# Integrated Reference Model
# Deliverable D1.3
# Sopheon

**June 2011**

| **Contributors** | **Affiliation** |
|---|---|
| Luuk Groenewegen | CeTIM / LIACS |
| Christoph Stettina | CeTIM / LIACS |
| Roger Erens | Zorggemak |
| Peter Eshuis | Philips |
| Huub Rutten | Sopheon |
| Elsemiek ten Pas | Sopheon |
| Steven Kleynenberg | Sopheon |

## Table of contents

# 1. Introduction

This document reports on the research done to analyse and define Reference Models that are relevant to the objectives of the Edafmis project. The report belongs to WP1 of the Edafmis project.

Background:

WP1's main objective: to deliver the conceptual data model (Reference Model) allowing to define and execute a clinical pathway for (semi-) automated intervention; where possible the WP will reuse existing industrial standards, however will develop related to additional needs; to deliver the protocol handling system that creates and manages the protocol in the form of a program that executes the clinical pathway; this can be used and evaluated during experimenting and prototyping.

> *It is clear that the conceptual and technical integration of various data producing and absorbing devices around the clinical pathway and the precise interpretation of their various datasets are conditional for meaningful communication and cooperation. This cannot be realised without having an agreed Reference Model, that allows for a stable and unambiguous "deep" definition of the clinical pathway "program", however while still allowing source systems to stick to their proprietary software languages. Therefore the WP will need close cooperation with those partners that deliver a device to the workflow such as the ERP system, the mobile scan exchanging system, the intervention system, the protocol management system, to agree the input/output schemata.*

There are 4 dimensions or perspectives on the research route to the objective mentioned above. Together they form the structure that are imperative for all the deliverables of Workpackage 1, including deliverable D1.3 *Integrated Reference Model* . These dimensions are represented in the tasks as described in the project description as follows:

**Task 1.1 Clinical pathway representation**

*This Task will deliver those models that define the medical aspects of the diagnosis and treatment of a disease. To do so it needs to establish the "language" in terms of function and form that can be used to express these aspects, both at a human level and at a technical (software) level. How to express the basic or starting configuration? The models will contain property-value combinations as well as rules that manage the behaviour of the treatment instruction. On one hand it should enable for medical experts to express their detailed definitions, on the other hand it should ensure the right data collection during diagnosis and intervention and the consequences thereof. For example to stop the treatment or to change strategy. The Clinical Pathway models will be tested by producing a Clinical pathway program, that will be used for prototyping and experiments in the integrated Edafmis environment.*

**Task 1.2 Process data representation**

*This tasks focuses on the flow-aspects of the diagnosis and intervention procedures. It will model the sequence and conditional aspects of activities and events in a way that we can consider the Clinical Path to be a "secured" workflow. It will define in detail when which data sets should be pushed out to or pulled in from the various devices and tools participating in the process. The task will deliver the conceptual interface requirements as well as the common reference model.*

**Task 1.3 Patient data representation**

*New and existing imaging modalities make an increasing amount of potentially important* *information available for any given patient. Furthermore, systems for image-guided minimal invasive surgery and therapy are increasingly being used in order to safely navigate surgical instruments into the human body for improved patient treatment. Essential questions in multi-modal navigation are: What information is needed for optimal surgical planning and guidance in the operating theatre? How should all this multi-modal information be presented to the surgeon? And how can technology ensure that information displayed in the navigation system represents an accurate picture of the intra-operative anatomy throughout the operation?*

**Task 1.4 Dynamic patient model representation**

*When the patient is in the Intervention phase the body will be monitored on a continuous basis from various perspectives, in parallel. The dynamic patient model instructs or suggests the intervention systems what to do and what not to do within certain tolerance margins. This task should define the patient model dynamics as a set of probable data combinations (patterns) that might appear and have preprogrammed logical consequences. It also should define how to handle “uncertain” or “unexpected” combinations of monitoring data. These rules need to be expressed in the Clinical Pathway software and executed by the Intervention System.*

However, in the course of the Edafmis project the task structure as described at the beginning of the project is very much operationalised in a different structure, the structure of the system architecture that has been developed. This system architecture contains a series of components or system modules each of which has internal data structures and each of which has exchange structures allowing to integrate with and interface to other components.

These components are:

1. a Protocol Management system, that defines the clinical pathway and includes the calculation rules for the real time decision support
2. a real time DSS -- Decision Support system
3. an Intervention system, including the EP Navigator
4. a virtual EHR -- Electronic Health Record system
5. a Mobile Device -- for collaborative patient image evaluation (this will be treated in a separate deliverable D3.2)

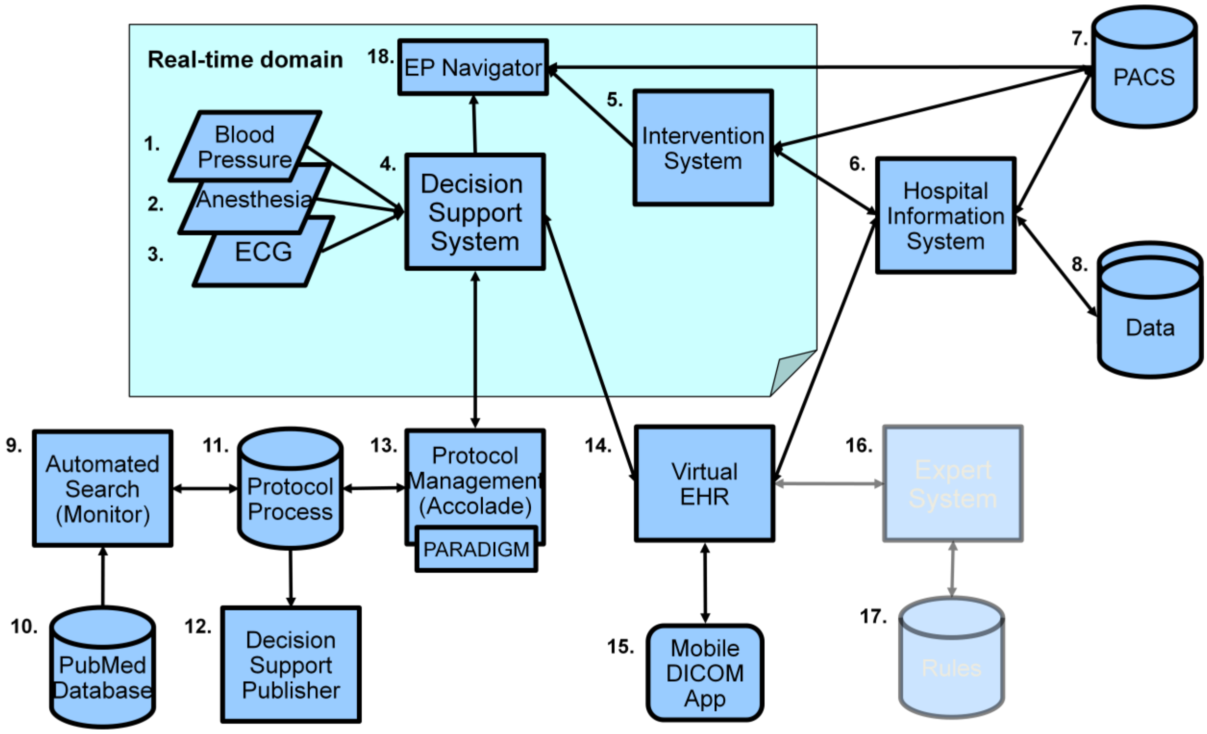


**Figure 1: Edafmis System Architecture**

It makes more sense to describe the reference models used by these Edafmis components, internally as well as externally for communication with the other components. Therefore this deliverable uses this structure as basis for reporting.

# 2. Protocol Management

*Sopheon and CeTIM*

Protocol Management is a business process, that produces operational (medical) protocols for how to diagnose and treat patients for a specific disease, in our example and use case VT Ablation. The protocol contains not only the description of the what and how to do things in the form of instructions for humans, but also the instruction for the DSS software program .

## 2.1 The Protocol Management process as supported by the Accolade system

The Accolade system is a web based system as shown in Fig. 2 below.

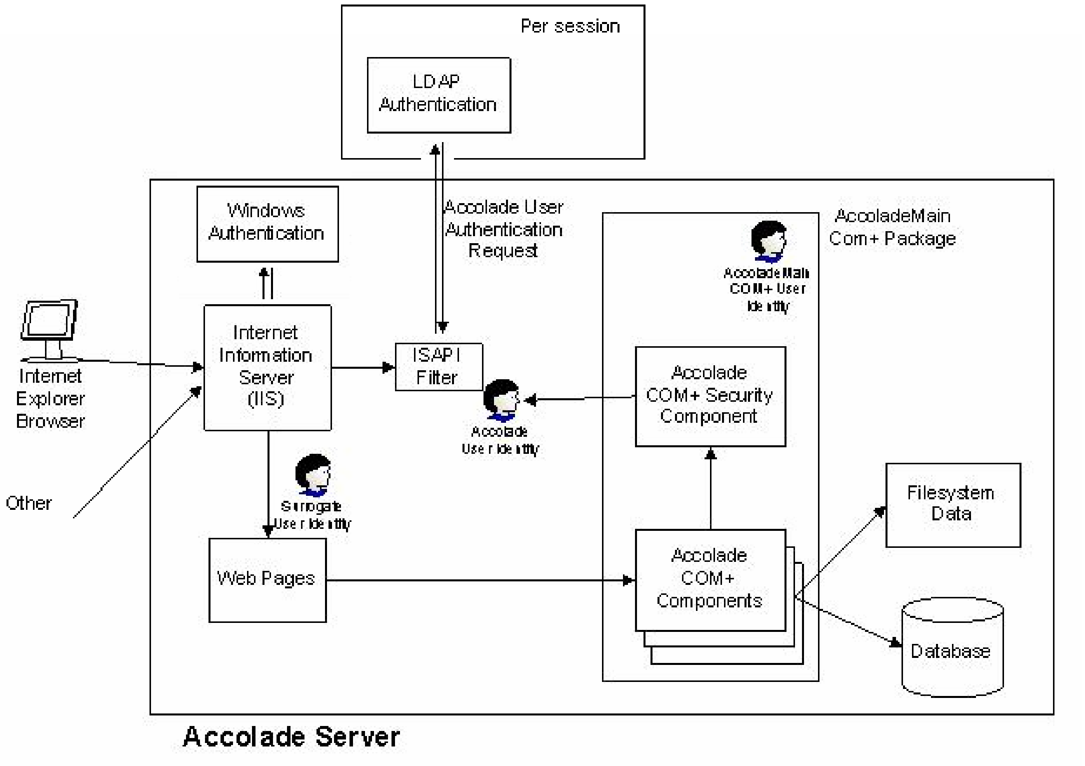


**Figure 2: Accolade Server Architecture**

It is a configurable system that allows organisations to add "metrics" that are required to manage a protocol through its lifecycle. It also comes out of the box with metadata placeholders that are required to manage a stage-gated process. In our case we need that very much, because protocol management has to be very precise and traceable. The picture below gives an idea of the underlying stage-gate datamodel.

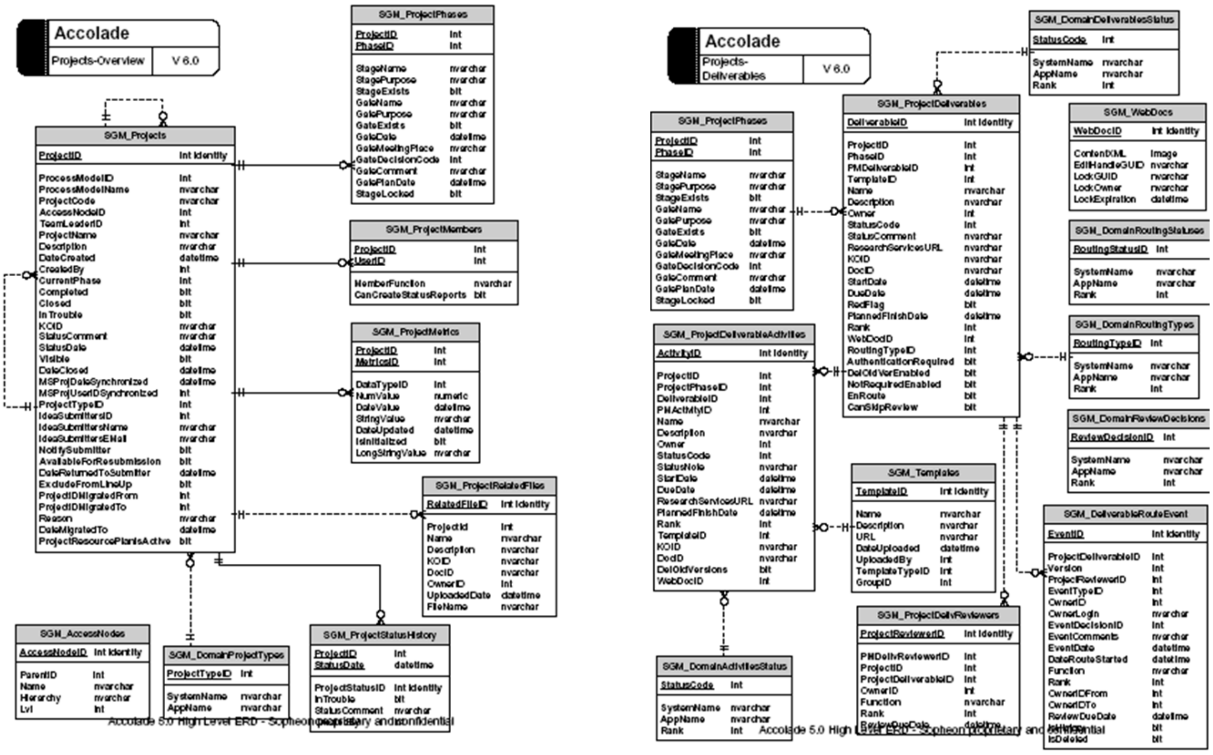
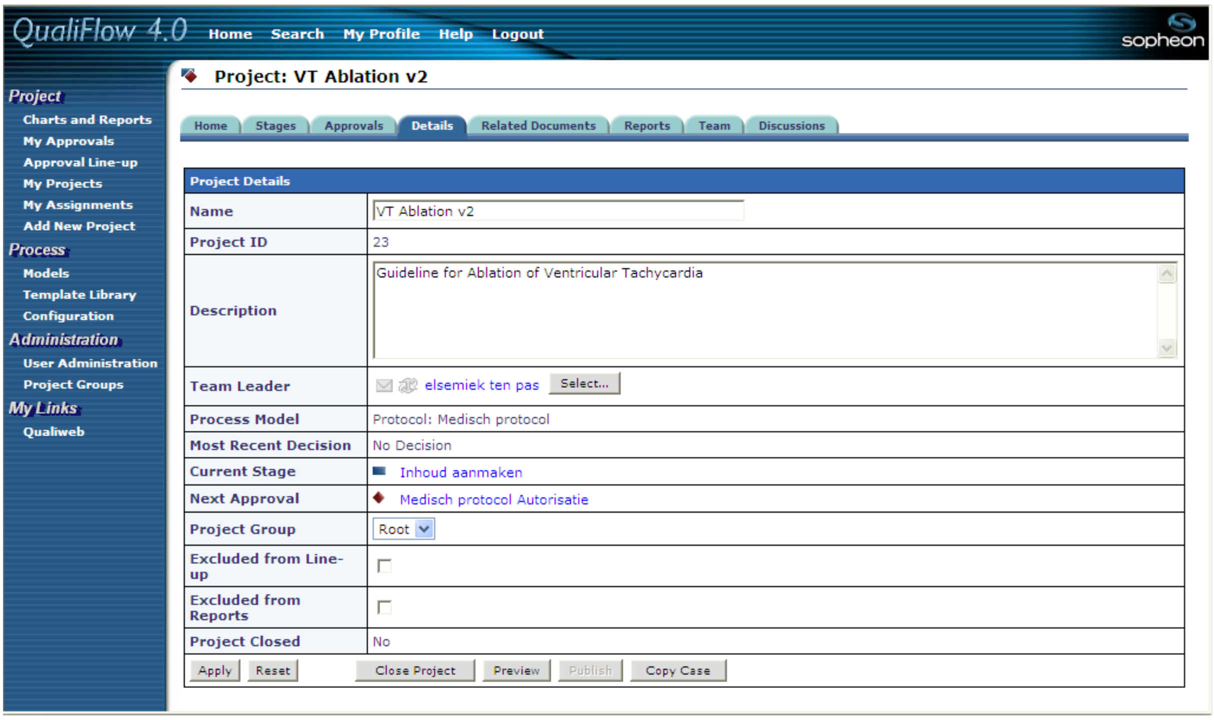

Accolade
Projects-Overview
V 6.0
SGM_Projects
SGM_ProjectPhases
SGM_ProjectMembers
SGM_ProjectMetrics
SGM_ProjectRelatedFiles
SGM_AccessNodes
SGM_DomainProjectTypes
SGM_ProjectStatusHistory
Accolade 5.0 High Level ERD - Sopheon proprietary and confidential
Accolade
Projects-Deliverables
V 6.0
SGM_DomainDeliverablesStatus
SGM_ProjectPhases
SGM_ProjectDeliverables
SGM_WebDocs
SGM_DomainRoutingStatuses
SGM_DomainRoutingTypes
SGM_DomainReviewDecisions
SGM_ProjectDeliverableActivities
SGM_Templates
SGM_DeliverableRouteEvent
SGM_ProjectDelivReviewers
SGM_DomainActivitiesStatus
Accolade 5.0 High Level ERD - Sopheon proprietary and confidential

**Figure 3: Stage-gate Data Model**

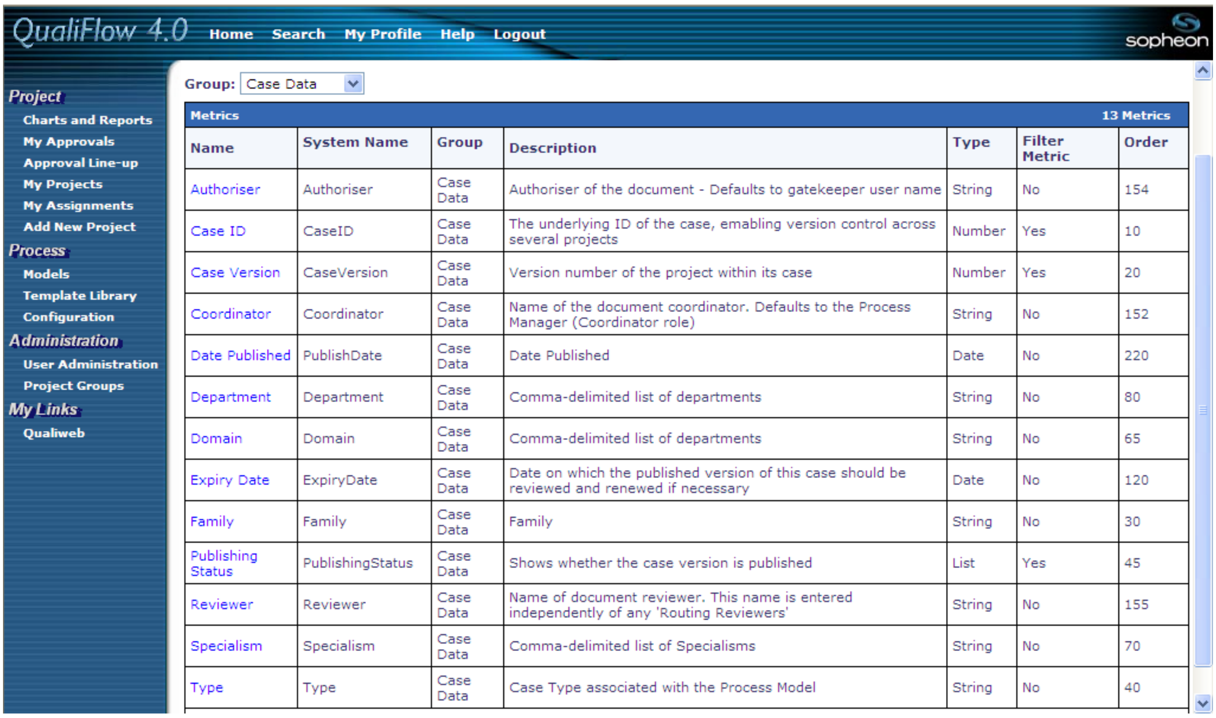
On top of this stage-gate model we have developed the metrics that are specific for the protocol management process.



Project: VT Ablation v2

| Project Details | |
|---|---|
| Name | VT Ablation v2 |
| Project ID | 23 |
| Description | Guideline for Ablation of Ventricular Tachycardia |
| Team Leader | elsemiek ten pas Select... |
| Process Model | Protocol: Medisch protocol |
| Most Recent Decision | No Decision |
| Current Stage | Inhoud aanmaken |
| Next Approval | Medisch protocol Autorisatie |
| Project Group | Root |
| Excluded from Line-up | □ |
| Excluded from Reports | □ |
| Project Closed | No |

**Figure 4: Project details for a protocol version**

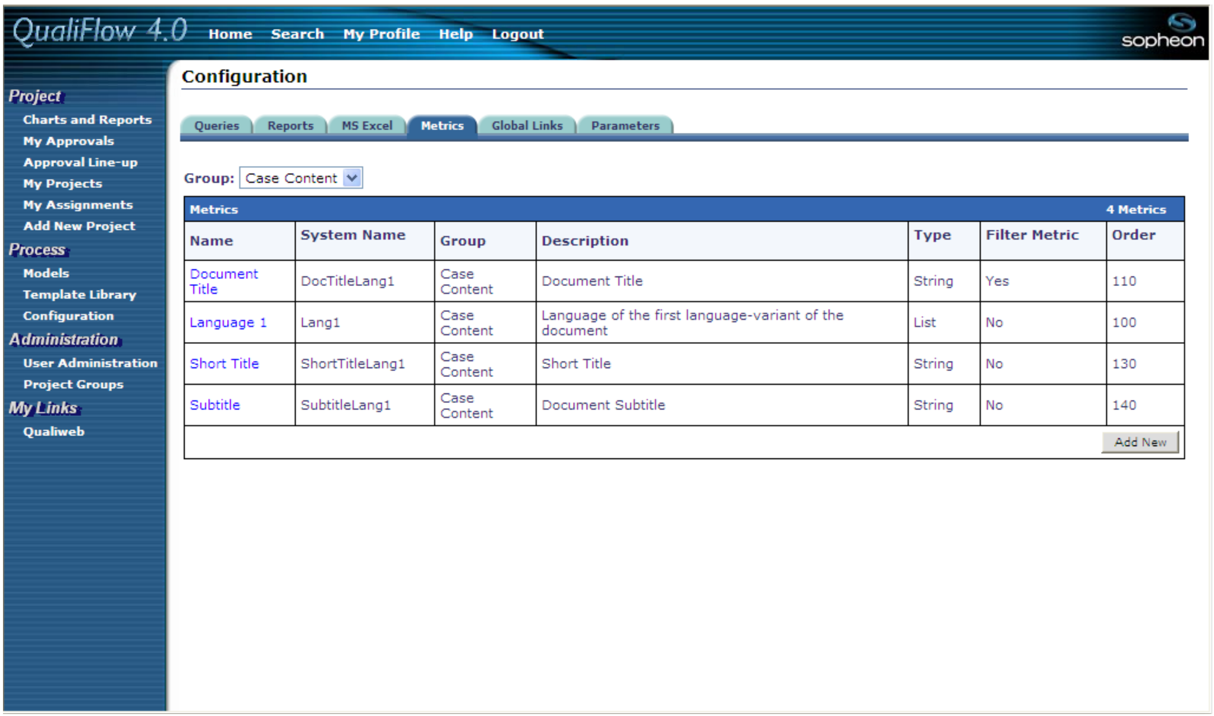


Group: Case Data

Metrics — 13 Metrics

| Name | System Name | Group | Description | Type | Filter Metric | Order |
|---|---|---|---|---|---|---|
| Authoriser | Authoriser | Case Data | Authoriser of the document - Defaults to gatekeeper user name | String | No | 154 |
| Case ID | CaseID | Case Data | The underlying ID of the case, emabling version control across several projects | Number | Yes | 10 |
| Case Version | CaseVersion | Case Data | Version number of the project within its case | Number | Yes | 20 |
| Coordinator | Coordinator | Case Data | Name of the document coordinator. Defaults to the Process Manager (Coordinator role) | String | No | 152 |
| Date Published | PublishDate | Case Data | Date Published | Date | No | 220 |
| Department | Department | Case Data | Comma-delimited list of departments | String | No | 80 |
| Domain | Domain | Case Data | Comma-delimited list of departments | String | No | 65 |
| Expiry Date | ExpiryDate | Case Data | Date on which the published version of this case should be reviewed and renewed if necessary | Date | No | 120 |
| Family | Family | Case Data | Family | String | No | 30 |
| Publishing Status | PublishingStatus | Case Data | Shows whether the case version is published | List | Yes | 45 |
| Reviewer | Reviewer | Case Data | Name of document reviewer. This name is entered independently of any 'Routing Reviewers' | String | No | 155 |
| Specialism | Specialism | Case Data | Comma-delimited list of Specialisms | String | No | 70 |
| Type | Type | Case Data | Case Type associated with the Process Model | String | No | 40 |

**Figure 5: Process metrics**



Configuration

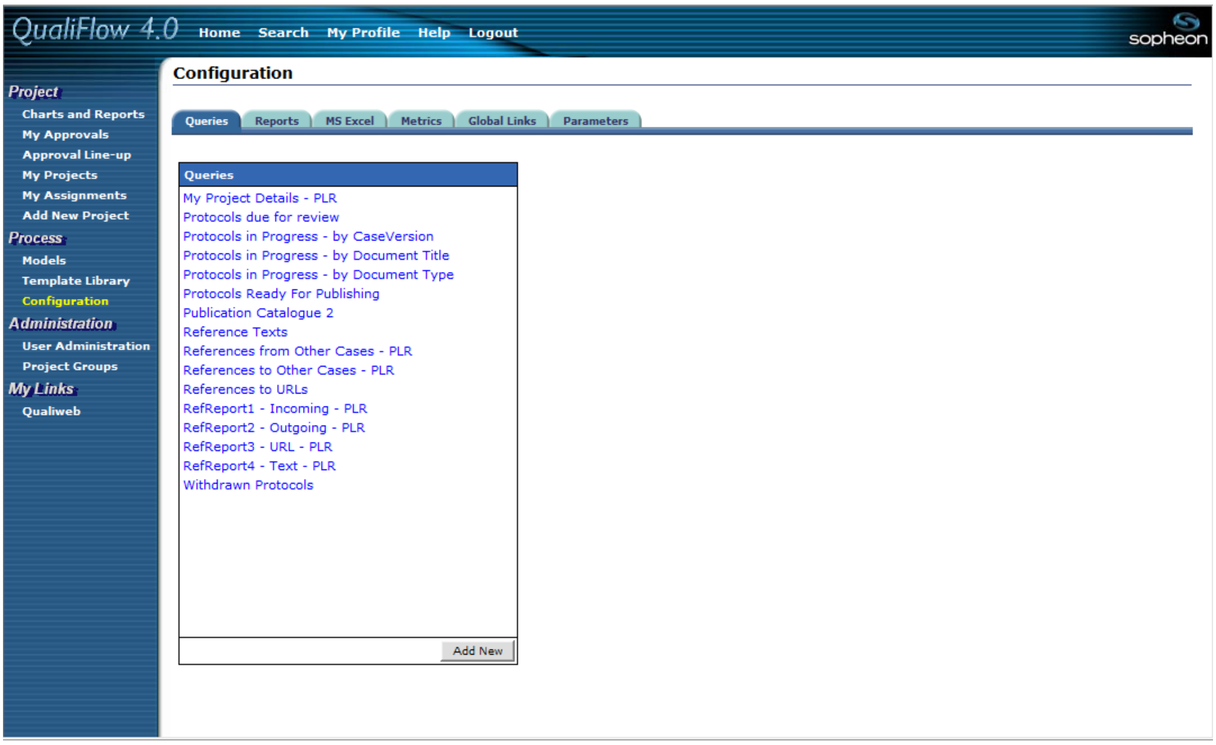
Group: Case Content

Metrics — 4 Metrics

| Name | System Name | Group | Description | Type | Filter Metric | Order |
|---|---|---|---|---|---|---|
| Document Title | DocTitleLang1 | Case Content | Document Title | String | Yes | 110 |
| Language 1 | Lang1 | Case Content | Language of the first language-variant of the document | List | No | 100 |
| Short Title | ShortTitleLang1 | Case Content | Short Title | String | No | 130 |
| Subtitle | SubtitleLang1 | Case Content | Document Subtitle | String | No | 140 |

**Figure 6: Content metrics**

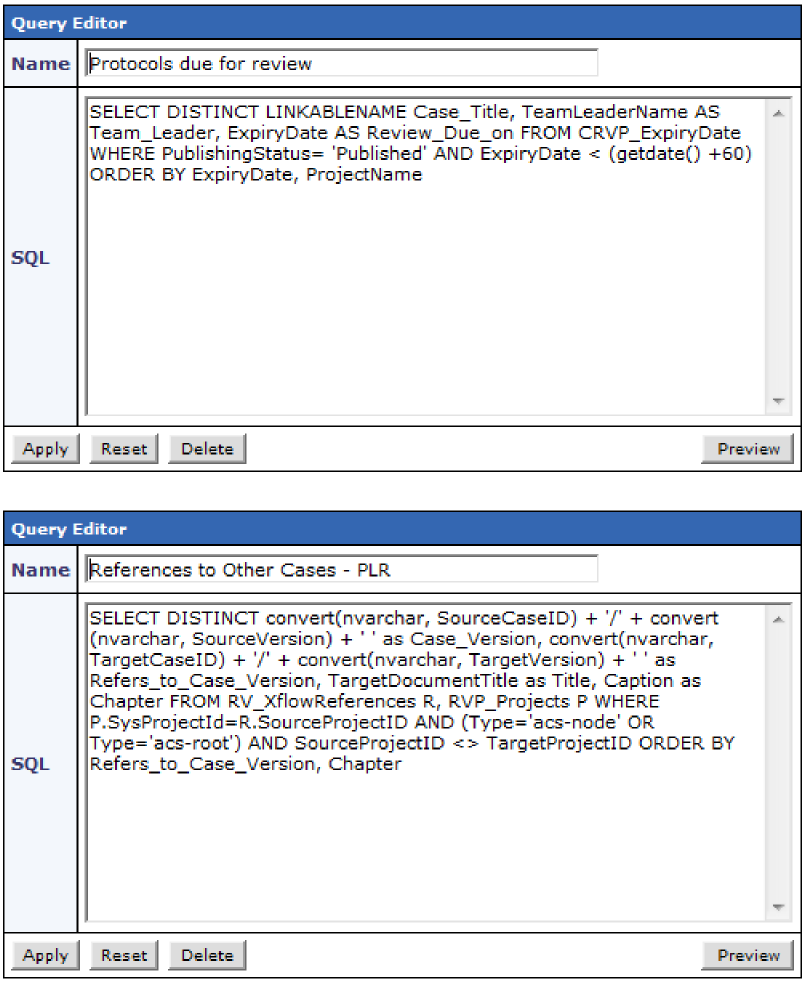
The metrics are addressable to configure queries that collect data from the database in views that are required to see progress and status of protocols.



Queries

- My Project Details - PLR
- Protocols due for review
- Protocols in Progress - by CaseVersion
- Protocols in Progress - by Document Title
- Protocols in Progress - by Document Type
- Protocols Ready For Publishing
- Publication Catalogue 2
- Reference Texts
- References from Other Cases - PLR
- References to Other Cases - PLR
- References to URLs
- RefReport1 - Incoming - PLR
- RefReport2 - Outgoing - PLR
- RefReport3 - URL - PLR
- RefReport4 - Text - PLR
- Withdrawn Protocols

## Figure 7: List of queries for collecting protocols



Name: Protocols due for review

SQL:

```
SELECT DISTINCT LINKABLENAME Case_Title, TeamLeaderName AS
Team_Leader, ExpiryDate AS Review_Due_on FROM CRVP_ExpiryDate
WHERE PublishingStatus= 'Published' AND ExpiryDate < (getdate() +60)
ORDER BY ExpiryDate, ProjectName
```

Name: References to Other Cases - PLR

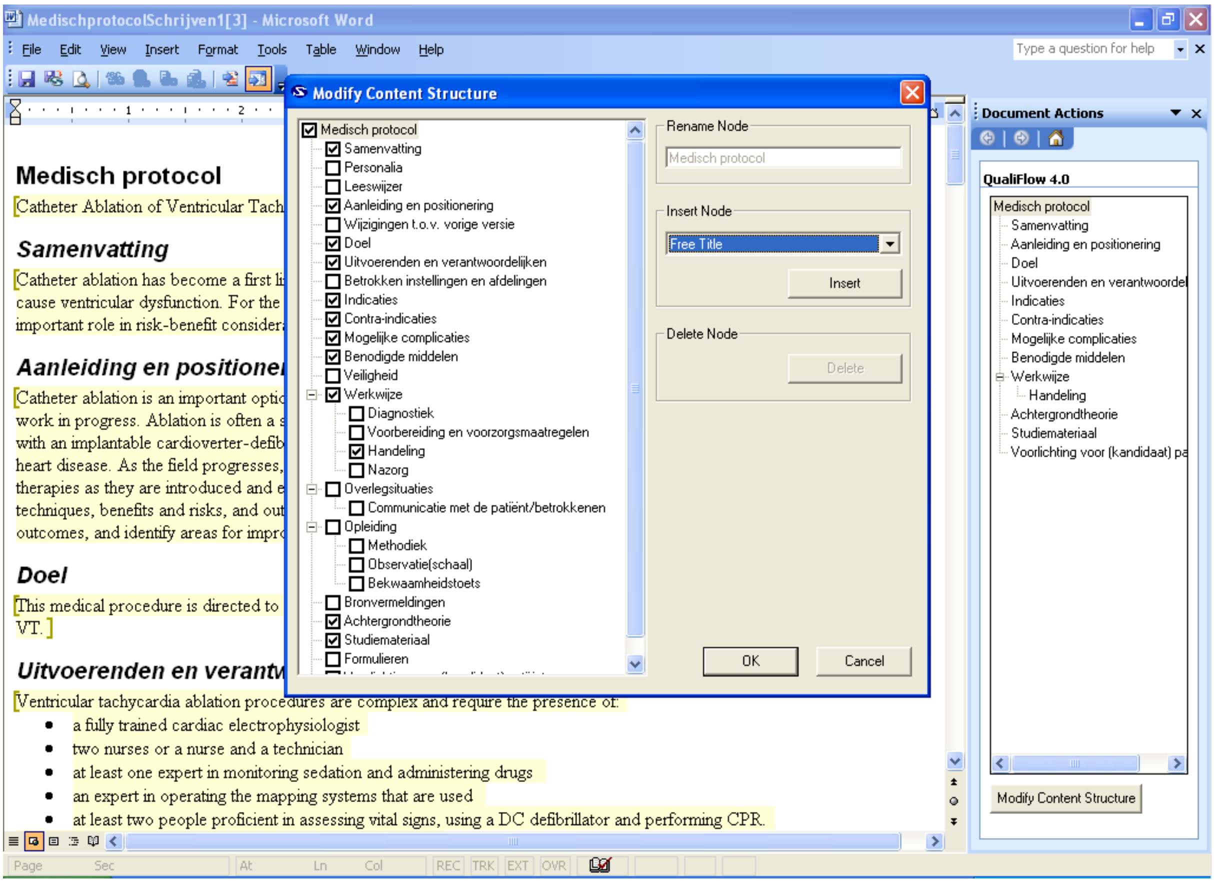
SQL:

```
SELECT DISTINCT convert(nvarchar, SourceCaseID) + '/' + convert
(nvarchar, SourceVersion) + ' ' as Case_Version, convert(nvarchar,
TargetCaseID) + '/' + convert(nvarchar, TargetVersion) + ' ' as
Refers_to_Case_Version, TargetDocumentTitle as Title, Caption as
Chapter FROM RV_XflowReferences R, RVP_Projects P WHERE
P.SysProjectId=R.SourceProjectID AND (Type='acs-node' OR
Type='acs-root') AND SourceProjectID <> TargetProjectID ORDER BY
Refers_to_Case_Version, Chapter
```

## Figure 8: Examples of metric queries

Managing a protocol is in fact separated from its content. Editing and publishing the content however are embedded in the management process.

To edit a protocol we have developed a special editing workbench with Word. It works with prefab (XML) templates that can be selected or changed.



Medisch protocol

[Catheter Ablation of Ventricular Tach

Samenvatting

[Catheter ablation has become a first li cause ventricular dysfunction. For the important role in risk-benefit consider

Aanleiding en positione

[Catheter ablation is an important optic work in progress. Ablation is often a s with an implantable cardioverter-defib heart disease. As the field progresses, therapies as they are introduced and e techniques, benefits and risks, and out outcomes, and identify areas for impr

Doel

[This medical procedure is directed to VT.]

Uitvoerenden en verantw

[Ventricular tachycardia ablation procedures are complex and requre the presence of:

- a fully trained cardiac electrophysiologist
- two nurses or a nurse and a technician
- at least one expert in monitoring sedation and administering drugs
- an expert in operating the mapping systems that are used
- at least two people proficient in assessing vital signs, using a DC defibrillator and performing CPR.

☑ Medisch protocol
- ☑ Samenvatting
- □ Personalia
- □ Leeswijzer
- ☑ Aanleiding en positionering
- □ Wijzigingen t.o.v. vorige versie
- ☑ Doel
- ☑ Uitvoerenden en verantwoordelijken
- □ Betrokken instellingen en afdelingen
- ☑ Indicaties
- ☑ Contra-indicaties
- ☑ Mogelijke complicaties
- ☑ Benodigde middelen
- □ Veiligheid
- ☑ Werkwijze
  - □ Diagnostiek
  - □ Voorbereiding en voorzorgsmaatregelen
  - ☑ Handeling
  - □ Nazorg
- □ Overlegsituaties
  - □ Communicatie met de patiënt/betrokkenen
- □ Opleiding
  - □ Methodiek
  - □ Observatie(schaal)
  - □ Bekwaamheidstoets
- □ Bronvermeldingen
- ☑ Achtergrondtheorie
- ☑ Studiemateriaal
- □ Formulieren

Rename Node: Medisch protocol

Insert Node: Free Title

Medisch protocol
- Samenvatting
- Aanleiding en positionering
- Doel
- Uitvoerenden en verantwoordel
- Indicaties
- Contra-indicaties
- Mogelijke complicaties
- Benodigde middelen
- Werkwijze
  - Handeling
- Achtergrondtheorie
- Studiemateriaal
- Voorlichting voor (kandidaat) pa

**Figure 9: Example of a prefab template for a medical protocol**

On top of this there is a specific rule editor that uses prefab categories/attributes for creating deeper embedded structures. The DSS engine can find these rules easily and can execute them as a software program. In fact we convert the XML written rules into a Javascripted program.

The table below provides an overview of the prefab categories available in the rule editor.

| | |
|---|---|
| Access<br>Access code<br>Accessibility<br>Action<br>Activity<br>Additional research<br>Aftercare<br>Alternative diagnosis<br>Anamnesis<br>Antibiotics<br>Article number<br>Author<br>Authorizer<br>Care<br>Causes | How<br>Indication<br>Initial<br>Input<br>Instrumentation<br>Instruments<br>Interactions<br>Lab<br>Location<br>Management<br>Materials<br>Means<br>Medication<br>Name<br>Number |

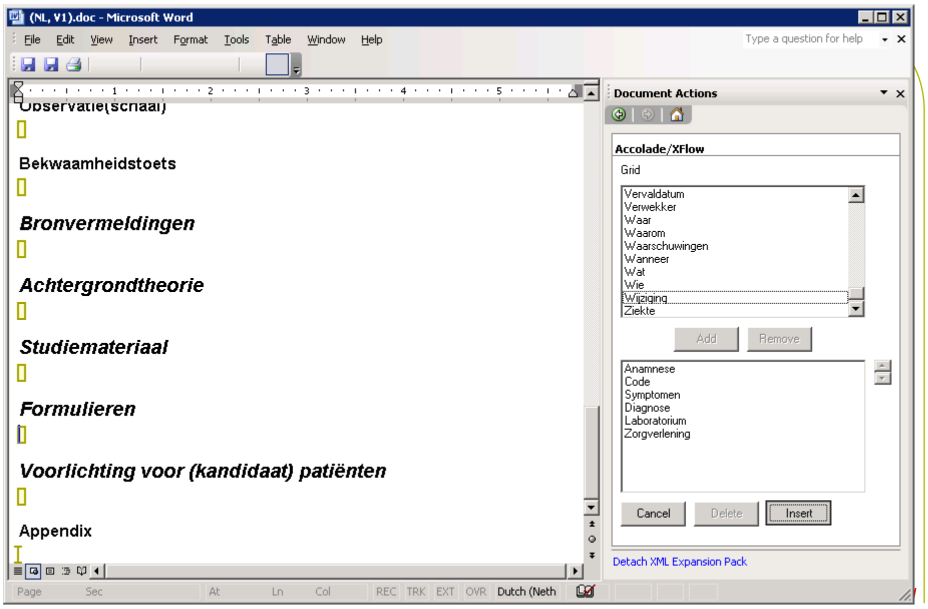
Change
Chapter
Code
Comments
Complications
Composition
Conditions
Contents
Contra-indications
Course
Department
Description
Diagnosis
Discussion
Disease
Document
Dosage
Duration
Evaluation
Examination
Execution
Executor
Expiration date
Feature
Frequency
Function
Goal
Health Unit

Number
Option
Order packet
Output
Participants
Parts
Pathogen
Patient Info
Physical behaviour
Physical Examination
Policy
Post-operative
Pre-operative
Procreator
Reviewer
Status
Symptoms
Task
Title
Treatment
Type
Type
Warnings
What
When
Where
Who
Why

These categories are presented to the user when he creates a so called grid. A grid is a table built by the user by selecting from the above-mentioned prefab categories. Each row of the GRID is like a rule. The categories are seen by the DSS as parameters that should meet certain values.

## Select GRID Components

sopheon



Observatie(schaal)

Bekwaamheidstoets

*Bronvermeldingen*

*Achtergrondtheorie*

*Studiemateriaal*

*Formulieren*

*Voorlichting voor (kandidaat) patiënten*

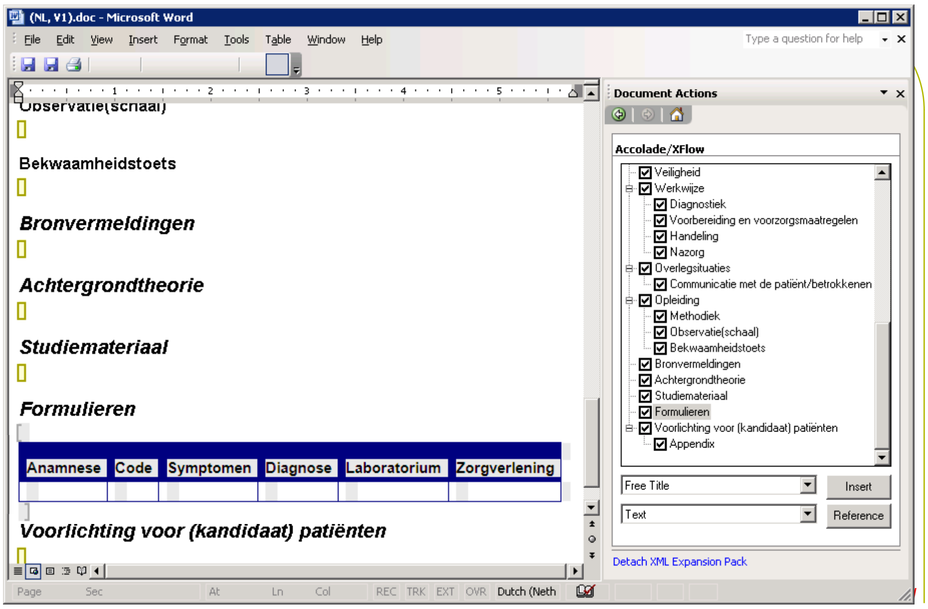
Appendix

Grid

Vervaldatum
Verwekker
Waar
Waarom
Waarschuwingen
Wanneer
Wat
Wie
Wijziging
Ziekte

Anamnese
Code
Symptomen
Diagnose
Laboratorium
Zorgverlening

## Edit the GRID like a normal table


(NL, V1).doc - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
Document Actions
Detach XML Expansion Pack
Page Sec At Ln Col REC TRK EXT OVR Dutch (Neth

Observatie(schaal)

Bekwaamheidstoets

*Bronvermeldingen*

*Achtergrondtheorie*

*Studiemateriaal*

*Formulieren*

| Anamnese | Code | Symptomen | Diagnose | Laboratorium | Zorgverlening |
|---|---|---|---|---|---|
| | | | | | |

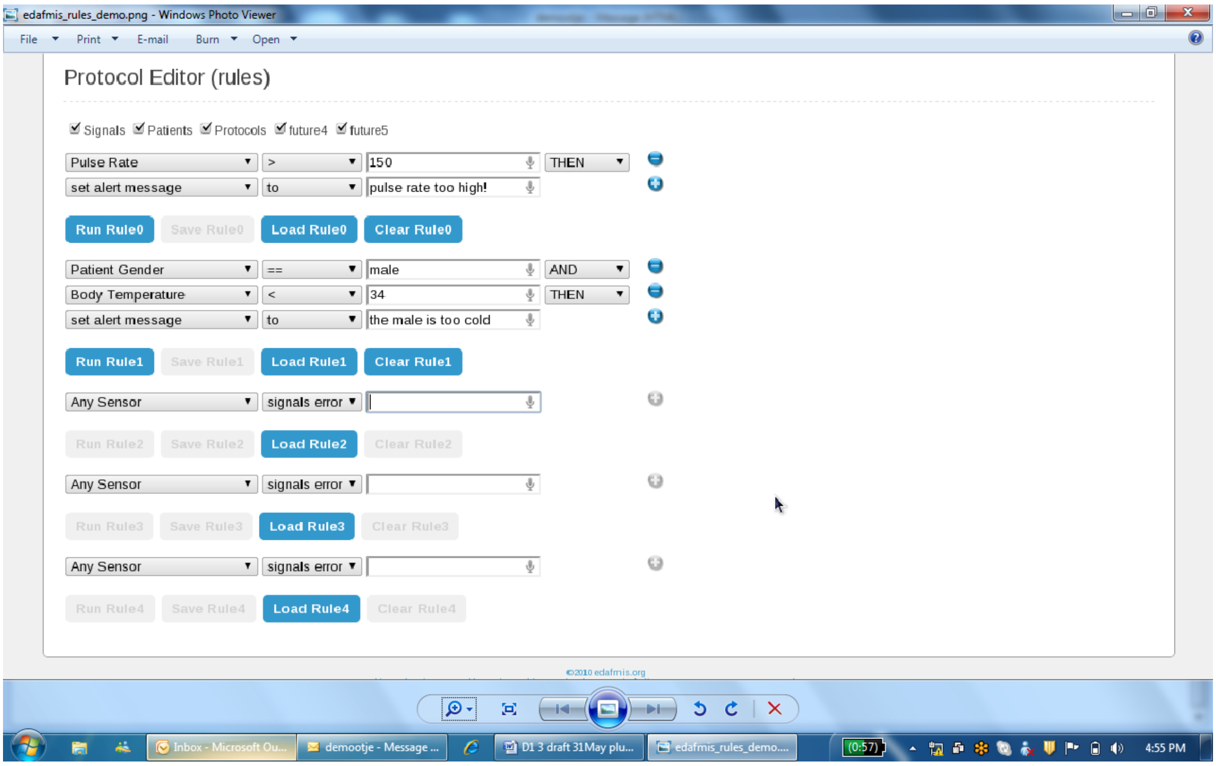
*Voorlichting voor (kandidaat) patiënten*

Accolade/XFlow

- ☑ Veiligheid
- ☑ Werkwijze
  - ☑ Diagnostiek
  - ☑ Voorbereiding en voorzorgsmaatregelen
  - ☑ Handeling
  - ☑ Nazorg
- ☑ Overlegsituaties
  - ☑ Communicatie met de patiënt/betrokkenen
- ☑ Opleiding
  - ☑ Methodiek
  - ☑ Observatie(schaal)
  - ☑ Bekwaamheidstoets
- ☑ Bronvermeldingen
- ☑ Achtergrondtheorie
- ☑ Studiemateriaal
- ☑ Formulieren
- ☑ Voorlichting voor (kandidaat) patiënten
  - ☑ Appendix

Free Title [Insert]
Text [Reference]

We also have developed another rule editor to make sure that it is more easy to use going forward. It is supported by a wizzard and specific for instructions how to interpret signals from the sensors in the operation room. See below:



Protocol Editor (rules)

☑ Signals ☑ Patients ☑ Protocols ☑ future4 ☑ future5

Pulse Rate | > | 150 | THEN
set alert message | to | pulse rate too high!

Run Rule0 | Save Rule0 | Load Rule0 | Clear Rule0

Patient Gender | == | male | AND
Body Temperature | < | 34 | THEN
set alert message | to | the male is too cold

Run Rule1 | Save Rule1 | Load Rule1 | Clear Rule1

Any Sensor | signals error |

Run Rule2 | Save Rule2 | Load Rule2 | Clear Rule2

Any Sensor | signals error |

Run Rule3 | Save Rule3 | Load Rule3 | Clear Rule3

Any Sensor | signals error |

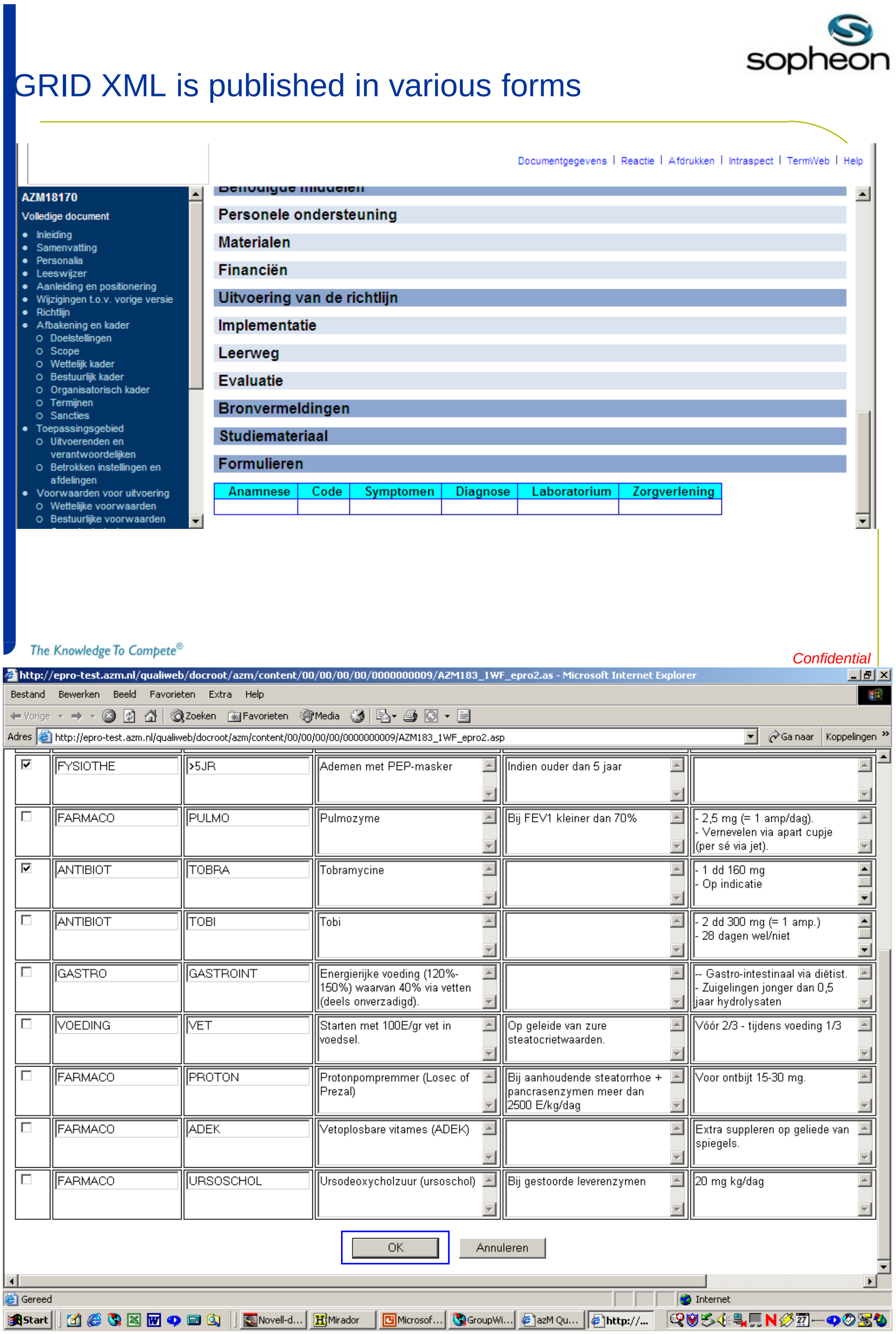
Run Rule4 | Save Rule4 | Load Rule4 | Clear Rule4

**Figure 10: Sopheon rule editor with regard to vital signs**

Of course the result of this editor can be saved as a specific deliverable in the Protocol Management system in Accolade.

## 2.2 *Publication of protocols*

Publishing is separated from the Protocol Management system. The publishing tool only knows "valid" and "published" protocols and allows users to search for them. It will not publish for instance protocols that are withdrawn or not published yet. The publishing tool needs to have 100% uptime, while the Protocol Management system is not mission critical but more back office.



GRID XML is published in various forms

| | | | | | |
|---|---|---|---|---|---|
| ☑ | FYSIOTHE | >5JR | Ademen met PEP-masker | Indien ouder dan 5 jaar | |
| □ | FARMACO | PULMO | Pulmozyme | Bij FEV1 kleiner dan 70% | - 2,5 mg (= 1 amp/dag).<br>- Vernevelen via apart cupje (per sé via jet). |
| ☑ | ANTIBIOT | TOBRA | Tobramycine | | - 1 dd 160 mg<br>- Op indicatie |
| □ | ANTIBIOT | TOBI | Tobi | | - 2 dd 300 mg (= 1 amp.)<br>- 28 dagen wel/niet |
| □ | GASTRO | GASTROINT | Energierijke voeding (120%-150%) waarvan 40% via vetten (deels onverzadigd). | | -- Gastro-intestinaal via diëtist.<br>- Zuigelingen jonger dan 0,5 jaar hydrolysaten |
| □ | VOEDING | VET | Starten met 100E/gr vet in voedsel. | Op geleide van zure steatocrietwaarden. | Vóór 2/3 - tijdens voeding 1/3 |
| □ | FARMACO | PROTON | Protonpompremmer (Losec of Prezal) | Bij aanhoudende steatorrhoe + pancrasenzymen meer dan 2500 E/kg/dag | Voor ontbijt 15-30 mg. |
| □ | FARMACO | ADEK | Vetoplosbare vitames (ADEK) | | Extra suppleren op geliede van spiegels. |
| □ | FARMACO | URSOSCHOL | Ursodeoxycholzuur (ursoschol) | Bij gestoorde leverenzymen | 20 mg kg/dag |

In the published protocol the GRID is displayed in form of a table, to be read and understood by doctors and nurses. The same content is also used by the DSS to do some useful calculations.

## 2.3 *The Use Case: VT Ablation Protocol*

*CeTIM / LIACS*

The project researched how to define a clinical pathway like VT Ablation based on Paradigm. We can see how this modelling method uses parameters to distinguish the elements of the pathway: the reference elements.

This chapter is based on a working paper entitled "Towards Flexibility in Computer-interpretable Guidelines: A VT-Ablation Pathway Example in Paradigm"[1], on adding flexibility to current computer-interpretable guidelines (CIG) approaches via the Paradigm coordination modeling language. While developing executable and sharable guidelines, the current approaches in literature discuss computerized pathway descriptions known as computer- interpretable guidelines (CIG). Based upon workflow notations, most of the respective languages underlying those systems use "Task-Network Models (TNMs)". While such notations usually allow creation of complete and automatically executable protocols, there is little support for collaboration and adaptation to exceptions on-the-fly. This is seriously hampering the adequate usage of decision-support systems (DSS) in realistic circumstances.

**Objectives**

In this chapter we want to contribute to the understanding of collaboration and adaptation to exceptions on-the-fly in notations for computer-interpretable guidelines (CIG) (Grando et al., 2010). We aim to bridge the gap between the existing medical pathways and ICT solutions, driven by highly agile coordination models embedded in medical guidelines as proposed in (Stettina, Groenewegen, & Katzy, 2011).

To improve our understanding of possibilities for integration, we have chosen to document our modelling process taking the example of a specific medical intervention: the ventricular tachycardia ablation. Cardiac ablation is a therapy method enabled by advances in technology, due to its minimally invasive origins the interventions requires special attention towards computer guided navigation. We thus develop an interaction/coordination model starting with a medical protocol towards a fully featured Paradigm model. By doing so we want to contribute to understanding of how to model medical processes.

**Paradigm**

As many systems today, medical equipment is affected by dynamic changes in its operational environment. Such systems cannot be simply shutdown to be changed, updated or upgraded and restarted again. This is particularly important for a live saving environment in which adaptation has to be done smoothly, quickly and without quiescence to support ongoing collaboration to meet clinical effectiveness. Adaptation on-the-fly to exceptions arising during ongoing collaboration, does hamper the adequate usage of decision-support systems (DSS) in realistic circumstances (Grando et al., 2010).

The name Paradigm is an abbreviation of PARallelism, its Analysis, Design and Implementation by a General Method (Andova, Groenewegen, & de Vink, 2010). The language has a strongly visual representation, analogous to other models such as those of UML. However, Paradigm is underpinned by precise mathematical constructs, constituting the formal definitions of its notions and their dependencies. On the basis thereof dynamic consistency between participants in a collaboration can be understood and analysed. As such, Paradigm consists of five basic notions to address coordination

[1] http://edafmis.org/Projects/530/Work%20Packages/Work%20Package%201/Related%20Documents/cardio06.pdf

of collaborating components: state transition diagrams, phases, (connecting) traps, roles and consistency rules.

Here we incorporate a brief explanation of Paradigm, using a small medical example. The example, although fictitious medically speaking, concentrates on the coordination of three different forms of collaboration within a medical team of three persons. Adaptation of the example will not be addressed as yet, as this would take too long for this document.

A state transition diagram (STD) is represented in Paradigm in a UML-like manner, as a simplified and strictly sequential statemachine; see Figure 11a with three highly similar STDs of one Clinician and two Nurses, performing a sequence of activities. Activity names serve as labels for the STD transitions. In view of performing some of the activities, additional regulations are in place, coordinating the ongoing behaviour of the Clinician and the two Nurses. The requirements for these regulations are as follows. Clinician, Nurse1 and Nurse2 can perform their activities enter, leave and . . . Act... in an independent manner, roughly speaking. But, (i) doTogether is to be performed simultaneously by the three; (ii) doPre, doMain and doPost are to be performed the one immediately after the other in that order; (iii) . . . DoXclsv are to be performed the one not immediately after the other and only after doPost has been performed; apart from that, the order is free.

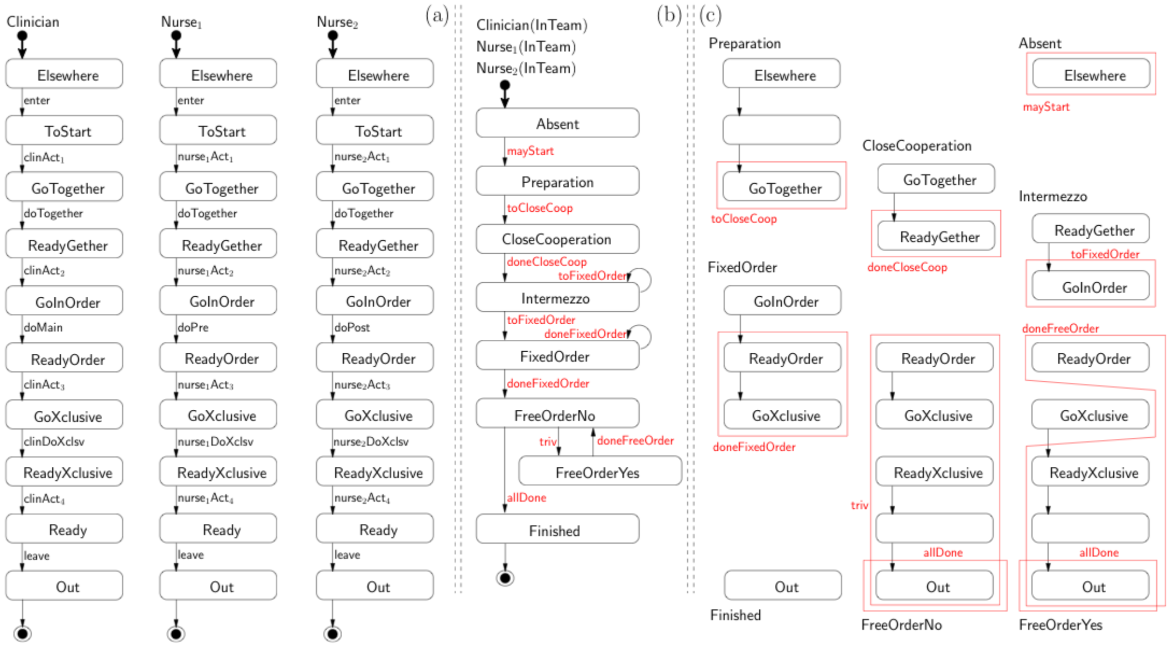


**Figure 11: (a) STDs Clinician, 2 Nurses; (b) their InTeam roles and (c) phases and traps**

Clinician, Nurse1 and Nurse2 each contribute to a collaboration via a specific InTeam role: Clinician(InTeam), Nurse1 (InTeam) and Nurse2 (InTeam), respectively. Figure 1(b) specifies the three InTeam) roles through a different STD each, whose states are so-called phases of the underlying original STDs Clinician, Nurse1 and Nurse2 : such phases are dynamic, i.e temporarily valid, constraints imposed on Clinician, Nurse1 and Nurse2 , respectively. The three role STDs in part (b) of the figure each mention eight phases: Absent, Preparation, CloseCooperation, Intermezzo, FixedOrder, FreeOrderNo, FreeOrderYes and Finished. Figure 11(c) couples Clinician, Nurse1 and Nurse2 each with their respective InTeam roles. It specifies each phase as part of Clinician, Nurse1 and Nurse2 , respectively; additionally, the phases are decorated with one or more polygons in red, grouping some states of a phase. Polygons visualise so-called traps: a trap, once entered, cannot be left as long as the phase remains the valid constraint. A trap having been entered, serves as condition for a phase transfer. Therefore, traps label transitions in a role STD, cf. Figure 11(b); if such phase transfer is to a different next phase, the trap moreover has to be connecting, i.e. each state within the trap also belongs to the next phase but there the former connecting trap can be left (is not necessarily a trap any longer).

Through synchronisation, singular steps from different roles are being coupled into one protocol step. If only one role step appears in the protocol step, as a matter of fact there is no synchronisation. It is through a consistency rule, Paradigm specifies a protocol step. In the example as presented here, we restrict the (Paradigm) consistency rules to those having the following format: (i) each rule starts with an $*$; (ii) the right-hand side of the $*$ lists one or more role steps being synchronised, separated by a comma; (iii) all role steps in such a list come from different roles. Technically this means, in our example all protocols are so-called choreographies (Andova, Groenewegen, & de Vink, 2010); so, here neither orchestrations occur nor any conductor driving one or more protocol steps; also, here no variable updates will be related to protocol steps in the form of a so-called change clause (Groenewegen, van Kampenhout & de Vink, 2005). Below is the complete set of consistency rules for the above example, specifying how Clinician, Nurse1 and Nurse2 can coordinate their activities while meeting the requirements. Immediately after the set of rules we present a visualisation of one of the many possible cooperation scenarios for Clinician, Nurse1 and Nurse2 as specified by the rules. On the basis of the visualisation we shall briefly clarify
the rule notation. There are 15 of these rules.

$$* \ \mathsf{Clinician(InTeam)}: \mathsf{Absent} \overset{mayStart}{\rightarrow} \mathsf{Preparation} \tag{1}$$

$$* \ \mathsf{Nurse}_1\mathsf{(InTeam)}: \mathsf{Absent} \overset{mayStart}{\rightarrow} \mathsf{Preparation} \tag{2}$$

$$* \ \mathsf{Nurse}_2\mathsf{(InTeam)}: \mathsf{Absent} \overset{mayStart}{\rightarrow} \mathsf{Preparation} \tag{3}$$

$$\begin{aligned} * \ & \mathsf{Clinician(InTeam)}: \mathsf{Preparation} \overset{toCloseCoop}{\rightarrow} \mathsf{CloseCooperation}, \\ & \mathsf{Nurse}_1\mathsf{(InTeam)}: \mathsf{Preparation} \overset{toCloseCoop}{\rightarrow} \mathsf{CloseCooperation}, \\ & \mathsf{Nurse}_2\mathsf{(InTeam)}: \mathsf{Preparation} \overset{toCloseCoop}{\rightarrow} \mathsf{CloseCooperation} \end{aligned} \tag{4}$$

$$\begin{aligned} * \ & \mathsf{Clinician(InTeam)}: \mathsf{CloseCooperation} \overset{doneCloseCoop}{\rightarrow} \mathsf{Intermezzo}, \\ & \mathsf{Nurse}_1\mathsf{(InTeam)}: \mathsf{CloseCooperation} \overset{doneCloseCoop}{\rightarrow} \mathsf{Intermezzo}, \\ & \mathsf{Nurse}_2\mathsf{(InTeam)}: \mathsf{CloseCooperation} \overset{doneCloseCoop}{\rightarrow} \mathsf{Intermezzo} \end{aligned} \tag{5}$$

$$\begin{aligned} * \ & \mathsf{Clinician(InTeam)}: \mathsf{Intermezzo} \overset{toFixedOrder}{\rightarrow} \mathsf{Intermezzo}, \\ & \mathsf{Nurse}_1\mathsf{(InTeam)}: \mathsf{Intermezzo} \overset{toFixedOrder}{\rightarrow} \mathsf{FixedOrder} \end{aligned} \tag{6}$$

$$\begin{aligned} * \ & \mathsf{Clinician(InTeam)}: \mathsf{Intermezzo} \overset{toFixedOrder}{\rightarrow} \mathsf{FixedOrder}, \\ & \mathsf{Nurse}_1\mathsf{(InTeam)}: \mathsf{FixedOrder} \overset{doneFixedOrder}{\rightarrow} \mathsf{FixedOrder} \end{aligned} \tag{7}$$

$$\begin{aligned} * \ & \mathsf{Clinician(InTeam)}: \mathsf{FixedOrder} \overset{doneFixedOrder}{\rightarrow} \mathsf{FixedOrder}, \\ & \mathsf{Nurse}_2\mathsf{(InTeam)}: \mathsf{Intermezzo} \overset{toFixedOrder}{\rightarrow} \mathsf{FixedOrder} \end{aligned} \tag{8}$$

$$\begin{aligned} * \ & \mathsf{Clinician(InTeam)}: \mathsf{FixedOrder} \overset{doneFixedOrder}{\rightarrow} \mathsf{FreeOrderYes}, \\ & \mathsf{Nurse}_1\mathsf{(InTeam)}: \mathsf{FixedOrder} \overset{doneFixedOrder}{\rightarrow} \mathsf{FreeOrderNo}, \\ & \mathsf{Nurse}_2\mathsf{(InTeam)}: \mathsf{FixedOrder} \overset{doneFixedOrder}{\rightarrow} \mathsf{FreeOrderNo} \end{aligned} \tag{9}$$

$$\begin{aligned} * \ & \mathsf{Clinician(InTeam)}: \mathsf{FreeOrderYes} \overset{doneFreeOrder}{\rightarrow} \mathsf{FreeOrderNo}, \\ & \mathsf{Nurse}_1\mathsf{(InTeam)}: \mathsf{FreeOrderNo} \overset{triv}{\rightarrow} \mathsf{FreeOrderYes}, \end{aligned} \tag{10}$$

$$\begin{aligned} * \ & \mathsf{Nurse}_1\mathsf{(InTeam)}: \mathsf{FreeOrderYes} \overset{doneFreeOrder}{\rightarrow} \mathsf{FreeOrderNo}, \\ & \mathsf{Nurse}_2\mathsf{(InTeam)}: \mathsf{FreeOrderNo} \overset{triv}{\rightarrow} \mathsf{FreeOrderYes}, \end{aligned} \tag{11}$$

$$\begin{aligned} * \ & \mathsf{Nurse}_2\mathsf{(InTeam)}: \mathsf{FreeOrderYes} \overset{doneFreeOrder}{\rightarrow} \mathsf{FreeOrderNo}, \\ & \mathsf{Clinician(InTeam)}: \mathsf{FreeOrderNo} \overset{triv}{\rightarrow} \mathsf{FreeOrderYes}, \end{aligned} \tag{12}$$

$$\begin{aligned} * \ & \mathsf{Clinician(InTeam)}: \mathsf{FreeOrderYes} \overset{allDone}{\rightarrow} \mathsf{Finished}, \\ & \mathsf{Nurse}_1\mathsf{(InTeam)}: \mathsf{FreeOrderNo} \overset{allDone}{\rightarrow} \mathsf{Finished}, \\ & \mathsf{Nurse}_2\mathsf{(InTeam)}: \mathsf{FreeOrderNo} \overset{allDone}{\rightarrow} \mathsf{Finished} \end{aligned} \tag{13}$$

$$\begin{aligned} * \ & \mathsf{Clinician(InTeam)}: \mathsf{FreeOrderNo} \overset{allDone}{\rightarrow} \mathsf{Finished}, \\ & \mathsf{Nurse}_1\mathsf{(InTeam)}: \mathsf{FreeOrderYes} \overset{allDone}{\rightarrow} \mathsf{Finished}, \\ & \mathsf{Nurse}_2\mathsf{(InTeam)}: \mathsf{FreeOrderNo} \overset{allDone}{\rightarrow} \mathsf{Finished} \end{aligned} \tag{14}$$

$$\begin{aligned} * \ & \mathsf{Clinician(InTeam)}: \mathsf{FreeOrderNo} \overset{allDone}{\rightarrow} \mathsf{Finished}, \\ & \mathsf{Nurse}_1\mathsf{(InTeam)}: \mathsf{FreeOrderNo} \overset{allDone}{\rightarrow} \mathsf{Finished}, \\ & \mathsf{Nurse}_2\mathsf{(InTeam)}: \mathsf{FreeOrderYes} \overset{allDone}{\rightarrow} \mathsf{Finished} \end{aligned} \tag{15}$$

Even without understanding their meaning, one might be able to recognise: behind the obligatory $*$, one or two or three role steps follow; each role step –of the format: phase, labeled arrow, next phase– is preceded by the name of the role and a colon. Thus, in the rules we see role names, phases and traps; per role the phases are pair-wise related by an arrow labeled by a (connecting) trap, suggesting the direction of the phase transfer specified.

Furthermore, we would like to point out the following idea underlying the 15 rules. Rules 1–3 coordinate the initialisation, all activities preceding doTogether. Rule 4 coordinates the first collaboration, activities doTogether. Rule 5 coordinates the closing if this first collaboration. Rules 6–8 coordinate the second collaboration, activities doPre, doMain and doPost. Rule 9 coordinates the closing of this second collaboration. Rules 10–12 coordinate the third collaboration, the activities . . . DoXclsv. Rules 13–15 coordinate the closing of this third and last collaboration up to finishing all activities.

On the basis of the three InTeam roles, Figures 12a and 12b specify an example scenario of the cooperation between Clinician, Nurse1 and Nurse2 in the style of a UML activity diagram. As such, a scenario shows a concrete sequence of protocol steps, thus visualizing a protocol as specified through suitable consistency rules. For each protocol step –i.e. at least one phase does change indeed– it is indicated which rule specifies the protocol step taken.

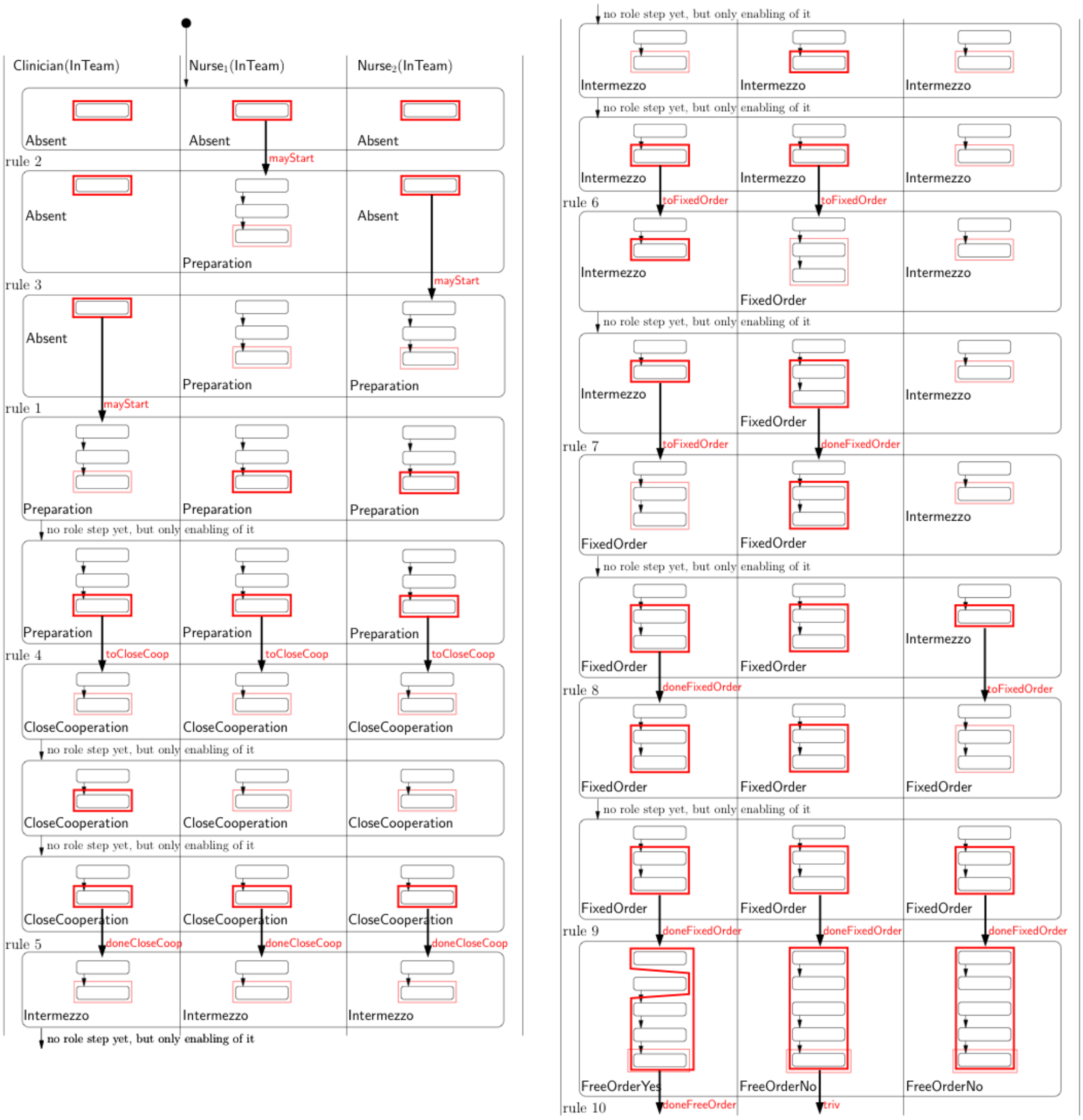


**Figure 12a: UML-like activity diagram for the three InTeam roles**

As a further, rather technical but interesting feature, one might recognise from Figure 12b how the so-called round robin strategy hidden in rules 10–12 works. (i) One can observe the circular application of rules 10, 11, 12, 10, . . . .. (ii) Where the circular application of rules 10–12 is interrupted by the additional step “no role step yet, but only enabling of it”, the enabling does take time as, the first time in this scenario, Nurse1 performs the nurse1 DoXclsv activity, the second time Clinician performs the clinDoXclsv activity and the third time Nurse2 performs the nurse2 DoXclsv activity. (iii) Later, after Clinician, Nurse1 , Nurse2 all three have entered their trap allDone, rule 15 is applied, thereby stopping the round robin strategy. Another detail here is, the round robin strategy gives the permission for performing the exclusive activity unconditionally –via trap triv– which means, even if not needed at that moment; but then, the permission cannot be used effectively, as continuing towards where it is actually needed, is temporarily blocked in (current) phase FreeOrderYes. This moreover results in, the permission is immediately withdrawn in favour of the next candidate, as –like-wise immediately– trap doneFreeOrder was entered.

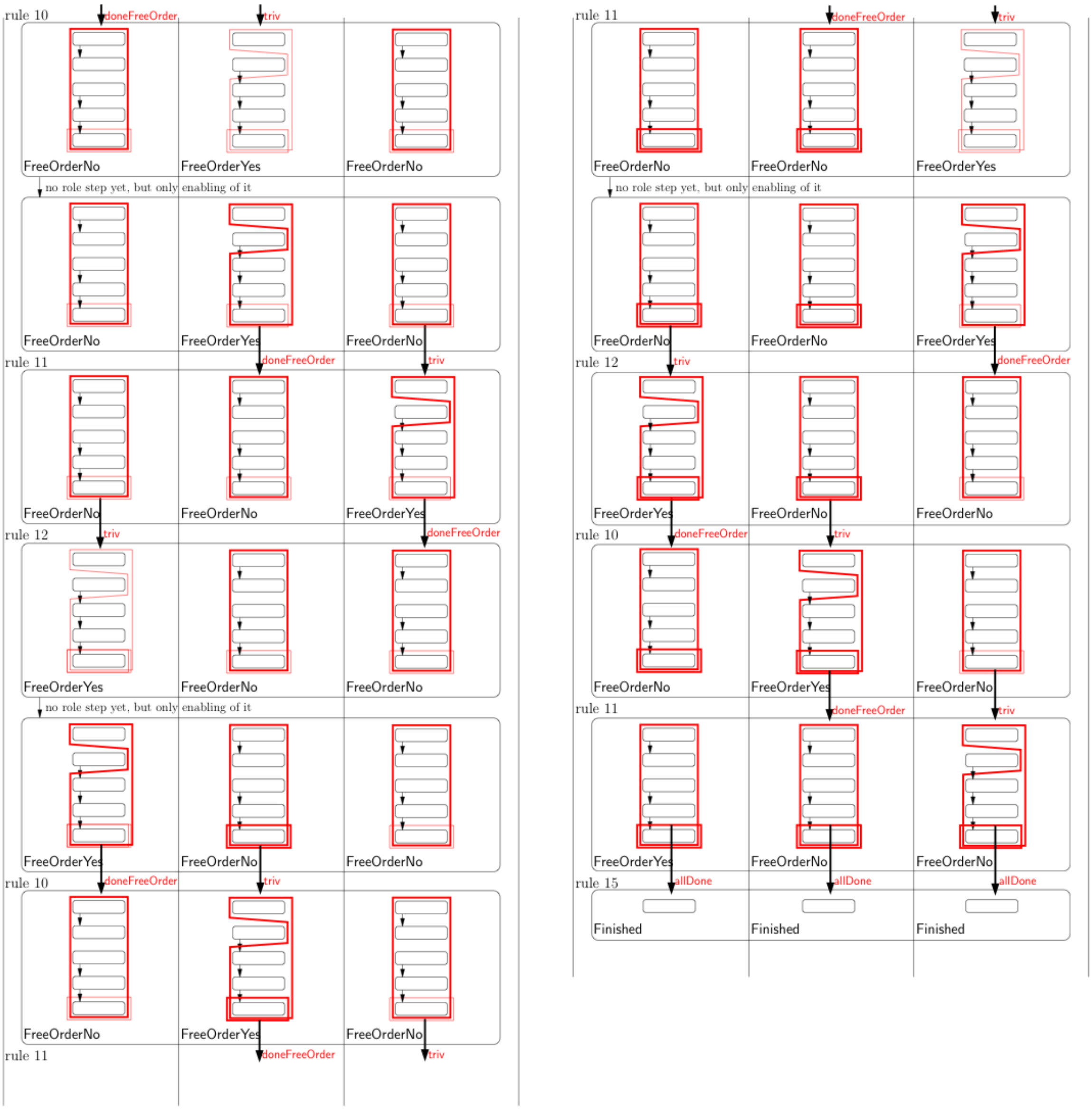


**Figure 12b: UML-like activity diagram for the three InTeam roles**

### Ventricular Tachycardia Protocol: Global Model

In the context of the EDAFMIS project we have received a VT ablation process description as supported by volumetric patient data. In this section we will discuss our approach in developing an intervention model from the initially received material.

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 (ICD) (Arya et al., 2009). The therapy is performed by a electrophysiologist who locally burns heart tissue on or around the locaton of the origin of the heart rhythm disorder. While the intervention can take up to 3-4 hours with traditional fluoroscopy, 3D imaging techniques based on CT data allow faster and more accurate interventions due to better spatial orientation.

**Stage 1: Step-wise Intervention - Single Swimlane**

The data as we initially received included discussion notes taken in the hospital, presentation slides, workflow descriptions and a flowchart diagram containing five roles in a threaded swim lane visualisation. To get a holistic view on the process in order to zoom into detail later on we decided to create a global single threaded model of the intervention first. See Figure 13.

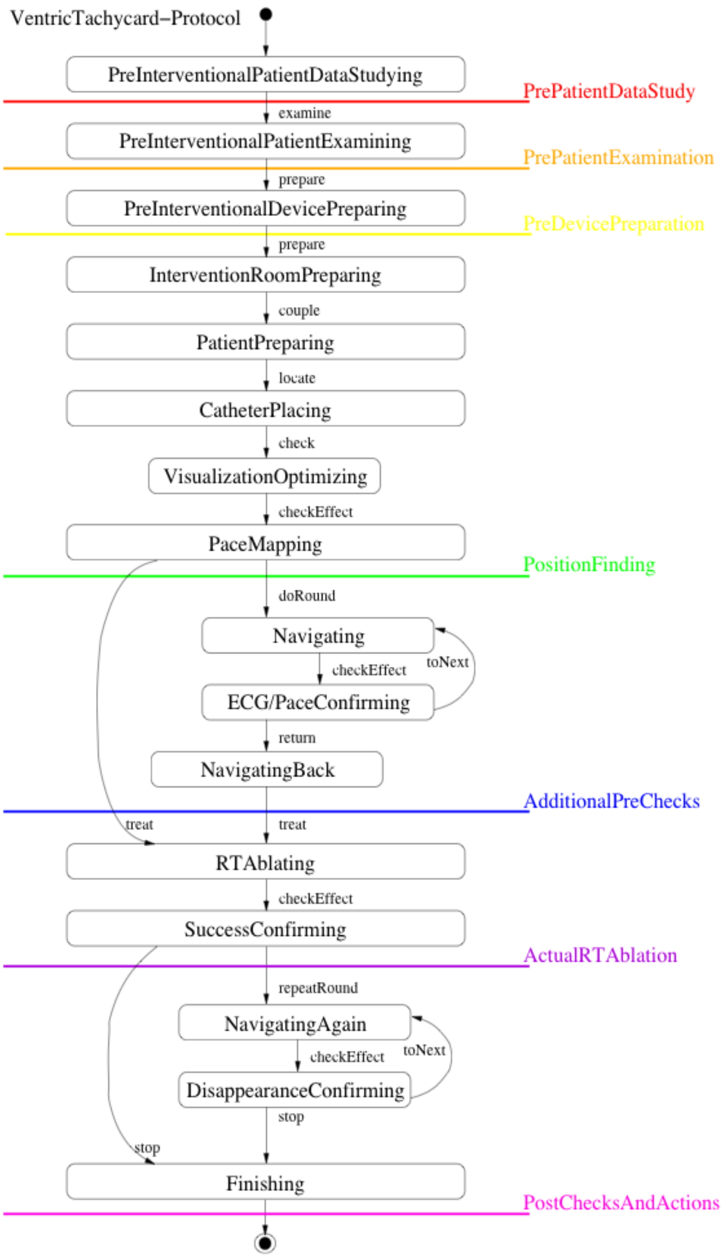


**Figure 13: Step-wise Intervention - Single Swimlane**

Before presenting some relevant examples of exception scenarios, we introduce an abbreviation of the intervention through Figure 14.

On the basis of the abbreviated form of the complete and uninterrupted intervention collaboration from Figure 14, Figure 15 (see appended paper) visualises an example scenario of such an exception.

Please note the following. It is via the AlarmWindow the ongoing intervention collaboration can be interrupted on the basis of emergency-like signals transmitted by a Sensor or ECG. This means, such interruption can occur only between AlarmWindow's actions switchOn and switchOff. Such interruption can have two consequences: Time-out, during which Patient should sufficiently recover, after which the intervention is to be resumed; Abandoning, stopping the actual intervention altogether, but not without some necessary finishing up depending on the intervention activities actually interrupted. Note, a third consequence might be seen in neither a Time-out nor Abandoning: just continuing; we prefer to take this as Time-out of zero duration, thus covering the third consequence too.

Please note, Figure 15's scenario is a combined example of one Time-out, after resuming followed by Abandoning. Other scenarios can be worked out in the same vein.

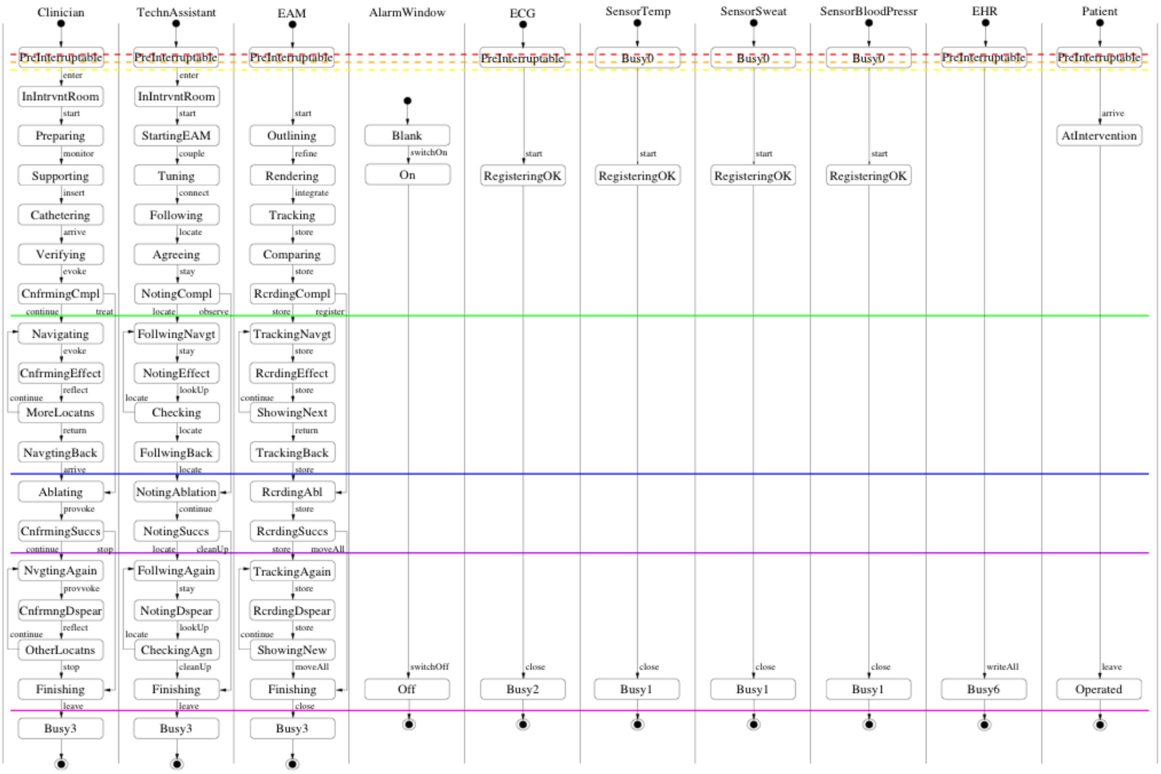


**Figure 14: Abbreviated collaboration, complete uninterrupted intervention**

## Conclusions

In this chapter we discuss our approach to modelling of computer-interpretable guidelines (GIG) (Grando et al., 2010) by using the notions of the coordination modelling language Paradigm. We document our iterative approach by modelling the intervention guideline of ventricular tachycardia ablation. We contribute an educational example to promote the use of the Paradigm language, to improve the modelling process for further medical procedures. We have positioned our approach at coordination of human and non-human actors thus allowing for integration of medical devices into the model.

# 3. Real-Time DSS

*Sopheon*

The DSS reads data from sensors (that are connected to the patient), the Protocol Management system and the EHR system. It sends messages to the EP navigator.

| type | entity | unit | abbr | origin | producer | target |
|---|---|---|---|---|---|---|
| sensor | ECG | millivolts | mV | Length: 8:28.928 (127232 sample intervals)<br>Sampling frequency:<br>250 Hz 1 signal | CU Ventricular Tachyarrhythmia Database (cudb) | |
| sensor | Pulse rate | beats per minute | bpm | derived from ECG | | |
| sensor | Pulse rhythm | regular/ irregular | | derived from ECG | | |
| sensor | Blood Pressure Systolic | Mercury | mmHG | simulated | | |
| sensor | Blood Pressure Diastolic | Mercury | mmHG | simulated | | |
| sensor | Temperature | degrees Celsius | °C | simulated | | |
| sensor | Respiration Rate | breaths per minute | | simulated/random | | |
| stream-in | Video | | | | Philips | |
| xml -in | Computer executable medical protocol | | | | Sopheon rule editor | |
| xml -in | Patient EHR data | | | | Zorggemak EHR dr. Halamka | |
| xml -out | Alerts/messages | | | | Sopheon DSS | Philips EP-navigator |

The DSS reads the data real time. It takes rules from the protocol and data from the EHR in order to calculate the conditions for alerting "messages".

# 4. EP Navigator and Intervention System

*Philips*

The EP Navigator is a separate system that has its own patient data set. In the future there should be integration with the EHR system. Today this is manual. The Navigation Imaging data are internal only to the EP Navigator system and are not used for integration yet.

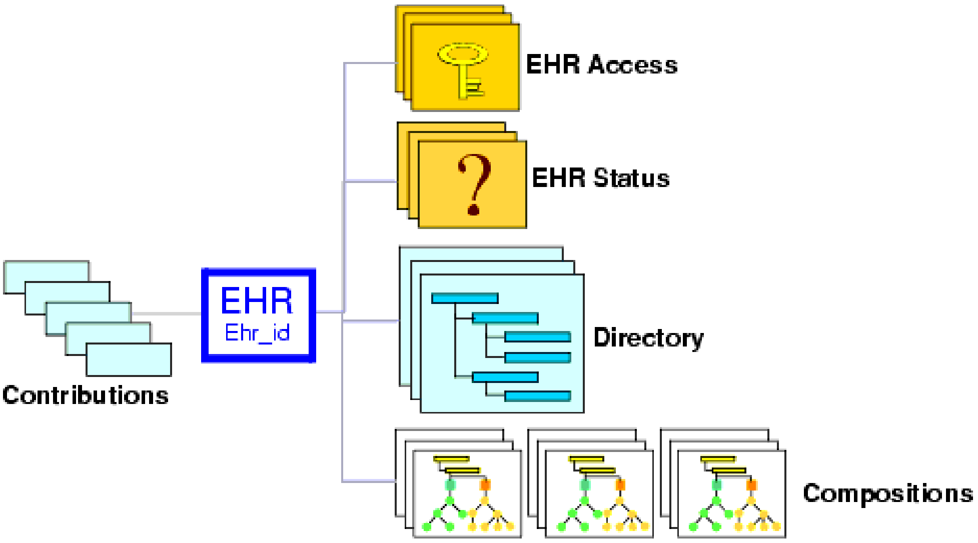
| Tag | Values | Attribute Name |
|---|---|---|
| (0010,0010) | (no value available) | PatientsName |
| (0010,0020) | [2231101570] | PatientID |
| (0010,0030) | [19000101] | PatientsBirthDate |
| (0010,0040) | [F] | PatientsSex |
| (0008,0050) | (no value available) | AccessionNumber |
| (0040,1001) | [R201009301454034] | RequestedProcedureID |

# 5. Virtual EHR System

*Zorggemak*

## *5.1 The openEHR EHR: coarse-grain level*

The openEHR EHR is structured according to a relatively simple model. A central EHR object identified by an EHR id specifies references to a number of types of structured, versioned information, plus a list of Contribution objects that act as audits of change-sets made to the EHR. The high-level structure of the *open*EHR EHR is shown in the figure below.



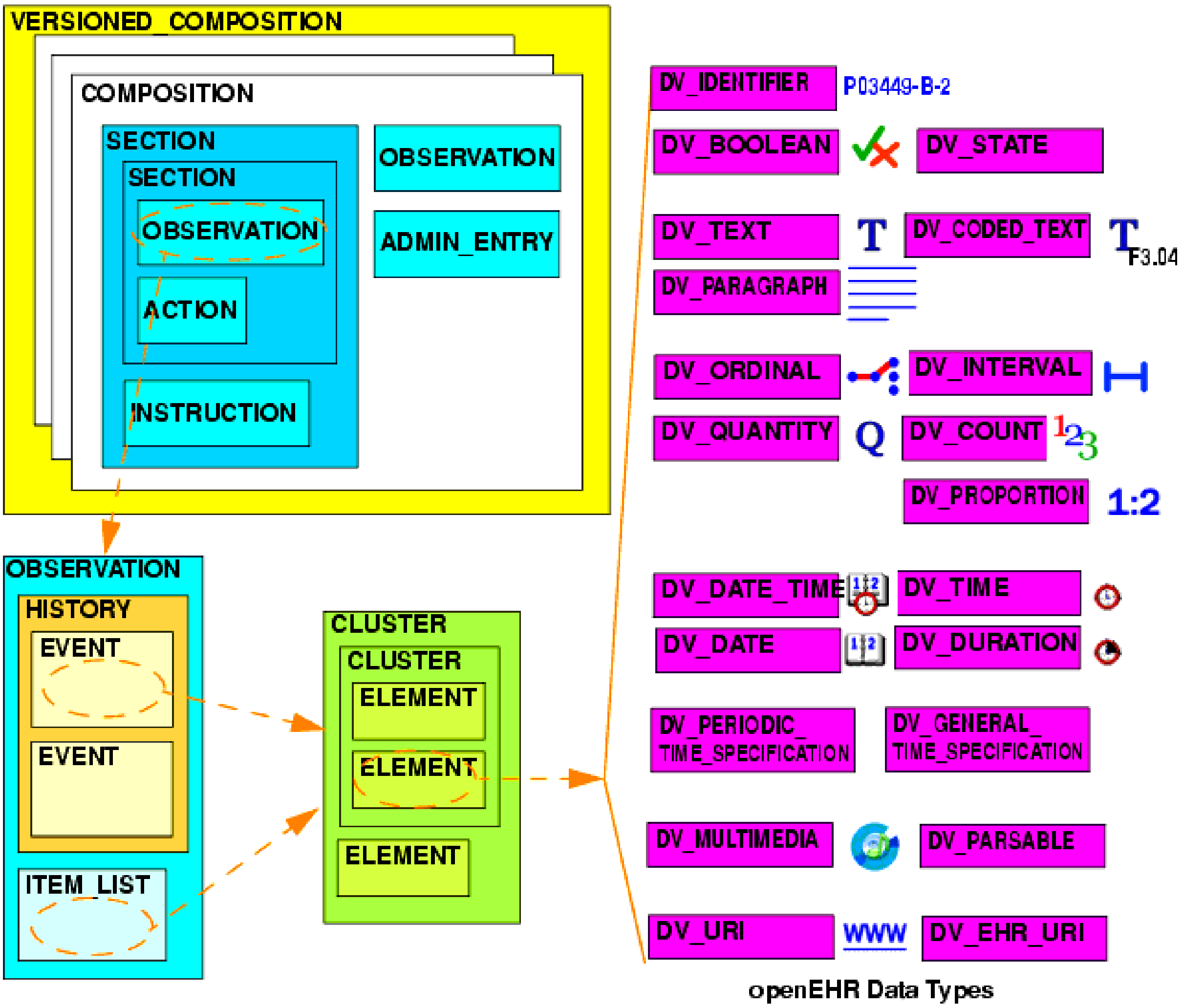
In this figure, the parts of the EHR are as follows:

- *EHR*: the root object, identified by a globally unique EHR identifier;
- *EHR_access (versioned)*: an object containing access control settings for the record;
- *EHR_status (versioned)*: an object containing various status and control information, optionally including the identifier of the subject (i.e. patient) currently associated with the record;
- *Directory (versioned)*: an optional hierarchical structure of Folders that can be used to logically organise Compositions;
- *Compositions (versioned)*: the containers of all clinical and administrative content of the record;
- *Contributions*: the change-set records for every change made to the health record; each Contribution references a set of one or more Versions of any of the versioned items in the record that were committed or attested together by a user to an EHR system.

The internal structure of the Composition along with the Directory object correspond closely to the levels in internationally agreed models of health information such as the CEN EN13606 and HL7 CDA standards.

## *5.2 The openEHR EHR: fine-grain level*

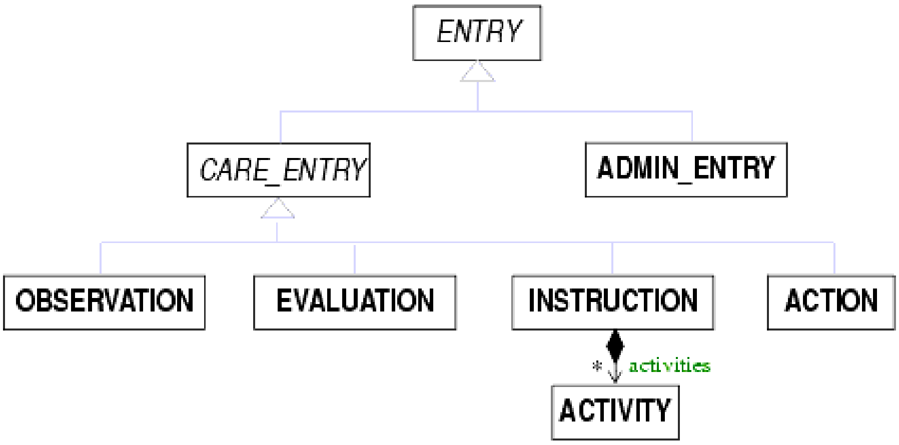
The logical structure of a typical Composition is shown in more detail in the next figure. In this figure, the various hierarchical levels from Composition to the data types are shown in a typical arrangement. The 21 data types provide for all types of data needed for clinical and administrative recording.



openEHR Data Types

All clinical information created in the openEHR EHR is ultimately expressed in "Entries". An Entry is logically a single `clinical statement', and may be a single short narrative phrase, but may also contain a significant amount of data, e.g. an entire microbiology result, a psychiatric examination note, a complex medication order. In terms of actual content, the Entry classes are the most important in the openEHR EHR Information Model, since they define the semantics of all the `hard' information in the record. They are intended to be archetyped, and in fact, archetypes for Entries and sub-parts of Entries make up the vast majority of archetypes defined for the EHR.

The openEHR ENTRY classes are shown in the figure below. There are five concrete subtypes: ADMIN_ENTRY, OBSERVATION, EVALUATION, INSTRUCTION and ACTION, of which the latter four are kinds of CARE_ENTRY.



The choice of these types is based on the clinical problem-solving process: a problem is solved by making observations, forming opinions (hypotheses), and prescribing actions (instructions) for next steps, which may be further investigation, or may be interventions designed to resolve the problem, and finally, executing the instructions (actions). This process may repeat itself as needed.

### *5.3 Integration with other systems*

Getting data in and out of the EHR is one of the most basic requirements openEHR aims to satisfy. In general, external or 'legacy' data (here the term is used for convenience, and does not imply anything about the age or quality of the systems in question) have different syntactic and semantic formats than openEHR data, and seamless conversion requires addressing both levels. Existing data sources and sinks include relational databases, HL7v2 messages, HL7 CDA documents and are likely to include CEN EN13606 data. Not all legacy systems are standardised; most hospital and GP products have their own private models of data and terminology usage.

In technical terms, a number of types of incompatibility have to be dealt with. There is no guarantee of correspondence of scope of incoming transactions and target openEHR structures – a single incoming document for example might correspond to a number of clinical archetypes. Structure will not usually correspond, with legacy data (particularly messages) usually having flatter structures than those defined in target archetypes. Terminology use is extremely variable in existing systems and messages, and also has to be dealt with. Data types will also not correspond directly, so that for example, a mapping between an incoming string "110/80 mmHg" and the target *open*EHR form of two DV_QUANTITY objects each with their own value and units has to be made.

### *5.4 Integration Archetypes*

The foundation of a key approach to the integration problem is the use of two kinds of archetypes. The "designed" archetypes, generally clinical, demographic or administrative. The common factors for all such archetypes are:

- they are based on the main part of the reference model, particularly the Entry subtypes OBSERVATION, EVALUATION, INSTRUCTION and ACTION;
- they are consciously designed from scratch by groups of domain specialists, and integrated into the existing library of *open*EHR archteypes;
- there is one archetype per identifiable health "concept", such as an observation type, person type etc.

A second category of archetypes is "integration" archetypes. These are characterised as follows:

- they are based on the same high-level types (COMPOSITION, SECTION etc), but use the sixth Entry subtype GENERIC_ENTRY;

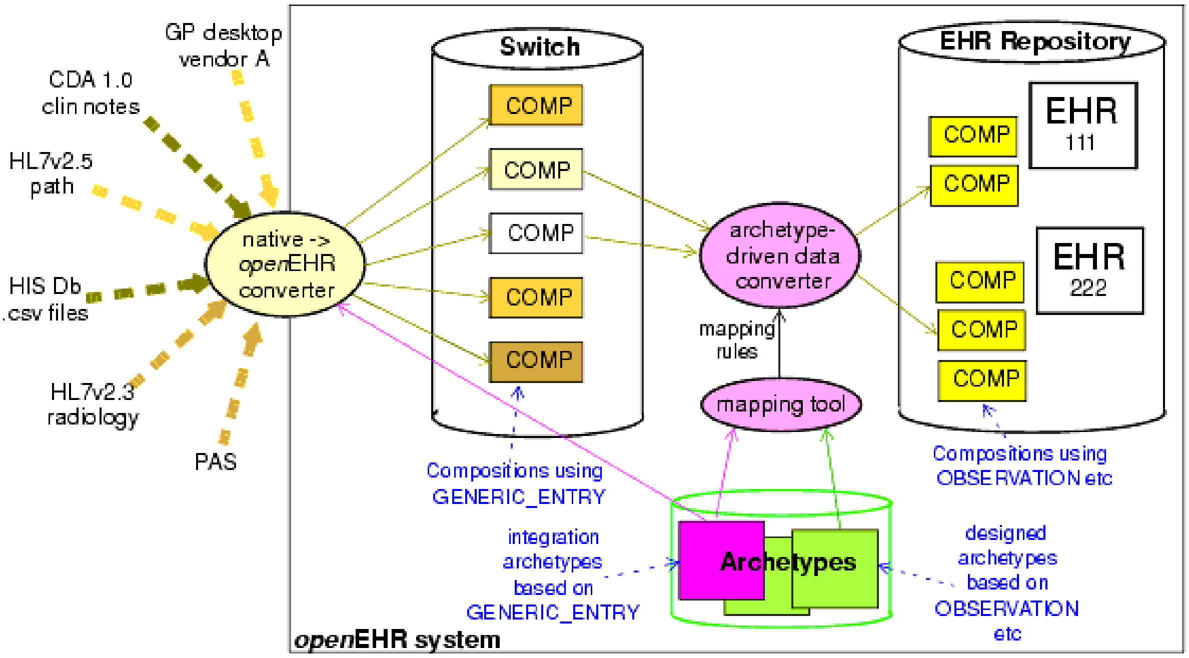
- they are designed to mimic the structure of legacy or existing data or messages; the design effort therefore is completely different, and is more likely to be done by IT or other technical staff who are familiar with the structures of the incoming data;
- there is one integration archetype per message type or identifiable source data that makes sense as a transaction to the EHR.

In the data integration environment, “designed” archetypes always define the target structures, coding and other semantics of data, while “integration” archetypes provide the means mapping of external data into the *open*EHR environment.

### *5.5 Data Conversion Architecture*

The integration archetype-based strategy for importing data into an openEHR system consists of two steps.

Firstly, data are converted from their original syntactic format into openEHR COMPOSITION/ SECTION/ GENERIC_ENTRY structures, shown in the openEHR integration switch. Most of the data will appear in the GENERIC_ENTRY part, controlled by an integration archetype designed to mimic the incoming structure (such as an HL7v2 lab message) as closely as possible. The result of this step is data that are expressed in the openEHR type system (i.e. as instances of the openEHR reference model), and are immediately amenable to processing with normal openEHR software.
In the second step, semantic transformation is effected, by the use of mappings between integration and designed archetypes. The mapping rules are the key to defining structural transformations, use of terminological codes, and other changes.


GP desktop vendor A
CDA 1.0 clin notes
HL7v2.5 path
HIS Db .csv files
HL7v2.3 radiology
PAS
native -> openEHR converter
Switch
COMP
COMP
COMP
COMP
COMP
Compositions using GENERIC_ENTRY
integration archetypes based on GENERIC_ENTRY
archetype-driven data converter
mapping rules
mapping tool
Archetypes
EHR Repository
COMP
COMP
EHR 111
COMP
COMP
COMP
EHR 222
Compositions using OBSERVATION etc
designed archetypes based on OBSERVATION etc
openEHR system

For the mapping tool, ZorgGemak will use an Expert System in conjunction with a Rules database (components 16 and 17 in figure 1). For example, when multiple legacy systems define the same data artifact, such as a patient’s address but with slightly differing details, the Expert System has to decide which data item is leading. The Expert System will in the future also be able to process signs.

Exporting information items from openEHR can be done in multiple ways; in Edafmis we propose to let the outside system (component 4 in figure 1) execute queries formed as ZorgGemak openEHR-Kernel’s API-calls, as is described in D2.2. One of the possible replies that the Kernel can send back is a GENERIC_ENTRY-based COMPOSITION as has been described in D1.2 for the case of patient's name, date of birth, weight and gender.

# 6. Conclusion

## 6.1 Integration and use of reference models

In the Edafmis practice the DSS is the material integration tool in the architecture.

1. It has APIs to the sensors, reads the data generated by the sensors real time,
2. It finds the patient record and sees the gender and age (it could see more parameters of course) of the patient, once the doctor has indicated the patient;
3. It also finds the protocol and the rules defined in there in the form of a software program, generated from the rules in XML format, again activated by the doctor. In the future we hope that the DSS can also track image information from the navigation system so that it can even more supportive of the doctor.
4. The doctor can also initiate another State (see Paradigm chapter); this initates a different set of rules, that will be used for calculation for a limited period of time. Then the system automatically resets for using the default rules.

## 6.2 Future

It is clear that still a lot of work needs to be done to get to a fully integrated system --- EHR, EP Navigation, Sensors and the DSS backed up by Protocol Management and Publishing. However, overseeing the components we can conclude that the various systems do have anchors that can be used for functional integration. In our project the integration role is given to the DSS . It has as an advantage that the reuse of data from various systems is functional from a diagnosis and treatment point of view. Functionally we can integrate without full technical integration, which is good from an IT perspective, but perhaps not necessary perse. Medical practice can make progress in this way, without waiting for massive integration projects.

## 7. References

S. Andova, L.P.J. Groenewegen, and E.P. de Vink. Dynamic consistency in process algebra: From Paradigm to ACP. *Science of Computer Programming*, 2010. To appear, 45pp.

A. Grando, M. Peleg, and D. Glasspool. A goal-oriented framework for specifying clinical guidelines and handling medical errors. *J. of Biomedical Informatics*, 43:287–299, April 2010.

L.P.J. Groenewegen, N. van Kampenhout, and E. de Vink. Delegation Modeling with Paradigm. In J.-M. Jacquet and G.P. Picco (eds.) *Proc. Coordination 2005*, LNCS 3453, pages 94-108, 2005.

C.J. Stettina, L.P.J. Groenewegen, and B.R. Katzy. (2011, January). Structuring medical agility. In V. Traver, A. Fred, J. Filipe, & H. Gamboa (eds.), *International conference on Health Informatics, HEALTHINF 2011,* pages 614–618. INSTICC Press, January 2011.

Arya, A., Piorkowski, C., Kircher, S., Sommer, P., Bollmann, A., Gaspar, T., et al. (2009). Results of recent studies in catheter ablation of ventricular tachycardias: in whom to abandon the icd? Herz , 34 (7), 539-44. Available from http://www.biomedsearch.com/nih/Results-recent-studies-in-catheter/20091253.html