During a single healthcare episode many professionals, involved in a variety of medical acts, administer medical care. Healthcare administration personnel, healthcare professionals, social care professionals, as well as patients need to selectively interact with health-related information. Each of these user groups has not only different needs in terms of information access, security, and quality of service, but also is involved in different tasks, medical acts, and healthcare procedures. In addition, the

requirements of each user affect information retrieval and presentation strategies, as well as the overall interaction with the information space.

Consider now the Virtual EHCR problem. A physician that accesses a patient's healthcare record, needs an overview of the patient's EHCR segments, since in most cases only a small fraction of the complete record will be selected and presented in detail. That also means that when accessing a particular clinical information system there is a need for extracting only a subset of the information stored in it. The real issue here is not only how to access specific information systems that maintain EHCR segments, but also how to identify and index the essential information in them.

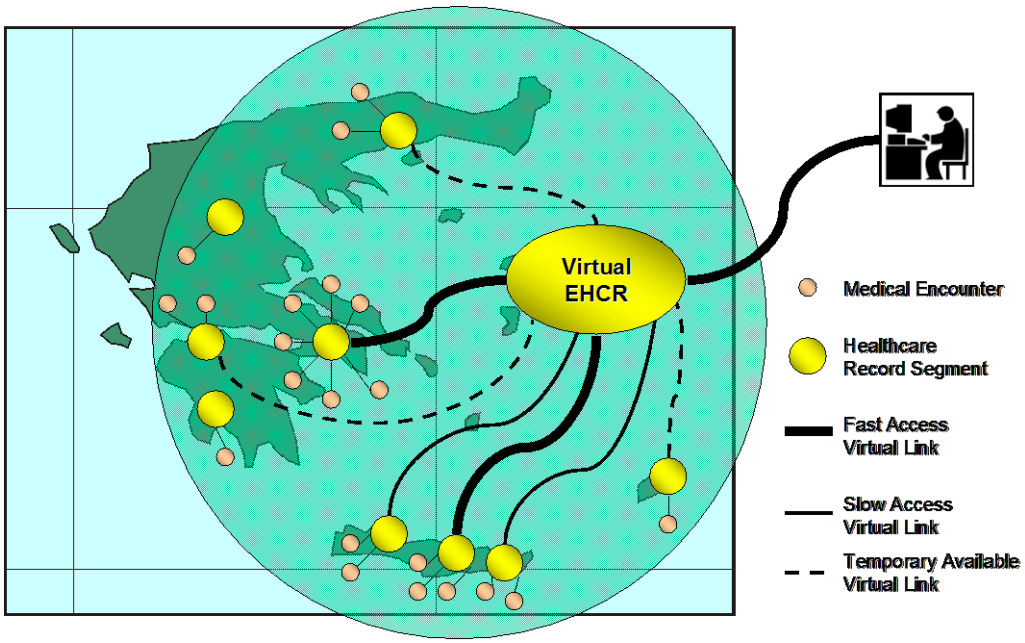


Figure 1: Access to the EHCR segments of a patient's health record involves diverse access methods, and highly variable access times.

A promising approach to this integration problem is to gain control of the organisation's information resources at a meta-data level, while allowing autonomy of individual systems at the data instance level. The objective of the meta-database model is to achieve enterprise information integration over distributed and potentially heterogeneous systems, while allowing these systems to operate independently and concurrently. However, achieving integration at a semantic level is a challenging problem mainly because the logic, knowledge, and data structures used in various systems are complex and often incompatible. In addition, the further someone wishes to hide heterogeneity, the more he/ she has to deal with semantic integration issues. Thus, a realistic solution should hide heterogeneity at the top level, while making the individual sources of information appear to end users as a large collection of objects that behave uniformly. Furthermore, as shown in Figure 1, the ideal world of interconnected information systems is far from being a reality, and hence a feasible solution should also take into account the fact that response time may vary considerably. This means that several parallel actions and, in most cases, data replication (or caching) should be considered.

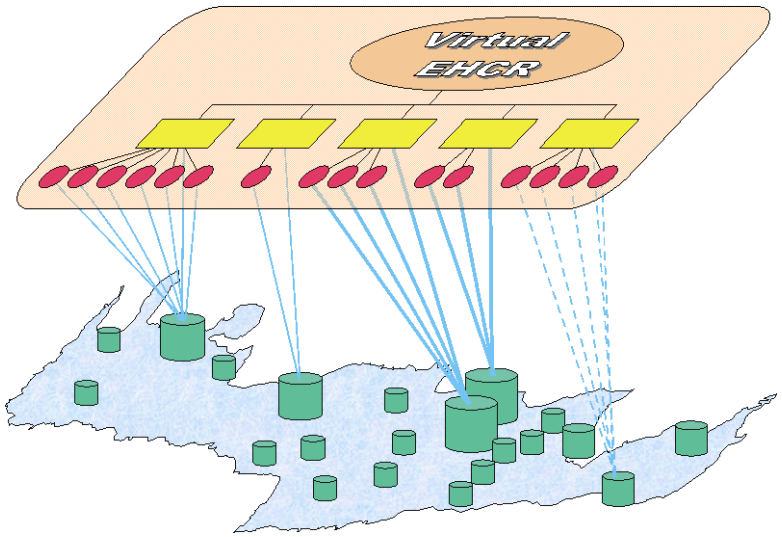


Figure 2: The Virtual EHCR mediates access to the distributed segments of a patient's healthcare record. This mediation is enabled by a directory infrastructure.

These considerations led to the design of the CPDD federation, which provides information integration at the enterprise-level and is enabled by a directory infrastructure that implements a distributed registry of EHCR segments. This approach realises a federation of clinical information systems, is capable of supporting any consolidation efforts of today for Virtual EHCR services, and can function successfully as part of a global directory service in the future (see Figure 2). The hierarchical structure of the directory, although limited at first sight, provides a powerful way to index the geographically distributed EHCR segments and can facilitate the provision of a Virtual EHCR service. The directory maintains a distributed registry for feeder systems and patients in the federation, as well as references to specific clinical objects in the EHCR segments of these patients. Since the directory stores access information for clinical objects in the EHCR segments, it may act as an access mediator. Thus, the treating physician, assuming authorisation is granted, may use one of the available access methods, to access relevant clinical objects in the feeder systems.

4.2 Crete CPDD Services Demo Application

CPDD: Clinical Patient Data Directory, include a set of middleware services that indices the EHCR segments of patients' health records and facilitates global access to patient clinical objects in feeder systems, i.e. geographically distributed, heterogeneous, autonomous clinical information systems. The CPDD maintains a distributed registry of feeder systems, patient key demographics, their EHCR segments, clinical meta-data, and references to clinical objects. The data model of the CPDD is based on the Subjective Objective Assessment Plan (SOAP) model that originated from the primary healthcare domain. Access to detailed information on particular healthcare encounters is delivered on a case by case basis. Each feeder information system is responsible to provide an export schema and map it to the schema of the directory. With regard to updates, different update schedules can be applied at different sites. The core technologies of the CPDD are X.500/ LDAP directories, CORBA, data extraction gateways (such as ODBC-to-LDAP), and related Internet technologies.

Relational databases are currently used extensively for storing enterprise data. A key strength of relational databases is their ability to make complex queries about the relationship between objects. Directories are specialized databases with a hierarchical information model, which are defined in terms of open, standardised, and vendor-neutral protocols, namely X.500 and LDAP. Key strengths of directory technology are its distributed provision and fast lookup based on name. On the other hand, Network Operating System (NOS) directories (e.g. Novell's Directory Service) are "low level" directories used to support location of computer servers, printers and other Local Area Network (LAN) resources. Microsoft's Active Directory, which is a new directory service from Microsoft (it

will be part of Windows NT 5.0) aims towards enhanced system integrity, in order to offer personalised user environments, simplified service and application configuration, security service integration, and improved bandwidth allocation.

In essence, the directory is a distributed database capable of storing in a hierarchical data model, information about people and objects in various nodes or servers distributed across a network. It is these servers that provide the potentially global access to information, which is made possible by X.500 technology. According to both the X.500 directory and LDAP have been identified as key components in the development of meta-directories. X.500 provides the central directory, controls and facilitates access to data in proprietary directories, while LDAP controls the communication from directory users, and retrieves and populates the proprietary directories with the core information.

Current directory technologies use the Internet standards (i.e. TCP/ IP), conform to a global naming structure (i.e. DNS), provide public interfaces for directory access (e.g. LDAP), integrate and maintain synchronisation with other transitional directories, and support security (i.e. X.509 and Kerberos). Since many users wish to access directory information via the Web, the WWW has an important relationship to the directory and is a key element to any directory solution.

4.2.1 Objectives, purpose and model of the application

The main objective of CPDD middleware is to provide basic support for Virtual EHCR services in a consistent, reusable, and extensible way. The CPDD indexes patient and feeder system identification, as well as information about clinical objects in a patient's EHCR segments. This information is accessible to authorised users and applications through stable IDL interfaces. The central component of CPDD is an X.500/ LDAP directory, which maintains a *registry* of feeder systems, patient key demographics, their EHCR segments, clinical meta-data, and references to clinical objects.

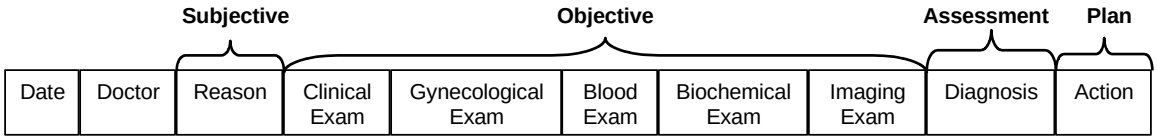


Figure 3: Medical encounter entries in the directory follow the SOAP model.

Information about clinical objects in the directory refers to clinical patient data that are produced during the communication about the patient, between two or more healthcare professionals. This communication is called *encounter*. The most common type of encounter is a visit to a medical office, clinic, or primary healthcare center. Medical encounter entries in the directory follow the Subjective Objective Assessment Plan (SOAP) model (Figure 3). The SOAP model is an approach for recording clinical data generated during the contact of a patient with a healthcare provider. *Subjective* refers to the reason of the contact (i.e. the context of the encounter[1]). *Objective* applies to medical examinations requested or reviewed during the contact e.g. blood examination, biochemical examination, computerised radiography (CR), etc. *Assessment* refers to the clinical diagnosis and associated reports. *Plan* refers to the clinical actions that must be taken, i.e. the treatment plan (drugs, surgery, etc).

The CPDD also provides the information required to access clinical multimedia information directly from its source (i.e. CPDD feeder systems). Feeder systems may support a wide variety of access methods ranging from human-mediated, to CORBA object references, and HTTP URLs. Furthermore, access to these systems is subject to site-specific authorisation policies and is available through the regional HII. The CPDD does not deal with the semantic mapping problem directly. It is the responsibility of a feeder system to map an export schema of its information model to the directory information model. The underlying assumption is that this mapping is provided by a human who understands the semantics of both the directory model and the export schema of the feeder system. A

minimal export schema of a feeder system should express the existence of patient's EHCR segment by providing patient identification attributes.

4.2.2 Directory Information Model

Each patient in CPDD has multiple *HealthcareRecordSegments*, which correspond to the segments of the patient's EHCR. Each *HealthcareRecordSegment* resides in a feeder system (*SystemInstallation)* and consists of a set *MedicalEncounters* of the specific patient. The directory supports a number of feeder *SystemTypes*, each of which has been installed in a number of healthcare organisations (feeder *SystemInstallations).* All feeder systems need not be identical. Location and access methods may differ too. The minimum information that a feeder system should provide is the fact that a given patient has an EHCR segment in the particular information system.

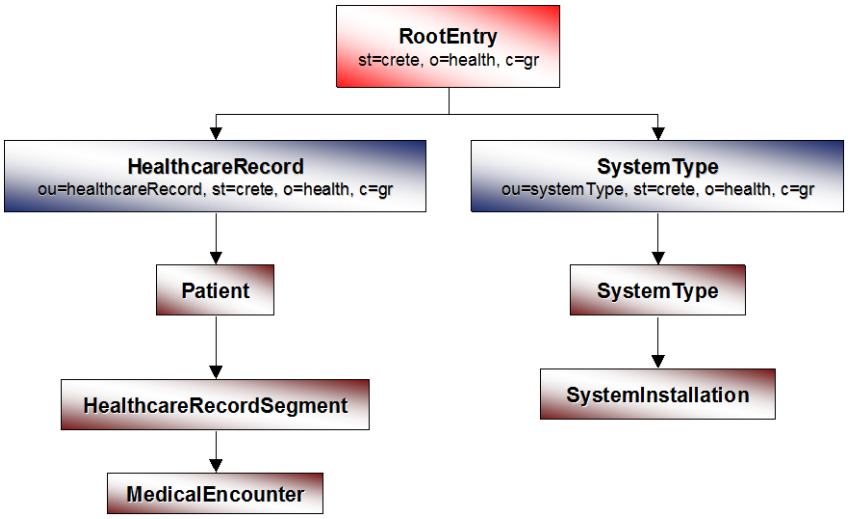


Figure 4: Directory Information Tree supported by CPDD.

The structure of the directory information tree for CPDD, appears in Figure 4. Each entry is identified among entries of the same object class by its Relative Distinguishing Name (RDN). The RDN for patient entries is the *cn* attribute. For all other object classes , it is the *uid* attribute. In addition, each entry (node) is uniquely identified in the directory information tree by its Distinguishing Name (DN), which is the sequence of RDNs in the tree path from the root to the current entry. The *RootEntry* of the directory is "st=crete, o=health, c=gr." Note that the present directory may be linked to other directories, which correspond to other regions. For example, the CPDD in the region of Epirus has root entry "st=epirus, o=health, c=gr." The two main branches of the directory information tree are *SystemType* and *HealthcareRecord* and maintain entries relevant to feeder systems and patient data, respectively.

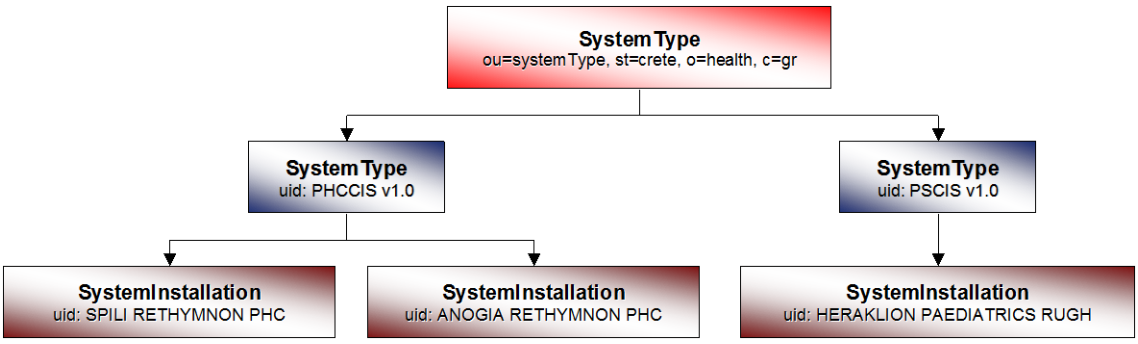


Figure 5: System Type subtree in a directory shows two different *SystemTypes* entries: PHCCIS v1.0 and PSCIS v1.0. In this example, PHCCIS v1.0 has two installations, while PSCIS v1.0 has one.

A possible *SystemType* branch appears in Figure 5. Observe in the figure that the root of the subtree (*SystemType*) is marked with its *DN*, namely "ou=systemType, st=crete, o=health, c=gr," where all other nodes are marked by *uid*, the *RDN* of the specific entries. The *name* and *version* of the clinical information system (e.g. PHCCIS v1.0) identify *SystemType* entries (i.e. PHCCIS v1.0). *SystemInstallation* entries are identified by the geographic location and organizational unit, where the feeder system has been installed (e.g. SPILI RETHIMNON PHC).

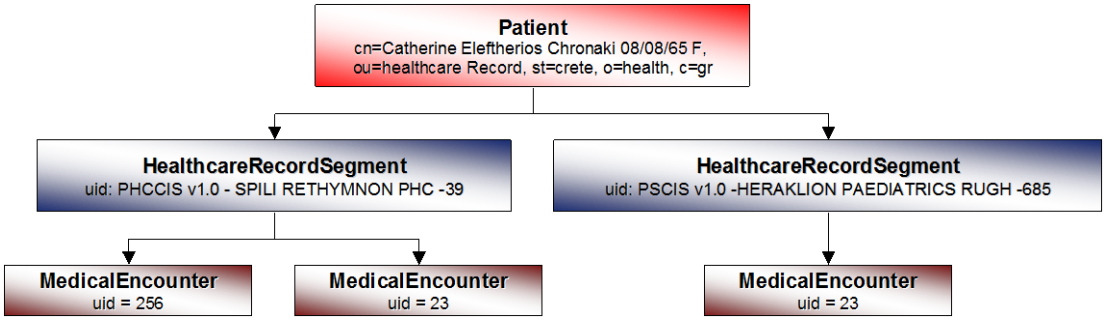


Figure 6: Patient subtree in a directory, shows a patient who has visited twice the Spili PHC and once the Pediatrics Clinic at the RGUH of Crete.

The *HealthcareRecord* branch of the directory information tree organizes patient-related data. *Patient* entries express a unique CPDD-wide patient identity. Each patient is uniquely identified in the directory by five attributes, namely *firstname*, *lastName*, *parentName*, *gender*, and *birthDate*. This is due to the fact that there is no single region-wide person identification. Thus, if two patients have the same sequence of values for these attributes, their *EHCR* segments are classified under the same *Patient* entry. Even this seemingly simple solution, however, raises several problems. First, with low probability, there will be a single *Patient* entry for two physically different persons. Second, several systems do not keep *parentName* or full *birthDate* data. Other systems, especially those that deal with emergencies, do not always have even the *lastName* registered in the database. CPDD deals with these problems on a case by case basis. If a system does not have *lastName* data in a record, the record is considered incomplete and the corresponding data is not inserted in the directory. If just the *parentName* does not exist in the record, then the indication "*UNKNOWN*" is entered as the value of the attribute. *birthDate* has also several problems associated with it. The directory stores *birthDate* attribute values in the form dd/ mm/ yyyy. In small villages, however, elderly citizens do not recall their exact *birthDate*; in the best case, they may recall the year approximately. The directory stores dates as strings, thus, *birthDate* "dd/ mm/ 1933" means that only the year of birth is known.

For each EHCR segment of a given patient, a new *HealthcareRecordSegment* entry is placed under the corresponding *Patient* entry. Thus, the subtree of the directory information tree rooted at a specific *Patient* entry has as many branches as the feeder systems that store EHCR segments of that patient. The clinical meta-data of a medical encounter appear in the *MedicalEncounter* leaves of the subtree rooted at the corresponding *HealthcareRecordSegment* entry. A possible *Patient* subtree appears in Figure 6. In the figure, the *Patient* entry (root node) is marked with its *DN* (i.e. "cn=Catherine Eleftherios Chronaki 08/08/65 F, ou=healthcarerecord, st=crete, ou=health, c=gr"), while all other entries are marked by their *RDN* (i.e. *uid=…*). *HealthcareRecordSegment* entries are partially identified by the *patientID* in the feeder system (e.g. PHCCIS v1.0 – SPILI RETHIMNON PHC – 39). Note that for a *HealthcareRecordSegment* entry to be valid, the corresponding *SystemType* and *SystemInstallation* entries should be present in the directory. *MedicalEncounter* entries are also identified by their *ID* in the feeder system.

SystemType entries identify clinical information systems uniquely by their name and version. Additional attributes that may be updated, are a description of the system and information on the person responsible for the software. Figure 7. shows an example S*ystemType* entry.

```
dn: uid=PHCCIS v1.0, ou=SystemType, st=crete, o=health, c=gr
objectClass:        SystemType
uid:                PHCCIS v1.0
systemName:         PHCCIS
systemRelease:      v1.0
systemDescription:  Primary Healthcare Information System
systemContact:      Nikos Stathiakis
contactNumber:      081 391644
systemEmail:        statiaki@ics.forth.gr
```

Figure 7: Example of a *SystemType* entry in the directory.

Feeder systems are installations of clinical information systems of known *SystemType*, in a healthcare facility. They are represented in the directory by *SystemInstallation* entries. The concatenation of the *location* and *organizationalUnit* identify the *SystemInstallation* entry of a feeder system. Observe that feeder systems correspond to *SystemInstallation* entries and not *SystemType* entries. For example, the *SystemInstallation* entry of PHCCIS v1.0 in SPILI RETHIMNON PHC and the *SystemInstallation* entry of PHCCIS v1.0 in ANOGIA RETHIMNON PHC are two different feeder systems, which are instances of the same *SystemType.* The fact that a specific *SystemType* entry may have been installed at multiple organisational units, is shown in the conceptual model with the relation between the SYSTEM TYPE and SYSTEM INSTALLATION classes. Authorisation and certification of the feeder systems that have the right to access and update the CPDD is provided by a unique key *accessKey* that is assigned to each feeder system. An example *SystemInstallation* entry appears in Figure 8.

```
dn: uid= SPILI RETHIMNON PHC, uid= PHCCIS v1.0, ou=SystemType, st=crete, o=health, c=gr
objectClass:        SystemInstallation
uid:                SPILI RETHIMNON PHC
location:           SPILI RETHIMNON
organizationalUnit: PHC
installationDate:   15/7/97
contactPerson:      G. Frantzeskakis
contactNumber:      0823-22233
contactEmail:       gef@ret.forthnet.gr
networkAddress:     I.P. address 139.91.191.3, port 3576
accessKey:          39eceda1e5c12ccf71444ed8cd012666
```

Figure 8: Example of a *SystemInstallation* entry in the CPDD.

Besides patient identification data, *Patient* entries in the directory include optional demographics attributes. These are *placeOfBirth*, *identification,* and *address*. The value of the identification attribute includes both the unique IDs and the issuing authority within brackets, separated by commas. Since attributes allow multiple values, multiple identifications of a patient within an information system can be supported. Just like a *SystemInstallation* entry, a *Patient* entry includes an *accessKey,* a special attribute for authorisation and certification purposes. A sample *Patient* entry appears in Figure 9.

```
dn: cn= Catherine Eleftherios Chronaki 08/08/1965 F, ou=healhtcareRecord, st=crete, o=health, c=gr
objectClass:        Patient
cn:                 Catherine Eleftherios Chronaki 08/08/1965 F
lastName:           Chronaki
givenName:          Catherine
parentName:         Eleftherios
birthDate:          08/08/1965
placeOfBirth:       Xanthi
gender:             F
address:            Vrioulon 43, Heraklion, Crete, 71306, Greece
identification:     "P315835","DX Hrakleiou"
identification:     "215-43-3423","US Social Security No"
accessKey:          03a183fd8c94c95fb0b235380717f6b8
```

Figure 9: Example of a *Patient* entry in the Directory of CPDD.

A *HealthcareRecordSegment* entry corresponds to an EHCR segment of a patient. The attribute *uid* identifies *HealthcareRecordSegment* entries. The value of *uid* is the concatenation of the attributes *systemInstallationID and patientID* separated by "-." *PatientID* is the ID of the patient in the feeder system, while *systemInstallationID* is the location and name of the organisational unit. Note that the insertion of a new *HealthcareRecordSegment* entry requires the existence of the corresponding feeder *SystemInstallation* entry in the directory. The attribute *recordCode* refers to the paper record of patient in the organisational unit, if such a record exists. A sample *HealthcareRecordSegment* entry appears in Figure 10.

```
dn: uid=PHCCIS v1.0 - SPILI RETHIMNON PHC - 39, cn= Catherine Eleftherios Chronaki 08/08/1965 F,
ou=healhtcareRecord, st=crete, o=health, c=gr
uid:                          PHCCIS v1.0 - SPILI RETHIMNON PHC - 39
objectClass:                  HealthcareRecordSegment
cn:                           Catherine Eleftherios Chronaki 08/08/1965 F
systemInstallationID:         PHCCIS v1.0 - SPILI RETHIMNON PHC
patientID:                    39
recordCode:                   E2543
```

Figure 10: Example of a *Healthcare Record Segment* entry in the Directory.

A *MedicalEncounter* entry corresponds to a single contact of a patient with a healthcare facility. Multiple medical encounters may be part of a single *episode,* i.e. the treatment of a long-term problem. This viewpoint is not supported by the current version of the directory information model. The identification attribute of *MedicalEncounter* entries is *uid,* the identification of the particular encounter within its feeder system. Other important encounter-related information is the *date* and *time* of the encounter and the attending physician (*doctor).* The rest of the attributes refer to the SOAP model. *Reason* (e.g. "Feeling of Arrhythmia") refers to the subjective symptoms. *Biochemical, blood, clinical, gynecological, radiology* are Y/N indicators of whether the corresponding clinical objects were produced during the encounter. All these indicators refer to the objective signs (physical signs and test results). "N" refers to absence, while "Y" refers to presence of examination results. *Diagnosis* indicates whether a diagnosis has been associated with the contact. In specific, diagnosis refers to the assessment and the rationale for a selected plan (a summary of the patient's condition or course of illness); again "N" refers to absence, while "Y" refers to presence of diagnosis. Finally, *action* is a Y/N indicator of whether certain actions have been prescribed. Apart from the attribute *reason,* all the other attributes referring to the SOAP model simply indicate the existence of the relative information. An example *MedicalEncounter* entry appears in Figure 11.

```
dn: uid=256, uid=PHCCIS v1.0 - SPILI RETHIMNON PHC - 39, cn=Catherine Eleftherios Chronaki 08/08/1965
F, ou=healhtcareRecord, st=crete, o=health, c=gr
uid:                256
objectClass:        MedicalEncounter
dateOfVisit:        03/2/1997
doctor:             George Frantzeskakis
reason:             Feeling of Arrythmea
action:             Y
biochemical:        N
blood:              Y
clinical:           Y
radiology:          N
gynecological:      Y
diagnosis:          Y
HTTPuri:  http://asklas.ics.forth.gr/phccis/ldap_visit.asp?database=phccis_anogia&patId=1&vId=111
CORBAuri:           "asklas.ics.forth.gr", "PCDD", "medical_encounter", "12", "44343"
```

Figure 11: Example of a *MedicalEncounter* entry in the Directory of CPDD.

4.3 Use Cases and scenarios

Figure 12, shows the use case scenarios that involve the CPDD. The *End User* accesses information in the directory through user-oriented services that use the CPDD*Query* interface. The CPDD *Administrator* can also incorporate new feeder systems in the federation using the CPDD *Admin* interface. Feeder *Clinical Information Systems* may use special components that feed data into the directory through the CPDD *Update* interface. They may also use the CPDD *Query* interface as part of more complicated use cases, related to the correlation of patient identities in feeder information systems and the co-ordinated execution of complicated tasks.

Diagram:

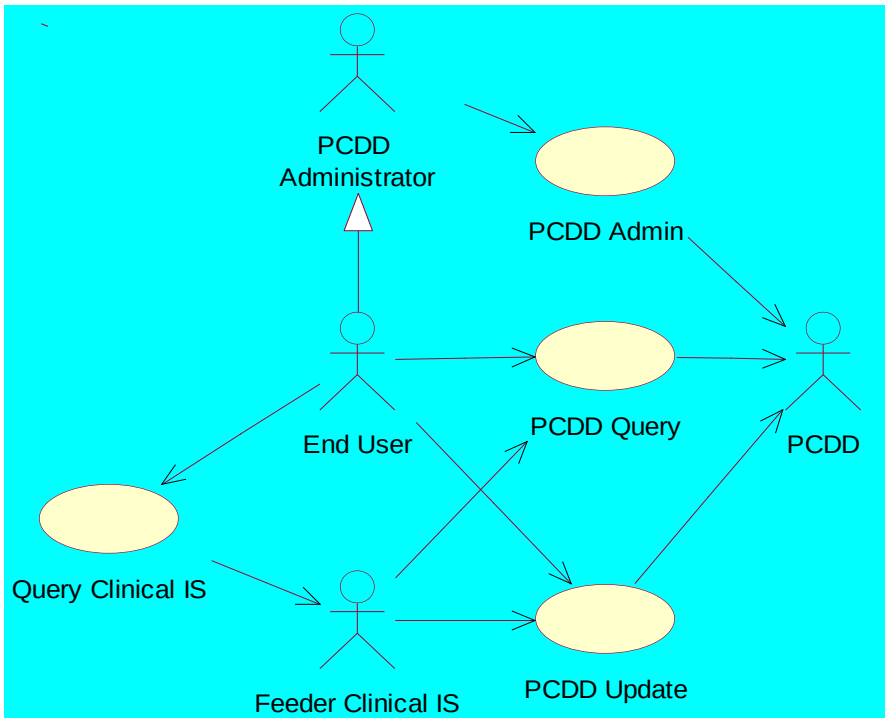


Figure 12: UML use case diagram demonstrating the basic operations that involve the CPDD services

Descriptions:

- CPDD services relate to several different **scenarios** of use:
  - A healthcare professional that interacts with a clinical information system (e.g. Paediatric Surgery CIS or Primary Health Care CIS) **refers** a patient to an expert in another facility. The referee needs access to acts related to the referral. As an example we attach the referral of the Regional Pediatric Management System.
  - A healthcare professional in a primary healthcare center has requested a diagnostic examination in a central facility. At subsequent visits of the patient, the professional is able to **monitor the status** and **products** (objective data) of the examination. Diagnostic examinations that will be considered involve radiology, laboratory, and cardiology departments.
  - A healthcare professional has to treat an emergency. Information regarding the emergency encounter is provided to the professional, through access to the relevant patient record segment.
  - A treating physician needs to **browse** the patient record of a patient, according to aspects of the SOAP (Subjective Objective Assessment Plan) model.

Note, that these four directions provide the framework for scenarios relating any clinical information system with any diagnostic department and the emergency care information system. The demonstration of such scenarios involves the maintenance of a directory (PMR in Star). The maintenance of the directory requires provision for **update** and **query** operations, which are outlined below.

The directory provides the treating physician with a summary of a patient's ***virtual patient record***. From this summary, the physician may request further information on a particular item by clicking on the appropriate link.

Patient referral between health care structures is one of the most used scenarios in the context of Patient Management in the region of Crete. The actors involved include Health Care Centers (HCCs), Prefecture Hospitals, and University Hospitals. There is an urgent need for medical information sharing and navigation through the *fragmented* patient record. Such operations result actually into the concept of the *Virtual Patient Record* (VPR).

Acute Abdominal Pain in Children is one of the most common cases in the childhood population of Crete. In many cases operative decisions are taken. Children living into rural areas have to be transferred at places where the appropriate expertise and facilities are present such as the Prefecture Hospitals and the University Hospital of Crete. Referrals may take place between HCCs and Hospitals (Prefecture, University) or between Prefecture and University Hospitals.

A physician located at a remote health care structure faces a child patient with Acute Abdominal Pain (AAP) symptoms. There are two alternate scenarios, treat the case at place or refer the case to the most appropriate health care structure. In the former the physician has first to update the local clinical information system and second has to make the produced information available to other physicians located in the region. In the second case a referral process is initiated. It involves two parts: the *requester* (remote physician) and the *performer* (expert).

The **Requester** has to

1) be able to navigate and inspect all past relevant patient information,
2) initiate the local patient management,
3) update the local information system,

4) inform the performer about the referral and
5) somehow, make the produced information available to the performer.

The **Performer** has to

1) either accept or reject the referral,
2) inform the requester,
3) in case of acceptance the performer must be able to navigate and inspect all relevant patient information; this means the information produced and published by the performer, as well as any past published information concerning the patient.
4) initiate the local patient management,
5) update the local information system,
6) somehow, make the produced information available to others for future use
7) when the case is discharged, close the referral.

- Different **functions** are to be performed within the CPDD services.
  - **Directory Updates**

    Directory updates may occur manually or automatically. Manual updates occur when a healthcare practitioner spontaneously requests a directory update (rare). Typically, directory updates occur automatically. An *update-directory* daemon is associated with each information system. The *update-directory* daemon runs regularly (e.g. every night) retrieves recent information from the information system to update the directory.

  - **Directory Queries**

    A treating physician logs on into the directory server through a web browser and requests information on a specific patient. A patient is identified in the directory with a set of attributes (tentatively: firstname, surname, parentname, birthdate, placeofbirth, id_number). The query is translated into LDAP, forwarded to the directory and the directory returns a set of links matching the selection criteria. Each link facilitates a connection to the directory (to refine the query) or the information system that maintains the relevant patient record segment.
- Different types of **services** are to be provided by CPDD. These services are in direct relation with, and will actually utilise respective **InterCare *common products***.

  In order to accomplish the tasks that are relevant to the aforementioned scenario the following set of services has to be available:

  - Local Information System (Legacy or Feeder System).
  - A service, which will enable both the requester and the performer to publish, navigate and inspect patient relevant medical information. The ***IC-EPDS*** and ***IC-PIDRM*** could provide such service.
  - A service which will enable the requester to locate the most appropriate medical functional unit to refer the case. The ***IC-HIS*** could provide such service.
  - A service which will secure the exchange of sensitive medical information. The ***IC-SS*** could provide such service.

4.4 Class Model

The object class model of the Crete CPDD application consists of five (5) classes packages:

1) **General Classes Package**
2) **HealthCare Record Segment & Medical Encounter Classes Package**
3) **Patient Record Classes Package / Patient**
4) **HealthCare Facility & Resources Classes Package**
5) **Referral Classes Package**

The **General Classes Package** is the main package in which all the classes of the CPDD Crete demo application is specified, using UML static structured diagrams. This package is in direct relation and communication with the other classes packages.

The **HealthCare Record Segment & Medical Encounter Classes Package** realises and specifies all the information items needed for storing, organising, accessing and retrieval of related medical information. Especially, information related to the exact location of the related medical encounter information is specified and accordingly structured.

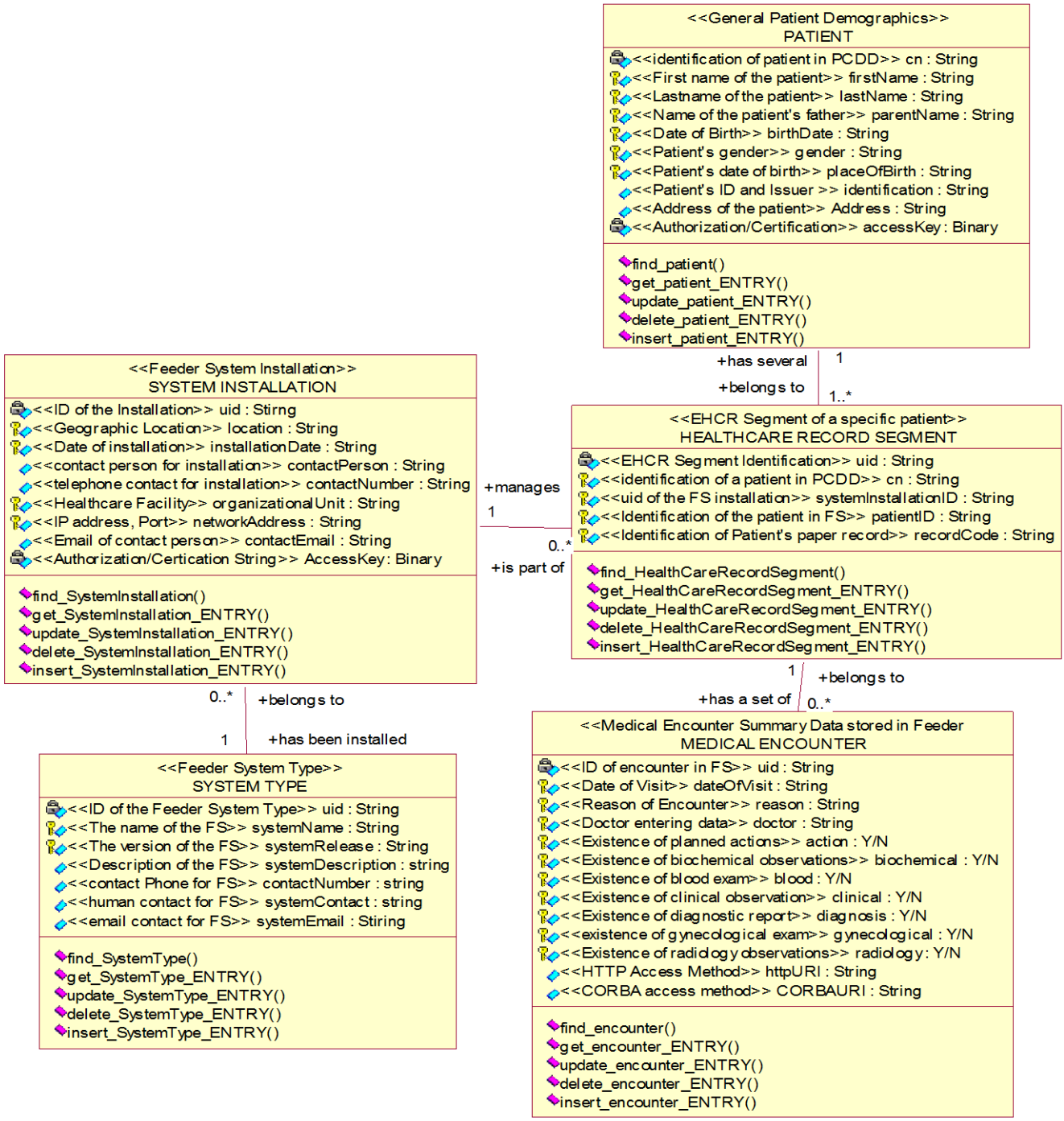
The **Patient Record Classes Package** includes all the patient information items used to specify a particular **patient**. The various items are related with respective items of the previous packages.

The **HealthCare Facility & Resources Classes Package** specifies all the needed info items for recording and retrieving patients' clinical info.

The **Referral Classes Package** specifies and structures the needed information for establishing the communication between CPDD and general professionals, towards patients clinical information retrieval.

All the above packages are related, and actually instantiated, by classes packages realising the specific distributed **clinical information systems** to be integrated by the CPDD services. For reasons of manageability, we did not include the classes description of these packages. In the course of WP4 local applications development, the complete specification of these packages (also in terms of UML class diagrams and models) will be delivered.

Diagram: General Classes Package



The figure above shows the UML class diagram of the entry types in the directory and displays attributes, methods, and relations among entry types. There are three types of attributes namely, private, protected, and public. A lock next to an attribute indicates a private attribute, which is an identifying attribute for that entry. A key next to an attribute indicates a private attribute, which cannot be updated. All other attributes can be updated through the CPDD *update* interface. Note also, that all attributes may take multiple values (multivalue attributes). As already mentioned, there are five entry types in the information model of the directory, namely *Patient*, *HealthcareRecordSegment*, *MedicalEncounter*, *SystemType*, and *SystemInstallation.*

Diagram: HealthCare Record Segment & Medical Encounter Classes Package

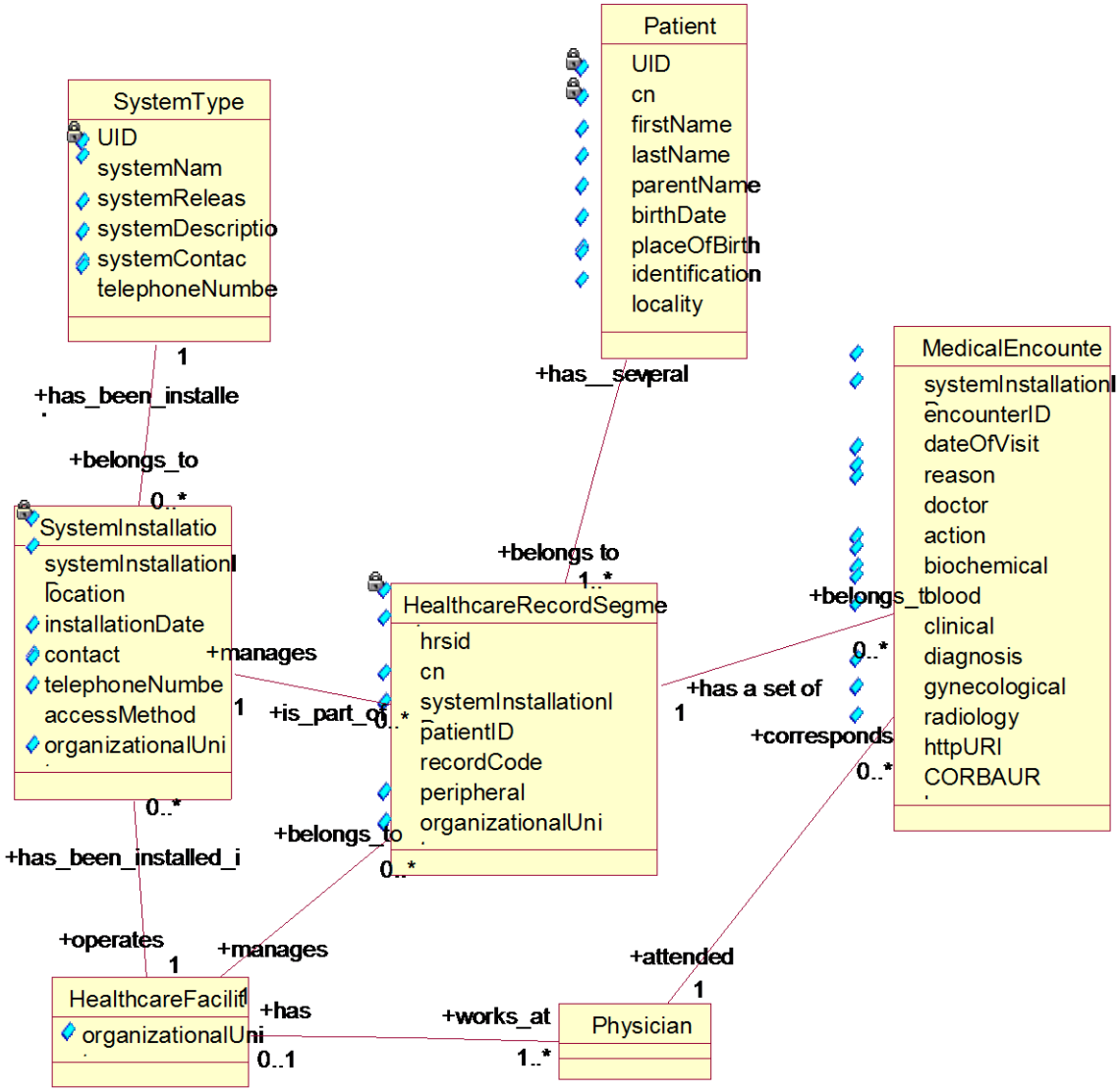


Diagram: Patient Record Classes Package / Patient

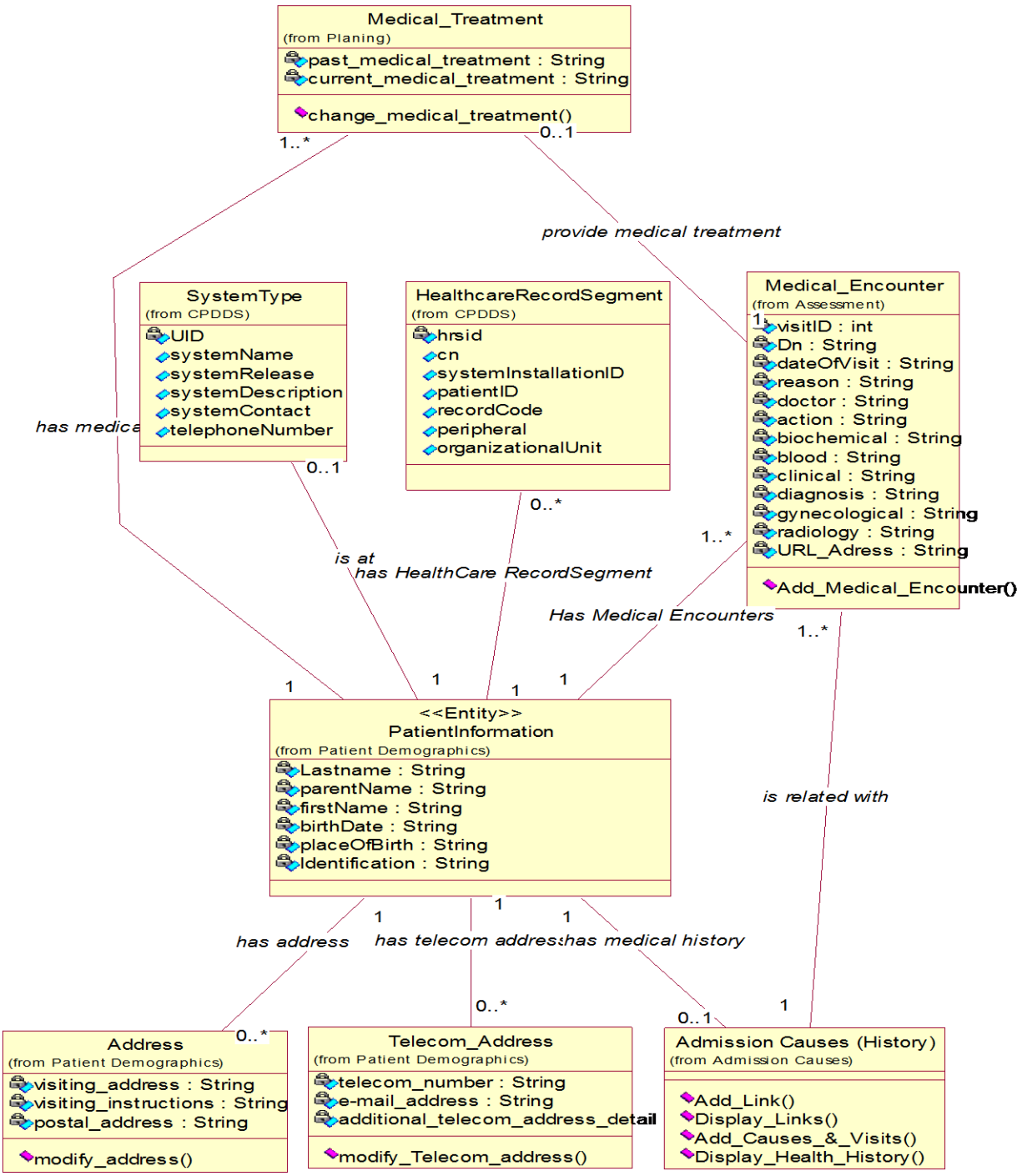


Diagram: HealthCare Facility & Resources Classes Package

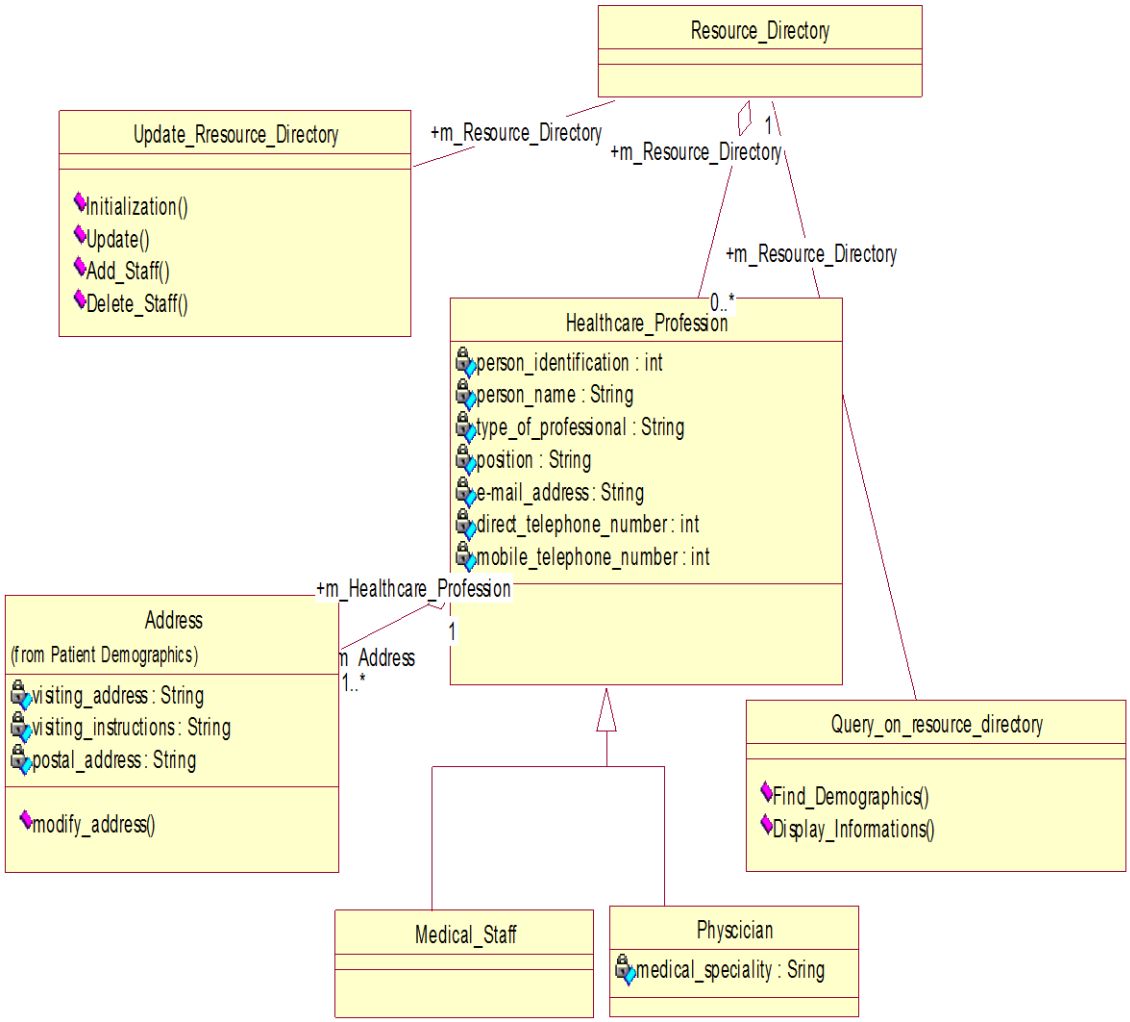
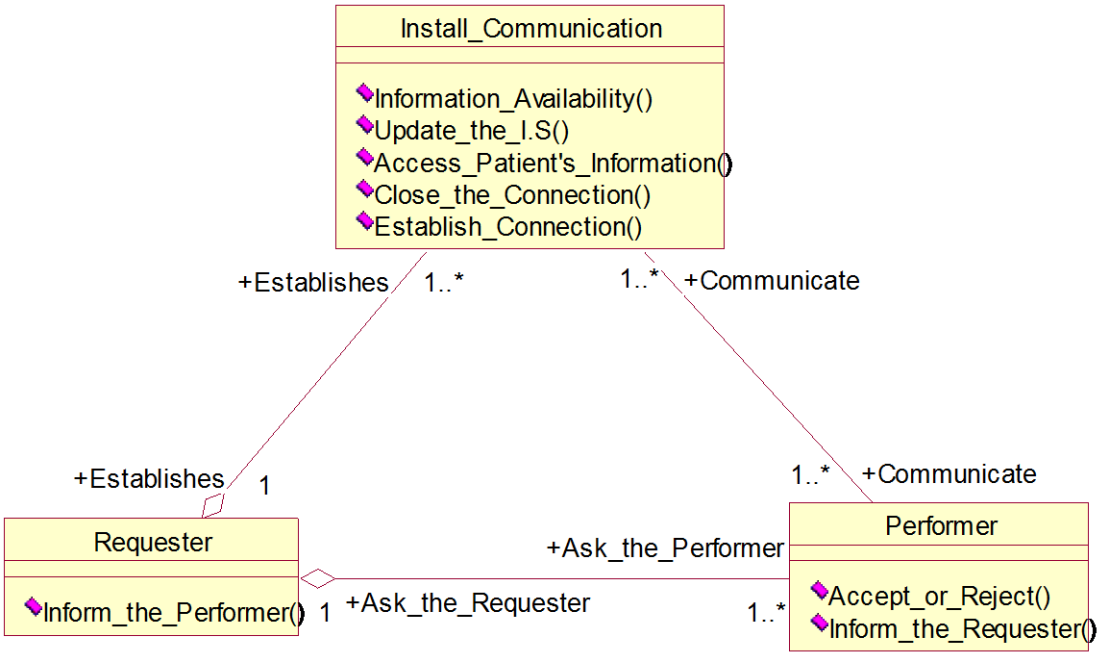


Diagram: Referral Classes Package



4.5 Implementation Plan

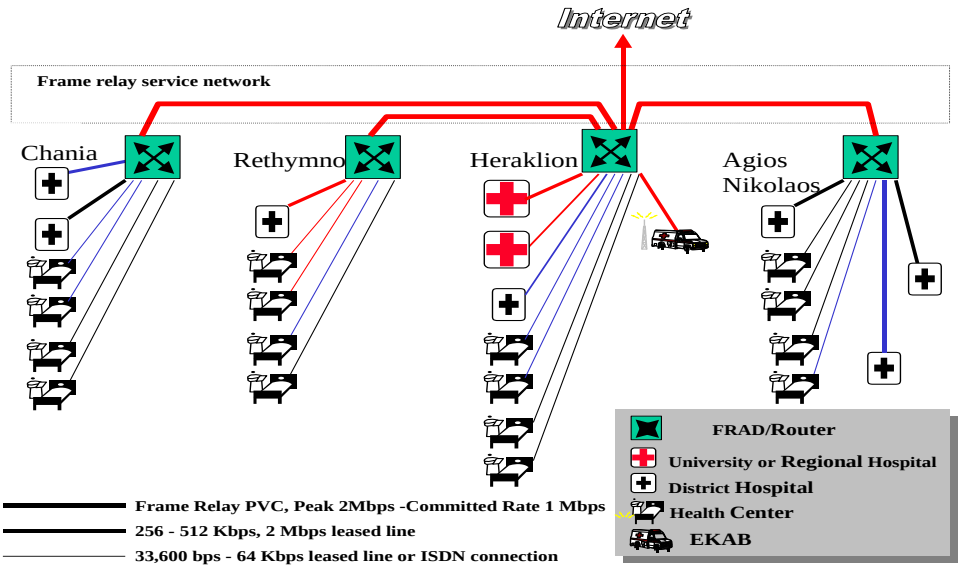
The implementation of the InterCare services in Crete demonstration site will be made in an iterative way during 1999, with a continuous communication with end-users. The overall development / implementation plan is divided into three phases.

1. ***1st Phase development:*** The clinical information systems to be integrated with the CPDD services will be developed. Some of them are already in a prototype version status (Paediatric Surgery CIS, Primary Health Care CIS, Health Emergency CIS, Laboratory CIS, Radiology CIS) and other are under development (Cardiology CIS). Furthermore, a Teleconsultation CIS is under development; its integration within CPDD will permit teleconsultation services over the same CPDD infrastructure. Furthermore, in this phase the basic network infrastructure (as will be provided by FORTHnet associate partner) will be accomplished. See the figure below for a layout of the overall **Crete HealthCare network infrastructure**, provisioned in the context of the InterCare project and other ongoing national and European projects.
2. ***2nd Phase development:*** The various Clinical Information Systems will be integrated within the CPDD middleware. Here the main CPDD development stream will take place. There is already a first prototype/demo version of CPDD services. Further refinement of the clinical information systems will also take place in this phase.
3. ***3rd Phase:*** The various common InterCare products will be utilised within the CPDD services in Crete. In particular, IC-EPDS, IC-PIDRM and IC-SS will be utilised.

The extensive modelling that has been carried out will be refined further in order to come up with clear-cut and comprehensive models of the Crete applications. These models will be used in further extensions of the Crete InterCare demonstrator towards an integrated HealthCare network in Crete.

Demonstration and evaluation will take place during the remaining InterCare project period in year 2000.

Crete HealthCare network infrastructure



4.6 Architectural aspects

Information technology planners use the term architecture to refer to abstract technical descriptions of systems, which are conceptually based and provide the basic framework for the creation, movement, distribution, use, and management of information. A reference architecture model describes a system in terms of basic functional elements and the interfaces among them. It clarifies where protocols need to be defined and identifies groups of functionality, without implying a specific physical implementation. The reference architecture for the HII shown in Figure 13 guides the development of a health-telematics network for the provision of integrated services. It provides a general framework in which healthcare-related information systems may be integrated to provide media-rich services to healthcare professionals, social workers, and the public. It is important to note that the HII provides a conceptual roadmap, since it does not imposes any execution platform, and consists of a vast conglomeration of autonomous information systems and supporting services.

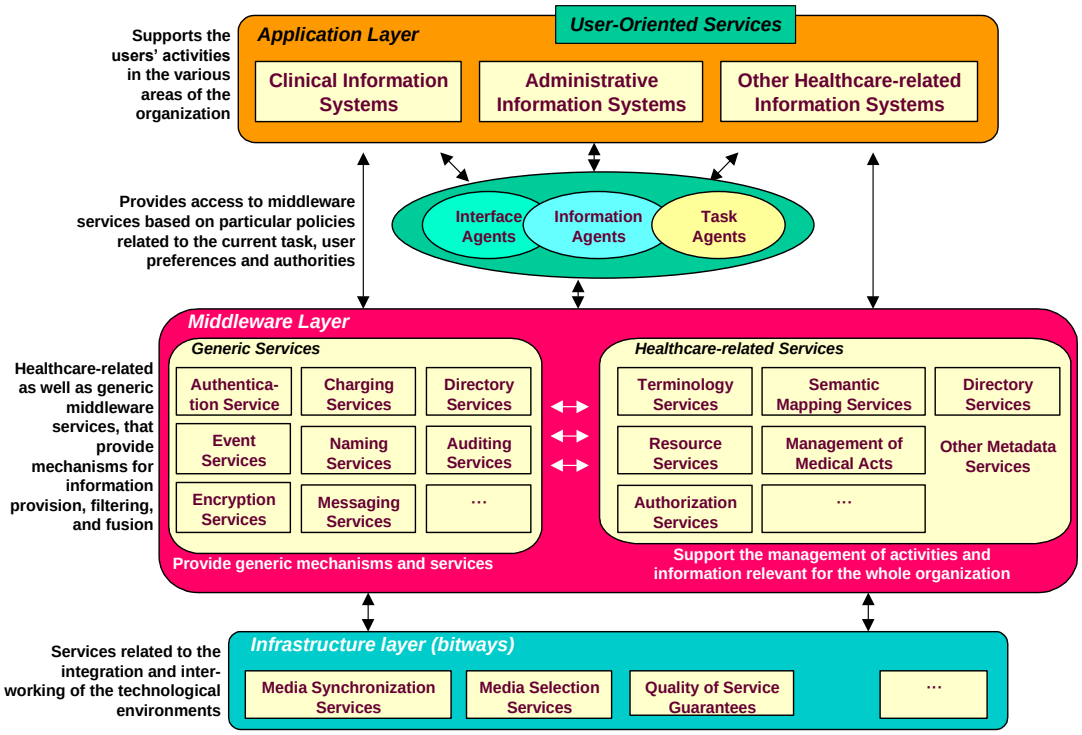


Figure 13: Crete InterCare Reference Architecture.

Users are primarily interested in information processing applications, which they may own or gain access to as end-users via communications networks. These services are 'enabled' by other underlying, transparent services provided by information and network service providers. Applications and enabling services employ certain information processing services and systems for data transport, which may be distributed throughout the existing HII. Thus, the HII consists of three basic components: applications, enabling (or middleware) services, and network infrastructure. At the bottom of Figure 13, the *Infrastructure layer* provides services that are related to the integration and inter-working of the technological environments. The *Application layer*, through the User-Oriented Services, supports the users' activities in the various areas of the organisation.

In the middleware layer of the reference architecture reside generic and healthcare-specific middleware services. Generic middleware services support the applications with general-purpose facilities, which are common to any information system in any type of business domain. *Directory* services based on X.500/ LDAP provide white page directories (i.e. information on healthcare professionals), as well as yellow page directories (i.e. information on healthcare facilities and services). *Certification* services provide certificates to various actors and enable the evolution of a

trust infrastructure. *Encryption* services support the secure data-transfer over communication networks. *Charging* services enable financial transactions, while *auditing* services facilitate the tracking of access profiles on information objects providing traces of activities concerning their creation, modification, deletion, etc. *Naming, indexing,* and *searching* services are essential for the efficient location and retrieval of information objects. *Event* services support de-coupled communication between information objects through the delivery of event notification messages. Finally, *messaging* and *collaboration* services allow the peer-to-peer synchronous or asynchronous communication of information objects based on standard communication protocols.

Healthcare-specific middleware services support applications with services related to activities of the healthcare domain. Regional *resource services* provide information on the availability of physical resources such as hospital departments, diagnostic modalities, mobile emergency units and their characteristics. Educational resources, such as public health information and access to digital medical libraries, are also enabled by resource services. Access to and use of the medical data maintained by healthcare information systems in different authorisation domains, is subject to strict confidentiality policies. The enforcement of these policies necessitates the synergy of healthcare-specific services, such as *authorisation* services, with generic services such as security and certification services. Flexible domain-specific *authorisation services* based on generic security and certification services conform to the evolving legislation regarding telemedicine. *Terminology-mapping services* map one terminology standard to another. *Semantic mapping services* provide mediation and translation services that facilitate the mapping of heterogeneous information sources into a global domain model. *Management of medical acts* enables the co-ordinated execution of healthcare-related tasks, facilitate functional integration and co-ordinate medical procedures in terms of quality, throughput, and reliability.

Agent-based technology provides access to generic and healthcare-specific middleware services based on particular policies related to user preferences, tasks, and authorities. Agents in the form of mediators and facilitators have been widely employed as a means of providing assistance to users in the complex tasks of information tracking, filtering, and fusion. *Interface agents* interact with the user to receive user specifications and deliver results. They acquire, model, and utilise user preferences to guide system co-ordination in support of the user's tasks. *Task agents* help users perform a task by formulating problem solving plans and carrying out these plans through querying and exchanging information with other agents. *Information agents* provide intelligent access to a heterogeneous collection of information sources.

Several clinical information systems have already been developed and utilised as autonomous networked applications. System conformance is measured by the capability of an information system to exploit, when necessary, one or more of the common components specified in the reference architecture. The degree of conformance to the reference architecture indicates the capability of the information system to be integrated, from both the functional and information viewpoint with the rest of the information systems. Appropriate extensions or modifications to these information systems should be carried out, so that they are seamlessly integrated to the proposed reference architecture.

Below, Figure 14, the general CPDD services architecture is visualised with all the respective clinical information systems, provisioned for integration within the CPDD middleware.

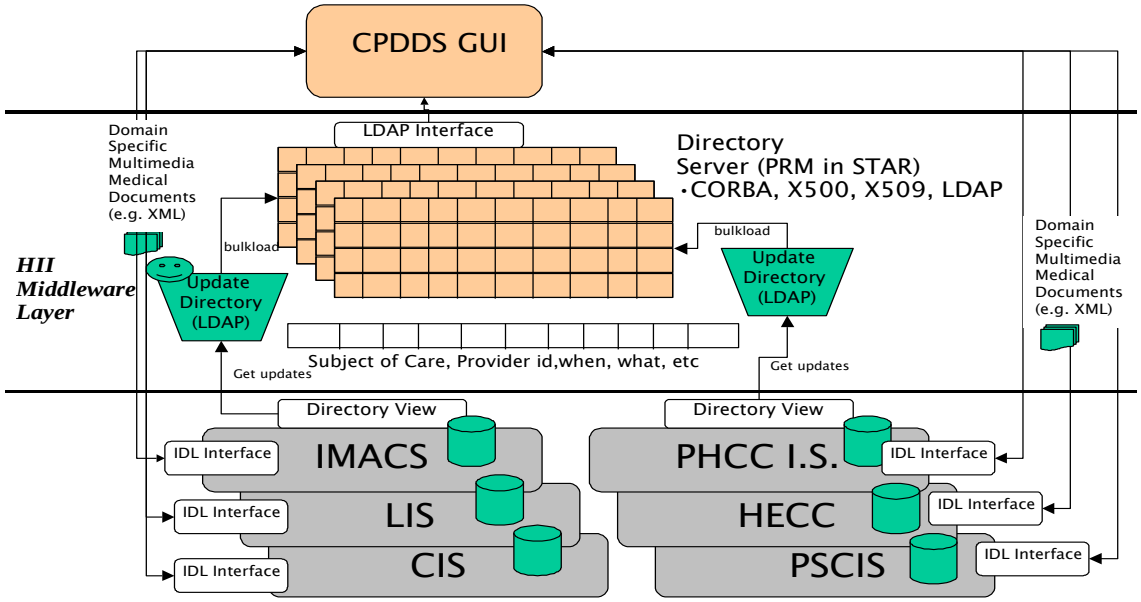


Figure 14: Overall CPDD architecture

5. Irish Demonstrator Design Specification

5.1 St James's Hospital Demonstrator

5.1.1 Business Objectives

A major change has taken place within the Dublin region. The three small city centre FDVH Hospitals (Meath, Adelaide, National Children's Hospitals) have recently moved to a new Tallaght Hospital (600 beds) 12 km south-west of Dublin. Many of the patients who currently attend St. James's Hospital will transfer to the Tallaght Hospital and likewise many FDVH patients will transfer to St. James's. Both hospitals are under the Trinity College Dublin Medical School umbrella so it is obvious that considerable addtional communication facilities will be required.

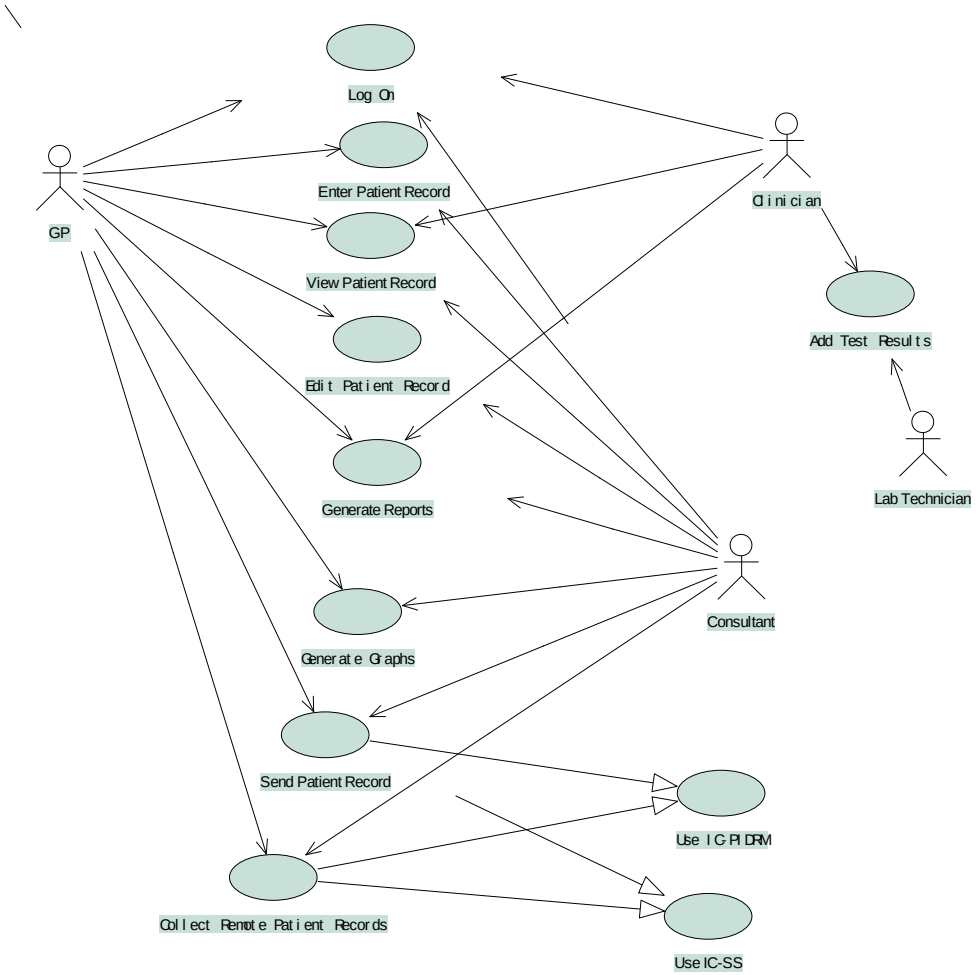
Patients with diabetes in Dublin have the option of arranging their care with preferred GP's and hospitals and are likely to make their choice based on convenience, or their perception of quality of services available. In addition, more than one GP or consultant may be responsible for the care of a diabetic patient. A diabetic patient may be the concern of a number of consultants, e.g. a renal consultant, dietician, cardiologist etc. It is important that each healthcare professional responsible for this patient in this shared care situation, has access to their patients data, especially lab data.

For example, a consultant in one hospital may refer a diabetic patient's case to a consultant in another healthcare institution. The second consultant will require access to the patient's lab data. There is a need for an InterCare environment to make this data available.

Another common scenario, occurs when a GP in the catchment area of the hospital, sends a patient sample to St. James's hospital pathology lab for investigation. A GP may wish to make reference to a patients previous results carried out by another consultant. Instead of receiving the results over the telephone or by post, it would be preferable to have access to this lab data through th e Diabetes Patient Manager system. It is expected that InterCare will fulfil this need.

5.1.2 Use Cases

5.1.2.1 Use Case Diagram



5.1.2.2 Use Case Descriptions

5.1.2.2.1 Actors

GP
The General Practitioner responsible for the health care of the diabetes patient.

Consultant
The consultant, referred to by the GP, in charge of the diabetic care of the diabetes patient.

Lab Technician
A laboratory technician carrying out special tests / investigations on diabetic samples.

Clinician
Authorised application user representing the many clinics that may have a special interest in the diabetic patient e.g., lipid clinic, eye clinic, diabetic clinic, diabetic day clinic etc.

5.1.2.2.2 Use Case Views

Use Case: Log On
An authorised user logs on to the diabetes patient management system.

Use Case: Enter Patient Record
Patient data is entered into the Diabetes Patient Manager, this covers initial registering of a patient, entering additional demographics, entering examinations results and comments etc.

Use Case: View Patient Record
The application user selects a patient from a list and can view associated data for that patient for which authorisation roles allow.

Use Case: Edit Patient Record
The application user may modify selected patient data for which authorisation roles allow.

Use Case: Generate Reports
Authorised application user may generate reports for selected patients to a previously defined specification.

Use Case: Generate Graphs
Authorised application user may generate graphs for selected patients to a previously defined specification.

Use Case: Send Patient Record
The GP wishes to make available the information on a specific patient to a specific consultant at a remote location.

Use Case: Collect Remote Patient Records
The authorised application user wishes to download results for his patients from a remote laboratory or clinic site.

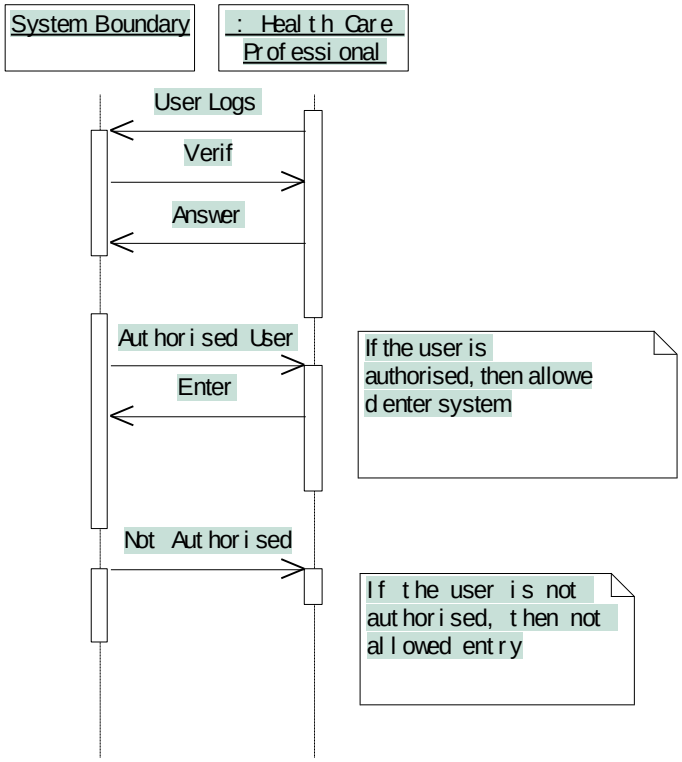
Use Case: Add Test Results
An authorised application user may add laboratory / clinical results to the Diabetes Patient Manager Database.

Use Case: Use IC-SS
When patient data is being sent from one site to another via the internet / extranet the IC-SS will be used to enforce that this data is secure.

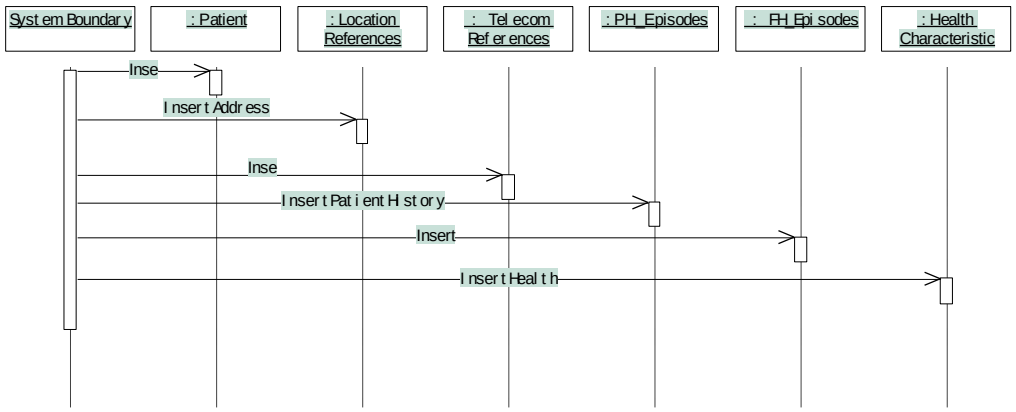
Use Case: Use IC-PIDRM
Patient data held on more than one site may have different unique patient id, the IC-PIDRM is used to link / merge these patients data.

5.1.2.2.3 Sequence Diagrams

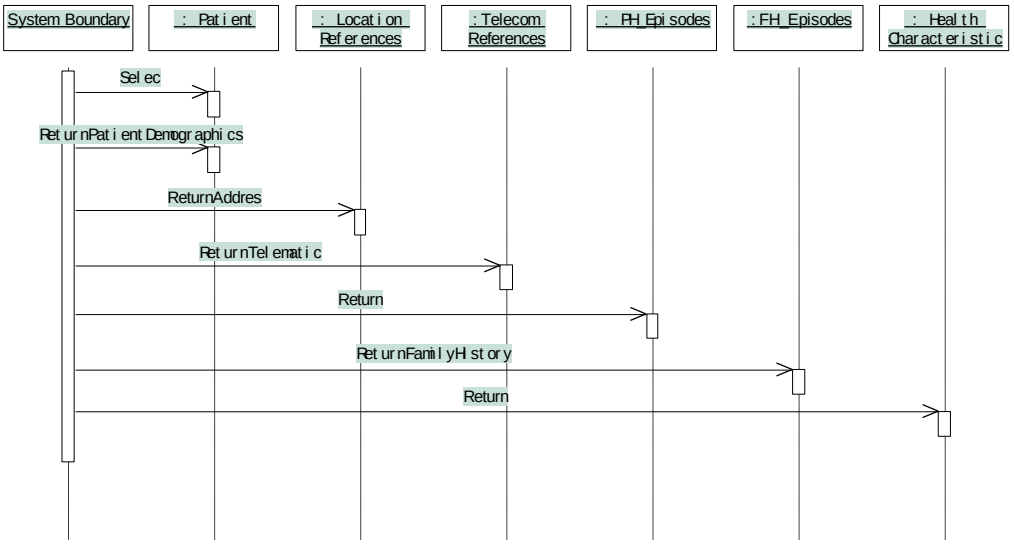
Use Case: Log On



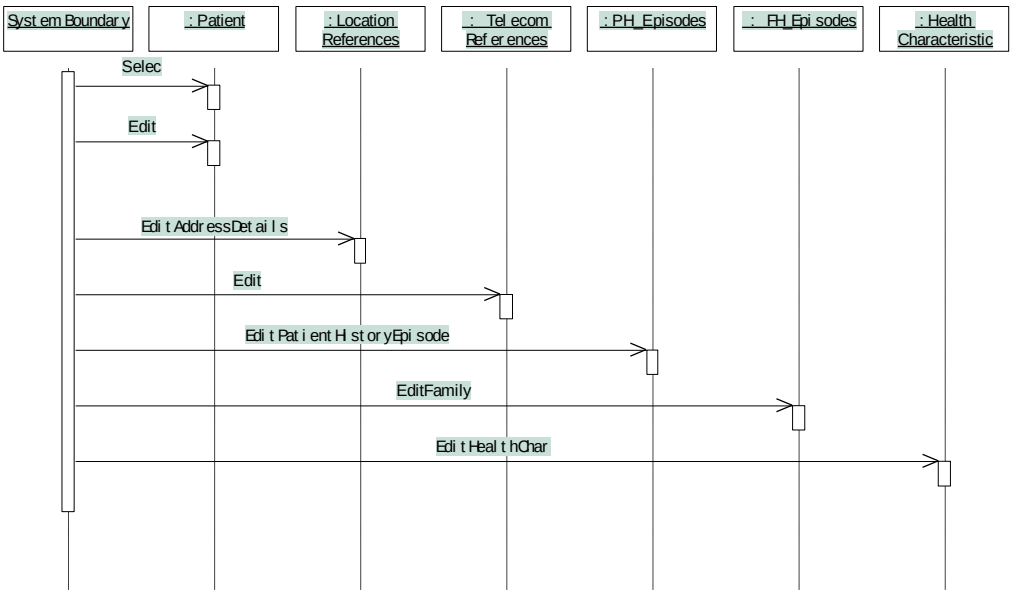
Use Case: Enter Patient Record



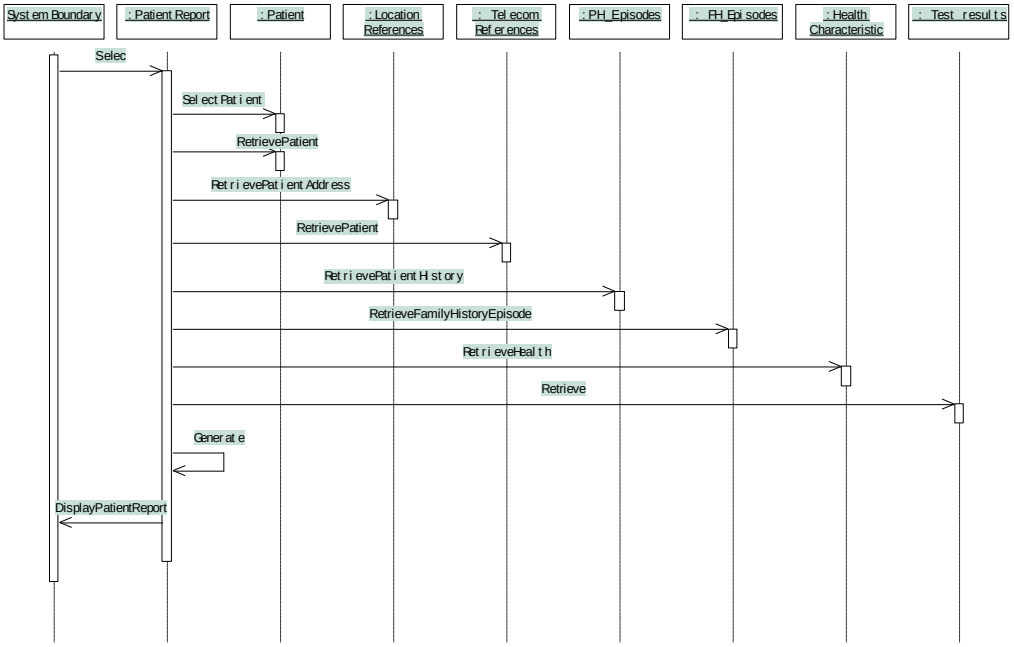
Use Case: View Patient Record



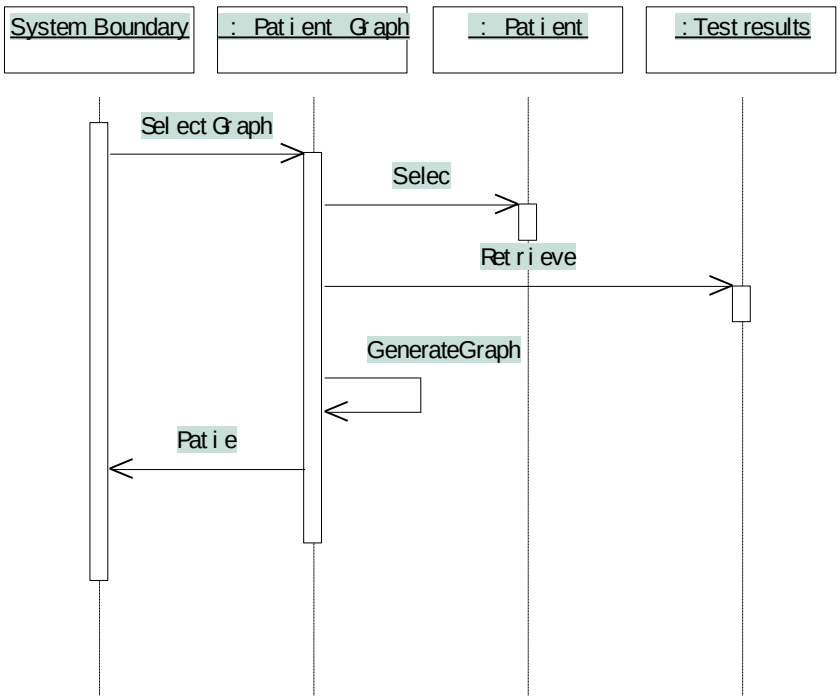
Use Case: Edit Patient Record



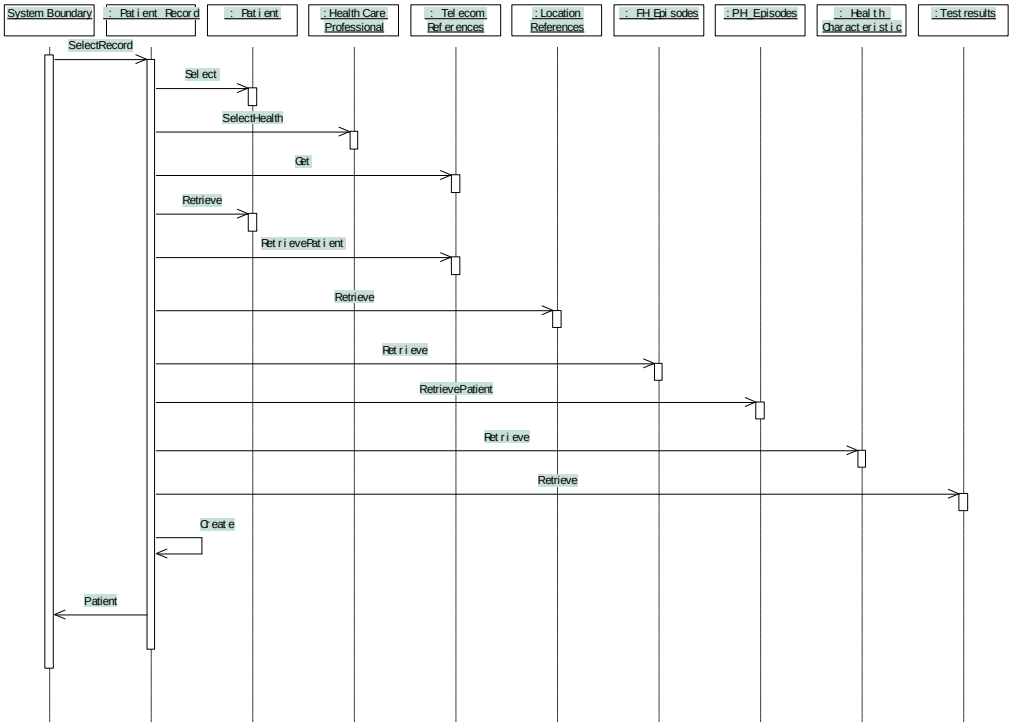
Use Case: Generate Reports



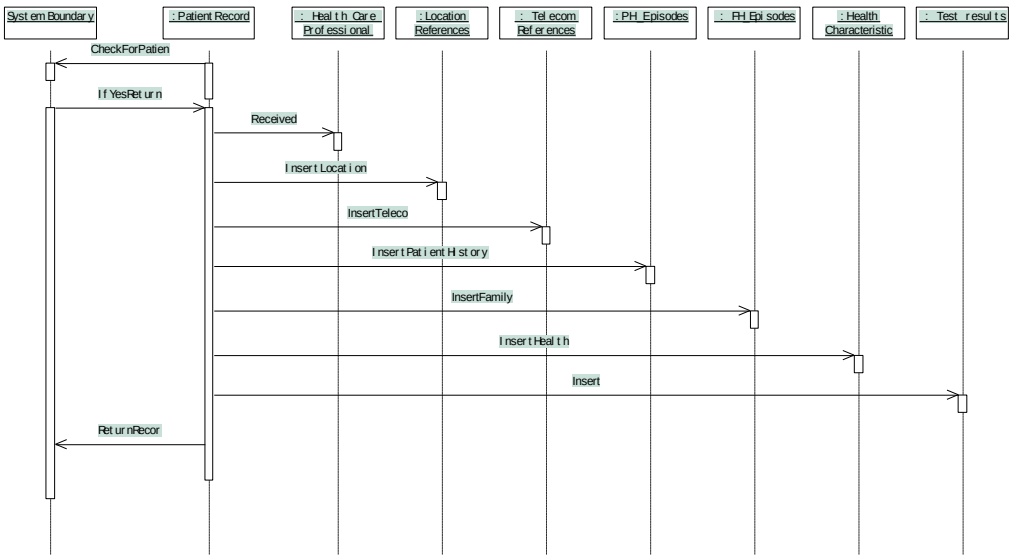
Use Case: Generate Graphs



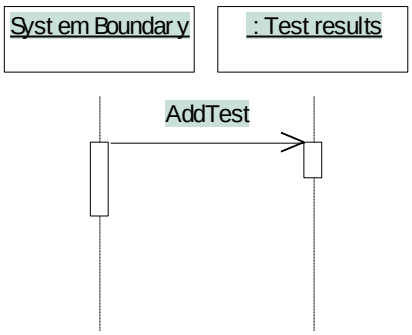
Use Case: Send Patient Record



Use Case: Collect Remote Patient Records

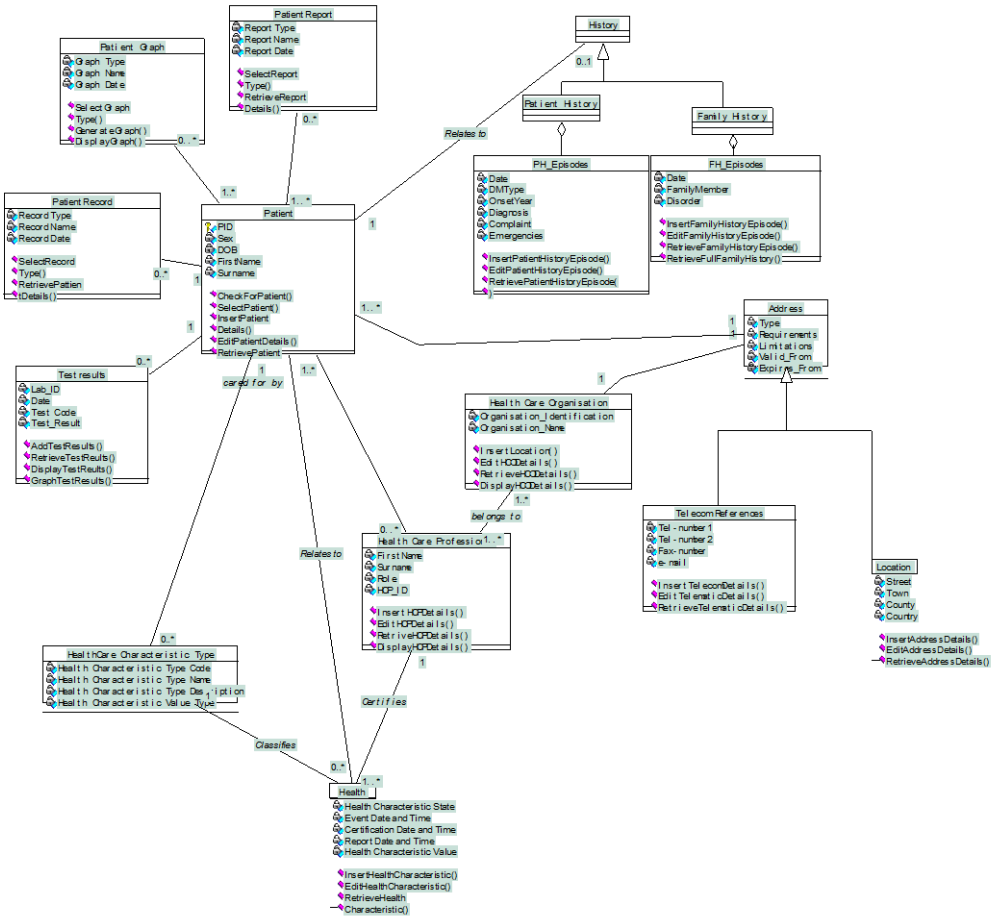


Use Case: Add Test Results



5.1.3 Object Class Model

5.1.3.1 Class Diagram



5.1.3.2 Class Descriptions

Patient

Person under diabetic health care.

Operations:

CheckForPatient( )
SelectPatient( )
InsertPatientDetails( )
EditPatientDetails( )
RetrievePatient( )

Patient Report

Report designed by a health care professional, giving a predefined report for a selected patient or list of patients.

Operations:

SelectReportType( )
RetrieveReportDetails( )
DisplayReport( )

Patient Graph

Graph designed by a health care professional, giving a predefined graph for a selected patient or list of patients.

Operations:

SelectGraphType( )
RetrieveGraphDetails( )
DisplayGraph( )

Patient Record

A record containing a patients full set of details.

Operations:

SelectRecordType( )
RetrievePatientDetails( )

Test Results

Results for tests carried out on a patient sample in the pathology laboratory, found on the LIS.

Operations:

AddTestResults( )
RetrieveTestResults( )
DisplayTestResults( )
GraphTestResults( )

Health Characteristics

Characteristic of a patients health status.

Operations:

InsertHealthCharacteristic( )
EditHealthCharacteristic( )
RetrieveHealthCharacteristic( )
DisplayHealthCharacteristic( )

HealthCare Professional

This is a GP, Consultant, Clinician or Lab Technician with a special interest in a diabetic patient.

Operations:

InsertHCPDetails( )
EditHCPDetails( )

RetrieveHCPDetails( )
DisplayHCPDetails( )

HealthCare Organisation

This is a health care institution with a special interest in a diabetic patient, this could be a ward, hospital, clinic, lab etc.

Operations:

InsertHCODetails( )
EditHCODetails( )
RetrieveHCODetails( )
DisplayHCODetails( )

Address

Contains location address and telecom details of patient, healthcare professional and organisations.

Operations:

InsertAddressDetails( )
InsertTelematicDetails( )
EditAddressDetails( )
EditTelematicDetails( )
RetrieveAddressDetails( )
RetrieveTelematicDetails( )

History

Contains history of the patients and the patients family's medical history.

Operations:

InsertPatientHistoryEpisode( )
InsertFamilyHistoryEpisode( )
EditPatientHistoryEpisode( )
EditFamilyHistoryEpisde( )
RetrievePatientHistoryEpisode( )
RetrieveFamilyHistoryEpisode( )
RetrieveFullPatientHistory( )
RetrieveFullFamilyHistory( )

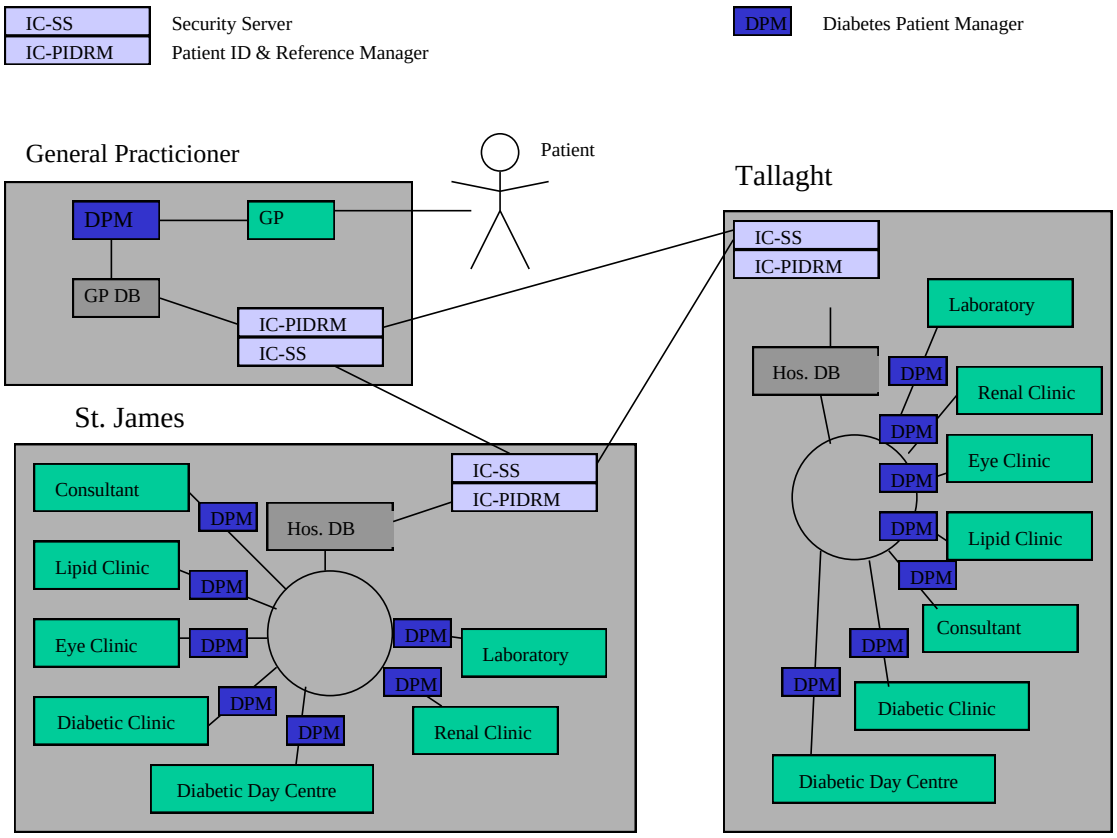
5.1.4 Implementation Plan

This is the implementation plan of the St James's / Tallaght Hospitals portion of the Irish Demonstration Site.

The users of the demonstration will be GP's, consultant and clinicians with in health care institutions. Within a hospital site there will be a distributed Diabetes Patient Manager system throughout the hospital with a central data base, accessible to authorised user each with individual user names and passwords. Each user will have his/her own permissions on the system. The GP system will have it's own local database. Each site will have a security server (IC-SS) and patient ID and reference manager (IC-PIDRM).

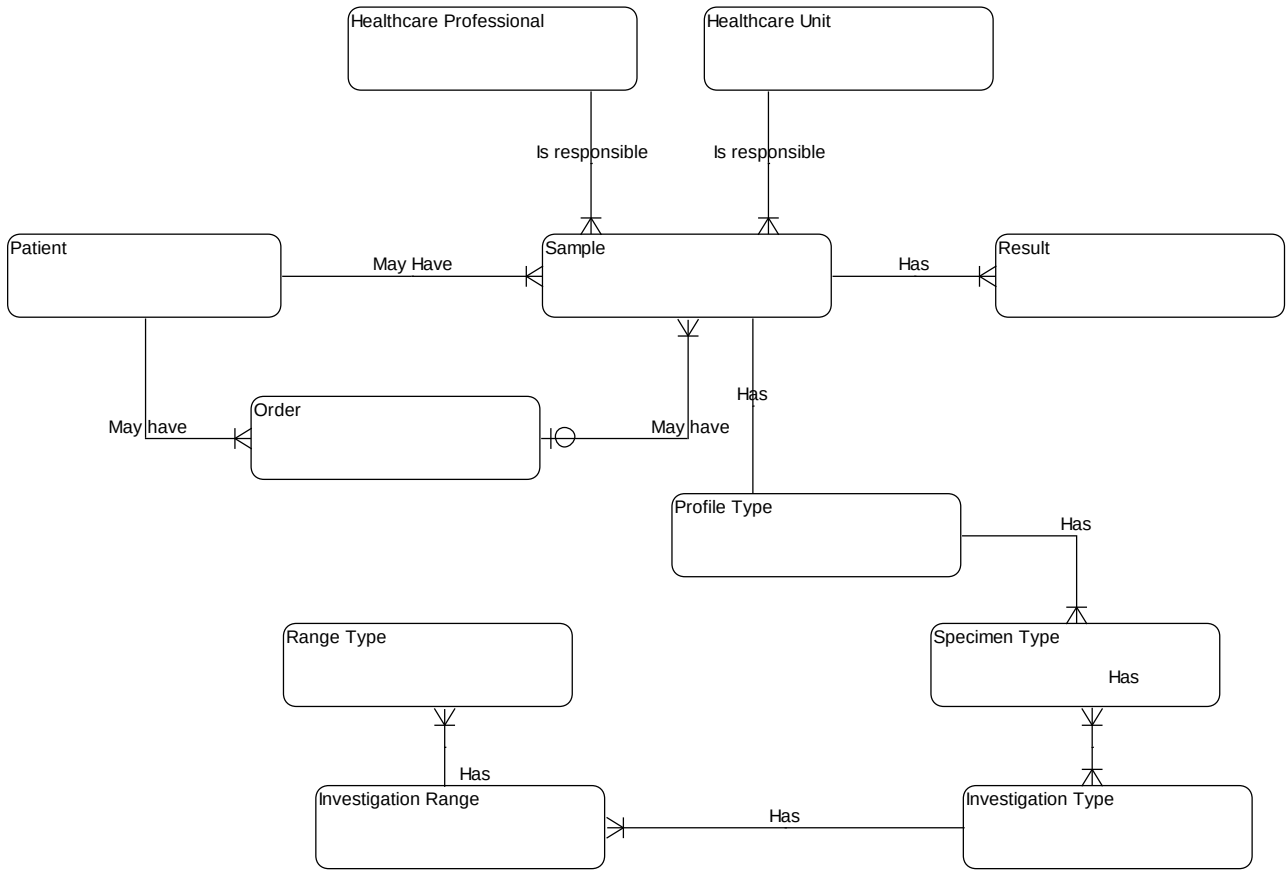
5.1.5 Architectural Aspects

5.1.5.1 Overview of Implementation Plan



5.1.5.2 Schema for SQL Server

The following schema is the design for the backend SQL Server database.



5.2 Eastern & North Eastern Health Boards' Demonstrator

5.2.1 Business Objectives

The National Health Strategy for Ireland demands the sharing and integration of patient information across the range of services provided by many different centres and agencies. The key theme is the provision of seamless care to individuals in the most appropriate and cost effective setting. This can be problematic when the range of care givers may be diverse and located in geographically separated locations. InterCare can significantly contribute to the attainment of the objectives of the aforementioned Health Strategy.

The primary objectives of the InterCare Project in the Eastern & North Eastern Health Board's will be to:

- Integrate and exploit the key deliverables from parent framework developments (security architecture, "Yellow Pages") into a combined strategy with project partners.
- Assess other project deliverables, review security mechanisms and incorporate the core data set into the standards for demonstrations.
- Bring together the isolated islands of information in Hospitals, GP's and Community Services into a single seamless environment. This will be achieved through the use of Internet technologies.
- Build on results of local projects and developments.
- Integrate with national initiatives.
- Facilitate inter organisation approaches to care in the context of revised service and organisational structures.

We will be developing the functionality of the Yellow Pages. These pages will contain information about the availability of care in the regions, such as –

- Health & Social Services available with contact names, addresses & numbers
- Public information bank on common medical complaints
- Information on the health & social status of the population in the regions
- Contact point for further information
- This site will also be mirrored with our Intranet site so as staff of the organisations can have access to this wealth of information

The Yellow Pages will be accessible from GP practices, pharmacies, health centre sites, public information points and other internet accessible points.

The Yellow Pages will also be used to give health care professionals access to our new Medical Card System, which is used to determine eligibility of a client to free medical care. This model will then be used for health professionals to gain access to other health care systems through our Yellow Pages.

5.2.1.1 Metrics

The efficiency of health professionals will increase, because less time will have to be spent in gathering information on health & social services available.

The Yellow Pages will be a comprehensive service directory with access to a patients status on eligibiltiy to free medical care. Clients can access information in a more flexible way with 24 hours access. Information will be right up to date.

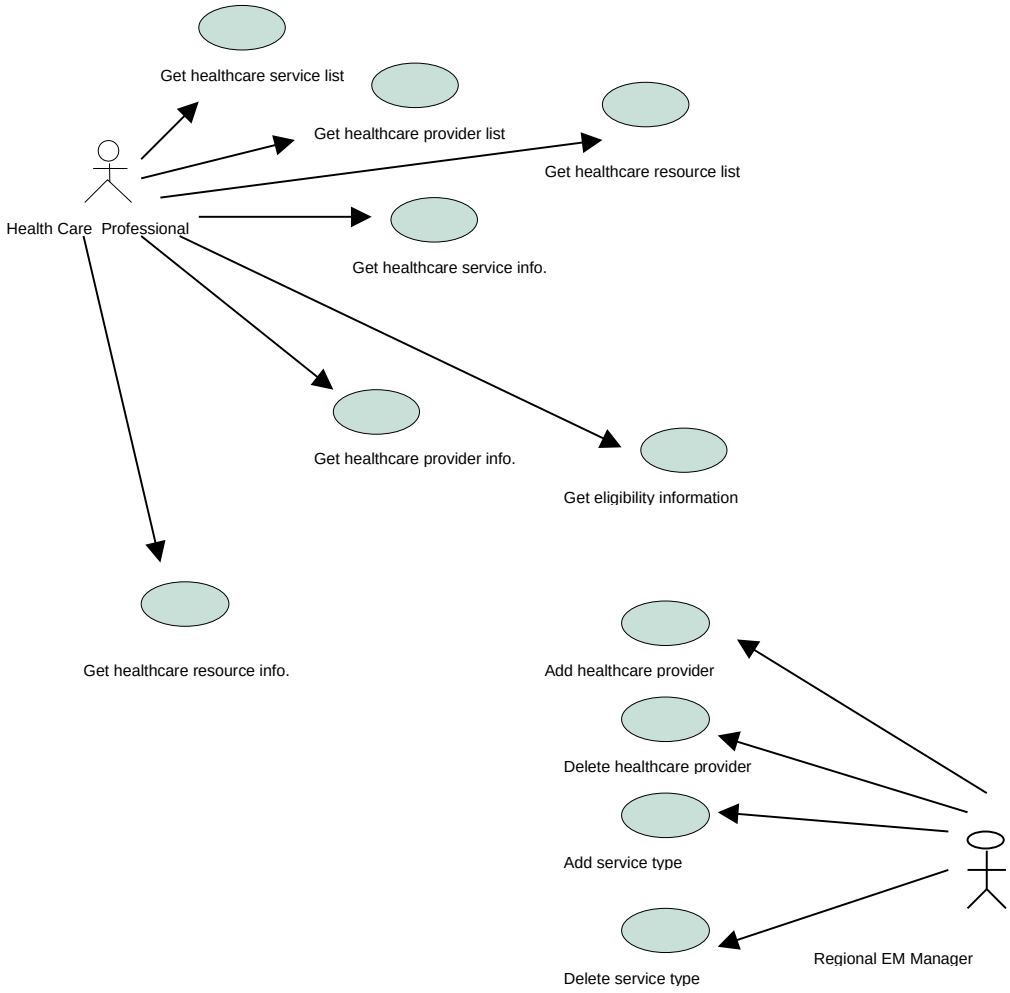
5.2.2 Use Cases & Scenario's

Scenario 1 – Access to Health & Social Services Information

Currently a client/health care professional might have to ring or visit a health care site to gain access to health & social care information – this is timely & costly and leads to inefficiency. We will demonstrate how this information will now be available at the push of a button.

Scenario 2 – Access to an Healthcare Application (Medical Card System)

When a client presents themselves to a health care professional (HCP), the HCP needs to know if they are eligible to free health care or not. This can be a timely process which will now be improved by means of an on-line access to the system via the Yellow Pages.



5.2.2.1 Actors

GP

The General Practitioner is the gatekeeper to our services.

Other Healthcare Professionals

Providers of healthcare services.

Clients

The user of our services.

5.2.2.2 Use Case Views

Get healthcare service list

Available service types are displayed.

Get healthcare provider list

Available providers according to the given search criteria are displayed.

Get healthcare resource list

Contact and availability details for the selected provider are displayed.

Get healthcare service info.

The detailed description of the service selected are displayed

Get healthcare provider info.

Professional skills details for the selected provider are displayed.

Get healthcare resource info.

Available providers for the service type selected are displayed

Get eligibility info.

The status of the clients eligibility to free healthcare is displayed.

Add healthcare provider

New provider added.

Delete healthcare provider

Provider deleted.

Add service type

New service added.

Delete service type

Service deleted.

5.2.3 Implementation Plan

This is the implementation plan of the Eastern & North Eastern Health Board.

The users of the demonstration will be Clients, GP's and other healthcare professionals.

The model will be built around the IC-SS for secure access and the IC-HIS for building the Yellow Pages for our services.

6. Italian Demonstrator Design Specification

6.1 The Lombardia Region Health In☆net - General Introduction for Regional Demonstrator in InterCare

6.1.1 Strategic View: approach and treats in designing the system

Even if the architecture of a new networked system should be designed accordingly to openness, distribution and integration requirements, it is necessary to consider all the conditions imposed both by organisational situations and by the choice of the technology itself, always in evolution.

These conditions become the limits of the actual system; the elimination of such limits could imply, in the future, a possible review or re-engineering of the current architecture.

In the following paragraphs are reported the most important topics that must be considered.

6.1.1.1 Laws, rules and regulations

The actual Italian ***law on privacy "675/96"*** gives clear indications on the possibility to exchange or treat personal information. These indications must be considered as complementary to the security components in the telematics data management and directly integrated into them.

Regulations about juridical validity of ***electronic signature*** can simplify the design of the system, taking out a lot of complex organisational controls in other way imposed on the matter. The electronic signature is taken as the standard tool for user's (professionals and patients) identification in all the transactions/transmissions in which there is necessity of certified and secure knowledge of the executor.

Electronic signature can also be associated to the data ownership and juridical responsibility on them, without any dispute on i.e. definition of prescriptions, possible modification by authorised personnel etc.

The recorded track (logging) of all the events happened during the various flows in the system provides a certain mechanism supporting the necessary controls.

The use of the ***smart card*** (microprocessor) must be foreseen with the aim of providing a "patient/citizen services card", and must respect the requirements of multi-functioning with associated multi-parties for the management of peculiar services.

6.1.1.2 Organisation

The ***request and provision process*** in healthcare services is considered the core process of the global system and due to its impact on the generation of costs, implies a particular control and a particular management of the flow of information associated in the network.

The system must be designed considering also the parallel presence of ***paper based information/documents***, that are alternative to the ones in electronic format. This existence is related to the complexity of the change to a paperless environment, that could take several years.

Moreover, administrative control procedures require actually the presence of prescriptions for health services on paper.

Continuity of the service implies technical solutions and physical distribution on the territory of components/replications of the system that must be carefully designed.

From the functional point of view, the design of the system infrastructure must allow the identification and the attribution of ***responsibility*** on roles, duties and rights of the various actors, on management of the "objects", on control of events etc. related to the new system.

6.1.1.3 Security

The necessary activation of the wide networked system (we can call it "Virtual Network") based on infrastructures that are the same of the public network, imposes to the design phase the adoption of clear and strong criteria for assuring to the system the following vital characteristics:

- Privacy of information;
- Secure identification of sender/receiver (users);
- Protection of data during transport;
- Protection from external intrusion

All these requirements must be respected in a global design in which are activated and co-operate security functions distributed on hardware, on code, on smart card, on applications and, mainly on the logic model of the management of communication.

6.1.1.4 Technology

One of the requirements of such a telematic system is to bring, electronically, the health care services closer as possible to citizen and health professionals. To do this, at the moment exists only the possibility to make use of public communication networks.

Another important factor is the short time at disposal to provide results that can be seen as a constraint to create blocks of working systems to be later on integrated in a complete manner. For this reason, the design must be driven by the usage of standard architecture, of tools well known and widely supported, and possibly, of a unique platform on which implement the initial phase of the project.

Costs must also be considered: the choice of implementation tools and platforms must consider carefully this factor, both in terms of financial impact and, technically, with the proven ability of co-operate with a set of remote systems that, by definition, have normally limited dimensions (in terms of technical characteristics and direct support at local level).

The necessity of support activities both on-line and off-line (in case of unavailability of communication connections) guides the design towards an asynchronous client-server communication.

Applications and services that must be present on the "Virtual Network" are, in principle, all asynchronous and should work with a mechanism of request/answer, in which communication time is not granted and to which is just possible to assign a value for maximum expected time of elaboration. 'Cause it is not possible to consider granted, at least in short term period, the data transmission in a public network nor its level of simultaneous charge, will be necessary to reduce as possible the volume of data exchanged, in such a way to obtain acceptable performances.

The various security levels to introduce in the system vary from a dedicated hardware for encryption and electronic signature to the tracking and authorisations for the transactions in the network.

Two other items about security to be considered are related to the certainty of the fact that the transmitted information has been taken in charge by the proper addressee and the verification of the physical presence of the person vs. the use of the smart card. These additional items can be left apart in the first phase of implementation.

6.1.2 Expected Architecture

In the actual situation is normally the citizen/patient that must follow the process of health care service provision, split in several moments equal to the number of the structures/professionals contacted. The General Practitioner, that is in most of the cases the originator of the process, as well as specialists and emergency department professionals, after their direct contact with the patient have no acknowledgement about where and when the patient will obtain the requested services, nor about the results of such services, nor the possibility to discuss of the case during its life-cycle: only when and if the patient brings to them the results of services received, they can know what exactly happened.

The implementation of telematic services in a "Virtual Network" allows the re-positioning of healthcare professionals in their real care role as well as to keep informed of the performance of healthcare services and to provide their managerial control to central regional authority (just on the administrative or quantitative data of central interest).

So, the aim of the project should be the definition and the implementation of a logical and technological architecture for a communication and co-operation system that equally involves health professionals, health care structure at local and central level and patients/citizen in the optimised process of request/offer/provision/ acquisition of health care services available in the Lombardia Region.

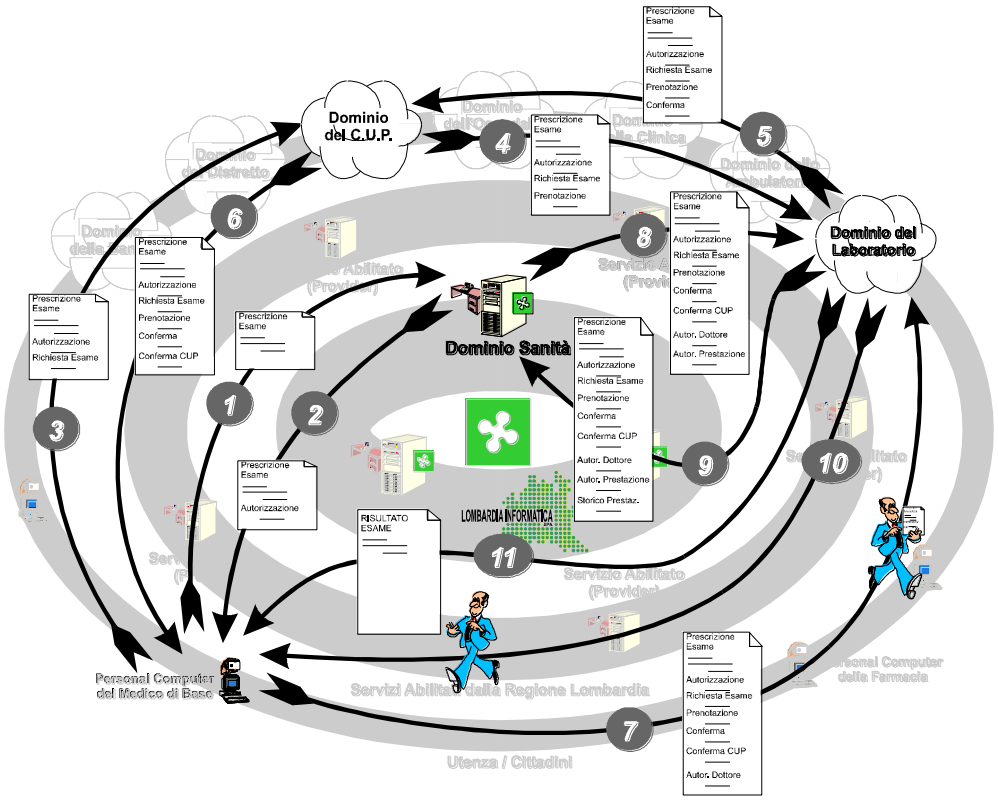
The logical general architecture of the system should be based on the following hierarchical levels:

- The healthcare central domain (regional services – 1st level), in which are present the services for identification, authorisation, validation and logging, in addition to the management of central databases and central systems;
- The local health care structures, including hospitals and services providers (2nd level), to which are connected
- Health care professionals, including general practitioners, specialists, pharmacists, diagnostic centres (3rd level)

All these actor are connected to the "Virtual Network" that provides to the possibility to use services and to communicate.

6.1.2.1 General Services Expected

The way to understand the improvement in terms of efficiency and effectiveness provided by the new telematic system is related to the scenario of the fundamental services that will be supported in such environment. These services, some of them tested in InterCare, are presented in a general manner in the following sub-chapters.



Scenario for the workflow for a lab test request

6.1.2.1.1 Prescription

More than 80% of health care costs are generated by prescriptions. The prescription is also the reference item for the integrated management of all the health care related events, as well as for the connected administrative matters. If the flow of the prescription is tracked and it is recorded in a central regional database as the starting point of the life/cycle of a request/provision process, it will be possible, with the recording of additional events related to this, to obtain an integrated view of the complete process up to its end, both for health and administrative information.

The relation between a patient/citizen and the prescription/services provided will constitute a complete folder for his/her contacts with the health care system.

6.1.2.1.2 Booking-appointments

The current, not always easy, way to obtain appointments/booking for health services is a clear overhead for the patient in the health care circuit. This situation can be dramatically changed with the support of the Virtual Network, connecting in an equal way requesters and providers of health care services: to the network can be interfaced the existing booking systems offered by each health care structure.

The research of the reference provider that satisfy in the best manner the requirements of the patient, could be supported by:

- the general practitioner at the end of a visit, with the production of a diagnosis and a set of prescriptions. In this way we can have a seamless process from the first to the next contact of the patient with health care professionals;

- the health care structure (local, hospital, department) to which the patient can access directly alternatively to the general practitioner;
- the private health structure freely chosen by the patient;
- the patient him/herself, with provided of basic knowledge on "internet" environment and supported with necessary tools (i.e. the health card). This option is not foreseen in the current project, in relation to the complexity of cultural and logistic preparation work.

6.1.2.1.3 Service Provision

The registration and transmission of the data about the provision of the requested service close the process started with the prescription. To this event can be associated the possibility, for the original requester, to obtain the results of the provided service. This allows also an administrative matching with prescription/provision, permitting a real knowledge and a monitoring of costs/incomes for health care structures and regional organisation.

6.1.2.1.4 Knowledge on healthcare history

The availability on a regional base of precise and complete data about service request/provision provides to enabled users the possibility to know and query about health care events.

This access can be open on the public (not protected) data on patient that are also recorded on the health card, or implicitly authorised via the provision of the health card by the patient to his/her physician, that constitutes an authorisation to access patient-related data.

This scenario offers an answer to the present gap about medical history on the patient and communications/shared responsibility between primary and secondary health care, and provides, via a different view of the information, the instrument by which health care structure can monitor, control and manage the request/provision process, applying all the necessary corrective actions.

6.1.2.1.5 Additional Services

The Virtual Network being based on the public network provides the possibility of utilisation of an additional set of services, at marginal costs for the project, with added values to the network connection for each user. The users could in fact access:

- services offered by the providers (remote support, help-desk etc.);
- services offered by other actors that can play a role on components of the network, such as additional providers of services encapsulated in the smart card (i.e. banks, municipalities etc.);
- other telematic services offered by sector of the Lombardia region;
- all the services provided directly by Internet, with possibility of incremental customisation (i.e. mailing and restricted, authorised group of mailers supporting professional communications etc.).

6.1.2.2 The Central Regional DataBase

Every time a service is requested for a patient, accordingly with the controls of access, identification and authorisation supported by the technological platform, in the central DB should be recorded the data related to this event. Once accessed the data related to such request, every additional event connect to this (booking, provision etc.) is linked in such a way to create a history path complemented with statistics and administrative information.

This DB can provide dedicated view to monitor the vents and provides the main references for finding the connected extended information.

In addition, in the central DB will be recorded, in protected space, all the logs of network access by the users, identifying each one interacting in the request/provision process.

6.1.2.3 The Health Card

The health card (microprocessor based) has the double aim to allow patients to access the health services in the net and to certify their provision, also via electronic signature where necessary.

The card permits also to record the general information of the owner (demographics and administrative data) plus the minimal data set considered effective for emergency purposes.

As previously indicated in the strategic objectives of the system, the card must be designed in such a way to permit the hosting of additional services not directly related to the health care environment.

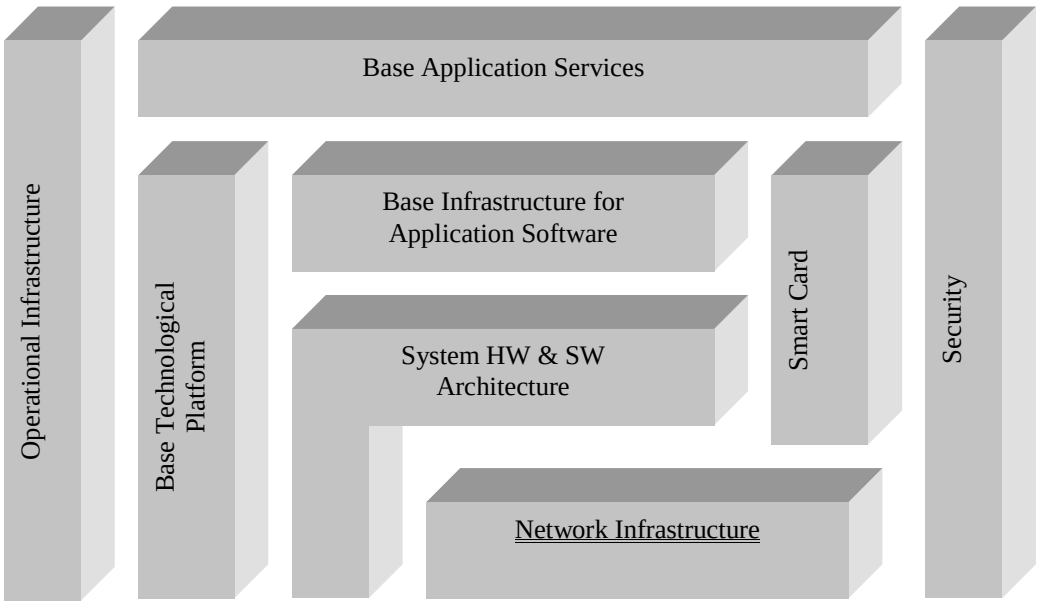
6.1.2.4 The Local Systems

The Virtual Network should allow the citizen to access the services provided via the "domain" represented by the systems already present in each health care professionals setting (individual and/or structure). This means that management autonomy must be preserved at the level of local systems, while the project must specify the technological requirements and functional specifications for allowing the connection of such systems to the Virtual Network and obtaining an integrated environment.

The integration between the Virtual Network and each "domain" system should be performed via the adoption of a software interface (gateway, bridge) that ensures the respect of rules and standards defined for the connection/communication/data exchange specified under the project.

6.1.3 Components of the "Virtual Network"

The system constituting the Virtual Network can be seen as composed of several building blocks, each own with its own characteristics that must be anyway harmonised to obtain the final expected working environment.



Some of these components will be shortly described, mainly the ones characterising the fundamentals of the Virtual Network.

6.1.3.1 Base Technological Platform

We means with this term all the HW & SW mechanisms allowing the communication among the several actors in the net. The base infrastructure is supported by a transport layer (physical, communication and protocol) widely available and permitting transmissions point to point.

The base infrastructure is made effective by three main areas of action:

- the creation of a Virtual Network on a public network, that means a set of telematic services co-operating and transmitting information through a common protocol, that can also be different from the physical one of the net;
- the implementation and management of security at every necessary level, physical and logical;
- the specifications on the way in which the applications can use the Virtual Network.

The several users and/or local systems accessing the Virtual Network are seen as "access nodes" of the net, where is installed a particular software (that will be called Virtual Network Agent – VNA), recognised by the central network domain.

So, the Virtual Network is composed by:

- software distributed on each enabled node. This creates the "virtual" view of the public network. This component is at the lower level of the architecture and permits the exchange of information among similar nodes;
- methods and rules about security, in such a way that the communication is secured among the network nodes through applications that make use of encrypting systems and electronic signature of set of information exchanged;
- telematic services that satisfy application requirements and support that data transmission. These services are the base functions necessary to implement the application programs that use the Virtual Network.

Not all the communications in the Virtual Network must pass through the VNA, but only the ones requiring privacy and security.

Due to general requirements and market trend, the reference platform for the Virtual Network should be CORBA, and Java should be the reference programming language, even if all languages that allow access to VNA services could be used for the development of end-user applications.

6.1.3.2 System HW & SW infrastructure

Different combinations of O.S. should co-exist, in principles. Anyway some preferred platforms aligned with the actual standards and trends must be mentioned. Necessity for a qualification in the Virtual Network of a HW & SW systems is only dependent by its ability to support the implementation of the VNA and to interact with the transport mechanisms applied.

Some distinction must be made between legacy systems and new systems based on the Virtual Network: for the first ones, the VNA can insist directly on the host machine or on a dedicated front-end; in the other case, it is possible already now to establish that the reference O.S. should be Unix and/or NT, supporting their own instance of VNA through their own Java Virtual Machine.

6.1.3.3 Network Infrastructure

There are no restrictions in the choice of the reference network, 'cause it is seen as encapsulated in the Virtual Network via the VNA. It is just important to consider reliability, availability, efficiency.

The network for the project will be a public network. The choice of the IP protocol is related to that fact that, due to diffusion and knowledge, is a "de facto" standard.

6.1.3.4 Cards and Card-readers

The card in the system is used both for identification, authentication and authorisation for the access to the Virtual Network, and as vehicle and repository of patient/citizen information.

The state of the art technology allows the usage of chips with a good capacity (8K), with a high possibility of protection of the stored information. The combination card-reader permits to access the card both in a encrypted manner, secure, or in an open, but controlled, manner i.e. for the data of emergency or other data that, outside the healthcare sector, can be considered public by the solution provider.

In the Virtual Network the card represent a flexible tool storing information and containing the rules for the access to the services. The card must always be seen in conjunction with its O.S. that grants access and protection.

Requirements for the card:

- Multi-provider;
- Security;
- Interoperability;
- Expandability;
- Adherence to standards

The provider of card and its O.S. must grant that:

- The O.S. must be inner protected from external intrusion that tries to modify the structure. If this happens, the O.S. must be not authorised by the reader to access the Virtual Network;
- The zones of the card dedicated to peculiar providers must be accessed only by those providers. Also the O.S., while knowing the physical location of such areas, must not access them. The view of the data in the open zone of the provider area is under the responsibility of the code downloaded from the reader.

We can foreseen three main types of cards:

- Patient/citizen cards
- User's card, that enable single users to provide to patient cards access to the Virtual Network, certifying in a unique manner the identity and the privileges for making use of telematic services;
- Providers cards that, via special R&W devices, can create physical the portions on the processor for their services.

In this complex environment of multi-providers there is the necessity to create and identify a unique administrator of the cards.

The HW security provided by the cards and the readers furnish to the whole Virtual Network the protection levels aligned with the actual law's requirements.

The use of a microprocessor card, having mathematical co-processor, it is necessary to enable all the mechanisms of asymmetric key encryption that are considered as an optimised solution for the system (the co-processor is necessary for the elaboration of integer numbers used in this solution).

6.1.3.5 Security

Security in the project is related to technical and organisational aspects that must grant the transmission of private information inside the Virtual Network.

Two levels at least must be considered:

- the recognition of two nodes in the Virtual Network and the transmission of secure information between them;
- the certified and confidential transmission (via electronic signature) of information in application programs present in the virtual Network.

These levels have validity in each node of the network: the security of the whole network must also be taken in account.

To respect the security requirements, three main items are fundamental (requirements):

- the usage of a specific hardware, the can be seen as the usage of micro-chip card and related reader;
- the integration in the application software of the necessary functions supporting the security;
- authentication, authorisation and logging of the activities performed in the Virtual Network.

Two aspects are actually left out of the project, but they can result very important in the future:

- physical identification of the person that is using the chip card;
- implementation of the accountability (non-repudiation) protocol in the information exchange between two network nodes/application programs.

As foreseen solution for the implementation of this secure environment, we should consider the usage of asymmetric keys technology; in case this solution results unfeasible or too expensive, the usage of alternative techniques (such as TripleDES technology, dynamic generation of passwords and peculiar identification protocols) as to be taken into account.

In parallel, some organisational structures with the aim of controlling and granting the security aspects should be put in place.

In this scenario, the Virtual Network Agent (VNA) and the Virtual Network at central level will act in the following manner:

- the VNA should encrypt and electronically sign the messages;
- the VNA could encrypt and electronically sign structured information exchange;
- the Virtual Network (central-regional domain functions) should log the VNA activities;
- the Virtual Network should identify the users.

6.1.3.6 Base Application Software

With Base Application Software we mean all the software mechanisms allowing data communication among the several application programs (telematic services) the are resident/use the Virtual Network.

This layer is implemented via the so called

- Virtual Network Agent, and
- a set of services in the regional central domain that activate and grant the network communication.

The VNA is a "static software agent" that is present in each node of the network, so distributed in the network, while the regional/central services are organised hierarchically.

In details, the VNA has responsibility for:

- messaging services;
- information structuring services (semantics and organisation of the information);
- the device driver for the usage of security enabling hardware (card);

The central/regional services have responsibility for:

- authorisation services;
- authentication services;
- log services;
- audit services.

In this view, the central/regional services rely on at least two main databases, one containing all the public keys of all the "parties" recognised in the Virtual Network, the other containing a table for the authorisation level for the transmission/services of recognised parties in the Virtual Network.

Additional services for test of the correct functioning of the net and for secure archiving of information should also be present in the central domain.

The Virtual Network is then based on a logic net, constituted by all the VNAs physically present. The VNAs communicate information sending "messages": all the messages are encrypted by the originator VNA and are understandable only by the addressee VNA.

The information exchanged among the various application programs, that are services in the virtual Network, are defined as "documents". The "documents" are the basic unit of information related to the services to be activated. The applications have the duty to encrypt the transmitted information. To do it, the application can make use of the services put at disposal by the VNA.

The interfaces provided by the VNA to application programs are, practically, the necessary APIs for receiving and sending "documents" (structured information) necessary to create transactions in the applications.

6.1.3.7 Base Application Services

The higher software layer present in the system is related to the applications/services directly used by the health care operators/patients. Up to now, these applications have been built to solve and support particular user's problems and aims, without considering the possibility to work in a co-ordinated way with the applications of other operators.

To solve this gap, in the Virtual Network should be implemented an intermediate layer between the end-users applications and the Virtual Network services, that has to support the following:

- Verify that the requester of a Virtual Network service is authorised to do it;
- Receive and transmit from/to end-users applications all the information that must pass in the Virtual Network;
- Check the correct link of processes that must be run when a certain type of information is transmitted;
- Manage the recording of the data that pass through the Virtual network;

- Grant the link between information logically connected but physically located in different nodes of the network;
- Provide to users/patients the possibility of consulting information that can contribute to the improvement of the care process.

Main type of services to be supported via the base application services:

- Management of reservation/healthcare resources in the regional health care net;
- Consultation of health information of the citizen/patient;
- Update of health related information on the card;
- Professional communications/authenticated flows;
- Information to citizen/patient and operator in the net.

Each of these services will be described in detail (processes, information, relation with other services) in the design phase in InterCare.

6.1.4 Main Types of Base Application Services

The base application services can be categorised into 4 main streams:

1. management of health services;
2. authorised/authenticated flows transmission;
3. information to citizen/healthcare operator provision;
4. consultation on citizen health data.

Mapping with application/services presented at the end of the previous chapter is direct, considering the processes performed on the card as part of – sub-processes of the previous main services.

The "**management of health services**" is built on the fact that most of the health services provided starts with a **prescription process** and the (information contained in the) prescription is the tool for

- request
- provision
- control
- financing

the health care system.

This service can be seen as a superset of basic services linked to the whole "production" phase of a healthcare service, on which is possible to verify and control all the actions necessary for the delivery of what necessary.

Being the prescription physically a "document", it is possible to consider the prescription as a "container" of the information that characterises itself and that are in following steps elaborated, extended and completed up to the moment in which they can be stored at the process completion.

The "**authorised/authenticated flows transmission**" must represent a generalised infrastructure allowing a healthcare operator to send a data flow authorised and/or encrypted to another operator in the net or to a remote server. Data flows coming from different operators generally have to converge in a unique destination where they can be controlled, matched and validated.

The "**provision of information to citizen/healthcare operator**" are generally considered based on public information stored on a internet web site, related to the Region at central level and to authorised providers (hosting information) and health care operators sites, and are seen as facilities for consultation on general health care information (yellow pages – services charts) and vehicles for accessing reserved information (private) by authorised users (i.e. health care rules and regulations, communications to healthcare professionals).

The "**consultation on citizen health data**" is an instrument for viewing all the healthcare events related to a citizen in his/her life: the access to these information must be authorised by the citizen to the related general practitioner and the professionals operating in a hospital in which the citizen is cared.

Due to the complexity, the implications with other systems (security, cards) and the importance in generating the information exploited by the other three categories of base services, we will put the attention in the "**management of health services**" for this initial level of analysis, focussing on one of its components, that is the "booking process".

6.1.4.1 Management of Health Services

Actually, the access of citizen to health care services in the public sector is possible only with a prescription provided by an authorised professional, excluding the cases of direct access (i.e. emergency).

Normally the prescription is provided by the General Practitioner; anyway other professionals can be enabled to provide the same.

Once defined the necessity of care for the citizen, an authorised healthcare professional can **prescribe** one or more **healthcare services** (on those available in the HC system) to the citizen; then the citizen relates directly with provision structures able to provide what requested.

So the prescription, in addition to the vehicle for obtaining by the citizen the care requested, can also be seen as the instrument (the initial matrix) for recording the events connected to a care process. In fact, when the healthcare services contained in the prescription are provided, their information can be related to the prescription that generated the activation of such services, until the moment of the final conclusion of the process. In this way is possible also to create a chain among different prescriptions and services provided that are all related to a care process based on a diagnostic protocol or differently combined by a healthcare professional, anyway aiming to define and care a particular case.

Having the possibility to access a database in which these information are stored and can be retrieved, is an important opportunity both for medical evaluation and for statistics/epidemiology analysis.

This is an important scenario for an extension of the Patient Identification and Reference Manager for InterCare, where "contacts" are documented by a prescription.

6.1.4.1.1 The life-cycle of a medical prescription/service

Before entering into the description of the main processes of interest for InterCare in Lombardia Region (booking), it is the case to briefly note the global concepts of a medical action viewed in the prescription light.

As said before, a prescription can be seen as a "container" of information (like a folder) that are completed during a certain period, parallel to the completion of the process of care.

Every "section" of such a folder is managed and updated by the applications (legacy systems) of the healthcare professionals collaborating in the care, that are acting on the single phases of the

prescription and that should be transferred to the "**management of health services**" application and database (we will call it GSSC, using an Italian acronym).

Practically, and exploiting the Virtual Network as transport media, the services contained in the GSSC application furnish the folder to the legacy applications of the healthcare professionals and, once verified completeness and correctness of the information inserted by such external applications, transmit the updated folder to the professionals involved in the following steps.

In the following table, a mapping is made between processes in the prescription/service cycle and the "actors" that can interact on such processes:

Phase Actor	Prescription	Booking/ Reservation	Provision	Payment	Result	Admin. controls	Financing
Citizen		✓					
Gen. Pract.	✓	✓					
Hosp/Struc.	✓	✓	✓	✓	✓	✓	
Out.P.Dept.	✓	✓	✓	✓	✓	✓	
Book. Syst.		✓					
Lab. Syst.			✓	✓	✓	✓	
Pharmacy			✓	✓		✓	
L.HC.U.							✓
Reg. Dept.							✓

As presented in the table, the **prescription** can be provided in addition to the General Practitioner, also by a specialist working in hospital or in outpatient care structure.

In the case the prescription requires secondary care services, admissions, labs examinations, and everything else cannot be furnished directly at the moment of the presentation of the prescription (i.e. drugs and pharmaceutical products), a **booking/reservation** is necessary for the service, to be requested in a department in which such a service can be provided.

The **provision** of services is made by the structures in which a reservation has been made (they can be more than one, 'cause the prescription can contained requests for more than one service to be provided in different structures). The providing structure records the event in terms of certification of provision.

The **payment** for the service(s) provided is made in the providing structure by the citizen in relation to the class of eligibility (contribution) of the patient.

With the process of "**results**" provision, made by the service providing structure, the medical/diagnostic information are recorded and available. In terms of interaction with the GSSC, it is important to note that these data are enriching the content of the "folder" not storing at central level such critical and private data, but just recording the co-ordinates where is possible to retrieve the results.

Final steps of the life-cycle described are related to **administrative procedures**, by which the services provided are controlled in terms of costs, and **financing** procedures, by which the Local Healthcare Unit (now buying services) should process payments to services provision structures on the basis of the performances reported.

Connecting all these events to the initial prescription, it is possible to obtain the complete tracking of the process in its medical and financial components.

The prescription completed with all these data can be then stored at central regional level and being at disposal for every additional inquiry-study-retrieval.

6.2 Main Data Description for Regional Demonstrator in InterCare

6.2.1 Data for Base Application Services

6.2.1.1 The Data at Local Level (HC operators)

The data recorded at central (regional) level need to be put at disposal to the HC operators, due to the process of controls for completeness and correctness of the information transmitted, as described in the previous chapter. How this can be done is depending by a technical choice: even if just a local copy of such data can be sufficient, to coop with problems of integrity and maintenance of these data, a direct access and services to the central (regional) level DBs should be foreseen.

In fact, in the central (regional) domain of the system, all the reference data for knowledge, interpretation, retrieval of information are stored, including the static domain tables containing classifications and coding systems on which controls for correctness are executed.

Two main types of archive should be put at disposal of HC operators in the net (via copy or access):

- codification archives (domain codes and values), containing everything coded by the National Health Service, like hc services, drugs, patient-GP relations, patients;
- operational archives, more dedicated to clinical support information, including diagnosis classifications ICD9, ICD9CM, classification of diagnostic protocols.

So, physically and/or logically, the hc operators should have available locally:

- their application's managed data;
- private/proprietary patient data;
- copy and/or access to the central static codification archives;
- content of the "folder" with the information transmitted/transactions to central (regional) level, indicating the active process of prescription/provision and up to the moment in which message of completion of the process is provided from central level.

In particular, for the central and local data stored in "folder" following a cycle of prescription (up to provision), three main events can happen:

- the cycle is correctly completed and in parallel to the its archiving at central (regional) level, all the hc operators that interacted on that are updated from central level: The "folder" can be locally archived or cancelled;
- the cycle is stopped during a controls phase by the central services of GSSC. A communication to the hc operator that transmitted wrong or incomplete data is given (this can be also done on-line in a transaction – not transmission – process). The hc operator must cancel the wrong prescription and insert a new correct one;
- the cycle is suspended 'cause the citizen does not acquire the service requested. Control on validity date of prescription is provided at central level and local communication should be provided.

In the simpler approach of implementation of these services via communication mechanisms, the life-cycle of the prescription progresses with a "passage" of the "folder" from one to another hc operator. The local archives for such folders must be split into:

- received folders;

- sent folders;

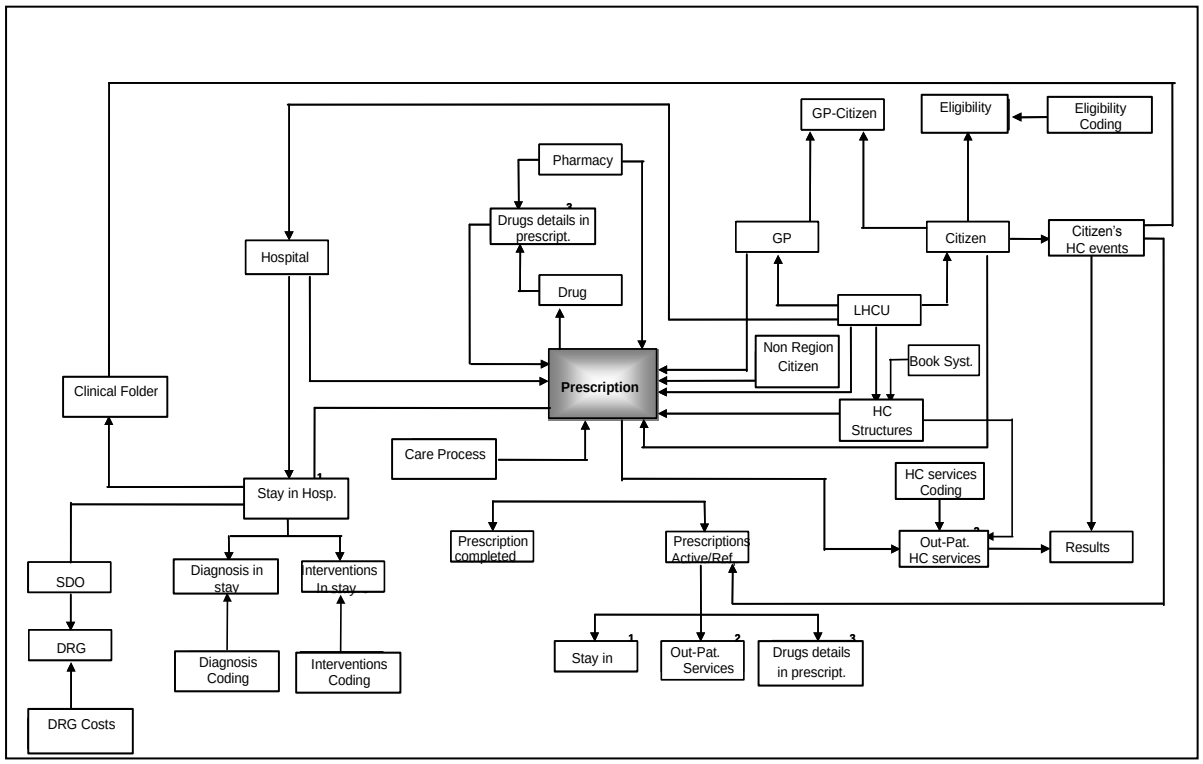
In the "received folders" archives, the hc operator will find all the folders that need a local processing before being sent to other operators and passing to the "sent folders" archive.

The previously describe mechanism implies off-line processes and elaboration: this can be valid for part of the processes composing the life-cycle of the prescription but not for the ones in which immediate answer (i.e. booking/reservation) is needed, to be communicated immediately to the citizen. In this case the central (regional) services should offer facility for routing this massage that, interpreted and stored as communication at central level, activate at the local destination a transactional interface to an existing system able to answer the request, and giving it back passing through the central services.

6.2.1.2 The Data at Regional Level (HC Regional Department)

Focussing on the prescription process as reference for all the information generated in a care process, the regional services are differing from the ones described in Star☆, giving absolute relevance to the coded contact/request (that is the prescription) and using/supposing the identification data of the patient as additional necessary data for completing the knowledge of the process. This is also due to the fact that a card system and a card as support for direct acquisition of these information is present as component of the new environment.

So, the central model of data is extended and complex as a model for a Regional healthcare Information System needs to be and can be schematised as the following (just for the part acting as process control of the prescription life-cycle and the related communications/transactions):



The prescription's information in the regional database

To be compliant with the major events in the life-cycle of a prescription, the regional DB for prescriptions will need to have three archives (versions) of the prescription "folder" for the main statuses of:

- Historical folder (completed processes);
- Active folder (on-going processes);
- Refused folder (processes stopped).

Once the life-cycle of the prescription has been completed, the folder will be passed to the historical archive. The information contained in the folder should be then passed to the relational model of information defined in the regional health care database. In fact, the folder is just an instrument to monitor and control the flow and the life-cycle of the prescription, not available for consultation. A dedicated service at regional level must extract the data from the folder and store them correctly in the relational model of the DB.

Only in this way the data will be available for enquiry (for services as "**provision of information to citizen/healthcare operator**" and "**consultation on citizen health data**") and for a possible regional datawarehousing.

6.2.1.3 Data logically distributed but physically present in the originator site

As said the prescription "folder" is a temporary data repository for all the hc operators directly involved in a certain care process: the data will be then available once transferred in the regional healthcare database.

The prescription can have as results:

- Diagnosis;
- Clinical result (in terms of written medical consultation);
- Examination results (lab examination numeric and graphical data, radiology images etc.);
- Clinical folder for an admission in hospital.

All these information are recorded locally in the originator's system domain, but they must be available on request for citizen and hc operators (ref. last sentence of previous chapter).

The link among all the hc events (contacts that are demonstrated by the existence of a prescription and a service provided) is maintained at central (regional) level in chronological order (date and time) per each citizen, with the references of the local sites (systems) in which are recorded the results of a service prescribed.

6.2.1.4 Data in Health Card

The health card is the main mean that grants the connection of the citizen to the Virtual Network, allowing identification and authorisation in accessing the services provided.

The health card should also be used in addition as a vehicle to record and transport the last "*n*" prescriptions requested by an authorised medical professional.

If the card chosen can store a relevant number of prescriptions, it is possible to foresee a dedicated service in the GSSC able, at the moment of the connection to the net, to check the status of the prescriptions at central (regional) level and to interact with the information stored on the card, cancelling the prescriptions completed in their life-cycle and/or expired in terms of validity time.

In the case the space for recording prescriptions is limited, the software used for managing the card should provide a service able, when the space available for registration is exhausted, to record a new prescription deleting in parallel the older prescription recorded.

6.2.2 Information characterising a prescription

Once defined the flows and the processes of a prescription and the operators enabled to interact with it, it is possible to analyse the macro-information (set of information), typical of each phase. This info is reported in a graphical manner, as "sheets" composing the folder.

HEADER: characterises the "document"

Prescription n° (unique id)
Citizen data (demo and admin. data from the health card)
Region of the citizen (if not from Lombardia region)
Foreign citizen (yes/not)

Operative Data

Active phase of prescription (prescription/booking/provision…)
Ending date of validity for active phase
Transmission status to GSSC (transmitted/not transmitted)
Status (active accepted/refused)

PRESCRIPTION: characterises the "phase"

Prescription date
Prescriptor category (GP, Specialist etc.)
HC professiona/structure prescribing
Prescription type (drugs, ambulatory services etc.)
Ending date of validity

Clinical Data

Anamnesis or clinical question
Care process code
Diagnosis code
Reference to diagnostic protocol code

H.C. Services data included in prescription

Service code
Additional information on the service requested
Status (Active accepted/refused)

BOOKING: characterises the "phase"

Booking/reservation date
Ending date of validity

H.C. Services data included in prescription

Structure/substructure booked
Date and time for booked provision

PROVISION: characterises the "phase"

Flag certifying the performance of the service

H.C. Services data included in prescription

Date of provision
Provider category (specialist, hospital etc.)
Providing structure
Provider professional
Status (active/refused)
Citizen signature (certification) of provision

PAYMENT: characterises the "phase"

Flag certifying the payment made

H.C. Services data included in prescription

Status (Active/refused)
Structure accepting payment
Payment manner
Payment date
Global cost of service
Cost of service directly charged to the citizen

RESULT: characterises the "phase"

H.C. Services data included in prescription

Date of result provision/recording
HC Professional signing the result

Clinical Data

Diagnosis/result
Reference to the logical address of the system where information are recorded/available

ADMIN. Procedure: characterises the "phase"

LHCU code
Admin. info. of performed registration
Registration date
Code explaining refusal
Professional/operator refusing
Data characterising the service in relation to financing (i.e. DRG code)

FINANCING: characterises the "phase"

Payment date to provision structure
Value of payment (Lit/Euro)
Code explaining refusal
Professional/operator refusing

The "Clinical data" can be described partially or in an extensive manner in relation to the level of complexity of – interest in the process of care.

In the header of prescription is indicated the type of services requested: in particular case (i.e. request for drugs) the booking phase can be excluded from the beginning.

In any case the booking/reservation phase cannot be considered necessary/mandatory and in the cases of emergency the services can be provided based only on prescription.

It is necessary that the prescription, at least in the initial period of the new system, is recorded both on the health card and on paper. The two registrations are mutually exclusive and the paper one can be used only in the lack of card support. The emergency cases, evaluated in about the 5% of the services provided, can be managed with a contemporary registration of prescription/provision.

Every phase of the life-cycle is recorded logging date and time of the interaction with GSSC.

It is clear that the information produced by the legacy applications of the healthcare professional must be provided with interfaces to the GSSC, controlling completeness and correctness of information, that will re-route them to the net.

6.3 Main Process Description

6.3.1 The processes in the GSSC (Management of Health Services)

The following chapters describe the processes and define the flows for information/operations related to a phase of the prescription life-cycle and in particular the booking/reservation phase.

Each of these processes needs to be supported by specific services that must be put at disposal by the GSSC to the authorised operators.

The flows highlight:

- Who interacts in the process = columns
- Which actions are done: = rectangles
- Decision processes = rhomboids
- Deviations from the main process = hexagons

In case the diagrams cannot be concluded in one page, a circle with a letter is given as reference point for continuation.

6.3.1.1 The Booking/reservation phase

In the past, the booking can only be done by the citizen presenting the prescription to the acceptance office of the structure where the hc service can be delivered. This situation created problems both in terms of time spent by the citizen and limitations in concurrent choices and the impossibility to optimise reservations in relation to availability offered by the possible provision structures.

The service supported by the GSSC should provide the possibility to perform the reservation process for all the services requested and contained in a prescription, via the networked system.

The service indicated as "**provision of information to citizen/healthcare operator"** are supporting by an application able to know hc services and resources (Regional Enterprise Manager in Star☆ definition) that each hc structure can offer. Knowing where each requested hc service can be delivered, it is possible via the reservation process, to book in a remote system.

Two main typology of service can be foreseen, as described here under.

6.3.1.1.1 Direct reservation via and/or interfacing a Booking System

The reservation can be made by a citizen via a Booking System (legacy) connecting more than one provider. An interesting option to be evaluated in feasibility due to the variety of Booking Systems present on the regional territory and their, sometimes, inadequate architecture, in to create local gateways and interfaces to such legacy systems (as per the approach developed in Star☆). Anyway the integration cannot be always guaranteed. Reference about this solution can be found in Star☆ project documentation.

6.3.1.1.2 Indirect reservation

Also in this case, at least three different possibilities must be considered, in relation to the user activating the process, that can be

- **the hc operator prescribing the hc service(s)**:
  the hc operator connects to the Virtual Network and executes the reservation for the hc services requested for the citizen, defining a set of parameters based referred to time constraints or location/provision structure on behalf and on request of the citizen. The information is sent to the destination system that will elaborate them. The data related to the execution of hc services requested will be then returned to the hc operator via the Virtual Network;
- **at an authorised desk**:
  the citizen goes to an authorised desk, connects to the Virtual Network via his/her health card and, with time or location/structure parameters of selection, make a reservation of hc service in the chosen provision structure;
- **via Internet**
  the citizen reach the Virtual Network via Internet after identification and authorisation procedures, and do the same previously described.

6.3.1.2 The booking/reservation Process

The booking/reservation service can be requested by an authorised user that can exploit an application able to execute the reservation.

The telematic service must be integrated with the GSSC and then the information about the reservation must be recorded on the prescription "folder" (as document specifying the booking/reservation phase, see chapter 1.1.2, this document).

As said before, the reservation can be made directly in legacy booking system, where an operator interacts with the system and the transaction is concluded when the reservation is defined and recorded.

The booking/reservation telematic service must put at disposal a different manner to request a hc service, via an operator in the net, sending a message (that can be interpreted as a transaction a at local level of interface with legacy systems) where in addition to the data about identification of the hc service requested are associated information about geographical and time constraints defined by the citizen (geographical constraints can be substituted by a list of desired structures).

Such request, via the central GSSC services and routing, arrives to all the possible target structures and to their booking systems. Then the request is processed (direct interface or application elaboration) and an answer is given via a data transmission on the local system of the requester, indicating the first availability. The hc operator shows the results to the citizen that choices the preferred. Data for confirmation is sent to the selected provider and cancellation of the other possible discarded reservation is given to the other potential providers.

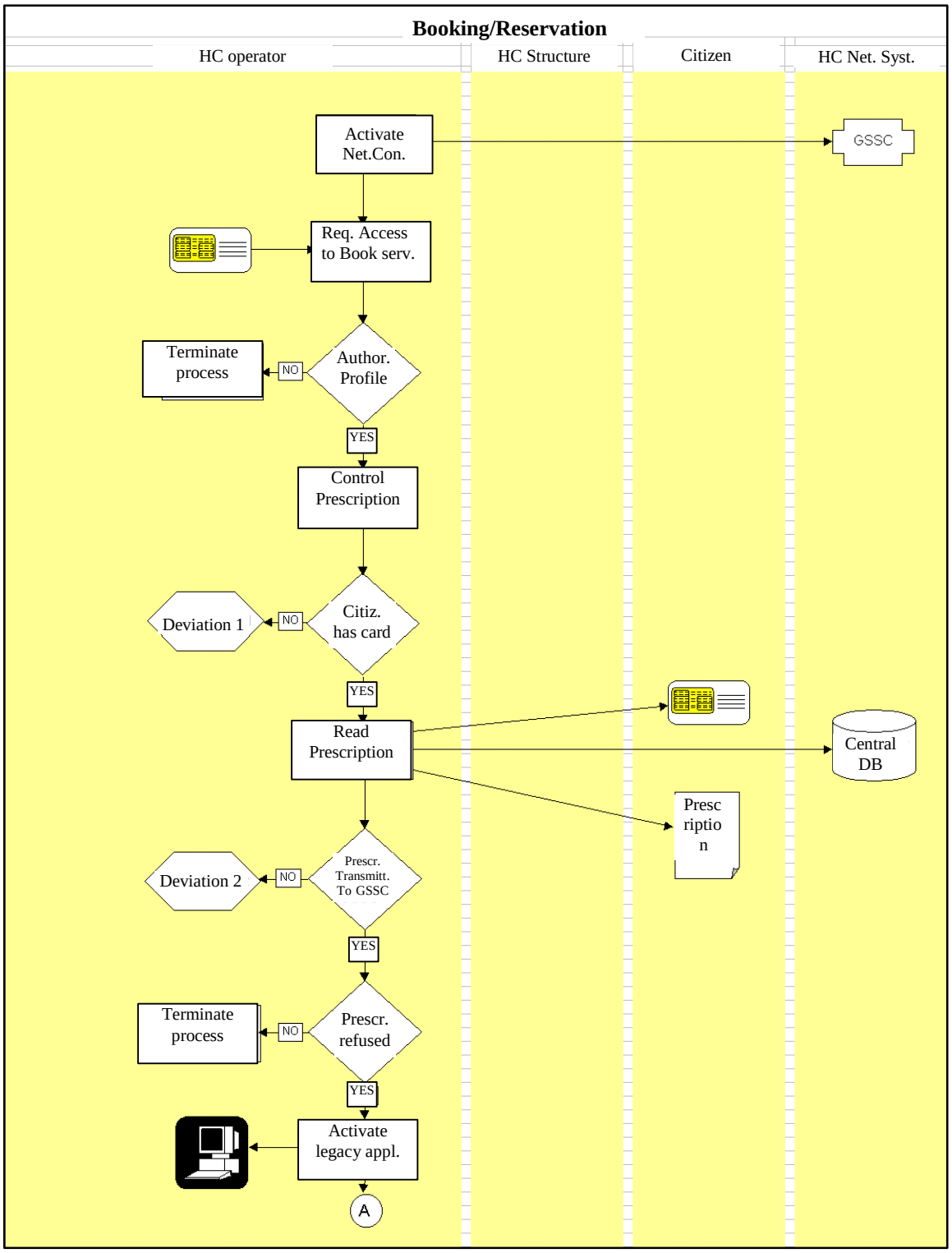
Two conditions are necessary for making the process efficient:

- the structures receiving requests and having availability process them as pre-reservations;
- the hc operator must acquire the concurrent proposals for provision inside a certain time (possibly an acceptable time by which the patient, present, can be informed): after such time, the GSSC via a dedicated service cancel all the pre-reservations made.

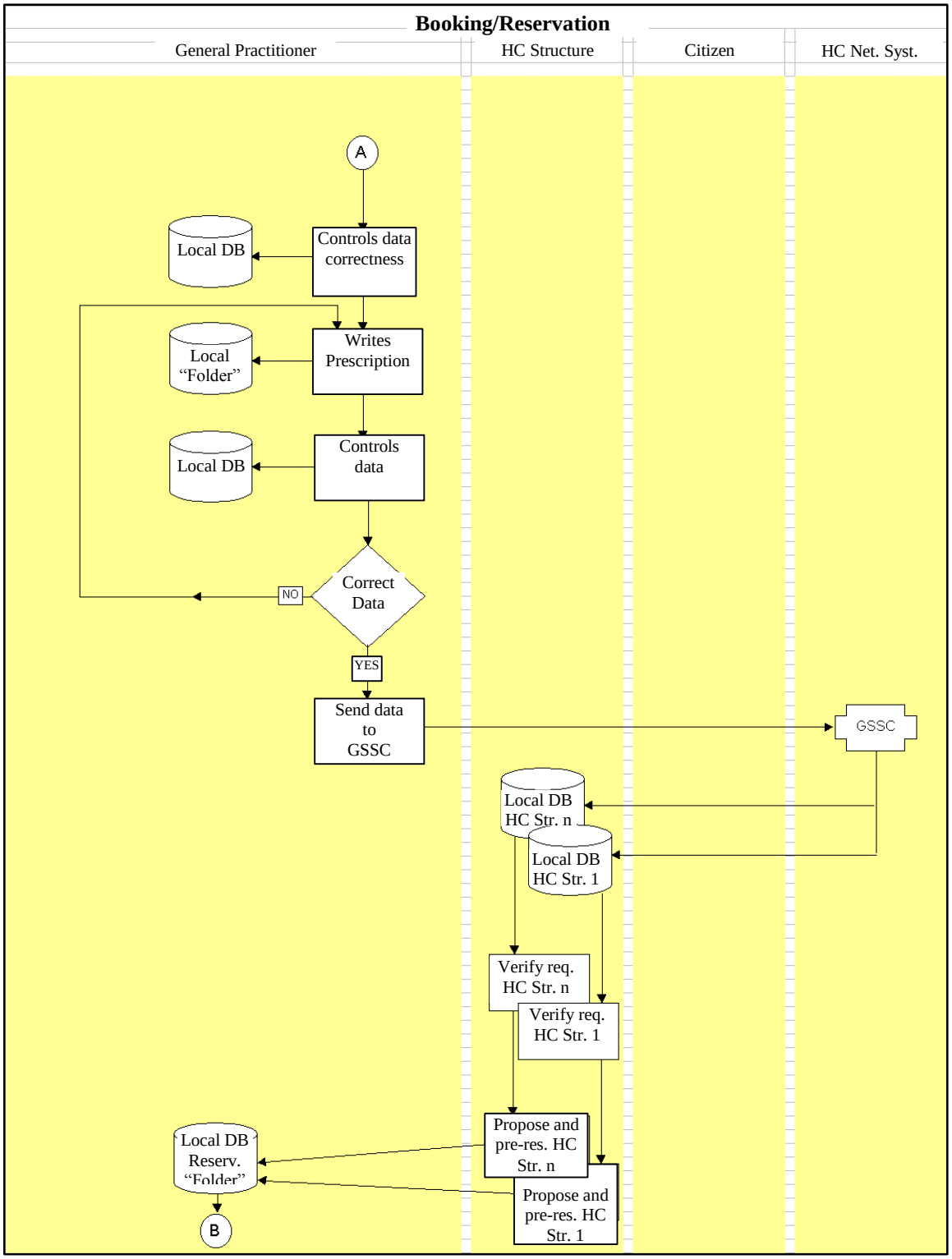
In the following chapters, the flows of information in the process of booking/reservation are graphically described.

6.3.1.2.1 Booking/reservation process by a HC operator (with dedicated view for GP)

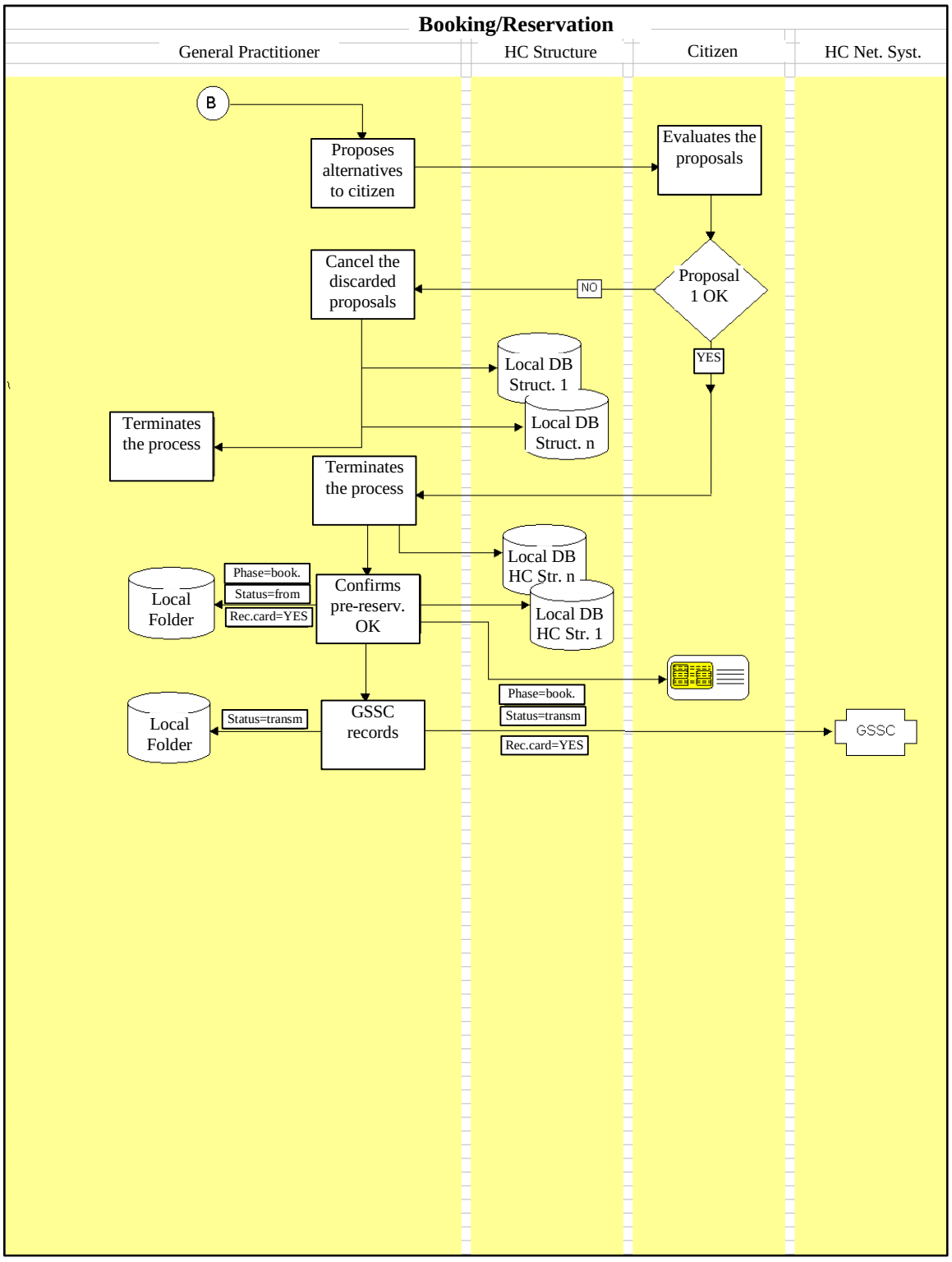
(part 1/3)



(part 2/3)


Booking/Reservation
General Practitioner
HC Structure
Citizen
HC Net. Syst.
A
Local DB
Controls data correctness
Local "Folder"
Writes Prescription
Local DB
Controls data
Correct Data
NO
YES
Send data to GSSC
GSSC
Local DB HC Str. n
Local DB HC Str. 1
Verify req. HC Str. n
Verify req. HC Str. 1
Propose and pre-res. HC Str. n
Propose and pre-res. HC Str. 1
Local DB Reserv. "Folder"
B

(part 3/3)

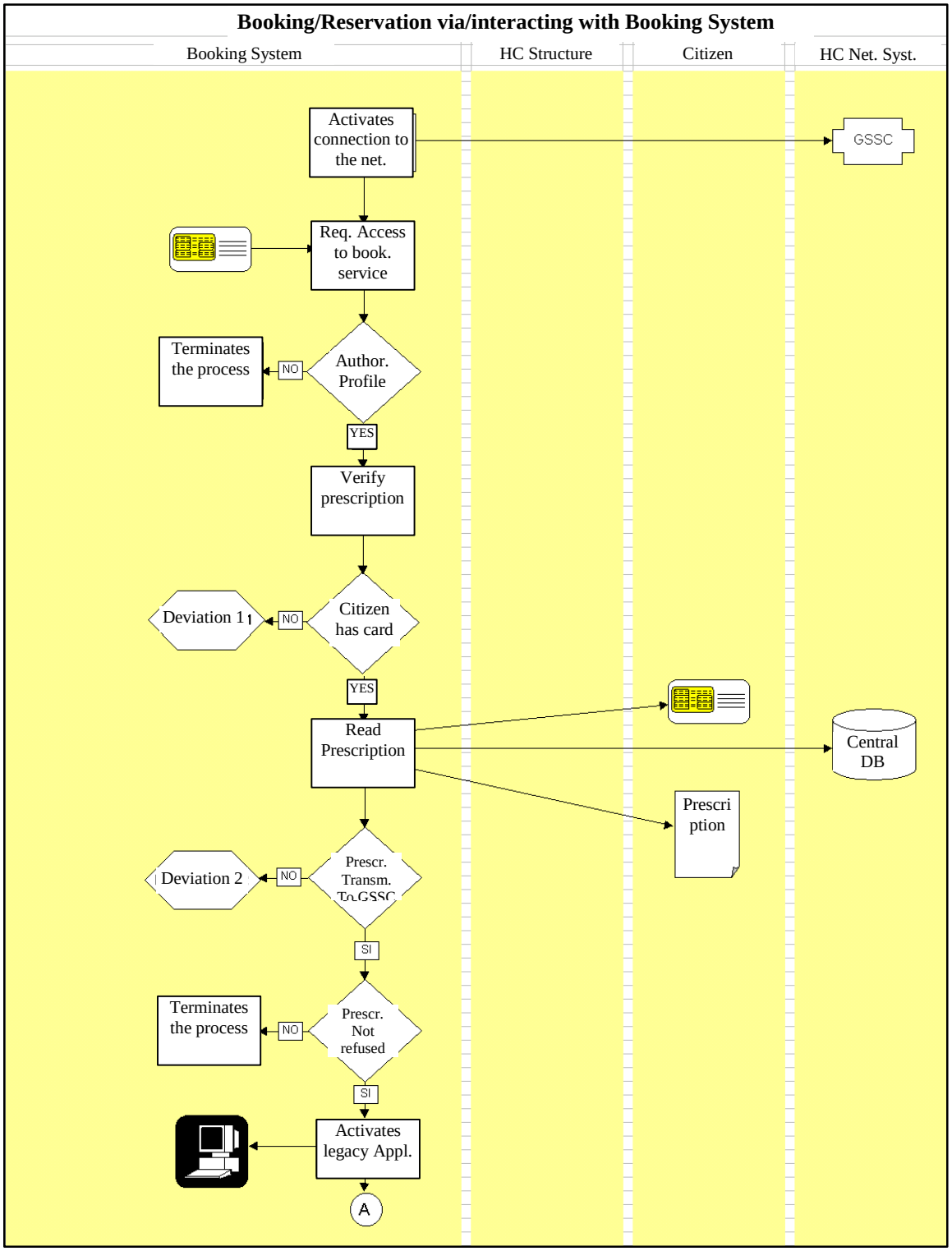

Booking/Reservation
General Practitioner
HC Structure
Citizen
HC Net. Syst.
B
Proposes alternatives to citizen
Evaluates the proposals
Cancel the discarded proposals
NO
Proposal 1 OK
YES
Local DB Struct. 1
Local DB Struct. n
Terminates the process
Terminates the process
Local DB HC Str. n
Local DB HC Str. 1
Phase=book.
Status=from
Rec.card=YES
Local Folder
Confirms pre-reserv. OK
Phase=book.
Status=transm
Rec.card=YES
Local Folder
Status=transm
GSSC records
GSSC

Explanatory notes to diagrams, regarding “deviations”:

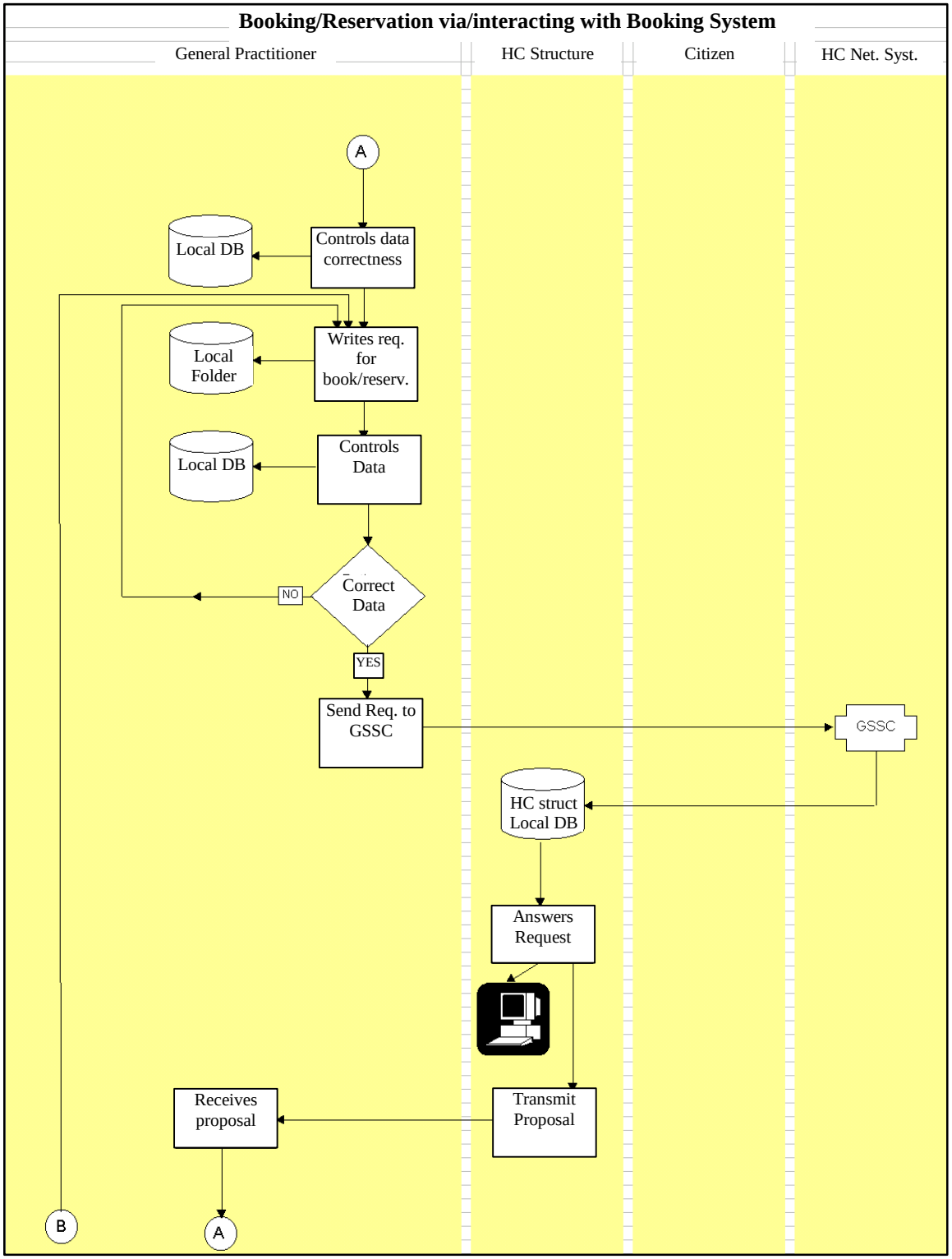
- if citizen has not the card during the prescription phase (***deviation 1***), gives to hc operator the written on paper prescription by which the hc operator search the prescription on the central (regional) database. If the prescription has been transmitted at regional level to GSSC, the process flows normally.
  If the citizen at booking/reservation phase has not the card, the request will be recorded only via the GSSC at central (regional) level. If the citizen during the booking/reservation phase has the health card, via the GSSC should be recorded on card both reservation and prescription (from regional database).
- If the prescription has been not transmitted (***deviation 2***), but has been recorded on the health card, the hc operator can control correctness of prescription data via the card, and the process can normally flow. The GSSC cannot write in the central database a reservation without connected prescription: The registration at central (regional) level of reservation data can be done only with the presence of prescription data. So, services for recording prescription and reservation should be provided: in any other case the (request of) reservation (not transmitted) remains at local level.

6.3.1.2.2 Booking/reservation process via/interacting with a Booking System

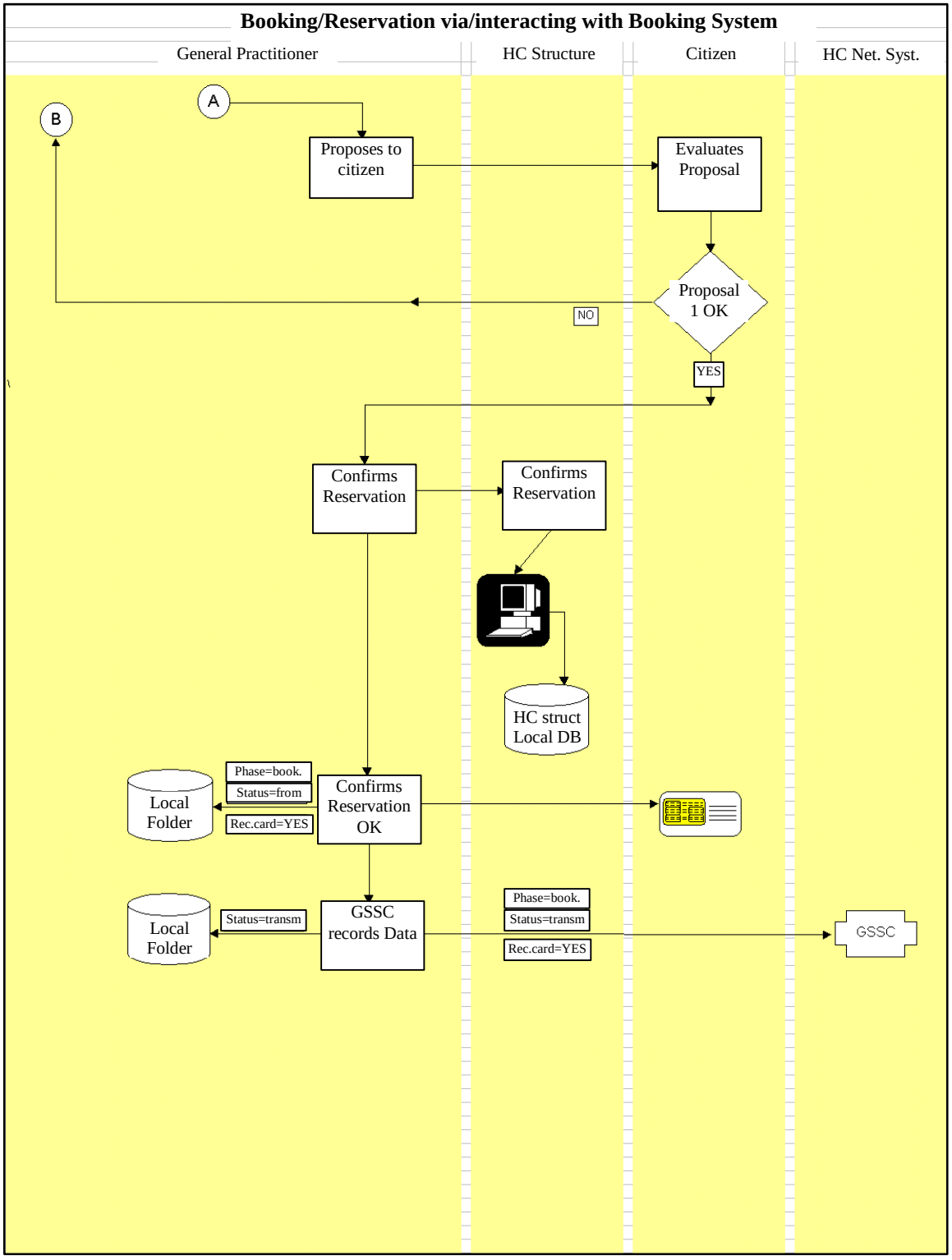
(part 1/3)



(part 2/3)


Booking/Reservation via/interacting with Booking System
General Practitioner
HC Structure
Citizen
HC Net. Syst.
A
Local DB
Controls data correctness
Local Folder
Writes req. for book/reserv.
Local DB
Controls Data
Correct Data
NO
YES
Send Req. to GSSC
GSSC
HC struct Local DB
Answers Request
Receives proposal
Transmit Proposal
B
A

(part 3/3)


Booking/Reservation via/interacting with Booking System
General Practitioner
HC Structure
Citizen
HC Net. Syst.
A
B
Proposes to citizen
Evaluates Proposal
Proposal 1 OK
NO
YES
Confirms Reservation
Confirms Reservation
HC struct Local DB
Local Folder
Phase=book.
Status=from
Rec.card=YES
Confirms Reservation OK
Local Folder
Status=transm
GSSC records Data
Phase=book.
Status=transm
Rec.card=YES
GSSC

6.3.2 Use Cases and Sequence Diagrams

In the following chapters a detailed specification of the Use Cases identified to model and analyse the process previously identified are presented.

The tool adopted by Lombardia Informatica S.p.A. and by the InterCare Consortium is Rose98, so the graphical representation of this set of data is the one offered by Rose.

We identified 7 Use Cases and for each one of them one or more Sequence Diagrams, in particular:

1. Booking via generic agent
   - Operator Booking

2. Booking via GP
   - GP Booking

3. Booking via Legacy Booking System
   - Legacy Booking

4. Booking without network connection
   - Batch Request

5. Prescribe
   - Prescription Phase

6. View Result
   - Result Phase

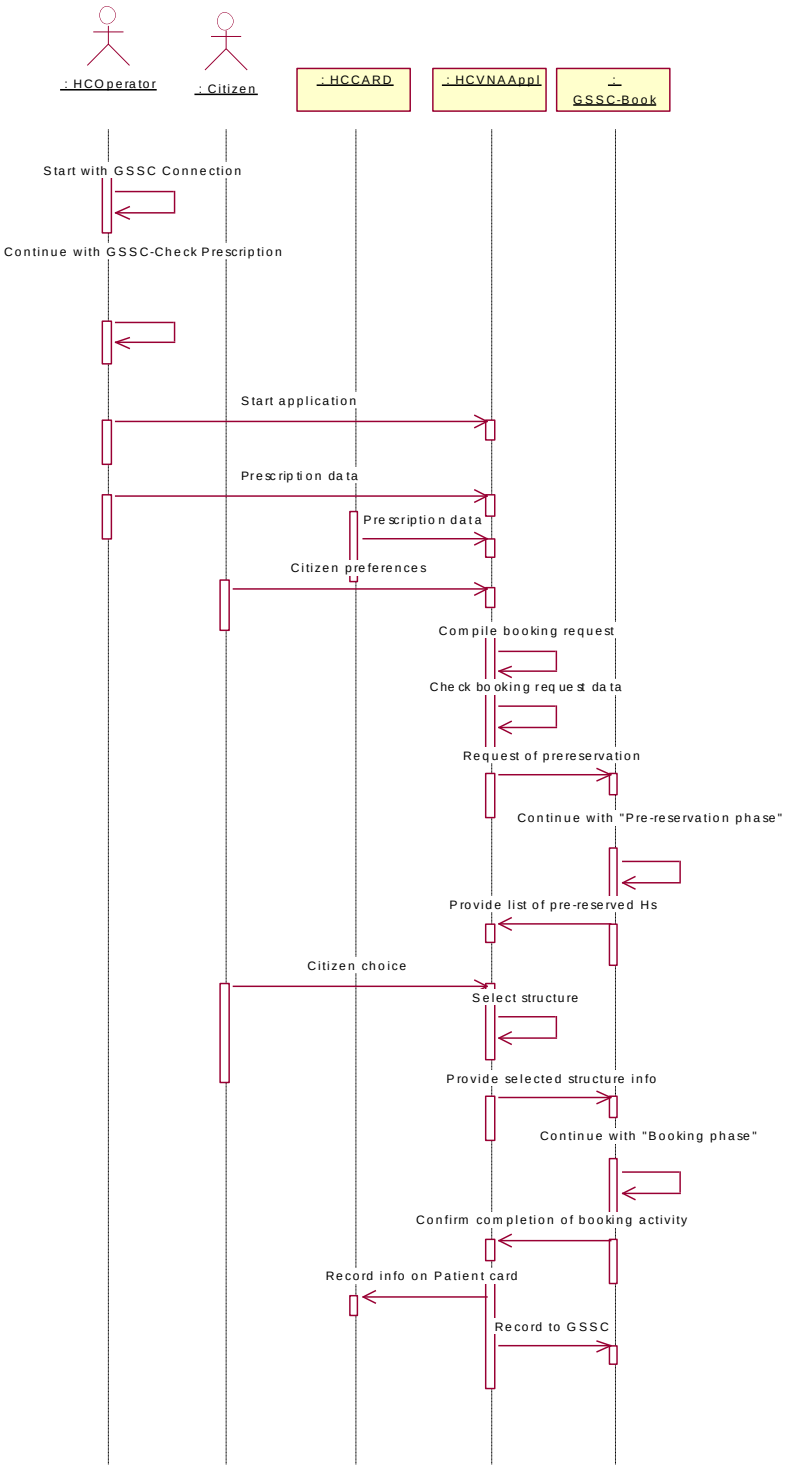
7. GSSC Main services
   - Pre-Reservation Phase
   - Booking Phase
   - GSSC-Connection
   - GSSC-Check Prescription

Within each sequence diagram it is possible to verify a sort of "external call" to other sequence diagrams, since a set of common process is foreseen and modelled just one time and then called it (i.e. "Start with GSSC-Connection" or "Continue with……").

For this reason it is quite difficult to follow a logical flow in the process description by the graphical representation provided in the following models: we decided to introduce in this document the Use Cases and related Sequence Diagram as presented in the previous list.

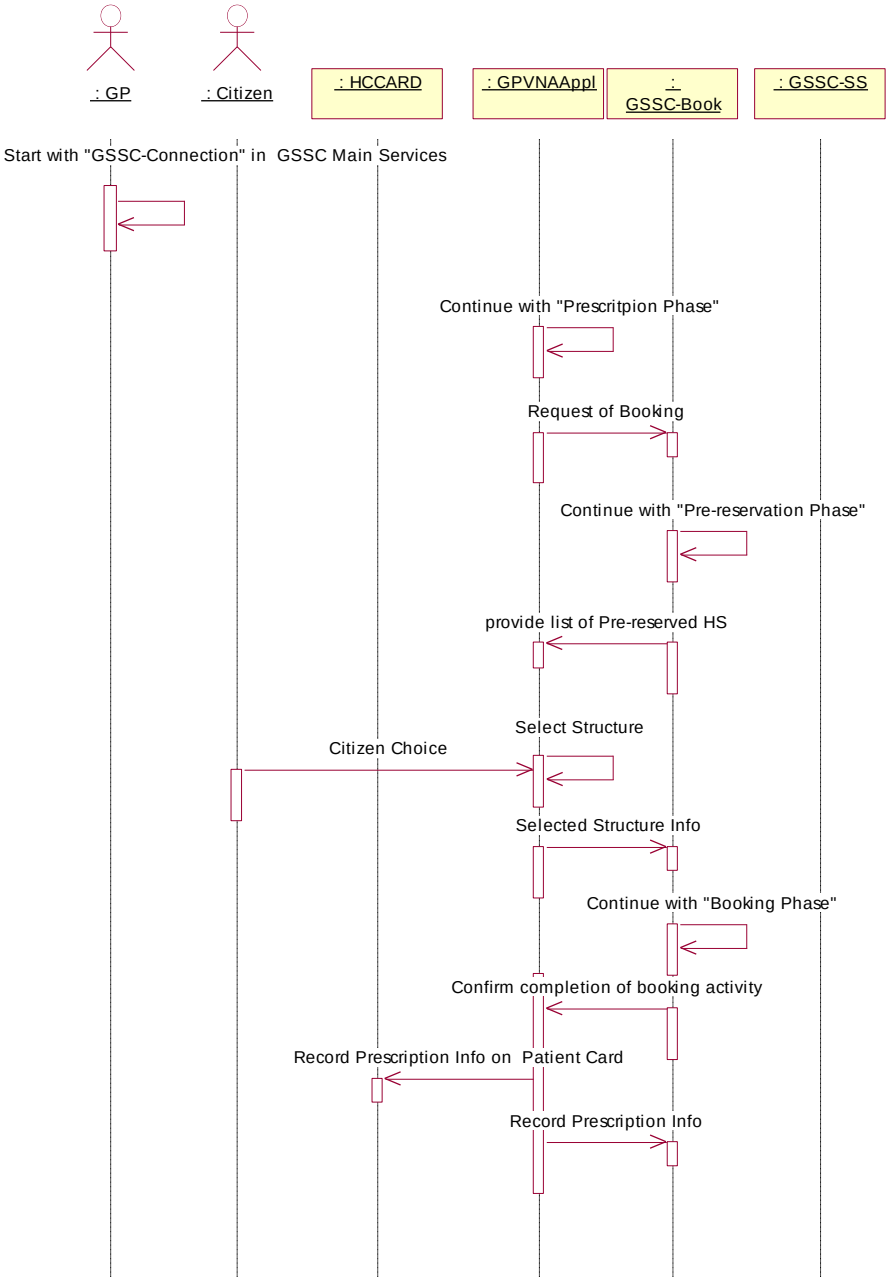
6.3.2.1 Booking via generic agent

6.3.2.1.1 Operator Booking



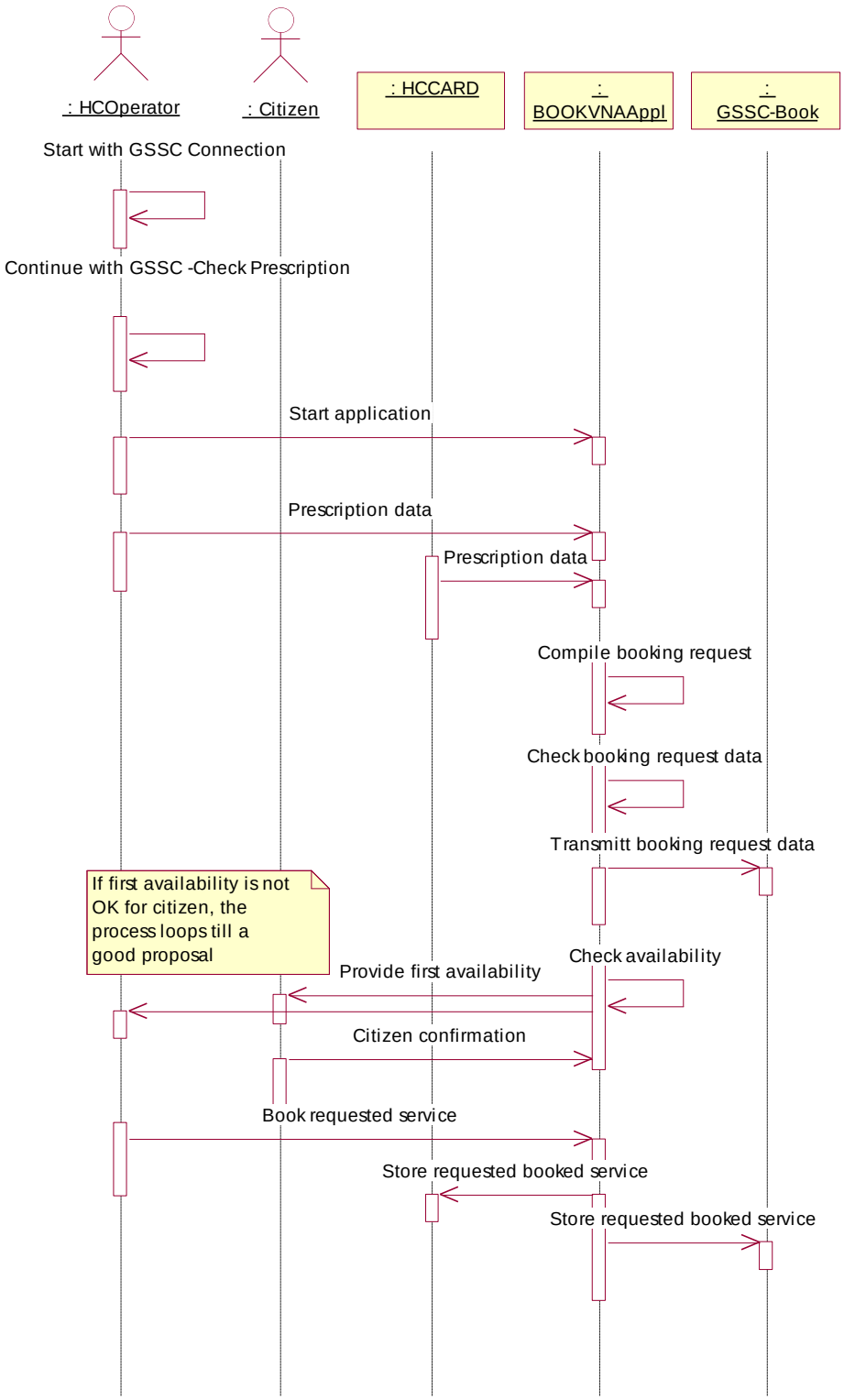
6.3.2.2 Booking via GP

6.3.2.2.1 GP Booking



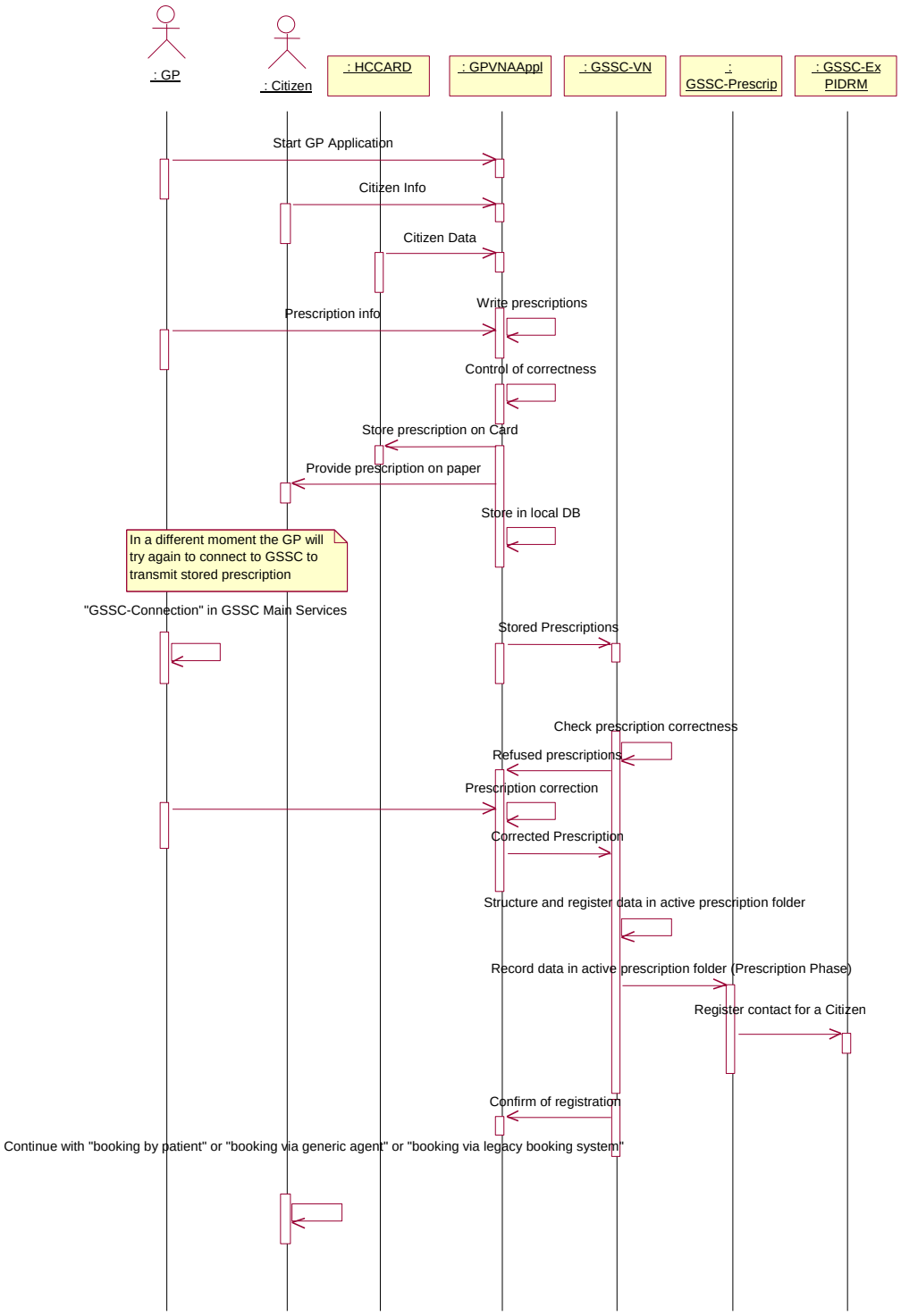
6.3.2.3 Booking via Legacy Booking System

6.3.2.3.1 Legacy Booking



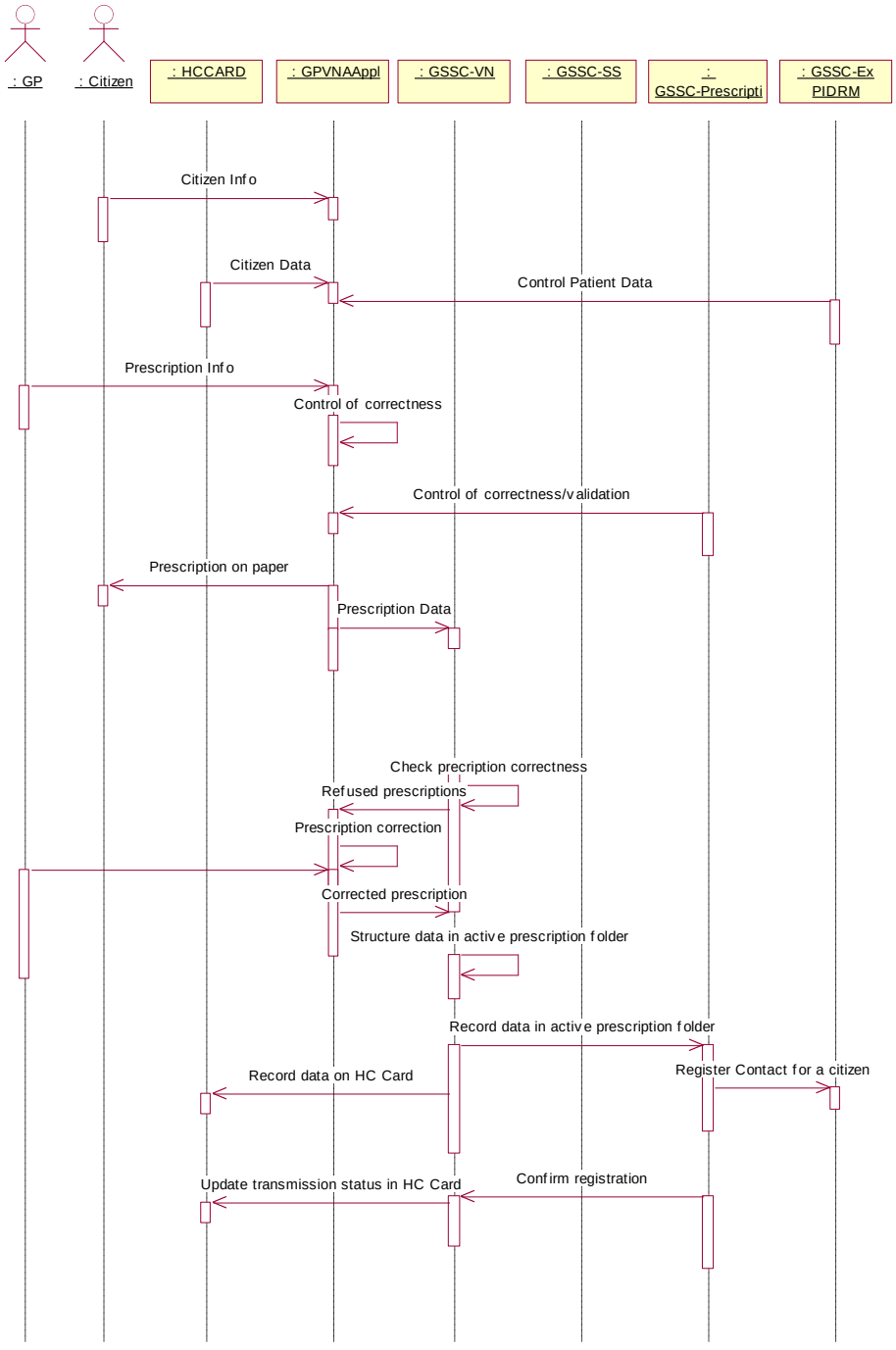
6.3.2.4 Booking without network connection

6.3.2.4.1 Batch Request



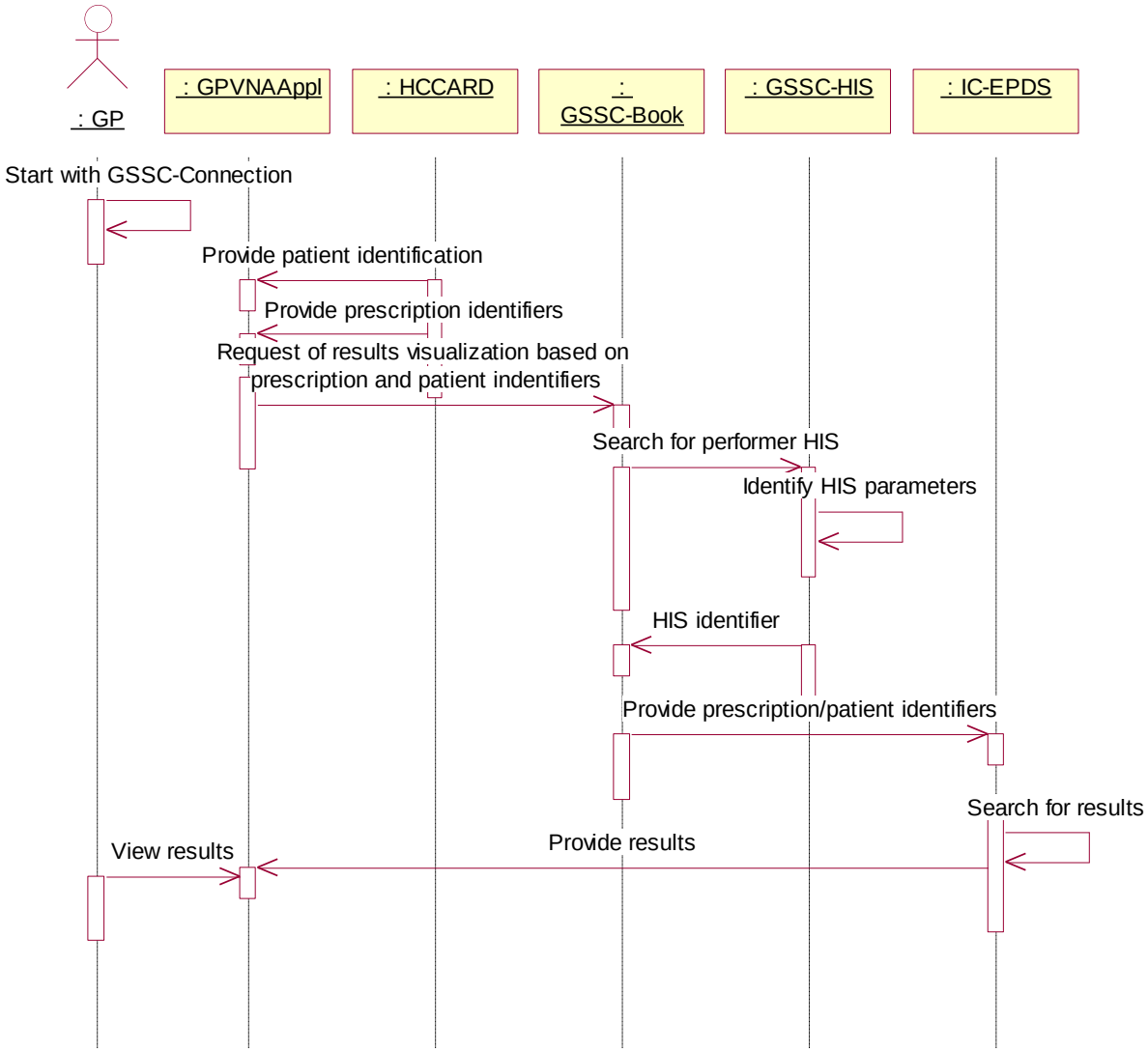
6.3.2.5 Prescribe

6.3.2.5.1 Prescription Phase



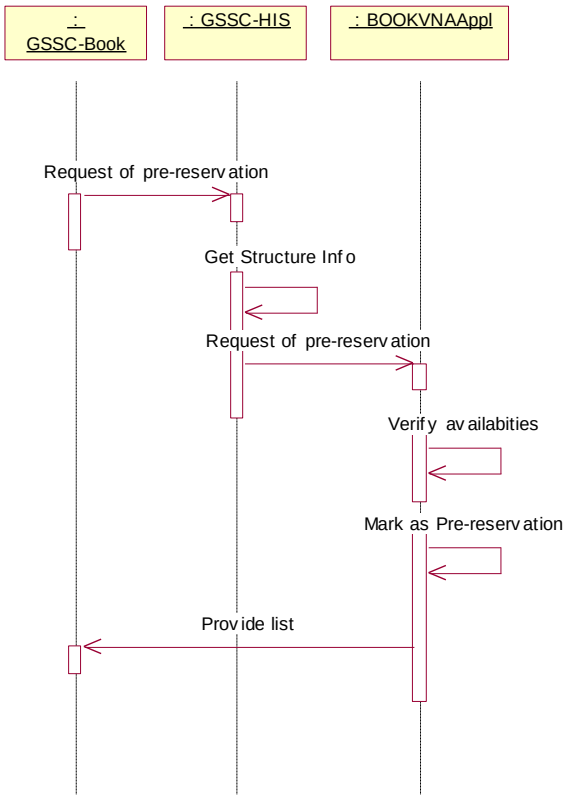
6.3.2.6 View Result

6.3.2.6.1 Result Phase

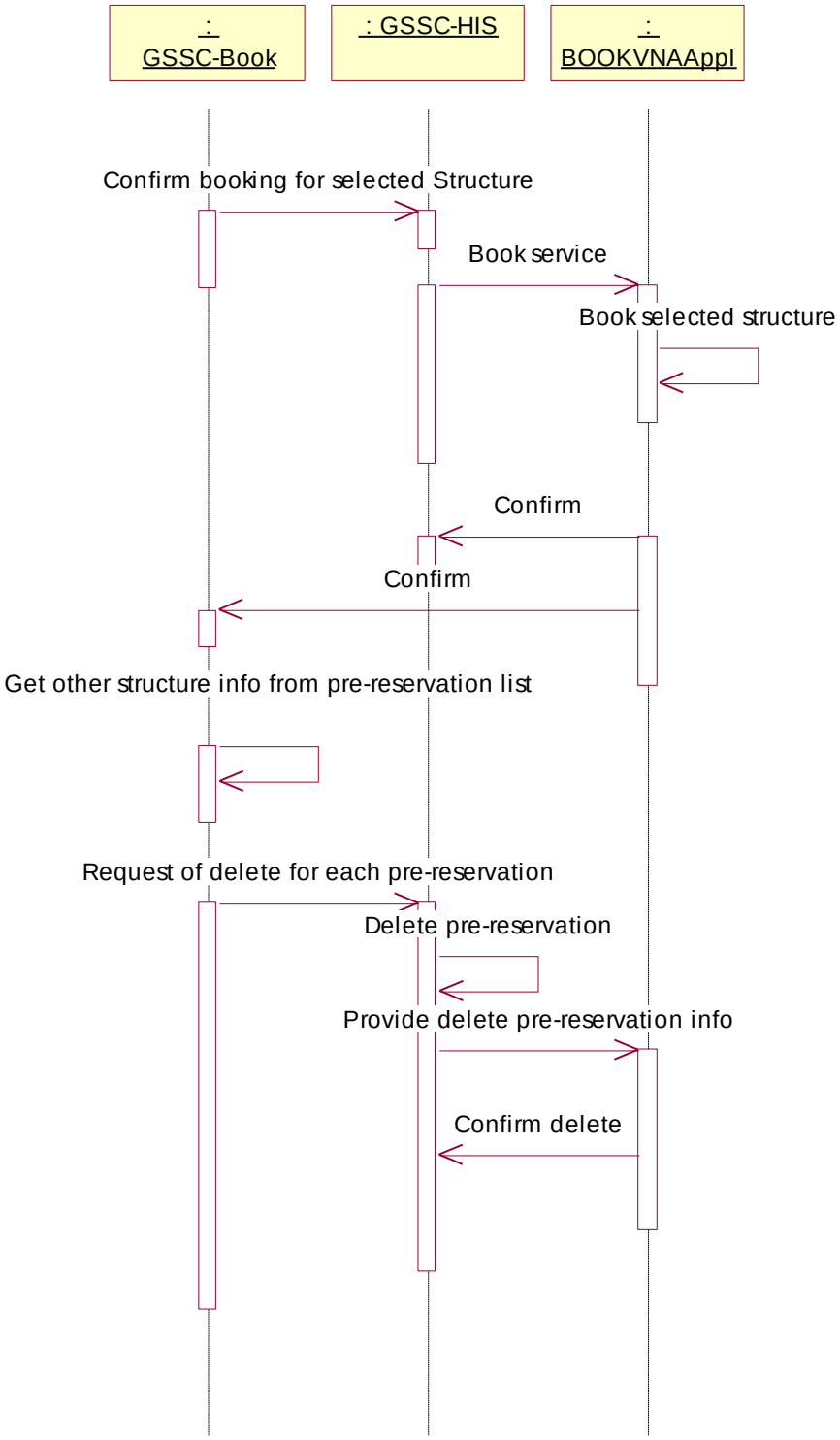


6.3.2.7 GSSC Main services

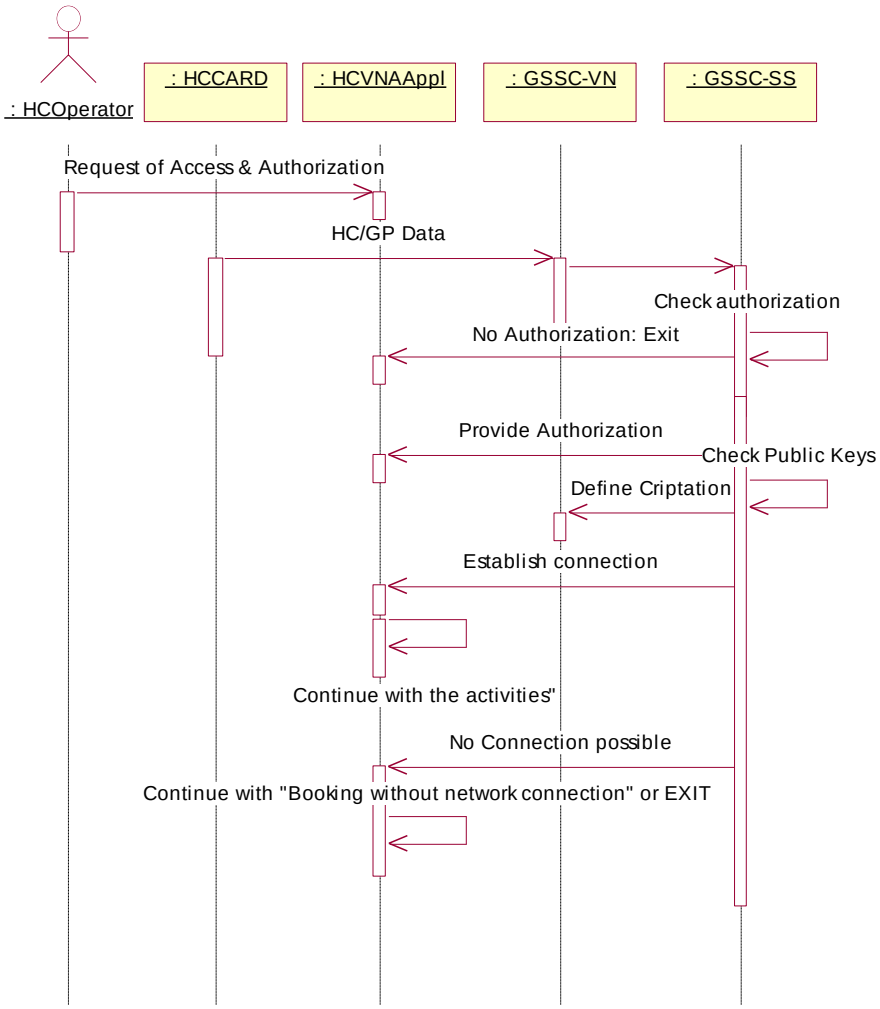
6.3.2.7.1 Pre-Reservation Phase



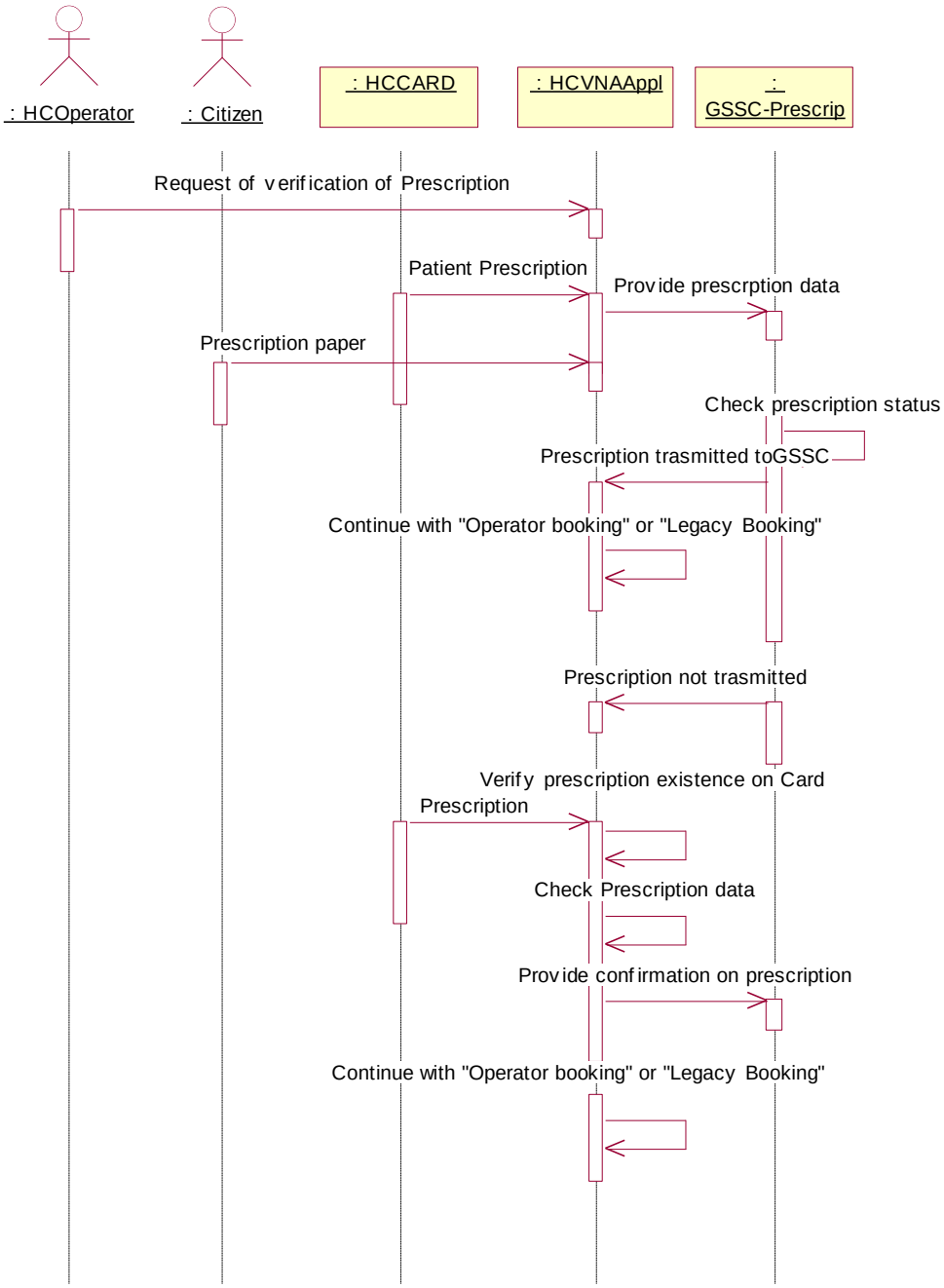
6.3.2.7.2 Booking Phase



6.3.3 GSSC-Connection



6.3.3.1.1 GSSC-Check Prescription



6.3.4 Object Class Model (Package/Class Specification)

In the following chapters a detailed specification of the main packages and related classes/attributes identified as principle set of data required in the definition of the Regional Demonstrator in InterCare are presented.

The tool adopted by Lombardia Informatica S.p.A. and by the InterCare Consortium is Rose98, so the graphical representation of this set of data is the one offered by Rose.

We identified 5 Logical Packages, in particular:

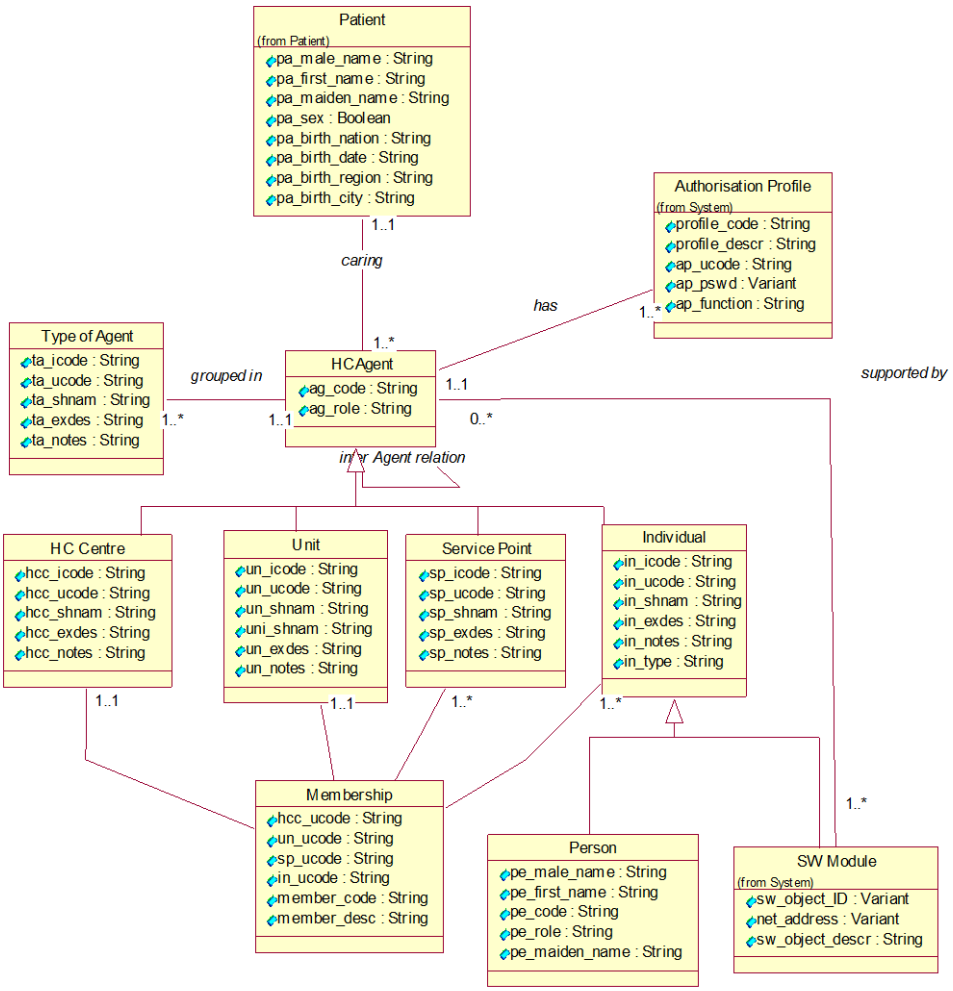
- HC Agent from ICHIS
- Extension of Contact from PIDRM
- Activity
- Patient
- System

For each one of them a different number of classes have been described and characterised by a set of attributes, for a total of 54 classes in 5 Logical Packages.

The following chapters are organised in the following way:

- the Main graphical representation of the Logical Package with all the classes and attributes
- the textual description of each class and related attributes.

6.3.4.1 HC Agent from ICHIS – Class Diagram



6.3.4.2 HC Agent from ICHIS – Class Descriptions

HCAgent
Public Attributes:
ag_code : String
ag_role : String

Type of Agent
Public Attributes:
ta_icode : String
ta_ucode : String
ta_shnam : String
ta_exdes : String
ta_notes : String

HC Centre
Derived from HCAgent
Public Attributes:
hcc_icode : String
hcc_ucode : String
hcc_shnam : String
hcc_exdes : String
hcc_notes : String

Unit
Derived from HCAgent
Public Attributes:
un_icode : String
un_ucode : String
un_shnam : String
uni_shnam : String
un_exdes : String
un_notes : String

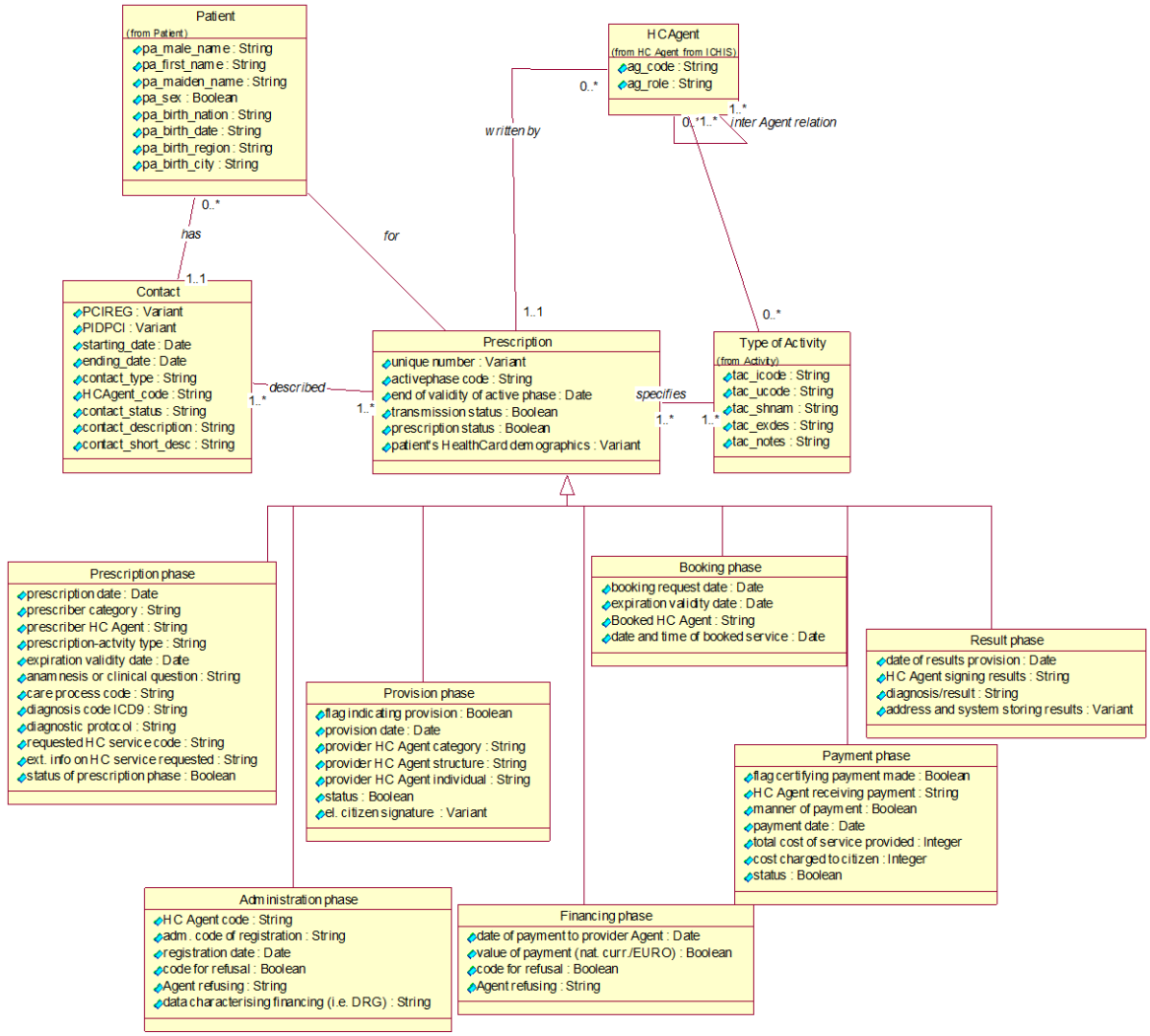
Service Point
Derived from HCAgent
Public Attributes:
sp_icode : String
sp_ucode : String
sp_shnam : String
sp_exdes : String
sp_notes : String

Individual
Derived from HCAgent
Public Attributes:
in_icode : String
in_ucode : String
in_shnam : String
in_exdes : String
in_notes : String
in_type : String

Membership
Public Attributes:
hcc_ucode : String
un_ucode : String
sp_ucode : String
in_ucode : String
member_code : String
member_desc : String

Person
Derived from Individual
Public Attributes:
pe_male_name : String
pe_first_name : String
pe_code : String
pe_role : String
pe_maiden_name : String

6.3.4.3 Extension of Contact from PIDRM – Class Diagram



6.3.4.4 Extension of Contact from PIDRM – Class Descriptions

Contact

Public Attributes:

PCIREG : Variant
PIDPCI : Variant
starting_date : Date
ending_date : Date
contact_type : String
HCAgent_code : String
contact_status : String
contact_description : String
contact_short_desc : String

Prescription
Public Attributes:
unique number : Variant
activephase code : String
end of validity of active phase : Date
transmission status : Boolean
prescription status : Boolean
patient's HealthCard demographics : Variant

Prescription phase
Derived from Prescription
Public Attributes:
prescription date : Date
prescriber category : String
prescriber HC Agent : String
prescription-actvity type : String
expiration validity date : Date
anamnesis or clinical question : String
care process code : String
diagnosis code ICD9 : String
diagnostic protocol : String
requested HC service code : String
ext. info on HC service requested : String
status of prescription phase : Boolean

Booking phase
Derived from Prescription
Public Attributes:
booking request date : Date
expiration validity date : Date
Booked HC Agent : String
date and time of booked service : Date

Provision phase
Derived from Prescription
Public Attributes:
flag indicating provision : Boolean
provision date : Date
provider HC Agent category : String
provider HC Agent structure : String
provider HC Agent individual : String
status : Boolean
el. citizen signature : Variant

Payment phase
Derived from Prescription
Public Attributes:
flag certifying payment made : Boolean
HC Agent receiving payment : String
manner of payment : Boolean
payment date : Date
total cost of service provided : Integer
cost charged to citizen : Integer
status : Boolean

Result phase

Derived from Prescription

Public Attributes:

date of results provision : Date
HC Agent signing results : String
diagnosis/result : String
address and system storing results : Variant

Administration phase

Derived from Prescription

Public Attributes:

HC Agent code : String
adm. code of registration : String
registration date : Date
code for refusal : Boolean
Agent refusing : String
data characterising financing (i.e. DRG) : String

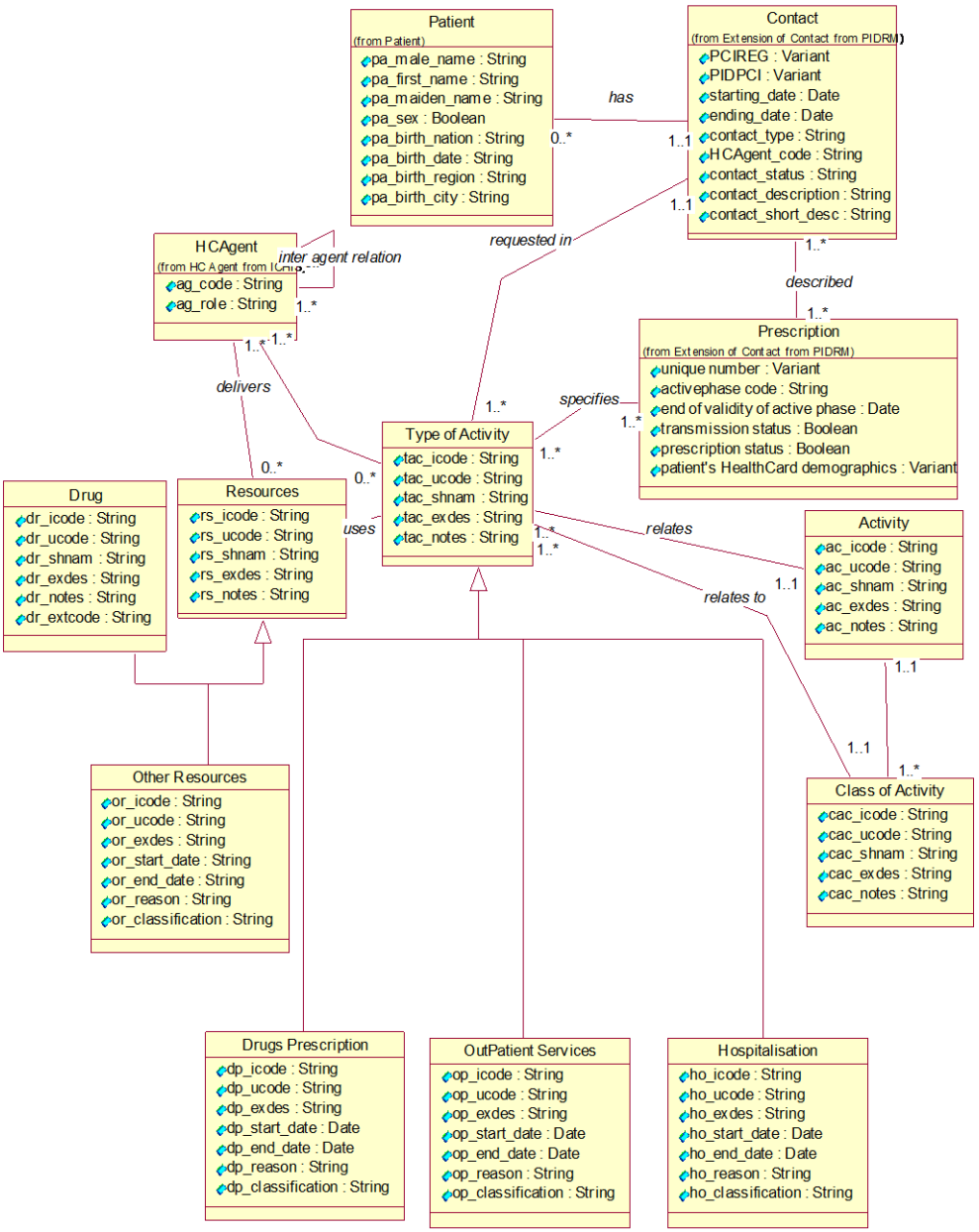
Financing phase

Derived from Prescription

Public Attributes:

date of payment to provider Agent : Date
value of payment (nat. curr./EURO) : Boolean
code for refusal : Boolean
Agent refusing : String

6.3.4.5 Activity - Class Diagram



6.3.4.6 Activity – Class Descriptions

Activity
Public Attributes:
ac_icode : String
ac_ucode : String
ac_shnam : String
ac_exdes : String
ac_notes : String

Type of Activity
Public Attributes:
tac_icode : String
tac_ucode : String
tac_shnam : String
tac_exdes : String
tac_notes : String

Resources
Public Attributes:
rs_ucode : String
rs_icode : String
rs_shnam : String
rs_exdes : String
rs_notes : String

Class of Activity
Public Attributes:
cac_icode : String
cac_ucode : String
cac_shnam : String
cac_exdes : String
cac_notes : String

Drug
Derived from Resources
Public Attributes:
dr_icode : String
dr_ucode : String
dr_shnam : String
dr_exdes : String
dr_notes : String
dr_extcode : String

Hospitalisation
Derived from Prescription, Type of Activity
Public Attributes:
ho_icode : String
ho_ucode : String
ho_exdes : String
ho_start_date : Date
ho_end_date : Date
ho_reason : String
ho_classification : String

OutPatient Services

Derived from Type of Activity

Public Attributes:

op_icode : String
op_ucode : String
op_exdes : String
op_start_date : Date
op_end_date : Date
op_reason : String
op_classification : String

Drugs Prescription

Derived from Type of Activity

Public Attributes:

dp_icode : String
dp_ucode : String
dp_exdes : String
dp_start_date : Date
dp_end_date : Date
dp_reason : String
dp_classification : String

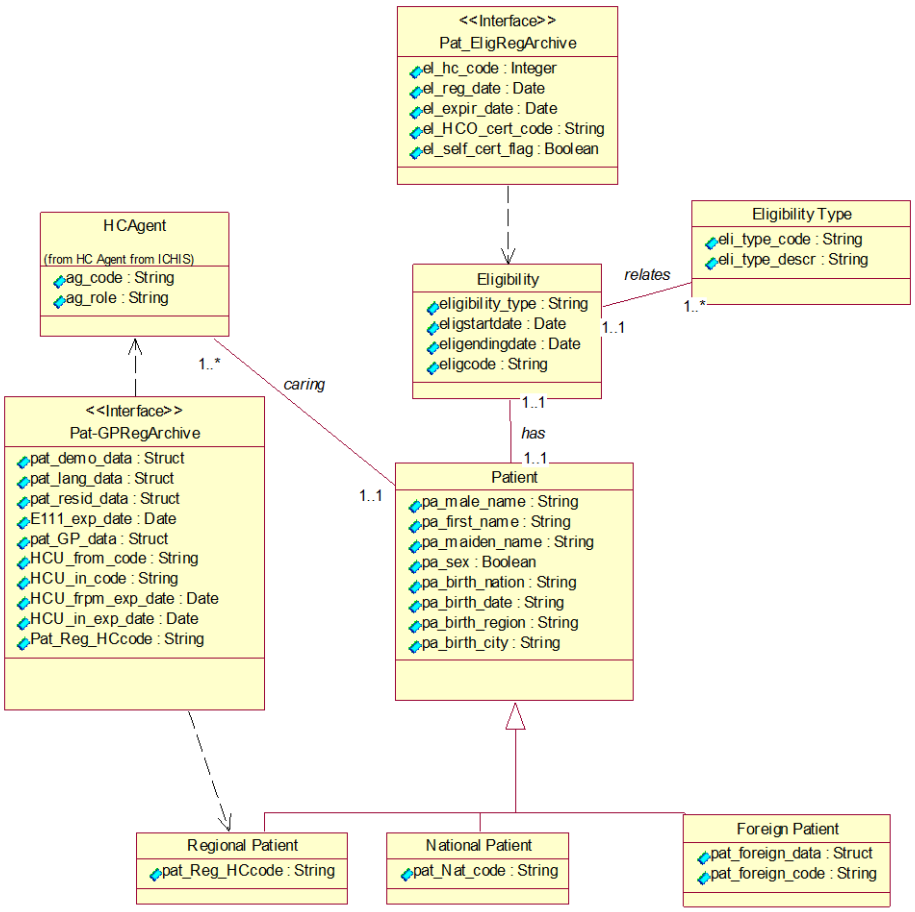
Other Resources

Derived from Resources

Public Attributes:

or_icode : String
or_ucode : String
or_exdes : String
or_start_date : String
or_end_date : String
or_reason : String
or_classification : String

6.3.4.7 Patient – Class Diagram



6.3.4.8 Patient – Class Descriptions

Patient

Public Attributes:

pa_male_name : String
pa_first_name : String
pa_maiden_name : String
pa_sex : Boolean
pa_birth_nation : String
pa_birth_date : String
pa_birth_region : String
pa_birth_city : String

Regional Patient

Derived from Patient

Public Attributes:

pat_Reg_HCcode : String

National Patient
Derived from Patient
Public Attributes:
pat_Nat_code : String

Foreign Patient
Derived from Patient
Public Attributes:
pat_foreign_data : Struct
pat_foreign_code : String

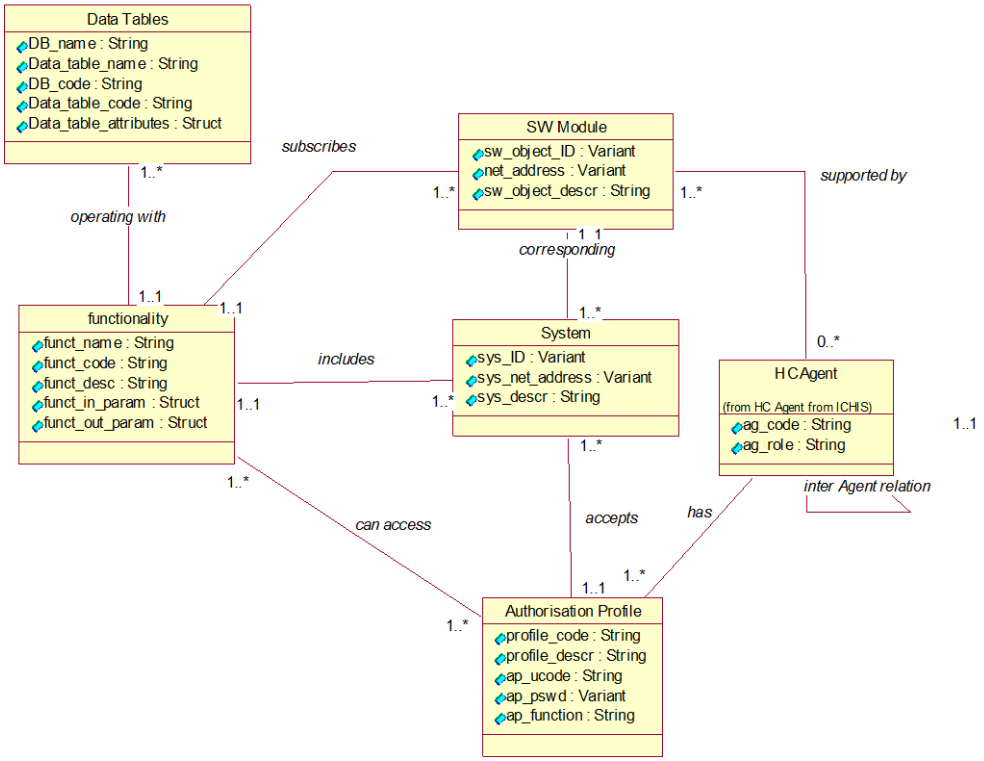
Eligibility
Public Attributes:
eligibility_type : String
eligstartdate : Date
eligendingdate : Date
eligcode : String

Eligibility Type
Public Attributes:
eli_type_code : String
eli_type_descr : String

Pat_EligRegArchive
Public Attributes:
el_hc_code : Integer
el_reg_date : Date
el_expir_date : Date
el_HCO_cert_code : String
el_self_cert_flag : Boolean

Pat-GPRegArchive
Public Attributes:
pat_demo_data : Struct
pat_lang_data : Struct
pat_resid_data : Struct
E111_exp_date : Date
pat_GP_data : Struct
HCU_from_code : String
HCU_in_code : String
HCU_frpm_exp_date : Date
HCU_in_exp_date : Date
Pat_Reg_HCcode : String

6.3.4.9 System – Class Diagram



6.3.4.10 System – Class Descriptions

System
Public Attributes:
sys_ID : Variant
sys_net_address : Variant
sys_descr : String

Authorisation Profile
Public Attributes:
profile_code : String
profile_descr : String
ap_ucode : String
ap_pswd : Variant
ap_function : String

SW Module
Derived from Individual
Public Attributes:
sw_object_ID : Variant
net_address : Variant
sw_object_descr : String

Functionality
Public Attributes:
funct_name : String
funct_code : String
funct_desc : String
funct_in_param : Struct
funct_out_param : Struct

Data Tables
Public Attributes:
DB_name : String
Data_table_name : String
DB_code : String
Data_table_code : String
Data_table_attributes : Struct

7. Swedish Demonstrator Design Specification

7.1 Swedish EPR Demo Application (Stockholm North)

7.1.1 Business Objectives and Purpose of the Application

The defined business objectives of a business are important steering instruments to make the purpose clear and to define the tasks to be supported when developing information systems.

Business objectives can be defined through a business analysis process in which actors from the business are important participants.

Below those business objects are stated which are of particular importance to the Palliative care unit of the Stockholm NorthEastern Healthcare Area. It is expected that the support, which the target application will give to the professional people in the organisation when fully operational, will contribute substantially to their fulfilment.

The business objectives are to:

1. Increase the care quality for the patient, which means that as a palliative unit be able to perform care for patients treated at home with main focus on creating security and to reduce pain and nausea for the patient.

2. Use complementary skilled medical persons organised in teams to offer a high degree of reliability in observations, judgements and medical decisions concerning the patient.

3. Realise new care planning objectives concerning co-ordination of care and treatment between several care units dealing with the patient.

4. Shorten lead times in admittance and medical treatment actions, resulting in reduced waiting times for the patient.

The purposes of the application are:

1. To extend communication exchange between team members and to improve the timing in exchange of information, in order to avoid medical mistakes and inconsistencies between medical decisions taken by different team members.

2. To be able to store information about medical decisions and observations made by different healthcare professionals being part of a care team. Parts of the information must be able to be shared among the team members, while other parts will not be shared.

3. To be able to access specific information concerning healthcare decisions and treatment plans located in other units, based on agreement and business rules.

4. To support the follow-up of care actions and treatments in terms of business goals, for example level of pain.

7.1.1.1 Metrics

One important business goal is to improve the quality of care for the patient by reducing the pain that the patient has. The level of pain is measured in the daily care activities by means of a certain measuring scale in which the patient indicates the level which is closest to what he/she experiences. By studying changes over time in such level-of-pain values, this will give some evidence whether this objective has come closer to fulfilment. However, to sort out any specific factor explaining such a change in trends is a difficult task. To what extent the information support given by the application is contributing may only be judged by the opinions of the care actors.

Another important goal is to reduce lead times in admittance and treatment activities. This can be more objectively measured.

7.1.2 Use Cases and Scenarios

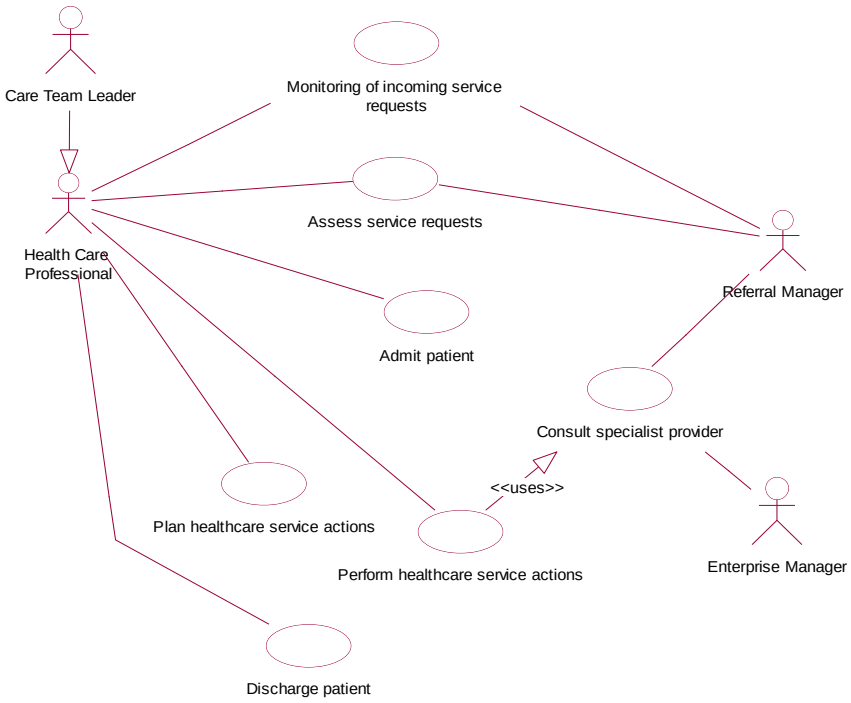
The Use Case View of the Swedish Electronic Patient Dossier Application is described in the following diagram.

The use cases depicted in the diagram are then described more in detail in form of a use case description. The descriptions have been made using a template containing the following sections:

- Preconditions
- Triggering
- Ends with/Results in
- Variations
- Actions.

The use cases are on a somewhat coarse level, and may be ”decomposed” into e.g. Create Care Plan, Revise Care Plan, Display Care Plan, Create Action, Create Health Characteristic, Create Healthcare Contact, etc. See HIS application for more detailed use cases.

7.1.2.1 Use Case Diagram



7.1.2.2 Use Case Descriptions

Use Case: Monitoring of incoming service requests

Preconditions:
Service requests from external healthcare parties have arrived.

Triggering:
Actor triggers monitoring interaction.

Ends with:
Assessments visit scheduled and partial referral answer sent/planned service event reported, or Referral request is rejected and final referral answer sent including motivation for rejection.

Actions:

- If referral arrives via fax or post, enter referral and store it (Secretary)
- Add basic information about ”potential patient” (*patient identification*, *patient name*, …)
- Actor (team leader, doctor, chief nurse) displays referral and checks referral completeness (patient’s permission to share care documentation, patient’s permission to be cared at home, permission from family/relatives for the patient to be cared at home).
- If missing, actor/requester seeks missing information and completes the referral.
- Actor follows up and documents patient’s health status (Health characteristic for ”potential patient”).

- Actor puts a priority to the request (*performer priority* of Referral Request) from available capacity data (current and planned admittances).
- If decision to proceed - create Assessment Visit and report Planned Service Event.
- If decision to reject request – create final referral answer, sign and distribute referral answer to requester.

(See also Use Cases for Swedish HIS Application where use cases for the referral process have been extracted and refined).

Use Case: Assess service requests

Preconditions:
Assessment visit for patient has been scheduled; patient exists as ”potential patient”.

Triggering:
Actor triggers assessment interaction.

Ends with:
Admittance visit scheduled and final referral answer (acceptance, rejection) created and distributed to requester.

Actions:

- Add additional or modified patient demographic, address, telecom address and healthcare location information.
- Add Patient related person information.
- Plan assessment visit: Display patient details, Display patient health characteristic details, etc.
- Document results from assessment visit for relevant care categories: Medical, Nursing,… (Health Characteristics of type: anamnesis, status. …, social situation, …..)
- If decision to accept - create Admittance Visit (*start date and time, additional information*)
- Create final referral answer and attachments
- Sign and distribute referral answer to requester.

(See also Use Cases for Swedish HIS Application where use cases for the referral process have been extracted and refined).

Use Case: Admit patient

Preconditions:
Patient to be admitted exists as ”potential patient”.

Triggering:
Actor triggers interaction.

Ends with:
Patient admitted and relevant patient and admittance information registered.

Variations:

1. New admittance
2. Re-admittance

Actions:

- Add additional or modified patient demographic data.
- Add additional or modified patient Address information.
- Add additional or modified patient Telecom Address information.
- Add Patient healthcare location information.
- Add Patient related person(s), add patient related person information including Address and Telecom Address information, add information about relationship type, add contact person relationship links.
- Create Home Care Episode (*start date and time, additional information*)
- Document results from admittance visit for relevant care categories: Medical, Nursing,… (Health Characteristics of type: anamnesis, status. …, social situation, …..)

Use Case: Discharge patient

Preconditions:
Decision taken to discharge patient.

Triggering:
Actor triggers discharge interaction.

Ends with:
Patient discharged, home care episode ended.

Variations:

1. Patient cured (healthy).
2. Patient has rejected further treatment.
3. Patient taken over by other healthcare party.
4. Patient deceased.
5. Patient discharged.
6. Patient temporarily discharged.

Actions:
If patient cured:

- Issue clinical summary/"epicrisis"
- Inactivate patient's care documentation

If patient rejects further treatment:

- Issue clinical summary
- Inactivate patient's care documentation

If patient is discharged to another healthcare party:

- Issue discharge message
- Issue clinical summary
- Document patient's permission to let the overtaking healthcare party take part of patient's care documentation

- Inactivate patient's care documentation

If patient deceased:

- Issue death certificate

  If doctor requests autopsy:

  - Create autopsy request
  - Document autopsy results
- Plan support actions for the deceased's family/relatives (Curator)
- End patient's Home Care Episode
- Inactivate patient's care documentation

Use Case: Consult specialist service provider

(See Use Cases for Swedish HIS Application where use cases for referral process have been defined).

Use Case: Plan healthcare service actions

Preconditions:
Patient admitted, home care episode started.

Triggering:
Actor triggers planning interaction.

Ends with:
Care plan for patient is up-to-date

Variations:
According to care category (medical, nursing, curative, etc)

Actions:

- Display patient's existing care documentation (for selected category, across categories).
- Display patient's existing care plan if revision.
- Create care plan (*start date and time*, *estimated care time*, *planning contact*).
- Add or modify main goal for care (Health Characteristic, state "Target").
- Add or modify subgoals for care (Health Characteristic, state "Target").
- Add identified problem (Identified Problem: *description*, *severity of problem*).
- Add actions to be taken (Action: classified by Action Type, *request date and time*, *requested execution date and time*, *reason for action*, *urgency level*, *action description*, set *action state*).

Use Case: Perform healthcare service actions

Preconditions:

Triggering:
Actor triggers interaction.

Ends with:
Relevant information about contacts, actions and health characteristics is displayed, added, modified or deleted.

Variations:
According to care category (medical, nursing, curative, etc).

Actions:
Display, add, modify or delete (when applicable):

- Indirect healthcare contacts e.g. Team conferences.
- Direct healthcare contacts, e.g. Home visits.
- Actions (e.g. *start execution date and time*, *end execution date and time*, *report date and time*, set *action state*).
- Health characteristics (e.g. *report date and time*, *health characteristic value*).

7.1.3 Object Class Model

The object class model of the Stockholm EPD application consists of three packages:

- Patient Record
- Professional and Organisations
- Referrals

The Patient Record package is the main package in which the classes of the EPD core application is specified using UML static structure diagramming. For reasons of manageability, the model is presented in 6 diagrams, each focusing a specific aspect/view of the model. Accordingly, the presentation diagrams are named: Patient, Contacts and Actions, Healthcare Characteristics, Healthcare Contact Specialisations, Care Plan and Patient Record Datatypes.

The Professionals and Organisations package contains those classes related to the organisational units and healthcare professional's part of the enterprise, which are referred to in the Patient Record package. It is a sub-set of the Healthcare Parties and Services package of the Stockholm HIS Application.

The package Referrals is just a reference to the Referrals category of the Stockholm HIS Application model.

The model at this stage concentrates on the business objects that the application is about. Interface objects and operations for the classes will be added.

7.1.3.1 Class Diagrams

Diagram: Care Documentation/Patient

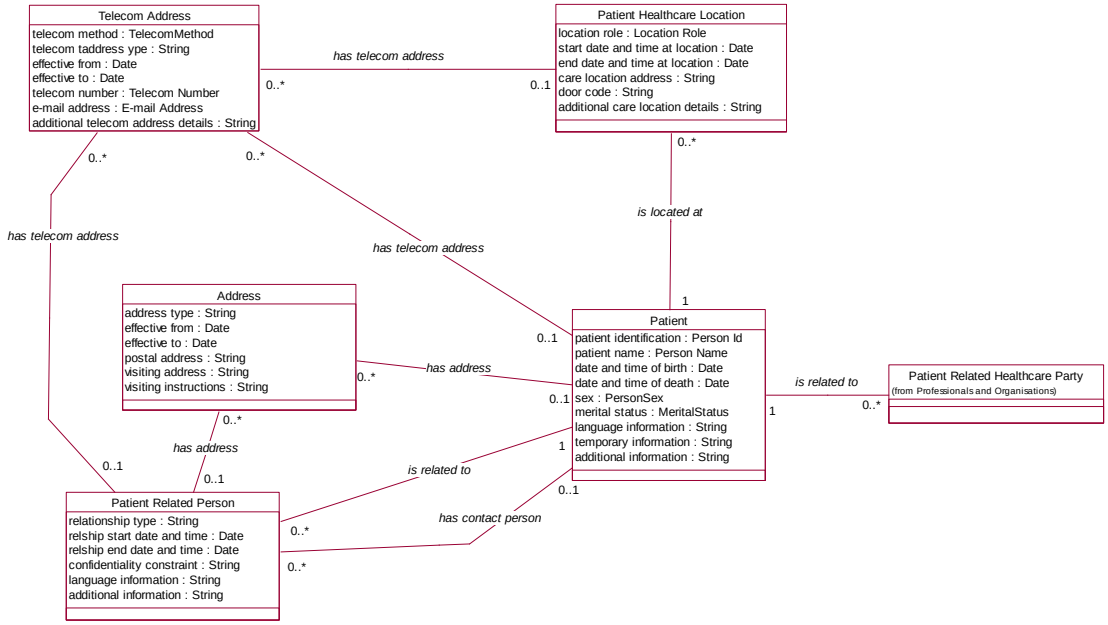


Diagram: Care Documentation/Contacts and Actions

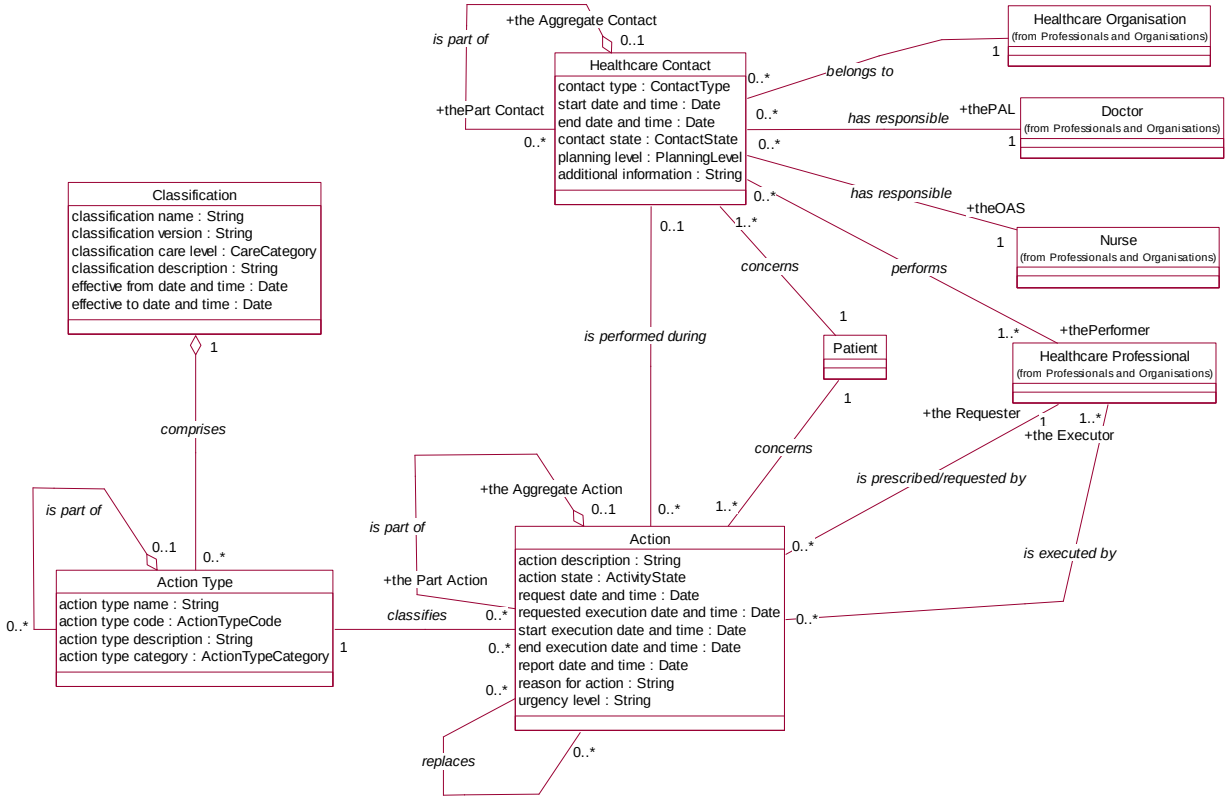


Diagram: Care Documentation/Health Characteristics

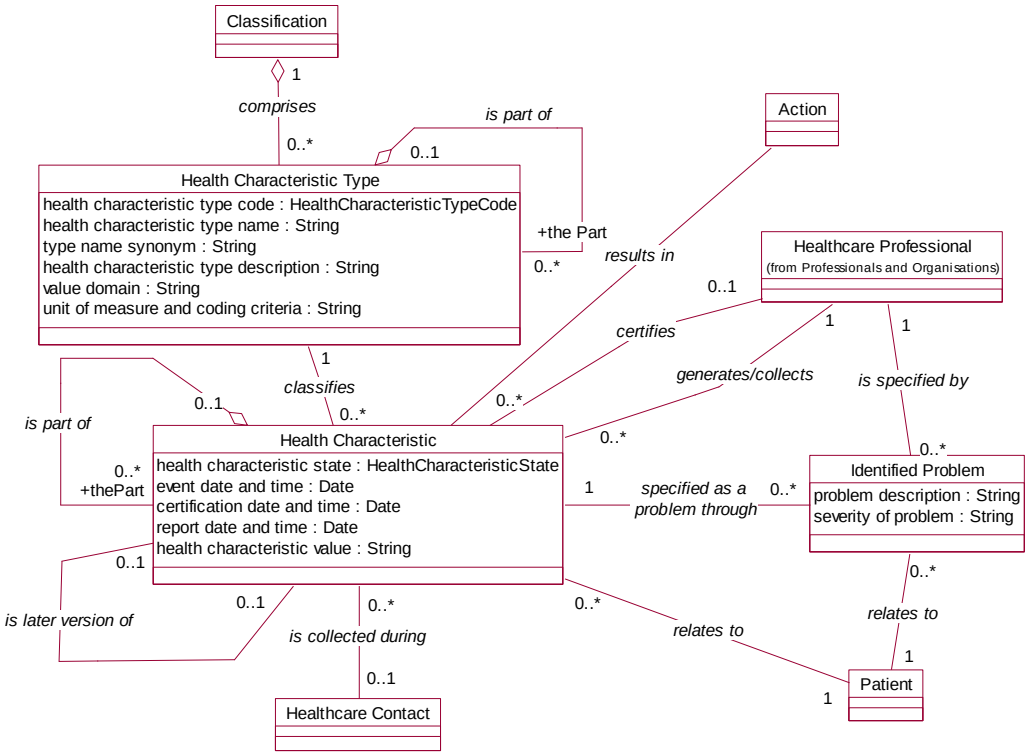


Diagram: Care Documentation/Healthcare Contact Specialisations

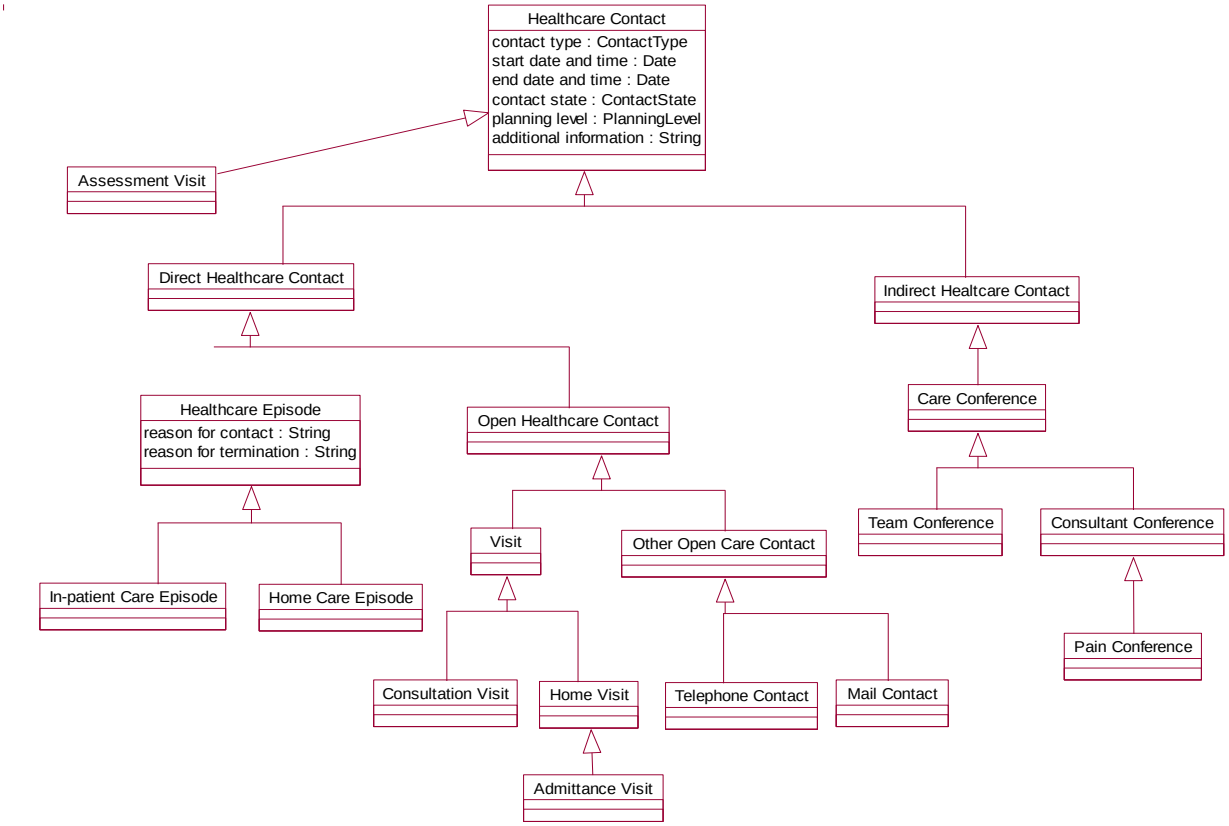


Diagram: Care Documentation/Care Plan

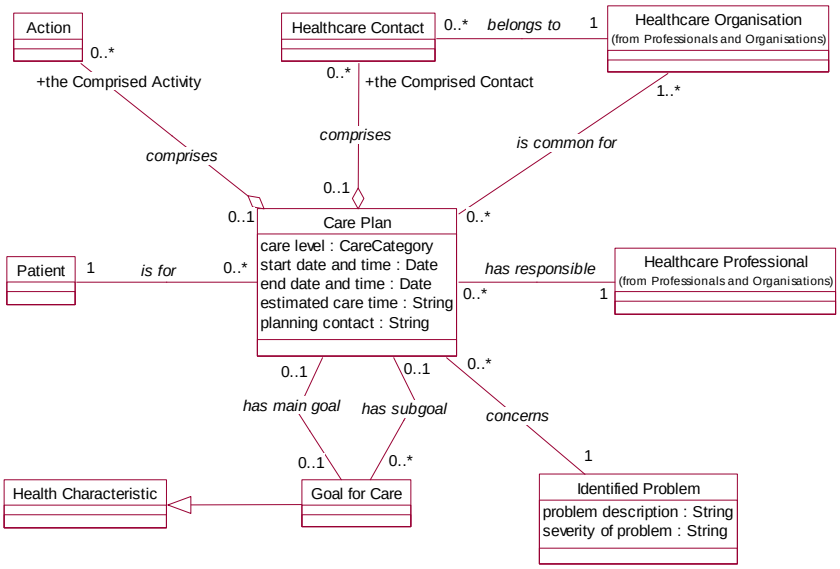
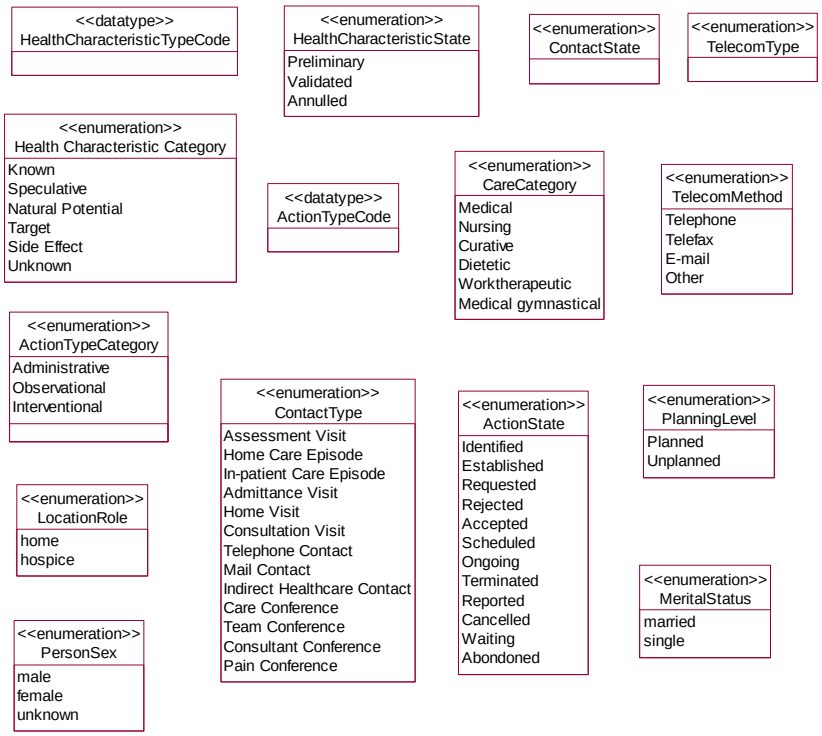


Diagram: Care Documentation/Care Documentation Datatypes



7.1.3.2 Class Descriptions

Healthcare Episode

Derived from Direct Healthcare Contact

Public Attributes:

reason for contact : String
reason for termination : String

Healthcare Contact

Public Attributes:

contact type : ContactType
start date and time : Date
end date and time : Date
contact state : ContactState
planning level : PlanningLevel
additional information : String

Assessment Visit

Derived from Healthcare Contact

Direct Healthcare Contact

Derived from Healthcare Contact

Indirect Healtcare Contact

Derived from Healthcare Contact

In-patient Care Episode

Derived from Healthcare Episode

Home Care Episode
Derived from Healthcare Episode

Home Visit
Derived from Visit

Consultation Visit
Derived from Visit

Telephone Contact
Derived from Other Open Care Contact

Mail Contact
Derived from Other Open Care Contact

Care Conference
Derived from Indirect Healtcare Contact

Team Conference
Derived from Care Conference

Consultant Conference
Derived from Care Conference

Pain Conference
Derived from Consultant Conference

Other Open Care Contact
Derived from Open Healthcare Contact

Open Healthcare Contact
Derived from Direct Healthcare Contact

Visit
Derived from Open Healthcare Contact

Admittance Visit
Derived from Home Visit

Health Characteristic
Public Attributes:
health characteristic state : HealthCharacteristicState
event date and time : Date
certification date and time : Date
report date and time : Date
health characteristic value : String

Action
Public Attributes:
action description : String
action state : ActivityState
request date and time : Date
requested execution date and time : Date
start execution date and time : Date

end execution date and time : Date
report date and time : Date
reason for action : String
urgency level : String

Identified Problem
Public Attributes:
problem description : String
severity of problem : String

Goal for Care
Derived from Health Characteristic

Patient
Public Attributes:
patient identification : Person Id
patient name : Person Name
date and time of birth : Date
date and time of death : Date
sex : PersonSex
merital status : MeritalStatus
language information : String
temporary information : String
additional information : String

Care Plan
Public Attributes:
care level : CareCategory
start date and time : Date
end date and time : Date
estimated care time : String
planning contact : String

Classification
Public Attributes:
classification name : String
classification version : String
classification care level : CareCategory
classification description : String
effective from date and time : Date
effective to date and time : Date

Health Characteristic Type
Public Attributes:
health characteristic type code : HealthCharacteristicTypeCode
health characteristic type name : String
type name synonym : String
health characteristic type description : String
value domain : String
unit of measure and coding criteria : String

HealthCharacteristicTypeCode

HealthCharacteristicState
{Preliminary, Validated, Annulled}

Health Characteristic Category
{Known, Speculative, Natural Potential, Target, Side Effect, Unknown}

Action Type
Public Attributes:
action type name : String
action type code : ActionTypeCode
action type description : String
action type category : ActionTypeCategory

CareCategory
{Medical, Nursing, Curative, Dietetic, Worktherapeutic, Medical gymnastical}

ActionTypeCategory
{Administrative, Observational, Interventional}

ActionState
{Identified, Established, Requested, Rejected, Accepted, Scheduled, Ongoing, Terminated, Reported, Cancelled, Waiting, Abondoned}

PlanningLevel
{Planned, Unplanned}

ContactState

ActionTypeCode

ContactType
{Assessment Visit, Home Care Episode, In-patient Care Episode, Admittance Visit, Home Visit, Consultation Visit, Telephone Contact, Mail Contact, Indirect Healthcare Contact, Care Conference, Team Conference, Consultant Conference, Pain Conference}

Telecom Address
Public Attributes:
telecom method : TelecomMethod
telecom taddress ype : String
effective from : Date
effective to : Date
telecom number : Telecom Number
e-mail address : E-mail Address
additional telecom address details : String

Address
Public Attributes:
address type : String
effective from : Date
effective to : Date
postal address : String
visiting address : String
visiting instructions : String

Patient Related Person
Public Attributes:
relationship type : String
relship start date and time : Date

relship end date and time : Date
confidentiality constraint : String
language information : String
additional information : String

Patient Healthcare Location
Public Attributes:
location role : Location Role
start date and time at location : Date
end date and time at location : Date
care location address : String
door code : String
additional care location details : String

LocationRole
{home, hospice}

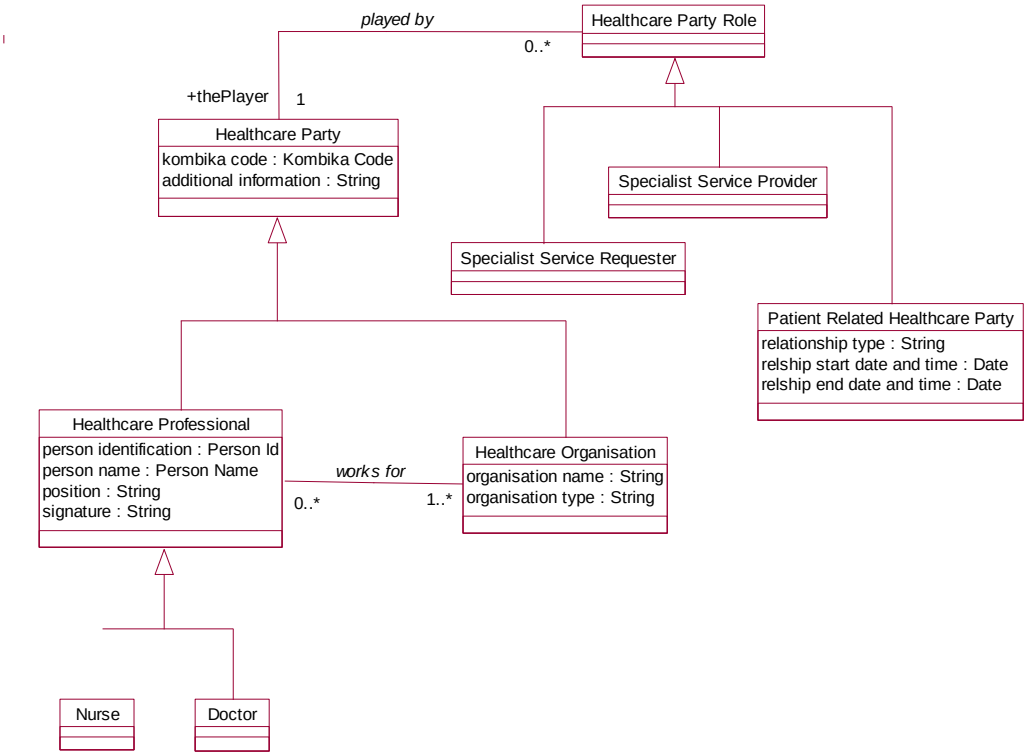
TelecomType

PersonSex
{male, female, unknown}

MeritalStatus
{married, single}

TelecomMethod
{Telephone, Telefax, E-mail, Other}

Diagram: Organisations and Professionals/Main



7.1.4 Implementation Plan

The implementation of the InterCare services in Stockholm North demonstration site will be made in an iterative way during 1999. This approach has been successful in STAR project and allows end-users to continuously develop their procedures and refine their requirements on the information system. The adopted distributed object technology supports this approach.

The identified use cases indicate what kind of IT-support is requested by users in the Palliative Care Unit. The most important requirements are services that supports team-based care planning, a shared patient record, distributed scheduling of activities and referrals. In addition to this, a role-based access control is necessary to secure integrity of the patients.

There is a need for access to clinical data from previous contacts, when assessing referrals of patients to the unit, but these requirements have a low priority compared to services mentioned above. The planned InterCare product EPDS based on components from Synapses will not be sufficient to meet the requirements of an electronic patient dossier. What is needed is a component offering Act Manager services.

An extensive modelling has been carried out, but still more refinements are necessary in order to develop products and applications. This will take place during the coming months. Parallel to this an evaluation of available act managers will take place. In the beginning of 1999 servers will be constructed or adjusted to meet requirements and applications will be developed. Before summer a shared patient dossier prototype (version 1) will be available and verified. This prototype will meet basic requirements, but it should offer services to end-users that are vital to the operations of the unit. The 80/20-rule will be applied. Accessibility to clinical information should be monitored through a security layer using a role-based access control server.

In addition to this, standard software (MS-Outlook) will be used to develop a scheduling module. This module should have the capacity to share objects with Corba. A referral system based on STAR Referral manager and InterCare Enterprise manager will be implemented before summer.

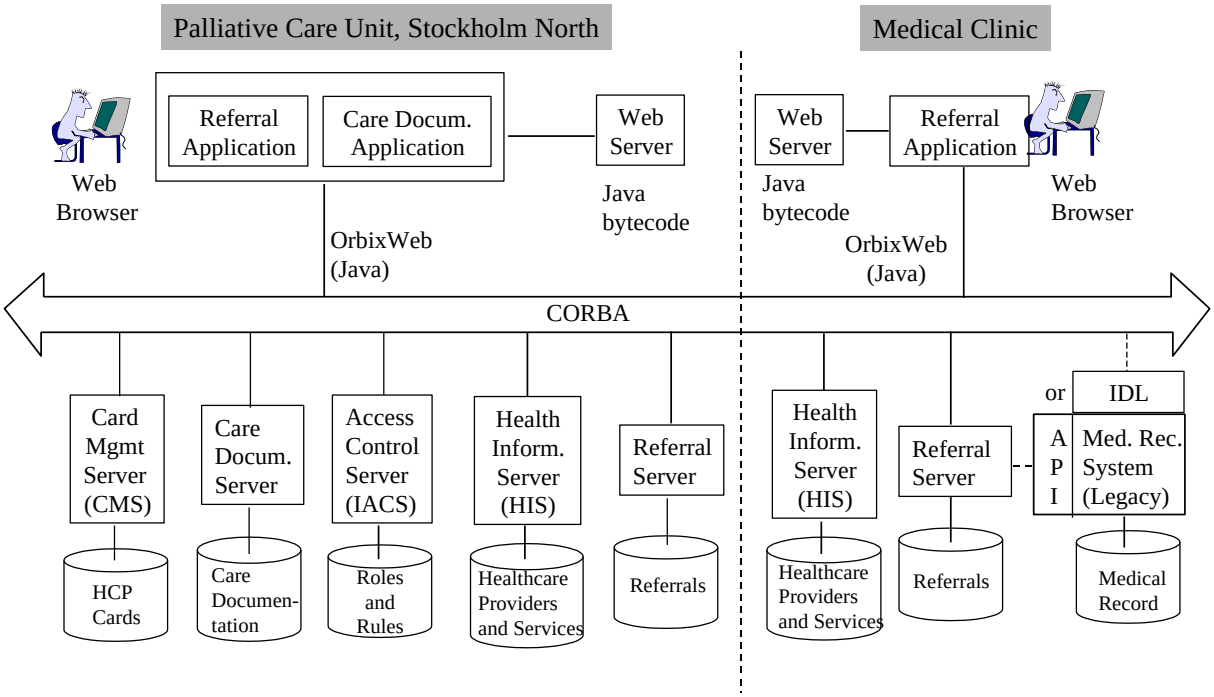
In the second half of year 1999 a refinement of the prototype will take place, the software (version 2) will be brought to a status which allows demonstration to take place at the site.

Demonstration and evaluation will take place during the remaining InterCare project period in year 2000.

7.1.5 Architectural Aspects

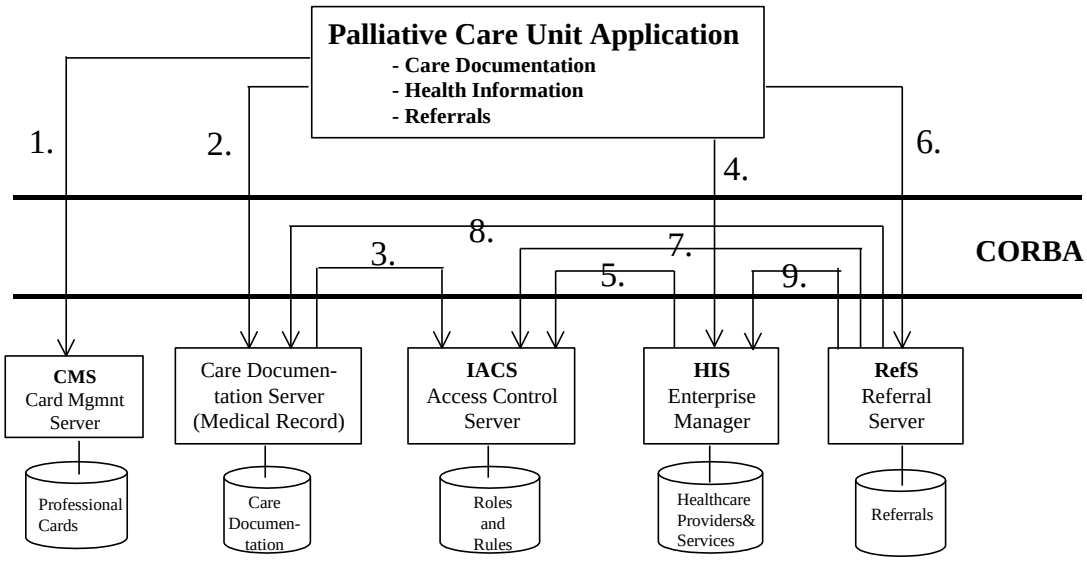
The current situation at the Palliative Care Unit is that all care documentation is handled manually, no computerised patient record is used. Referrals are received and sent by ordinary mail or by fax.

The target architecture is described in the following figure:



Target architecture of the Stockholm North application

The interoperability between the target architecture components is described and explained below.



Component interoperability

1. Identification/authentication of healthcare professional. Certificate returned.
2. Query (get/create/change/delete/..) concerning care documentation. Result (answer to query or message ”permission denied”) returned after access control.
3. Request for access control. Permitted/not permitted returned.
4. Query concerning healthcare information. Result (answer to query or message ”permission denied”) returned after access control.
5. Request for access control. Permitted/not permitted returned.
6. Query concerning healthcare service requests (referrals). Result (answer to query or message ”permission denied”) returned after access control.
7. Request for access control. Permitted/not permitted returned.
8. Query concerning patient demographic information/store signed referral. Result (answer to query or message ”permission denied”) returned after access control.
9. Query concerning service requests (referral template). Access control HIS/IACS ev. delegated to RefS.

7.2 Swedish HIS Demo Application (Stockholm South)

7.2.1 Business Objectives and Purpose of the Application

Below those business objects are stated which are of particular importance to the participating units of the Stockholm South Healthcare Area. It is expected that the support, which the target application will give to the professional people in the organisation when fully operational, will contribute substantially to their fulfilment.

The business objectives are to:

1. Increase the quality of handling referrals between GP units and hospital units i.e. to decrease lead times in the referral process.
2. Improve the quality of referral contents.
3. Improve decision support to healthcare professionals.
4. Increase the patient's influence in matters concerning him or her.

The purposes of the applications are:

1. To have a reliable service events reporter informing about the status of the referral.
2. To place templates at healthcare professionals disposal, prepared with adequate questionnaire from joint liability about various treatments. To make it possible to create attachments to referrals e.g. x-rays, to send them and to open them.
3. To place information about healthcare services at healthcare professionals' disposal with or without the referral application running.
4. To have information easy of access to patients in order to help them influence on different matters concerning him or her.

7.2.1.1 Metrics

The metrics discussed in the project group are

1. Comparing the times it takes for a paper referral and an electronic referral to be properly handled.
2. The possibility to measure decreased number of visits to a specialist clinic if the specialist gets such a good quality on the referral that he/she is able to judge it without a patient present.
3. The use of a Yellow pages database might be measured by comparing the time it takes to pick up information the ordinary way and when using the Yellow pages service.
4. Other valuable things to find out are quality of referral, satisfaction with one's work and patient satisfaction.

Since the goal of this project is to demonstrate a prototype, and not to install a system at several Health Care centres and hospitals in an area, it is probably very difficult to find out moments in the referral process that really influence the way of working at the units taking part in InterCare.

7.2.2 Use Cases and Scenarios

7.2.2.1 Scenarios

The following four scenarios have been developed to describe alternative use situations of the Stockholm HIS application.

Scenario 1: Create referral

Dr Andersson, GP at a Health Care Centre, has given the diagnosis diabetes mellitus to a previously healthy 22-year-old man. The Centre, where Dr Andersson works, has a well-established collaboration with the Medical unit at the nearby hospital. When a debut of diabetes mellitus is discovered one has the routine to hospitalise the patient for a couple of days. Dr Andersson is familiar with this routine and needs no more information to decide whether to make a referral or not to the medical unit.

- Dr Andersson opens STAR referral application, among possible *Healthcare Organisations* she chooses the medical unit.
- From the application she picks the *template* for diabetes mellitus and fills in the specified fields. Since the patient is previously healthy, there is no need for Dr Andersson to get more information from the patient record. The appropriate information is filled in directly into the template provided by the application.
- She *signs* the referral to make it accessible for the hospital medical unit. At the same time, information from the referral is transferred to the patient record.

Scenario 2: Create referral and use Yellow Pages

Example 1

Mr A has recently been through knee surgery and is on his way home after 24-hours stay at the hospital. The physiotherapist, B, and Mr A conclude that Mr A needs continuous help from a physiotherapist, with special orthopaedic competence, to success with rehabilitation.

- B opens the database "*healthcare providers"* (with or without STAR application running).
- The *search function* becomes visible from where B can make a search for e.g. Hospital, Primary Health Care Centre, Private practitioners, physiotherapists, specialists and specific persons or qualifications
- B finds a physiotherapist with special orthopaedic competence.
- B picks a *template.* The referral is filled in, signed and sent to the physiotherapist whom B recommended and A accepted.
- Information from the referral is transferred to the patient record.

Example 2

Maj is 72 years old. She was bitten by a dog, therefore she went to the emergency unit at the closest hospital.

- The wound is taken care of and her tetanus vaccination is refilled.
- Maj is worried about her wound. The physician gives her a recommendation to go to the Primary Health Care Centre on duty tomorrow, Sunday, so the wound will be taken care of.
- STAR application is running and the physician clicks on the icon for *Yellow Pages* to find out about which Health Care Centre is on duty tomorrow.
- A *template* is filled in, signed and thereby made accessible to the appropriate Health Care Centre.

Scenario 3: Create a referral and create attachment

Example 1

The GP needs a short and quick consultation from a specialist physician concerning a patient with skin alterations. Does the patient need to see the specialist at all or can the GP be helped from distance to solve the problem herself?

- From STAR referral application the GP sends a referral where the skin alterations are described. At the same time a *digital photo* of the skin alterations is accessible for the specialist.
- The description of the patient's skin alterations and the possibility to study the picture makes it possible for the specialist to make a judgement without having the patient go to the clinic.
- The specialist immediately writes an answer, which is sent to the GP.
- The *follow up function* in STAR referral application indicates the new referral status to the GP so that he/she will notice the arrival of a referral answer.

Example 2

A 65-year old woman with diabetes ulceration on her feet. The foot therapist treats the ulcerations regularly. By taking digital pictures of the ulceration the therapist has documented the development. The diabetes ulceration has deteriorated and the woman needs to continue her treatment at a specialist clinic.

- The therapist chooses a *referral template*, fills in the needed information according to the template fill-in - instructions. A reference to the digital pictures is attached.
- The specialist examines the referral and the pictures.
- A summon is sent to the patient.
- Answer about how to continue the treatment and how to share the responsibility for it in the future is sent as a referral answer to the foot therapist.
- The follow up function in the STAR referral application makes it easy for the therapist to see when a referral answer has arrived.

Scenario 4: Create referral answer

The last four days a 22-year old man with a recently débuted diabetes mellitus has been hospitalised at the medicine clinic. The patient is on insulin and has received some information about the disease. Dr Olofsson, a new employer at the ward, is responsible for necessary actions when the patient is to leave the hospital. Dr Olofsson has been informed by his superior about the close co-operation between the hospital and the general care units in the area in case of diabetes mellitus. One has a joint liability about the treatment of diabetes written down in a care program. In connection with the patient's leave, Dr Olofsson is urged to take responsibility of letting the care professionals at the general care unit receive appropriate information quickly since the patient shortly needs an appointment there.

- Dr Olofsson starts STAR referral application and selects the referral from GP Dr Andersson.
- Dr Olofsson is recently employed and not well acquainted with all details in the care program. Thanks to the connection between STAR referral application and an information database about care producers he can easily reach the care program about diabetes mellitus.
- With a simple search function he is immediately informed about the three professionals to whom he is to send referrals. He finds out that Dr Andersson is not the patient's ordinary GP, but Dr Bengtsson is. Dr Olofsson chooses to address the referral to Dr Bengtsson, but lets a copy go to Dr Andersson too. The copy becomes the referral answer.
- There are two possible dieticians in the area, one in the North area and one in the South area. The patient prefers the dietician in the North area since that will give him the shortest way to travel.

- Next Dr Olofsson gets a template for diabetes mellitus. I that moment he becomes aware of that he has to choose profession to be able to get the right template. In this case he has to use three slightly different templates since the three professions need different information. It is not necessary for the doctor to explain suggestions from the hospital how to continue the treatment since his superior has recommended the general health care centre to follow the joint liability care program. In the referral Dr Olofsson refers to the care program.
- Then Dr Olofsson sends the three referrals, plus a copy to Dr Andersson, by signing the referral.
- A message will appear on the screen at Dr Bengtsson, nurse Olle and dietician Sven that announces ”referral answer” has arrived.
- By a simple click the professionals can read the referrals and give the patient an appointment time, a summon is automatically printed.

7.2.2.2 Use Cases

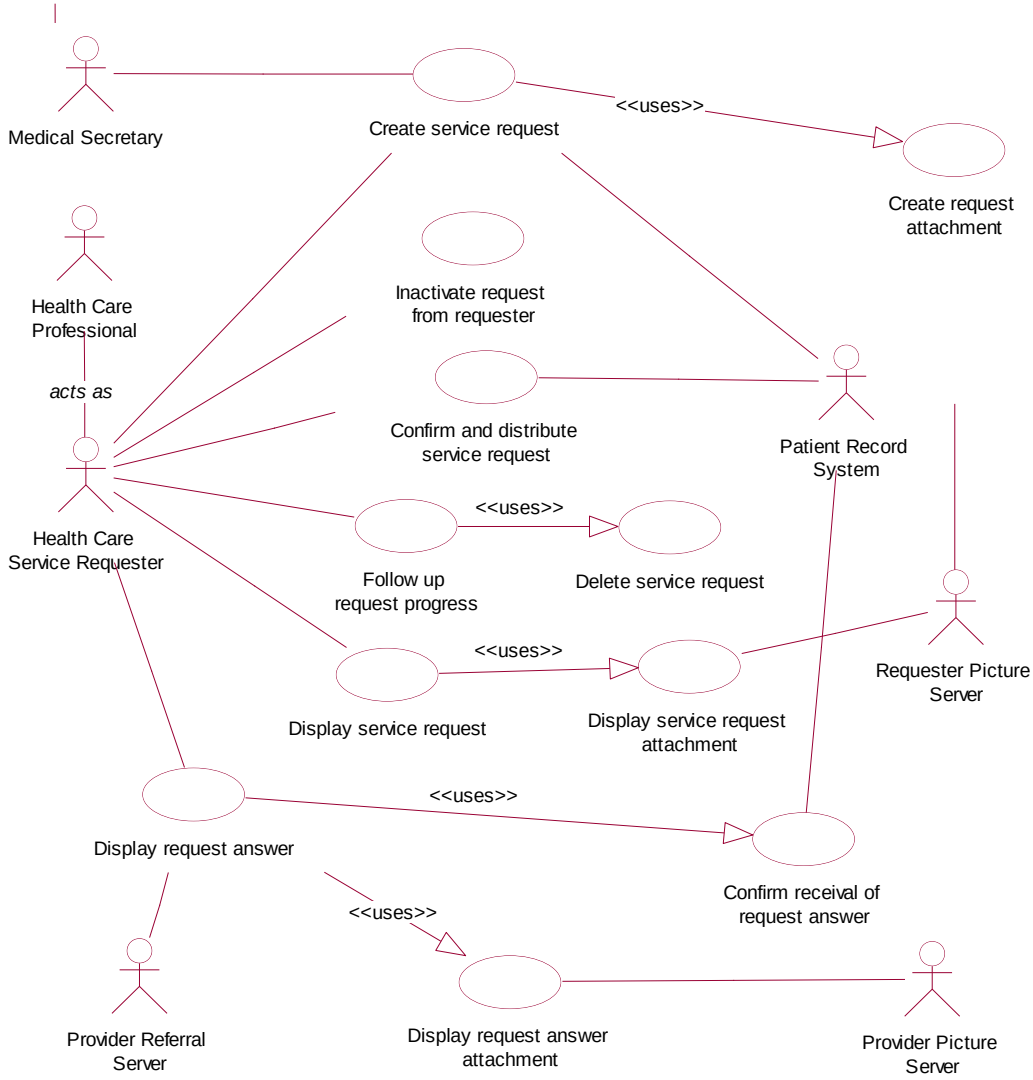
With these scenarios as a background, the detailed Use Case View of the Swedish Health Information System Application has been developed and are described in four diagrams:

1. Request Healthcare Service (incl. attaching pictures to referrals)
2. Provide Healthcare Service (incl. attaching pictures to referral answer reports)
3. HIS Application
4. Picture Management

The four diagrams follow below in sequence. After each diagram the use cases in the diagram are described in natural language. Each use case description is made according to a template containing the following sections:

- Preconditions
- Triggering
- Ends with/Results in
- Variations
- Actions

7.2.2.2.1 Request Healthcare Service - Use Case Diagram



7.2.2.2.2 Use Case Descriptions

Use Case: Create service request

Preconditions:
When applicable, Legacy patient record system running.

Triggering:
User wants to create a service request i.e. a referral.

Ends with:
Referral contents is completely defined and the referral is ready to be confirmed/signed and distributed.

Variations:
Started but incomplete referral is temporarily saved, or

Started referral is cancelled.

Actions:
If *new* referral:

1. User identifies provider and service.
2. The prescribed referral template for selected provider/service is displayed with filled-in provider data.
3. User identifies patient (*patient identification*).
4. Patient data (*patient identification, patient name, address, home telephone number* and *job telephone number*, is, if applicable, fetched from legacy patient record system and added to form.
5. User fills in the needed information according to form template and fill-in instructions.
6. User may add attachment(s) to referral (Use Case: Create service request attachment**)**
7. User marks the referral as ”completed and ready to be signed”.
8. Referral completeness is checked. If incomplete – user corrects.
9. When correct – referral state of Referral is set to ”ReferralCompleted”.

User may stop anywhere in this sequence of actions by saving the started referral (referral state is set to ”ReferralStarted”) or by cancelling the whole operation.

If *started* referral:

1. Referrals with state ”ReferralStarted” are displayed.
2. User selects and opens referral.
3. Started referral is displayed.
4. User continues to enter referral contents according to actions 5-8 above.

Use Case: Create service request attachment

Preconditions:
User is in the process of creating a referral.

Triggering:
User wants to add attachment to the current referral.

Ends with:
One or more attachments added to referral.

Actions:

1. User selects among available attachment sources (legacy patient record, stills stores, etc).
2. User selects source.
3. Available items from this source are presented (addressable items from patient record, still picture files, etc).
4. User selects item to be attached (*reference to document*).
5. System numbers selected item in sequence according to in which order selected (*attachment number*).

6. User names attachment (*attachment name*).
7. User describes attachment (*attachment text*).

Use Case: Confirm and distribute service request

Preconditions:
One or more referrals, which are completed and ready to be signed, do exist.

Triggering:
User wants to confirm and distribute referral to provider.

Ends with:
Referral signed, distributed to provider and, if applicable, stored in (legacy) patient record.

Actions:

1. Referrals in the state of ”ReferralCompleted” are listed.
2. User selects referral to sign.
3. The contents of the selected referral is displayed.
4. User signs referral.
5. *referral state* of Referral is set to ”ReferralSigned” and *signing date* is set to current date
6. The signed referral is distributed to provider (Referral Request is created and linked to Referral via association Referral *establishes* Referral Request).
7. Referral is communicated to Patient record system where it is stored.

Use Case: Follow up request progress

Preconditions:
One or more referrals, which are signed and distributed, do exist.

Triggering:
User wants to follow up the progress of referrals.

Ends with:
List of active referrals according to selection criteria has been displayed.

Actions:

1. User selects the set of active referrals to be followed up:
   - all referrals for the Healthcare Organisation (i.e. handled by this Referral Manager)
   - the requestor’s ”own” referrals, i.e. referrals which *is remitted by* the current user
   - the referrals which *concerns* a specific *patient*, in which case the *patient identification* is entered
2. User sorts list in preferred sort order (default is by *signing date*)

Use Case: Delete service request

Preconditions:
Referral is in the state of ”ActDoneBy Requester” and Referral Request linked to referral via association establishes is in the state of ”ActDoneByProvider”.

Triggering:
When list of ”active” referrals is generated in Use case: Follow up request progress.

Ends with:
Referral is deleted.

Actions:

1. Delete referral (objects Referral, Referral Request, Referral Answer (one or more), Referral Answer Request (one or more) plus other dependent linked objects)

Use Case: Display service request

Preconditions:
Referral or Referral and Referral Request do exist.

Triggering:
User wants to look at the contents of a referral.

Ends with:
Referral contents is displayed.

Variations:

1. Referral is started but not confirmed (signed). Actor: Health Care Service Requester.
2. Referral is confirmed. Actors: Health Care Service Requester, Health Care Service Provider.

Actions:

1. User selects referral to be displayed.
2. Referral contents is displayed.
3. If there exist one or more attachments to the current referral, user may trigger Use case: Display service request attachment.

Use Case: Display service request attachment

Preconditions:
Attachment(s) to the current referral do exist.

Triggering:
User wants to look at attachment(s).

Ends with:
The contents of the selected attachment(s) is displayed.

Actions:

1. User selects attachment to be displayed.
2. Attachment contents is displayed.
3. User ends use case and continues in Use case: Display service request.

Use Case: Display request answer

Preconditions:
Referral Answer or Referral Answer and Referral Answer Request do exist.

Triggering:
User wants to look at the contents of a referral answer.

Ends with:
Referral answer contents is displayed.

Variations:

1. Referral answer is started but not confirmed (signed). Actor: Health Care Service Provider.
2. Referral answer is confirmed. Actors: Health Care Service Requester, Health Care Service Provider.

Actions:

1. User selects referral answer to be displayed.
2. Referral answer is displayed.
3. If there exist one or more attachments to the current referral answer, user may trigger Use case: Display service request attachment.

Use Case: Display request answer attachment

Preconditions:
Attachment(s) to the current referral answer do exist.

Triggering:
User wants to look at attachment(s).

Ends with:
The contents of the selected attachment(s) are displayed.

Actions:

1. Available attachments are listed.
2. User selects attachment to be displayed.
3. Attachment contents is displayed.
4. User ends use case and continues in Use case: Display request answer.

Use Case: Confirm receipt of request answer

Preconditions:
The referral answer for which the receipt by the requester is going to be confirmed do exist and has been made current through Use case: Display request answer (Referral Answer Request is in the state of ”AnswerRequestStarted”).

Ends with:
The receipt of a referral answer has been confirmed (signed) by the requester.

Actions:

1. User signs receipt of current referral answer.
2. The state of the Referral Answer Request is set to ”AnswerRead”.

Use Case: Inactivate request from requester

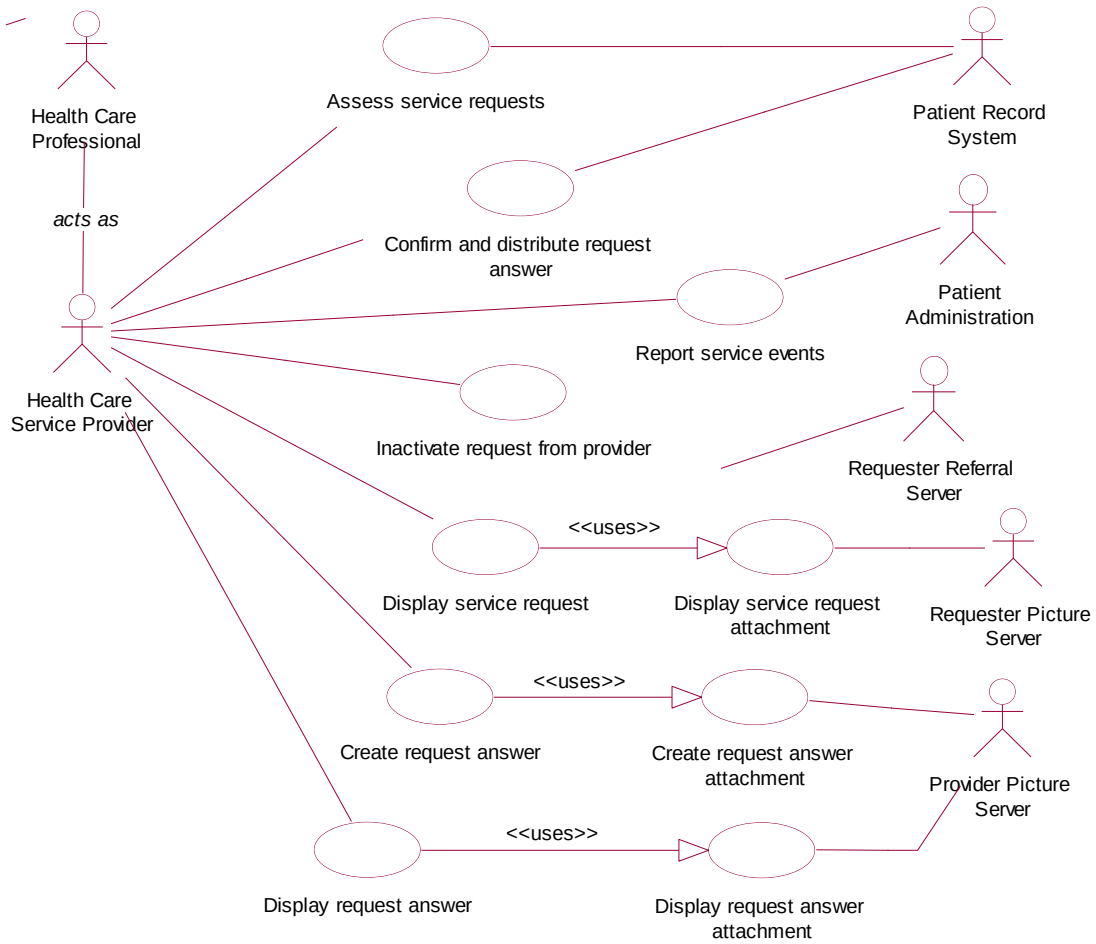
Preconditions:
Referral is active.

Ends with:
Referral is inactivated at requester side and will not appear in follow up lists.

Actions:

1. Requesting user indicates that referral is no longer of interest for her/him and may be deleted.
2. *referral state* of Referral is set to ”ActDoneByRequester”.

7.2.2.2.3 Provide Healthcare Service – Use Case Diagram



7.2.2.2.4 Use Case Descriptions

Use Case: Assess service request

Preconditions:

Triggering:
User wants to assess incoming referrals.

Ends with:
Referral assessed and decided to be accepted or rejected, or decision postponed.

Actions:

1. Referral requests in the state of ”RequestStarted” are listed.
2. User selects referral request to assess.
3. User displays referral and referral attachments (Use case: Display referral)
4. User marks Referral request as accepted or rejected, identifies Healthcare professional to be responsible and puts a priority on the request.
5. If decision is not taken user ends the use case or selects a new referral request to assess.
6. *referral request state* of Referral Request is set to ”RequestAssessed”, *assessment date* to current date, *performers priority* to the value entered, and Referral Request is linked to Specialist Service Provider via associations *is responsible for* and *assesses*.
7. User selects a new referral request or ends the use case.

Use Case: Report service events

Preconditions:

Triggering:
User, Legacy Patient Record System or Legacy Patient Administration System wants to report on planned or performed service events.

Ends with:
Planned service event details or outcome of executed service action has been reported and made available for requester when following up request progress.

Actions:

Use Case: Create request answer

Preconditions:

Triggering:
User wants to create a referral answer.

Ends with:
Referral answer contents is completely defined and the referral answer is ready to be confirmed/signed and distributed.

Variations:
Started but incomplete referral answer is temporarily saved, or
Started referral answer is cancelled.

Actions:
If *new* referral answer:

1. User identifies referral to be answered
2. The prescribed referral answer template for selected provider/service is displayed.
3. User fills in the needed information according to form template and fill-in instructions.
4. User may add attachment(s) to referral answer (Use Case: Create request answer attachment**)**
5. User marks the referral answer as ”completed and ready to be signed”.
6. Referral answer completeness is checked. If incomplete – user corrects.
7. When correct – referral answer state of Referral answer is set to ”ReferralAnswerCompleted”.

User may stop anywhere in this sequence of actions by saving the started referral answer (state is set to ”ReferralAnswerStarted”) or by cancelling the whole operation.

If *started* referral answer:

1. Referral answers with state ”ReferralAnswerStarted” are displayed.
2. User selects and opens referral answer.
3. Started referral answer is displayed.
4. User continues to enter referral answer contents according to actions 3-7 above.

Use Case: Create request answer attachment

Preconditions:
User is in the process of creating a referral answer.

Triggering:
User wants to add attachment to the current referral answer.

Ends with:
One or more attachments added to referral answer.

Actions:

1. User selects among available attachment sources (legacy patient record, stills stores, etc).
2. User selects source.
3. Available items from this source are presented (addressable items from patient record, still picture files, etc).
4. User selects item to be attached (*reference to document*).
5. System numbers selected item in sequence according to in which order selected (*attachment number*).
6. User names attachment (*attachment name*).
7. User describes attachment (*attachment text*).

Use Case: Confirm and distribute request answer

Preconditions:
One or more referral answers, which are completed and ready to be signed, do exist.

Triggering:
User wants to confirm and distribute referral answer to requester.

Ends with:
Referral answer signed, distributed to requester and, if applicable, stored in (legacy) patient record.

Actions:

1. Referrals in the state of ”ReferralCompleted” are listed.
2. User selects referral to sign.
3. The selected referral is displayed.
4. User signs referral.
5. Referral state of Referral is set to ”ReferralSigned”
6. The signed referral is distributed to provider (Referral Request is created and linked to Referral via association Referral establishes Referral Request).
7. Referral is exported to Patient record system where it is stored.

Use Case: Inactivate request from provider

Preconditions:
Referral is active (Referral Answer Request is in the state of ”AnswerRead”).

Ends with:
Referral is inactivated at requester side and will not appear in follow up lists.

Actions:

1. Providing user indicates that referral is no longer of interest for her/him and may be deleted.
2. *referral state* of Referral is set to ”ActDoneByProvider”.

Use Case: Display request answer attachment

Preconditions:
Attachment(s) to the current referral do exist.

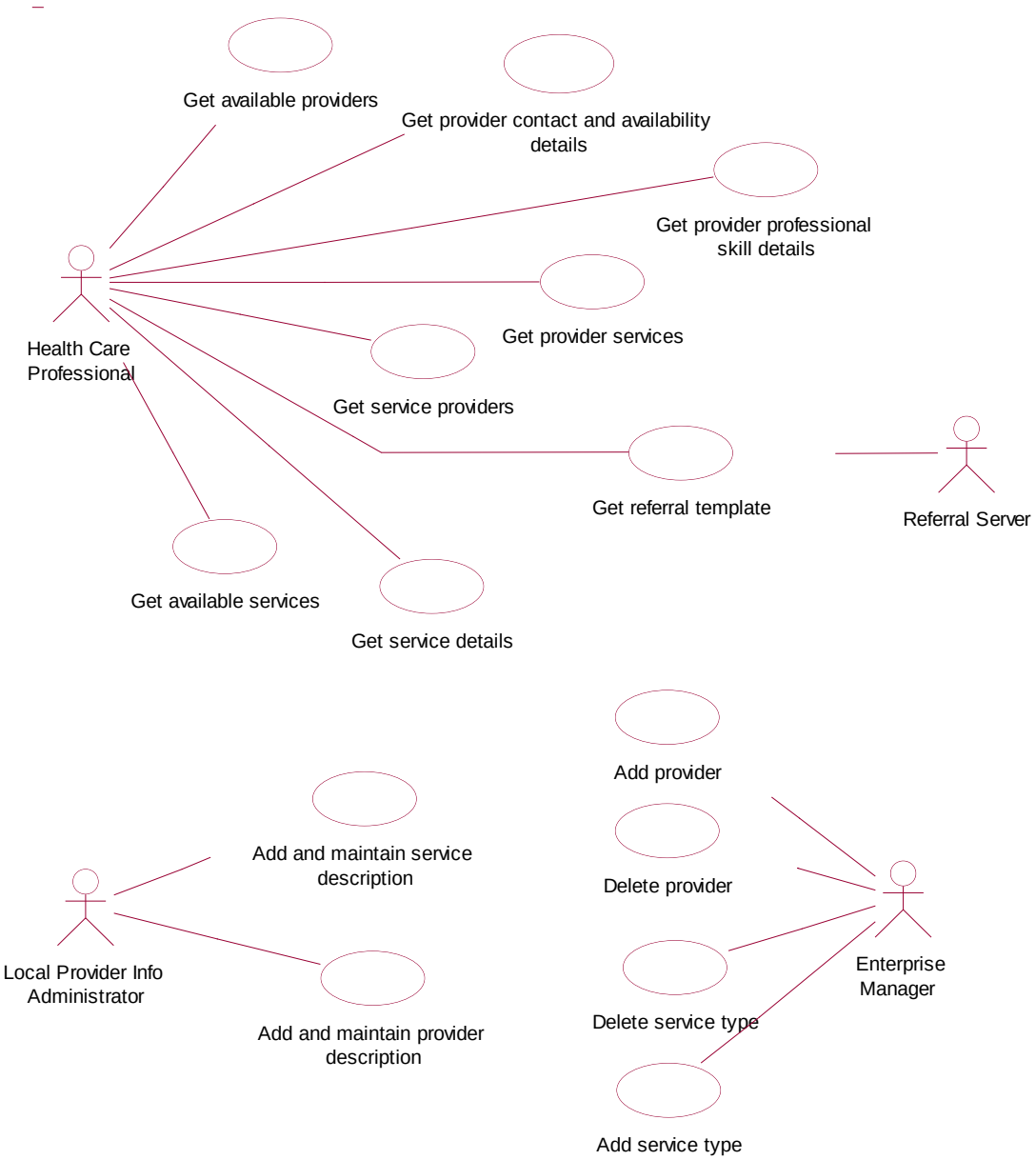
Triggering:
User wants to look at attachment(s).

Ends with:
The contents of the selected attachment(s) are displayed.

Actions:

1. User selects attachment to be displayed.
2. Attachment contents is displayed.
3. User ends use case and continues in Use case: Display service request.

7.2.2.2.5 HIS Application – Use Case Diagram



7.2.2.2.6 Use Case Descriptions

Use Case: Get available providers

Ends with:
Available providers according to the given selection criteria are displayed.

Use Case: Get provider contact and availability details

Ends with:
Contact and availability details for the selected provider are displayed.

Use Case: Get provider professional skill details

Ends with:
Professional skill details for the selected provider are displayed.

Use Case: Get provider services

Ends with:
Available service types of the provider selected are displayed.

Use Case: Get available services

Ends with:
Available service types according to given selection criteria are displayed.

Use Case: Get service details

Ends with:
The detailed description of the service selected (provider/service type) are displayed.

Use Case: Get service providers

Ends with:
Available providers for the service type selected are displayed.

Use Case: Get referral template

Ends with:
The referral template for the selected provider/service is displayed.

Use Case: Add and maintain service description

Ends with:
The service description (service details, referral act templates) for a certain provider/service type is added or updated.

Use Case: Add and maintain provider description

Ends with:
The description (provider basic data, telecom addresses, addresses, availability, professional skills) of a certain provider is added or updated.

Use Case: Add provider

Ends with:
New provider added.

Use Case: Add service

Ends with:
New service added.

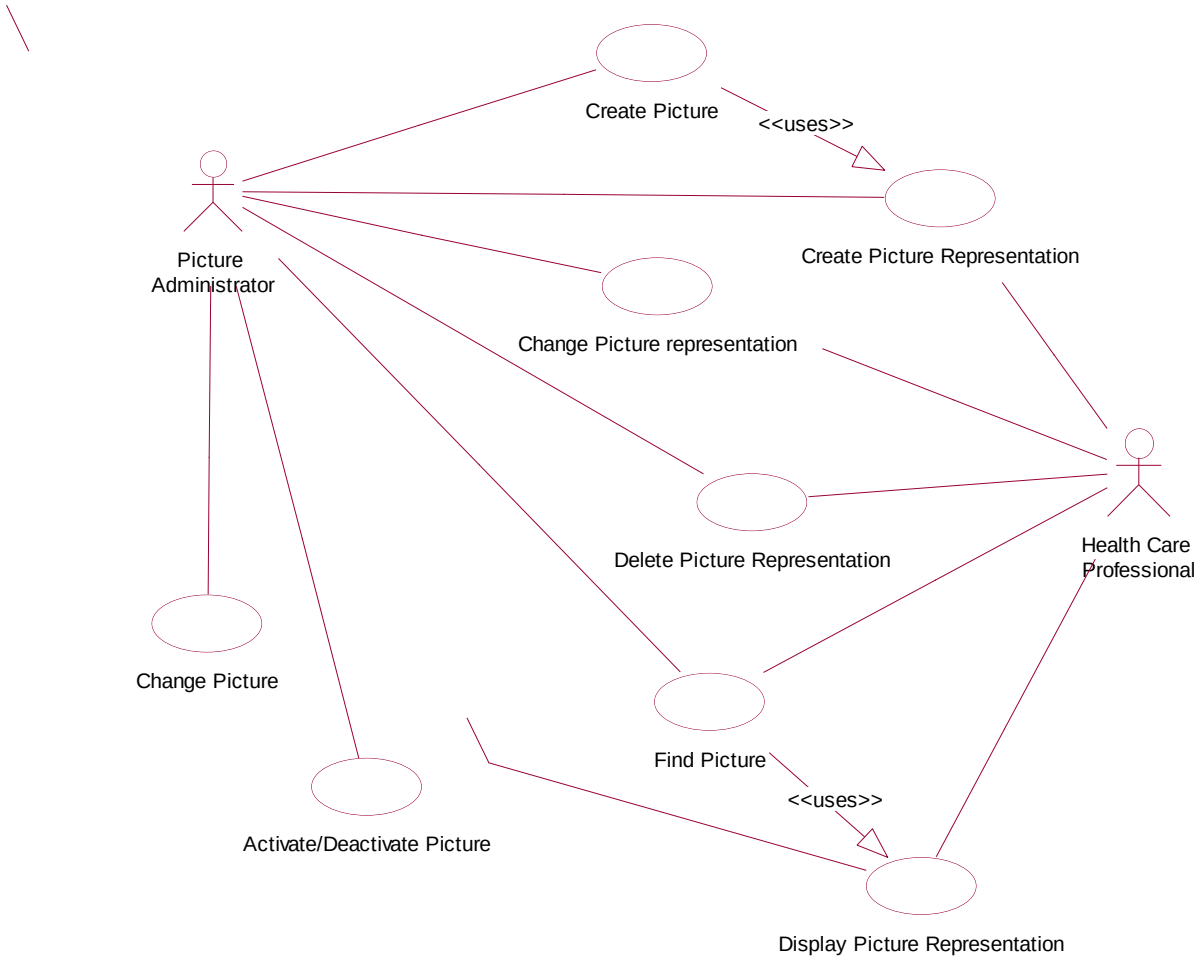
Use Case: Delete provider

Ends with:
Provider deleted.

Use Case: Delete service

Ends with:
Service deleted.

7.2.2.2.7 Picture Management - Use Case Diagram



7.2.2.2.8 Use Case Descriptions

Use Case: Create picture

Preconditions:
The Picture identified must not exist before.

Triggering:
The Picture Administrator wants to create a new picture and triggers the interaction.

Ends with:
A Picture and its original Picture Representation including the Picture Blob have been created and stored in the picture database managed by the local Picture Manager.

Variations:

1. The Patient concerned by the Picture does exist in the Picture database.
2. The Patient concerned by the Picture does not exist in the Picture database.

Actions:
To create a Picture means to create a Picture object, the original Picture Representation object, which the Picture object is represented by, and the Picture Blob object having the contents of the Picture Representation object. If the Patient concerned does not exist, a new Patient object is created.

Picture:

- Actor enters *picture identification*, *patient identification* for the patient concerned, and *organisation identification* for the owning healthcare unit.
- Check: the picture identified must not exist before, the identified organisation unit must be the unit of the Picture Manager used, or a sub-unit of this unit.
- If the identified Patient does not exist, a new Patient object is created. Actor enters *patient name* in addition to *patient identification.*
- Actor enters optionally *picture name*, *picture description* and *keywords.*
- *creation time* is set to current time, *picture state* is set to ”Active”, *created by* is set to the person name of the actor.

Picture Representation and Picture Blob:

- <<uses>> Create Picture Representation, variation 1.

Use Case: Change picture

Preconditions:
The Picture to be changed must exist and be in the state of ”Active”.

Triggering:
The Picture Administrator wants to change attribute value(s) of an existing picture and triggers the interaction.

Ends with:
The attribute value(s) of the Picture has been changed.

Variations:

Actions:

1. Actor identifies picture to be changed by entering the picture identification (or via Use case: Find picture).
2. Actor enters new value for changeable attribute(s): *picture name, picture description, keywords*.

Note: The attributes that may be changed are included with the sole purpose of making it easier to find relevant pictures. They bear no clinical information to be traceable.

Use Case: Activate/Deactivate picture

Preconditions:
The Picture to be deactivated must exist and be in the state of ”Active”. The Picture to be activated must exist and be in the state of ”Inactive”.

Triggering:
The Picture Administrator wants to deactivate an active picture, or activate an inactive picture, and triggers the interaction.

Ends with:
The Picture has been deactivated/activated.

Variations:

1. Deactivation of an active picture.
2. Activation of an inactive picture.

Actions:

1. Actor identifies picture to be deactivated/activated by entering the picture identification (or via Use case: Find picture).
2. If deactivation: *deactivation time* is set to current time and *picture status* is set to ”Deactivated”, if activation: deactivation time is set to ”None” and *picture status* is set to ”Activated”.

No history of deactivations/activations is kept.

Use Case: Create picture representation

Preconditions:
The Picture to which to add the representation to be created must exist and be in the state of ”Active”.

Triggering:
The Picture Administrator/Health Care Professional wants to add a new picture representation and triggers the interaction.

Ends with:
A new Picture Representation and its corresponding Picture Blob has been created and added as a representation of the Picture.

Variations:

1. Creation of the original representation
2. Creation of an additional representation

Actions:

1. Actor selects Picture to which to add a new representation by entering the picture identification (or via Use case: Find picture).
2. The interaction proceeds according to description below. The original representation for a Picture can only be created by the Picture Administrator when creating the Picture.

Picture Representation:

- Actor, or system, enters/sets the attributes: *compression method*, *picture size*, *resolution* and *colors*.
- Actor enters *additional information.*
- If the original representation, *original representation* is set to ”True”, else to ”False”.
- *creation time* is set to current time
- Object is linked to *the picture represented.*

Picture Blob:

- System derives blob identification from picture identification and compression method.
- System derives blob size.
- Object is linked to the representation contained.

Use Case: Change picture representation

Preconditions:
The Picture Representation for which to change an attribute value must exist and the Picture, which it represents, must be in the state of ”Active”.

Triggering:
The Picture Administrator/Health Care Professional wants to change an attribute value of a Picture Representation and triggers the interaction.

Ends with:
The attribute value of the Picture Representation has been changed.

Variations:

1. Changing an attribute value of the original Picture Representation.
2. Changing an attribute value of a Picture Representation other than the original one.

Actions:

1. Actor selects Picture by entering the picture identification (or via Use case: Find picture).
2. Actor selects the Picture Representation for which to change an attribute from selection list.
3. Actor enters the new attribute value. There is only one changeable attribute: *additional information*.
4. The attribute is updated.

Rules:
The original Picture Representation may only be changed by the Picture Administrator.

Use Case: Delete picture representation

Preconditions:

The Picture Representation to be deleted must exist. The state of the Picture represented must be ”Active”. The Picture Representation must not be the original one.

Triggering:
The Picture Administrator/Health Care Professional wants to delete a picture representation and triggers the interaction.

Ends with:
The Picture Representation has been deleted.

Variations:

Actions:

1. Actor selects Picture by entering the picture identification (or via Use case: Find picture).
2. Actor selects the Picture Representation to delete from selection list.
3. Actor confirms deletion.
4. The Picture Representation is deleted.

Use Case: Find picture

Preconditions:

Triggering:
The Picture Administrator/Health Care Professional wants to find a picture and triggers the interaction.

Ends with:
The Picture has been identified, its description is displayed and its existing representations listed.

Variations:

1. Actor knows the picture identification
2. Actor enters search arguments and identifies the picture by successive qualifications and displaying picture representations (thumbnail representations may eventually be used).

Actions:
Possible search arguments should be:

- patient identification
- organisation identification
- picture name (substrings)
- picture description (substrings)
- keywords

The search arguments should be able to use separately or in combination.

Use Case: Display picture representation

Preconditions:

Triggering:
The Picture Administrator/Health Care Professional wants to display a representation of a picture and triggers the interaction.

Ends with:
The selected Picture Representation is displayed.

Variations:

Actions:

1. The actor selects the picture representation to be displayed.
2. The picture representation is displayed together with its description.

7.2.3 Object Class Model

The object class model (Logical View) of the Stockholm South application, which includes the HIS application, consists of three packages: Healthcare Parties and Services, Referrals and Pictures.

The Healthcare Parties and Services package is presented in five diagrams, each focusing a specific aspect/view of the model. The presentation diagrams are named: Yellow Pages, Providers and Services, Role Specialisations, Enterprise Manager, and Parties and Services Datatypes.

The Referrals package contains four diagrams: Main, Attachments, Referral Manager, and Referral Data Types and Exceptions.

The Pictures package contains three diagrams: Main, Picture Manager, and Picture Data Types and Exceptions.

The model in this specification concentrates on the business objects that the application is about.

7.2.3.1 Healthcare Parties and Services

7.2.3.1.1 Healthcare Parties and Services – Class Diagrams

Diagram: Healthcare Parties and Services/Yellow Pages

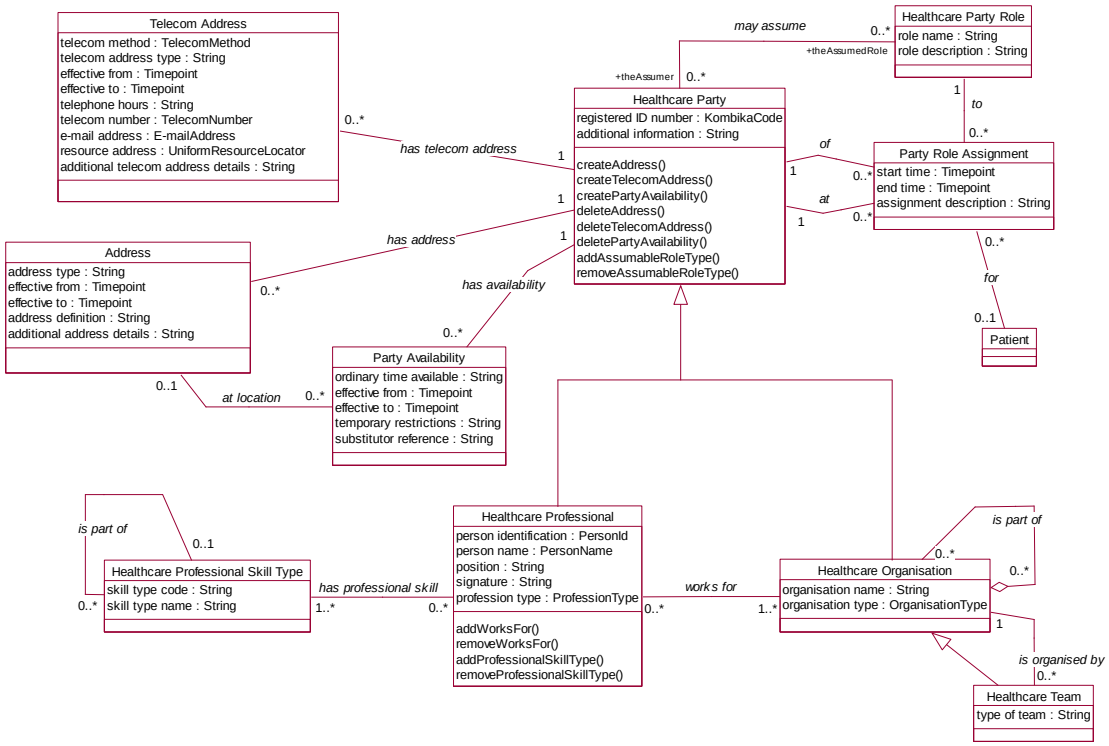


Diagram: Healthcare Parties and Services/Providers and Services

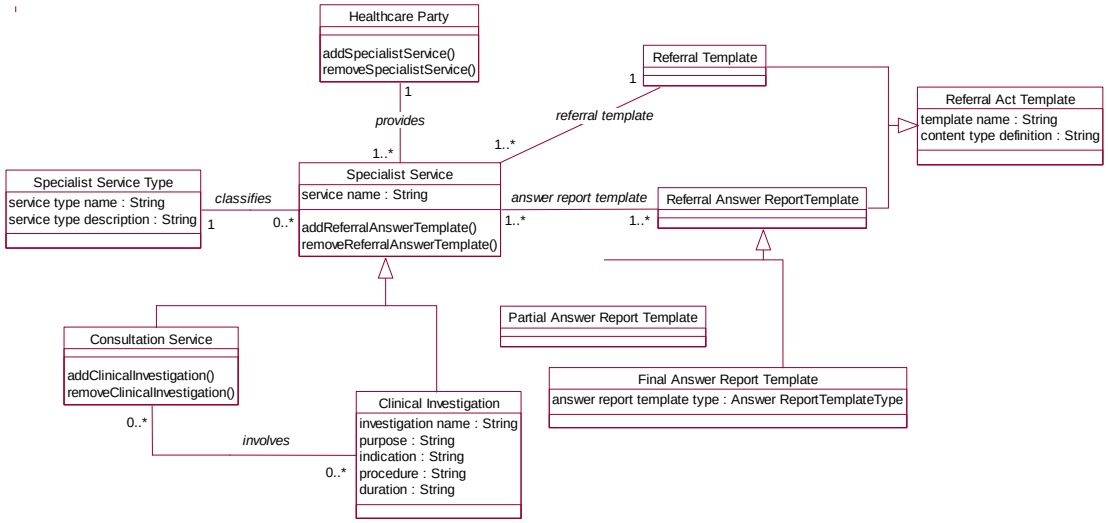


Diagram: Healthcare Parties and Services/Enterprise Manager

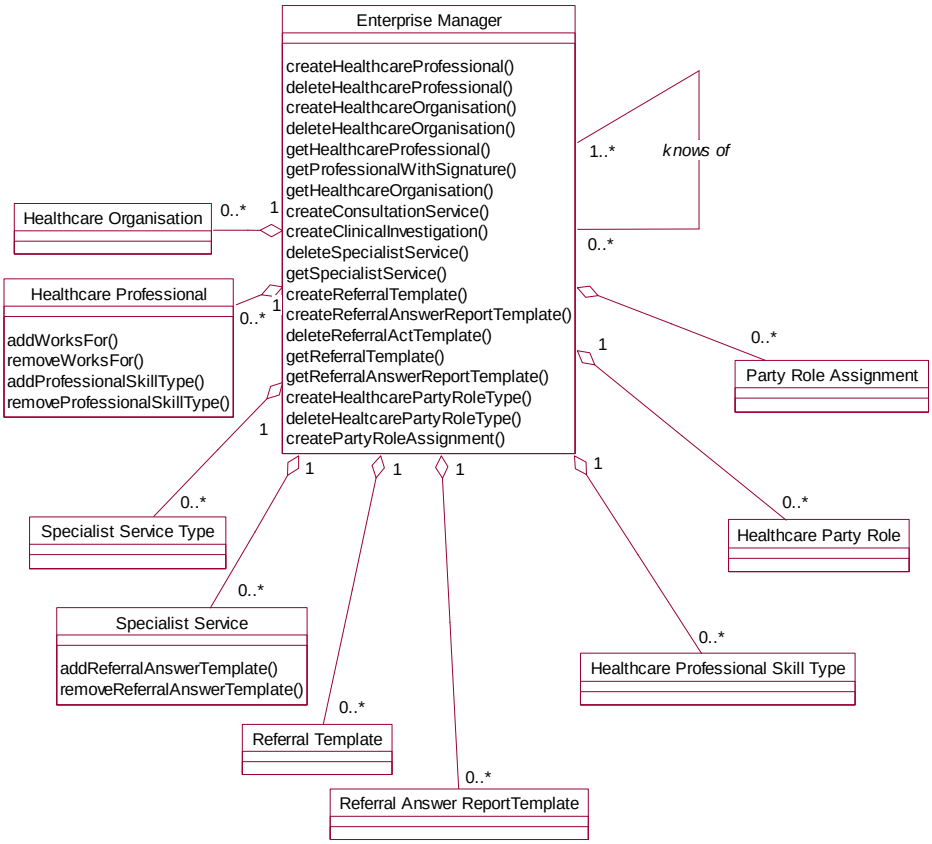
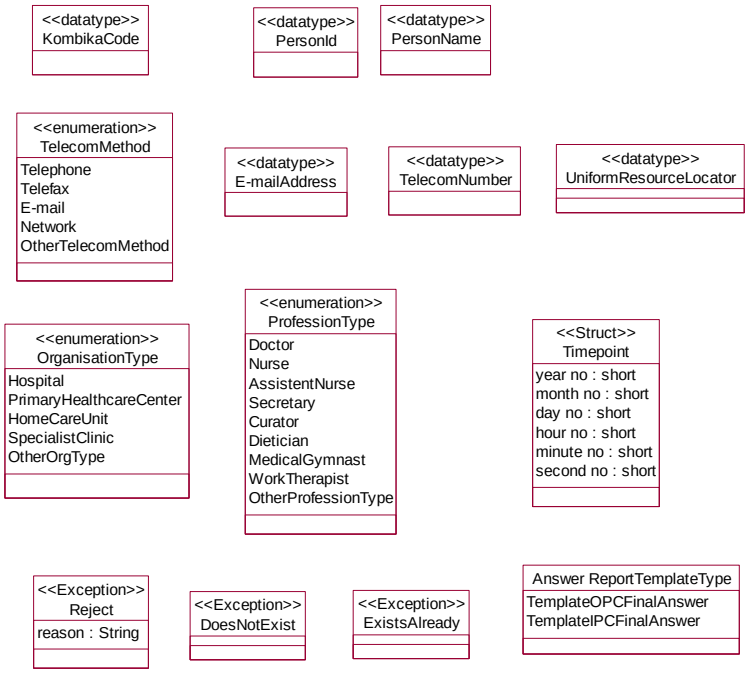


Diagram: Healthcare Parties and Services/Parties and Services Datatypes



7.2.3.1.2 Class Descriptions

Address
Public Attributes:
address type : String
effective from : Timepoint
effective to : Timepoint
address definition : String
additional address details : String

Clinical Investigation
Derived from Specialist Service
Public Attributes:
investigation name : String
purpose : String
indication : String
procedure : String
duration : String

Consultation Service
Derived from Specialist Service
Public Operations:
addClinicalInvestigation (investigation : Clinical Investigation) :
removeClinicalInvestigation (investigation : Clinical Investigation) :

Enterprise Manager
Public Operations:
createHealthcareProfessional (person_identification : PersonId, person_name : PersonName, profession_type : ProfessionType, signature : String) : Healthcare Professional
deleteHealthcareProfessional () :
createHealthcareOrganisation () :

deleteHealthcareOrganisation () :
getHealthcareProfessional () :
getProfessionalWithSignature () :
getHealthcareOrganisation () :
createConsultationService () :
createClinicalInvestigation () :
deleteSpecialistService () :
getSpecialistService () :
createReferralTemplate () :
createReferralAnswerReportTemplate () :
deleteReferralActTemplate () :
getReferralTemplate () :
getReferralAnswerReportTemplate () :
createHealthcarePartyRoleType () :
deleteHealtcarePartyRoleType () :
createPartyRoleAssignment (role_name : Healthcare Party Role Type, of_party : Healthcare Party, at_party : Healthcare Party, for_patient : Patient, start_time : Timepoint) :

Final Answer Report Template
Derived from Referral Answer Report Template
Public Attributes:
answer report template type : AnswerReportTemplateType

Healthcare Organisation
Derived from Healthcare Party
Public Attributes:
organisation name : String
organisation type : OrganisationType

Healthcare Party
Public Attributes:
registered ID number : KombikaCode
additional information : String
Public Operations:
createAddress (address_type : String, effective_from : Timepoint) : Address
createTelecomAddress (telecom_method : TelecomMethod, telecom_address_type : String, effective_from : Timepoint) : Telecom Address
createPartyAvailability (time_available : String, effective_from : Timepoint) : Party Availability
deleteAddress () :
deleteTelecomAddress () :
deletePartyAvailability () :
addSpecialistService (service : Specialist Service) :
removeSpecialistService (service : Specialist Service) :
addAssumableRoleType (assumable_role_type : Healthcare Party Role) :
removeAssumableRoleType (assumable_role_type : Healthcare Party Role) :

Healthcare Party Role
Public Attributes:
role name : String
role description : String

Healthcare Professional
Derived from Healthcare Party
Public Attributes:
person identification : PersonId

person name : PersonName
position : String
signature : String
profession type : ProfessionType

Public Operations:

addWorksFor (org : Healthcare Organisation) :
removeWorksFor (org : Healthcare Organisation) :
addProfessionalSkillType (skill_type : Healthcare Professional Skill Type) :
removeProfessionalSkillType (skill_type : Healthcare Professional Skill Type) :

Healthcare Professional Skill Type

Public Attributes:

skill type code : String
skill type name : String

Healthcare Team

Derived from Healthcare Organisation

Public Attributes:

type of team : String

Partial Answer Report Template

Derived from Referral Answer Report Template

Party Availability

Public Attributes:

ordinary time available : String
effective from : Timepoint
effective to : Timepoint
temporary restrictions : String
substitutor reference : String

Party Role Assignment

Public Attributes:

start time : Timepoint
end time : Timepoint
assignment description : String

Referral Act Template

Public Attributes:

template name : String
content type definition : String

Referral Answer Report Template

Derived from Referral Act Template

Referral Template

Derived from Referral Act Template

Specialist Service

Public Attributes:

service name : String

Public Operations:

addReferralAnswerTemplate (template : Referral Answer Report Template) :
removeReferralAnswerTemplate (template : Referral Answer Report Template) :

Specialist Service Type
Public Attributes:
service type name : String
service type description : String

Telecom Address
Public Attributes:
telecom method : TelecomMethod
telecom address type : String
effective from : Timepoint
effective to : Timepoint
telephone hours : String
telecom number : TelecomNumber
e-mail address : E-mailAddress
resource address : UniformResourceLocator
additional telecom address details : String

Answer ReportTemplateType
{TemplateOPCFinalAnswer, TemplateIPCFinalAnswer}

E-mailAddress

KombikaCode

OrganisationType
{Hospital, PrimaryHealthcareCenter, HomeCareUnit, SpecialistClinic, OtherOrgType}

PersonId

PersonName

ProfessionType
{Doctor, Nurse, AssistentNurse, Secretary, Curator, Dietician, MedicalGymnast, WorkTherapist, OtherProfessionType}

TelecomMethod
{Telephone, Telefax, E-mail, Network, OtherTelecomMethod}

TelecomNumber

Timepoint

UniformResourceLocator

7.2.3.2 Referrals

7.2.3.2.1 Referrals – Class Diagrams

Diagram: Referrals/Main

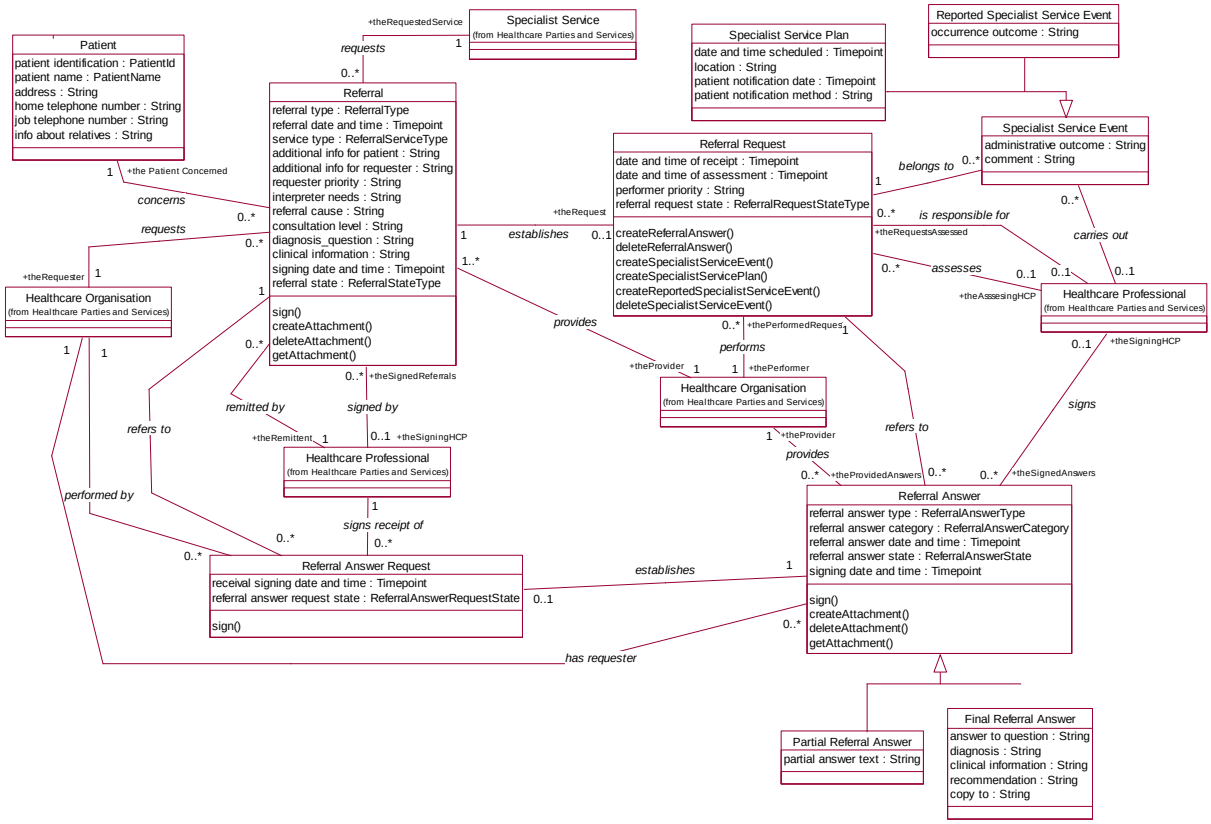


Diagram: Referrals/Attachment

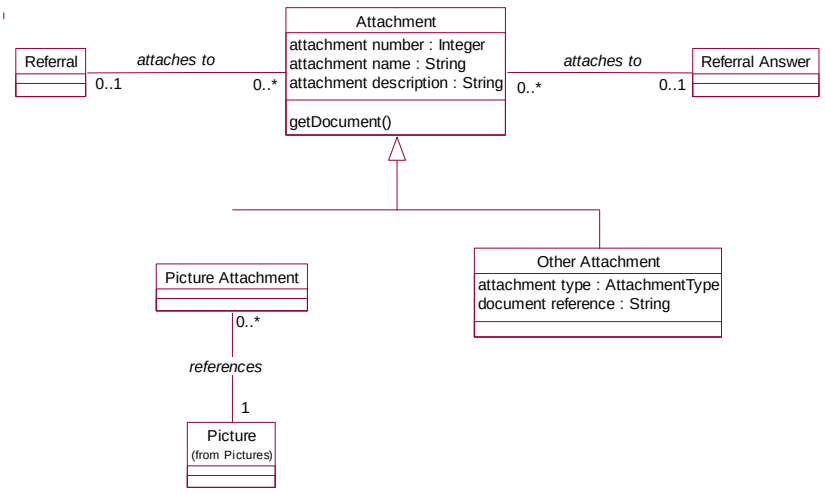


Diagram: Referrals/Referral Manager

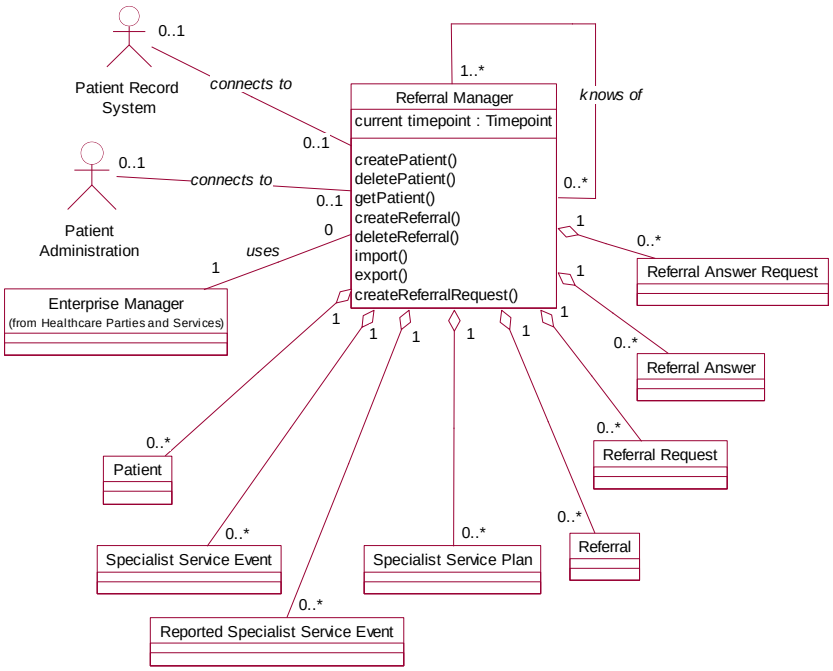
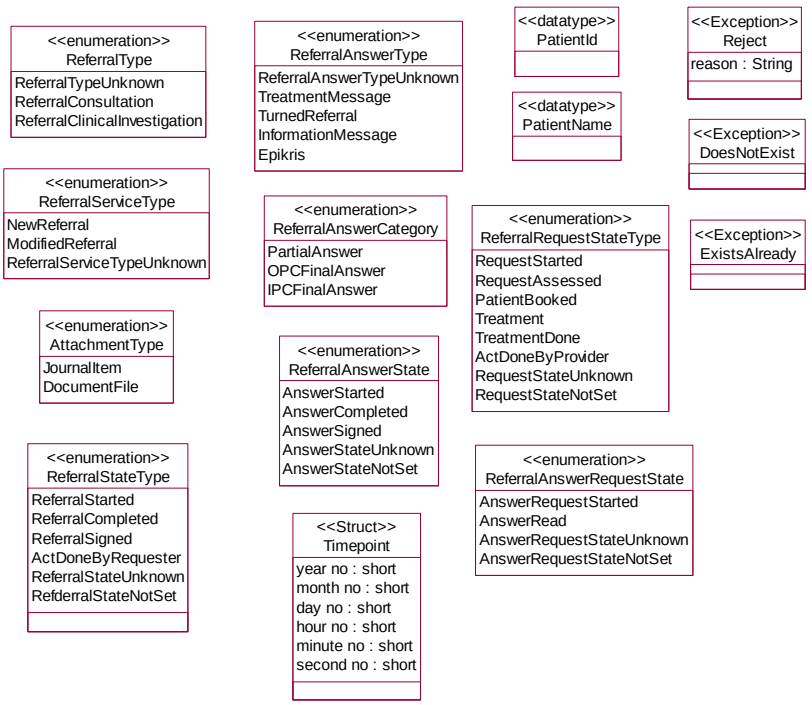


Diagram: Referrals/Data Types



7.2.3.2.2 Class Descriptions

Attachment
Public Attributes:
attachment number : Integer
attachment name : String
attachment description : String
Public Operations:
getDocument () : String

Final Referral Answer
Derived from Referral Answer
Public Attributes:
answer to question : String
diagnosis : String
clinical information : String
recommendation : String
copy to : String

Other Attachment
Derived from Attachment
Public Attributes:
attachment type : AttachmentType
document reference : String

Partial Referral Answer
Derived from Referral Answer
Public Attributes:
partial answer text : String

Patient
Public Attributes:
patient identification : PatientId
patient name : PatientName
address : String
home telephone number : String
job telephone number : String
info about relatives : String

Picture Attachment
Derived from Attachment

Referral
Public Attributes:
referral type : ReferralType
referral date and time : Timepoint
service type : ReferralServiceType
additional info for patient : String
additional info for requester : String
requester priority : String
interpreter needs : String
referral cause : String
consultation level : String
diagnosis_question : String
clinical information : String

signing date and time : Timepoint {frozen}
referral state : ReferralStateType
Public Operations:
sign (signingDateAndTime : Timepoint, signingHCP : Healthcare Professional) : Referral Request
createAttachment (attachmentNumber : Integer, attachmentType : AttachmentType) : Attachment
deleteAttachment (attachm : Attachment) :
getAttachment (attachmentNumber : Integer) : Attachment

Referral Answer
Public Attributes:
referral answer type : ReferralAnswerType
referral answer category : ReferralAnswerCategory
referral answer date and time : Timepoint
referral answer state : ReferralAnswerState
signing date and time : Timepoint
Public Operations:
sign (signing date : Date, signed by : Healthcare Professional) : Referral Answer Request
createAttachment (attachment number : Integer, attachment type : AttachmentType) : Attachment
deleteAttachment (suppl : Attachment) :
getAttachment (attachment number : Integer) : Attachment

Referral Answer Request
Public Attributes:
receival signing date and time : Timepoint
referral answer request state : ReferralAnswerRequestState
Public Operations:
sign (receiptSigningDate : Date, signedBy : Healthcare Professional) :

Referral Manager
Public Attributes:
current timepoint : Timepoint
Public Operations:
createPatient (patient_identification : PatientId, patient_name : PatientName) : Patient
deletePatient (pat : Patient) :
getPatient (patient_identification : PatientId) : Patient
createReferral (concerns : Patient, remitted_by-professional : Specialist Service Requester, requesting_organisation : Specialist Service Requester, providing_organisation : Specialist Service Provider, requests : Specialist Service, provider_RM : Referral Manager) : Referral
deleteReferral (ref : Referral) :
import () :
export () :
createReferralRequest (performing_organisation : Specialist Service Provider, date_of_receipt : Date, refer : Referral) : Referral Request

Referral Request
Public Attributes:
date and time of receipt : Timepoint
date and time of assessment : Timepoint
performer priority : String
referral request state : ReferralRequestStateType
Public Operations:
createReferralAnswer (category : ReferralAnswerCategory) : Referral Answer

deleteReferralAnswer () :
createSpecialistServiceEvent () : Specialist Service Event
createSpecialistServicePlan () : Specialist Service Plan
createReportedSpecialistServiceEvent () : Reported Specialist Service Event
deleteSpecialistServiceEvent (event : Specialist Service Event) :

Reported Specialist Service Event
Derived from Specialist Service Event
Public Attributes:
occurrence outcome : String

Specialist Service Event
Public Attributes:
administrative outcome : String
comment : String

Specialist Service Plan
Derived from Specialist Service Event
Public Attributes:
date and time scheduled : Timepoint
location : String
patient notification date : Timepoint
patient notification method : String

AttachmentType
{JournalItem, DocumentFile}

PatientId

PatientName

ReferralAnswerCategory
{PartialAnswer, OPCFinalAnswer, IPCFinalAnswer}

ReferralAnswerRequestState
{AnswerRequestStarted, AnswerRead, AnswerRequestStateUnknown, AnswerRequestStateNotSet}

ReferralAnswerState
{AnswerStarted, AnswerCompleted, AnswerSigned, AnswerStateUnknown, AnswerStateNotSet}

ReferralAnswerType
{ReferralAnswerTypeUnknown, TreatmentMessage, TurnedReferral, InformationMessage, Epicrisis}

ReferralRequestStateType
{RequestStarted, RequestAssessed, PatientBooked, Treatment, TreatmentDone, ActDoneByProvider, RequestStateUnknown, RequestStateNotSet}

ReferralServiceType
{NewReferral, ModifiedReferral, ReferralServiceTypeUnknown}

ReferralStateType

{ReferralStarted, ReferralCompleted, ReferralSigned, ActDoneByRequester, ReferralStateUnknown, RefderralStateNotSet}

ReferralType
{ReferralTypeUnknown, ReferralConsultation, ReferralClinicalInvestigation}

7.2.3.3 Pictures

7.2.3.3.1 Pictures – Class Diagrams

Diagram: Pictures/Main

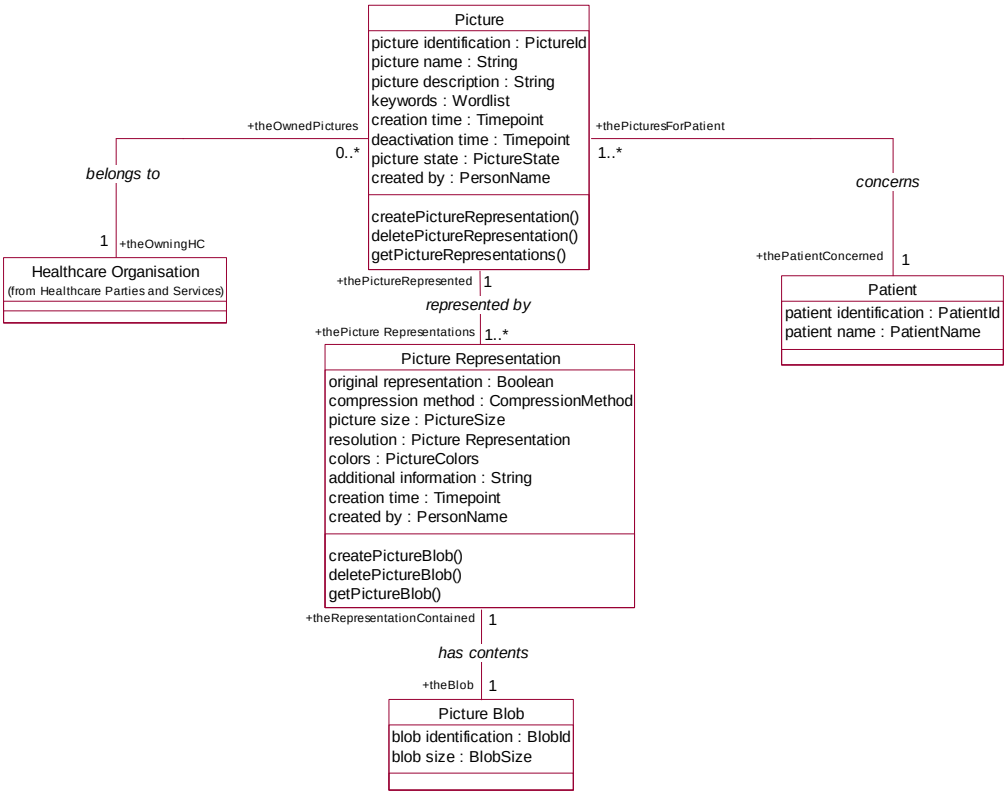


Diagram: Pictures/Picture Manager

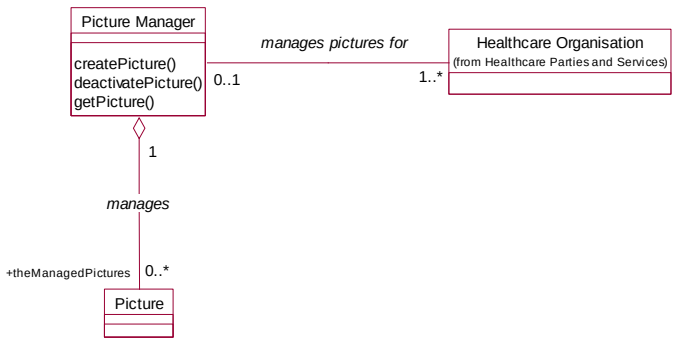
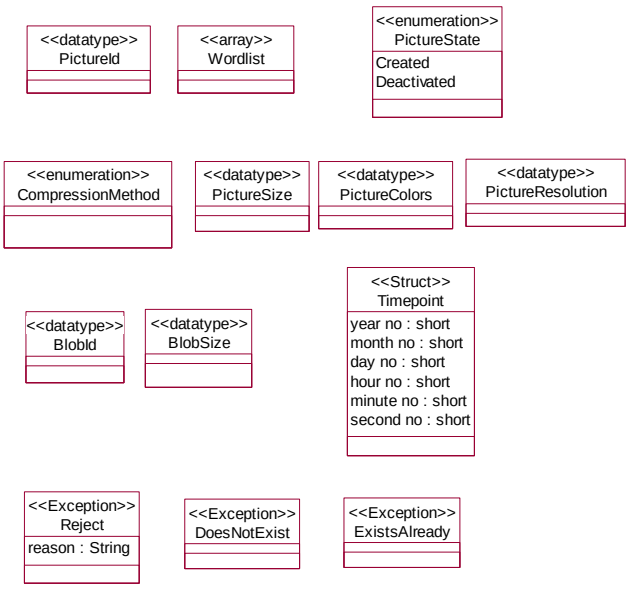


Diagram: Pictures/Picture Datatypes



7.2.3.3.2 Class Descriptions

Picture
Public Attributes:
picture identification : PictureId {frozen}
picture name : String
picture description : String
keywords : Wordlist
creation time : Timepoint {frozen}
deactivation time : Timepoint
picture state : PictureState
created by : PersonName
Public Operations:
createPictureRepresentation (original_representation : Boolean) : Picture Representation
deletePictureRepresentation () :
getPictureRepresentations () :

Picture Blob
Public Attributes:
blob identification : BlobId {frozen}
blob size : BlobSize

Picture Manager
Public Operations:
createPicture (picture_identification : PictureId, patient_concerned : Patient, owning_hco : Healthcare Organisation) :
deactivatePicture (pict : Picture) :
getPicture (picture_identification : PictureId) :

Picture Representation
Public Attributes:

original representation : Boolean {frozen}
compression method : CompressionMethod {frozen}
picture size : PictureSize {frozen}
resolution : Picture Representation {frozen}
colors : PictureColors
additional information : String
creation time : Timepoint {frozen}
created by : PersonName

Public Operations:

createPictureBlob () :
deletePictureBlob () :
getPictureBlob () :

BlobId

BlobSize

CompressionMethod

PictureColors

PictureId

PictureResolution

PictureSize

PictureState

{Created, Deactivated}

Timepoint

Public Attributes:

year no : short
month no : short
day no : short
hour no : short
minute no : short
second no : short

Wordlist

DoesNotExist

ExistsAlready

Reject

Public Attributes:

reason : String

7.2.4 Implementation Plan

The implementation of InterCare services in Stockholm South demonstration site will be made in an iterative way during 1999. This approach has been successful in STAR project and allows end-users to continuously refine their requirements on the information system.

Having specified the user requirements by using UML we have a good basis for the application and server development.

By agreeing on a common development platform and a common architecture in InterCare the foundation for parallel implementation has been put in place. In STAR Referral we have developed the basic functionality needed to create, forward, monitor and respond to a referral. STAR Referral has also the capability to interface existing legacy systems.

The object model has now been extended also to make it possible to attach references to multimedia objects. This extension of the application will be carried out during spring 1999.

This requires an elaborated security solution to secure the control of access to e.g. image databases. The design of the security solutions will be based on specifications from TrustHealth.

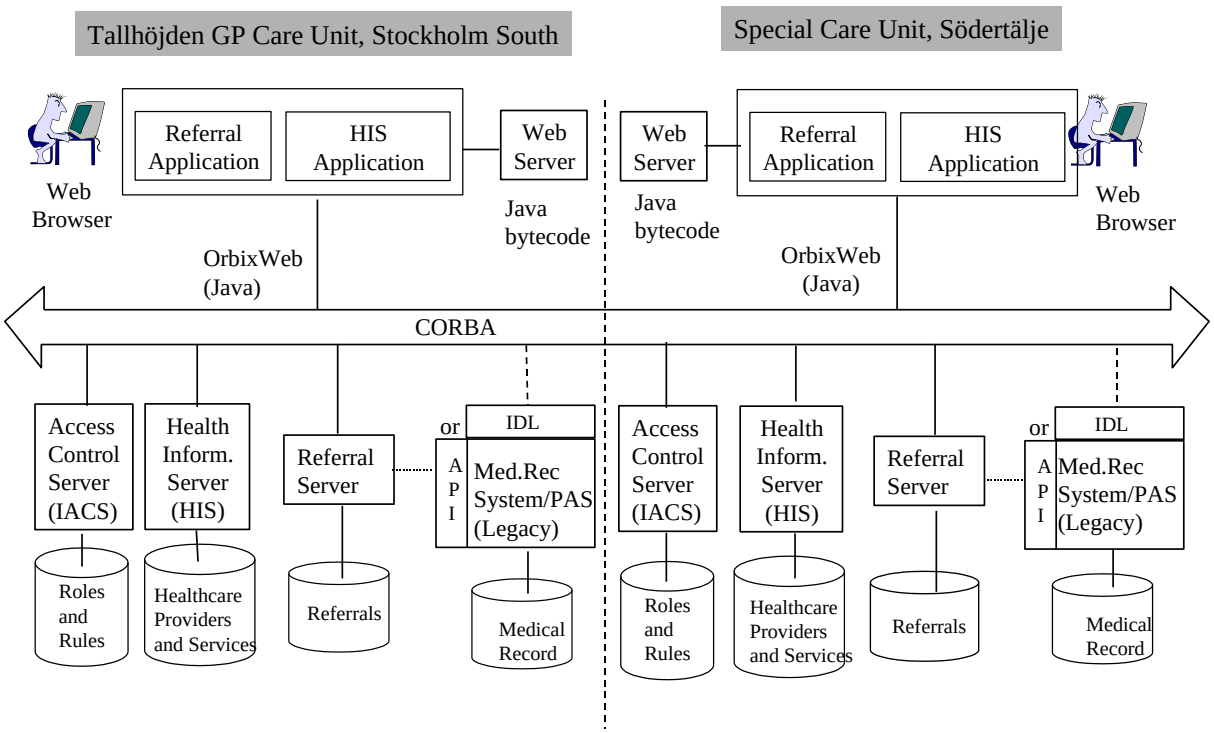
Finally the functionality to search provider information will be extended either as an act initiated through the referral process or as a "Yellow Pages" activity. The applications supporting this will be developed on top of InterCare HIS component during first half of 1999. The InterCare HIS component could be either an extended Enterprise Manager now supporting the STAR Referral application or a HIS-component developed by Saphis. This will be decided after the integration workshop planned to be carried out in the end of January.

A considerable amount of work also has to be done to integrate the HIS-component and applications to existing services, initiated both on national and local level, offering provider information.

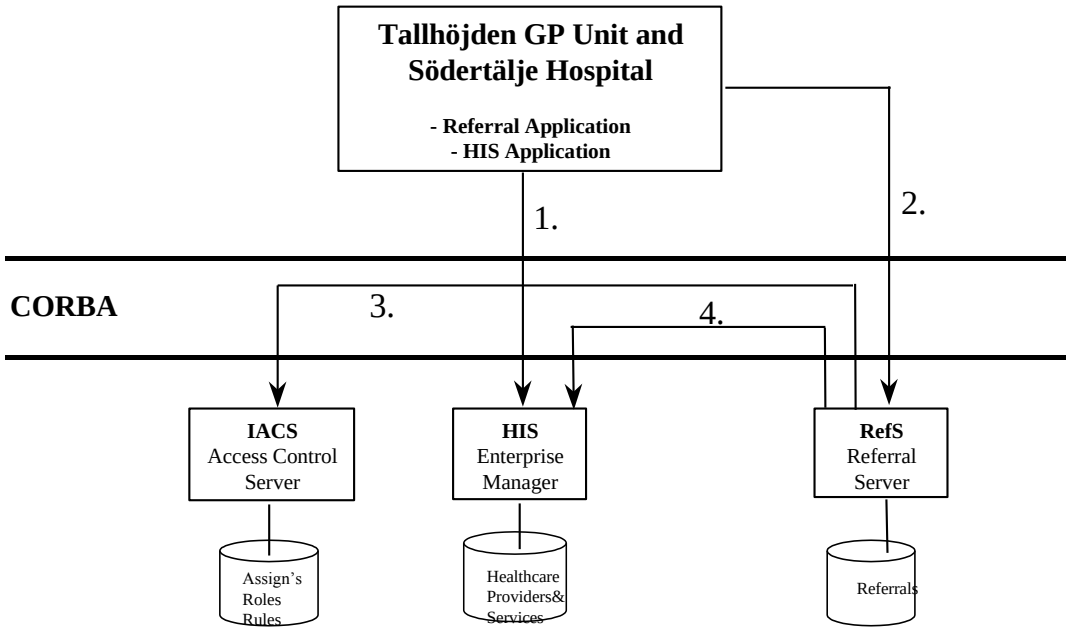
Due to plans the first versions of the extended referral systems should be verified in the beginning of the second year of the project.

7.2.5 Architectural Aspects

The target architecture is described in the following figure:

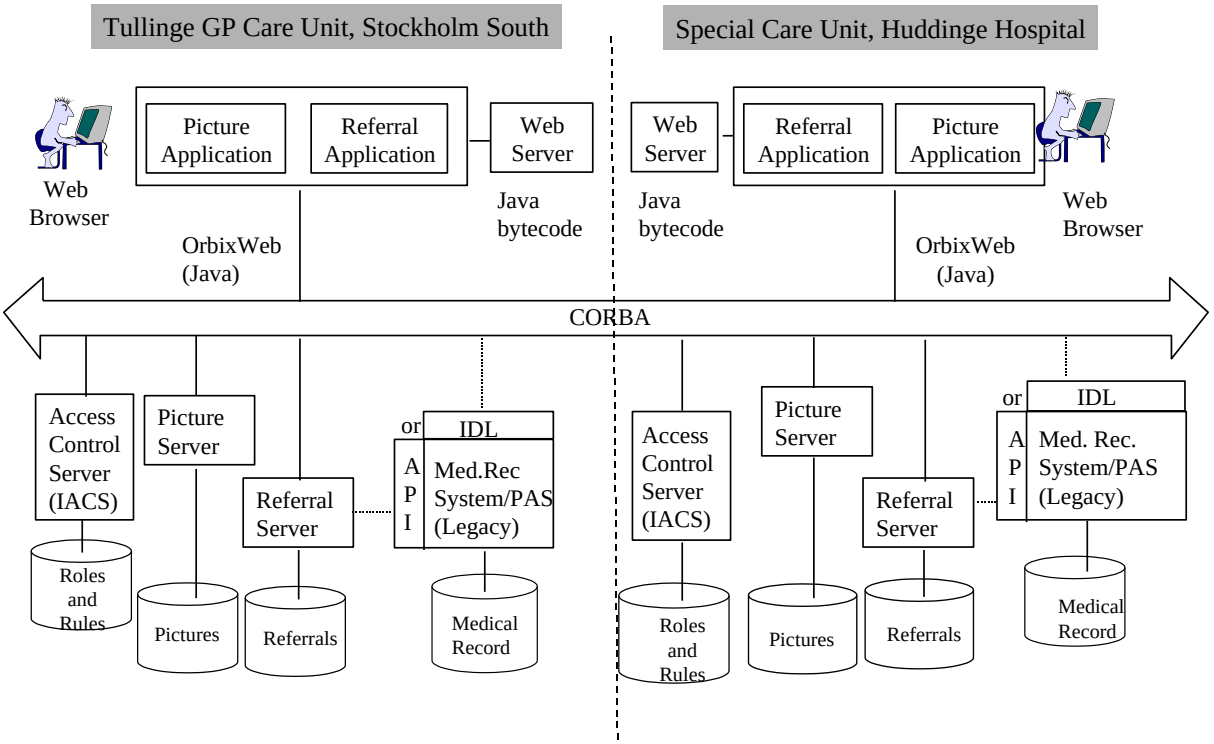


Target architecture of the Stockholm South application at Tallhöjden/Södertälje

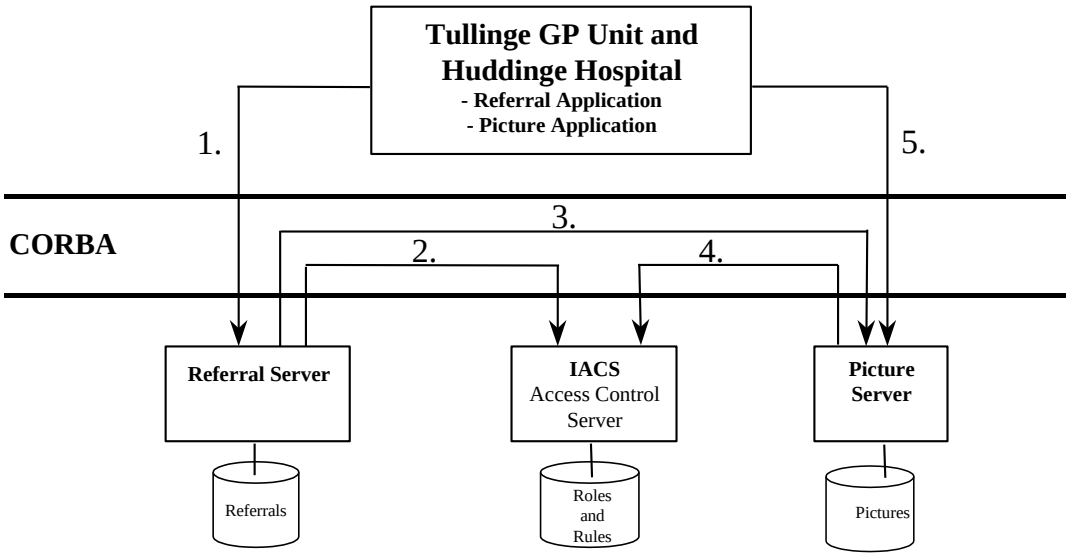


Component interoperability

1. Query concerning healthcare information.
2. Query concerning healthcare service requests (referrals). Result (answer to query or message ”permission denied”) returned after access control.
3. Request for access control. Permitted/not permitted returned.
4. Query concerning service requests (referral template). Access control HIS/IACS ev. delegated to RefS.



Target architecture of the Stockholm South application at Tullinge/Huddinge



Component interoperability

1. Query concerning healthcare service requests (referrals). Result (answer to query or message ”permission denied”) returned after access control.
2. Request for access control. Permitted/not permitted returned.
3. Query concerning picture attachment. Result (answer to query or message ”permission denied”) returned after access control.
4. Request for access control. Permitted/not permitted returned.
5. Query concerning picture attachment /picture. Result (answer to query or message ”permission denied”) returned after access control.