openehr-kernel
Search results

OpenEHR Overview and Reference Model
Description of the OpenEHR Reference Model  more
www.rosa.nl

OpenEHR Service API
Description of the Zorggemak OpenEHR API  more
www.zorggemak.com

Zorggemak OpenEHR API Implementation
Information on how Zorggemak implemented the OpenEHR-API in context of the client-architecture  more
www.zorggemak.com
OpenEHR Overview
(two level modelling)

➔ Information Model
OpenEHR Overview
(two level modelling)

➔ Information Model
➔ Reference Model
OpenEHR Overview
(two level modelling)

- Information Model
- Reference Model
- Clinical Information
OpenEHR Overview
(two level modelling)

- Information Model
  - Reference Model
- Clinical Content Model
  - Archetypes
OpenEHR Overview
(ECO System)
OpenEHR Overview
(Structure of an EHR)

**FIGURE 14** High-level Structure of the openEHR EHR
OpenEHR Overview
(Structure of an EHR)

FIGURE 14 High-level Structure of the openEHR EHR

All Versioned
OpenEHR Overview

(Software Structure)

- Archetype svc
- Term svc
- EHR svc
- Demo. svc
- Template OM
- Archetype OM
- EHR
- Demographic
- Common
- Data types
- Data structures
- Basic types (Id, Terminology Access)
- General Purpose Datatypes (ISO 11404)
OpenEHR Overview
(Software Structure)

EHR
Folders
Composition
Sections
Entries

Clusters
Elements
Data values
OpenEHR Overview

(Software Structure)

EHR
Folders
Composition
Sections
Entries

Clusters
Elements
Data values
OpenEHR Overview
(Composition Structure)
OpenEHR Overview

(Information Level 1 & 2)
OpenEHR Overview
(Information Level 2 & 3)
OpenEHR Overview

(Information Level 2 & 3)
OpenEHR Overview (Archetypes)
OpenEHR Overview
(AQL (Archetype Query Language))

SELECT

o/data[at0001]/events[at0002]/time, 
   o/data[at0001]/events[at0002]/data[at0003]/items[at0013.1]/value

FROM

Ehr[uid=@EhrUid] CONTAINS Composition 
   c[openEHR-EHR-COMPOSITION.encounter.v1] 
   CONTAINS Observation o[openEHR-EHR-OBSERVATION.laboratory-lipids.v1]
OpenEHR Overview

• **Semantic operability in all software layers**
  
  *different pieces of software* know what the data mean

• **Patient centric view**
  
  patient-centric longitudinal EHR across enterprises, for decision support, care pathways, medical research

• **Easy adoption of standards, changes and complexity**
  
  build systems that keep up with reality

• **High level of re-use of artefacts**
  
  build once, re-use many times, archetypes, templates.

• **Low costs new software development**

• **Easy involvement in development of domain experts**
Zorggemak

The company I work for (freelance-base) is “Zorggemak”. One could translate it into “Easy Care”.

Zorggemak works together with companies like Philips, and universities in Italy, Spain and Israel and others.

The projects involve (for Zorggemak) implementation of the OpenEHR kernel, archetypes and user-interfaces in/during surgery, in another projects, cardiac monitoring, and easy accessible data-collections and monitoring in home-care situations.
Zorggemak

The simplicity of the OpenEHR kernel API as implemented by Zorggemak and the simplicity of software-development.

- GUI-controls are mapped/identified by archetype/template-path-identifiers
- The kernel accepts path-identifiers and corresponding values as representation of RM-objects and validates against archetype before storing.
- Values are stored in a database-structure holding less then ten tables.
- Storage is optimized for use of AQL and easy RM-instance retrieval and collection.
- EN13606 authorization-schemes will be supported.
- Terminologies from OpenEHR internal, and SNOMED will be implemented.