An Open Source Record Locator Service Built With CDE Code

"Unintended Consequences of Technology Distribution, Picnic Day, UC Davis, 2005"

A First Generation Bench Test

California HealthCare Foundation

Oakland, California

August 3, 2007

[n]

MENDOCINO *informatics*
suite 206 - 216 west perkins street - ukiah, california 95482 - usa
[v] 707.462.6369 - [f] 707.462.5015 - www.minformatics.com

minformatics.com

OS-CDE Bench Test

July 6, 2007

CONTENTS

Project Team

Scott Duvall -- PhD Student, University of Utah

Shaun Grannis, MD -- Assistant Professor of Family Medicine, Indiana University School of Medicine

Richard Haley -- Network Engineer, minformatics.com

Jim Hazen -- Programmer, CareScience

Odysseas Pentakalos, PhD -- CTO, Sysnet International

Will Ross -- CIO, minformatics.com

Stuart Turner, DVM, MS -- Director, California Bioterrorism Laboratory Integration Project

Jeff Wallace -- Vice-President, EnvoyTek, Inc.

Summary

Overview of the First Generation OS-CDE Bench Test

The Open Source Care Data Exchange (OS-CDE) Bench Test is the serendipitous result of simultaneous events. In presentations at health care technology conferences in November 2006,[1] Jonah Frolich, Senior Program Officer at the California HealthCare Foundation[2] (CHCF), proposed an open source road map for the Care Data Exchange[3] (CDE) software. The CDE software had been developed as a proprietary software product[4] for CHCF during the preceding eight years by CareScience,[5] a health care technology vendor. In December 2006 Stuart Turner & Will Ross launched the OpenEMPI[6] (Open Enterprise Master Person Index) website, an online investigation to discover, catalog and evaluate open source identity matching software. As part of this investigation, Turner and Ross asked CHCF for permission to extract and publish the patient matching algorithm(s) from the CDE. Answering for CHCF, Frolich demurred, and instead challenged Turner and Ross to demonstrate an open source record locator service[7] (RLS) built with components from the legacy CDE software.

Accepting the challenge, Turner and Ross assembled a team of investigators for a rapid bench test of an explicitly open source fork of the CDE code. The investigators were invited to participate based on prior and concurrent discussions with either Turner or Ross (or both) on the topic of open source solutions for a full Master Person Index (MPI), or for a simple identity matching service. The OpenEMPI team launched the OS-CDE Bench Test with a two day planning meeting in Walnut Creek, California in February 2007. Weekly conference calls enabled the team to further define the RLS requirements, fork the CDE architecture, insert a demonstration patient matching algorithm, and build a workbench demo of the OS-CDE running as an explicitly open source solution. The OS-CDE solution was demonstrated to Jonah Frolich by Stuart Turner, Will Ross and Richard Haley on Friday May 11, 2007.

[1] November 10, 2006, Washington, DC and November 15, 2006, San Francisco, California.

[2] http://www.chcf.org

[3] http://www.wired.com/science/discoveries/news/2000/12/40433 -- see Appendix A

[4] "Proprietary" as used here indicates the presence of privately owned, licensed intellectual property within the CDE software. It would not be legal to distribute CDE code under an open source license without first resolving all current intellectual property restrictions, which is a core goal of the OS-CDE investigation.

[5] http://www.carescience.com

[6] http://www.openempi.org

[7] http://www.connectingforhealth.org/commonframework/docs/T6_RecordLocator.pdf

Thumbnail Project Plan

- Limit the technology investigation to the CDE identity correlation and record location services
- Remove internal package dependencies (e.g., CDE Information Architecture, Cloverleaf)
- Remove external dependencies (e.g., BEA WebLogic)
- Suppress excessively complex internal components (e.g., neural net for patient matching)
- Build on open source stack (e.g., Linux + PostgreSQL + JBoss)
- Replace legacy or proprietary dependencies with new open source services
- Postpone complexity, sensitivity and specificity while focusing on functionality of an RLS

Result of the OS-CDE Bench Test

The RLS built on the OS-CDE fork was demonstrated to CHCF on May 11, 2007 by Stuart Turner, Will Ross, Richard Haley and Jeff Wallace. This report completes the OS-CDE Bench Test, and provides recommendations for next steps prior to release of the code.

The OS-CDE demonstration is temporarily hosted by Mendocino Informatics at:

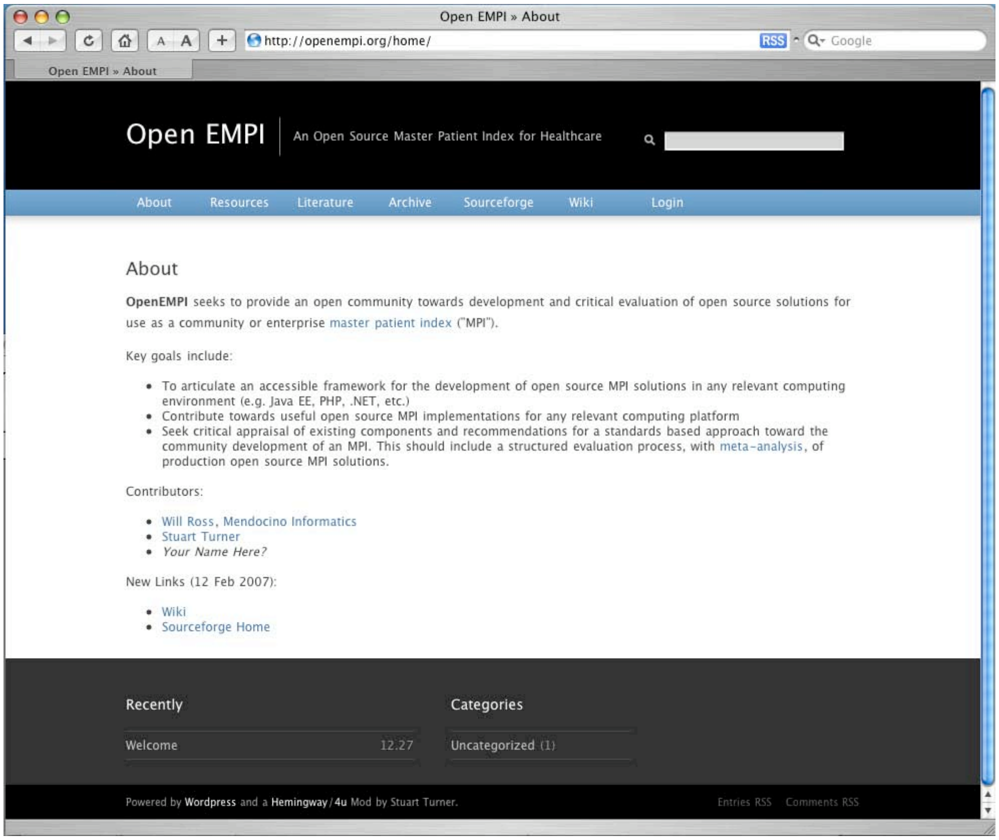
http://64.142.87.238:8080/openempi/

About OpenEMPI.org

OpenEMPI.org is intended to foster an open collaborative environment towards the development and critical evaluation of open source solutions for use as a community or enterprise master patient index ("MPI"). Key goals include:

- To articulate an accessible framework for the development of open source MPI solutions in any relevant computing environment (e.g. Java EE, PHP, .NET, etc.)
- Contribute towards useful open source MPI implementations for any relevant platform
- Seek critical appraisal of existing components and recommendations for a standards based approach toward the community development of an MPI. This should include a structured evaluation process, with cross-comparision or meta-analysis, of production open source MPI solutions.
- To identify, harmonize and develop normative business and technical requirements for an MPI and RLS within a number of emerging, but well defined use cases.

The OpenEMPI web site (see FIGURE 1) is located at http://www.openempi.org.



About

OpenEMPI seeks to provide an open community towards development and critical evaluation of open source solutions for use as a community or enterprise master patient index ("MPI").

Key goals include:

- To articulate an accessible framework for the development of open source MPI solutions in any relevant computing environment (e.g. Java EE, PHP, .NET, etc.)
- Contribute towards useful open source MPI implementations for any relevant computing platform
- Seek critical appraisal of existing components and recommendations for a standards based approach toward the community development of an MPI. This should include a structured evaluation process, with meta-analysis, of production open source MPI solutions.

Contributors:

- Will Ross, Mendocino Informatics
- Stuart Turner
- *Your Name Here?*

New Links (12 Feb 2007):

- Wiki
- Sourceforge Home

FIGURE 1 -- The OpenEMPI web site includes a blog framework, a list of resources, and a wiki.

OS-CDE Project

Pilot Test of an Open Source CDE Record Locator Service

The OS-CDE proposal was submitted to CHCF on February 8, 2007.[8] Specifically, Turner and Ross proposed a rapid prototyping project to fork the Care Data Exchange information architecture, and to pilot test a Record Locator Service (RLS) built with explicitly open source components. A small development team of J2EE programmers plus appropriate Subject Matter Experts was proposed. As a discovery based process, the OS-CDE proposal followed the roadmap suggested by Forrester Research[9] in their 2005 report[10] to CHCF. The Forrester schematic for the CDE information architecture is shown in FIGURE 2.

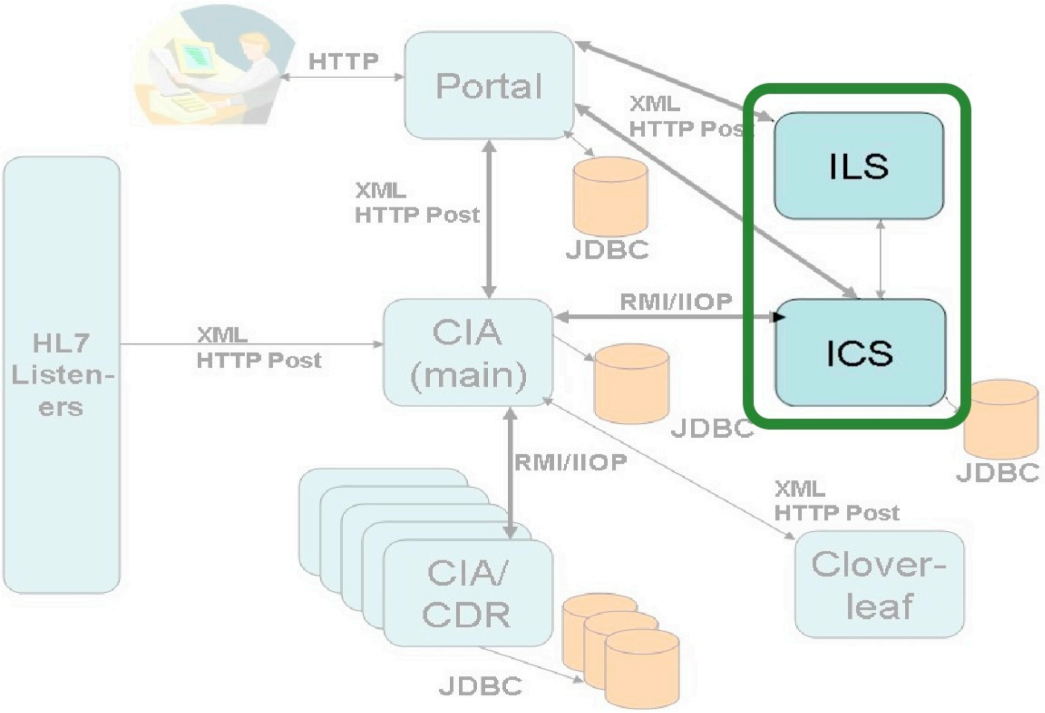


FIGURE 2 -- Forrester diagram of CDE Information Architecture, highlighting the two modules necessary for the OS-CDE fork to demonstrate an RLS.

"The project will focus on two CDE modules: the Identity Correlation Service (ICS) and the Information Locator Service (ILS). These modules perform the two key actions of an RLS: the ICS creates a master person index containing patient demographics and health

[8] OS-CDE Proposal submitted to CHCF in February 2007

[9] http://www.forrester.com

[10] "Care Data Exchange Technical Review" by Forrester Research, May 17, 2005

> record locations; and the ILS operates as a separate service to retrieve patient health records from known locations."[11]

Open Source Solution

An open source software solution was an essential goal of the OS-CDE bench test.

> "The primary deliverable from this project will be a pilot demonstration of an explicitly open source record locator service built from a combination of legacy CDE components and new open source code."[12]

Regarding the desired end state of an explicitly open source software solution, the proposal stated:

> "Any new code produced by this project will be released under an open source license to be specified by CHCF. … The final open source CDE code produced by this project will be donated to an appropriate hosting entity to be named by CHCF."[13]

On February 15, 2007 the OS-CDE proposal was approved by CHCF. However, the final licensing status of the OS-CDE code is unknown at this time.

OS-CDE Project Team

Co-Investigators

- Stuart Turner, DVM -- *UC Davis*
- Will Ross -- *Mendocino Informatics*

Team Members

- Scott Duvall -- *University of Utah*
- Shaun Grannis, MD -- *Indiana University*
- Richard Haley -- *Mendocino Informatics*
- Jim Hazen -- *CareScience*
- Dave Minch -- *John Muir Health System*
- Odysseas Pentakalos, PhD -- *Sysnet International, Inc.*
- Jeff Wallace -- *EnvoyTek, Inc.*

[11] "OS-CDE Proposal" *ibid.*

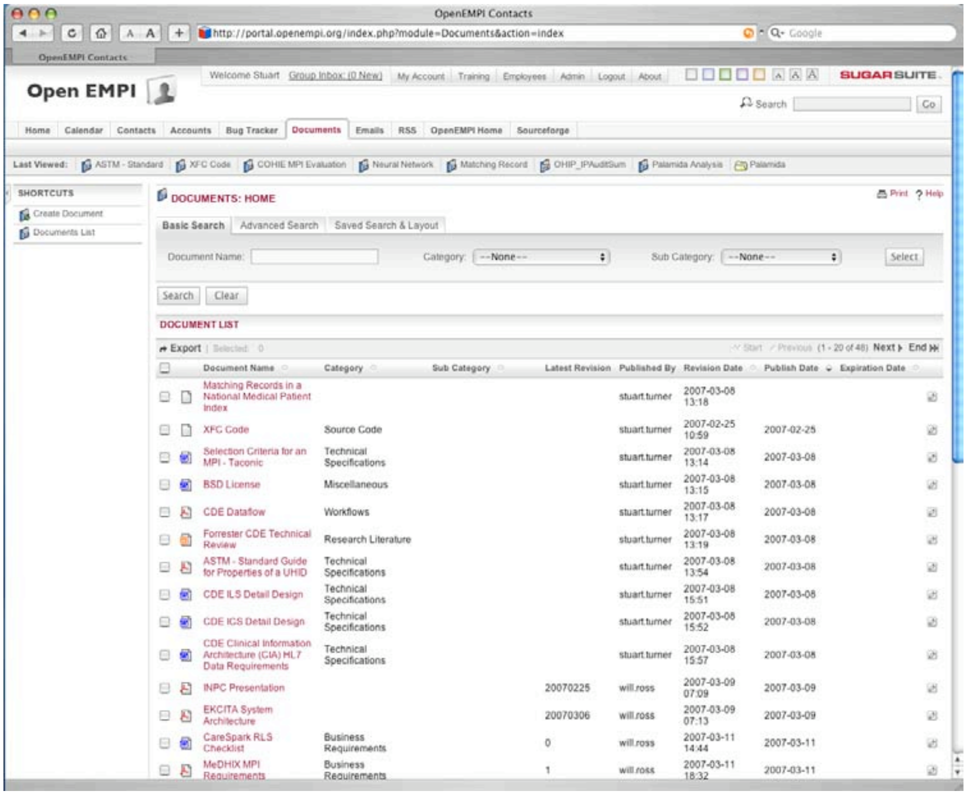
[12] "OS-CDE Proposal" *ibid.*

[13] "OS-CDE Proposal" *ibid.*

OS-CDE Collaboration Tools

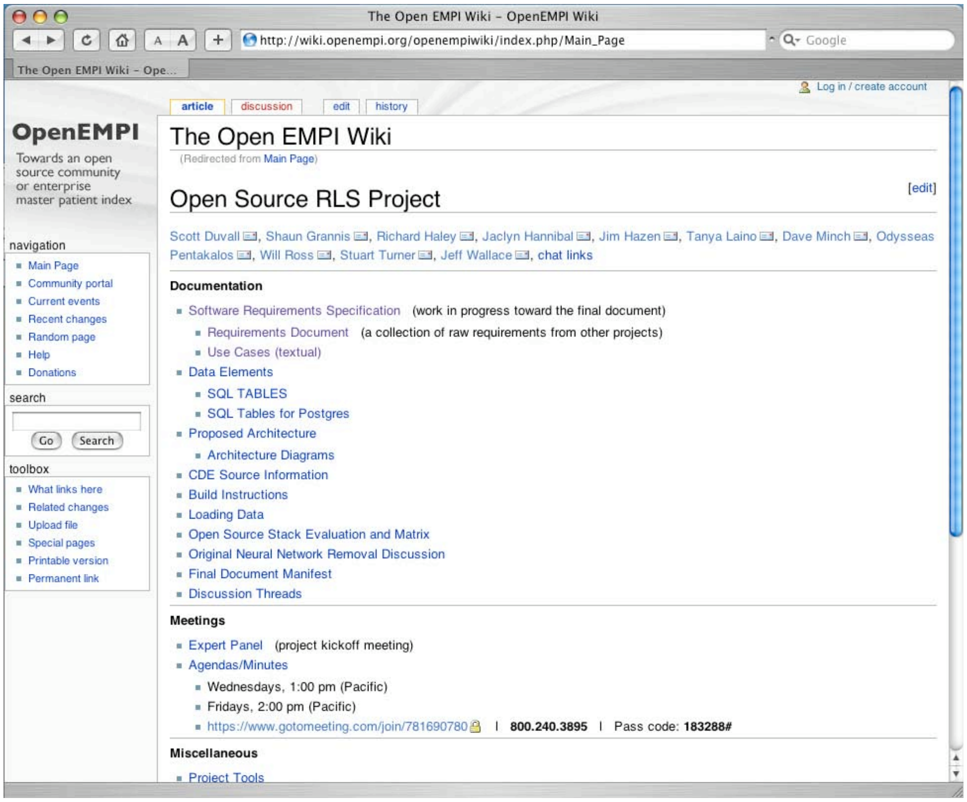
The project deployed a full suite of online collaboration tools.

- Project Portal (see FIGURE 3)
- Project Wiki (see FIGURE 4)
- Ftp Server
- Subversion Repository
- Project mailing list



Document Name	Category	Sub Category	Latest Revision	Published By	Revision Date	Publish Date	Expiration Date
Matching Records in a National Medical Patient Index				stuart.turner	2007-03-08 13:18		
XFC Code	Source Code			stuart.turner	2007-02-25 10:59	2007-02-25	
Selection Criteria for an MPI - Taconic	Technical Specifications			stuart.turner	2007-03-08 13:14	2007-03-08	
BSD License	Miscellaneous			stuart.turner	2007-03-08 13:15	2007-03-08	
CDE Dataflow	Workflows			stuart.turner	2007-03-08 13:17	2007-03-08	
Forrester CDE Technical Review	Research Literature			stuart.turner	2007-03-08 13:19	2007-03-08	
ASTM - Standard Guide for Properties of a UHID	Technical Specifications			stuart.turner	2007-03-08 13:54	2007-03-08	
CDE ILS Detail Design	Technical Specifications			stuart.turner	2007-03-08 15:51	2007-03-08	
CDE ICS Detail Design	Technical Specifications			stuart.turner	2007-03-08 15:52	2007-03-08	
CDE Clinical Information Architecture (CIA) HL7 Data Requirements	Technical Specifications			stuart.turner	2007-03-08 15:57	2007-03-08	
INPC Presentation			20070225	will.ross	2007-03-09 07:09	2007-03-09	
EKCITA System Architecture			20070306	will.ross	2007-03-09 07:13	2007-03-09	
CareSpark RLS Checklist	Business Requirements		0	will.ross	2007-03-11 14:44	2007-03-11	
MeDHIX MPI Requirements	Business Requirements		1	will.ross	2007-03-11 18:32	2007-03-11	

FIGURE 3 -- portal.openempi.org, built with SugarCRM



The Open EMPI Wiki

(Redirected from Main Page)

Open Source RLS Project [edit]

Scott Duvall, Shaun Grannis, Richard Haley, Jaclyn Hannibal, Jim Hazen, Tanya Laino, Dave Minch, Odysseas Pentakalos, Will Ross, Stuart Turner, Jeff Wallace, chat links

Documentation

- Software Requirements Specification (work in progress toward the final document)
  - Requirements Document (a collection of raw requirements from other projects)
  - Use Cases (textual)
- Data Elements
  - SQL TABLES
  - SQL Tables for Postgres
- Proposed Architecture
  - Architecture Diagrams
- CDE Source Information
- Build Instructions
- Loading Data
- Open Source Stack Evaluation and Matrix
- Original Neural Network Removal Discussion
- Final Document Manifest
- Discussion Threads

Meetings

- Expert Panel (project kickoff meeting)
- Agendas/Minutes
  - Wednesdays, 1:00 pm (Pacific)
  - Fridays, 2:00 pm (Pacific)
  - https://www.gotomeeting.com/join/781690780 | **800.240.3895** | Pass code: **183288#**

Miscellaneous

- Project Tools

FIGURE 4 -- wiki.openempi.org, built with MediaWiki

Expert Panel

Team members met for two days[14] in Walnut Creek to launch the OS-CDE project. The first day was generously hosted by the John Muir Health System. As a frame of reference the meeting began with an overview of the Indiana Network for Patient Care (INPC), presented by Shaun Grannis, MD.[15] The INPC is a leading health informatics implementation, with nine million patients in their Master Person Index. After lunch an introduction to the CDE code[16] was presented by Jim Hazen, the primary CareScience developer.

[14] March 9-10, 2007

[15] See Appendix D, "INPC Notes"

[16] See Appendix E, "CDE Overview"

FIGURE 5 -- OS-CDE team on Friday March 9th at John Muir Health System: (l-r) Dave Minch, Will Ross, Stuart Turner, Jim Hazen, Odysseas Pentakalos, Shaun Grannis, Jeff Wallace, Scott Duvall. Photo by Richard Haley.

On Saturday the team reviewed next steps for a fast path forward, creating an aggressive eight week project calendar. Several key project tasks were identified.

- Draft harmonized RHIO requirements for an RLS
- Complete the Palamida IP analysis
- Modularize the OS-CDE build with abstraction layers between major functions (e.g., portal, database, matching algorithm, federated search, etc.)
- Fork the CDE code to allow elimination of non Open Source IP
- Replace the proprietary strong comparator with a simple rules engine to demonstrate modularity for the OC-CDE fork
- Build demonstration server on a new open source stack (Linux + JBoss + PostgreSQL)

Weekly OS-CDE Iterations

Following the kickoff meeting, the team used the project collaboration tools combined with regular conference call meetings to track progress. Weekly calls were held on:

- Wednesdays at 1:00 PM (Pacific)
- Fridays at 2:00 PM (Pacific)

Extra meetings were scheduled as needed. The project wiki was utilized to prepare meeting agendas and minutes. The project mailing list handled discussion threads between meetings.

Palamida IP Analysis

Palamida, Inc.[17] is an IT services company that evaluates software code for legal issues, focusing especially on potential conflicts between differing intellectual property (IP) licenses contained in the code. Palamida generously donated to the OS-CDE project a screening and assessment of the ICS and ILS packages from the CDE code to determine IP license obligations and potential IP compliance issues. A summary of findings by Palamida,[18] and the OS-CDE response, listed in Table 1, was discussed with Palamida staff on the regular OS-CDE project status conference call on April 4, 2007.

Palamida Result	IP Issue	OS-CDE Result
1 "Severity One" issue	GPL package called by the ICS	Package discarded
13 "Severity Two" issues	Non-GPL OS licenses	Not incompatible if OS-CDE published under a non-GPL license
13 "Severity Three" issues	Non-GPL OS licenses	Not incompatible if OS-CDE published under a non-GPL license
Numerous issues	Proprietary code	All proprietary code dependencies eliminated from OS-CDE
Investigate IP assertions by LANL	Potential inclusion of IP from LANL project	No LANL or University of California copyright assertions found in CDE

TABLE 1 -- Palamida's automated evaluation of IP risk confirmed the manual evaluation of OS licensing issues conducted by OS-CDE team members. Although no substantial new IP risks were identified, the project benefited from Palamida's impartial analysis. It is also appropriate for CHCF initiate a subsequent IP audit by Palamida to evaluate the final code delivered by the OS-CDE project.

For years the proprietary CDE code has been dogged by assertions of the potential inclusion of University of California IP from the Los Alamos National Laboratory[19] development of OpenEMed[20] in the 1990s. The absence of clear IP links to LANL or UC by the Palamida and OS-CDE investigations is not an unassailable assertion; it simply represents the first public opportunity to openly query the raw code for IP traces. The persistence of the rumor that IP from LANL is in the CDE is present even in the Forrester review.

[17] http://www.palamida.com

[18] Received on March 29, 2007

[19] http://www.lanl.gov -- A national security and strategic science development laboratory operated for the U.S. Government by the University of California

[20] http://www.openemed.org -- an OMG compliant medical record system, which evolved out of an earlier Telehealth project at LANL. OpenEMed is available under the BSD open source license.

OS-CDE Demonstration

A summary report was presented at the CHCF office on Monday April 23, 2007 by Stuart Turner, Will Ross and Jeff Wallace. All salient points in the OS-CDE project were discussed. However, the workbench demonstration of the OS-CDE was not stable at that time. Three weeks later, on Friday May 11, 2007, the functional OS-CDE was demonstrated to Jonah Frolich by Stuart Turner, Will Ross and Richard Haley. This led to several follow up tasks:

- Stuart Turner provided CHCF with an archive of the final production version (v.66) of the OS-CDE code
- Jeff Wallace provided CHCF with a quote for a final round of code cleanup prior to release of the OS-CDE
- A subsequent discussion with Jonah Frolich was held on Friday May 18, 2007 regarding the OS-CDE decision to disable the University of Pennsylvania Neural Net in the ICS, with Stuart Turner, Will Ross, Shaun Grannis, Odysseas Pentakalos and Jeff Wallace

The balance of this final report on the OS-CDE Bench Test will examine the following topics

- RLS Requirement
- OS-CDE System Architecture
- Patient Matching
- Neural Net
- OS-CDE Package

Following these discussions, the OS-CDE Report concludes with recommendations.

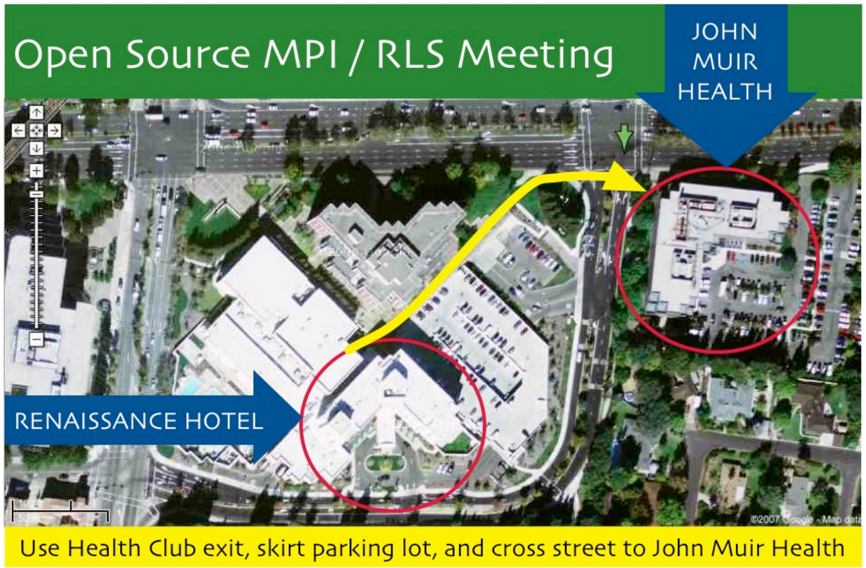

Open Source MPI / RLS Meeting
JOHN MUIR HEALTH
RENAISSANCE HOTEL
©2007 Google - Map data
Use Health Club exit, skirt parking lot, and cross street to John Muir Health

FIGURE 6 -- The long walk from the hotel to John Muir Health System, host of the Expert Panel

RLS Requirements

Requirements Source Documents

Based on prior negotiations with CHCF, the OS-CDE proposal explicitly stated:

> "To frame a desired end state of the prototyping project, RLS requirements will be gathered from the Taconic Health Information Network and Community (THINC)."[21]

As the OS-CDE project got underway it became apparent that THINC,[22] not previously consulted as a source of RLS requirements, was not able to provide them. Therefore, the project rapidly acquired the following normative and non-normative references. The following are reference documents used to inform ad-hoc RLS requirements, and may be important references for future development.

Normative References

- ASTM E 1714 -- Standard Guide for Properties of a Universal Healthcare Identifier. Although adoption of a UHID is arguably an impossible task for political and social reasons, this document, last published in November 2000, provides a substantive body of work regarding patient identifier terminology; data element use; high level encryption requirements; scenarios for use within several types of health record applications; and is scoped for the population of the United States.
- ASTM E 1239 -- Standard practice for description of Registration-Admission, Discharge, Transfer (R-ADT) data for Electronic Health Record (EHR) systems
- ASTM E 1239 -- Standard practice for description of Registration-Admission, Discharge, Transfer (R-ADT) data for Electronic Health Record (EHR) systems
- ASTM E 1394 -- Standard specification for transferring information between clinical instruments and computer systems
- Patient Identifier Cross-Reference (PIX) and Patient Demographics Query (PDQ) -- HL7 implementation/IHE Technical Framework

Non-Normative References

- Metropolitan District of Columbia Health Information Exchange (MeDHIX) -- granular user interface recommendations that are generally outside the primary scope of the OS-CDE project

[21] "OS-CDE Proposal" *ibid.*

[22] http://www.taconicipa.com

- CareSpark[23] (Southwestern Virginia, Northwestern Tennessee, Western North Carolina, and Southeastern Kentucky)
- Utah Health Information Network[24] (UHIN)
- Colorado Health Information Exchange[25] (COHIE)
- Greater Rochester (New York) Regional Health Information Organization[26] (GRRHIO)
- OpenEMed[27]
- SBCCDE "CDE Product Requirements Document" (circa 2000) within the CDE Technical Specifications Package[28]

In addition to these formal artifacts, discussion of the INPC architecture during the Expert Panel, and comparison to the RLS requirements built into the CDE architecture, led to an informal working assumption that the OS-CDE will intentionally migrate the underlying system architecture from the complex, integrated and enterprise CDE topology to the simple, modular and federated INPC architecture. That is, as a working thesis the default end state RLS specification for the OS-CDE will be consistent with a Regenstrief style federated topology of logically separate repositories with locally authoritative demographic updates and no persistent enterprise record number.

Requirements Specification

IDENTITY MATCHING

- The system shall provide a statistically based matching algorithm -- to enable tuning against false positives
- The system shall err on the side of specificity rather than sensitivity -- to minimize false positives
- The system shall be configured to keep false positives below 0.001% (1 in 100,000) -- at least until this is shown to be statistically unattainable
- Matching algorithms shall be able to operate independently of the application -- the application architecture shall allow removal, modification or alteration of deterministic, probabilistic or hybrid algorithms as needed

[23] http://www.carespark.com

[24] http://www.uhin.org

[25] http://www.corhio.org

[26] http://www.grrhio.org

[27] http://www.openemed.org

[28] Received from CHCF on July 31, 2007, and not available to OS-CDE team during project, but now archived to assist with future OS-CDE development activities.

- Heuristic changes to matching thresholds shall be non-programmatic -- similar to the above, the application shall not require programmatic changes, recompiling or other system level alterations in order to tune the matching functionality of the system
- The system shall return a null value if no records are found which meet the community defined minimum acceptable matching probability -- there is no "you're getting warmer" response to an identity matching query; if a user lacks appropriate access credentials to view known patient data, then the system shall not return any hint that more data might be visible if only the user had stronger credentials
- The system shall trust site level demographic data as authoritative -- changes in site level demographic details shall flag a record for regrouping
- The system shall allow dissassociation of false positive identity linkages (network level overrides) at the system level
- The system shall provide error correction feedback functionality to record input on site level demographic accuracy -- INPC example of twins returning incorrectly as exact matches, or of typos in site level records causing incorrect matches

SYSTEM CHARACTERISTICS

- The system shall support user level authentication to allow or restrict access to site level data -- multiple strategies will be available to conform to site, user and patient privileges to opt-in or opt-out as appropriate. For example, access rules to a patient record may be systemically relaxed during a predetermined "follow up window" after a site level patient care visit for specific, clinically predetermined conditions.[29]
- The system shall create audit logs for all RLS queries by user and by individual demographic identity
- The system shall not support "break the glass" overrides to exceed user permissions -- if role based permissions are set correctly then "emergency" overrides are unnecessary. For example, while physician access to patient records is legally restricted by privacy laws regarding certain types of data (e.g., HIV status, mental health, etc.), law enforcement and public heath users have a clear public safety mandate that overrides most restrictions.

[29] This standard procedure to temporarily unlock patient records based on recent hospital discharge status is broadly utilized by INPC.

OS-CDE Architecture

RLS Straw Person

The OS-CDE team was provided with two resources to articulate the CDE software architecture:

- "Care Data Exchange Technical Review" by Forrester Research , May 17, 2005
- Direct participation by Jim Hazen, current CDE developer, and one of the technical subject matter experts interviewed in the Forrester review.

A high level CDE architecture diagram as developed by Forrester is shown in FIGURE 7.

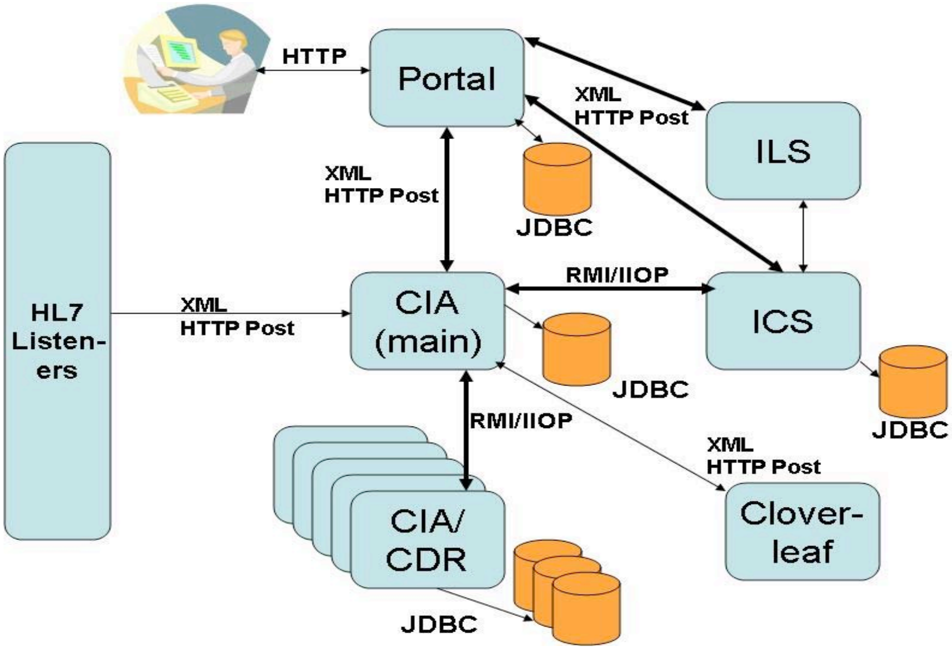


FIGURE 7 -- Forrester diagram of CDE Information Architecture

As built, every module in the CDE diagram is either dependent upon or utilizes proprietary software. The OS-CDE project was conceived as a code fork, comprehensively replacing proprietary CDE services and dependencies with explicitly open source solutions. Some of the key proprietary CDE components to replace were:

- BEA[30] Weblogic portal
- UPenn Neural Net in ICS
- Intelligent Search Technology in ICS

[30] http://www.bea.com

- Data Junction[31] (XML transformation tool)
- Oracle[32] DBMS

The OS-CDE discarded these and other proprietary components or dependencies in the pursuit of an explicitly open source RLS.[33] A classic RLS, based on the Markle conceptual diagram, is shown in FIGURE 8.

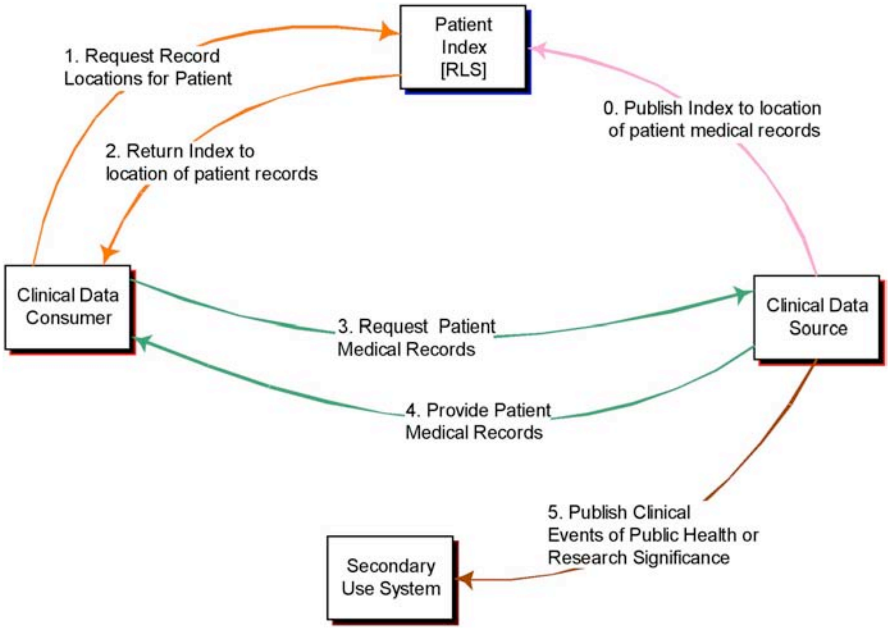

Patient Index [RLS]
1. Request Record Locations for Patient
2. Return Index to location of patient records
0. Publish Index to location of patient medical records
Clinical Data Consumer
3. Request Patient Medical Records
Clinical Data Source
4. Provide Patient Medical Records
5. Publish Clinical Events of Public Health or Research Significance
Secondary Use System

FIGURE 8 -- Classic asynchronous RLS diagram, published in CSC - Connecting for Health NHIN Prototype Architecture proposal submitted to ONC in January 2007

Retaining only the CDE Information Correlation Service (ICS) and Information Locator Service (ILS) modules, the OS-CDE team proposed a strawperson RLS architecture (see FIGURE 9) by imagining four named services to substitute for the discarded components from the CDE Information Architecture.

- An Integration Engine to acquire incoming clinical data (replacing current proprietary software pathways -- the Mirth Project[34] immediately comes to mind as an open source candidate for external HL7 listening responsibilities)
- Standardized Repositories to store the data in an open source DBMS[35]

[31] http://www.pervasive.com

[32] http://www.oracle.com

[33] The possibility of discarding the CDE entirely and starting over from scratch was briefly entertained as an option.

[34] http://www.mirthproject.org

[35] For continuity and scope consistency the OS-CDE project utilized the native CDE database schema

- An Access Broker to handle role based authentication and authorization services before an approved query can be issued to the ICL or ILS
- An RLS Portal to provide user facing interaction

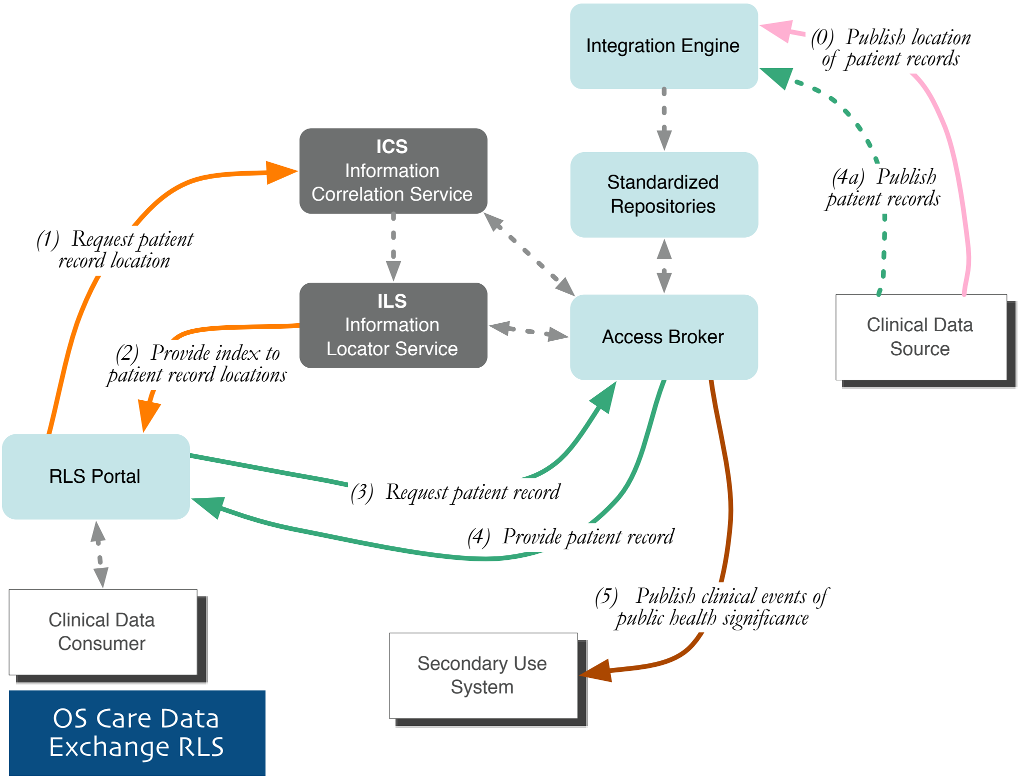

Integration Engine
(0) Publish location of patient records
ICS Information Correlation Service
Standardized Repositories
(4a) Publish patient records
(1) Request patient record location
ILS Information Locator Service
Access Broker
Clinical Data Source
(2) Provide index to patient record locations
RLS Portal
(3) Request patient record
(4) Provide patient record
Clinical Data Consumer
(5) Publish clinical events of public health significance
Secondary Use System
OS Care Data Exchange RLS

FIGURE 9 -- Adapting the OS-CDE to the RLS query & response pattern.

Building fully operational versions of all these services was not in scope for the OS-CDE project; the key role of the bench test was to demonstrate RLS capability, not to complete engineering on a comprehensive alternate RLS package. Accordingly, for the static demonstration of an open source RLS only two of the four imaginary services (in FIGURE 9) were needed:

- Clinical data repositories must be loaded with sample data; and
- A portal must allow:
  - i. User issues query to RLS -- *"Do you have a record for John Smith DOB 1-1-2001?"*
  - ii. User receives response from RLS -- *"Yes, I have records for John Smith DOB 1-1-2001. They are located at Clinic1, Clinic2 and Clinic3."*

In other words, a static data load makes HL7 listening unnecessary, and there is no need to encumber the demonstration process with user authentication or access controls.

Substituting Open Source Components

Checking in the ICS and ILS source code into the OS-CDE Subversion repository allowed an opportunity to identify all external package calls and dependencies. Disabling calls to discarded proprietary components prompted the creation of either new calls to successor components (i.e., open source) or to temporary classes (i.e. placeholders).

LOAD PATIENT DATA

An early conundrum was the effort necessary to populate derivative repository fields in the ICS. A custom data loader tool[36] was built (see FIGURE 10) to automate the process.

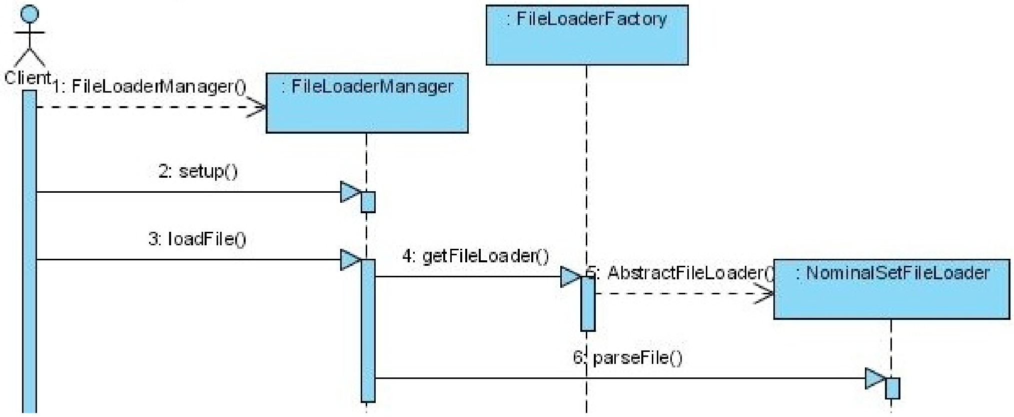


FIGURE 10 -- Temporary data loader for ICS

CLINICAL REPOSITORY INTERFACE

- Complex ICS database schema[37] was retained unaltered
- Loaded into PostgreSQL (replacing Oracle)

CORRELATION SERVICE

- Disable Neural Network correlation service
- Substitute call to new drop in algorithm service
- Replace proprietary database interface (Data Junction)

[36] See Appendix G

[37] See Appendix F

LOCATOR SERVICE

- Aside from calling the new JBoss portal, few changes were needed

USER INTERFACE / PORTAL

- Replace BEA Weblogic with RedHat JBoss (see FIGURE 11)

System Topology Issues

- The CDE database schema appears to contain many legacy tables of questionable value for a modern, agile RLS implementation
- The database service is hard coded to the CDE schema, which should be replaced with a Hibernate style abstraction layer for modularity
- ICS assumes standing links between matched users which is more labor intensive than on-the-fly matching services (while on-the-fly requires clean, normalized data and an exquisitely sensitive and well tuned algorithm)

System Documentation

The OS-CDE project functioned as a discovery based process exploring code with almost a total absence of any development artifacts. As a capability demonstration with a limited budget and a short time frame, the OS-CDE sketched some high level diagrams,[38] but did not engage in a rigorous documentation effort.

[38] See OC-CDE presentation to CHCF on April 23, 2007

Patient Matching

Modularizing the OS-CDE Matching Service

Modularizing the matching service to associate a unique person with one or more records was a core objective of the OS-CDE project. Instantiating a simple deterministic algorithm was sufficient to demonstrate the architectural modularity of the OS-CDE code base. While the development of an appropriate algorithm was not a central project goal, the project enjoyed an extended collaborative discussion on identity matching.

The development tasks were also driven by a desire to not only modularize the matching algorithm component, but to remain relatively agnostic to the type of algorithm deployed, whether it be deterministic, probabilistic, a hybrid of these two methods or a combination of two or more methods. It also supported the notion that the algorithm(s) chosen for the demonstration phase of the project could evolve and a production instance of the OS-CDE fork could serve as a bench tool or developers kit for testing new algorithm approaches as well as to eventually perform as a patient identity resolution and record locator service within a production implementation.

Cultural Naming Barriers to Accurate Linking

In an ideal world individual person naming attributes would be defined explicitly to resolve differences in data types, including variances such as use of initials vs. full spelling of the atomic name element for middle name, variances in spelling, etc.. However, current “best practices” in health care patient data collection are a potential barrier to identification of an appropriate minimum set of patient demographic elements. While outside the OS-CDE scope, the team discussed cultural practices with fundamental differences in naming conventions, especially in Asian and Hispanic cultures. These practices are especially relevant within California’s diverse population.

- In Vietnamese cultures, women often retain their maiden name, but children assume their patrilineal name.
- Hmong and Cambodian culture historically had one name. Today the Hmong name represents the clan, not the family unit, they belong to.
- Mien (or Yao) have 12 clan names, often spelled differently based on differences in individual phonetic interpretation (e.g. Saefung vs. Saefong). Name order may place Thai name first, followed by Mien clan and then by Mien given name.[36]

[36] http://wiki.openempi.org/openempiwiki/index.php/Comments_on_naming_conventions_in_SouthEast_Asian_Cultures

- In Hispanic cultures, a child's proper surname is derived through both matrilineal and patrilineal lines and therefore has two components, a paternal surname (*apellido*) followed by a maternal surname (*apelativo*) (e.g., the proper name for the former President of Mexico is Vincente Fox Quesada).

Imposing the casual American minimum data set of first name, middle initial and last name is a potentially substantial barrier to accuracy in identity linking (e.g., where Eduardo G. Alvarado, Eduardo A. Garcia, and Eduardo Alvarado-Garcia can be the same original identity captured inexactly by computerized admission systems at separate health care institutions.) Clearly it is beyond the scope of the OS-CDE project to solve the absence of normative standards that correctly capture culturally common person naming conventions, but it is worth mentioning in the context of any system that seeks to accurately link person identities across separate institutional computer systems.

Data Elements

When discussing identity linking across disparate systems it is difficult to avoid the desire to target a unique patient identifier that is consistent and reliable. Each patient demographic element can be weighted either qualitatively or quantitatively based on the data element discriminating power or the identifier ability to reduce the global population to a sufficiently small number to statistically guarantee a match. These data elements fall under three basic categories.[37]

Permanent Data Segment

Relatively immutable person attributes (e.g., date of birth, place of birth, mother's maiden name, etc.)

Longitudinal Data Segment

Relatively persistent person attributes over the lifetime of a patient (e.g., Social Security Number, driver's license, permanent address, occupation, name of children, etc.)

Health Service Data Segment

Health organization data elements intended to classify, identify and locate clinical artifacts (e.g., service type, provider, organization identifier, dates of admission and discharge, etc.)

[37] RIDE (Roadmap for Interoperability of eHealth Systems in Support of COM 356 with Special Emphasis on Semantic Interoperability), http://www.srdc.metu.edu.tr/webpage/projects/ride/...PatientIdentifiers

SSN Discussion

Clearly, the most ubiquitous patient identifier that fits well with the qualities of a discriminating identifier is the Social Security Administration Number ("SSN"). A critical evaluation of the uniqueness of the SSN quickly yields serious concerns, yet it is widely collected and when available often contributes as a key identifier in matching algorithms. Grannis reports that in the Indiana Network for Patient Care (INPC), all 71 hospitals in the system collect the SSN and 67% of all records in the INPC global registry have SSN present.

Advantages to using a Social Security Administration Number (SSN)

- There are a large number of Social Security offices in the US (~ 1,300) with a well established electronic network and with relatively short notice, patients can be assigned a number if one does not exist.
- Use of SSNs is ubiquitous and commonly collected by healthcare organizations

Disadvantages to using a SSN

- The SSN is not always unique. That is, a one to one (1...1) relationship between individual and number. A relatively large number number of people have more than one SSN (approximately four million). The matching algorithm may correctly identify the right person, but the presence of two or more SSNs assigned to the same individual presents the potential for failure to identify all document artifacts especially those distributed horizontally (between different healthcare organizations) and longitudinally (temporally) over time.
- SSNs are not issued at birth. Even if a parent requests a number be assigned at birth, the birth registration system must complete processing with communication to the Social Security Administration who then subsequently issues a number. There is obviously some period of latency in this process, although California is included with all other states in a program to provide "Enumeration at Birth".
- Up to 20% of individuals do not know their SSN.
- 20% of SSNs fail to validate against the SSA master database. However most of those that failed (85%) could eventually be resolved manually.
- The Social Security Administration does not remove SSNs from the system. A large number of SSNs exist that are assigned to people who are expired.
- Not everyone seeking health care is eligible to receive a SSN.
- Individuals can dissent against the use of their SSN as a unique health care identifier.
- Some states prohibit the collection of the SSN for health care (e.g., Massachusetts)
- The SSN does not provide a check digit. Check digits are algorithmically derived to check for keying (e.g., manual data entry risks transposition and other errors) or errors of validation. This creates the risk of either failing to identify the patient and associate his

or her records, but also to positively identify, but associate the wrong records to the patient.

- Some hospitals assign newborns their mother's SSN.

National Death Index

The National Death Index (NDI) has been capturing vital status and cause of death information through the National Center for Death Statistics since 1979. Research users pass study data to the registry which uses an adaptation of a two-staged probabilistic method developed by Fellegi and Sunter (1969) and Rogot, Sorlie and Johnson (1986).[38] The first pass determines matches based on agreement against at least one of seven criteria (listed below). Then, each match on these items is scored and summed (Horm, 1996). According to the NDI, "All matches on last name and father's surname are performed on the basis of either exact spelling or NYSIIS (New York State Identification and Intelligence System) Phonetic Codes . These criteria are listed below:[39]

(1) Social Security Number

(2) Month of birth (exact) and +/- year of birth, first name and last name

(3) Month of birth (exact) and +/- year of birth, first and middle initials, last name

(4) Month and day of birth (exact), first name and last name

(5) Month and day of birth (exact), first and middle initials, last name

(6) Month and year of birth (exact), first name, father's surname

(7) If the subject is female: Month and year of birth (exact), first name, last name (user's record) and father's surname (NDI record)

Matching Algorithm

The core purpose of the OS-CDE was to demonstrate open source functionality, not to investigate matching algorithms. The following simple exact-match algorithm was used to demonstrate the modularity and functionality of the OS-CDE solution.

```
SSN + FN + LN
SSN + YB + MB + DB
SSN + FN + YB + ZIP
FN + LN + YB + MB + DB
```

[38] "Zombies, Immortals and the Which Hunt: NDI Record Linkage - It's More Than False Positives and False Negatives", Sayer B et al, Proceedings of the 2002 Joint Statistical Meeting (JSM), New York, New York.

[39] National Death Index (NDI), US Department of Health and Human Services, Centers for Disease Control and Prevention, National Center for Health Statistics. URI: http://www.cdc.gov/nchs/r&d/ndi/ndi.htm

Neural Network

Gone But Not Forgotten…

In the original CDE, the ICS module uses a set of person matching functions based on an artificial neural network (NN). During the OS-CDE Bench Test it was decided to persist, yet disconnect, the NN and not rely on its functionality to provide patient matching services. Some key reasons were:

- IP Concerns. The NN, originally developed by the University of Pennsylvania, was implemented by CareScience in the production instance of the CDE application. A review of the source code, including feedback from the Palamida analysis, suggested that one or more copyrights or patents on the neural network algorithms used in the CareScience application exist (held by both CareScience and/or the University of Pennsylvania). It was presumed, but not confirmed, that these patents, if they exist, were transferred to CHCF intrinsically as part of the intellectual property transfer. However no specific patents were detailed, and a cursory search of the United States Patent and Trademark Office database yielded no patents related to the U. Penn. neural network or CareScience in general.
- Out of Scope. The neural network as implemented in the CDE was initially trained by several people over several months[40] working with thousands of data points from a third party commercial phonetic matching utility. This phonetic matcher was proprietary. To meet the explicit open source target for the OS-CDE project, an alternative non-proprietary phonetic matcher would be needed to retrain the NN. Thus, continued use of the NN in an explicit open source workbench demonstration was clearly outside the scope of the immediate OS-CDE project, both in the spirit of the investigation and in the time and resource constraints.
- Not Self Adaptive. Curiously, the key benefit of a neural network to learn over time on accumulative demographic data was disabled in the CDE implementation of the NN. The existing CDE decision logic as trained against the original phonetic matching utility was implemented in the NN as a static rules engine where heuristics (probability thresholds) could be applied to make changes to the outcome. The NN was not implemented as a self-adaptive system to respond to new patient identity data. In the absence of "retraining" the static rules governing its operations, the NN would be expected to degrade in performance over time.
- Not Critically Appraised. No documentation was available to the OS-CDE team that critically evaluated or validated the NN performance.[41] Independent study with statistical outcomes (e.g., sensitivity, specificity, precision, recall, positive predictive value,

[40] Per the recollection of Jim Hazen. Specifically, he referred to it as "more of a burden than an aid," and as a prime candidate for deletion from any future CDE architecture roadmap.

[41] Shaun Grannis was unaware of critical appraisals of neural networks in general for use in person identity matching.

etc.) would have had considerable impact on the OS-CDE team's confidence in the NN approach to person identity matching.

- Insufficiently Modular. The OS-CDE project has a compelling architectural priority to modularize the capabilities of the code packages to function as a utilitarian toolkit for generalized use in an RLS implementation. Disabling the CDE's direct calls to the NN while enabling an external drop in (i.e., modular, user configurable) approach to identity matching would meet one of the key production goals of the OS-CDE project, and would not foreclose future use of the NN as a drop in tool, should it be desirable.

Using the existing NN code would have taken less time for the OS-CDE Bench Test, but for the reasons cited above the team considered circumvention to be the correct path forward. The original NN code remains intact, but it is not currently called by any of the production OS-CDE paths. If warranted, the NN can be reactivated with nominal additional work. However, pending resolution of the IP status, the OS-CDE team recommends complete removal of the NN from the explicit open source fork.

OS-CDE Package

As built, the OS-CDE tool kit utilizes the explicitly open source components listed in Table 2.

Component	Function	Advantage
Apache HTTP Server	Secure, efficient and extensible web server for Linux, Windows and other modern OS implementations	Used for nearly 70 percent of websites; robust feature set; commercial support available
Apache Tomcat Server	Reference implementation of Java Servlet and Java Server Pages technologies	Enables rapid deployment of highly dynamic functionality to websites; highly scalable; robust support for high clustering and high availability implementations
Java 2 SDK v 1.4.2 with NetBeans 5.0	NetBeans provides a modular, integrated development environment (IDE) for Java.	NetBeans 5.0 introduces comprehensive support for developing IDE modules and rich client applications based on the NetBeans platform; supports JBoss server
Xdoclet	Code generation engine allowing attribute oriented programming in Java	Simplifies deployment of servlets in a Java application server environment
JBoss	J2EE application Server	Built by an operating division of RedHat, JBoss is the leading open source J2EE application server; commercial support available
PostgreSQL	RDBMS	Mature relational data base system
Spring	Application framework for J2EE	Simplifies development of applications by providing abstractions around EJBs, data access and other services
Xalan	Allows conversion of XML documents using XSLT transform specifications	Implements the Java API for XML processing (JAXP); can be used to convert between XML and XML-derived document types, including HTML
Xerces J	High performance XML processor	A library for reading and writing XML documents, Xerces helps implement web services bindings between applications

TABLE 2 -- OS-CDE Package Components.

Demonstration

The formal presentation to CHCF on April 23, 2007, with a complete narrative about the project, lacked a working demonstration of the OS-CDE server. Three weeks later, on Friday May 11, the OS-CDE team finally released a stable build (#66), and was able to demonstrate to CHCF consistent RLS queries running correctly on an explicit open source stack built from legacy CDE components (ICS + ILS).

Sample Data File

For the OS-CDE demonstration, a custom file of 3,000 synthetic patient identities was built[42] with the following parameters.

- 14 Data fields

  LNAME,FNAME,SECNAME,ADDRESS_1,ADDRESS_2,CITY,STATE_PROV,ZIP,

  COUNTRY,PHONE_AREA_CD,PHONE_NUM,GENDER,SSN,DOB
- Zip codes limited to California
- No match between gender assignation and usual gender of first name
- All vowels suppressed from first and last names (to prevent an accidental match with a real person identity)

The file was split into two repositories, and the following ten identities were loaded into both repositories to enable discovery based testing of the RLS.

```
CWGR, WLLM, , 448 E FORK CANYON CREEK, , MOORETOWN, CA, 91180,
USA, 542, 185-4557, F, 998204542, 3-7-1971

FSTR, CRYSTL, , 418 UNION PACIFIC RAILROAD AVE, ,CALIFORNIA, CA,
90470, USA, 375, 780-7721, M, 995016986, 10-10-1949

KNG, LS, , 890 17TH ST, , ENTERPRISE RANCHERIA, CA, 92300, USA,
508, 442-9740, F, 992543828, 12-15-1973

LN, BRDGT, , 510 CAMP CREEK WAY, GRIDLEY, CA, 92380, USA, 592,
601-7621, M, 994128449, 11-29-1997

RZ, VCTR, , 473 RAND, , ENTERPRISE, CA, 92380, USA, 292,
445-1853, M, 992351070, 9-28-1957

TRBGH, JSN, , 386 1ST ST, , BERRY CREEK, CA, 92300, USA, 626,
171-1920, M, 995356748, 12-3-1959

TRRY, LS, , 548 UNION PACIFIC RAILROAD LN, , MOORETOWN
RANCHERIA, CA, 92300, USA, 341, 303-6513, F, 996471893, 1-3-1946

FRR, RBN, , 581 FRENCH CREEK, , CALIFORNIA, CA, 91180, USA, 294,
780-1039, M, 995652654, 2-21-1983
```

[42] Shaun Grannis has an amazing tool that rapidly generates huge numbers of dummy identity files. In fact, he made in a few minutes the 3,000 name production file during one of our conference calls.

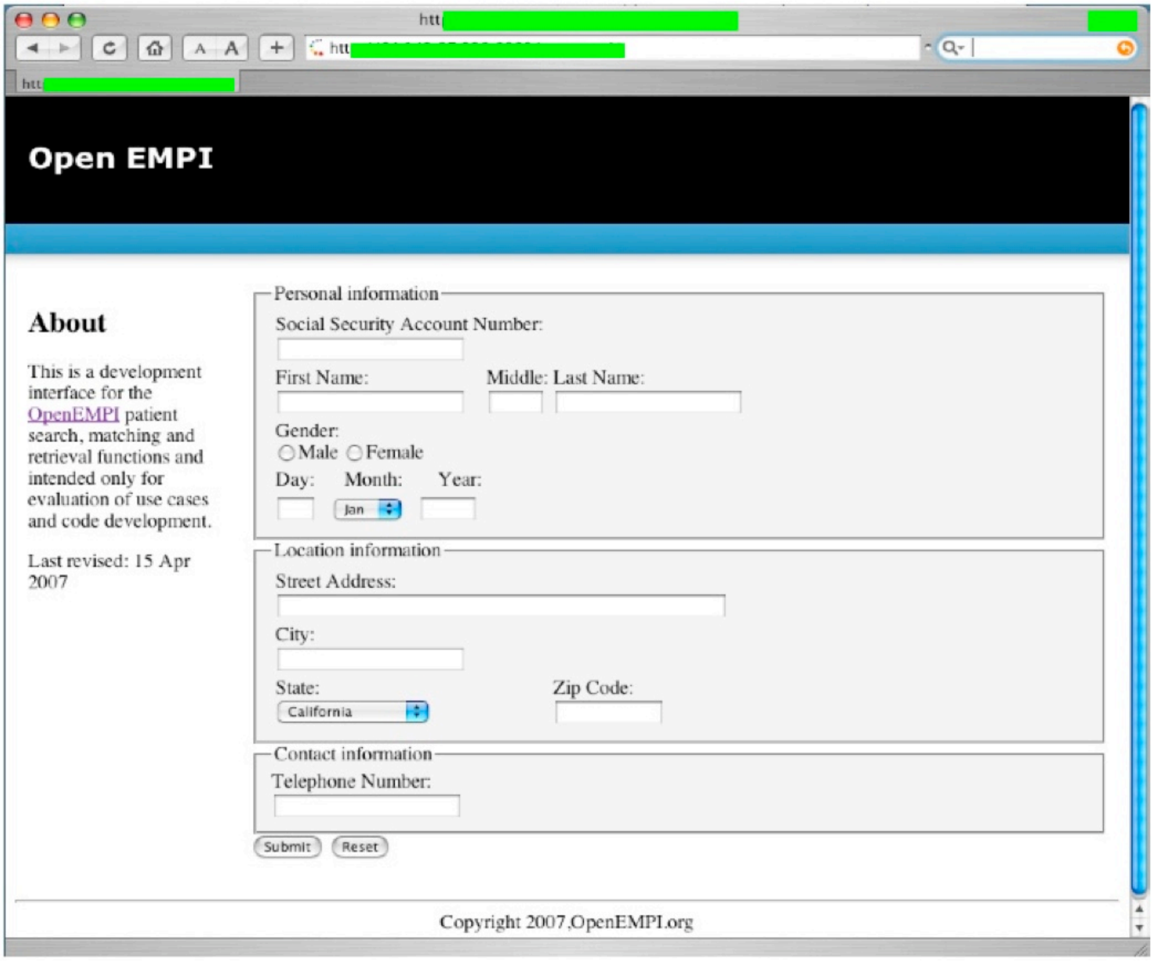
```
BRWN, FRNK, , 111 PARKWOOD CT, , CHICO, CA, 92380, USA, 631,
609-9606, F, 998454136, 5-31-1967
LMB, DN, , 531 GLENN, , PALERMO, CA, 92380, USA, 560, 711-7264,
F, 991012574, 7-26-1975
```

Two Screens

Without the requirement to function as a production system, the OS-CDE Bench Test features a JBoss portal with two screens

(1) RLS Query -- A data entry form grouping the 13 known data elements into three information categories (see FIGURE 11)

(2) RLS Response -- A data display form which echoes the values entered along with a match result message in the bottom row of the table (see FIGURE 12)



About

This is a development interface for the OpenEMPI patient search, matching and retrieval functions and intended only for evaluation of use cases and code development.

Last revised: 15 Apr 2007

Personal information

Social Security Account Number:

First Name: Middle: Last Name:

Gender:
○Male ○Female

Day: Month: Jan Year:

Location information

Street Address:

City:

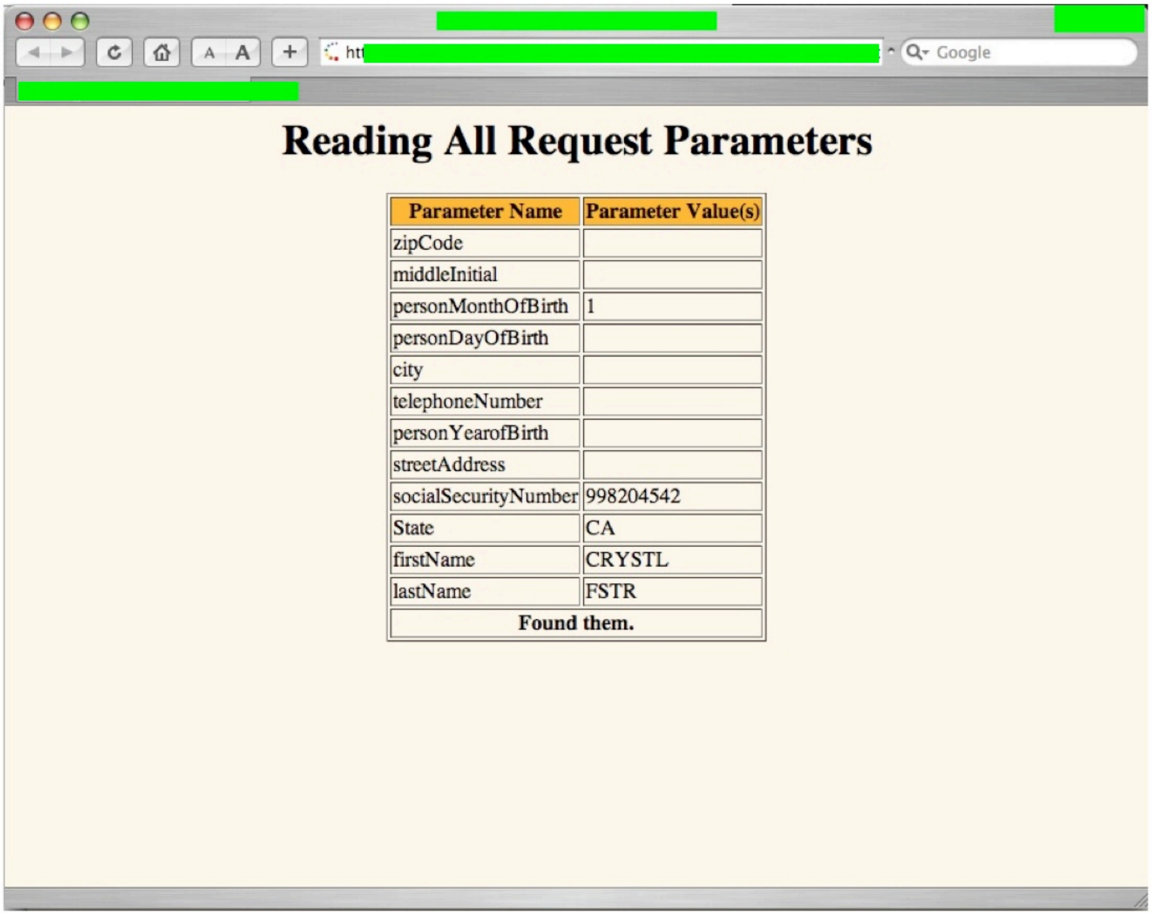
State: California Zip Code:

Contact information

Telephone Number:

Copyright 2007.OpenEMPI.org

FIGURE 11 -- RLS Query form at OS-CDE demonstration server



Reading All Request Parameters

Parameter Name	Parameter Value(s)
zipCode	
middleInitial	
personMonthOfBirth	1
personDayOfBirth	
city	
telephoneNumber	
personYearofBirth	
streetAddress	
socialSecurityNumber	998204542
State	CA
firstName	CRYSTL
lastName	FSTR
Found them.	

FIGURE 12 -- RLS Response at OS-CDE demonstration server. "Found them." displayed in the bottom row is the match message. In this case, the identity data elements entered were MB + SSN + ST + FNAME + LNAME. If there was no match, "Didn't find them." would be displayed in the match message row.

Recommendations

The OS-CDE project has produced a functional yet relatively immature executable software library. 'Immature' in this context recognizes the limited OS-CDE project scope (e.g., no requirements building, unit testing, etc.) and the short three month development cycle. 'Immature' also recognizes a hope for renewal expressed by the OS-CDE fork. The legacy CDE codebase emerged prior to current best practices (e.g., HL7's Reference Information Model, etc.). The brief OS-CDE bench test has inherently not been harmonized with the object classes and the inheritance of entity (e.g. person) and role (e.g. patient) classes in the more modern paradigm. This does not prevent the application from functioning correctly, but it emphasizes the potential near term obsolescence facing the system design without time and attention to improving the underlying model as well.

Another broad concern regarding compatibility with emerging standards is the physical data model and data attributes. A total of 44 persistent person attributes, some of which are administrative data elements, plus 12 additional attributes in a separate class for updating purposes (e.g., maiden name) are modeled in the schema. Reviewing these data elements for harmonization (e.g. data types, element descriptions, value domain if enumerated, and concepts) and mapping to existing data element standards will improve the library's ability to interoperate with a larger number of disparate applications.

Overall, the OS-CDE project objectives were less about creating a production instance and more about a demonstrating that a legacy closed source code base could be released from its proprietary and commercial dependencies and deployed within a fully open source stack.

It is important to recognize that the OS-CDE fork is a dependent executable library, because the identity correlation and record locating modules are by themselves inoperable without other layers. These include a robust user interface, auditing and logging functions with reporting tools, a critically appraised matching algorithm that is appropriate for its deployment environment, a web services hook into a person index (repository) as well as a method to feed patient demographic information (e.g. from an ADT stream) with data transformation (messaging) and mapping (semantics) to create a reliable and synchronous flow of information.

The following are salient concerns and recommendations for the next phase in the evolution of the CDE code base in general, and the OS-CDE fork in particular.

- Refactor the existing code to improve its structure, modularity, and readability. This is an essential exercise for a competitive open source product. It is the best way to truly expose the code base, evaluate it critically, and fully understand its design since the original developers and documentation were absent. Refactoring will provide better structure and is essential to generate comprehensive documentation, not only for current operations but for future developers. The current OS-CDE testers were unable to share in this type of knowledge transfer in the brief project cycle.

- Complete a software requirements specification. This activity should include subject matter experts in patient matching algorithms as well as software engineers to be sure that the objective of algorithm modularity is not only feasible, but functional and extensible. The application should be sufficiently modular so that one or more patient matching components may be readily embedded without changes in source code. Additional testing, including regression, black box and gray box testing of the algorithm with realistic patient data sets, is expected.
- Build formal use cases. Develop one or more scenarios or representations of the existing project design against a more idealized design based on evolving requirements gathering and feedback from the user community. UML engineering diagrams as well as textual use cases need to continue to be developed openly (e.g., project wiki, etc.).
- Review and annotate the logical and data model. As mentioned previously, changes to the data model should also include improved represtation of data element metadata (e.g., data type, descriptions, concept mapping, enumeration of value domains when they exist, etc.).
- Productize the source code for community release. This work should include an organized approach to release numbering, release of branches, the transition from an internal alpha to a public beta and a roadmap toward stabilization. Other activities should focus on well written technical and end-user documentation, including concise installation or build artifacts, and a well-written end-user guide that clearly outlines features, proper use and cautionary advice regarding risks associated with inappropriate use of the application. The project community tools should have clearly organized user-facing components with methods to encourage users and administrators to submit feedback (e.g., usability, feature requests, alternative use cases, etc.). As stated before, this activity should be managed under a formal open source collaborative environment where issue tracking, documentation, feedback forums and versioning are managed collectively. The project already is registered at Sourceforge and its persistence and further utilization as a public project is recommended.
- Complete a gap analysis comparing the OS-CDE with other initiatives, frameworks and standards. Compare the OS-CDE architecture design, components and functionality with the Open Health Framework's PIX/PDQ and clinical document metadata standard (XDS), HL7's Clinical Document Architecture (and others). This will help developers and potential users to understand inherent similarities and differences, and will provide better guidance on local implementation, and will improve understanding of interoperability in a federated environment.

Refactoring and requirements harmonization should include community feedback regarding user facing interface components. As a library, a portal wrapper with user interface is not a key requirement; however, usability analysis and use case scenarios during the next phase should be gathered to be certain that front-end question text, form components and workflow are sufficiently assessed. These exertions, while not trivial, will be necessary for the next iteration of the OS-CDE fork to both prioritize appropriate functionality and to optimize the overall package for a robust future roadmap.

Acknowledgments

Stuart Turner and Will Ross, OS-CDE Co-Investigators, thank the following individuals for support, sympathy and participation

- Dave Minch
- Richard Haley
- Shaun Grannis
- Scott DuVall
- Odysseas Pentakalos
- Jim Hazen
- Jeff Wallace
- Jonah Frolich

Appendix

A. Med Records Get Cue from Napster, Wired News, December 14, 2000

B. Open Source Health Information Exchange Fact Sheet, CHCF, November 2006

C. Agenda for Open Source RHIO Forum, November 15, 2006, San Francisco, California

D. INPC Notes from OS-CDE Expert Panel, Friday March 9, 2007, Walnut Creek, California

E. CDE Notes from OS-CDE Expert Panel, Friday March 9, 2007, Walnut Creek, California

F. ICS Database Schema, From CareScience

G. Loading Data, from OS-CDE Wiki

Appendix A

Med Records Get Cue from Napster

Wired News, December 14, 2000

WIRED

« Back to Article

Med Records Get Cue From Napster

Michelle Delio 12.14.00 | 2:00 AM

Reader's advisory: *Wired News has been unable to confirm some sources for a number of stories written by this author. If you have any information about sources cited in this article, please send an e-mail to sourceinfo[AT]wired.com.*

Many medical workers must access patient records using technology that is so outdated it makes punching presidential votes into a butterfly ballot look downright sophisticated.

According to a recent report by the National Academy of Science's Institute of Medicine, at least 50,000 deaths occur each year due to medical errors that are often linked to physicians not having immediate access to information that they desperately need to care for their patients.

CareScience aims to change those statistics using Napster-like peer-to-peer (P2P) technology to give medical personnel easy access to crucial health-related data.

"We know that early detection is the key to treating many conditions," said John Erkins, a medical records supervisor at Women's College Hospital in Toronto. "So if a report gets lost, or a referral for further tests gets delayed, even for a short while, it can literally be a matter of life and death. We truly need to find a way to move information quickly and reliably."

CareScience began development of the "CareScience Care Data Exchange" in November 1999. The beta release of the system is scheduled for January 2001.

P2P allows users to transfer data directly from one computer to another. Although the technology itself isn't new, it has recently been infused with new life -- mostly due to the media blitz that centered on Napster and other innovative, mostly-music-sharing, P2P systems.

The cost of a P2P system is also significantly less expensive than that of a central server-based system, since P2P simply provides an interface that allows people to share information stored on individual computers or networks. The developer of a P2P system doesn't have to house, configure, or care for a database crammed full of files.

David Brailer, M.D., chief executive officer of CareScience, said that easy, reliable access to complete information is considered the "Holy Grail" in health care because so many errors and so many unnecessary treatments are driven by inaccurate or unavailable information.

"Here's a typical scenario: A physician decides to have a specialist evaluate a patient. The first step is to get authorization from the health plan, which is typically done over the phone. The doctor fills out a referral, Xeroxes the chart, and mails it to the specialist. The specialist sees the patient and returns his or her evaluation in writing. The whole process can take as long as 90 days, and in four out of 10 times a result or request simply gets lost in the system."

The consequences of missing data can be severe.

"Not knowing what drug someone is on leads physicians to prescribe other drugs that could initiate an allergic reaction because of the unknown drug," Brailer said. "Or, if people get admitted to a hospital because they have a slightly bizarre EKG result, and the physician doesn’t know if it has changed from the last one they had, they have to admit the patient to the hospital."

Brailer said there have been numerous efforts in the health care industry to develop viable ways to move and share information.

One of the largest efforts was the Community Health Information Networks (CHINs) -- large, centralized, mainframe-driven databases that were intended to contain all the medical records on everyone in a particular community.

But this centralized approach was found to be enormously expensive. "In most cases, unaffordable" said Vernon Elden, who works as a community liaison at Richmond Memorial Hospital.

Elden said that CHIN also sparked debates, "and even some verbal brawls" over who controlled the stored information, who got access to it, and who owned what was stored on the central server.

"It sometimes got fairly ugly," Elden recalled. "Scientists and researchers, especially, can be amazingly vicious in a quiet sort of way."

More recently, some hospitals tried to form Integrated Delivery Networks (IDNs) that would exchange information within the hospital and affiliated laboratories, treatment units and doctors. That too failed, Elden said, again because of expense and information ownership issues.

CareScience chose the peer-to-peer architecture specifically because it allows users to "own" their own data while sharing it.

Brailer said that CareScience's P2P network isn't plagued by scalability problems, as many other P2P networks are.

Often, because P2P systems methodically move information from one machine to another until it arrives at its ultimate destination, a system can function well with a limited number of users but becomes sluggish when user numbers increase.

The Care Data Exchange works around this, as some other P2P systems have successfully done, by ensuring that no single "exchange," or computer using the system, is completely responsible for all of the transactions of all of the data, but each exchange also has open access to all other exchanges.

Care Science is also sensitive to privacy issues.

Brailer noted that he firmly believes that "doctors cannot have access to every human beings' information simply because they are doctors. So the authentication and the authorization issues are quite large, all the way up to the ability of the patient to give informed consent—saying, 'yes you can move my information from point A to point B'."

Consumers will not have access to the system when it launches in January. But Brailer says that since CareScience believes that the data that's being moved around ultimately belongs to the consumer they may soon open the system to non-medical users if there is enough interest.

Elden and Brailer both believe that this technology will be nothing less than transformative.

"It could solve such an enormous problem," said Elden. "This is analogous to a breakthrough drug -- after it is proven, the adoption curve is huge, and it changes the very nature of the problem so radically that the old way of doing things becomes instantly obsolete."

Appendix B

Open Source Health Information Exchange Fact Sheet

CHCF, November 2006

California HealthCare Foundation

Open Source Health Information Exchange

Prospects for an Open Source RHIO

FACT SHEET

November 2006

Background

In the spring of 2005, the California HealthCare Foundation (CHCF) hired Forrester Research to study the feasibility of making software it had funded for the Santa Barbara Care Data Exchange (CDE) available to regional health information organizations (RHIOs) under a free open source software license. Availability of a basic software solution should allow RHIOs to build their regional health information networks (RHINs) much more quickly than with conventionally licensed and custom developed software products. CHCF could also inform the state and national RHIO policy debate with its real-world experience in the field of RHIO formation and governance. Forrester introduced the name "Open RHIN" to describe an open source RHIN solution.

Open Source and Governance

Open source products are developed using a community model whereby source code is licensed to users to encourage modification and improvement. Open source software licenses grant users access to source code with the right to redistribute the source code and make derivatives of it—usually without paying a license or royalty fee. Most open source licenses require the user to contribute all derivatives, extensions, and improvements of the source code, which are then distributed back to the open source project for redistribution without royalties. Because of this process, each release of open source software is compatible with prior releases, consistent at its core, and available for free.

A successful Open RHIN will require the formation of a strong community to assume ownership and governance. Community governance means that development, management, and future direction of the Open RHIN software will be managed by a community of users, vendors, and systems integrators. This governance model will help insure that Open RHIN is aligned with real requirements. In addition, as Open RHIN is a complement of modules that together represent an actual application (and not a set of tools), its future development will require careful prioritization of features in close alignment with Open RHIN customers.

Software and Technical Design

The current CDE application has five distinct modules:

1. A portal linked to user roles (physician, patient, admin), implemented in BEA Weblogic Portal.™
2. A core application module that processes all incoming messages, demographics, and information feeds.
3. A clinical data repository.
4. An information correlation service (or master patient index—MPI), which uses a neural net for patient identification.
5. A record locator service.

A technical review of the source code determined that some parts of CDE uses code that CHCF licenses from others but does not own. The licensed components, including the Internet portal among others, must therefore be replaced with new code developed by the open source community. Some components also were not designed optimally and should be replaced or restructured by the Open RHIN community.

The Market

The RHIO market is made up of a series of markets. Some vendors focus on clinical data interfaces, including HL7 integration engines; some focus on customer data integration and master patient index engines; some focus on digitizing medical records through EMR/EHR applications or clinical repositories; and, lastly, some focus on providing staff and patient access to medical information through portals. This diversity reflects the many ideas about what RHIOs should do and what RHINs should be.

Forrester Research documented the level of interest in the Open RHIN concept by conducting interviews with key audiences: users, open source developers, product vendors, and systems integrators. Forrester interviewed 26 individuals from these audiences to assess their interest in RHIOs, open source software, and the prospect of an Open RHIN offering. While supporters were excited about an Open RHIN offering, some IT users and vendors were skeptical that it could succeed. Forrester concluded that there was enough interest in open source RHINs to proceed, but that an Open RHIN must be accompanied by sustained market education.

Taking the Next Step

The California HealthCare Foundation is seeking partners willing to contribute significant time and energy to an open source offering. CHCF is willing to donate the source code and facilitate the transition process to an organization willing to lead Open RHIN development. The partners would be responsible for the following:

- Re-implementing the current proprietary code base with open source code, using available open source components when available and appropriate.
- Adopting a General Public License (GPL) to attract maximum partner involvement in the Open RHIN project and facilitate adoption.
- Creating a sustainable developer/contributor community.
- Creating a sustainable business model for ongoing development, maintenance, and support of the Open RHIN applications, as well as the customers using the technology.

As with any new initiative, the partners need a pilot community interested in implementing the Open RHIN application. A successful project would require a group of highly engaged and committed organizations willing to support Open RHIN and customers willing to adopt the new technology to improve clinical decision-making throughout their community.

Appendix C

Agenda for Open Source RHIO Forum

November 15, 2006, San Francisco, California

Open Source RHIO Forum
Sponsored by
The California HealthCare Foundation and CalRHIO

November 15, 2006
Omni San Francisco Hotel
500 California Street
San Francisco, CA

AGENDA

Registration	**9:00 – 10:00**
I. Welcome and Introductions - Sam Karp (California HealthCare Foundation) - Lori Hack (CalRHIO)	**10:00 – 10:15**
II. Opening Remarks: Open Source and Health IT - Keynote speaker: Ken Kizer (Medsphere)	**10:15 – 10:45**
III. Toward an Open Source Regional Health Information Network: Findings and Analysis from the Field - John Rymer (Forrester Research)	**10:45 – 11:30**
IV. User Community Requirements & Considerations Facilitator: Lori Evans (Manatt Health Solutions) - Lyman Dennis (Partnership HealthPlan of California) - Katherine Johnson (Long Beach) - David Katz (Smart Valley) - Larry Ozeran (Clinical Informatics) - Greg Wenneson (Mendocino SHARE)	**11:30 – 12:30**
V. Lunch	**12:30 – 1:00**
VI. Open RHIN Approach for RHIOs – Creating an O/S community Facilitator: Walter Sujansky (Sujansky & Associates) - Tom Jones (Tolven) - Skip McGaughey (Eclipse Foundation) - Eishay Smith (IBM)	**1:00 – 2:00**
VII. Opportunities and Risks of Open RHIN Approach for RHIOs - Facilitator: Jonah Frohlich (California HealthCare Foundation)	**2:00 – 2:45**
VIII. Wrap Up and Next steps	**2:45 – 3:00**

Appendix D

INPC Notes from OS-CDE Expert Panel

Friday March 9, 2007, Walnut Creek, California

INPC

From OpenEMPI Wiki

Open Source Enterprise or Community Master Patient Index Project
Main Page
+ Expert Panel > **INPC** :: CDE Overview :: Fast Path
+ Requirements Document > Software_Requirements_Specification
+ Data Elements >
+ Proposed Architecture >
+ CDE Source Information >
+ References >
+ AMIA >

Indiana Network for Patient Care

- Regenstrief founded to computerize healthcare and improve efficiencies
- Regenstrief Medical Record System (RMRS) started by Clem McDonald in 1972, now the world's oldest EMR in continuous use
- INPC Repository built as a federated infrastructure using logically separated data vaults
- Each participating health care facility uses their own information system, and sends content via HL7 messaging to INPC
- Terminology mapping is the greatest barrier to semantic interoperability (INPC has 5 full time mappers handling exceptions derived from a volume of about 500,000 HL7 messages per day)
- Sites are in the patient care business, not the semantic interoperability business
- First use case was emergency room reporting to improve patient care, which resulted in reduced duplicate tests, documented in a $26 per visit reduction in cost -- *Overhage, et. al., Annals of Emergency Medicine. 2002)*
- Second use case was public health reporting; documented a 400% increase in reportable events -- submitted to AMPH for publication
- INPC is curator of data on behalf of local institutions; patient requests are the responsibility of local institutions

INPC Architecture Challenges

- LOINC for observations
- Freetext for organism "answers" (every clinical test has a question and an answer; LOINC models questions, SNOMED models non-numeric answers; standardizing the question is necessary in order to contextualize the answer)
- 7,000 ED visits/day = 10 MB of data; health care (excluding images) is not memory intensive -- low latency is a priority, but bandwidth, memory & cpu are not primary usability constraints
- the quality of a field is due to its discrimination capability -- see Shanon's Entropy (how surprised are you to see that value; a uniformly distributed variable has the highest entropy)
- User login and access is locally controlled by each participating institution
- Treat NHIN query as just another query against the global index
- Uses a known "don't link" list to defend against twins (i.e., in 9,000,000 name database Regenstrief has identified 400 twins that can be logically identified as the same patient -- Ronald & Donald Smith, Natalia and Nataltia Jones, etc.)
- MRN is not included in the identity correlation fields
- HL7 -- v2 is available and can structurally represent most coded elements; v3 while richer is not in general use. The key field is the payload (PID, OBX, etc.).

Record Linkage

- What are the core algorithms that allow matching?
- Trails -- multiple zip codes represent partial matches to a single patient
- INPC Linkage presentation *(Upload file and link to it here after Shaun sends it)*
- Global Patient Index -- one record per assigned medical record number per institution
- Trusting the institutional source of the data, making automated decisions based on the data
- Fuzzy deterministic matching (longest common substring algorithm allowing looseness on the agreement)
- Algorithm tuned to fiercely defend against false positives; racheting up specificity at the expense of sensitivity

Retrieved from "http://wiki.openempi.org/openempiwiki/index.php/INPC"

- This page was last modified 15:20, 23 March 2007.
- This page has been accessed 139 times.
-
- Privacy policy
- About OpenEMPI Wiki
- Disclaimers

Appendix E

CDE Notes from OS-CDE Expert Panel

Friday March 9, 2007, Walnut Creek, California

CDE Overview

From OpenEMPI Wiki

Open Source Enterprise or Community Master Patient Index Project
Main Page
+ Expert Panel > INPC :: **CDE Overview** :: Fast Path
+ Requirements Document > Software_Requirements_Specification
+ Data Elements >
+ Proposed Architecture >
+ CDE Source Information >
+ References >
+ AMIA >

Meeting with Jim Hazen, CareScience
Friday March 9, 2007, John Muir Health System, Walnut Creek, California

Contents

Use Case

There are two primary features of the CDE Open Source Use Case

- The MPI contains known patient identities, which can be queried, and can return answers to queries
- New data can be published to the MPI

A simple portal view to access these features is within scope, while all other software and business process features of a health information exchange (e.g., Global Provider Index, user authentication, etc.) are out of scope for this focused eight-week project

Identity Correlation Service (ICS)

- firstname, lastname, home address, dob, birthplace, ssn, phone, gender, maiden name, drivers license
- Stored in a flat database featuring an alias for each patient identity record
- ICS takes aliases and assembles them into logical groups (though "alias" does not occur in the code)
- Queries are hosted by the portal
- Narrows it down to a series of aliases that are assembled as one logical person seen at "x" different health care sites -- though each alias corresponds to a source system
- Viewed as a series of results feeds
- Scores for possibility of a match are analyzed, producing a master score
- High master score = merge; medium master score = offered to user as possible match; low master score = recommendation to review can be published if appropriate
- Jim recommends dumping Neural Net for a deterministic rules engine
- If an existing patient registers at the hospital with a new last name (e.g., married name) then a second record is created
- Multiple likely persons can be returned in response to a query, and the results can be assembled longitudinally even if they are not all the same person
- Persistent links can be asserted by a merge process at the global alias level, with a physician user discriminating against a list containing false positives?
- Securing the VIP process is a hard & intractable problem and is probably best handled with auditing and appropriate response to abuse of access privileges (privacy is a user expectation, confidentiality is a record condition)
- Correlation scores on each demographic field are normalized over a range from 0 to 1, then totals are averaged and records with a score of 0.9 or greater are merged at the global alias level
- Merging process due to a situation where one hospital in Santa Barbara dumped 200,000 unmatched records in the system
- System design difficulties due to the issue of name space collisions caused by sub-enterprise system variances (GIGO)
- **IST** is a licensed module which is called for string comparison to return a value, may want to dump it
- Shaun's three string comparators
  - Jaro-Winkler [1] (http://en.wikipedia.org/wiki/Jaro-Winkler)

    - Levenstein Edit Distance [2] (http://en.wikipedia.org/wiki/Levenstein_distance)
    - Longest common substring[3] (http://en.wikipedia.org/wiki/Longest_common_substring)
- **Person IDentity Service** (PIDS Match) calls a second service to assert a correlation
- PIDS is the nexus of flow control for asserting identity correlations

ILS

- Locator service is aware of remote repositories
- ILS query to CDR is a web services POST query
- Response to a query of remote systems is typically a list of URLs
- Central ILS assembles the remote results and assembles it for the portal

End of Day Suggestions

- Drop the neural net
- Drop the custom data layer (e.g., use hibernate)
- Make data entities more closely match actual database schema, to mimic the correct flow control of data -- fix data model, use POJOs
- "Person.java" can be dropped
- Start with database, use entity releationship matcher, generate classes based on the actual schema
- Adjust the domain model to match the database, not the code
- Find FOSS soundslike engine
- Find FOSS rules engine

Wiki Help

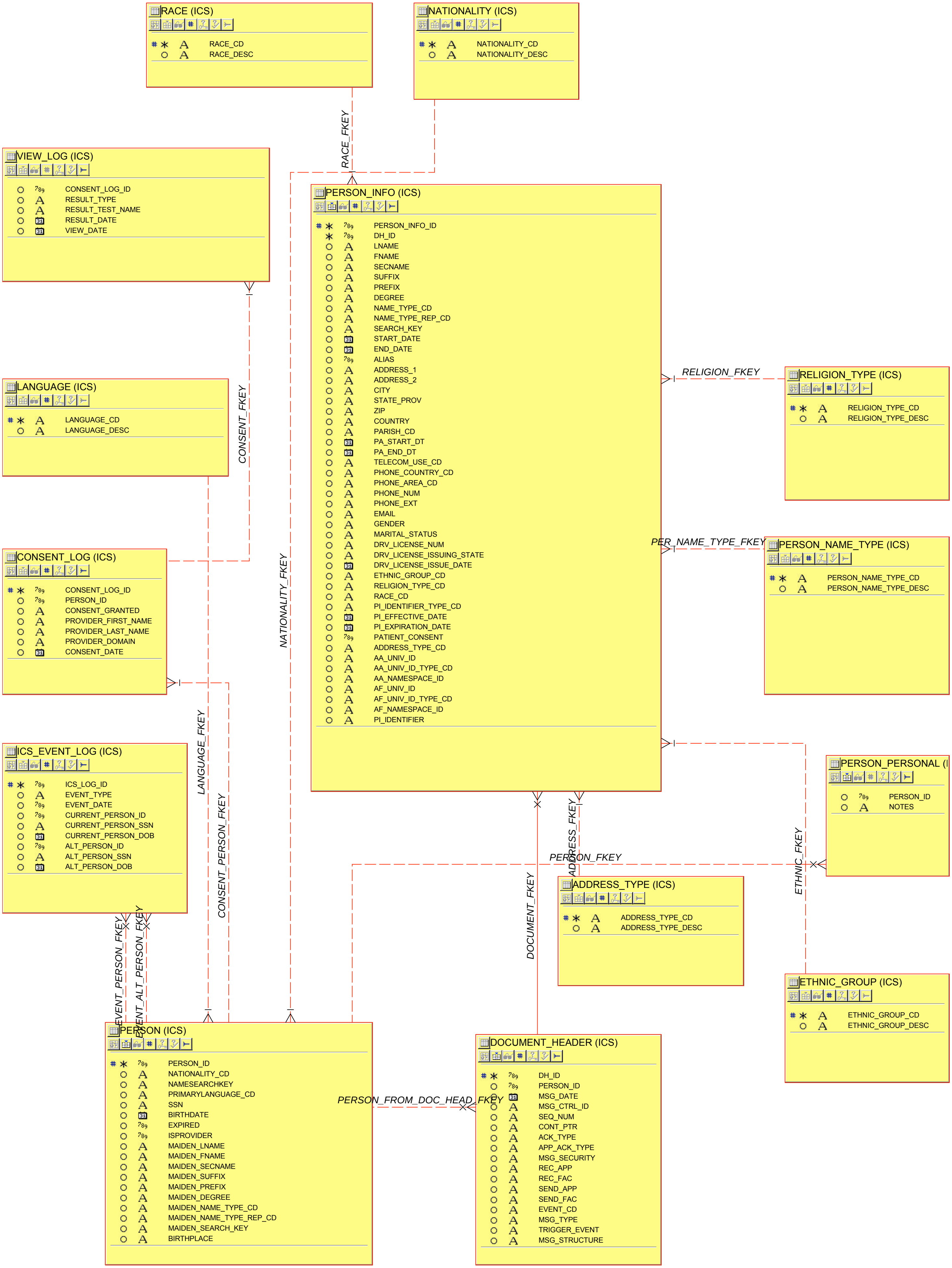
Retrieved from "http://wiki.openempi.org/openempiwiki/index.php/CDE_Overview"

- This page was last modified 15:20, 23 March 2007.
- This page has been accessed 119 times.
-
- Privacy policy
- About OpenEMPI Wiki
- Disclaimers

Appendix F

ICS Database Schema

From CareScience


RACE (ICS)
RACE_CD
RACE_DESC
NATIONALITY (ICS)
NATIONALITY_CD
NATIONALITY_DESC
RACE_FKEY
VIEW_LOG (ICS)
CONSENT_LOG_ID
RESULT_TYPE
RESULT_TEST_NAME
RESULT_DATE
VIEW_DATE
PERSON_INFO (ICS)
PERSON_INFO_ID
DH_ID
LNAME
FNAME
SECNAME
SUFFIX
PREFIX
DEGREE
NAME_TYPE_CD
NAME_TYPE_REP_CD
SEARCH_KEY
START_DATE
END_DATE
ALIAS
ADDRESS_1
ADDRESS_2
CITY
STATE_PROV
ZIP
COUNTRY
PARISH_CD
PA_START_DT
PA_END_DT
TELECOM_USE_CD
PHONE_COUNTRY_CD
PHONE_AREA_CD
PHONE_NUM
PHONE_EXT
EMAIL
GENDER
MARITAL_STATUS
DRV_LICENSE_NUM
DRV_LICENSE_ISSUING_STATE
DRV_LICENSE_ISSUE_DATE
ETHNIC_GROUP_CD
RELIGION_TYPE_CD
RACE_CD
PI_IDENTIFIER_TYPE_CD
PI_EFFECTIVE_DATE
PI_EXPIRATION_DATE
PATIENT_CONSENT
ADDRESS_TYPE_CD
AA_UNIV_ID
AA_UNIV_ID_TYPE_CD
AA_NAMESPACE_ID
AF_UNIV_ID
AF_UNIV_ID_TYPE_CD
AF_NAMESPACE_ID
PI_IDENTIFIER
LANGUAGE (ICS)
LANGUAGE_CD
LANGUAGE_DESC
CONSENT_FKEY
RELIGION_FKEY
RELIGION_TYPE (ICS)
RELIGION_TYPE_CD
RELIGION_TYPE_DESC
CONSENT_LOG (ICS)
CONSENT_LOG_ID
PERSON_ID
CONSENT_GRANTED
PROVIDER_FIRST_NAME
PROVIDER_LAST_NAME
PROVIDER_DOMAIN
CONSENT_DATE
NATIONALITY_FKEY
PER_NAME_TYPE_FKEY
PERSON_NAME_TYPE (ICS)
PERSON_NAME_TYPE_CD
PERSON_NAME_TYPE_DESC
ICS_EVENT_LOG (ICS)
ICS_LOG_ID
EVENT_TYPE
EVENT_DATE
CURRENT_PERSON_ID
CURRENT_PERSON_SSN
CURRENT_PERSON_DOB
ALT_PERSON_ID
ALT_PERSON_SSN
ALT_PERSON_DOB
LANGUAGE_FKEY
CONSENT_PERSON_FKEY
PERSON_PERSONAL (I
PERSON_ID
NOTES
ADDRESS_FKEY
PERSON_FKEY
ETHNIC_FKEY
DOCUMENT_FKEY
ADDRESS_TYPE (ICS)
ADDRESS_TYPE_CD
ADDRESS_TYPE_DESC
EVENT_PERSON_FKEY
EVENT_ALT_PERSON_FKEY
ETHNIC_GROUP (ICS)
ETHNIC_GROUP_CD
ETHNIC_GROUP_DESC
PERSON (ICS)
PERSON_ID
NATIONALITY_CD
NAMESEARCHKEY
PRIMARYLANGUAGE_CD
SSN
BIRTHDATE
EXPIRED
ISPROVIDER
MAIDEN_LNAME
MAIDEN_FNAME
MAIDEN_SECNAME
MAIDEN_SUFFIX
MAIDEN_PREFIX
MAIDEN_DEGREE
MAIDEN_NAME_TYPE_CD
MAIDEN_NAME_TYPE_REP_CD
MAIDEN_SEARCH_KEY
BIRTHPLACE
PERSON_FROM_DOC_HEAD_FKEY
DOCUMENT_HEADER (ICS)
DH_ID
PERSON_ID
MSG_DATE
MSG_CTRL_ID
SEQ_NUM
CONT_PTR
ACK_TYPE
APP_ACK_TYPE
MSG_SECURITY
REC_APP
REC_FAC
SEND_APP
SEND_FAC
EVENT_CD
MSG_TYPE
TRIGGER_EVENT
MSG_STRUCTURE

Appendix G

Loading Data

From OS-CDE wiki

Loading Data

From OpenEMPI Wiki

A framework was developed for loading test sample data into the system. The code resides in package org.openempi.ics.loader. The following class diagram Media:LoaderFrameworkClassDiagram.jpg and sequence diagram Media:LoaderFrameworkSequenceDiagram.jpg describe the organization of the package and the operation of the framework. There are two scripts checked into Subversion that can be used for loading the CDC dataset and the 3000 entry synthetic dataset that Shaun provided us. The scripts are named accordingly. This is what the 3000 entry synthetic dataset loader script looks like.

```
#!/bin/bash
IT=`pwd`
 OPENEMPI=$IT/build/dist/openempi-ejb.jar
 JNPSERVER=/opt/jboss-4.0.5.GA/server/default/lib/jnpserver.jar
 JBOSSCOMMON=/opt/jboss-4.0.5.GA/lib/jboss-common.jar
 CLASSES=$IT/build/classes
 LIB=$IT/lib/third-party
 LOG4J=$LIB/log4j-1.2.8.jar
 JBOSSALL=$LIB/jbossall-client.jar
 COMMONS=$LIB/commons-lang-2.1.jar
CONF=conf
 CP=$JBOSSCOMMON:$LOG4J:$JBOSSALL:$CLASSES:$COMMONS:$CONF:$LIB:$JNPSERVER:$OPENEMPI
echo $CP
 java -classpath $CP org.openempi.ics.loader.FileLoaderManager "data/test_3000.txt" nominal
```

The scripts should be run from the root directory of your openempi tree. Both Eclipse and the ant script have been configured to compile Java source files into the build/classes directory so the script expects to find the compiled classes that comprise the file loader there. The FileLoaderManager class does all the work of creating the appropriate loader through the Factory and invoking the parse method that processes the file and loads it into a running instance of the OpenEMPI application. The first parameter is the relative filename to the data file that needs to be loaded and the second parameter is the name of an alias that identifies the loader that should be used to load it. The framework has been designed to be extensible so to load a new file you extended the class AbstractFileLoader and add a new alias to the factory class.

Retrieved from "http://wiki.openempi.org/openempiwiki/index.php/Loading_Data"

- This page was last modified 13:43, 2 May 2007.
- This page has been accessed 20 times.
-
- Privacy policy
- About OpenEMPI Wiki
- Disclaimers