

UFS Workflow

An overview of ongoing discussions from
the perspective of someone on the edge
of the it (4/9/2021)

*Dr. Ir. Hendrik L. Tolman
Senior Advisor for Advanced Modeling System
Office of Science and Technology Integration
National Weather Service / NOAA*

Hendrik.Tolman@NOAA.gov

Basic Principles

Unified Forecast System:

- Unified, not Unitary, balance of
 - Focusing resources to get critical mass
 - Allowing for diversity as needed
 - Modular tools versus “one size fits all”

Start from requirements, not solutions

- Functional requirements
 - Software engineering perspective
 - System engineering perspective

Software engineering

Hierarchy of software elements is generic,
examples are from NCO,

- The programs at the core of each step of a workflow (observation processing, DA, models, product generation etc.) (**UFS code**)
- “Scripting” to manage the basic operation of these programs (a “task”) (**“ex” scripts**)
- Configure and actual run of the code (when, where, storage, etc.) (**JJobs**)
- The functional scheduler (**Sequence JJobs**)
- The operating system scheduler (**ECflow**)

Systems engineering #1

Work is already underway!

Functional elements of a workflow:

- Input data processing
- Model initialization
 - Data Assimilation (DA) mandatory for initial value problems (ocean, atmosphere)
 - Model cycling (with or without DA) for forced problems (wind waves, coastal ocean modeling)
- Producing the forecast (model run, can include “raw” products)
- Producing products (including model corrections)

Systems engineering #2

For a (standardized) workflow to be used throughout the UFS, it needs to be

- Easy to use (as simple as possible for operations, fully documented)
- Flexible / modular / extensive (particularly for research)
- Efficient (always critical), from a systems perspective driven by
 - Latency, i.e., time between steps in a workflow
 - Resource utilization, i.e., avoid idle processors

Systems engineering #3

Critical efficiency aspects vary per application

- For rapidity cycling systems, latency is likely to be the critical factor
 - RRFS with DA step ?
- For slowly cycling systems, efficient use of assigned resources is likely the key efficiency factor
 - GFS with single DA step ?
 - “Offline” product generation vs. model run !

Implications of this for Unification:

- Use shared common elements to generate a small set of workflows

The Matrix take1

Software and systems engineering considerations create a basic matrix of things to consider

Next development steps

- Top down approach
- Identify shared elements
 - The “Library”
- Many previous discussions allow us to tentatively refine the simple minded matrix shown here
- Next page is just a first cut, nothing final!

	Input processing	Initialization /DA	Model run	Product Generation
program	X	X	X	X
script	X	X	X	X
Run config.	X	X	X	X
Functional scheduler	X	X	X	X
OS scheduler	X	X	X	X

The Matrix take 2

	Input processing	Initialization /DA	Model run	Product Generation	
Program (code config.)	code / repository management standards same for all				The "Library"
	Obs processing IODA, restart files, etc.	Stand alone DA configuration	Stand alone model config.	UPP, ensemble processing tools, etc.	
Integrated DA / Model config.					
"script"	Standard execution environment for each element above, engineered to allow for stringing individual elements together				
Run config.	Standardized naming conventions for configuration of all "scripts" Automate combining configurations for scripts when used together				
Functional scheduler	Workflows tailored for application / experiment Workflow is a sequence of "library" elements New capabilities are introduced as library elements UFS selects community OS Scheduler				
OS scheduler					NCO adopts UFS scheduler, or EMC "translates" between schedulers

The "matrix view" of the workflow is a simplification to allow visualization. In order to implement this high-level vision, the next step is to develop a comprehensive analysis of all tasks (both functional and developmental) associated with it.



Charge to team

Work is already underway!

We ask the workflow team to do the following

- Start with software / systems engineering approach as laid out here
- Develop an agreed upon general matrix layout as on the previous slide, including a comprehensive set of task associated with defining and implementing it.
- Estimate effort needed for building elements of a unified workflow
- Prioritization of work to be done, including
 - Which Library elements to built first, and
 - when and how to transition actual workflows

Questions?