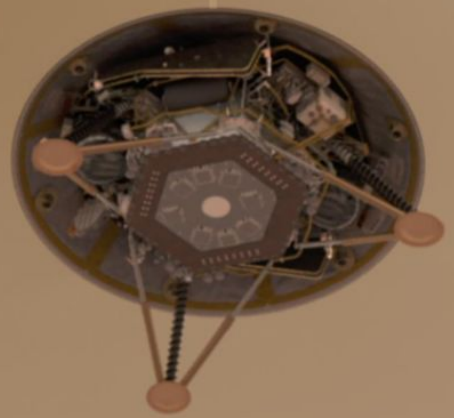
Jet Propulsion Laboratory
California Institute of Technology

Improvements in Concurrent Software Development and Version Control

Arash Hosseini Jafari
Software Engineering Intern
M.S. in Data Engineering Student
UC Riverside

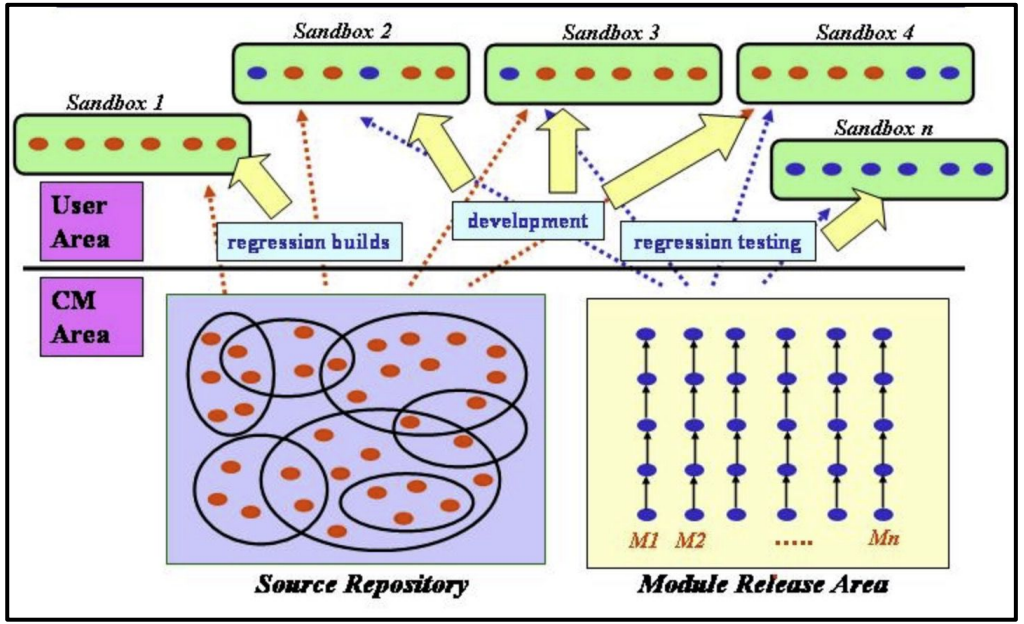
Mentors:
Dr. Abhinandan Jain
Dr. Jonathan Cameron



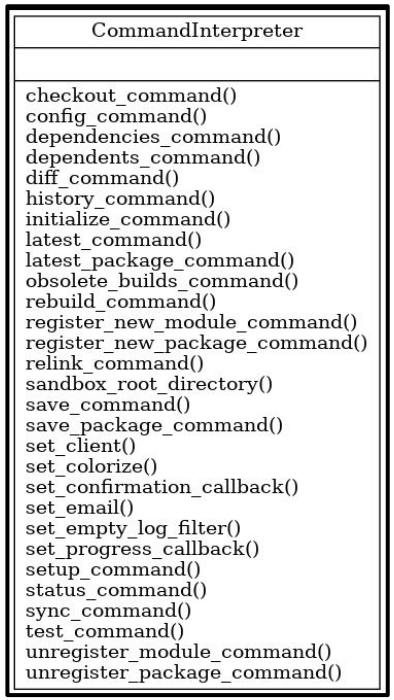
DARTS LAB

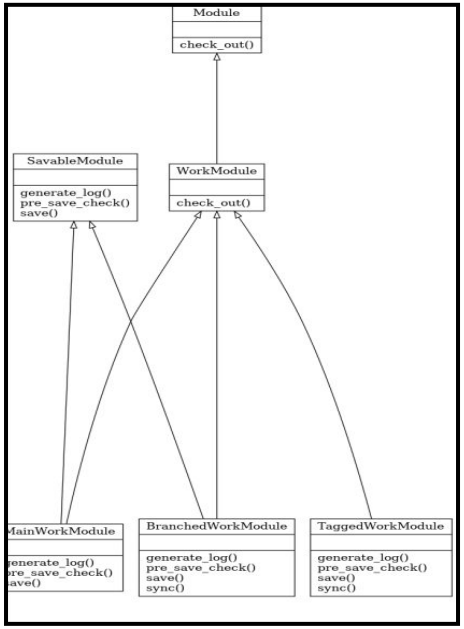
Why Pyam

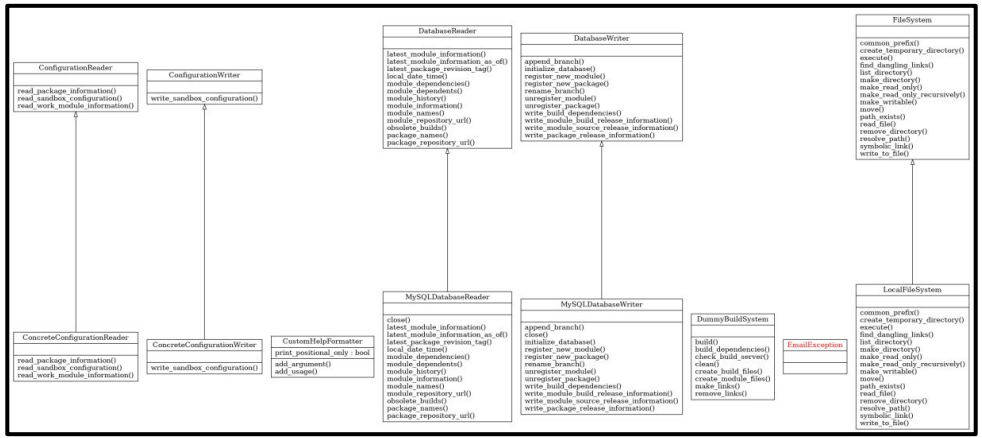
- Reusable Software development infrastructure
- Module branches
  - Allow concurrent development
  - Multiple adaptations
- Packages
  - Allow deployment of different configurations of software
- Release early and often without stepping on each other's toes.



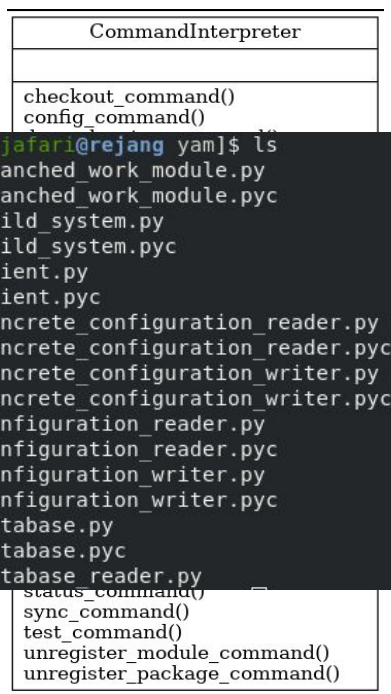
Why Pyam

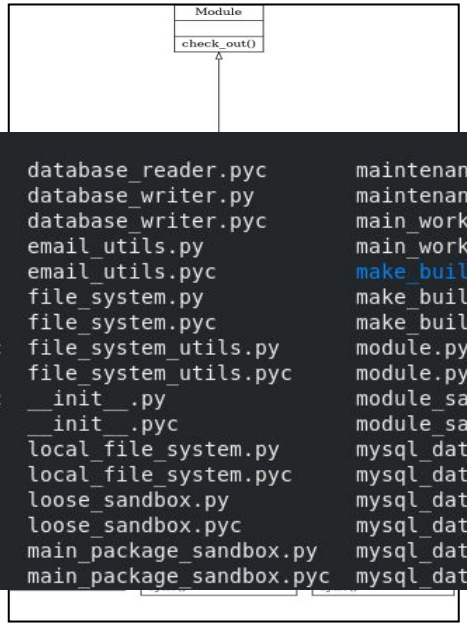

CommandInterpreter
checkout_command()
config_command()
dependencies_command()
dependents_command()
diff_command()
history_command()
initialize_command()
latest_command()
latest_package_command()
obsolete_builds_command()
rebuild_command()
register_new_module_command()
register_new_package_command()
relink_command()
sandbox_root_directory()
save_command()
save_package_command()
set_client()
set_colorize()
set_confirmation_callback()
set_email()
set_empty_log_filter()
set_progress_callback()
setup_command()
status_command()
sync_command()
test_command()
unregister_module_command()
unregister_package_command()

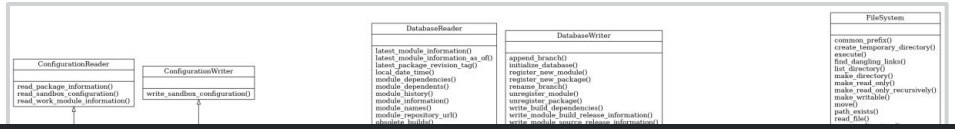

Module
check_out()
SavableModule
generate_log()
pre_save_check()
save()
WorkModule
check_out()
MainWorkModule
BranchedWorkModule
generate_log()
pre_save_check()
save()
sync()
TaggedWorkModule
generate_log()
pre_save_check()
save()
sync()


ConfigurationReader
read_package_information()
read_sandbox_configuration()
read_work_module_information()
ConfigurationWriter
write_sandbox_configuration()
DatabaseReader
latest_module_information()
latest_module_information_as_of()
latest_package_revision_tag()
local_date_time()
module_dependencies()
module_dependents()
module_history()
module_information()
module_names()
module_repository_url()
obsolete_builds()
package_names()
package_repository_url()
DatabaseWriter
append_branch()
initialize_database()
register_new_module()
register_new_package()
rename_branch()
unregister_module()
unregister_package()
write_build_dependencies()
write_module_build_release_information()
write_module_source_release_information()
write_package_release_information()
FileSystem
common_prefix()
create_temporary_directory()
execute()
find_dangling_links()
list_directory()
make_directory()
make_read_only()
make_read_only_recursively()
make_writable()
move()
path_exists()
read_file()
remove_directory()
resolve_path()
symbolic_link()
write_to_file()
ConcreteConfigurationReader
read_package_information()
read_sandbox_configuration()
read_work_module_information()
ConcreteConfigurationWriter
write_sandbox_configuration()
CustomHelpFormatter
print_positional_only : bool
add_argument()
add_usage()
MySQLDatabaseReader
close()
latest_module_information()
latest_module_information_as_of()
latest_package_revision_tag()
local_date_time()
module_dependencies()
module_dependents()
module_history()
module_information()
module_names()
module_repository_url()
obsolete_builds()
package_names()
package_repository_url()
MySQLDatabaseWriter
append_branch()
close()
initialize_database()
register_new_module()
register_new_package()
rename_branch()
unregister_module()
unregister_package()
write_build_dependencies()
write_module_build_release_information()
write_module_source_release_information()
write_package_release_information()
DummyBuildSystem
build()
build_dependencies()
check_build_server()
clean()
create_build_files()
create_module_files()
make_links()
remove_links()
EmailException
LocalFileSystem
common_prefix()
create_temporary_directory()
execute()
find_dangling_links()
list_directory()
make_directory()
make_read_only()
make_read_only_recursively()
make_writable()
move()
path_exists()
read_file()
remove_directory()
resolve_path()
symbolic_link()
write_to_file()

Why Pyam








```
[ajafari@rejang yam]$ ls
branched_work_module.py                database_reader.pyc          maintenance_saving_utils.py   name_utils.py                svn_revision_control_system.py
branched_work_module.pyc               database_writer.py           maintenance_saving_utils.pyc  name_utils.pyc               svn_revision_control_system.pyc
build_system.py                        database_writer.pyc          main_work_module.py           package_sandbox.py           tagged_package_sandbox.py
build_system.pyc                       email_utils.py               main_work_module.pyc          package_sandbox.pyc          tagged_package_sandbox.pyc
client.py                              email_utils.pyc              make_build_system_data        port_utils.py                tagged_work_module.py
client.pyc                             file_system.py               make_build_system.py          port_utils.pyc               tagged_work_module.pyc
concrete_configuration_reader.py       file_system.pyc              make_build_system.pyc         pysvn_verbose                text_editor_utils.py
concrete_configuration_reader.pyc      file_system_utils.py         module.py                     revision_control_system.py   work_module.py
concrete_configuration_writer.py       file_system_utils.pyc        module.pyc                    revision_control_system.pyc  work_module.pyc
concrete_configuration_writer.pyc      __init__.py                  module_saving_utils.py        revision_tag_utils.py        yam_exception.py
configuration_reader.py                __init__.pyc                 module_saving_utils.pyc       revision_tag_utils.pyc       yam_exception.pyc
configuration_reader.pyc               local_file_system.py         mysql_database_reader.py      sandbox.py                   yam_log.py
configuration_writer.py                local_file_system.pyc        mysql_database_reader.pyc     sandbox.pyc                  yam_log.pyc
configuration_writer.pyc               loose_sandbox.py             mysql_database_utils.py       savable_module.py
database.py                            loose_sandbox.pyc            mysql_database_utils.pyc      savable_module.pyc
database.pyc                           main_package_sandbox.py      mysql_database_writer.py      saving_utils.py
database_reader.py                     main_package_sandbox.pyc     mysql_database_writer.pyc     saving_utils.pyc
```

Developing Test Cases & Validation

- Wrote test cases for a variety of Pyam features
- Subsequently used Cram for constant development and testing of new features
  - Test cases
    - No keep release
    - Pyam save
    - Pyam Build with Email
    - Pyam Checkout
    - YamConfig
    - Maintenance Feature
  - misc tasks
    - Fixing broken test cases & Validation of notebooks

Developing Test Cases

- Cram (Link: https://bitbucket.org/brodie/cram/src/default/)
  - Functional testing framework for command line applications
  - Regression Testing
    - Runs a series of commands and compares the output of each command with the desired output.

```
srun dtest-cram --shell=/bin/bash --indent=4 script5.t
```

OR

```
srun dtest
```

```
Running  test
   test            exit status - True
 STATUS:     RUN: 1/1   CMP: 0/0  SUCCESS


SUMMARY: Ran 1 tests, 1 succeeded, 0 failed (in pyam)
Tests completed in 14.045321 seconds
```

```
Test the "save" command.

    $ cd "$sandbox_directory"
    $ $PYAM --no-keep-release save 'Dshell++'
    WARNING!!! Move to release directory has been disabled.* (re)
```

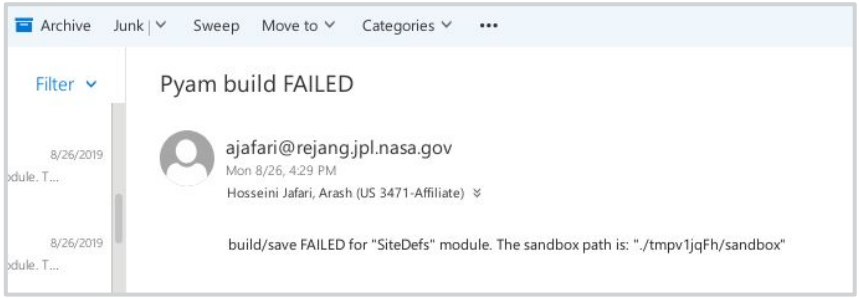
Scrap Feature

- Problem: Checking out specific releases or branches of a given work module did not work if you had checked out the module previously.
  - Requires:
    - Editing YAM.config
    - Moving or deleting module from sandbox
- Solution: Added Scrap Feature:
  - Argument that runs a config_to_link function followed by appending the module directory name in the sandbox with a timestamp

```
     $ $PYAM scrap 'Dshell++'
+    Scrapping work-module and converting work modules to link modules
+    --->  Converting 'Dshell++' to a link module
+    Appending module directory name in the sandbox with timestamp
+      Running 'make rmlinks' subprocess: start
+      Running 'make mklinks' subprocess: start
+      Running 'make mklinks' subprocess: start
```

Email Alerts

- Pyam-build
- Added feature to Pyam-build to send out email alerts when Pyam-build fails to release a module.
  - How it works
    - SMTP*
    - emai_utils.py
    - Pyarmc
    - local host
  - recently added sandbox path




```
if modules_with_errors:
    hostname, port = config_dict['email_server'].split(':')
    port = int(port)
    email_from_address = '{user}@{host}'.format(
            user=getpass.getuser(),
            host=platform.node()    )

    print('\nTemporary sandbox: {sandbox}'.format(sandbox=sandbox_path),
          file=sys.stderr)

    print('\nThese modules were not able to be built/saved:',
          file=sys.stderr)
    for module in modules_with_errors:
        print(module, file=sys.stderr)
        error_message = ('build/save FAILED for "{module}" module. \n \n \n --- Associated Directories
                         ' * The temp sandbox path: "{sbpath}".'
                         '\n * Pyam is running from: "{pyampath}".'
                         '\n * Current working directory: "{crntdir}" '
                         '\n * Other paths: "{s}"'.format(module=module, sbpath=sandbox_path, pyampath
        email_utils.send_email(
        subject='Pyam build FAILED',
        body= error_message,
        from_address= email_from_address,
        to_addresses= [config_dict['project_admin_email'],],
        hostname='127.0.0.1',port=port)
```

Maintenance Feature

- Problem: Need to facilitate the delivery of delta updates and patches to NASA project teams that must remain on older, or mission specific, versions of the lab's simulation tool.
- Added features:
  - Maintenance Checkout:

(pyam checkout --maintenance "project name")

```
$ cd "$sandbox_directory"
$ $PYAM checkout 'Dshell++' --maintenance 'Mars2020'
YOU ARE CHECKING OUT A MAINTENANCE BRANCH
--->  Converting 'Dshell++' to a work module
--->  Checking out source code for 'Dshell++' module
  Running 'make rmlinks' subprocess: start
  Running 'make mklinks' subprocess: start
  Running 'make mklinks' subprocess: start
  Running 'make all' subprocess: start
```

Maintenance Feature

- Problem: Need to facilitate the delivery of delta updates and patches to NASA project teams that must remain on older, or mission specific, versions of the lab's simulation tool.
  - Reconfigured scripts related to pyam save
    - mainly scripts in Yam folder including: Branched_Work_modules, Client, Module_saving_utils, SVN_Version_Control_System
  - Added Maintenance_saving_utils.py

```
    $ $PYAM sync --all
+   --->  Module 'Dshell++' is already synced up
    $ $PYAM save 'Dshell++'
+   --->  Checking validity of working copy
+   --->  Running pre-save hooks
+   --->  Checking that link modules are up to date
+   --->  Checking for dangling links
+   --->  MAINTENANCE BRANCH RECOGNIZED
+   --->  THIS IS THE RELEASE PATH: /tmp/cramtests-lHAjwL/fake_release/Module-Releases/Dshell++
+   --->  Moving module to release area '/tmp/cramtests-lHAjwL/fake_release/Module-Releases/Dshell++/Dshell++-R4-06iM00-Mars2020-Maint00'
```

Maintenance Feature - Round 1

1. Making sure any development branches/releases of the module are “removed” from the sandbox and pyam YAM.config is in the right configuration
   - pyam scrap “Module_Name”
2. Checkout a maintenance branch for the first time based on the latest release of the module.
   - pyam checkout “SHECrover” --maintenance “Abhi” --release R1-00c
     i. Will checkout module in featureBranches/SHECrover-R1-00c-Abhi-Maint
     ii. Will add “BRANCH_SHECrover = SHECrover-R1-00c Abhi-Maint” entry to YAM.config
   - Note - cannot for now checkout a branch off of an older version of a maint branch however older versions of the module files and folders do exist in the release area for efference.
3. Make changes and commit as normal
   - svn status, svn add, svn commit, svn update and other svn commands work as normal.
4. Save as normal with: pyam save “Module_Name”
   - No need for maintenance subcommand. Pyam will recognize a maint branch is being saved and handles the save differently.

Maintenance Feature - Round 2

Now we are done working on the maint branch and want to CO normal Module

1. Must reconfigure sandbox back to pre-maintenance state (i.e: wrk modules to link modules, and remove the directory if it's still there.)
   - pyam scrap "Module_Name"
2. Checkout a regular development branch
   - pyam checkout "Module_Name"
3. Continue as normal ...

Maintenance Feature - Round 3

Return to working on the maintenance branch

1. The revision tag in YAM.config has moved up and we need to manually bring it down and change the branch name to “projectName-Maint”.
    - Todo: Automate this (NOT DONE YET)
        - (Using the config --to-work function and concrete_configuration_writer)
2. Checkout a NEW maintenance branch based on an older release OR the latest module.
    - Two options
        - Add above BRANCH entry with mainteance tag to YAM.config, and run ‘pyam checkout SHECrover’. This will checkout a new maint branch based on the latest development version.
        - Other option is to run “pyam checkout SHECrover --release R1-00c --maintenance Abhi”. This will also work.
3. Make changes and commit as normal
    - svn status, svn add, svn commit, svn update and other svn commands work as normal.
4. Save as normal with: pyam save “Module_Name”

TO DO

1. No SVN repo changes or tags currently. Goal was to apply M02 svn tag to this release
2. It is also not currently adding a ChangeLog entry
3. Leaves YAM.config unchanged, i.e. still leaves the old BRANCH entry and does not convert into a link module
    - (requires manual eddit of YAM.config after a save.)
4. Also get a spurious error about “ YaM error: Some modules ('SHECrover',) could not be saved due to the following errors.”.
    - Even though save went through successfully… Need to see what is happening here.
5. For other misc to do’s that are not related to maintenance branch (see work_progress document)

Thank You!

Contact:
Arash Hosseini Jafari
arashh@uci.edu
arash.hosseini.jafari@jpl.nasa.gov (may change)
818-913-8272

DARTS LAB