

We strongly agree that it is important to improve the collaboration and coordination between Unidata's IDV and SSEC's McIDAS-V development efforts, and that input and contributions from knowledgeable community members must be streamlined and encouraged. While we agree with much of the roadmap you lay out for achieving that goal, there are a number of areas where our thinking does not quite align and, as you say, further discussion will be needed.

Below are some of the main areas we feel need further discussion.

Communication

An area where there seems to be strong agreement is that increasing the level of communication and trust between the groups is key to moving forward with the vision of enhanced collaborations, and it is something that we are committed to work on. To that end, we propose that the Unidata IDV developers visit Madison this fall for a non-trivial amount of time (e.g., several days) so that members of both development groups could get to better know one another and lay the groundwork for greater coordination of mutually-agreed upon work. To ensure continued close communication, we propose developing a somewhat regular meeting schedule for the full team (using "team" as defined in SSEC whitepaper, roadmap item 2), e.g.,

- Quarterly/semiannual/annual face-to-face meetings (alternate between Madison and Boulder).
- Team teleconference on monthly time frame.
- Weekly high-level coordination phone calls (the designated coordinators you discuss in your roadmap item 3).

Joint Technical Advisory Group

Improvements to the inter-group communication and coordination discussed above will work to align plans and priorities. In the case of disagreement around technical direction or goals/priorities not aligning, the team will work to build consensus. To help with this effort, we might consider forming a joint technical advisory group from which we can seek guidance on priorities and overall design/architecture.

Merged Product

We believe it is important to allow development of new capabilities for diverse users while making sure any new capabilities do not affect other users. This would allow each site's offerings to stay branded in a manner familiar with existing user bases -- satellite meteorology focused plug-ins for McIDAS-V and mesoscale meteorology focused plug-ins for the IDV. The IDV plug-in architecture already supports much of this goal. If needed, we can look for ways the package architecture may be evolved and for other desirable refactoring opportunities.

Merging the products in ways that change the current branding or control over the future direction of the respective packages would need to be considered in depth and, on the Unidata side, involve the Unidata Users and Policy Committees and must take into account the needs and interests of the Unidata community.

Support

Given Unidata's history with McIDAS and the amount of common source code in the IDV and McIDAS-V, an extension of the McIDAS-X support model would seem to be a reasonable path for the future.

Given Unidata's limited resources and ambitious goals, the potential need to support users of McIDAS-V raises critical resource implications for the UPC that will require decisions on how many analysis/visualization tools Unidata can and should support. This issue will need to be discussed with the Unidata Users and Policy Committees before we can make a decision on UPC's support of McIDAS-V.

Whatever the support model:

- Both the IDV and McIDAS-V support systems must be accessible to all team members.
- We must agree on a process for handling cross-community support, e.g., if a MUG member asks an IDV question or a Unidata community member asks a McIDAS-V question.
- We must consider how IDV eSupport and McIDAS-V forums should interact.
- We must agree on how support issues should bubble up to issue/bug tracking system.

Source Code Repositories

While we agree that source code repositories must be usable by the entire IDV/McIDAS-V team, gaining write access to a code repository maintained by an external group requires a level of trust that needs to be developed over an extended period of time. Such was the case for Tom Yoksas getting access to the SSEC CVS repository for McIDAS-X, and for Jeff McWhirter getting access to Bill Hibbard's repository for VisAD. In both cases, however, the maintainer of the code repository (SSEC/MUG for McIDAS-X and Bill for VisAD) had and continues to have the ultimate say in what changes are incorporated into the main development trunk. This model is also what is followed for really big development projects like the Linux kernel where Linus Torvalds holds the ultimate authority for what does or does not get included in the released product.

There is already a high level of trust between our two groups. That trust will continue to develop as we move forward with this collaboration. In the meantime, whether we eventually unify our source code repositories or not, we must ensure that source code repositories for all joint project code are visible to all team members. Further, we must agree on workflows for contributing changes and new code to those repositories. These workflows would include direct write access to the repositories, patch submission, and possibly others (e.g., those enabled by DVCS). Approved developers would have write access to the repositories, others might be required to submit patches. Another possible step would be to conduct joint code reviews on larger code changes/additions. This could be done using distributed code review tools that would not require meetings or even synchronized schedules.

Unidata is moving towards a distributed code repository model of development using git and GitHub. While still in the evaluation phase, we believe that

git allows the right kind of collaborative sharing, by giving other developers full read rights to the source repository, allowing branches to be easily created which contain discrete pieces of development, and allows the owner of the central repository to decide which features should go into which releases. We recommend that Unidata and SSEC jointly evaluate this workflow for this project.

Issue/Bug Tracking Systems

Similar to source code repositories, we must ensure that issue/bug tracking systems for each project are accessible to all team members and allow submission of and comment on issues by all team members. (Issue/bug tracking systems must also support referencing issues in other systems.) We should also develop agreement on:

- Time period within which submitted issues will be assigned and scheduled.
- How submitters can request elevation of priority.
- What to do if developers can't agree on priority (see the "Joint Technical Advisory Group" section).
- Some consideration should also be given to the quality of submitted issues. The more work put into submitting a quality issue, the better the likelihood the issue will get a higher ranking/prioritization. E.g., an issue that is only a problem statement is likely to not be ranked as high as one that also suggests where in the code the problem lies. Adding a test for the issue that fails and even a possible patch makes for even better ranking.

Other Release Engineering Details

We agree that we must "create a shared process for routine testing of changes and enhancements" [Roadmap 2c]. As part of this, we would like to ensure coordination of automated nightly build, test, and release systems for all layers of the joint project. The results of these systems must be accessible to all team members.

As part of the planning and prioritization of bugs and enhancements, we should work together to coordinate software versioning and release schedules. The shared testing process should detail how the various levels of testing (e.g., unit, integration, and system level testing) aligns with versioning and release schedules.

Next Steps

Given our mutual commitment to collaboration and a strong desire to move forward, here are some suggestions for next steps in no particular order (some of these are already underway):

- [Joint] Developer meeting in Madison.
- [Joint] Continue discussion to further develop roadmap.
- [Unidata] Simplify SSEC (and community) access to IDV code repository.
 - [Unidata] Create git repository (on gitHub) for IDV code [done].

- [Joint] Evaluate use of git with one or more collaborative efforts [underway].
- [SSEC] Simplify Unidata (and community) access to VisAD and McIDAS-V code repositories.
- [Unidata] More consistent use of JIRA issue tracking system.
- [Joint] Integrate McIDAS-V ADDE Chooser modifications into IDV [ongoing].
- [SSEC] Refactor Hydra as an IDV plug-in.