

Implementing the XDS Infrastructure

Bill Majurski

IT Infrastructure

National Institute of Standards
and Technology



- **Repository Actor**
- **Registry Actor**

- **XDS.a**
- **XDS.b**

- **2 Styles**
 - File Server
 - EHR
- **(3) XDS Interfaces**

Repository as a File Server

- Accepts files (documents) via Provide and Register transaction
- Stores files for later retrieval

EHR as Repository

- **Exposing content as Documents**
- **Must generate size, hash for document**
- **Must offer Document via URI**
 - **Must match size and hash**

Repository in XDS.a

- **SOAP (Register transaction)**
 - HTTP Post
 - Body is XML
 - Top level element is SOAP-Envelope
- **SOAP with Attachments (Provide and Register transaction)**
 - Body is MIME Multipart
 - First part is SOAP-Envelope
 - Additional parts are Documents

Document Retrieve

- **HTTP GET**
- **Be careful with MIME Type and size returned in HTTP Header**
- **Size and Hash of body MUST match what was sent to Registry**

Web Services interface changes

- **Provide and Register transaction**
 - Namespaces
 - MTOM
- **Register transaction**
 - Repository ID
 - uniqueID

...Repository in XDS.b

- **Retrieve**
 - **Repository ID**
 - **uniqueID**
 - **Web service**
 - **Multiple documents**

XDS.a Register transaction

- Send updated metadata to Registry
- Via SOAP/WS message

XDS.b Register transaction

- **Send metadata to Registry**
- **Web service interface**
- **ebRS/ebRIM version 3.0**
- **Can publish both**
 - **URI (XDS.a)**
 - **repositoryID / uniqueID (XDS.b)**

Registry

- **Functionality**
- **Approaches**
- **Interfaces**
- **Validation**

Registry Functionality

- **Accept and Store metadata**
- **Service Queries for metadata**
- **Hard Parts**
 - Parsing and generating metadata
 - Very complex XML
 - Validating metadata
 - Queries have many parameters

XDS.a vs XDS.b

- Details of SOAP wrapper
- ebRIM/ebRS 2.1 vs 3.0
- Differentiate .a vs .b requests based on XML namespace
- Stored query adds new optional parms, be able to ignore them

Approaches to building a Registry

- **Start with open source ebXML Registry**
 - Source Forge project ‘ebxmlrr’
 - Add XDS functionality
 - Source Forge project ‘IheOs’ is example
- **Build on top of favorite database**

SQL Query (optional)

- Raw SQL
- Does assert database schema

Registry - Validation

- **Metadata is archival content**
 - As a community we must insure that metadata is ‘correct’
- **‘Correct’**
 - Key structural issues
 - Interpretable by any consumer
- **Heart of XDS**

Registry - Validation (2)

- Registry Adaptor - functionality XDS adds to Registry standard
- Specifies validation rules

- **Pre-Connectathon ('MESA' tests)**
 - **Registry/Repository**
 - Command line tools (Java)
 - Large corpus of test data
 - Submit (upload) results file to Kudu
 - **Source/Consumer**
 - Test against Public Registry
- **Connectathon**
 - Use Stored Query to validate XDS tests

Testing Related Profiles

- **ATNA testing is integrated**
 - Node authentication
 - Audit log
 - At Connectathon - may want to be able to log to multiple Audit Repositories
- **Patient Identity Feed testing**
 - Relies on PIX tests from MIR
- **XUA**
 - Simple assertions to be tested in Stored Query and Retrieve Document Set
- **Consistent Time**
 - Not tested as part of XDS

Testing

- **New tools - easier to use**
- **New tests - add detail**

- IHE Wiki articles
 - <http://wiki.ihe.net>
- Open Source implementation
 - Source Forge project 'IheOs'
- Public Registry server
 - Runs 'IheOs' implementation

...Resources

- <http://ihexds.nist.gov>
 - Home page for XDS testing
 - Test definitions
 - FAQ
 - Documentation
- [**bill@nist.gov**](mailto:bill@nist.gov)