



Requirements and design – Operational execution deliverable D2.2

Frank van der Linden

6-5-2011

Final

Contributors

Frank van der Linden
Frenk Sloff
Peter Eshuis
Jan-Marc Verlinden
Roger Erens
Christoph Stettina
Luuk Groenewegen
Tom Sutedja
Huub Rutten
Elsemie ten Pas

affiliation

Philips
Philips
Philips
ZorgGemak
ZorgGemak
Cetim
Cetim
VUMC
Sopheon
Sopheon

Table of contents

1	Introduction	1
2	Requirements	2
2.1	Improvements for the intervention planning	2
2.2	Intervention system requirements	2
2.3	Virtual EHR requirements	3
2.3.1	Introduction	3
2.3.2	Applying the standards to Edafmis	4
2.4	Decision support requirements	6
2.5	Protocol management requirements.....	7
3	Design & Interfaces.....	8
3.1	Process support	8
3.2	Intervention system	11
3.3	Virtual EHR.....	11
3.4	Decision Support System	15
3.5	Protocol management	18
4	Summary.....	21

1 Introduction

Frank van der Linden – Philips Healthcare

This document describes the requirements and the design of the demonstrators for operational execution. The demonstrators of Edafmis are presented in Figure 1. Different partners deliver different parts of the demonstrators, each described with an ellipse in Figure 1:

- Philips provides the EP navigator, intervention and navigation system, including the connection to the in-hospital medical image databases.
- ZorgGemak provides the Virtual data repository providing access to all kinds of relevant data for the intervention. In particular ZorgGemak delivers intelligent access to data – subject of WP1.
- Sopheon provides access to work flow and process data. Presentations of these (subject of WP1) will be delivered to the intervention system through the virtual data repository.
- Mobilera provides a mobile DICOM viewer – subject of WP3. It will access and deliver images through the virtual data repository.
- Some external sources of data are not part of the project. These are all kinds of sensor data of the patient (ECG, temperature, state of anaesthesia etc.). Others are long term storage of medical images and of medical knowledge. For access to these present day solutions will be used.

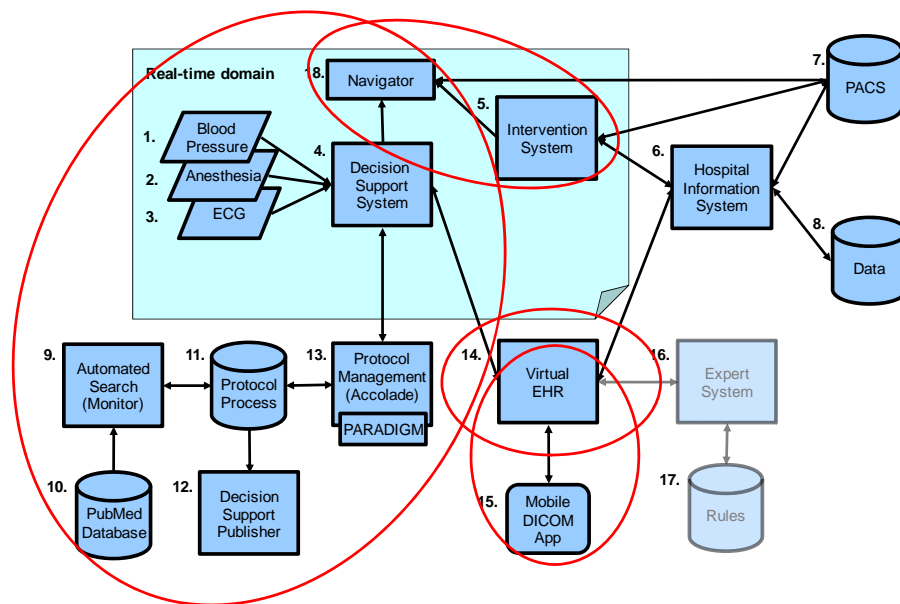


Figure 1: Edafmis - demonstrator set-up

This document is structured according to the demonstrator set-up.

First we get a part on requirements, section 2, and second there is a part, section 0, with designs.

The requirements from the end-user are provided in section 2.1. The requirements of the intervention system (5, 18) are described in section 2.2, for the virtual data repository (14) are described in section 2.3, those for decision support (4) in section 2.4 and those for the protocol management (13) in section 2.5. The requirements and design for the mobile DICOM viewer (15) are subject of WP3. The global design of the demonstrator (i.e. the design of Figure 1) is provided in section 3.1. The design and interfaces for the intervention system (5, 18) are described in section 3.2, for the virtual data repository (14) in section 3.3, for the decision support (4) in section 3.4, and for the protocol management (13) in section 3.5. A summary of the document is given in section 4.

2 Requirements

2.1 Improvements for the intervention planning

Tom Sutedja – VUMC

In this section the requirements for Interventional Pulmonology for lung cancer diagnosis is discussed. Nevertheless, the various procedures can also be applied for diagnostic purposes in other diseases than lung cancer but this is beyond the scope of current section. We envisage in the next program within EDAFMIS with regard to the lymph node navigational project of which the following algorithm can be developed:

1. Integral use of sharing data on a audio-visual platform using high resolution monitor by many specialists involved in the multidisciplinary setting in which each specialist should use unique identifier within secure virtual personal network. Each user should be clearly identifiable and concurrent tracking of user and in retrospect should be make possible.
2. Possibilities for audio-visual tele-feedback without the necessity of any specialist physically being present (Skype, Yahoo Messenger, I-Chat, VOIP) thru high resolution tablets such as I-PAD, Samsung Galaxy, or HD monitor in the room) allowing real time discussions for taking multidisciplinary strategic decision either among multiple platforms (i.e. two specialists or many specialists, many locations) such as in telemedicine, tele-conference with potential of multi institutional input and output.
3. Added information either data, or visual images according to each intervention will be blended/added to existing data archives, such that a sophisticated high quality augmented audio-textual-visual comprehensive data of each patient become cumulatively available. Archived data modules should be safely and rigidly guarded, uniquely linked and/or tagged to the patient data cumulatively stored in the archives. There should be tracking of data input, which data and by whom.
4. Archived data are secured, can only be used for viewing, export of data storage outside the central archives should be completely prohibited. Each user at any time should be identifiable. There will be central control from the respective institutions. Multidisciplinary use cannot be enabled without pertinent permission from the coordinator of the institution that has accumulated patients' data.

2.2 Intervention system requirements

Peter Eshuis – Philips

The intervention system plays the central role during an interventional procedure and the main requirement is to provide images of the region of interest at a sufficiently high quality (quality of the images can be tuned by the physician). For the treatment of Atrial Fibrillation an X-ray Intervention System is used to provide live X-ray images at for example 15 frames/second.

The X-ray Intervention System can be used in a CT-like mode shooting images from all sides of the patient, i.e. through a so-called rotational acquisition, and these images can be reconstructed in 3D by the Interventional Tools computer connected to the Intervention System. For Atrial Fibrillation a workstation with EP navigator software provides the 3D reconstruction of the Left Atrium and provides the 3D volume information as an overlay on top of the live X-ray images, see Figure 2 below. The doctor navigates through the heart based on the image (live+3D overlay) supplied by the EP navigator, so the following requirements are related to the EP navigator:

- The EP navigator must have real-time connection with the Intervention System to be able to show live X-ray images to the doctor.
- Optionally the EP navigator may be connected to a PACS to retrieve/store image data.
- The EP navigator is the main source of image data for the doctor to perform the procedure on, so this is the appropriate spot to provide decision support to the doctor during the complete procedure.

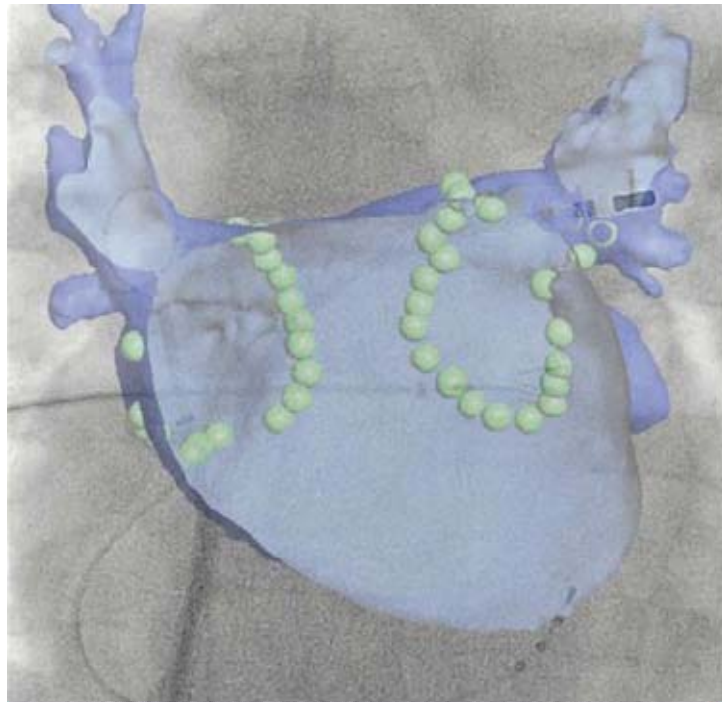


Figure 2: 3D image of left atrium overlaid with the EP navigator on live X-ray image (green spheres mark the catheter positions for ablation)

2.3 Virtual EHR requirements

Jan-Marc Verlinden & Roger Erens – ZorgGemak

2.3.1 Introduction

Standardization is an important way to foster interoperability among multi-vendor devices, solutions, and services making it a key condition for the growth of the market. A component that implements such a standard hence plays a key role in facilitating the Decision Support System.

The interoperability-areas of main interest for the Edafmis project are the following ones:

- A. Technical and semantic interoperability among the health-related devices and systems
- B. Wireless interoperability (this area will be covered in deliverable D3.1)
- C. Security and privacy

First, we recap the ISO/CEN EN13606 standard: with regard to the standardisation of Electronic Health Record (EHR) architecture, this standard describes all aspects of EHR in general, in a very theoretical way. It is structured into five levels whose evolution is described below.

1. **ISO 13606-1:2008** specifies the communication of part or all of the electronic health record (EHR) of a single identified subject of care among EHR systems, or between EHR systems and a centralized EHR data repository. It may also be used for EHR communication between an EHR system or repository and clinical applications or middleware components (such as decision support components) that need to access or provide EHR data, or as the representation of EHR data within a distributed (federated) record system. It has been recognised that the main applications of these principles can be achieved by using HL7-v3 messages (implemented in XML).

2. **ISO 13606-2:2008** specifies the information architecture required for interoperable communications among systems and services that need or provide EHR data. ISO 13606-2:2008 is not intended to specify the internal architecture or database design of such systems. ISO 13606-2:2008 defines an archetype model to be used to represent archetypes when exchanged between repositories or archetype services. The archetypes can be serialised using two different structuring methods:
 - either openEHR, or
 - Clinical Document Architecture (HL7-CDA R2).
3. **ISO 13606-3:2009** defines term lists, each specifying the set of values that particular attributes of the Reference Model defined in ISO 13606-1 may take. It also defines informative Reference Archetypes that correspond to ENTRY-level compound data structures within the Reference Models of openEHR and HL7 Version 3, to enable those instances to be represented within a consistent structure when communicated using ISO 13606-3:2009.
4. **ISO 13606-4:2009** describes a methodology for specifying the privileges necessary to access EHR data. It also refers to general security requirements that apply to EHR communications such as consent and auditability.
5. **ISO 13606-5:2010** ISO 13606-5:2010 defines a set of interfaces to request and provide EHR and archetype communications of the following artefacts:
 - An EHR_EXTRACT for a given subject of care as defined in ISO 13606-1;
 - One or more ARCHETYPE(s) as defined in ISO 13606-2;
 - An EHR_AUDIT_LOG_EXTRACT for a given subject of care as defined in ISO 13606-4 and for providing the data to the requesting party or for declining the request.ISO 13606-5:2010 effectively defines the payload to be communicated at each interface. It does not specify the particular information that different transport protocols will additionally require, the security nor authentication procedures that might be agreed between the communicating parties or required by different jurisdictions.

Furthermore, within the medical domain of imaging systems the interoperability between machines of different vendors is dominated by the major protocol DICOM (<http://medical.nema.org/>). DICOM is known as NEMA standard PS3, and as ISO standard 12052.

2.3.2 Applying the standards to Edafmis

From the preceding section, it follows that area A is covered by the first three levels of EN13606. This leads to the requirement to use HL7v3 XML-based messages. Only in case interaction is needed with legacy-based data sources, HL7v2 messages should be used.

Typically the content (payload) of those messages being passed around is only parts of the patients EHR: extracts. The extracts should be serialised either in parts of openEHR-based archetypes, or HL7v3-based CDA-R2 documents. In the upcoming version of the openEHR-standards, parts or collections of archetypes are called templates, but here we will stick to the current openEHR-standard that does not yet make this distinction between archetypes and templates.

The detailed structure and contents of the archetype(-parts) and CDA-R2 documents will be worked out in an iterative process using the requirements for component (4), **Decision Support System**, given in chapters 2.4 and 3.4.

Versions for the archetypes receiving dynamic patient data from component (4), **DSS**, are being developed in conjunction with the software. A preliminary list of information that needs to be persisted in the EHR are:

- patient ID
- timestamp
- vital signs

- resulting decision/action taken
- clinician ID
- DICOM-image ID

Area C from the preceding section is covered by the fourth level of EN13606. This gives rise to the following requirements:

- medical and demographic data should be kept separate
- the role of the clinician requesting patient data is used to check whether the clinician is authorised to access that data
- when access is granted, the requests and changes made to the patient data are logged to enable auditing

We note from Figure 1 that component (14), **Virtual EHR**, connects to four other components. As noted above, the interaction with component (15), **Mobile DICOM Application**, will be discussed in deliverable D3.1.

The Virtual EHR must present itself in the information ecosystem as an API which allows access to other repositories. In this way, a single API is known to system-developers, and hidden behind are many data providers of many kind.

Since the Virtual HER is outside of the real-time domain we expect the use of the SOAP-protocol for the messages to be sufficiently fast. In the real-time domain, web-sockets or legacy protocols may be needed.

Electronic Health Records (EHR)

Search this site

1. EHR and Medical Communication Standards

1. CDA

2. openEHR Introduction

Architecture overview

Reference model

ArcheType model

Orbology service

Heart Failure Ontology

Named browser

XML schemas

3. The messaging standard (XML, HL7v3)

2. OpenEHR API

ZorgGemak Implementation

Service Model

Data services

Composition Service

Message Service

EHR Service

Locatable Service

History Service

Directory Service

ACL Service

Utility services

Terminology Service

Measurement Service

Configuration Service

ArcheType Service

Transaction Service

System Service

Session Service

Security/Access Service

Logging Service

Documentation Service

Template Service

Archetypes

Discussions

Other information

Page authors

2. OpenEHR API >

ZorgGemak Implementation

Flexible

This OpenEHR webservice implementation has been designed with absolute flexibility and scalability in mind. Webservices are location-transparent, platform-independent and scalable. Thereby webservices can have all kinds of client-software, e.g. native platform-software, web-clients or other webservices. In this way webservices can be an end of the line product, or a lego-piece in an enterprise environment. Webservices should be considered an interface behind the actual software. If needed, it is possible to add more interfaces to the same service, or combine different-services for one service-interface.

Design goals

The implementation consists of a group of services which are (loosely) coupled. These kind of services only have a few hard dependencies. On the other side, the coupling must have well-defined interfaces. The loosely coupling is preferred, because the implementation must be able to run in yet unknown situations. For example, specific service functionality may already exist which should be used instead of the native openEHR-service. Or customers may prefer another storage-vendor or even another storage-type. This can be the case for example the authorization/authentication-service, terminology-service, storage service, etc. How this is done is explained in detail at the specific service-descriptions. Important is that the openEHR webservice implementation can collect all necessary services and contact them over well defined interfaces.

Location transparent

The location of the services is not defined. As long as they can be reached, the openEHR webservice implementation is able to use them. These services can easily be used on a standard PC or in a server-park, independent of the platform used.

Isolation of responsibilities, autonomy, stateless

All services have their own defined responsibility. They have a specific task and use specific resources. They can be deployed, modified and maintained independent from each other. The service operations are stateless. They do not remember the last action, and do not know the next. They do not depend on the state of other services. Stateless services provide better performance, flexibility, scale ability and reliability, and also reduces complexity. A few parameters do depend on the state of other services, these are elaborated in the session context.

Interchangeable

Although a part of the implementation services are interchangeable, this implementation consists of a complete and fully functional group of services it is in no way needed to have already in house supporting services of any kind. This implementation

Figure 3: API specifications for openEHR standards

2.4 Decision support requirements

Huub Rutten & Elsemiek ten Pas – Sopheon

The goal of the Decision Support System (DSS) is to provide real-time process support to health professionals during medical interventions. More in particular, it should warn health professionals about exceeded limits or potentially dangerous situations, so that they can concentrate on the treatment itself. To achieve this, the DSS needs to integrate dynamic patient data coming from intra-operative imaging and monitoring devices with existing patient data and existing medical protocols.

In Figure 4 one sees that for the decision support we need a predefined Care Pathway – Clinical Pathway, describing the procedure for a specific consultation. Such a Care Pathway is one of many in a hospital Protocol Management System and such a Care Pathway uses in fact many related protocols, such as laboratory protocols, general operating room protocols, nursing protocols etc. The protocols include the rules for the DSS engine.

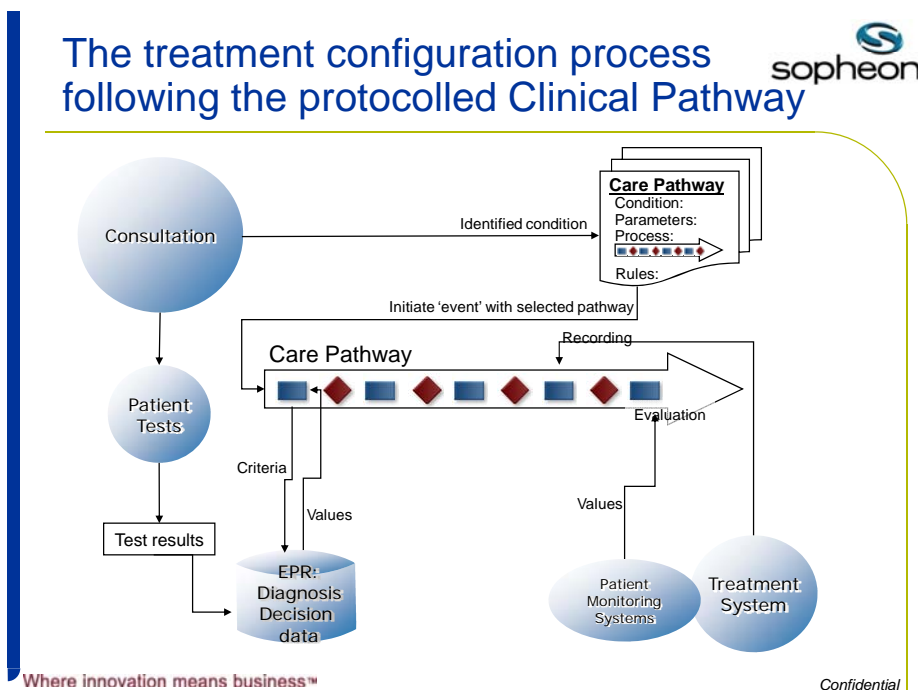


Figure 4: Clinical Pathway

At the centre of the DSS are the rules it uses for the calculations: values from the different data sources need to be calculated real-time, in parallel and in a cumulative manner, following algorithms and rules as officially described and approved in a protocol. In other words, the DSS requires that the protocol includes the rules to be executed by the DSS. These rules should specify under which conditions actions are to be performed, by whom or which device and within which limits or constraints.

In order to allow for real-time integration of different data sources, the DSS should:

- Be flexible, usable for any typical clinical operation, including “phased” processes (research concepts from Paradigm)
- Be configurable for the doctors through formalized protocol management – under their control
- Work with existing data and systems – no double or extra data!
- Be transparent about how it works and worked (alert history)

- Work as an “embedded system”, as a “smart app”
- Work on many devices – real-time calculation in the browser
- Calculate fast -- multiple times per second
- Be easy to install and maintain
- Be easy to use

This means the DSS requires the following APIs:

1. Instant and continuous access to the monitoring devices providing vital sign data during the intervention, such as temperature, blood pressure, sweating, iris movement, etc.
2. Instant and continuous access to the 3D workstation from Philips providing intra-operative imaging data
3. Access to the EHR system to load existing patient data (HL 7 based, or other)
4. Access to the protocol management system to load existing executable protocol data for its configuration.

In order to be functional for health professionals in the intervention room, the DSS should produce alerts that are:

1. instantly visible, preferably on the monitor of the intervention system used
2. simple to understand
3. readable – it should be clear why was the alert produced.

2.5 Protocol management requirements

Huub Rutten & Elsemiek ten Pas – Sopheon

The protocol management requires a protocol management system and a reading robot to keep track of new evidence to be incorporated in the protocols.

The protocol management system should:

1. be able to manage protocols that include rules that can be executed by the real-time DSS
2. provide capabilities enabling health professionals to edit and read the rules that can be executed by the DSS
3. be able to manage thousands of related protocols for an entire hospital for many diseases in one integrated database
4. be able to provide health professionals with instant access to the protocol at the point of care
5. be able to provide access to protocols at the point of care in an appropriate, instructive manner
6. comply with legal requirements and provide for each medical protocol to be reviewed and officially approved by someone who is not the author of the protocol
7. comply with liability requirements and provide for version management.

The reading robot should:

1. be able to generate automated alerts on new publications from an authoritative source (PubMed) that might contain relevant new evidence for the protocol

3 Design & Interfaces

3.1 Process support

Christoph Stettina & Luuk Groenewegen – CeTIM & Frank van der Linden – Philips

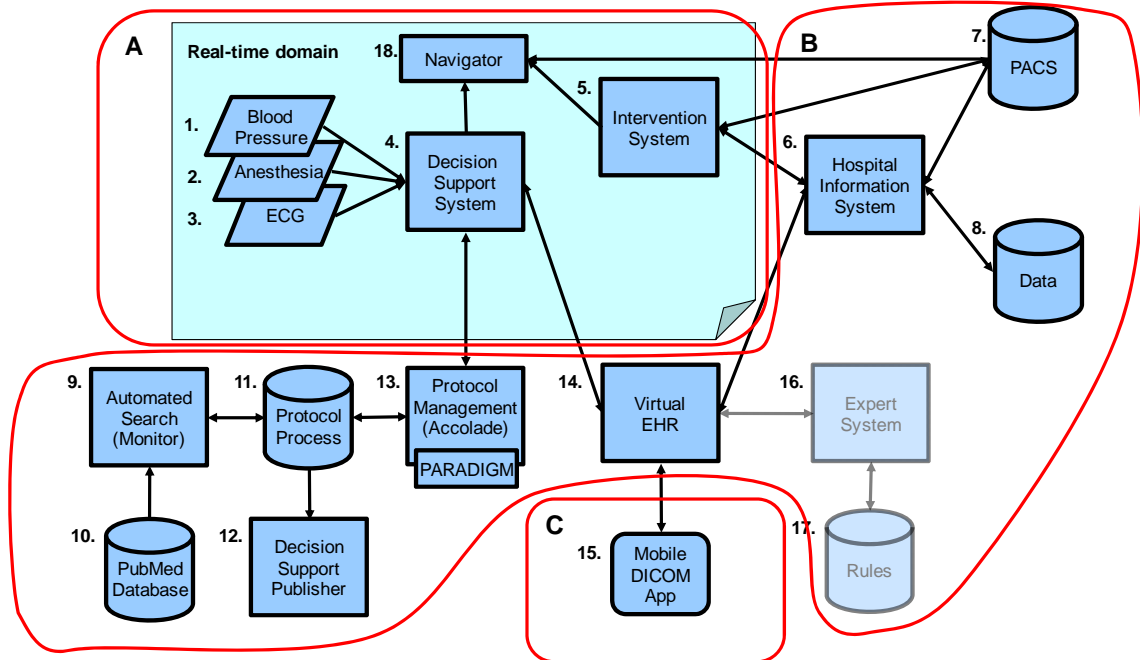


Figure 5: Edafmis - demonstrator set-up

Process support is crucial for minimal invasive intervention. Several systems have to work together to help the medical professionals in their work. Figure 5, shows the set-up of the process support

Operational Domains supporting the Intervention Process

The Edafmis process support is divided into three major operational domains with respective timing constraints:

- A. The real-time sensitive components where access is crucial during the intervention.
- B. The components outside the real-time domain, supporting the intervention with information less sensitive to timing constraints (e.g. patient images taken before the intervention can be transferred into the operating room in a pre-interventional step).
- C. The remote domain, where external collaboration is addressed.

Within the real-time domain the intervention system (5) is the most important component of the intervention. It provides images of the patient during the operation, and is connected to existing hospital information systems (6, 8) and PACS systems (7). Connected to the intervention system is a navigation system (18), which specifically supports the navigation control of instruments in the patient's body. The intervention and navigator system are described in section 3.2. The navigator is connected to a decision support system (4), described in section 3.4, that monitors other equipment in the intervention room (1, 2, 3). This equipment looks at vital signs of the patient. For Edafmis we use existing equipment, like blood pressure meters, anaesthesia monitors and ECGs. Whenever the decision support system decides that these signs indicate an alert situation, this alert is communicated to the navigation system that in turn informs the surgeon about the alert situation, as other, less standard actions might be necessary.

Outside the real-time domain, several systems are available to assist the clinician. The Virtual data repository (14) provides access to all kinds of relevant medical data relevant for the intervention. It can access the medical history of the patient at hand and it can access medical and surgery information on patients with similar diseases. An expert system (16) with rules (17) will help to provide intelligent access to this data. This is described in section 3.3. The protocol management systems (13) provide the real-time decision support system (4) with the right information about protocols based on medical evidence and followed during the procedure. The protocol management systems access a data base with protocol process information (11). The data base is connected to a decision support publisher (12) used by the hospital to keep their protocols updated. Whenever relevant the protocol process database can access external (PubMed) data for updates, via a monitor (9, 10) data for updates, via a monitor. This is described in section 3.5.

Finally to facilitate external consultation, a mobile DICOM viewer (15) is used to communicate with external experts. If relevant the intervention system informs the hospital information system that a certain set of images is to be consulted by a given external expert. This expert is informed via the virtual EHR, and will access the data outside real-time domain via the hospital information system and the PACS server. This is mainly work in WP3.

Interplay of the Systems to Support the Intervention Process

The interplay of professionals and systems is crucial for the intervention. Without a proper integration the potential gains cannot be realized and reduction of costs and staff productivity cannot be achieved, although improved navigation and effective decision support are targeted to this. The drawing below is an example of the VT ablation intervention to help the reader to clarify the connections between the ICT systems and the medical world. To visualize the operation of the system the diagram, Figure 6, depicts an overview of the process involving the technology layer (hardware), applications (software), and the process (inside the intervention room) down to human actors.

Ventricular tachycardia is a cardiac dysrhythmia, a fast heart rhythm caused in one of the ventricles of the heart. It is potentially life-threatening as it can lead to ventricular fibrillation, asystole, and sudden death. Catheter ablation, the removal of biological tissue, is a key therapy for patients with recurrent VT and can be an alternative for patients with an implantable cardioverter defibrillator.

Intervention Room Roles. At the level of human actors we can see 3 roles assigned to the VT ablation process: the patient, the assigned clinician and a nurse.

Intervention Room Processes. The actual intervention is divided into a pre- and intra-operative phase consisting of several sub-phases and actions. In the preparation phase the patient data is being studied by the clinician and the medical staff, followed by an examination of the patient on-site and a preparation of the intervention room and systems before the intervention. During the intra-operative phase the doctor probes several locations inside the heart chamber in order to provoke a controlled reaction. Then he performs the actual VT ablation by burning heart tissue at the location where the source of the heart rhythm disorder is. Finally he verifies the success of the intervention by probing all relevant locations of the former disorder again.

Application Layer (Software). This layer provides the intelligent systems supporting the intervention with an emphasis on the real-time domain. The intervention system creates live x-ray images of the patient during the medical operation and provides them to the EP-Navigator. While the original intervention can take up to 3-4 hours with fluoroscopy, 3D imaging techniques allow faster and more accurate interventions due to the improved spatial orientation. The decision support system of Edafmis furthermore allows monitoring of other equipment in the intervention room such as blood

pressure meters, anaesthesia monitors and ECGs according to medical protocols. Whenever the DSS encounters exceptional situations a system alert is communicated through the navigation system to the doctor. He then can decide on further actions.

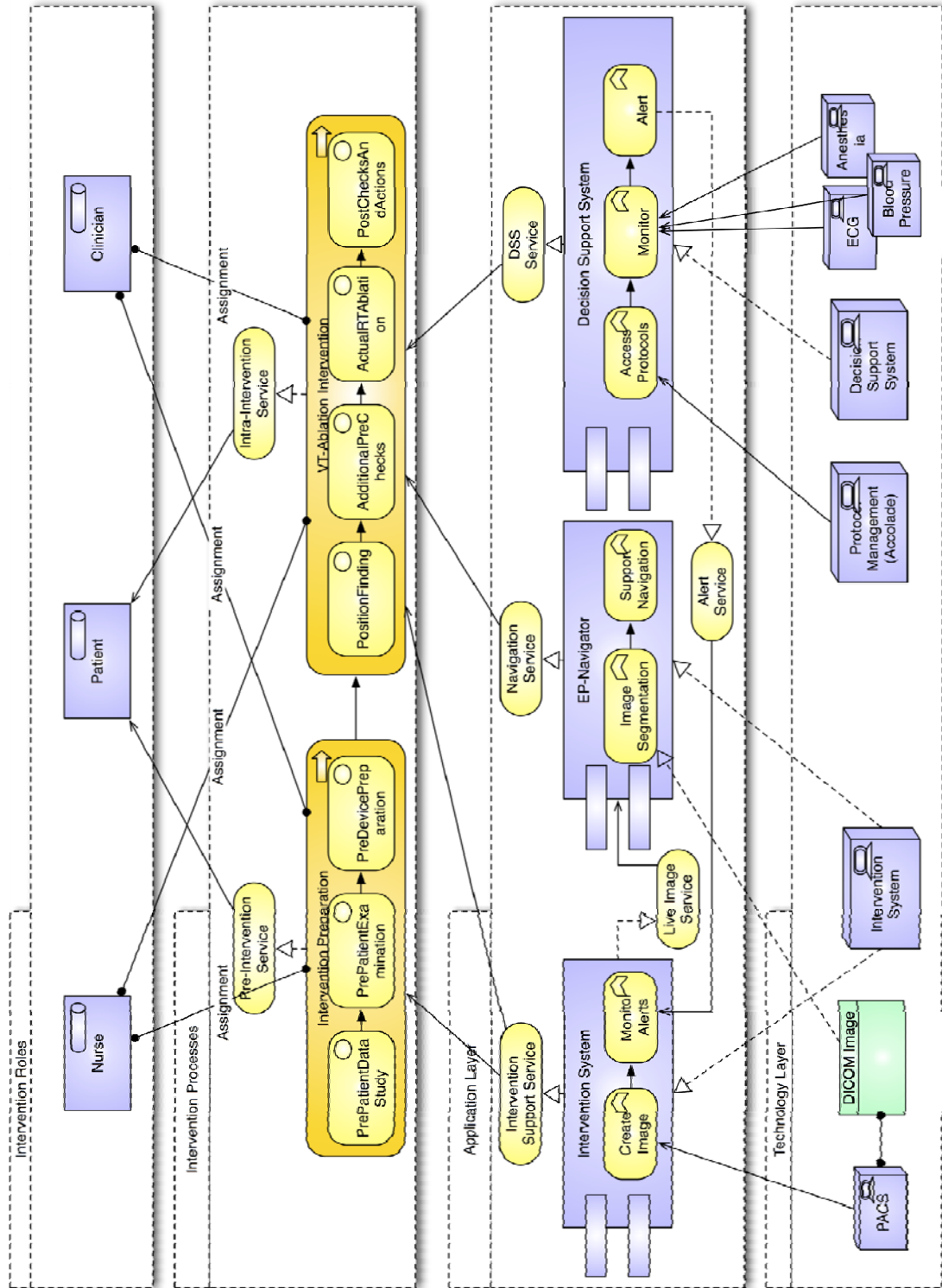


Figure 6: Intervention Process

Technology Layer (Hardware). At the hardware layer we can find the Philips intervention system hosting the intervention software as well as the EP-Navigator software. The PACS provides DICOM images from the patient. The Sopheon system hosts the DSS application and provides protocol management functionality. The sensors in the operating room such as ECGs, blood pressure meters and anaesthesia monitors then provide direct input to the DSS.

3.2 Intervention system

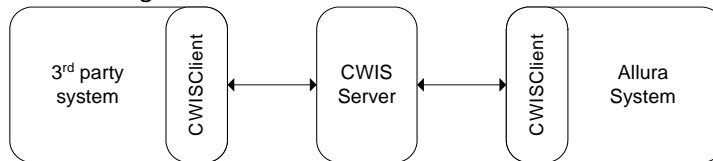
Peter Eshuis – Philips

The X-ray Intervention System and the EP navigator computer are connected through the hospital network over which metadata is broadcasted by the Intervention System. In this way the EP navigator computer picks up the Patient Name and Patient ID from the Intervention System for example.

The live X-ray images are broadcasted by the Intervention System over a glass fibre network to which the EP navigator computer is connected to.

To publish the messages/events produced by the decision support system on the EP navigator computer the Cathlab Workstation Integration Service (CWIS) interface is used. CWIS is a Philips proprietary protocol for exchanging data between two systems in the CathLab. CWIS is specially designed for integrating third party systems and the Philips X-ray System.

CWIS is designed as a client-server architecture:



The 3rd party system and the Allura X-ray System are using the CWIS Client to interface with the CWIS data exchange mechanism. The CWIS Client hides implementation details for the CWIS users and provides a small and clear interface. This CWIS Client is a ready to use component supplied by Philips. The CWIS Server is the central 'post office', i.e., it will facilitate in the data exchange.

CWIS will detect a disconnection of a function due to e.g. physical problems (broken cable). By the graceful degradation strategy implemented by CWIS it will not lead to unexpected system behaviour. CWIS will make sure the disconnected function gets informed and the user will get feedback that the function is disconnected. When the connection is restored CWIS is able to start communicating again. This means that a system restart is not necessary to restore the connection.

3.3 Virtual EHR

Jan-Marc Verlinden & Roger Erens – ZorgGemak

In order to meet the requirements as mentioned in Section 2.3, we propose the introduction of the Virtual Electronic Health Record (Virtual EHR). This term will be used to refer to the concept of a record that integrates patient record information of various information sources but is presented to the user as a single patient record. It is shown as component (14) in Figure 1 and Figure 5.

In principle, the patient data in the Virtual EHR is not persistent. It only has all the relevant patient data in memory when the application is running. The Virtual EHR obtains that static data from external sources (sometimes also called Distributed Data Repositories, DDRs) and presents them in an integrated way.

The Virtual EHR can also collect the real-time patient data streams (e.g. the blood pressure, anaesthesia concentrations, ECG). Those data are fused with the pre-intervention data that were collected in the HIS (or a HIS-like) application. The Virtual EHR uses a patient model that is archetype-based. Interfacing is not restricted to archetype-based messages: with the use of plug-ins, also messages can be exchanged that are based on e.g. HL7v2, HL7v3, CEN13606, or DCMs.

The Virtual EHR is based on ZorgGemak's OpenEHR-environment which has advantages. Because all software layers are transparent, the changes needed to be written (or configured) do not touch the system on deep level, this avoids risks of disturbing the system stability.

The generic kernel contains only generic knowledge of medical data. All specific knowledge is in the archetypes. The ZorgGemak OpenEHR-kernel is based on openEHR and has functionality like:

- It receives a dataset and verifies that the associated archetype already is in the kernel. If so, then the kernel verifies the dataset to the requirements formulated in the archetypes. If the validation succeeds then the data will be stored. If the archetype is not found, or when the validation fails, an exception will occur and the data will not be saved.
- The kernel provides an interface for data search. In fact, AQL (Archetype Query Language, similar to SQL) queries are to be used to find and retrieve conditional data in the kernel.
- The kernel distinguishes demographical and medical data. That is the only substantive knowledge of data that is in the kernel. The medical and demographic data are separated, enabling them to be stored on different servers and data warehouses.
- The kernel maintains a security access mechanism to a dense distribution of tasks and groups.
- The kernel maintains a directory structure in which virtual medical data can be grouped. A medical data-set can exist in multiple folders simultaneously. Subfolders can also be created. The kernel provides an API for linking or unlinking medical data to folders which can be created or deleted.
- The kernel includes a time machine. This database is capable to present data-constellations at any time in the past. It is possible to restore the complete database to a situation it was in at any moment in the past. The purpose is use for historic medical research (e.g. which decisions were taken on a specific medical situation, what were the consequences off a specific decision), medical research (e.g. which were the conditions on which a specific medication was issued) and legal issues (e.g. did a medical professional take a right decision based on information available at that moment).
- The kernel does not remove anything: modified data is stored in versions. Items removed within the kernel are only marked as deleted.

All these functionalities are accessible through various (levels of) APIs. This is depicted in Figure 7.

OpenEHR has several API-layers, native generic (yellow API) and added (orange) customized API's. The OpenEHR kernel-functionality connects to all API's (using some library-layers in between).

External data-provider can be used to connect to the kernel, for example a demographic service, or LDAP-service. To allow kernel interaction, an interaction layer needs to be on top of the Distributed Data Repository (DDR). This can be very generic on archetype-level, but can also be more specific. A DDR can also be added to the system on service layer, where it will be a data-provider for the native OpenEHR API, but passing the OpenEHR kernel.

The OpenEHR kernel (yellowish white) can connect to its native API, but also to the customized APIs; likewise all other components, document-based, DDRs, raw-databases can connect to the native API, a customized API, or both, and via the kernel or passing the kernel.

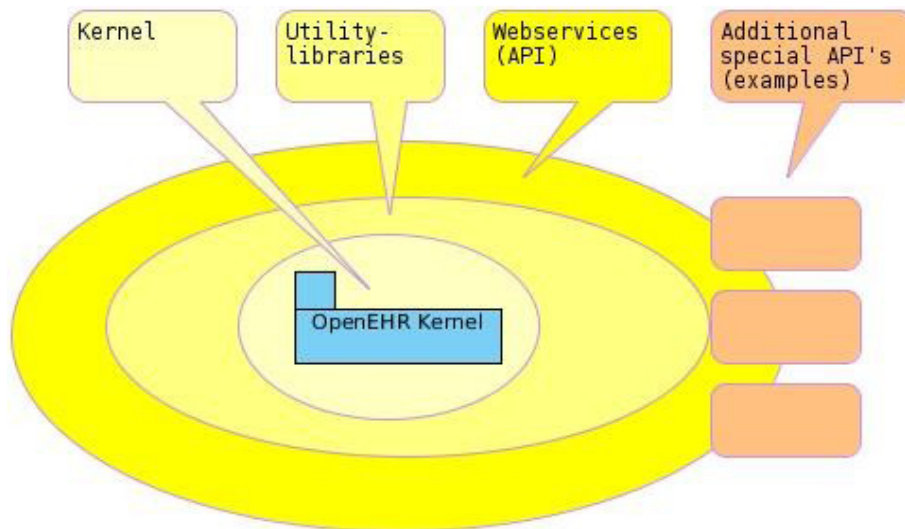


Figure 7: OpenEHR Kernel interacting with the outside world

This makes the kernel a flexible, extendible modular software-environment which seems suitable for getting the role of Virtual EHR in the system.

The kernel exposes itself via an API. This API is typically in the OpenEHR concept-context. This can be reached using web services (SOAP). On request other protocols can be added. Because of the layered architecture, services do not always need to reside on the same system. It is possible to write sub service-layers which can expose a more client-optimized API. This sub service-layer must be seen as an additional business-layer or as a part of the client software. (In Figure 7 and in orange)

If a sub service-layer is used it is possible to add knowledge about the data inside the coding. For example: family-relations can be focused on and make easy accessible in this layer. The sub service-layer then converts the non-generic API to the generic archetype-based API, and then forwards the request to the generic OpenEHR-kernel. It is also possible to add functionality to the sub service-layer, for example, creating SOEP-sequences for medical-data. Filter data for different professionals like nurses or GPs. The sub service-layer can also expose a simple query-API which translates the simple queries to the more technically AQL-queries. It is possible to write as many sub service-layers as wanted/needed and use them simultaneously, for the same or different kind of clients. A sub service-layer has an API focused on special functionality, while the original service-layer has a generic API.

The Virtual EHR provides controlled access to the Virtual EHR itself and the various distributed data repositories. This is depicted in Figure 8.

The following draft versions for the archetypes delivering historical patient data to component (4), **DSS**, have been defined:

- edafmis-demographic-ADDRESS.address.v1.adl
- edafmis-demographic-ADDRESS.telephone.v1.adl
- edafmis-demographic-PARTY_IDENTITY.identity_identifier.v1.adl
- edafmis-demographic-PARTY_IDENTITY.identity_person_name.v1.adl
- edafmis-demographic-PERSON.person.v1.adl
- edafmis-EHR-COMPOSITION.report.v1.adl
- edafmis-EHR-GENERIC_ENTRY.admission.v1.adl

They are available on the Edafmis internal website.

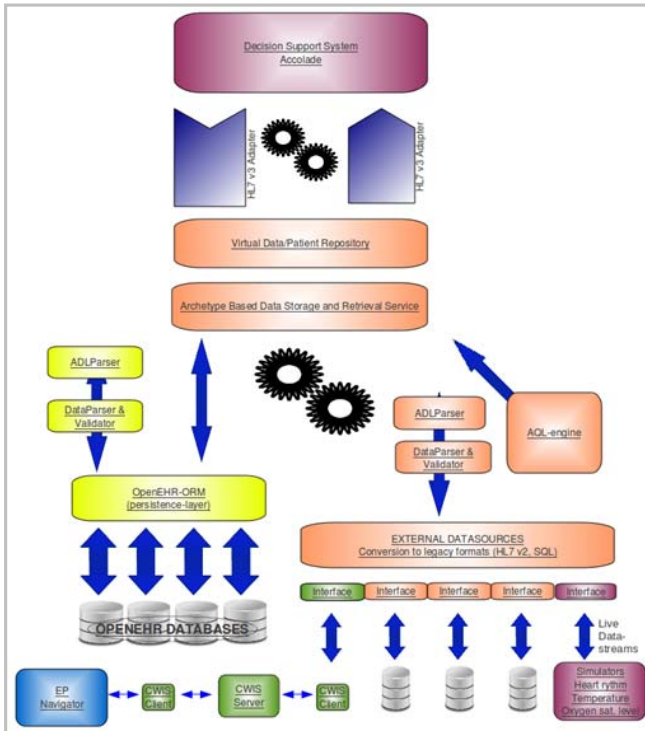


Figure 8: Virtual EHR interacting with the outside world

46.38.162.26:8080/axis2/services/listServices

The Apache Software Foundation
<http://www.apache.org/>

Available services

[DemographicService](#)

Service EPR : <http://46.38.162.26:8080/axis2/services/DemographicService>

Service Description : DemographicService

Service Status : *Active*

Available Operations

- modifyActor
- addContactToParty
- init
- removeAddressFromContact
- addPartyRelationship
- removePartyIdentityFromParty
- addPartyIdentityToParty
- createActor
- undeleteActor
- countActors
- retrieveVersionedLocatable
- deleteActor
- listActors
- removePartyRelationship
- addRoleToActor
- removeContactFromParty
- removeRoleFromActor
- addAddressToContact

[DirectoryService](#)

Service EPR : <http://46.38.162.26:8080/axis2/services/DirectoryService>

Service Description : DirectoryService

Service Status : *Active*

Available Operations

- retrieveRootFolder
- removeFolder
- createDirectoryRoot
- addItemToFolder
- removeItemFromFolder
- init
- addFolderToFolder

[TerminologyService](#)

Figure 9: Virtual EHR server (in the Cloud)

3.4 Decision Support System

Huub Rutten & Elsemiek ten Pas – Sopheon

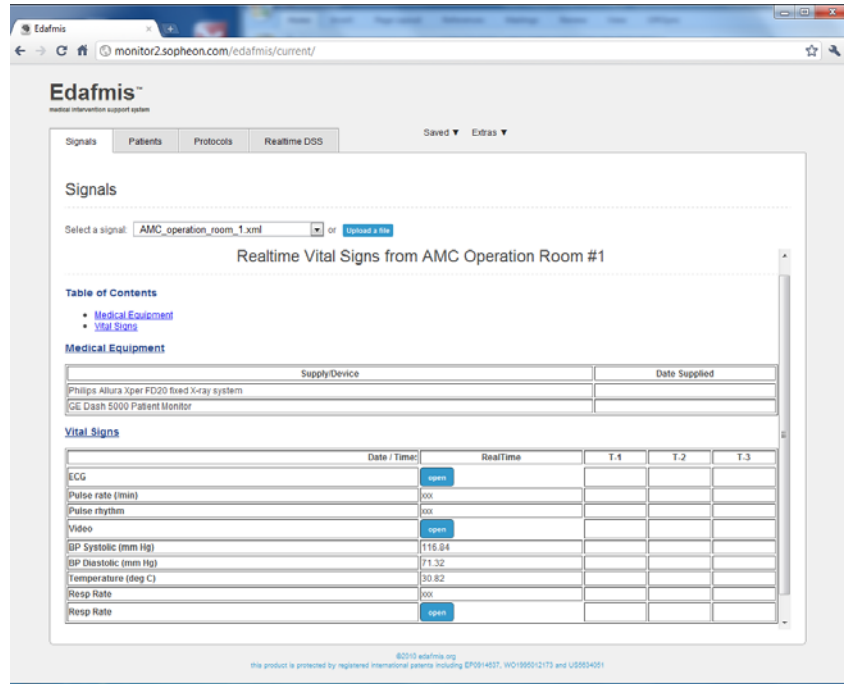
As shown in Figure 5 in 3.1, the Decision Support System (DSS) is the central hub in the real-time domain of the EDAFMIS Embedded Decision Support architecture. It interfaces with components 1, 2, 3 and 18 in the real-time domain, and with components 13 and 14 outside the real-time domain.

ID	Component	Interfaces
1,2,3	Sensors	<u>Outputs:</u> → Monitoring devices in the intervention room send vital signs data to the DSS
4	Decision Support System	<u>Inputs:</u> → DSS receives vital signs data from monitoring devices in the intervention room → DSS receives rules from the Protocol Management System → DSS receives real-time images from the Intervention System → DSS receives historical patient data from the Virtual EHR System <u>Outputs:</u> ← DSS sends a configuration log to the Protocol Management System ← DSS sends alerts to be displayed on the monitor(s) of the Intervention System ← DSS sends real-time patient data to the Virtual EHR System
18	EP Navigator	<u>Inputs:</u> → EP Navigator receives alerts from the DSS <u>Outputs:</u> ← EP Navigator sends real-time images from the Intervention System and PACS to the DSS
9	Automated Search (Monitor)	<u>Inputs:</u> → Monitor receives protocols from the protocol database → Monitor receives search results from the PubMed database, i.e. new publications that may be relevant for a given protocol <u>Outputs:</u> ← Monitor sends alerts on new references for a given protocol stored in the protocol database
13	Protocol Mgmt System	<u>Inputs:</u> → Protocol Management System receives protocol descriptions from the protocol database → Protocol Management System receives a configuration log from DSS <u>Outputs:</u> ← Protocol Management System sends rules to the DSS ← Protocol Management System sends protocol feedback to the protocol database
14	Virtual EHR	<u>Inputs:</u> → Virtual EHR System receives real-time patient data from the DSS <u>Outputs:</u> ← Virtual EHR System sends historical patient data to the DSS

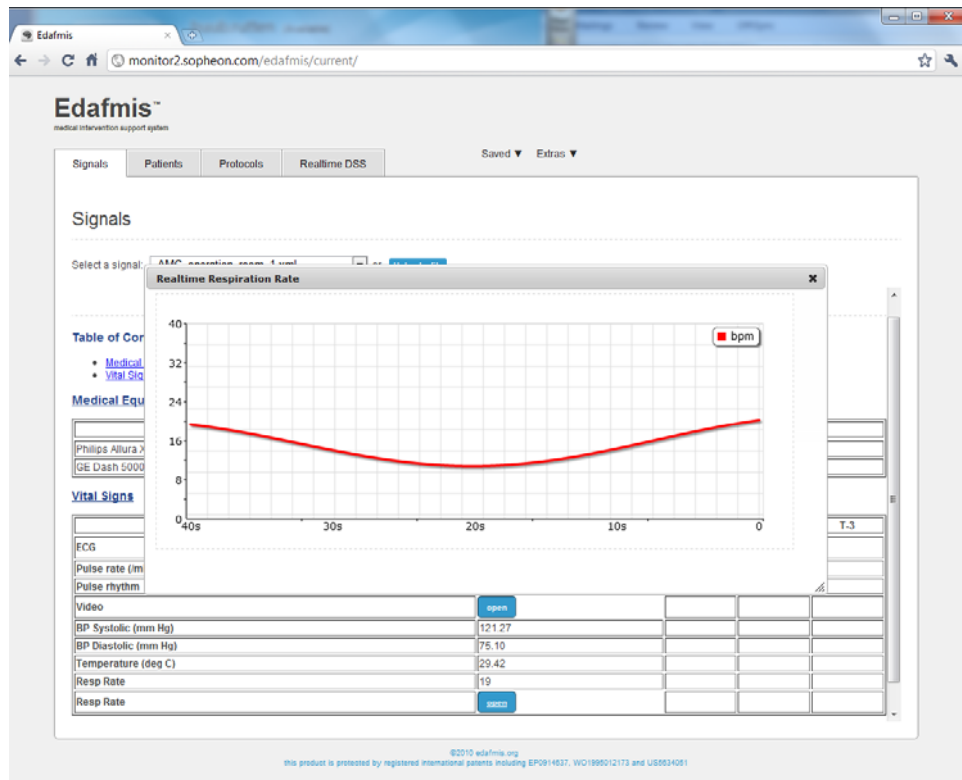
It is in the DSS that the fusion of signals, historical patient data and protocol data takes place.

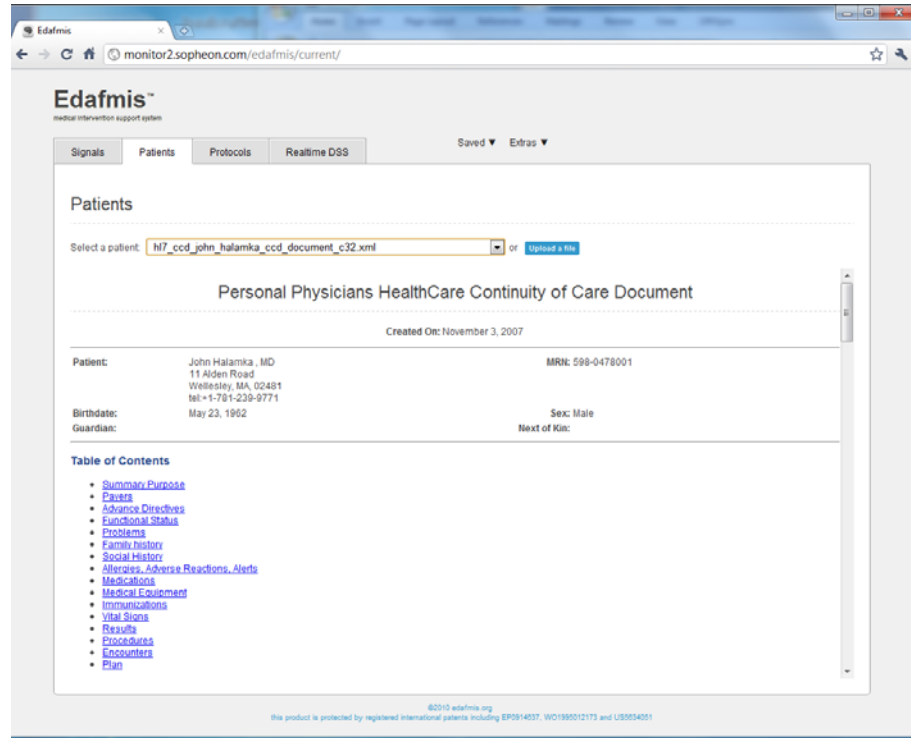
Sopheon created a web application to illustrate the tasks performed by the DSS. The screenshots below are taken from that web application.

1. The DSS calls selected signals and imaging data as specified in the (first phase of the procedure described in the) protocol.

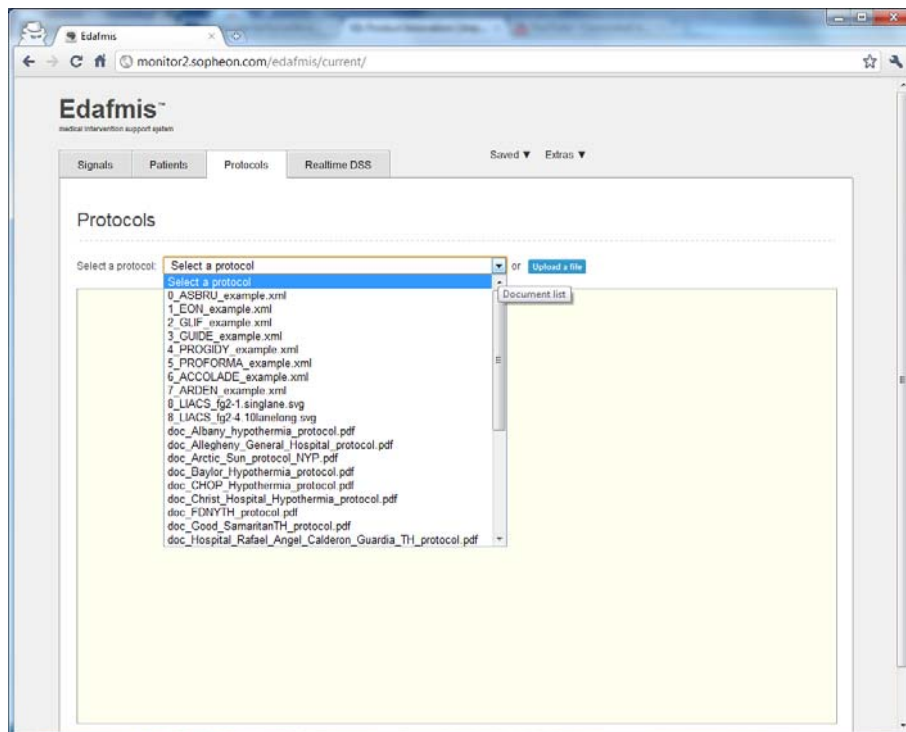


2. The clinician selects the relevant patient record.



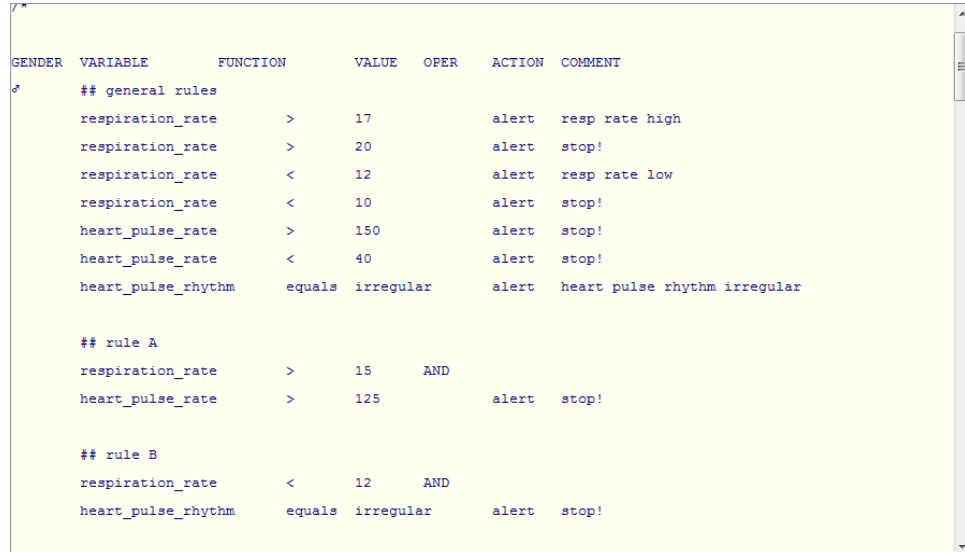


3. The clinician selects the relevant protocol.



4. The DSS rule-engine is programmed, initialised and tuned through 1) the interpretable sections in the patient record and 2) the interpretable sections in the protocol.

5. Once running the DSS-system is fed either raw real-time data from the various sensors/ monitoring devices, or algorithmically deduced values, e.g. heart rate (frequency), heart rhythm (regularity) computed from the raw ECG signal.
6. It is through the combination of real-time data and rules that the DSS can act as an alerting system. If the monitoring and imaging data exceed a limit laid down in the active protocol, the DSS generates an alert.



```

GENDER  VARIABLE      FUNCTION      VALUE  OPER  ACTION  COMMENT
♂
## general rules
respiration_rate  >      17          alert  resp rate high
respiration_rate  >      20          alert  stop!
respiration_rate  <      12          alert  resp rate low
respiration_rate  <      10          alert  stop!
heart_pulse_rate  >      150         alert  stop!
heart_pulse_rate  <      40          alert  stop!
heart_pulse_rhythm equals irregular  alert  heart pulse rhythm irregular

## rule A
respiration_rate  >      15          AND
heart_pulse_rate  >      125         alert  stop!

## rule B
respiration_rate  <      12          AND
heart_pulse_rhythm equals irregular  alert  stop!

```

Figure 10: Extract from the executable rules

3.5 Protocol management

Huub Rutten & Elsemiek ten Pas – Sopheon & Christoph Stettina – CeTIM

Protocol Management is depicted by the components ID 9-13 in Figure 5.

The Protocol Management System (ID13) is implemented in Accolade, Sopheon’s platform for innovation process management. A dedicated configuration, named QualiFlow, was made to support the creation and maintenance of protocols including a formal authorization step and proper versioning.

The Protocol Management System directly interfaces the Decision Support System (ID 4) and provides the rules that can be executed by the DSS within the real-time domain.

In QualiFlow each protocol version is a project consisting of one stage (the making of the protocol) and one gate (approval of the protocol version paving the way for publication). The author and authorizer of the protocol version have to be different persons.

The screenshot below, Figure 11, shows the project stage page in QualiFlow.

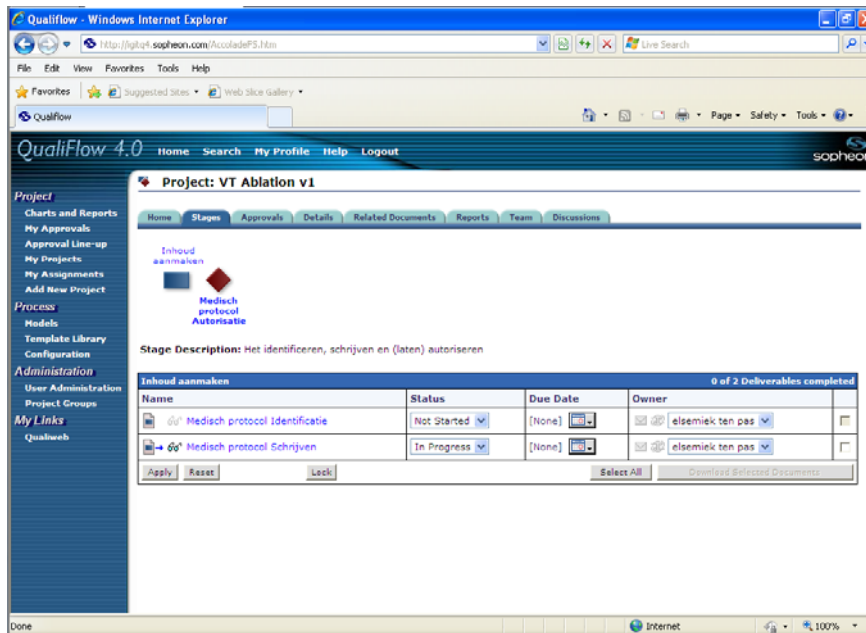


Figure 11: QualiFlow stage page

The content of the protocol is edited in MS Word expanded with a special QualiFlow add-in. The add-in provides the protocol author with special functions to structure and quality-check the content, as well as the ability to save a version directly to QualiFlow.

Protocols are always based on a template providing a predefined document structure. QualiFlow offers a range of templates corresponding with different protocol types. The selection of a protocol type is compulsory while creating a project in QualiFlow.

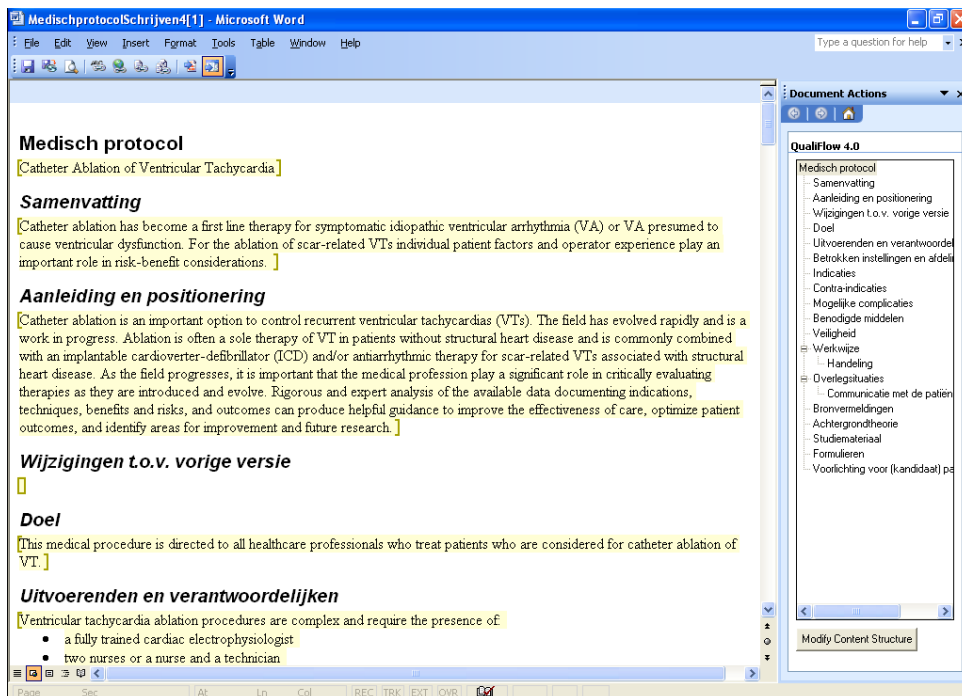


Figure 12: QualiFlow protocol page

Linked to the Protocol Database is the Automated Search component called Monitor (ID9), a reading robot that automatically keeps track of new evidence on the medical subjects described in the protocols. It can automatically translate a document such as a protocol into a query and notify the protocol owner of new publications that may be relevant for keeping the protocol up-to-date and evidence-based. The reading robot is connected to the renowned PubMed database.

ID	Component	Interfaces
13	Protocol Management System	<u>Inputs:</u> → Protocol Management System receives protocol descriptions from the Protocol Database → Protocol Management System receives a configuration log from DSS <u>Outputs:</u> ← Protocol Management System sends rules to the DSS ← Protocol Management System sends protocol feedback to the Protocol Database
9	Automated Search (Monitor)	<u>Inputs:</u> → Monitor receives protocols from the Protocol Database → Monitor receives search results from the PubMed database, i.e. new publications that may be relevant for a given protocol <u>Outputs:</u> ← Monitor sends alerts on new references that may be relevant for a given protocol stored in the Protocol Database

4 Summary

Frank van der Linden – Philips Healthcare

This document provides the requirements and design of the Edafmis demonstrators, all combined support operational execution during minimal invasive interventions. After some end-user requirements, the requirements of the different systems are described. The section on design starts with a description of the complete set-up of systems form minimal-invasive intervention support. Some of them are designed as demonstrator in this project. The others are existing systems in a hospital environment. The designs of the demonstrators are given as well – except the mobility connection for virtual team support.

Note that the set-up and the designs is an initial set-up based on available equipment and the knowledge of the participants in this work package. However, the set-up gives the necessary insights in the complete set-up of commercial tooling in intervention support planning and navigation.