What constitutes an OpenHIE CR?

Overview

Accurate and efficient unique identification of patients is an essential function for a fully realized eHealth architecture. A client registry (CR) is designed to support patient identity management. The OpenHIE CR community seeks to foster innovative technology that provides accurate, reliable and stable identification and de-duplication of individuals and other entities in a variety of contexts, particularly resource-constrained settings.

Standards and Workflows

A core principle of the OpenHIE architecture is to allow the various infrastructure services (such as the CR) to be interchangeable. To support this, the OpenHIE Standards and Profiles use by the Client Registry are outlined in the workflows below.

To be OHIE CR component, the CR application must be able to support the OHIE workflows listed below. Implementations may support only the workflows needed to support their use case:

a. Create patient demographic record workflow
b. Update patient demographic record workflow
c. Query patient demographic records by identifier workflow
d. Query patient demographic records by demographics workflow

A core principle of the OpenHIE architecture is to allow the various infrastructure services (such as the CR) to be interchangeable. See: OpenHIE Standards and Profiles.

Recommended Functional Requirements

Depending upon the desired use case(s), the system may support many or all of these functional features.

a. **Configurable Entity matching** - A service to assist in identifying duplicate patients
    i. The rules for determining whether two records match each other should be configurable.(e.g., ability to use both statistical and/or rules based, etc.)
    ii. The blocking strategy for loading potential matches before the matching rules are applied should be configurable.
    iii. Any configurable component should have an interface so that advanced users can write their own implementation from scratch if desired.
    iv. Any interface should have at least one default implementation.
    v. The default implementation should be flexible and configurable so that non-programmers can adjust it to meet their needs.
    vi. To the extent possible, CR system configuration information should be managed using consistent and easy to access methods, such as a database, properties files, or XML files).
    vii. It should allow “wizard-based” or “guided” setup of matching rules
b. **Patient Linking and De-duplication**
    1. The system should implement accurate and efficient patient linking and de-duplication methods.
    2. It should provide an easy to use and intuitive way to see merge/linkage operations
    3. It should allow an easy to use and intuitive way of manually accepting or rejecting merge suggestions, with the ability to choose fields from either record to be merged
c. **Configure and monitor inbound/outbound transactions.**
    i. The system must have the capacity to record receipt and transmission of transactions.
d. **Synchronize client IDs with a SHR.** (Support patient-level clinical data OHIE workflow)
e. **UI to search patients, manually edit** (e.g., create, update, merge, split, and deprecate)
f. **UI to review and manually adjudicate uncertain** (“potential”) matches, and override incorrect matches.
g. **Configurable Attributes** -
    i. The attributes that form a patient record and are used for matching should be configurable.
    ii. The implementation can include an example/default patient schema.
    iii. It should be easy to add attributes to the schema.
    iv. It should also be easy to remove attributes from the default model (or start over from scratch).
h. **Error Management:** Ensure that error handling comprehensively captures and logs all related exceptions, and to the extent possible, shows relationships between exceptions.
i. **Logging:** Logging should be consistent; it should be easy to find information in the log.
j. **Privacy/Security:** The system should have functions including user management and access controls.
k. **Pediatric Option:** it is mandatory for an OpenHIE-conformant CR to support the PIX “Pediatric Option”

Architecture Characteristics

1. **The following are the recommended core software architectural characteristics of a CR:**
    a. System configuration: Defines entity features, identity sources, decision models, business process rules.
    b. Data persistence: Supports reliable low latency, high bandwidth access to potentially large volumes of patient identity information.
    c. Object Representations of Patients
        i. An incoming patient record should not need to be converted into many different formats prior to storing in a database.
        ii. A process like this should be sufficient:
            1. An HL7 message is received, representing the patient as a PID segment within the text
            2. An HL7 library (like HAPI) parses the message, creating an instance of a PID Java object
            3. The Client Registry converts the PID object into an instance of its own patient/entity/whatever class
            4. That object is loaded into the database

Scalability

Scales to millions of patients: Client registries are increasingly expected to support unique at edification of large patient populations. The CR design must support efficient operation (sub-second response time to identity queries) when managing millions of patients.

Support of OpenHIE Non-functional Requirements

An OpenHIE component should consider the following OpenHIE Non-Functional Requirements - Draft