LiU EEE

An Educational EHR Environment
Based on openEHR & REST

http://www.imt.liu.se/mi/ehr/tools/

Presented by:
Erik Sundvall
erik.sundvall@liu.se
http://www.imt.liu.se/~erisu/

Department of Biomedical Engineering
Linköpings Universitet, Sweden
EEE

An Educational EHR Environment

Extendable EHR Ecosystem?

Exploding EHR Environment

(EHR = Electronic Health Record)
openEHR semantics & EEE

- openEHR semantics of data is well defined
  - but somewhat intertwined with semantics of operations on data due to object orientation

- openEHR semantics of operations on data
  - requirements on operations (e.g. versioning & contributions) are implicit in the object oriented data model
  - no service interface published as public specification yet

EEE aims to

- use REST for operations and to allow for several different syntax encodings of the openEHR data semantics ('representations' in REST terms)
- separate operations (OO functions) from the representations
Encourage implementation diversity while preserving semantic interoperability

- open specifications (e.g. openEHR) enables implementation technology diversity **between** interoperable systems (great!)
- EEE aims to simplify diversity and flexibility **within** a system deployment

- Not new or unique, also present in:
  - Upcoming openEHR service specifications for SOAP
  - Opereffa (Java)
  - OSHIP (Python, Java, Ruby, Lua, C++ ?)
  - GastrOS
  - Eiffel + .NET at commercial vendor?
  - Open EHR-Gen
  - ...

LiU
Why EEE?

We (IMT/LiU) needed an EHR system based on Archetypes etc (openEHR and/or ISO 13606) for education and research

- Modifiable & Open Source (e.g. OSHIP & Opereffa are also alternatives)
- Modular & easy to get started with
  - It was hard for new people to get started with openEHR projects in short time (e.g. master thesis projects or studies for a single paper).
  - "Pluggable" components with limited scope could help.
  - Integrate with other complete and partial implementations (any http-capable platform/programming language etc)
- Massively scalable? (Yet unproven...) It would be nice if EEE's architecture nicely scaled out to massively distributed deployments, including for nation-wide epidemiological purposes.
  - Slowness is a usability problem present in many EHRs
  - EEE aims to make use of REST's load reducing constructs.
Archetypes (chaining points)

ehr://1234567/87284370-2D4B-4e3d-A3F3-F303D2F4F34B@latest_trunk_version/content/openEHR-EHR-SECTION.vital_signs.v1/items/openEHR-EHR-OBSERVATION.heart_rate-pulse.v1/data/events[at0006]/data/items[at0004]/value/magnitude
Contributions
≈ transaction logging

Versioned objects
Contributions

FIGURE 8 Contributions to the EHR
Many ways to represent openEHR semantics

- Entered data internally mostly resembles a tree technically,
- but many prefer to think of it and present it as a structured document.
- The semantics of the reference model (RM) used is specified as an object oriented model.
- openEHR data can contain pointers to graph content such as terminology system codes.
Document - Traditions, clinical context, authorship etc. matter.

> EHR > Compositions > Sections > Entries > Data structures > Values

Tree - Paths are convenient for queries, processing etc.

> ehr://1234567/87284370-2D4B-4e3d-A3F3-F303D2F4F34B@latest_trunk_version/content[openEHR-EHR-SECTION.vital_signs.v1]/items[openEHR-EHR-OBSERVATION.heart_rate-pulse.v1]/data/events[at0006]/data/items[at0004]/value/magnitude

Objects - Easily implementable in IT systems (somewhat consistently)
storing openEHR data
but how?
Structured documents
- XML
- JSON
- YAML
- dADL
- ...

Objects / object graphs
- Object databases
- Network databases
- ...

Fine grained data atoms
- Mapped to Relational Database (SQL)
- RDF in a triple store / DB
- ...

Depends on major use case (single vs. multi), indexing etc

- Initial EEE implementation: XML documents in XML database with indexing. Why?
  - openEHR XML schemas exist (easy start)
  - XQuery is powerful for path-based queries
  - ...but other storage formats are also expected within an EEE ecosystem
# XML-database

<table>
<thead>
<tr>
<th>XML Database</th>
<th>License</th>
<th>Language</th>
<th>XQuery API</th>
<th>XML:DB API</th>
<th>RESTful API</th>
<th>Transaction Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache Xindice (no longer maintained)</td>
<td>Open source, free</td>
<td>Java</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>BaseX</td>
<td>Open source, free</td>
<td>Java</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Gemfire Enterprise</td>
<td>Commercial</td>
<td>Unknown</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>DOMSafeXML</td>
<td>Commercial</td>
<td>Unknown</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>eXist</td>
<td>Open source, free</td>
<td>Java</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MarkLogic Server</td>
<td>Commercial</td>
<td>C++</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MonetDB/XQuery</td>
<td>Open source, free</td>
<td>C++</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>myXMLDB</td>
<td>Open source, free</td>
<td>Java</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Unknown</td>
</tr>
<tr>
<td>OZONE</td>
<td>Open source, free</td>
<td>Java</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Sedna</td>
<td>Open source, free</td>
<td>C++</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Software AG’s Tamino</td>
<td>Commercial</td>
<td>Unknown</td>
<td>No</td>
<td>Partial</td>
<td>No</td>
<td>Unknown</td>
</tr>
<tr>
<td>TeXtML</td>
<td>Commercial</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Many more with xQuery support (and QQJ?) e.g. Oracle

REST

Representational State Transfer
REST (Representational State Transfer)

Web principles

- The terms "representational state transfer" and "REST" were introduced in 2000 in the doctoral dissertation of Roy Fielding, one of the principal authors of the Hypertext Transfer Protocol (HTTP) specification.

Predefined "verbs" (next slide)...

...act on "resources" identified e.g. by URIs (below)

Resources can return different "representations"

- E.g. xml (openEHR, RDF), html, json, protobuf, plain text, serialised java objects

```
scheme://user:pass@example.net:8080/path/to/file;type=foo?name=val#frag
\___/ \________\___________\________\________\________\________\___/
   |   |          |       |     |      |       |       |       |
   |___________/               (matrix)
   authority
```

Example: http://www.google.se/search?q=openehr&ie=utf-8&oe=utf-8
**REST verbs**

Predefined "verbs" act on "resources" (e.g. URI identifiers)

- A **GET** to an identifier requests a copy of the information in the supplied content type.
- A **PUT** to an identifier replaces the information. The supplied content type determines how it is to be interpreted.
- **POST** adds information.
- **DELETE** eliminates information.

<table>
<thead>
<tr>
<th>Database Operations</th>
<th>REST/HTTP Equivalents</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE</td>
<td>PUT</td>
</tr>
<tr>
<td>READ</td>
<td>GET</td>
</tr>
<tr>
<td>UPDATE</td>
<td>POST (update) or PUT (replace)</td>
</tr>
<tr>
<td>DELETE</td>
<td>DELETE</td>
</tr>
</tbody>
</table>
REST (Representational State Transfer)

REST Buzzwords: Low Entry-barrier, Extensibility, Distributed Hypermedia(?), Internet-scale...

EEE

openEHR + REST = true love?
EEE

Main Components

- Resources accessible via HTTP calls to URIs
- Services accessible via interfaces (mostly Java currently)
EEE main components

- Decision support
- Trigger handler (during entry)
- Validators & Converters
- Trigger handler (on commit)
- Export (e.g. replication)
- Decision support

EHR Database

- basic DB write
- basic DB read
- Native DB calls

Versioned objects (compositions etc)
Contributions
EHR Access control settings

Contribution builder

Contribution

Tools & utilities:
- Bulk loader
- Instance builder
- Log extractor
- Admin (users etc)

Client

Single record access router

Custom resources

Log

Multi record access router

Log

Query (DB native QLs)

Query translators

AQL-Query

Versioned object

Decision support

Export (e.g. replication)
Grouping of use-cases: single vs aggregate EHR access

Often different requirements & user categories
- permissions, logging, patient identity handling etc.
Easier to redirect to optimized databases & query engines.

**Single record access router**

**Multi record access router**

**URIs always contain .../ehr/{ehrId}/... e.g.**
http://localhost/ehr/1234000/56780007@latest_version
http://localhost/ehr/1234000/AQLQuery/...
DB access can be restricted to the EHR in the URI
Easy access logging & interpretation.

**URIs can look almost as above but queries may also access many records. Log analysis and access control may require analyzing query semantics e.g. http://localhost/multi/AQLQuery/...**
Example usage: Research & statistics
Ethical approval needed? Anonymization?
Adding data using Contribution Builder (1)

Purpose: Lowering entry-barrier, simplify development
- Allow incremental learning (also via examples).
- Detect misunderstandings & errors early.

Client

Single record access router

Contribution builder

EHR Database

- basic DB read
- basic DB write
- Versioned objects (compositions etc)

Contributions

Log

Contribution

Trigger handler (on commit)

Decision support

Export (e.g. replication)

Trigger handler (during entry)

Purpose: Lowering entry-barrier, simplify development
- Allow incremental learning (also via examples).
- Detect misunderstandings & errors early.

URIs contain .../contributionBuilder/{composerId}/{ehrId}/... e.g.
http://localhost/contributionBuilder/dr_who/1234567/generateContributionBuildID

The generated ID serves as a temporary "writing area" where e.g. compositions can be added(PUT), changed(POST), read(GET) or deleted(DELETE) under URIs like
/contributionBuilder/{composerId}/{ehrId}/{tempContributionId}/{tempObjectId}/...
Adding data using Contribution Builder (2)

Parameters regarding versioning etc can be given via the URI matrix and/or query

.../contributionBuilder/{composerId}/{ehrId}/{tempContributionId}/{tempObjectId};matix-variablenam1=value1;m-var2=val2?query-var3=value3&q4=v4..

Example:

.../contributionBuilder/dr_who/1234567/b734db36-30a7-43d6-b9ec-51cb389871b6/vital-signs;change_type=creation;object_type=COMPOSITION?lifecycle_state=incomplete...

Parameters can also be passed between resources via "request attributes" or be contained in the incoming representation (e.g. an XML or json document via POST)
Adding data using Contribution Builder (3)

Initial content (intended to later be modified e.g. by a GUI) can be loaded from the EHR:

`../contributionBuilder/{composerId}/{ehrId}/{contributionId}/medication-list/load=56780007::ehr.us.lio.se::2;change_type=modification;...`

Prototype "skeletons" (fake EHR content instances) can be created by administrators from Templates & Archetypes using the "Instance builder" and then be manually edited.

Content can also be loaded from a prototype document:

`../contributionBuilder/{composerId}/{ehrId}/{contributionId}/{tempObjectId}/load-prototype={prototype-uri};change_type=creation...`

optionally the prototype can be run through engines like Freemarker or Velocity

`../load-freemarker-prototype={prototype-uri};change_type=creation?optional-param1-to-prototype-engine=value1...`
Content can (e.g. during development) be run through specific validators & converters. (New openEHR Java ref-impl enables archetype based validation.)

Expected first (easy) EEE-formats: XML, json, DADL,

When the entire contribution build is finished it can be committed to the database

.../contributionBuilder/{composerId}/{ehrId}/{contributionId}/commit...

it will first be run through the validators & converters chosen by system administrators
Adding data using Contribution Builder (5)

A "rich/fat" client can have its own internal contribution builder and commit and entire contribution in one step.

Advanced clients can reuse the REST semantics and the EEE components to make development and maintenance easier.

- Less network traffic and server load. Faster decision support response in GUI etc.
- A client side Contribution builder can store data temporarily if network is unavailable (e.g. in mobile applications)
- A goal is to make the Contribution builder fully embeddable in e.g. Android (and partially in GWT (Google Web Toolkit) - probably not all validators/converters though...)

- Contribution
- Versioned objects (compositions etc)
- Export (e.g. replication)
- Trigger handler (on commit)
- Trigger handler (during entry)
- Decision support
- Log
- Single record access router
- Contribution builder
- Validators & Converters
- EHR Database
- Client
Commonly occurring scenario:
- A query produces a list containing hyperlinks. The query might come from a custom resource like http://localhost/ehr/1234000/MaternityOverview/ More on queries in later slides...
- Some of the hyperlinks pointing to versioned objects are followed fetching details, e.g. http://localhost/ehr/1234000/87284370-2D4B-4e3d-A3F3-F303::ehr.us.lio.se::2/
Likely scenario:
- Ad-hoc or stored queries produce lists or reports, sometimes containing hyperlinks.
- Some of the hyperlinks pointing to versioned objects can be followed fetching details, e.g. http://localhost/multi/ehr/1234000/87284370-2D4B-4e3d-A3F3-F303::ehr.us.lio.se::2/

Research & statistics.
Ethical approval needed? Anonymization?
Queries

*QL
SELECT e/data[at0001]/items[at0002.1]/value/defining_code/code_string
FROM EHR [uid = 121212-1212]
CONTAINS COMPOSITION c [openEHR-EHR-COMPOSITION.epicrisis.v1]
CONTAINS EVALUATION e [openEHR-EHR-EVALUATION.problem-diagnosis.v1]

SELECT c
FROM EHR [uid=$ehrUid]
CONTAINS COMPOSITION c
CONTAINS INSTRUCTION i [openEHR-EHR-INSTRUCTION.medication.v1]
CONTAINS ITEM_TREE it [openEHR-EHR-ITEM_TREE.medication.v1]
WHERE (it/items[at0012]/value/defining_code/terminology_id = "SNOMED"
AND it/items[at0012]/value/defining_code/code_string
matches {'350162003', '350162003'})

More info: http://www.openehr.org/wiki/display/spec/Archetype+Query+Language+Description
Embedded AQL-queries

How?

- Translate AQL to other query-languages, then run query
- Note: Chosen storage structures affects translation
- EEE's initial tool used: JavaCC
- EEE's initial target query language: XQuery
  - An EEE implementation storing data as RDF might instead translate AQL to SPARQL. The AQL result can also be embedded in a SPARQL query instead of in an XQuery.

Why?

- Use powerful query optimizers available in databases
- AQL does not specify any return format, but some query languages do have very flexible return formats
AQL embedded in xQuery

- XQuery results can be formatted and embedded in different file and document types
  - e.g. outputting results inside any XML dialect, Javascript or Adobe Flash

Examples

- HTML openEHR COMPOSITION list
- RDF Generation (DebugIT experiment by Daniel Karlsson)
- Google earth KML placemark generation + Adobe Flash
- HTML 5 + SVG based visualisations
- (Old proven) tab- or comma-separated lists
AQL embedded in xQuery

- XQuery results can be formatted and embedded in different file and document types
  - e.g. outputting results inside any XML dialect, Javascript or Adobe Flash
- Examples
  - HTML openEHR COMPOSITION list
  - RDF Generation (DebugIT experiment by Daniel Karlsson)
  - Google earth KML placemark generation + Adobe Flash
  - HTML 5 + SVG based visualisations
  - (Old proven) tab- or comma-separated lists

![AQL embedded in xQuery example](http://localhost:8182/googleh)

```
Date               Label                                      
2007-10-23T09:05:00 Besöksanteckning: vårdcentral : leg. läk. David Alvar  
2007-12-07T09:45:00 Besöksanteckning: vårdcentral : leg. läk. David Alvar  
2007-12-21T09:15:00 Besöksanteckning: vårdcentral : d.sk. Emma Skog       
2008-01-04T09:15:00 Besöksanteckning: vårdcentral : d.sk. Emma Skog       
2008-02-01T09:15:00 Besöksanteckning: vårdcentral : d.sk. Emma Skog       
2008-03-01T09:15:00 Besöksanteckning: vårdcentral : d.sk. Emma Skog       
2008-03-28T11:00:00 Besöksanteckning: hemmet : at Karin Rehn             
2008-06-30T13:25:00 Besöksanteckning: hemmet : at Karin Rehn             
2009-09-09T11:03:00 Epikris : neurologkliniken : leg. läk. Gun Erixsson    
```
AQL in xQuery in XHTML

Integration by pre-parsing (and later post-parsing) custom EEE-tags

All code below is valid xQuery, thus xQuery editors and validators can be used.

```xml
<h:html xmlns:h="http://www.w3.org/1999/xhtml"
        xmlns:eee="http://www.imt.liu.se/mi/ehr/2010/EEE-v1.xsd"
        xmlns="http://schemas.openehr.org/v1">
  <h:head><h:title>Composition list</h:title></h:head>
  <h:body>
    <h:table>
      <h:tr><h:th>Date</h:th><h:th>Label</h:th></h:tr>
      { let $aqlFiltered :=
        <eee:AQL>SELECT c FROM Ehr [uid=$ehrUid] CONTAINS COMPOSITION c</eee:AQL>
        for $version in $aqlFiltered/..
        let $comp := $version/data
        order by $comp/context/start_time/value/text()
        return <h:tr><h:td>{$comp/context/start_time/value/text()}</h:td><h:td>
          <h:a href="/ehr/AnnaTest-v3/{$version/uid/value}">
            {$comp/name/value/text()}: {$comp/context/setting/value/text()}: {$comp/composer/name/text()}
          </h:a></h:td></h:tr>
      }
    </h:table>
  </h:body>
</h:html>
```
AQL embedded in xQuery

XQuery results can be formatted and embedded in different file and document types

- e.g. outputting results inside any XML dialect, Javascript or Adobe Flash

Examples

- HTML openEHR COMPOSITION list
- RDF Generation (DebugIT experiment by Daniel Karlsson)
- Google earth KML placemark generation + Adobe Flash
- HTML 5 + SVG based visualisations
- (Old proven) tab- or comma-separated lists
EEE DebugIT converter experiment

- RDF-generating queries
- extract DebugIT-relevant data to triple store
- SPARQL queries executed by triple store
- Later: trigger based export

---

**Multi record access router**

- Query (DB native QLs)
  - AQL-Query
  - Query translators

- RDF triple store
- Log

---

**EHR Database**

- basic DB write
  - Versioned objects (compositions etc)
- basic DB read
  - Contributions
- Native DB calls
  - EHR Access control settings

---

**Client**

- Multi record access router
- Query (DB native QLs)
- AQL-Query
- Query translators
- Log

---

**Client**
AQL in xQuery in RDF

All code below is valid xQuery, thus xQuery editors and validators can be used. Custom EEE-tags delimit AQL.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xs="http://www.w3.org/2001/XMLSchema#"
    xmlns:ddoliu="https://lincoln.imt.liu.se:8443/vocab/resource/liu_ddo#">
{
    for $c in
    <eee:AQL>SELECT c FROM Ehr [uid=$ehrUid] CONTAINS COMPOSITION c</eee:AQL>
    return
    <rdf:Description rdf:about="/ehr/{$ehrUid}/{$c/../.uid/value}">
        <ddoliu:hasStartDateTime rdf:datatype="xs:dateTime">{data($c/context/start_time/value)}</ddoliu:hasStartDateTime>
    </rdf:Description>
}
</rdf:RDF>
```
AQL in xQuery in RDF - result

```xml
    xmlns:xs="http://www.w3.org/2001/XMLSchema#
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <rdf:Description rdf:about="/ehr/AnnaTest/4317453e-7f32-442a-a2e8-0fcaef11deed::test2.eee.mi.imt.liu.se::1">
        <ddoliu:hasStartDateTime rdf:datatype="xs:datetime">2008-03-30T10:25:00</ddoliu:hasStartDateTime>
    </rdf:Description>
    <rdf:Description rdf:about="/ehr/AnnaTest/7ef31236-3a94-4dea-a9c5-8321250dc8ae::test2.eee.mi.imt.liu.se::1">
        <ddoliu:hasStartDateTime rdf:datatype="xs:datetime">2008-04-03T11:45:00</ddoliu:hasStartDateTime>
    </rdf:Description>
    <rdf:Description rdf:about="/ehr/AnnaTest/625b730f-574c-45e7-9068-95b7b1deff20::test2.eee.mi.imt.liu.se::1">
        <ddoliu:hasStartDateTime rdf:datatype="xs:datetime">2008-06-02T09:15:00</ddoliu:hasStartDateTime>
    </rdf:Description>
    <rdf:Description rdf:about="/ehr/AnnaTest/1926354a-144d-4edd-aec0-2b0cf125acba::test2.eee.mi.imt.liu.se::1">
        <ddoliu:hasStartDateTime rdf:datatype="xs:datetime">2008-06-30T13:25:00</ddoliu:hasStartDateTime>
    </rdf:Description>
    <rdf:Description rdf:about="/ehr/AnnaTest/7e943f85-3b7a-43dd-8d40-b4a33df33b92::test2.eee.mi.imt.liu.se::1">
        <ddoliu:hasStartDateTime rdf:datatype="xs:datetime">2008-03-15T21:15:00</ddoliu:hasStartDateTime>
    </rdf:Description>

...results shortened here...
</rdf:RDF>
```
AQL embedded in xQuery

XQuery results can be formatted and embedded in different file and document types
- e.g. outputting results inside any XML dialect, Javascript or Adobe Flash

Examples
- HTML openEHR COMPOSITION list
- RDF Generation (DebugIT experiment by Daniel Karlsson)
- Google earth KML placemark generation + Adobe Flash
- HTML 5 + SVG based visualisations
- (Old proven) tab- or comma-separated lists
Decision support integration
Decision support integration

- In the GUI code (nice but hard to reuse between systems)
- At monitorable REST calls (reusable) via trigger handlers

  - ContributionBuilder interactions - no system wide visibility
    (To log or not to log?)
    - When? (configurable)
      - Continuous or...
      - ...when validation is requested

- Contribution interactions - system wide visibility, logged
  - At commit (incomplete, complete)
  - At attestation
**EEE data entry - Triggers, how? (1)**

- **Decision support**
- **Trigger handler (during entry)**
- **Validators & Converters**
- **Trigger handler (on commit)**
- **Export (e.g. replication)**
- **Decision support**

![Diagram](image.png)

- **Client**
  - Single record access router
  - Log
- **Contribution builder**
  - **Versioned object**
  - AQL-Query
- **Contribution**
  - **基本 DB write**
- **EHR Database**
  - **Versioned objects (compositions etc)**

**Trigger Handlers using e.g Apache ActiveMQ (not REST)**

- Allows subscription based on filters, e.g.
  - archetypes used, terminology codes used in data, EHR ID etc.
- **STOMP, JMS, XMPP, REST, WS-notification interfaces allowing clients using e.g.**
  - Java, C, C++, C#, Ruby, Perl, Python, PHP, ActionScript/Flash, Smalltalk
Trigger listeners can access further info via GET calls

- Classical example: Look up allergies & interactions during prescription
- To make Decision support modules reusable use pure AQL-Query and standard resources (e.g. Versioned object).
  Don't use queries containing native query languages (e.g. Xquery). Don't let listeners read via custom resources.
Distributed & scalable backends

Use case: population-wide queries
- Epidemiology, statistics, patient safety (GTT?)

Implemented e.g. using map-reduce
- A REST interface with query logging

Apache Hadoop (open source map-reduce framework)
- HBase, Pig Latin and other query languages

Amazon Elastic Compute Cloud (EC2)

Google App Engine (Bigtable, …)

Xadoop, xQuery for Hadoop might become a shortcut...
- http://www.xadoop.org/index.html
- alternative Hadoop language next to Pig Latin, Hive etc.
Why believe it can scale?

- **Semantics in URIs** (e.g. EHR id or versioned object ID) makes **sharding** (partitioning e.g. storage and load) between physical machines easy and straightforward.

- **OpenEHRs append-only** (or "never physically delete") principle combined with timestamped operations... + timewindow... makes **replication easier**

- Web deployments properly following REST principles have been shown to scale.

- But: The "**messaging part**" might not scale to "web size" (but hopefully "enterprize size") if the interactions chosen in setup are complex and intertwined.
openEHR

why bother?

(EHR = Electronic Health Record)
What is possible...
- For a computer system? For a human?
- For an organization? For multiple organizations?

Does it scale? Is it manageable? Man-hours? (Costs & competence)

Is data quality effected?
Archetypes, two layer modeling

Domain Content Models
archetypes & templates

Information Representation Models
openEHR reference model

• Change when new clinical needs arise.
• By Clinicians
  Medical (Informatics) knowledge?

• By IT system vendors etc.
  Computer Science knowledge?
• Implemented as software
  (in Java, .NET etc.)
• Storage, transactions etc.
• Stable
openEHR

in too short time

(Extra material not presented as part of EEE)
Encounter

Vital signs
- Blood pressure
  - Any event: den 25 maj 2008
  - Data: systolic mm[Hg], diastolic mm[Hg]

Pulse
- Any event: den 25 maj 2008

Findings

Respirations
- Any event: den 25 maj 2008

Blood gas assessment
- Any event: den 25 maj 2008

Arterial
- SaO2 %

Protocol
- Site of measurement

Template
(use case specific)

Archetypes
(reusable pieces of documentation patterns)
Archetype editing sidetrack.

**Concept & Original Author**
- **Concept**: Blood pressure
- **Long Concept**: The measurement by any means (invasive or non-invasive) of systemic arterial blood pressure which is deemed to represent the actual systemic blood pressure
- **Author**: Sam Heard
- **E-mail**: sam.heard@oceaninformatics.biz
- **Organisation**: Ocean Informatics
- **Date**: 22/03/2006

**Keywords**
- observations
- blood pressure measurement

**Usage & Purpose**
- **Use**: All blood pressure measurements are recorded using this archetype. There is a rich state model for use with exercise ECGs and Tilt Table measurements.
- **Misuse**: Not to be used for intravascular pressure.
- **Purpose**: To record the systemic blood pressure of a person. The measurement records the systolic and the diastolic...
Archetype editing sidetrack.
Archetype editing sidetrack.

### Term Definitions

<table>
<thead>
<tr>
<th>Code</th>
<th>Text</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>at0000</td>
<td>Blood pressure</td>
<td>the measurement by any means (invasive or non-invasive) of systemic arterial blood pressure which is deemed to represent...</td>
</tr>
<tr>
<td>at0001</td>
<td>history</td>
<td>history Structural node</td>
</tr>
<tr>
<td>at0002</td>
<td>baseline reading</td>
<td>baseline event in event history</td>
</tr>
<tr>
<td>at0003</td>
<td>blood pressure</td>
<td>@ internal @</td>
</tr>
<tr>
<td>at0004</td>
<td>systolic</td>
<td>the peak systemic arterial blood pressure over one cycle - measured in systolic or contraction phase of the heart cycle</td>
</tr>
<tr>
<td>at0005</td>
<td>diastolic</td>
<td>the minimum systemic arterial blood pressure over one cycle - measured in the diastolic or relaxation phase</td>
</tr>
<tr>
<td>at0006</td>
<td>any event</td>
<td>other event in event history</td>
</tr>
<tr>
<td>at0007</td>
<td>state structure</td>
<td>@ internal @</td>
</tr>
<tr>
<td>at0008</td>
<td>Position</td>
<td>The position of the patient at the time of measuring blood pressure</td>
</tr>
<tr>
<td>at0009</td>
<td>Exertion level</td>
<td>The level of exertion at the time of taking the measurement</td>
</tr>
<tr>
<td>at0010</td>
<td>Exercise</td>
<td>The classification of the exercise level</td>
</tr>
<tr>
<td>at0011</td>
<td>list structure</td>
<td>list structure</td>
</tr>
<tr>
<td>at0012</td>
<td>Instrument</td>
<td>the instrument used to measure the blood pressure</td>
</tr>
<tr>
<td>at0013</td>
<td>Cuff size</td>
<td>the size of the cuff if a sphygmomanometer is used</td>
</tr>
<tr>
<td>at0014</td>
<td>Location of measurement</td>
<td>The site of the measurement of the blood pressure</td>
</tr>
<tr>
<td>at0015</td>
<td>Adult</td>
<td>A cuff that is standard for an adult</td>
</tr>
<tr>
<td>at0016</td>
<td>Wide adult</td>
<td>A cuff for adults with larger arms</td>
</tr>
<tr>
<td>at0017</td>
<td>Paediatric</td>
<td>A cuff that is appropriate for a child or thin arm</td>
</tr>
<tr>
<td>at0022</td>
<td>At rest</td>
<td>The person is at rest and not in the recovery phase from exertion</td>
</tr>
<tr>
<td>at0023</td>
<td>Post-exercise</td>
<td>Measurement is taken immediately after exercise</td>
</tr>
<tr>
<td>at0024</td>
<td>During exercise</td>
<td>The measurement is taken during exercise</td>
</tr>
</tbody>
</table>
Archetype editing sidetrack.
Document - Traditions, clinical context, authorship etc. matter.

> EHR > Compositions > Sections > Entries > Data structures > Values

Tree - Paths are convenient for queries, processing etc.

> ehr://1234567/87284370-2D4B-4e3d-A3F3-F303D2F4F34B@latest_trunk_version/content[openEHR-EHR-SECTION.vital_signs.v1]/items[openEHR-EHR-OBSERVATION.heart_rate-pulse.v1]/data/events[at0006]/data/items[at0004]/value/magnitude

Objects - Easily implementable in IT systems (somewhat consistently)
FIGURE 30 How Archetypes apply to Data