Design and Development of a Modular Harvester for Data Ingest

August 12, 2010
New York Hall of Science

VIVO: Enabling National Networking of Scientists is supported by NIH Award U24 RR029822.
VIVO Collaboration

Who are we?

– Software Engineering for UF College of Medicine
– Biomedical Informatics
– Interface and Packaging teams for VIVO
Semantic Web

Old Method
Only Human Readable Data
Tagging
Search Engine Crawlers

New Method
Human Readable Display
Machine Readable Metadata
What is VIVO

- Is/not “Facebook” for scientists
  - (IS) Enables the discovery of researchers, papers, grants, projects all across the university
  - (IS NOT) Presents information about researchers from authoritative sources
Its all about the Data

• Authoritative Data
  – Systems of Records
    • Employers
    • Publishers
    • Government Organizations
    • Award Funding Agencies
    • Professional Societies
    • Scholastic Institutions
Where do we get the data

- Human Resources
  - Grants Management
- Faculty Reporting
  - Project Management
- Local Data Sources
- National Data Aggregators
- CiteSeer
- Grantsfire
- National Data Repositories
- PubMed.gov
- Grants.Gov
- National Organizations
- NSF
- National Institutes of Health
- AAAS
- National networking of scientists
Manual Data Ingesting

• Collection of tools for VIVO administrators to manipulate rdf models and import xml and csv data into their vivo installation
Where do we start

• Data Source Target
• Specifications of Work
• Detailed Designs
Data Source Target

– Maintained by the National Library of Medicine
– 20+ million citations going back to 1865
– Consolidated Appropriations Act of 2007
  • Publicly available copies of all publications from Grants
– MeSH – Medical Subject Headings
– Interface for Automated Data Access
Specifications

• Functional Specification
  – Use Cases
  – How the user interacts with the program
  – sf.net/apps mediawiki/vivo/Functional_Specification

• Technical Specifications
  – How the program works
  – Overview of the system architecture
  – sf.net/apps mediawiki/vivo/Technical_Specification
Detailed Designs

sf.net/apps/mediawiki/vivo/Category:Detailed_Technical_Designs

**Score**

**Scoring**

- Arch Flow diagram
- Scoring java
  - 2.1 Flow
  - 2.2 Arguments
  - 2.3 Functions:
    - 2.3.1 CommitResultSet
    - 2.3.2 replaceResource
    - 2.3.3 recursiveSanitizeBuild
    - 2.3.4 ExecuteQuery
    - 2.3.5 ParseAndScore
    - 2.3.6 ExitAndMatch
    - 2.3.7 Machine Learning
    - 2.3.8 Natural Language Processing
    - 2.3.9 Regex Search

**Arch Flow diagram**

**Scoring.java**

Method used to score incoming data. Data is assumed to be in a VIVO-like ontology and stored in a Java model. Method can call any combination of scoring algorithms. Scoring function will attempt to match data to authors in VIVO. Several algorithms can be utilized to determine when the match will be inserted into VIVO. In addition, data produced by this method can be stored in a separate scoring model.

**Flow:**

1. Executed via command line
2. Read in command line parameters containing algorithms to run and the data to score
3. Call each scoring algorithm utilizing passed parameters for each call
4. Return Model containing scored statements to stdout unless output argument is supplied

Each scoring method will return a model consisting of statements that can be later passed back to store into VIVO. In addition, each method will accept an input model with statements to score. This input model is designed to allow sharing amongst all methods. Initially, each statement will only be scored by each method in order until a match is found or all methods of disambiguation have been exhausted. In this case, the remaining statements are simply discarded.
Development Cycle

• Best Attempt Agile Development
  – 3 week development cycles
  – Kickoff Planning Meeting
  – Daily stand-up meetings
  – Release Early / Release Often
    • Customer Feedback
    • “With Many Eyes all Bugs are Shallow”
      Cathedral and the Bazaar
Original Design – Milestone 1

External Data Store → Fetch → Translate → Score → VIVO

Controller
Milestone II

External Data Store

Controller

IO

Fetch

Translate

Score

Qualify

Transfer

VIVO
Milestone III

- Task
- Record Handler
- Fetch
- Translate
- Transfer
- Qualify
- Score
- VIVO
- External Data Store
Milestone IV

• Refinement of the Record Handler
  – Meta Data
  – MD5 Checksum Comparison

• Removal of the Task Framework

• Addition of Command Line / Config File Parsers

• Sanity of Execution

• Refinement of Relational Database
Release Dates

• May 10th – Milestone I
• June 4th – Milestone II
• July 10th – Milestone III
• August 6th – Milestone IV
What does it look like now

- External Data Store
- Fetch
- Translate
- Transfer
- Utilities
- RDF
- Score
- Qualify
- VIVO
Multi-Server Design

- mod.server.institution
  - Translate
  - Qualify
- disambig.server.institution
  - Scoring
- fileTrans.server.institution
  - Transfer
  - Fetch

- VIVO
- VIVO Database
- External Data Sources
Utilities

- **Record Handler**
  - File
  - Relational Database
  - Semantic Data Store
Utilities

• ArgParser
  – Wrapper/Extender of CLI
  – Command Line Args / Configuration Arguments
PubMed Fetch

• Medline Data in XML
• Soap Interface – Entrez Utility
• Data Limits
  – Batch Methods on Weekends and 9pm to 5am
  – 3 requests per second
Design and Development of a Modular Harvester for Data Ingest

**PubMed Fetch**

### Configuration

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Task type="org.vivoweb.ingest.fetch.PubmedSOAPFetch">
  <Param id="email">hainesc@ctrip.ufl.edu</Param>
  <Param id="location">University of Florida</Param>
  <Param id="output">config/recordHandlers/PubmedXMLRecordHandler.xml</Param>
  <Param id="termSearch">1:8000 [dp]</Param>
  <Param id="numRecords">20</Param>
  <Param id="batchSize">1000</Param>
</Task>
```

### Record Handlers

```xml
<?xml version="1.0" encoding="UTF-8"?>
<RecordHandler type="org.vivoweb.ingest.util.repo.TextFileRecordHandler">
  <Param name="fileDir">XMLVault/PubmedRDF</Param>
</RecordHandler>
```

### Execution

```
java -cp bin/ingest-0.4.4.jar;bin/dependency/*
org.vivoweb.ingest.fetch.PubmedSOAPFetch -X config/tasks/PubmedFetch.xml
```
Open Archive Initiative

• Interoperability Standards
  – Open Access & Institutional Repositories
  – OAI-PMH: Protocol for Metadata Harvesting
  – Default Format is Dublin Core XML
  – OpenArchives.org
OAI Fetch

Configuration

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Task type="org.vivoweb.ingest.fetch.OAIHarvest">
  <Param id="address">digital.library.upenn.edu/webbin/OAI-celebration</Param>
  <Param id="startDate">2000-01-01</Param>
  <Param id="endDate">2010-01-01</Param>
  <Param id="repositoryConfig">config/recordHandlers/OAIXMLRecordHandler.xml</Param>
</Task>
```

Execution

```
>java -cp bin/ingest-0.4.4.jar;bin/dependency/* org.vivoweb.ingest.fetch.OAIFetch -X config/tasks/OAIFetch.xml
```
Additional Fetch Sources

Relational Data Source Configuration

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Task type="org.vivoweb.ingest.fetch.JDBCFetch">
  <Param id="driver">com.mysql.jdbc.Driver</Param>
  <Param id="connection">jdbc:mysql://127.0.0.1:3306/jdbctestharvest</Param>
  <Param id="username">jdbcTestHarvest</Param>
  <Param id="password">xKzQVJhnBnxYREmYK</Param>
  <Param id="output">config/recordHandlers/JDBCXMLEventRecordHandler.xml</Param>
</Task>
```
Translate

• XSLT
  – Extensible Style Sheet Transformations
  – Map XML Files to new schemas
  – Uses
    • XPath – Navigating the XML Document
    • XSL-FO – Formatting the XML Document
  – javax.xml.transform
Translate

Record Handlers

```xml
<?xml version="1.0" encoding="UTF-8"?>
<RecordHandler type="org.vivoweb.ingest.util.repo.TextFileRecordHandler">
    <Param name="fileDir">XMLVault/PubmedRDF</Param>
</RecordHandler>

<?xml version="1.0" encoding="UTF-8"?>
<RecordHandler type="org.vivoweb.ingest.util.repo.TextFileRecordHandler">
    <Param name="fileDir">XMLVault/PubmedXML</Param>
</RecordHandler>
```

Execution

```bash
java -cp bin/ingest-0.4.4.jar;bin/dependency/*
or.g.vivoweb.ingest.translate.XSLTranslator
    -i config/recordHandlers/PubmedXMLRecordHandler.xml
    -x config/datamaps/PubMedToVivo.xsl
    -o config/recordHandlers/PubmedRDFRecordHandler.xml
```
Scoring

• Exact Match
  – Unique Identifier matching
    • Core:WorkEmail

• Name Match
  – Pass in threshold (Minimum Match)
  – Hard coded to Foaf:LastName, Foaf:FirstName
Scoring

- Very Early Implementation

```xml
<model>
  <param name="dbClass">com.mysql.jdbc.Driver</param>
  <param name="dbType">MySQL</param>
  <param name="dbUrl">jdbc:mysql://127.0.0.1:3306/vivo</param>
  <param name="modelName">http://vitro.mannlib.cornell.edu/default/vitro-kb-2</param>
  <param name="dbUser">root</param>
  <param name="dbPass">v3nd3tta</param>
</model>
```

Configuration of Jena Model

```
java -cp bin/ingest-0.4.4.jar;bin/dependency/* org.vivoweb.ingest.score.Score -i config/recordHandlers/PubmedRDFRecordHandler.xml -V config/jenaModels/VIVO.xml -e workEmail -a 2
```

Execution
Qualify

• Validate and Sanitize Data
  – Truncated Data
  – Abbreviated Data

```java
java -cp bin/ingest-0.4.4.jar:bin/dependency/*
org.vivoweb.ingest.qualify.SPARQLQualify
-j config/jenaModels/VIVO.xml -n staging -t "Prof" -v "Professor" -d
http://vivoweb.org/ontology/core#Title
```

```java
java -cp bin/ingest-0.4.4.jar:bin/dependency/*
org.vivoweb.ingest.qualify.SPARQLQualify
-j config/jenaModels/VIVO.xml -n staging -r .*JAMA.*
-v "The Journal of American Medical Association"
-d http://vivoweb.org/ontology/core#Title
```

Execution
Transfer

• Move Data into and out of VIVO

```java
java -cp bin/ingest-0.4.4.jar:bin/dependency/* org.vivoweb.ingest.transfer.Transfer
    -i config/jenaModels/VIVO.xml -I staging -o config/jenaModels/VIVO.xml

java -cp bin/ingest-0.4.4.jar:bin/dependency/* org.vivoweb.ingest.transfer.Transfer
    -i config/jenaModels/VIVO.xml -I staging -d dump.rdf
```

Execution
How do you use it

• Configuration
  – Record Handler
  – Jena Models (for Qualifying and Scoring)
  – Configuration Files (optional)

• Execution
  – Schedulers (Windows Scheduler, CRON)
  – Batch/Bash Files for executing inline
Where are we going from here

• More Data Sources
Where are we going from here

• Translation of established data formats

  - vCalendar & vCard
  - eduPerson & eduOrg
  - iCalendar
Where are we going from here

- Scoring Algorithms
  - Machine Learning
  - Data Classification
  - Natural Language Processing

Publications Research & Automated Author Disambiguation: Collaboration between University of Florida and Cornell University

Break Out B – 2:00 – 3:00 PM
Where are we going from here

- **Update Process**
  - Identifying Attributes that have change
- **Integration with External Tools**
  - D2RMap
- **Documentation**
  - How do I extend the Harvester
  - How do I add my program into the Harvester Process
Where can you find out more?

• http://www.vivoweb.org
• http://vivo.sourceforge.net
  – Wiki
  – SVN development branch
  – File Repository
  – IRC Channel #VIVO at freenode
Development Resources

- Eclipse
- Subversion Repository
- Jena 2.8.4
- Commons Cli
- Log4J
- MySQL Server
Introducing the Rest of the Crew

Christopher Barnes  
Narayan Raum  
Yang Li  
Christopher Haines  
Dale Scheppler  
Nicholas Skaggs
Questions?