XORT- How to Bridge your Database and Application

Pinglei Zhou
Josh Goodman

FlyBase

January 18, 2007
Generic Modern Language Database?

My Parents: hometown dialect

My wife: Mandarine (official form of Chinese)

My son: Plain English

With colleagues: ‘chadoXML’
Introduction

• An XML-database mapping system for data exchange between DB and XML-driven application

• Developed/Supported by Pinglei Zhou at FlyBase Harvard, 0.007 version now.

• Used: All FlyBase sites

• Written in Perl

• Required perl modules:
  XML::Parser::PerlSAX
  Unicode::String
  XML::DOM
  DBI
Components

- Database & Schema
- ChadoXML Specification
- DumpSpec collections
- Tools
Highlights of Chado XML Specification

• Unique represent of specific database schema

• Get away with those internal primary key value

• Static vs. Operational

• Encoding for non-ansci characters

• Macro mechanism (object reference)
Putting it together: New FlyBase dataflow – Part 1
Putting it together: New FlyBase dataflow – Part 2

FlyBase
Harvard Univ.

Chado

FlyBase Indiana Univ.

Chado

Denormalization

XORT

ChadoXML

GFF & FASTA

XORT

XHTML
Data & Report Generation

• Content of all output files is controlled by XML dumpspecs.
  – Dumpspecs are language independent.
  – Easily readable (with knowledge of Chado structure).

• All XML transformation steps are done with XSLT v2.
  – Saxon XSLT (http://saxon.sourceforge.net/)
  – ChadoXML is split into individual chunks before XSLT processing to accommodate large file sizes.
  – Extremely fast. We can process all data for ~60,000 Drosophila genes in under 30 minutes.
Hibernate & XORT

- Hibernate didn't scale well when dealing with 5,000+ features in bulk.
- Performance tweaks for Hibernate can be quite complicated to setup for bulk operations.
- XORT is currently handling ~6 million features in production with only minor performance problems.
- XORT is much more language independent.
Support for complex transactions using XORT

Eg:

- Find all records linked to a record using dumpspec

- Merge gene x into y, each with thousands of records attached
  1. Dump all data use simple dumpspec
     <chado>
     <feature dump="all">
       <uniquename test="eq">x</uniquename>
     </feature>
     </chado>
  2. Delete feature x from DB, with triggers to clean orphan records, if necessary
  3. Edit the output xml, change uniquename x to y, then load the edited file back to DB
CHIA
(Chado Interface Application)

Java application organizes SQL and XORT functionality for internal users, eg:

- Dump chado-XML for gene regions for Apollo curation
- Organize and execute “canned” SQL queries
- Serve IDs for curators (in development)
- Dynamic browser Chado without writing SQL statement

CHIA is being designed to be extensible for adding new functionality as needed.
Limitations

- DB Schema follow certain rules
  All have internal int primary key
  All have unique key(s)

- It may take long path to retrieve certain type of data
  gene->allele->genotype->phenotype via feature_relationship

- Structure not store in memory
  Flush out data as it goes
Documentation

• Previous presentations

• Using chado to Store Genome Annotation Data

• XORT specification docs

• XORT draft (unpublished)

• GMOD case demo procedure
  All in the doc directory of XORT package
  http://www.gmod.org
Acknowledgements

- Willian Gelbart
- David Emmert
- Stan Letovsky
- Frank Smutniak
- Peili Zhang
- Haiyan Zhang
- Andy Schroeder
- Susan Russo
- Mark Zythovicz

- Victor Strelets
- Robert Wilson
- Paul Leyland

- Chris Mungall
- Mark Gibson
- Nomi Harris
- Suzanna Lewis
- Stan Letovsky
- Aubrey de Grey
- Don Gilbert

- Scott Cain
- Lincoln Stein