ACCORD System Area Leader

Daniel Santos, 1st ACCORD ASW, 12-16 April 2021

RWP Strategy

In the last years RWPs we have included some COMMON actions to increase the number of shared activities. In 2021 we designed some COM actions for system activities. The **main objective to create a new more collaborative and productive working environment** will require:

Design a **shared multiple repository infrastructure and associated working practices**:

- Coordination **started between ECMWF and MF** to have the same projects in both repos This **transparency exercise between ECMWF and MF will facilitate the collaboration** for Global models that **should be beneficial for LAMS**

- The number of **ways that model is used**, the **configurations**, the **code versions and** the increased use of **accessory software** creates **not only a system** in fact it could be considered as an **ECOSYSTEM** that can be easier treated under a **multirepo framework**.

- This **ECOSYSTEM** should **allow enough freedom** for the different usages and configurations, but should be **an efficient and productive way of introduce code and scientific interchange**.

RWP Strategy

Design a **shared multiple repository infrastructure and associated working practices (cont)**:

- The **multirepo** will increase that freedom due to the **system customization** thanks to the **modularity** The modularity will help in the **system evolution and adaptation to new tendencies and technologies like GPUs, Machine Learning and Cloud Computing** and opens the door to the **collaboration with other communities.**

- To extract all these **potential benefits,** we will need to perform **some actions:**

  - Establish a **flexible shared platform for code information exchange.** The platform should be **easily accessible** for any partner. It should enable posting of code contributions, ticketing, assigning tasks, ... Now repos are controlled by **GIT** so these **functionalities, usability and a more user-friendly environment** are offered by **GITHUB, GITLAB or BITBUCKET.**

  - Define **new working practices** is needed **for coordinating code developments that could affect several repositories** and also, be adapted towards **a more continuous code integration process for LAM partners** and taking into account the link with ECMWF.

RWP Strategy

Design a **shared multiple repository infrastructure and associated working practices (cont)**:

- To extract all these **potential benefits,** we will need to perform **some actions (cont):**
  - The **accessibility to the repositories** by the main ACCORD developers and **sharing the maintenance responsibility** of the current three CSC's will **increase the transparency and sense of cooperation.**
  - Establish **regular meetings on Code and System aspects** with the aim of **creating a group of code maintenance experts**. So, establish **working weeks, task teams** … will increase the sense of a **good working and productive environment**. Build this **community culture** will **inspire us and increase the motivation for sharing experiences and knowledge**.
  - Obviously to **homogenize and spread the knowledge** we will define **solutions for training staff on tools**, for training staff on how to run components and assembled parts of the System. When needed, training will be **organized at different levels to facilitate the transition to a more common and efficient working practices.**

RWP Strategy

The **theoretical approach** will require some **technical actions** that can be summarized in **3 ACTIONS**:

- **ACTION 1: Organize the codes**:
  - Create a **prototype multiple repository organization for NWP core codes** (content and structure) and establish **a platform for exchange of technical information** which is well integrated with the multiple GIT repository infrastructure.
  - **Determine a flexible and efficient way to link all of repos** by **bundling tool.** The **ecbundle and bundle.yml** definition infrastructure appears to be quite mature and should be evaluated as overall solution.
  - Explore **repository solutions for supplementary codes and new tools** (testing tools, scripting, etc.).

RWP Strategy

- **ACTION2: test the codes:**
  - Development of the **unit testing based on DAVAÏ**
    - **Ensure Davai portability** creating an interface which allows users to **execute the Davai tool on other platforms** and **implement tests** of components **for different CSCs.**
    - Further develop a **user-friendly tool and interface for visualizing the results** of unit testing (eg. “ciboulaï”)
    - Arrange **training of key integrators to use the Davai tool**, to allow them to locally test **possible code contributions more frequently.**

RWP Strategy

- **ACTION 3: increase system usability and maintenance**:

  We consider a system the **ECOSYSTEM** of **codes, configurations and accessory software that allows**

  - One of the **key part of the ECOSYSTEM** are the **scripting systems**. They allow to **execute, control and validate all the process** that a numerical weather forecast generation requires.

  - **Not all the ACCORD partners** have the **same level of functionality** on their scripting systems To **increase the productivity, the model usability and facilitate the maintenance** the **convergence** to a **single scripting** system is desirable

    - Collect information from Members **to map their current scripting systems, their functionalities and their dependencies on IT** elements

SYSTEM SESSION SUMMARY

System activities in HIRLAM-C (Daniel Santos)

- **Harmonie releases:**
  - 3 years since 43h1.pre-alpha.1 and harmonie-43h2.2
  - Different types of releases: Operational, Technical and Scientific
  - Operational implementation problems
    - SAPP and WMO Bufr standard for the local observation treatment
  - Redesign the meteorological validation and testing
  - CANARI problems and use of pysurfex as replacement
- **Common code generation and maintenance:**
  - 10 branches committed in CY48T1
  - 3 pending branches to be committed on CY48T2 ?? Or pre phase for CY49
  - More validation and easier integration process (Thanks Nikko and Alexadre)
  - Better use of GIT for code interchange and maintenance:
    - HIRLAM Public GITHUB: OpenSource codes
    - HIRLAM Private GITHUB: Harmonie-Arome codes
    - Training and define working practices is needed
- **Code optimization:**
  - SP, DP and DualPrecision
  - 4DVar optimization
  - BSC 2nd phase Harmonie performance analysis

SYSTEM SESSION SUMMARY

GMKPACK, 16 years of a build system (Ryad El Khatib)

- **Genesis:**
  - F77 => F90 with dependencies and ODB software
  - mkpack based on make => gmak Perl wrapper solve duplicated files
  - GMKPACK : a wrapper of gmak for developers
- **Evolution:**
  - 2004: First Stable release;
  - 2006-2010: flexibility, supports, maintenance, robustness, optimizations
  - 2014: OOPS C++ support, vimpack
  - 2019 : fix C++ and ODB new issues
  - 2020 : Hub (Lockdown version!)
- **The « hub »**
  - a plug-in to host external libraries built with their own build system inside gmkpack in order to keep full control of compiler options consistency
- **Future:**
  - There is room for new developements, but anticipation is necessary to develop and test the new features

SYSTEM SESSION SUMMARY

Cycling (Claude Fisher)

- **Cycles:**
  - CY46T1_bf.06: additional fixes before a potential export version (bf.07?)
  - CY47; CY47T1
  - CY48: declared end of August 2020, CY48T1:
    - build started in October 2020
    - incremental integration approach (50 branches !!!)
    - intensive use of “davaï” testing tool
    - Declaration date not yet decided, pending validation of specific tests
  - CY49: Autumn 2021 or in 2022 (2 years gap !!!)
    - CY48T2:
      * pending decision on CY49 & discussion within MF/ACCORD
      * T2 for integration of pending contributions from the T1 + oper updates + bugfixes
- **2021-2023**:
  - Stable code for OOPSification
  - But expected impact due:
    * code adaptations to GPU
    * separation of concern implementation (data structures, DSL tool chain etc.)

SYSTEM SESSION SUMMARY

Building new cycles: towards new practices (Alexandre Mary)

- **Contribution workflow:**
  - Semi-private repo: 0. CYn decl **Developers:** 1. Develop branch, 2. Test, Fix issues
  - Public repo: 3. Push + Integration request. **Integrators**: 4.Review, 5. Merge, Test, Fix
    6. Update integration branch

- **Contribution : flow without barriers**
  - Central repository accessible : (no more manual transfers MF internal network)
  - Systematical tests: with a standardized set of CSC, **as a contributor's duty .**
  - Integration requests : web platform (github-like)
  - Reviews : avoid to discover issues later

- **Non- Continuous Integration** (Until 47T1)
  - Development/contribution window (2-3 months) => Merge (1 week) => Phasing, debugging and validation (3-4 months)
- **Continuous Integration** (from 4781)
  - Development/contribution window (5 months) => Final integrations (1 Month)
    * branches are continuously validated (extended contribution window)
    * Pool of integrators (merge conflicts solved by thematic Experts)

SYSTEM SESSION SUMMARY

Building new cycles: towards new practices (Alexandre Mary)

- **Testing : Davaï**
  - Unprecedented level of validation for common cycle CY48 (prevented a number of issues)
  - v1/Olive => proof of necessity for a portable system
  - v2/Davaï (Vortex, scripted definitions, Git-managed) : under development, prototype OK
    * Summer 2021: first release to partners
    * ACCORD collaboration for :
      - plug of a build system other than gmkpack
      - tests of CSC Alaro and Harmonie-Arome and other relevant tests
      - porting to ECMWF HPC (Vortex OK) Possible SPDAVAI for non-ECMWF
      - set of SP tests
    * Second phase:
      - porting to other HPCs
      - set of toy tests on workstation
      - input switch to bundle (separation of repositories, e.g. OOPS, SURFEX)
- **Bundled ecosystems**
  * Interface to pick projects (e.g. OOPS, SURFEX) to tie together the versions of projects that are compatible. Maintain NWP branches.
  * Tool ecbundle from ECMWF
  * Need to adapt contribution procedure in case it addresses several projects

XXXXX

- Text 1
  - Text 2
- Text 3
- Text a bit longer blablabla
  - Text xxxxx