

Conference call to resolve the latest proposed changes to natdiv module.

## Meeting notes

- Thursday, May 26, 6pm BST / 1pm EST / 10am PST Chado\_Natural\_Diversity\_Module/natdiv\_call\_notes
- Wednesday, June 1, 5pm BST / 12pm EST / 9am PST  
Chado\_Natural\_Diversity\_Module/natdiv\_props\_call\_notes

## Participants

- Seth
- Bob Made some notes here
- Scott
- Rob
- Naama
- Lukas
- Sook
- Yuri
- Maren
- Lacey
- Hilmar

## Agenda

1. Triage proposed changes into the following categories:
  - ◆ implement before paper publishing
  - ◆ implement after paper publishing
  - ◆ do not implement
2. Bio::Chado::Schema update
  - ◆ can someone do one after the changes have been made? Maccallr 14:37, 26 May 2011 (UTC)

## Proposed changes

### Prop table in genotype module

- change: addition of (vanilla) prop table to genotype module [cvterm\_id, value, rank]
  - ◆ proposer: Seth Redmond / Vectorbase
  - ◆ reason: enables us to store ontology terms for current genotypes, e.g. presence/absence of specific inversions - impossible under current schema
  - ◆ Did I understand correctly that for a genotypeprop table that cvterm\_id would allow NULL? Scott 17:17, 26 May 2011 (UTC)

## Hackathon changes

- phenotype
- phenotypeprop (Can somebody add the reason for adding this table?)
- SVN: gmodnatdiv branch:
  - ◆ GMOD Evo Hackathon changes to phenotype module 1) Removed UNIQUE and NOT NULL constraints on phenotype.uniquename. 2) Added phenotype.units\_id. 3) Added phenotypeprop table. 4) Deprecated phenotype\_cvterm table.
  - ◆ Diff:
    - <http://gmod.svn.sourceforge.net/viewvc/gmod/schema/branches/gmodnatdiv/modules/phenotype/phenotypeprop>

## Yuri's proposals

- Wouldn't it be preferable to give at least workable solutions to the two significant flaws of the phenotype module before publishing the paper? From the last call, these are:
  - ◆ Phenotype description and value are 1:1 and in the same table. Solution: Use nd\_experiment\_phenotypeprop to store values.
    - ◇ It is not necessarily 1:1. Phenotype description is stored in cvterm table. The phenotype table stores the value (value or cvalue\_id) and has foreign keys (observable\_id, attr\_id). We could choose to use either one foreign key to connect to one phenotype descriptor or use both. So current schema can store 1 to M, phenotype descriptor to value. (Sook)
      - I think the phenotype table should store the description. Eg, phenotype.observable\_id='stem diameter'. -- Yuri
  - ◆ No (straightforward) way available to do post-composition of phenotype descriptions. Solution: Use phenotypeprop with cvalue\_id
    - ◇ I think post composition can be done in cvterm\_relationship table. All of those terms 'stem diameter at harvest in mm', 'Stem\_diameter', 'at harvest', and 'mm' can be stored in cvterm table. The cvterm\_relationship table then can store the relationships between 'stem diameter at harvest' and 'stem diameter' with the term 'part of' 'belongs to' 'unit' or whatever relationship term appropriate.(Sook)
      - You would need a way to link from the phenotype table to the cvterm\_relationship table. In your example, 'stem diameter' would be stored in phenotype table. The modifier 'at harvest' can be stored in phenotypeprop: type\_id='at', cvalue\_id='harvest'. phenotypeprop also gives the flexibility to store literal values, such as type\_id='at', value='6/1/2009'. I think the unit should be stored near where the value is stored, in this case nd\_experiment\_phenotypeprop (see below). --Yuri
  - ◆ These solutions are easy to implement and can always be refined at some later date.
- Add environmentprop. This is useful when creating phenstatements.
  - ◆ Example phenstatement: ?The mean of the phenotype root length in genotype TN7.4 given an environment of NaCl treatment of 100 millimolar is 10.5 mm?
  - ◆ environment: uniquename='100 NaCl'
  - ◆ environmentprop: type\_id='NaCl treatment', value=100, cvalue\_id='mM'
    - ◇ How about '100 mM NaCl' as environment.uniquename? It can be linked to cvterm via environment\_cvterm table.'100 mM NaCl', 'NaCl treatment' and 'mM' can all be separately stored in cvterm table and associated via cvterm\_relationship table (see above). I think cvalue\_id is to store qualitative values that can be stored in cvterm table, not for the units(Sook).
      - A unit can be a cvterm. I'm using the units in the Unit Ontology. --Yuri

- ◊ or just store '100 mM' in the value field. Or the cvterm should have the unit as part of its definition 'NaCl in mM' --[NaamaMenda](#) 13:58, 1 June 2011 (UTC)
  - I prefer to separate the term, value and unit. Especially if the term and unit are already in cvterm. You are suggesting I should precompose all my terms and I don't think this is necessary (or realistic) for units. --[Yuri](#)
- Add phenstatementprop. This is useful when creating phenstatements.
  - ◆ phenstatement: type\_id = 'summary statistic', phenotype\_id='flower number', genotype\_id='TN7.4', pub\_id='experimental result'
  - ◆ phenstatementprop: type\_id='mean', value = 10.5, cvalue\_id='mm'
    - ◊ this is the actual phenotype value, and should be in the phenotype table , or whatever other table we choose for values or post-composed terms.
      - This is **not** a phenotype value. This is a summary statistic across experiments. --[Yuri](#)
- Add nd\_experiment\_protocolprop. I use this to store protocol values specific to an nd\_experiment.
  - ◆ Eg: nd\_protocol.type\_id='NaCl treatment', nd\_experiment\_protocolprop:{type\_id='treatment amount', value=100, cvalue\_id='mM'}
  - ◆ +1, could definitely use this for same reasons (e.g, same insecticide resistance assay protocol, but with different insecticides, exposure times, and/or concentrations; currently using nd\_experimentprop) [Maccallr](#) 13:24, 30 May 2011 (UTC)
    - ◊ Can we make multiple protocols, such as NaCl 100mM, NaCl 10mM, etc, (or insecticide resistance assay 1, 2, etc) and link to nd\_experiment table? If we want to group similar protocols, we could use protocolprop (type\_id = protocol\_type, value = insecticide resistance assay protocol). The details can be stored in protocolprop (type\_id=exposure time, value=1 hr: type\_id=concentration, value= 10 mM), etc). The insecticide can also be stored in reagent table(Sook).
    - ◊ Aren't these different protocols if you are using different amounts? I don't think the amounts are properties of the protocol. --[NaamaMenda](#) 13:58, 1 June 2011 (UTC)
      - I guess you could say the value is a property of both the protocol and the experiment. --[Yuri](#)
    - ◊ I also think prop tables for linking tables are not consistent with the rest of chado tables and make chado schema too complicated (Sook).
      - I originally got this idea from nd\_experiment\_stockprop, so there's definitely precedent for this. --[Yuri](#)
- Add nd\_experiment\_phenotypeprop. I use this to store phenotype observations specific to an nd\_experiment.
  - ◆ Eg: phenotype.observable\_id='root length', nd\_experiment\_phenotypeprop:{type\_id='observation', value=10.5, cvalue\_id='mm'}
  - ◊ 10.5 can be stored in phenotype.value and the unit can be associated with the cvterm itself in the cvterm table (Sook).
- Add cvalue\_id to NatDiv property tables and related property tables like projectprop. This allows for postcomposition of cvterms like units to the property type\_id.
  - ◆ Eg: type\_id='my experimental bucket color', cvalue\_id='purple'
    - ◊ phenotype\_cvterm with a type\_id column should do. Should we add this to the sql file in a new svn branch ? --[NaamaMenda](#) 13:58, 1 June 2011 (UTC)
      - This proposal was about more than just phenotypes. In several property tables I'm storing values that have units. Using a cvalue\_id would also cut down on the amount of literal strings I'm storing in value fields. -- [Yuri](#)
  - ◆ Clarification: I didn't propose this originally but Naama brought up the concern that the property tables weren't consistent if some have cvalue\_id and others don't.

- ◇ The issue here is whether the prop tables are the right place for this. I think we can add non-uniform columns to prop tables, assuming it is necessary, and Chado has no better way to answer. I think broad usage of prop tables should be limited. Look at props as a place to add some unstructured metadata (dates, names of places, nicknames, maybe synonyms, or anything else you do not wish to structure as a cvterm). If the data does need structure, it is better to store it in a designated well defined table. This is also better for querying the database. With cvalue\_id you need to ask does my prop has a cvalue? what is the meaning of my cvalue? etc. --[NaamaMenda](#) 13:58, 1 June 2011 (UTC)
  - So, it sounds like what you are saying is that either data must be precomposed in an ontology or it should be a literal value. And what I'm saying is why not give the flexibility to post-compose cvterms? I think it's clear in the example I gave the meaning of 'purple' when the property is 'bucket color'. -- [Yuri](#)
- ◆ Chado has some way to post-compose cvterms which [Maccallr](#) 11:56, 17 May 2011 (UTC) doesn't understand.
  - ◇ It looks rather complex. --[Yuri](#)

(Sook) I think that the solution is to store the phenotypic value in the phenotype table and store the cvterm\_id of the post-composed phenotypic descriptor in the phenotype table. The further-up cvterms can be associated via cvterm\_relationship table. We only use 'attr\_id' to store the final post-composed phenotypic descriptor. It might be better to have descriptor\_id in the phenotype table so that users who use both 'attr\_id' and 'observable\_id' can keep their practice.

- It was decided in the meeting to discuss these major changes to the phenotype module in another discussion topic. It was also proposed to see if phenotype\_cvterm with a type\_id field would be adequate. It is clear the phenotype table does not address well post-composing terms, yet we'd like to avoid making the same mistake again of adding multiple columns, which renders the schema not normalized. Also prop tables are not an ideal place for phenotype values (see more in the [Chado Natural Diversity Module/natdiv\\_call\\_notes](#) ). We should keep this discussion related directly to the ND schema, and hammer out the phenotype module as a second stage. --[NaamaMenda](#) 01:14, 1 June 2011 (UTC)
  - ◆ I started a discussion for this page to address the phenotype module proposed changes. (click on the 'discussion' tab at the top of the page) --[NaamaMenda](#) 14:08, 1 June 2011 (UTC)

## Bob's proposals

Just looking at the NatDiv prop tables, saw some inconsistencies:

- nd\_geolocationprop.value is varchar(250) while others in NatDiv are 255. Rest of chado is type 'text'. Propose change to text.

this means we need to change the value type in all nd prop tables to text (Naama)

- nd\_experimentprop.value is NOT NULL while all others (in NatDiv) allow NULL (rest of chado is mixed). Propose all allow NULL.

This was already fixed. I committed the SQL a couple of weeks ago (Naama)

- ◆ I just haven't rolled it into the default\_schema.sql yet [Scott](#) 17:10, 26 May 2011 (UTC)