OpenCMISS Repository Plan Proposal

Timeframe

Proposed: 20 August 2015
Approve by: ~~3 September 2015~~ 10 September 2015

Purpose

Agree on an initial set of github organisations and repositories so development on OpenCMISS can proceed with confidence about where content can be retrieved from and submitted to.

This plan is intended for use by developers; users will primarily find links to source code repositories via the OpenCMISS website.

General Guidelines

Ideally content is in repositories under the OpenCMISS github organisation, however to avoid cluttering it with large numbers of repositories, additional github organisations are used. The only additional organisations proposed are OpenCMISS-Dependencies and OpenCMISS-Examples, as described below.

Use a separate repository for a body of code or content that needs to be checked out together, and maintained consistently including with consistent dependencies on other repositories/libraries.

Do not commit into any repository derived works such as compiled binaries, documents converted from raw sources etc.. The OpenCMISS website, builders and services will be responsible for delivering derived content. If possible, avoid adding large binary data, documents, presentations - prefer links to outside hosts. For documentation prefer vector images (SVG) or otherwise choose image formats with appropriate quality/compression for the type of content (Usually: PNG for illustrations with large areas of plain colour, otherwise JPG).

GitHub Organisations

This section details the main OpenCMISS-related GitHub organizations.

OpenCMISS

Repositories:

- iron
- zinc
- documentation
- manage

The documentation repository is to contain all OpenCMISS documentation *except* library-specific API documentation (i.e. Doxygen) which is put with the respective library, and example-specific documentation which is put with the respective example.

OpenCMISS-Dependencies

All third-party libraries that require modification for building OpenCMISS libraries against. Each third-party library is in a separate repository.

Dependencies must adhere to license requirements which require that:

- the original source without modifications, plus the changes made are both obtainable
- inclusion of original license
- proper attribution of original authors
- in some cases (which need to be clearly listed) post-commit hooks are required to notify original author of changes.

OpenCMISS-Examples

Existing OpenCMISS examples will be migrated to individual repositories under this organisation and descriptions (annotations) of each example will be indexed in PMR. It is expected that these repositories will not change beyond simple fixes to update examples when APIs or library dependencies are changed. Substantial changes and any new examples will be stored wherever the example developer or group prefers, with the description indexed in PMR. Developers may choose to store new examples under this organisation if they have permission to do so.

Reviewer Notes

Reviewers should add a line in a Reviewer Notes section indicating they have completed reviewing the proposal, and the date at which it was completed so it is clear which revision was reviewed, and indicate whether they approve entirely (provided their preferred or acceptable options are chosen) or have requested changes via comments.

Daniel Wirtz: Approves on September 10th

Andreas Hessenthaler: Approves the proposal as at Sept 9.

Richard Christie. Approves the proposal as at Sept 9.

David Nickerson: Approves the proposal as at Sept 9.

Chris Bradley: Approves the proposal as at Sept 10.

Hugh Sorby: Approves the proposal as at Sept 10.

Attachments

Archive of previous content of the proposal, and meeting minutes, both in date order.

Removed 9/9/15 as this proposal should stick to listing organisations, repos, folders, not specifying processes.

Lightweight Change Process

To ease making future changes:

1. Discuss (in person, by e-mail, in the tracker) requirements and ideas with other developers and/or OpenCMISS Senior Development Group members, to get support or better ideas.
2. E-mail the proposed change to all members of the SDG.
3. If all agree to proposed change, then it is approved.

Folder structure removed from proposal

Iron and zinc libraries have these subfolders:

- /src All source code for the library
- /bindings All sources and scripts for building non-native language bindings
- /tests Unit tests (if implemented)

The documentation repository will initially have this subfolder:

- /development Contains OpenCMISS processes including the Standard Agreement Process (and as part of this proposal, that process will be updated to reflect this new location), this Repository Plan, future processes, coding standards etc.

Daniel Wirtz:

1. I don't think the folder-details of the iron/zinc repositories (src,tests etc) should be placed here, as anyone who opens this document expects details on git-related stuff only.
2. Having a /process folder within the documentation repo is a good idea; i'd consider naming it "development" or "standards", as the contents are less a user's but a developer's guide?! but i wont make a fuss if ya'll want "process".
3. Should we include a suggested git-workflow for developing OpenCMISS components (forking, pull requests etc) here or have a separate proposal for that? Opened a new proposal for that.
4. I've added some details in the Examples section mainly including a "collector" repo that maintains references to example revisions that are known to work with the current release.

Andre:

1. Demonstrating how to use PMR to reference external data (not trying to set the actual annotations to use, just demonstrating the concept).
    a. go to http://staging.physiomeproject.org/pmr2_virtuoso_search
    b. enter the following query to search for all OpenCMISS examples:

```
SELECT ?x
WHERE { ?x  <http://opencmiss.org/d/terms/1.0/example>  ?y}
```

2. using services, you can then navigate through to other information, e.g.,

```
SELECT ?y ?z
WHERE {
<https://github.com/nickerso/really-really-really-cool-opencmiss-exam
ple>  ?y ?z}
```

or

```
SELECT ?y ?z
WHERE {
<https://github.com/nickerso/not-so-cool-opencmiss-example/that-will-
never-be-accepted>  ?y ?z}
```

Auckland meeting 2015-09-04

At the Auckland OpenCMISS meeting we (Richard, Chris, Andre, David Ladd, Noel) discussed this proposal. Mostly discussed the examples repositories/organisation. All agreed that indexing example descriptions in PMR is the way to go and that examples themselves can be stored anywhere on the internet. Richard and Chris proposed that there should be a “special place” where examples are managed by the “OpenCMISS” team which would contain examples guaranteed to be valid and kept up to date. Tests would be drawn from this special set of examples. Richard and Chris believe this should be the OpenCMISS-examples organisation, Andre believes this should be an ABI software lab thing, if in fact something special is required (which he doesn’t agree with).

There was also some discussion regarding the addition of a buildbot repository to this proposal - generally accepted, but we need to be aware of any security implication if this repository needs to contain ssh keys or passwords or ip address etc. More information is needed in terms of how this relates to the existing gitlab repos.