Using NAACCR XML with Relational Databases

Isaac Hands, MPH
Lead Software Architect, Kentucky Cancer Registry
Chair, NAACCR XML Data Exchange Workgroup
NAACCR Annual Conference 2016
Background: NAACCR XML Data Exchange Standard

• NAACCR XML Version 1.0 Approved by NAACCR Board Nov. 2015

• Patient-Centered Data Model

  <Patient>
  <Item naaccrNum="dateOfBirth">19600117</Item>
  <Item naaccrId="sex">1</Item>
  ...

  <Tumor>
  <Item naaccrNum="dateOfDiagnosis">20100314</Item>
  <Item naaccrNum="primarySite">C340</Item>
  ...

http://naaccrxml.org
Background: NAACCR XML Data Exchange Standard

• Allows for User-Defined fields that fit within fixed-width record
  • State Requestor Fields, NPCR Fields, Reserved Fields
  • Create User-Defined Data Dictionary with Item Definitions

• Allows XML “Extensions” that go beyond the fixed width file

• Software tools have been created to read and write NAACCR XML
  • Created by IMS, Inc.
  • Available on GitHub as open source project

• Great advantages over fixed-width format, the future of NAACCR Data Exchange

http://naaccrxml.org
Challenges Transitioning to NAACCR XML

• All cancer registries use relational database systems, not obvious how to map XML data into a relational model

• A lot of custom software built around the fixed-width format

• Data analytics and statistical reporting is based on “flat” data, dealing with XML may be unfamiliar (SAS)
Typical Data Exchange Scenarios

Scenario 1: Sender and receiver have very different data management systems and storage methods
  • Registry to Out-of-State Registry
  • Registry to Data Submission Agency

Scenario 2: Sender and receiver want to setup an ongoing data exchange with the same method of storing data
  • Long-term research study
Mapping XML to Databases

<XML>
- Flexible
- Extensible
- Everything is a String of Characters

Relational Database:
- Difficult to Change
- Static Design
- Strongly Typed

http://naaccrxml.org
## Common Methods of Mapping XML to a Relational Database

<table>
<thead>
<tr>
<th>Method</th>
<th>Pro</th>
<th>Con</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Direct Storage of XML</strong></td>
<td>• Automated</td>
<td>• Vendor Lock-In</td>
</tr>
<tr>
<td>• Graph databases</td>
<td>• Fast</td>
<td>• Not Good for a Diverse Community</td>
</tr>
<tr>
<td>• XML columns</td>
<td>• Flexible</td>
<td>• Can be difficult to analyze data</td>
</tr>
<tr>
<td><strong>2. Schema-driven Mapping</strong></td>
<td>• Mostly Automated</td>
<td>• Loss of flexibility</td>
</tr>
<tr>
<td>• XSD</td>
<td>• Self-Documenting</td>
<td>• Limited by schema standard</td>
</tr>
<tr>
<td>• DTD</td>
<td>• Supported</td>
<td></td>
</tr>
<tr>
<td>• RelaxNG</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3. Custom Software</strong></td>
<td>Do whatever you want</td>
<td>Expensive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Support</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Maintenance</td>
</tr>
</tbody>
</table>
Objectives

1. Read a NAACCR XML file and write to a relational database
   • Auto-Create database if it does not exist (along with tables and columns)

2. Create a NAACCR XML file from a relational database
   • Use SQL to specify patients or tumor to extract from database
   • Performance: Batch reads from database for large queries

3. Support custom data elements and extensions

4. Use existing NAACCR XML software tools as much as possible

5. Support Multiple Relational Databases
Methods: Define a database table model

3 Item Containers in NAACCR XML

At least 3 Tables in Database, Items Become Columns
Methods: Define a way to configure the database table model

```xml
<PatientTable connection="cpdms-default-mysql" tableName="naacrr_patient">
    ...
</PatientTable>

<TumorTable connection="cpdms-default-mysql" tableName="naacrr_tumor">
    <NaacrrItemColumn naaccrlId="dateOfDiagnosis" columnName="DiagDate"/>
    <ExtraColumn columnName="PatID" foreignTableColumnName="AutoID"/>
    <TableListener javaClass="edu.uky.kcr.naxml.listener.CpdmsTableListener"/>
    <ExtraTable connection="cpdms-postgres" tableName="case_other"/>
```

http://naacrrxml.org
DEMO
Results

• Several relational database systems supported:
  • PostgreSQL, MySQL, Sqlite

• Custom NAACCR Data Elements are supported

• Existing NAACCR XML Software needed modifications
Conclusions

• Custom software for mapping from NAACCR XML to Relational Databases is a viable option
• NAACCR could oversee creation and maintenance of standard software tools
• Other relational databases could be supported
• Existing NAACCR XML Software could be expanded to support this solution
• Graphical User Interface would be nice
Contact Information

Isaac Hands
Kentucky Cancer Registry
Chair, NAACCR XML Data Exchange Workgroup
isaac@kcr.uky.edu