

XML GUS Data Loading

The Genomics Unified Schema User's and Developer's Workshop
July 7, 2005

Josef Jurek
Daphne Preuss Laboratory
Molecular Genetics and Cell Biology
The University of Chicago
jurek@cs.uchicago.edu

Terry Clark, Josef Jurek, Gregory Kettler, and Daphne Preuss,
**A Structured Interface to the Object-Oriented Genomics Unified Schema
for XML Formatted Data**, *Applied Bioinformatics*, in Press, Spring 2005.

Goals

Formulate an XML interface that includes relational database key constraint definitions

Create an XML for GUS generalized enough to input data into any table or group of tables

Regularize the traversal through that XML (syntax checking).

Allow for user/site specific processing of data.

What the User Requires

- The XMLGUS plugin, available at <http://amrit.ittc.ku.edu/flora>.
 - XML::YYLex (for XML processing)
 - XML::DOM processor (provides the lexical analysis for the parser)
 - Berkeley YACC compiler generator Perl-byacc
- A user designed XML scheme for marking up data.
- A context-free grammar or CFG. (Don't be alarmed). There are also some CFG's available at <http://flora.uchicago.edu/grammars>.
- Optional user-defined functions for additional processing of data.

An Example of User Designed XML Tags for XMLGUS

```
<gus>
<dots_nasequence depth="0">

.   <dots_sequencetype fkobj="dots::sequencetype" depth="1">
.       <name>DNA </name>
.   </dots_sequencetype>
.   <sequencetypeid pkobj="dots::sequencetype" key="sequence_type_id"/>

.   <sres_taxonname fkobj="sres::taxonname" depth="1">
.       <name>Olimarabidopsis pumila </name>
.   </sres_taxonname>
.   <taxonid pkobj="sres::taxonname" key="taxon_id"/>

.   <description>OPM18B21 Contig10 </description>
.   <sequence>
ATCGGAGTCAGGCTGGAAGACAACCTCCTCTGCGAAGTCGCGGTGAGTTTTAGT
GCATCGATGAATTTACGGATGACAACACTGTTTGTACTCTCTAAAACAACCAG
CCACCTAGCACAACAACCTTTACCCCGAATATCTTATCACATATCTTTTAAAGT
.   </sequence>
</dots_nasequence>
</gus>
```

Deriving Foreign Keys from Candidate Keys

- . <dots_sequencetype fkobj="dots::sequencetype" depth="1">
- . <name>DNA </name>
- . </dots_sequencetype>
- . <sequencetypeid pkobj="dots::sequencetype" key="sequence_type_id"/>

DoTS::NASequence (view on GUS::Model::DoTS::NASequenceImp)

column	null?	type	parent_table
na_sequence_id	no	number(10)	
sequence_version	no	number(3)	
subclass_view	no	varchar2(30)	
sequence_type_id	no	number(4)	DoTS::SequenceType
taxon_id		number(12)	SRes::Taxon
sequence		clob(4000)	
length		number(12)	
...

Example of a user designed XML for XMLGUS (Again)

```
<gus>
<dots_nasequence depth="0">
.   <dots_sequencetype fkobj="dots::sequencetype" depth="1">
.       <name>DNA </name>
.   </dots_sequencetype>
.   <sequencetypeid pkobj="dots::sequencetype" key="sequence_type_id"/>
.
.   <sres_taxonname fkobj="sres::taxonname" depth="1">
.       <name>Olimarabidopsis pumila </name>
.   </sres_taxonname>
.   <taxonid pkobj="sres::taxonname" key="taxon_id"/>
.
.   <description>OPM18B21 Contig10 </description>
.   <sequence>
ATCGGAGTCAGGCTGGAAGACAACCTCCTCTGCGAAGTCGCGGTGAGTTTTAGT
GCATCGATGAATTTACGGATGACAACACTGTTTGTACTCTCTAAAACAACCAG
CCACCTAGCACACAACCTTTACCCCGAATATCTTATCACATATCTTTTAAAGT
.   </sequence>
</dots_nasequence>
</gus>
```

Another XML Example: inserting rows into child tables

```
<gus>
<dots_nafeature depth="0">
.   <dots_externalnasequence depth="1" fkobj="dots::genefeature">
.     <name>Arabidopsis thaliana </name>

.     <sres_externaldatabaserelease depth="2" fkobj="dots::externalnasequence">
.       <sres_externaldatabase depth="3" fkobj="sres::externaldatabaserelease">
.         <lowercase_name>ncbi </lowercase_name>
.       </sres_externaldatabase>
.       <external_database_id pkobj="sres::externaldatabase" key="external_database_id"/>
.       <version>NC_003070.5 </version>
.     </sres_externaldatabaserelease>
.     <external_database_release_id pkobj="sres::externaldatabaserelease" key="external_database_release_id"/>
.   </dots_externalnasequence>
.   <na_sequence_id pkobj="dots::externalnasequence" key="na_sequence_id"/>
.   <name>misc.feature </name>
.   <dots_nalocation depth="1">
.     <start_min>1 </start_min>
.     <end_max>444 </end_max>
.     <is_reversed>0 </is_reversed>
.   </dots_nalocation>
.   <dots_nafeaturecomment depth="1">
.     <comment_string>
.       nucleotide sequence in this region was derived from BAC clone TEL1N.
.     </comment_string>
.   </dots_nafeaturecomment>
</dots_nafeature>
</gus>
```

Another Example of Deriving Foreign Keys from Candidate Keys

DoTS:ExternalNASequence is a parent of

. **SRes:ExternalDatabaseRelease** is a parent of

. **SRes:ExternalDatabase**

```
<dots_externalnasequence depth="1" fkobj="dots::genefeature">
.   <name>Arabidopsis thaliana </name>
.   <sres_externaldatabaserelease depth="2" fkobj="dots::externalnasequence">
.     <sres_externaldatabase depth="3" fkobj="sres::externaldatabaserelease">
.       <lowercase_name>ncbi </lowercase_name>
.     </sres_externaldatabase>
.     <external_database_id pkobj="sres::externaldatabase" key="external_database_id"/>
.     <version>NC_003070.5 </version>
.   </sres_externaldatabaserelease>
.   <external_database_release_id pkobj="sres::externaldatabaserelease" key="external_database_release_id"/>
</dots_externalnasequence>
<na_sequence_id pkobj="dots::externalnasequence" key="na_sequence_id"/>
```


Resolving Foreign Keys from Candidate Keys Once per File

```
<gus>
<sres_externaldataserelease depth="0" fkobj="dots::externalnasequence">
.   <sres_externaldatabase depth="1" fkobj="sres::externaldataserelease">
.     <lowercase_name>ncbi </lowercase_name>
.   </sres_externaldatabase>
.   <external_database_id pkobj="sres::externaldatabase" key="external_database_id"/>
.   <version>NC_003070.5 </version>
</sres_externaldataserelease>

<dots_externalnasequence depth="0" fkobj="dots::genefeature">
.   <external_database_release_id pkobj="sres::externaldataserelease" key="external_database_release_id"/>
.   <name>Arabidopsis thaliana </name>
</dots_externalnasequence>

<dots_nafeature depth="0">
.   <na_sequence_id pkobj="dots::externalnasequence" key="na_sequence_id"/>
.   <name>misc.feature </name>
.   <dots_nalocation depth="1">
.     <start_min>1 </start_min>
.     <end_max>444 </end_max>
.     <is_reversed>0 </is_reversed>
.   </dots_nalocation>
</dots_nafeature>

<dots_nafeature depth="0">
.   [...]
</dots_nafeature>

<dots_nafeature depth="0">
.   [...]
</dots_nafeature>
</gus>
```

The XMLGUS Context Free Grammars (CFG)

Written in YACC, compiled by Perl-byacc into PERL.

Consists principally of variables and terminals associated with GUSXML elements (table names, table attribute names).

Some pre-written XMLGUS Grammars are available from the University of Chicago at <http://flora.uchicago.edu/grammars>.

Production/Rule for Table

```
P1_DOTS_NASEQUENCE: dots_nasequence P1_DOTS_NASEQUENCE_SET _dots_nasequence
{
.   GUS::Common::Plugin::XMLGUS::process_xml_rule(
.   undef, undef,
.   "DoTS::NASequence",
.   $2->getNodeValue,
.   $1->getAttribute("pkobj"),
.   $1->getAttribute("fkobj"),
.   $1->getAttribute("key"),
.   $1->getAttribute("depth")
.   );
.   };
};
```

```
P1_DOTS_NASEQUENCE_SET:
.   P1_DOTS_NASEQUENCE_ATT |
.   P1_DOTS_NASEQUENCE_SET P1_DOTS_NASEQUENCE_ATT;
```

Production/Rule for Table Attributes

P1_DOTS_NASEQUENCE_ATT:

```
. P2_DOTS_NASEQUENCE__DESCRIPTION |  
. P2_DOTS_NASEQUENCE__LENGTH |  
. P2_DOTS_NASEQUENCE__SEQUENCE |  
. P2_DOTS_NASEQUENCE__A_COUNT |  
. P2_DOTS_NASEQUENCE__C_COUNT |  
. P2_DOTS_NASEQUENCE__G_COUNT |  
. P2_DOTS_NASEQUENCE__T_COUNT |  
. P2_DOTS_NASEQUENCE__OTHER_COUNT |  
. F1_DOTS_SEQUENCETYPE |  
. P2_DOTS_NASEQUENCE__SEQUENCE_TYPE_ID |  
. F2_SRES_TAXONNAME |  
. P2_DOTS_NASEQUENCE__TAXON_ID |  
. N1_DOTS_NASEQUENCEKEYWORD |  
. N1_F3_DOTS_KEYWORD;
```

P2_DOTS_NASEQUENCE__DESCRIPTION: description TEXT _description

```
{  
. GUS::Common::Plugin::XMLGUS::process_xml_rule(  
.     undef, undef,  
.     "DoTS::NASEquence::description",  
.     $2->getNodeValue,  
.     $1->getAttribute("pkobj"),  
.     $1->getAttribute("fkobj"),  
.     $1->getAttribute("key"),  
.     $1->getAttribute("depth")  
. );  
};
```

Presently Available Grammars at <http://flora.uchicago.edu/grammars>.

- nasequence.y
inserts rows into DoTS.NASequence with an option to insert a row into DoTS.NASequenceKeyword.
- externalnasequence.y
inserts rows into DoTS.ExternalNASequence.
- blast.y
inserts rows into DoTS.Similarity and child DoTS.SimilaritySpan.
- gtg_genefeature_nalocation_geneinstance.y
inserts rows into DoTS.Genefeature and children DoTS.NALocation, DoTS.GeneInstance.
- gtg_just_gene.y
inserts rows into DoTS.Gene.
- gtg_nafeature_nalocation.y
inserts rows into DoTS.NAfeature and children DoTS.NALocation, DoTS.NAFeatureComment.

Specialized/Site-specific Processing of Data.

```
P2_DOTS_NASEQUENCE_DESCRIPTION: description TEXT _description
{
.   GUS::Common::Plugin::XMLGUS::process_xml_rule(
.       undef, Specialized,
.       "DoTS::NASequence::description",
.       $2->getNodeValue,
.       $1->getAttribute("pkobj"),
.       $1->getAttribute("fkobj"),
.       $1->getAttribute("key"),
.       $1->getAttribute("depth")
.   );
. };
```

In the PERL module Specialized.pm:

```
sub DoTS_NASequence_02
{
.   my $object = $_[GUS::Common::Plugin::XMLGUS::getObjectConstant()];
.   # Process the string
. }
```

Writing Your Own Grammars

Easy to learn, soon becomes routine, yet time-intensive.

Can use pre-existing grammars as templates.

Terry Clark's present research includes automating grammar generation from the user defined XML and/or definitions from the GUS relational schema.

XMLGUS Application Experience at the University of Chicago

- GenBank Formatted Arabidopsis Chromosomes with Annotations
- Centromere/BAC Annotation Project
Shotgun Reads from local sequencing facility and associated BLAST output, contigs and annotation.
- Genome Skimming Project
7,000,000 Shotgun Reads and associated BLAST output, contigs, and annotation.