2018 ESS PI Meeting Breakout Session

Building Community Testbeds for Modeling Watershed Systems

Tuesday, May 1, 2018

The following write-up captures an ESS community discussion on the design requirements for watershed system and integrated hydrologic modeling testbeds.

The goal of this breakout session was to generate community recommendations on the ideal capabilities and platforms for future watershed testbeds. Participants were asked to specifically consider:

- Existing hydrologic resources and successful approaches to community modeling
- Potential users and their ideal workflow and simulation requirements
- Data requirements and data handling approaches
- Modeling platforms and tools to facilitating analysis and visualization

Questions posed for group discussion

1. Key capability requirements or biggest challenges that potential users are facing
2. Existing resources and successful approaches in other areas that could be applicable here
3. Biggest outstanding development needs or hurdles to implement a successful community testbed

Summary of group discussion

Most groups were stuck on the definition of 'testbed' and 'community.' There were many questions about what the goals of a testbed should be and who are the users. For example, should the testbed be geared towards improving the simulation capabilities for model developers, or facilitating easy access and comparison for users interested in setting up and running models or users who use model outputs for analyses? The answer is probably for all those different types of users. Despite the ambiguity of the testbed definition, the participants were able to come up with a list of design requirements that would be useful for a community testbed.

Design requirements useful for a community testbed

- The testbed should cover synthetic and real systems over a range of complexity. New model formulations can be verified on smaller synthetic one- or two- dimensional problems that run quickly (such as the sandbox idea). Complex watershed systems with real data can be used for understanding process coupling at watershed scale and benchmarking. In this way, the testbed would be appealing to both model developers and model users.

- The testbed should allow consistent model setup using different codes for fair comparison. Developing standardized inputs, metrics and algorithms for evaluation would allow people to more easily compare different models at the testbed.

- The testbed should enable easy parameterization for large-scale models. Access to standardized data is the key. Interacting with community databases (e.g., ESS-DIVE and others) is essential.

- Data and model versions as well as systems for building the models need to be preserved along with output in order to ensure data and model provenance and reproducibility.

- Containerizing models developed at the testbed would make the models easily transferrable to other systems and users without the hassle of configuring the exact computing environment. With this approach, ultimately many users would be able to build a national testbed.

- Uncertainty quantification and data assimilation should be built into the testbed to better facilitate model-data integration. Ensemble simulations need to be supported in this context. This type of infrastructure to enable experimental design informed by models will accelerate knowledge discovery.

- The testbed should provide test cases of systems that are known for insufficient model performance. This way, community efforts could be directed to improve model performance in these targeted areas.

- Testbed should support routine model testing such as regression tests and scaling tests.

- Models should 'live' within the framework with strong modularity to allow easy assembly of alternative conceptual models with multiple selections of contributing components or modules. This is critical for parsing model structure errors.

- The testbed should have multiscale multi-resolution modeling capabilities.

- As a community resources, documentation is the key to success. Therefore, the testbed should come with documentation tools that would make it a fun process rather than a pain. Tools like Jupyter Notebook could be explored.

- In the era of big data and emergence of artificial intelligence, one remaining question is how do we compete against Microsoft, Google, Amazon, etc? Alternatively, another question is really how we can leverage these industry resources?