A Workflow Management System for Automated Weather and Climate Simulations

Christopher W Harrop[1,2], Ligia Bernardet[1,4], Mark Govett[1], Jeff S Smith[1,3], Stephen Weygandt[1]

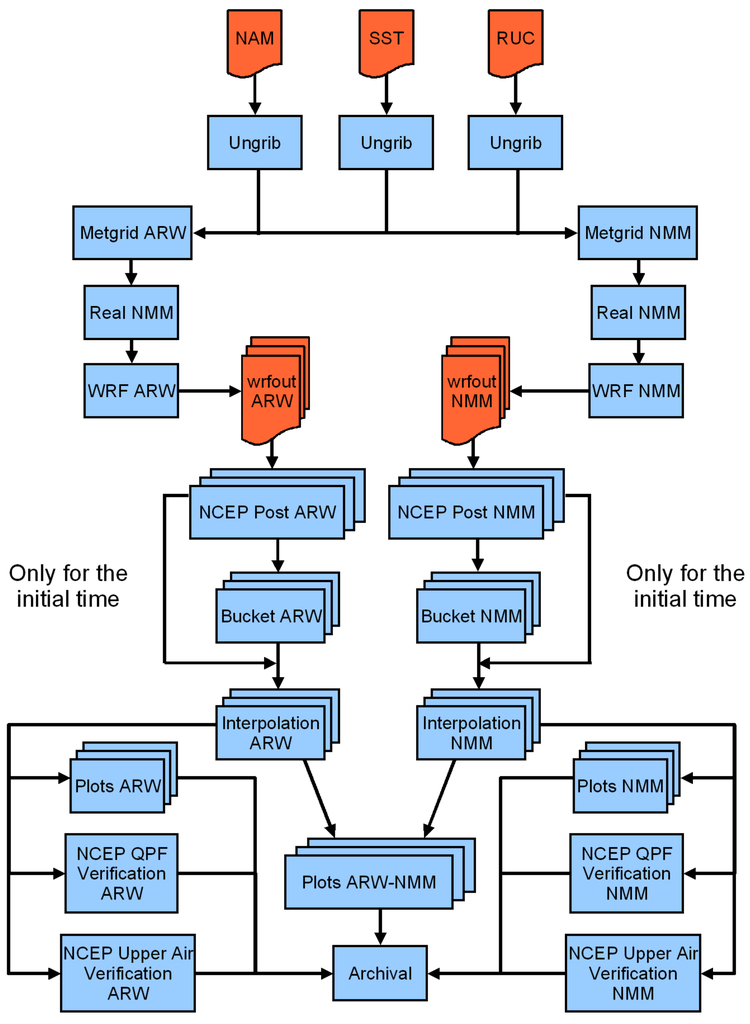
NOAA - Earth System Research Laboratory - Global Systems Division[1], Cooperative Institute for Research in Environmental Sciences[2], Cooperative Institute for Research in the Atmosphere[3], Systems Research Group[4]

Introduction

One of the challenges of modeling and simulating weather and climate phenomena is the onerous process of running the various codes that are required to carry out a simulation. There are several factors that contribute to this difficulty.

- An end-to-end run of a simulation often requires running a large number of components.
- The simulation components can have complex interdependencies and runtime requirements.
- Simulations that run in real time must complete before the resulting forecasts become irrelevant.
- Many experiments require 100's or even 1000's of end-to-end runs.
- High Performance Computing systems are notoriously difficult to use and are not always reliable.
- Monitoring and tracking simulation progress is labor intensive and prone to error.
- Ad hoc scripting methods for end-to-end simulation orchestration are not feasible.

A Conceptual View of a Realistic Complex Large-Scale Workflow



For a 48 hour forecast, with output every three hours, this workflow would consist of a total of 182 tasks.

Task Count	Task Type	Description
5	WPS Tasks	Three ungrib tasks, plus one metgrid tasks for each core
4	WRF Tasks	One real task and one WRF task for each core
34	Post Tasks	One post task per output time for each core
32	Bucket Tasks	One bucket task per output time, excluding the initial time, for each core
34	Interpolation Tasks	One interpolation task per output time for each core
51	Plotting Tasks	One plotting task per output time for each core, plus one difference plotting task per output time
4	Verification Tasks	One verification task per verification type for each core
1	Archival Task	One archival task to archivel workflow output

Workflow Management System Solution

Definition: A ***Scientific Workflow Management System*** is software that helps scientists compose scientific workflows and automate their execution.

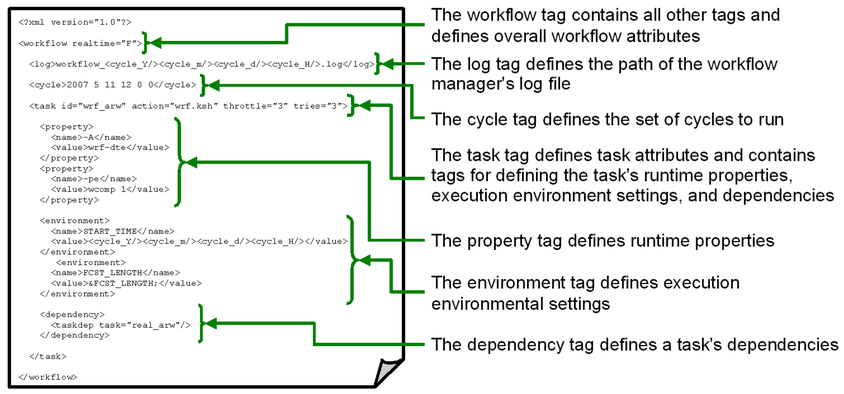
Definition: A ***scientific workflow*** is a collection of data and computational tasks along with a description of their interrelationships and runtime requirements.

A workflow management system provides scientists with two key functions:

1. a mechanism for composing workflows; and
2. an engine for managing workflow execution.

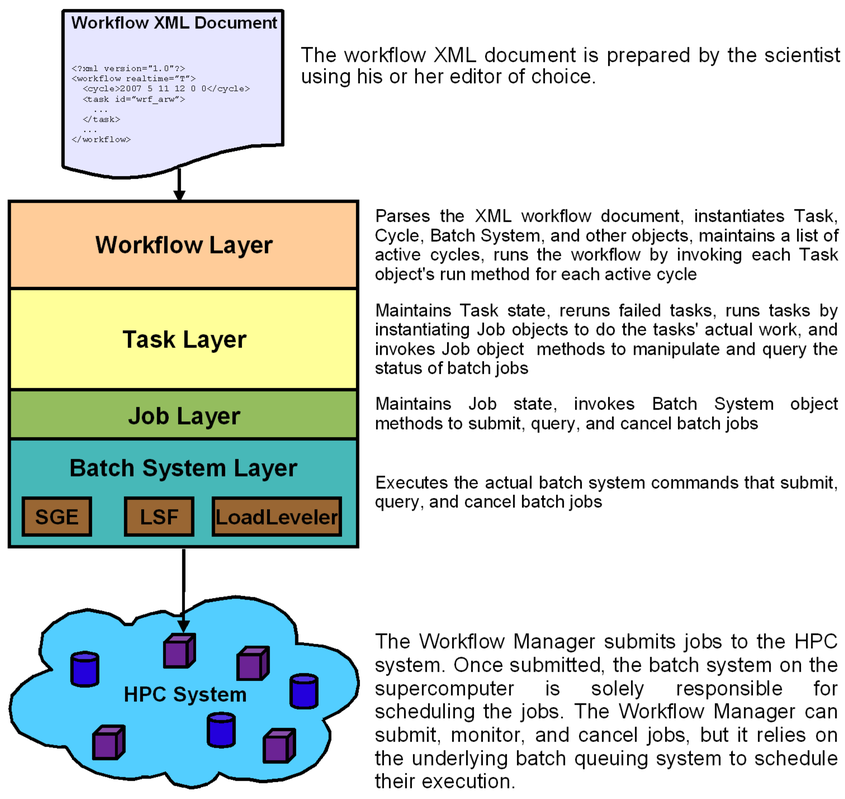
Composing Workflows

We have devised a custom XML-based language for composing workflows. The language is simple, non-esoteric, and uses terms meaningful to most scientists. It consists of a set of "tags" that are used to describe workflow structure. Scientists use an editor to arrange the tags and set the tags' attributes to define the structure of their particular workflow.



The Workflow Management Engine

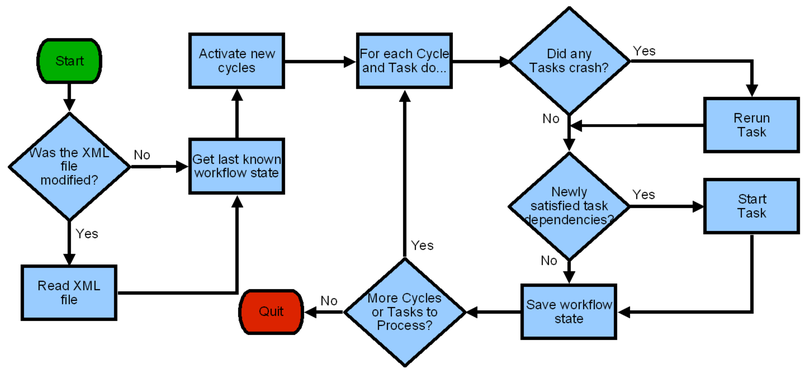
The Workflow Manager has a layered, object-oriented architecture. Layers communicate with each other via standard interfaces. Support for additional batch systems is achieved by implementing a new batch system class in the Batch System Layer.



Using the Workflow Manager

The Algorithm

The workflow manager implements the algorithm depicted below. Each time it runs, it detects task failures and newly satisfied task dependencies and starts or reruns tasks accordingly. Thus, repeated invocations (usually as a cron job) of the workflow manager make incremental progress until the workflow is complete. The current state of the workflow is maintained in permanent storage between runs, and each run usually finishes in a few seconds.



Command Line Usage:

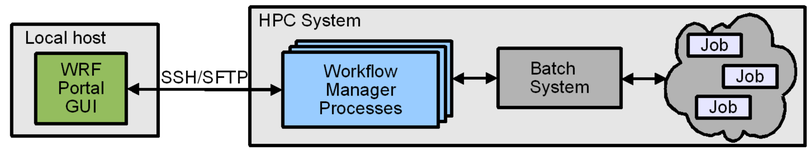
```
$ workflowmgr.rb --xml=/path/to/xmlfile --store=/path/to/statefile
```

Crontab Usage:

```
*/2 * * * * workflowmgr.rb --xml=/path/to/xmlfile --store=/path/to/workflow/statefile
```

A Core Component of WRF Portal

Our workflow manager also provides core functionality for WRF Portal. WRF Portal is a Java application that simplifies model testing an evaluation by providing modelers with an intuitive graphical user interface for composing, running, and monitoring workflows. Using WRF Portal, scientists can easily create, organize, and track large numbers of model configurations, model runs, and visualize the results. The diagram below shows how the workflow manager is used by WRF Portal to coordinate the execution of simulation components.



Practical Experience

Our workflow manager is designed to solve the practical simulation management problems that scientists face on a daily basis. This means we focus on solving the problems that scientists encounter during their real-world modeling experiences rather than focusing only on "grand challenge" applications. As a result of this philosophy, our system has played a key role in the success of several real-world modeling applications. Below is a list of some recent experiments that made use of our workflow manager.

Experiment Name	Type
WRF Test Plan	Retrospective
WRF Rapid Refresh Core Test	Retrospective
DTC Winter Forecasting Experiment (DWFE)	Realtime
NMM5-CONUS	Realtime
Terrain-induced Rotor Experiment (T-REX)	Retrospective & Realtime
Rapid Refresh Development	Realtime
Rapid Update Cycle Benchmark	Retrospective & Realtime

Conclusions

- The success of our system has been demonstrated by its practical use in several real-world experiments
- A focus on solving daily practical issues results in a greater amount of "buy in" by the end-user scientists
- Many scientists now consider our system to be a crucial tool for conducting their experiments
- Our system has dramatically increased the reliability and efficiency of several experiments
- We have plans to make the following improvement
  - Add workflow language constructs to make it easier to create and maintain a large number of nearly identical tasks
  - Add quality of service workflow language constructs to enable the specification of start and end deadlines for real-time task execution
  - Generalize task runtime property specification to allow workflow documents to be portable across different batch systems
  - Implement the capability to "roll back" a workflow so that a scientist can make changes and rerun tasks that have already completed successfully (currently it is only possible to rerun tasks that have failed).