The Virtual Leaf

Friday 11 June 2010

Issues:

1. Every class that has an association or composition eventually will need its own custom view page template, e.g. sed_view. For the moment I have set the default view for these classes to dev_view (a copy of base_view).
2. An association in UML is cast as a ReferenceField in the resulting class schema. Open the association and add the widget:label tag to the class that is being associated.
3. The default description text for a class should be a informative description of the class, not the instance, derived from the SED-ML manual.
4. Apply 'searchable' tags to the appropriate class attributes.
5. I choose 'isTidyHtmlWithCleanup' as the MathML validator. Will this work?
6. What should be added to our implementation of SED-ML to make it more than merely a mirror image of the SED-ML XML format?

    a. The ability to attach files at various points, e.g. parameter files.
    b. Additional output types, e.g. animations.
    c. Custom page templates
    d. Custom search form
    e. Custom catalog indexes
    f. Custom portlet(s)
    g. Custom viewlet(s)

Friday 24 September 2010

To Do:

1. Publish our mercurial repositories (Chris/Martin). This may mesh with deploying Plone.
2. Create a vleaf mailing list (Henk).

Found an article on SBRML - a markup language for associating systems biology data with models. At a glance the schema seems bigger than SED-ML's, but SBRML appears to be SBML specific.

Monday 27 September 2010

Spoke with Chris about publishing the Mercurial repositories and deploying Plone. Both can be accomplished in their own project space on a server in the DMZ. Chris said he would supply me with login credentials for both accounts. I will install Plone and Chris (or Maarten) will help me configure Apache/nginx and Varnish. Chris will install the required Mercurial packages; after which I can clone the repositories to their new home.

Tuesday 5 October 20010

tutorial0.pro

Added simplugin.h to list of header files. NB: this is the only tutorial in which simplugin.h appears.

auxingrowthplugin.h

Declared, and defined, a virtual function named `DefaultLeafML()` which merely returns a QString naming a LeafML filename sans path.

VirtualLeaf.cpp

Moved `Cell::SetMagnification()` and `Cell::SetOffset()` from `main()` to `MainBase::Init()`.

canvas.h

Declare `exportCellData()`.

canvas.cpp

Add an 'Export cell areas' to the file dropdown menu which invokes - surprise - `Main::exportCellData()`:

```
void Main::exportCellData(void) {
  QFile file("areas.csv");
  if ( file.open( IO_WriteOnly ) ) {
    QTextStream stream( &file );
    mesh.CSVExportCellData(stream);
    mesh.CSVExportMeshData(stream);
    file.close();
  }
}
```

mesh.h

Include `<QTextStream>`

Set the boundary_poloygon pointer to zero in the class constructor, and delete it, if it exists, in the class destructor.

Declare `Compactness()`, `CSVExportCellData()` and `CSVExportMeshData()`:

```
double Compactness(double *res_compactness=0, double *res_area=0, double *res_cell_area=0);
void CSVExportCellData(QTextStream &csv_stream) const;
void CSVExportMeshData(QTextStream &csv_stream);
```

mesh.cpp

In `mesh::clear()`, delete the boundary_polygon only if the pointer hasn't been assigned:

```
if (boundary_polygon) {
   delete boundary_polygon;
   boundary_polygon=0;
 }
```

Ditto for `mesh::clean()`.

Define the code for `Compactness()`, `CSVExportCellData()` and `CSVExportMeshData()`.

modelcatalogue.cpp

In `InstallModel()`, find and load the default LeafML file.

simplugin.h

Declare `DefaultLeafML()`:

```
// Default LeafML-file to be read after model startup
virtual QString DefaultLeafML(void);
```

simplugin.cpp

Define `DefaultLeafML()`. Returns an empty `QString`:

```
QString SimPluginInterface::DefaultLeafML(void) { return QString(); }
```

xmlwrite.cpp

In `Mesh::XMLReadCells()` - Delete the boundary_ploygon only if its pointer has been assigned.

Wednesday 6 October 2010

Wrote Simon van Mourik about his missing libiconv-2.dll.

Thursday 7 October 2010

Added a Q3FileDialog in `canvas.cpp:Main::exportCellData(void)` to choose where to write the exported cell data.

Friday 8 October 2010

Added new parameters to control perodic cell exprt data.

Added code to relize perodic cell export.

Monday 11 October 2010

Mercurial hooks

Tried several alternatives for Mercurial's `pretxnchangegroup.forbid_2heads` hook.

1. http://bitbucket.org/dgc/headcount: **Headcount** `ui` complains of a missing data member
2. http://stackoverflow.com/questions/1705921/useful-mercurial-hooks: **forbid2_heads.py** doesn't load
3. http://davidherron.com/node/961: **forbid2_heads.sh** works as advertised

Mercurial Repositories

Tried to install Rhode Code from: http://packages.python.org/RhodeCode.

1. http://bitbucket.org/marcinkuzminski/rhodecode
2. http://packages.python.org/RhodeCode, http://pypi.python.org/pypi/RhodeCode/1.0.0rc3

3. http://hg.python-works.com (demo,demo)
4. http://bitbucket.org/bfrog/cutehg
5. http://pypi.python.org/pypi/SIP, http://www.riverbankcomputing.com/hg/sip
6. http://pypi.python.org/pypi/PyQt
7. http://ask.github.com/celery

All goes tolerably until you tick in the url; at which point RhodeCode complains that:

```
Exception happened during processing of request from ('127.0.0.1', 35803)
Traceback (most recent call last):
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/Paste-1.7.5.1-py2.6.egg/paste/httpserver.py", line 1068, in process_request_in_thread
   self.finish_request(request, client_address)
 File "/ufs/guravage/opt/Python-2.6.2/lib/python2.6/SocketServer.py", line 320, in finish_request
   self.RequestHandlerClass(request, client_address, self)
 File "/ufs/guravage/opt/Python-2.6.2/lib/python2.6/SocketServer.py", line 615, in __init__
   self.handle()
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/Paste-1.7.5.1-py2.6.egg/paste/httpserver.py", line 442, in handle
   BaseHTTPRequestHandler.handle(self)
 File "/ufs/guravage/opt/Python-2.6.2/lib/python2.6/BaseHTTPServer.py", line 329, in handle
   self.handle_one_request()
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/Paste-1.7.5.1-py2.6.egg/paste/httpserver.py", line 437, in handle_one_request
   self.wsgi_execute()
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/Paste-1.7.5.1-py2.6.egg/paste/httpserver.py", line 287, in wsgi_execute
   self.wsgi_start_response)
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/Paste-1.7.5.1-py2.6.egg/paste/cascade.py", line 130, in __call__
   return self.apps[-1](environ, start_response)
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/Paste-1.7.5.1-py2.6.egg/paste/registry.py", line 379, in __call__
   app_iter = self.application(environ, start_response)
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/RhodeCode-1.0.0rc1-py2.6.egg/rhodecode/lib/middleware/https_fixup.py", line 33, in __call__
   return self.application(environ, start_response)
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/Pylons-1.0-py2.6.egg/pylons/middleware.py", line 163, in __call__
   self.app, new_environ, catch_exc_info=True)
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/Pylons-1.0-py2.6.egg/pylons/util.py", line 48, in call_wsgi_application
   app_iter = application(environ, start_response)
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/WebError-0.10.2-py2.6.egg/weberror/errormiddleware.py", line 156, in __call__
   return self.application(environ, start_response)
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/RhodeCode-1.0.0rc1-py2.6.egg/rhodecode/lib/middleware/simplehg.py", line 55, in __call__
   return self.application(environ, start_response)
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/Beaker-1.5.4-py2.6.egg/beaker/middleware.py", line 152, in __call__
   return self.wrap_app(environ, session_start_response)
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/Routes-1.12.3-py2.6.egg/routes/middleware.py", line 131, in __call__
   response = self.app(environ, start_response)
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/Pylons-1.0-py2.6.egg/pylons/wsgiapp.py", line 98, in __call__
   self.setup_app_env(environ, start_response)
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/Pylons-1.0-py2.6.egg/pylons/wsgiapp.py", line 203, in setup_app_env
   pylons_obj.translator = _get_translator(lang, pylons_config=self.config)
 File "/ufs/guravage/.archive/mas/mr/rhodecode/lib/python2.6/site-packages/Pylons-1.0-py2.6.egg/pylons/i18n/translation.py", line 165, in _get_translator
   raise LanguageError('IOError: %s' % ioe)
LanguageError: IOError: [Errno 2] No translation file found for domain: 'rhodecode'
```

RE-INSTALL

1. easy_install mercurial
2. mkdir src
3. cd src
4. hg clone http://www.riverbankcomputing.com/hg/sip
5. cd sip
6. python build.py prepare
7. python configure.py
8. make
9. make install
10. cd ..
11. wget http://www.riverbankcomputing.com/static/Downloads/PyQt4/PyQt-x11-gpl-4.7.7.tar.gz
12. tar xzf PyQt-x11-gpl-4.7.7.tar.gz
13. cd PyQt-x11-gpl-4.7.7
14. python configure.py
15. make
16. make install
17. cd ..
18. hg clone https://litsol@bitbucket.org/bfrog/cutehg
19. cd cutehg

20. python setup.py build
21. python setup.py install
22. cd ..
23. easy_install rhodecode, http://pypi.python.org/pypi/RhodeCode/1.0.0rc2

This procedure fails as before.

However, upon inspection I saw that the i18n directory was missing from my build but present in the rhodecode clone from bitbucket. While I think that the `easy_install rhodecode` invocation is necessary to install the various subsidiary packages, moving the cloned rhodecode into the site-packages directory seems to have made most functionality work. e.g. graphical logs and diff highlighting.

Tuesday 12 October 2010

RhodeCode

I think that if Chris can install RhodeCode, I can serve individual instance from my home directory.

SED-ML L1V1RC1

Walking through the new SED-ML schema I see that they've dropped the generic simulation class and added AddXML and RemoveXML model change classes. Adding several more SED-ML examples for the symposium might still be justified - though they become obsolete the moment I introduce the new schema changes.

Wednesday 13 October 2010

Tweaked `.bashrc` and `virtualenvwrapper.sh` to facilitate virtualenv and virtualenvwrapper - many unbound variables!

Pushed recent changes to repository from where Roeland can retrieve and test them.

OK. RhodeCode RC4 works out of the box, excepting cutehg and PyQt4.

Friday 15 October 2010

Rhodecode

Installing RhodeCode locally on the nhypnos is a real pain. In addition to the previous list of dependencies, for the caterpie, add these for the nhypnos:

1. Python
2. Bison (for sip)
3. Flex (for sip)
4. m4 (for Bison)
5. Qt (for PyQt)
6. GLIBCXX_3.4.9 (for Qt) - I draw the line at (re)installing GCC!

OK. Forget sip bison, flex, m4, Qt and the rest. Let's install just the minimum.

First, install sqlite; we'll see in a minute that Python requires it. The sources are available at: http://www.sqlite.org/download.html.

Second, install Python. My initial attempt complained it was missing the following bits and pieces:

```
Python build finished, but the necessary bits to build these modules were not found:
_bsddb             _sqlite3           _ssl
_tkinter           bsddb185           dl
imageop            readline           sunaudiodev
To find the necessary bits, look in setup.py in detect_modules() for the module's name.
```

So install sqlite first. And, since Python's configure script doesn't have an option for it, I had to tweak `setup.py` to enumerate the path where it would find sqlite;

Third, install RhodeCode. I first installed it using easy_install, but while the result worked, the changelog view did not display the branch/merge graphics. I then cloned the code from its bitbucket repository:

```
clone https://litsol@bitbucket.org/marcinkuzminski/rhodecode
```

Replacing the egg installed by easy_install by this code does display the branch/merge graphics.

Fourth, setup and serve rhodecode. Following the instructions posted online at: http://packages.python.org/RhodeCode/setup.html#setup works as advertised:

```
paster make-config RhodeCode production.ini
paster setup-app production.ini
paster serve production.ini
```

All that's left is an apache rewrite rule directing virtualleaf.project.cwi.nl/repository to port 5000, or whaterver we change it to, on the nhypnos.

The Virtual Leaf

Added `incrementIterations()`, `getIterations()` and `int iterations` to mesh.h. `incrementIterations()` is called in TIMESTEP in VirtualLeaf.cpp. The actual counter, `iterations` is initalized to zero in the Mesh class constructor.

The iterations are inquired in two places: for the frame count in `MainBase::Plot` in `VirtualLeaf.cpp` and for the export cell data interval in ``Main::TimeStepWrap` in `canvas.cpp`.

The questions remain: Should we use this count, and if yes, where should it be incremented? The previous count was incremented in `Main::TimeStepWrap`. Or should we simply use `(int)mesh.getTime()`? Since this is not an ordinal count, the results look strange.