

2020-08-20

Yannick opened by announcing that this week is a general round-table update.

He reminded all JEDI users to update their GitHub profiles with your full name and organization. Some users have ambiguous github user names and it's sometime difficult to know who they are. And, in the organization field, please put common acronyms, e.g. JCSDA, OAR, NWS, NCAR, UKMO, NRL, etc. This will allow us to search for these strings and properly categorize and monitor contributions from JCSDA partners and collaborating institutions, as demonstrated at the last JCSDA Quarterly review. Please see the [JEDI team discussion](#) for further information.

Then the discussion turned to summaries of the various JEDI projects.

Mark M summarized the JEDI1 meeting yesterday. One of the announcements made at that meeting is also relevant for the broader JEDI group: **We now have a JEDI user/developer forum** at <https://forums.jcsda.org/>. We are experimenting this as a possible way to deliver user support for JEDI users and developers that may not have write access to our private repositories and so, cannot open ZenHub issues. This support will become more important after the upcoming JEDI release planned for October.

All JEDI developers and users are encouraged to start using the forum. The forum is hosted by [Discourse](#) and you'll need to set up a Discourse account to post. However, you can sign up using GitHub so there is no need to set up a separate user name and password. Just follow the instructions. Currently it is set up so that anyone with a Discourse account can post so no further authentication is needed. We may change this in the future if there are unwelcome posts. Discourse is also integrated with slack - you can join the JCSDA jedi-forum slack channel if you want to receive slack notifications for when people post to the forum.

Specific bug reports can still go directly to ZenHub issues or GitHub teams but we can start migrating general usage questions to the forums. If the volume of posts warrants it, we can create separate categories for developers vs users, for models, observations, etc. Please give it a try and let us know what you think.

Mark M then proceeded with a further summary of the JEDI1 meeting. Another announcement is the availability of an integrated testing pipeline, thanks to Maryam. Now, if you push a commit in an active oops pull request with the commit message "trigger pipeline", this will run tests for oops, saber, ioda, and ufo. In the future we will add more options for running models when jedi base repos change.

He mentioned that we're behind on the August milestone, as is the JEDI project as a whole, but that is due to an ambitious estimate of what we think we can get done (August story points far exceeding completed story points in previous months). Mark and Yannick encouraged people to prioritize their issues - put the most pressing in the current milestone and others can go in next month. If you finish some things from this month you can always move issues into the current milestone. It's important to improve our estimates because it helps teams to focus on what needs to be done in our monthly sprints.

After reviewing the July and August milestones, the JEDI1 team then identified particular issues that need to be done prior to the release. Mark did not want to take the time to go into details but you can see these issues by going to the JEDI ZenHub workspace and filtering by the JEDI1 label and the Epic 5.1 (release). Mark encouraged team members to add issues to the Epic 5.1 (release) to help us prioritize our efforts in the next few months.

Then Ryan proceeded with an update for JEDI2. There has been a great deal of recent progress here, with 18 PRs merged into ufo in the past two weeks. This, together with the integrated testing pipeline mentioned above, led to some tests failing because of rate limits on pulling testing containers from Docker Hub (Maryam is addressing this by hosting our CI testing containers on AWS instead of Docker Hub). Ryan thanked the team for their efforts and summarized some of these PRs. He also mentioned that the JEDI2 team has been working closely with the OBS team on variable naming conventions. Some obs operators require variable names to match between the observations and models so it's important to clarify this. Ryan also mentioned continuing work on preparing ioda-engines for the release, including work on the cmake infrastructure and the tutorials and documentation.


Several people asked about the time line for integrating ioda-engines into ioda. Steve expects the first PRs to start to go in next week.

Guillaume asked about the plans for refactoring GeoVaLs. In particular, he'd like to see the TLAD trajectory allocated by GeoVaLs. Steve, Yannick, Ryan, and Anna responded that this is planned but probably won't be in before the release. There's other work that needs to be done before we tackle this, including the integration of ioda-engines, finalizing variable names, refactoring GeoVaLs to C++, and allocating the geovals in the constructor.

Then Dan proceeded with an update for JEDI3, describing progress from each of the model teams. The UKMO has been focusing on the UM model with a goal of reproducing their operational system. UM-JEDI is atlas based and there was some discussion at the JEDI3 meeting about how atlas data structure could be exploited to provide an interface between Fortran and C++ that does not require the use of linked lists. The UKMO is also working on a static B.

The MPAS group has been updating mpas-jedi in preparation for upcoming oops PRs. They have also been working on improving their interpolation and automated testing - they now have cycling DA applications in their CI testing. Chris H reported that the shallow water model is up to date with the latest oops PRs.

The Neptune team has been working to implement the NAVGEM static B for use with neptune-jedi. Yannick asked if there were plans to make this available in saber and Dan confirmed there were. Sarah later added that the NRL team has also been working to develop a pseudo-model that will allow them to integrate neptune-jedi into the JCSDA regression testing pipeline.

The SOCA team is working on scripts for LETKF cycling applications on different platforms, including Discover, in preparation for a September deadline. Another deliverable is to replace the Reynolds SST system at UMD  with a 2DVar JEDI system.

A question came up during the JEDI3 meeting about generic interpolation interfaces. They exist now in JEDI and provide an interface to atlas and bump interpolation. However, in order to use them, you need to express the input and output grids and fields as atlas data structures. So, in order for this to be used with GetValues, we plan to represent the Locations as an atlas PointCloud. Mark M is now experimenting with this for the qg model. If it goes well, the next steps would be to implement something similar for ufo Locations and then fv3-jedi. Dan explained that fv3-jedi is particularly tricky because in addition to interpolating real/double fields, there are also specialized interpolation routines for integers and vectors. Mark M expects that this can be worked into the generic interpolation framework. But, this will probably not all be done before the release.

The fv3 team is also working on a static B along with some other things, including improving the testing code coverage, performing runtime profiling and reading gfs cold starts. Ting and Cory are working on NWP for regional domains.

Chris S asked if the static B work is generic. Dan replied that initially no, but that's the eventual goal. He described three ongoing static B efforts:

- NAVGEM (NRL)
- UM (UKMO)
- GSI

For the first two, most of the current effort is going into implementation in specific models (Neptune, UM). After that is complete, we can look into making them generic and including them in saber. Dan further split the GSI effort into two components. The first component is the vertical balance, which currently resides in fv3-jedi and is model-specific. The second component is based on a recursive filter. This is generic and is now available in saber.

Anna then proceeded with a summary for JEDI 4. She skipped the static B update because that was already discussed. She mentioned Yannick's pull request that removes the need for model specification in the config files for 3DVar and 4DVar applications. This was merged today. Jake later asked if this PR had any impact on memory usage. Yannick said no - it should not. Anna also mentioned a few other PRs that were merged, including an RTPP inflation application added by JJ and progress with UM-jedi. She also mentioned several active PRs by Benjamin, including a major restructuring of global arrays in bump/saber that should improve memory efficiency, the addition of hybrid weights with fixed localization, and a **Hybrid 4DVar implementation in oops that needs reviews**. Anna then described a series of PRs by Sergey regarding local ensemble solvers and LETKF - the latter also **needs a review**. Another PR that **needs reviews** is one by Anna that removes the MakeObs application for creating artificial observations for testing. This functionality is now present in HofX. Anna then quickly summarized what some of the other JEDI4 team members are working on including time parallelization, KD-tree search for LETKF and local ensemble filtering. She also pointed interested parties to a discussion on the JEDI teams about when QC filters should be run.

Yannick then followed with a summary of JEDI5 activities, which are focused on the upcoming release and on creating documentation in preparation for the release. He described two upcoming (virtual) code sprints:

- 21-25 September: Writing more tests to improve test coverage
- 5-9 October: Writing more documentation

Both are in preparation for the release. In the sprint planning we are now working to identify code components that are not covered by the tests and areas most in need of documentation. All members of the JEDI core team will participate in both sprints, along with others. We may contact you or you may be encouraged by your organizations to participate. Let us know if you wish to participate.

We then addressed a question in the chat from Rahul about whether ioda-engines would be backward compatible. Steve described three stages in the development

1. support for current file formats
2. a converter to change current files into the new format
3. modifying the ioda-converter themselves to write to the new format

Jake then reported that as they move to higher-resolution applications with mpas, they are finding a need to optimize memory usage. There was some discussion that this is a known issue for JEDI (not just MPAS), particularly in saber, oops, ioda, and ufo. This had not been a top priority but after the release we intend to work more intensively on identifying and mitigating performance bottlenecks. Yannick mentioned that the interpolation in particular is currently too slow.

Chris S mentioned that they are looking for more obs data to run applications for different time periods. Plus NCAR is collaborating with EMC in the development of fast (ie, compiled code) BUFR file converters.

Then Mark O proceeded with an update of some of the work he is doing. This includes improving the handling of C++ include files and libraries to be more robust and to follow good coding standards (as defined by google). He has already worked with the shallow-water and ioda repos and is now working on mpas. Along the way, he is also working on a cmake build system for mpas. This was stalled by some work on the jedi-cmake repo but he expects to issue a PR next week. He also described improvements to the cmake code in ioda-converter that removes the need for ebuild. He explained that ebuild can have conflicts with general cmake that can cause problems. For example, ebuild hard-codes in lib as an install destination for libraries but this needs to be lib64 on some systems. He has issued an upstream PR to ecmwf ebuild to address this. All this cmake work should improve the portability of jedi for the release.

Mark O then described more generally the function of the new jedi-cmake repository (renamed from jedi-toolchains). The motivation for this is that ecmwf wants to keep ebuild lightweight and was resisting some of our extensions. So, jedi-cmake is intended as a place where we can put cmake functionality above and beyond ebuild without being subject to upstream reviews. Currently the repo includes CMake Find modules (for NetCDF and GPTL) as well as toolchains that can be used to specify cmake configurations for different platforms. jedi-cmake is generic (not jedi-specific) and public, with no dependencies. So, it can be used for other cmake projects as well.

Then there was some discussion of a question in the chat from Rahul about whether interpolations will be available to convert from generic unstructured grids to gaussian lat-lon grids for analysis. Mark M responded that this is more of an atlas issue than a JEDI issue. The interfaces are already there in JEDI to access the atlas interpolation, provided that you use atlas data structures (functionspaces and fields) to represent your grids and fields. But, Yannick explained that parallel interpolation in atlas currently only works if the input and output grid share the same parallel distribution. They are working on generalizing this to allow for generic interpolation between different functionspaces.

Yannick closed the meeting by reminding people that next week will be a focused topic discussion. A topic has not yet been chosen - possibilities include Jake leading a discussion on RTTOV or a discussion about how to do code reviews efficiently and productively. Please [let us know](#) if you have any preferences for next week or any ideas for future discussion topics.