

12B.1 INTEGRATION OF LEAD AND WRF PORTAL TECHNOLOGIES TO ENABLE ADVANCED RESEARCH, OPERATIONS AND EDUCATION IN MESOSCALE METEOROLOGY

Craig A. Mattocks*¹, Kelvin K. Droegemeier², Robert B. Wilhelmson³ and Jeff Smith⁴

¹University of North Carolina, Chapel Hill, NC

²University of Oklahoma, Norman, OK

³National Center for Supercomputing Applications, University of Illinois, Champaign, IL

⁴NOAA Earth System Research Laboratory, Global Systems Division, Boulder, CO

1. INTRODUCTION

Imagine configuring an advanced, non-hydrostatic numerical weather prediction model (WRF-ARW) through an elegant graphical user interface, laying out multiply-nested grid domains, specifying physics and numerical techniques options, selecting analyses for initial/boundary conditions, choosing which sources of real-time meteorological data (including NEXRAD Doppler radar data) to assimilate, spawning an ensemble of simulations on a vast array of remote supercomputing clusters, then creating richly textured 4D visualizations of the output as the model runs all through a web browser from anywhere in the world. This is not a fanciful daydream; it is now possible to do all of this, plus unleash event-triggered dynamically adaptive forecasts, through the Linked Environments for Atmospheric Discovery web portal (<http://portal.leadproject.org>).

LEAD (Droegemeier et al., 2005; Droegemeier, 2009) is a National Science Foundation Large Information Technology Research project that empowers researchers and students with capabilities heretofore available at only a few major universities and research or operational centers, and it does so using a service-oriented architecture (SOA) similar in many respects to the familiar Amazon.com, where storage and processing resources are provided via the “cloud”, a highly scalable pool of virtualized, computer resources accessible through a web-based interface. By managing the complexity of interoperative cyber tools and providing flexibility and ease in how they can be linked, LEAD allows users to focus their time on solving the science and engineering problems at hand, providing a means for more deeply understanding the tools

and techniques being applied, rather than struggling with the nuances of arcane data formats, communication protocols, software installation, and job execution environments. LEAD has completed five years of research, including the development and testing of a prototype system, as well as operational testing in collaboration with the NOAA National Severe Storms Laboratory and Storm Prediction Center.

The LEAD project has propelled research in cyberinfrastructure, virtual organizations, and large-scale software frameworks that support analysis pipelines (i.e., user-assembled workflows, Fig. 1) that contain a mix of community and custom models and execute on selected TeraGrid resources under a community account. We describe in this paper several recent developments including a collaborative effort with NOAA/ESRL to merge the best aspects of the LEAD portal (ability to submit large simulations across the TeraGrid, multi-level fault tolerance and recovery, drag-and-drop workflow assembly, data mining and feature extraction) with those of the WRF portal (rich desktop Java graphical user interface with horizontal and vertical domain editors, support for multiple/two-way grid nesting, diff tool for comparing workflows), while creating new portal capabilities (namelist file error-checking/validation to prevent input and configuration errors, vertical grid-stretching tool, ability to edit compile and manage application source code). We are working toward deploying LEAD as a formal community facility, if sufficient funding becomes available. This would make it openly available to support research, education, and operational experimentation, including the emerging concept of warn on forecast vs. warn on detection.

*Corresponding author address: Dr. Craig A. Mattocks, UNC Institute of Marine Sciences, 3431 Arendell St., Morehead City, NC 28557, USA. E-mail: cmattock@email.unc.edu.

1. LINKING LEAD AND THE DTC

The principal goal of this project is to develop enhanced, interoperable capabilities between the LEAD NSF cyberinfrastructure system and the NOAA/ESRL/GSD WRF Portal (Govett and Smith, 2008), the latter used at the Developmental Testbed Center (DTC) (<http://www.dtcenter.org/>). The DTC, part of the Joint Numerical Testbed (JNT) at the National Center for Atmospheric Research (NCAR), provides a national collaborative framework in which the numerical weather analysis and prediction communities can interact to accelerate the development/testing of new technologies and techniques for research applications and speed their transition to operational implementation – all in a way that mimics, but does not interfere with, actual forecast operations. The new fused LEAD-WRF portal, with its more advanced and intuitive graphical user interface (GUI), will provide unprecedented forecasting capabilities by enabling users of all levels of sophistication and institutional capability to configure and run numerical weather simulations on powerful computing resources. Initial efforts have focused on enhancing the editing and validation capabilities of the namelist.input files that are used to configure and run the WRF numerical weather prediction model.

The default WRF namelist editor in the LEAD portal resides in the Experiment Wizard (Fig. 2). This tool, which operates on a small subset of physics parameters that are commonly modified in operational forecast ensembles, is ideal for users who prefer a lightweight web interface and only need to make a few simple changes to a namelist.input file. Dynamic highlighting (white color) draws the user's attention to the parameter that is currently being edited.

LEAD and NOAA scientists decided that the most effective initial approach for linking the two portals would be to “loosely couple” them by including a hyperlink (URL) in the LEAD portal that points to a Java Web Start (JWS) Java Native Launch Protocol (JNLP) file that launches the WRF Domain Wizard (WDW) (Smith et al., 2008). A LEAD user would then configure the WRF model using the WDW and save his/her desired namelist.input files on a local file system for subsequent upload to the LEAD portal. However, the WDW was originally designed to connect directly to remote supercomputing clusters using SSH (Secure Shell) authentication, while LEAD uses X.509 certificates. In addition, the WDW

required that the WRF Preprocessing System (WPS) be installed either on the cluster or on the user's local desktop machine so the WDW could launch the WPS geogrid, ungrib, and metgrid executables. To bypass these restrictions, NOAA/ESRL developed a separate website that serves up a customized “LEAD version” of the WRF Domain Wizard (<http://www.wrfportal.org/DomainWizardForLEAD.html>), shown in Fig. 3. To avoid having to maintain multiple disparate code bases, a new optional command line parameter (“LEAD”) is passed into the Java Virtual Machine (JVM) on startup to launch this version of the WDW.

Significant architectural changes are required to implement the WRF Domain Wizard in the LEAD Experiment Builder (Fig. 4). LEAD computer scientists are currently working on implementing a portal file uploading capability to support the use of WDW-LEAD for configuring the WRF NWP model. The upload capability consists of three related parts: (1) an upload user interface in the LEAD portal, (2) a plugin to parse the namelist input file and store it as metadata, and (3) a query capability to make the experiment configuration parameters available to LEAD workflows.

2. WRF NAMELIST VALIDATION

The second major goal of the LEAD-WRF portal fusion project is to develop a new namelist input file error-checking/validation tool, infused with numerical modeling intelligence, to prevent the generation of illogical, corrupt namelist files. This tool's components guide users through parameter selection, drastically reduce user frustration and conserve computing resources by ensuring that simulations submitted to remote job queues don't fail due to easily preventable errors (CFL stability violations, excessively small MPI patch size, specification of incompatible physics options, incorrect grid nesting hierarchies and layouts, inconsistent start/stop dates, etc.).

The first step in creating a namelist validation capability was to develop a fairly extensive set of low-level string, number, and date validation functions in Java and run them through a suite of unit tests using the TestNG framework (<http://testng.org/doc/>). Unit testing, one of the fundamental principles of test driven development (TDD) and Extreme/Agile/Pragmatic programming, is a method of designing software that guards against introducing errors during refactoring and maintains code in a continuously functional state.

The results of running an early version of the suite of tests are displayed in the test report (html format) generated by TestNG shown in Fig. 5. Note that one test has been forced to fail on purpose to demonstrate how an error would be detected and displayed (red color).

Next, high-level namelist validation rules and namelist-specific functions, such as `isValidWRFgroup()`, `isValidWPSgroup()`, `isGroupStart()`, `isGroupEnd()`, were developed to provide basic, first-pass, stateless error-checking on a WRF namelist.input file. These functions were augmented by a set of “smart” Java enums, one for each parameter in the namelist, that screen for an allowed range of values and store the line number at which each parameter occurs in the namelist.input file for subsequent, high-level, multi-parameter validation tests.

A boolean `isNamelistValid()` function is used by the `NamelistAnalyzer` class to validate the entire contents of the namelist. All errors are stored in a mutable structure (an auto-sorting Java `TreeMap`) within an `ErrorInfo` object. If the `isNamelistValid()` method returns `false`, the `ErrorInfo` object is interrogated to find the problematic line number(s) in the namelist, as well as the error message(s) that indicate what is wrong and how to fix the problem. The WDW GUI pops up an error message dialog box to alert the user of any problems (Fig. 6).

A numerical stability checker, which analyzes the grid spacing and time step at all grid nesting levels in the WRF namelist.input file to prevent the submittal of simulations that are likely to go unstable, was developed and incorporated into the WDW code base. It was decided to initially employ practical “rules of thumb” in this validation function, rather than implement complicated CFL criteria based on the numerical schemes selected, their accuracy, the numerical order of the computational viscosity chosen, etc. For the WRF-ARW (WRF-NMM) core, the rule of thumb is to set the time step in seconds to about 4-6 (2.25) times the horizontal resolution (in km). A demonstration of its capabilities is shown in Fig. 6. After a user alters the horizontal grid spacing and/or time step in the Namelist Editor and saves the namelist.input file, he/she can press the “Validate” button to check for configuration problems. In the example shown, a user has attempted to use a time step of 60 seconds for a 2.7 km WRF-ARW simulation. In this case, the algorithm alerts the user and

suggests that a time step of 16.2 seconds should be used to maintain stability. After the user follows this recommendation and modifies the value for the variable `time_step` in the namelist.input file, saves this configuration, and presses the “Validate” button once again, the Java function `integralTimeStep()` recommends that the user adjust the time step to 16.1435 seconds if he/she wants an integral number of time steps per hour, in this case 223 (not shown).

A physics suite compatibility checker has also been developed to prevent LEAD users from specifying inconsistent/incompatible physics options before they attempt to launch a doomed WRF simulation on the TeraGrid. A new Java `Ensemble` class has been created that holds `EnsembleMembers`, which are reference WRF physics configurations that have been proven to run well. (The current list of 23 members will be gradually expanded over time.) The `NamelistAnalyzer` compares a user's configuration against these reference configurations and issues a warning if there is no match. Several sources of information have been used to construct these proven reference ensemble members for this tool: (1) a list of combinations of physics parameters used in the US Air Force Weather Agency's (AFWA) Joint Mesoscale Ensemble (JME), shown in Table 1, (2) a list of the members used in the LEAD/CAPS 2008 Spring Real-time Storm Scale Forecast Ensemble (Kong et al., 2007), shown in Table 2, which is partially based on NCEP's Short-Range Ensemble Forecasting (SREF) system, and (3) written documentation from the WRF-ARW User's Guide, lecture notes from WRF tutorials, and the namelist descriptions provided with each WRF software release. Information has also been harvested from advanced LEAD users, forecasters at operational centers, and scientists at national research laboratories regarding the combinations that are commonly used and known to produce high-fidelity WRF simulations. Beyond general compatibility matrices, the physics suite checker also issues warnings if a user has specified that a cumulus parameterization scheme be activated on a grid with a horizontal resolution finer than 5 km, if an incorrect number of soil layers has been requested for the land surface model, etc.

3. WRF VERTICAL GRID STRETCHING

With permission from scientists at the University of Oklahoma's Center for Analysis and Prediction of Storms (CAPS), the vertical grid stretching

algorithm was extracted from the ARPS (Advanced Regional Prediction System) model (Xue et al., 1995; Xue et al., 2003), rewritten as a standalone application for use with WRF, ported from Fortran 90 to Java, and integrated into the WRF Domain Wizard. The screenshot in Fig. 7 shows the vertical grid stretcher being used in the Non-linear Level Generator to configure the σ_p (eta) levels in a quadruply nested WRF simulation of Hurricane Frances (2004).

This new capability optimizes the computational efficiency of WRF by concentrating the layers where they are needed to resolve phenomena of interest. The hyperbolic tangential stretching option, for example, can be used to pack multiple uniformly spaced layers into the PBL, allow the levels stretch gradually apart to maximum separation at mid-levels, then recompress the further aloft where anticyclonic outflow, tropopause folding, and stratospheric potential vorticity/ozone intrusions occur. Cubic and linear vertical stretching options are also available.

4. SUMMARY AND CONCLUSIONS

After five years of R&D and testing involving researchers, teachers, students and operational weather forecasters, LEAD has firmly demonstrated the value of a service-oriented architecture in which meteorological analysis tools, forecast models, and data repositories can operate as dynamically adaptive, on-demand, grid-enabled systems that (a) change configuration rapidly and automatically in response to weather; (b) respond to decision-driven input from users; (c) initiate other processes automatically; and (d) steer remote observing technologies to optimize data collection for the problem at hand. In this endeavor, LEAD has been a driver for new capabilities within the TeraGrid, most notably on-demand computing with high quality of service.

As time progressed and both the LEAD and WRF portal efforts matured, it became clear that merging the capabilities of the two portals would allow both systems to reach their full, transformative potential. This effort is now well underway. In addition to providing unprecedented forecasting capabilities, the new fused portal, with its more advanced and intuitive GUI, will enable users of all levels of experience, education and institutional capability to configure and run advanced numerical weather simulations on powerful computing resources. Students in any

science class will be able to learn meteorology and explore numerical weather analysis and prediction via the web without expert knowledge, while advanced users (graduate students, post docs, research scientists, program managers) will be able to run so-called “deep” simulations – those that can change the underlying thinking that motivated the simulations in the first place – to investigate leading-edge atmospheric research topics.

Acknowledgements: This research was funded by the National Science Foundation under grant ATM-0331578 (University of Illinois at Urbana-Champaign with a sub-contract to the University of North Carolina).

REFERENCES

- Droegemeier, K.K., 2008: Transforming the sensing and numerical prediction of high impact local weather through dynamic adaptation. *Phil. Trans. of the Royal Soc., A*, 1-20.
- Droegemeier and Co-Authors, 2005: Service-oriented environments in research and education for dynamically interacting with mesoscale weather. *Computing in Science and Engineering*, 7, 12-29.
- Govett, M. W. and J. S. Smith, 2008: WRF Portal: A graphical front-end for WRF, *88th AMS Annual Meeting*, New Orleans, La., Amer. Meteor. Soc., 16 pp.
- Kong, F., M. Xue, Kelvin K. Droegemeier, D. Bright, M. C. Coniglio, K. W. Thomas, Y. Wang, D. Weber, J. S. Kain, S. J. Weiss, and J. Du, 2007: Preliminary analysis on the real-time storm-scale ensemble forecasts produced as a part of the NOAA hazardous weather testbed 2007 spring experiment. *22nd Conf. Wea. Anal. Forecasting/18th Conf. Num. Wea. Pred.*, Salt Lake City, Utah, Amer. Meteor. Soc., CDROM 3B.2.
- Smith, J. S., P. McCaslin, and M. W. Govett, 2008: WRF Domain Wizard: The WRF preprocessing system GUI, *88th AMS Annual Meeting*, New Orleans, La., Amer. Meteor. Soc., 21 pp.
- Xue, M., K. K. Droegemeier, V. Wong, A. Shapiro, and K. Brewster, 1995: *ARPS Version 4.0 User's Guide*. Center for Analysis and Prediction of Storms, University of Oklahoma, 380 pp.
- Xue, M., D.-H. Wang, J.-D. Gao, K. Brewster, and K. K. Droegemeier, 2003: The Advanced Regional Prediction System (ARPS), storm-scale numerical weather prediction and data assimilation. *Meteor. Atmos. Physics*, 82, 139-170.

The screenshot displays the XBay workflow composer interface. The main window shows a workflow graph with several components connected by arrows. The components are:

- Terrain_V5_2_9_Pub_Preprocessor** (Orange)
- AssimilatedADASData Config** (Orange)
- CrossCuttingConfigurations Config** (Orange)
- WRF_Static_V5_2_9_Pub_Preprocessor** (Orange)
- NAMLateralBoundaryData Config** (Orange)
- ADAS_Analysis_V5_2_9_Pub_Interpolator** (Green)
- NAM_Lateral_Boundary_V5_2_9_Pub_Interpolator** (Orange)
- ARPS2WRF_V5_2_9_Pub_Interpolator** (Orange)
- WRF_Forecast_V2_2_Pub_Model** (Orange)
- WRF_Output_File Config** (Orange)

The lower-left pane shows the service description for **MetGrid-V3.1-Vortex2**:

Service: MetGrid-V3.1-Vortex2:

Description:
A service to horizontally interpolate the WPS Format meteorological data that are extracted by the ungrb program onto the simulation domains defined by the geogrid program.

Operation: Interpolate

Input parameter(s)

The lower-right pane shows the monitoring table:

| Time | Component | Status | Message |
|-----------------------|---------------------------------|----------------|--|
| 17:36:46.597 05/19/09 | WRF_Static_V5_2_9_Pub_Prepro... | sendingResult | [Trying to send successful result of invocation] |
| 17:37:53.938 05/19/09 | GFacADAS_Analysis_V5_2_9_Pu... | serviceInvoked | Service Invoked |
| 17:38:07.750 05/19/09 | GFacADAS_Analysis_V5_2_9_Pu... | INFO | Service (http://www.extreme.indiana.edu/lead)ADAS_Analysis-V5.2.9-Pub created in https://tyr10.cs.indiana... |
| 17:38:07.762 05/19/09 | GFacADAS_Analysis_V5_2_9_Pu... | sendingResult | [Trying to send successful result of invocation] |
| 17:38:09.494 05/19/09 | ADAS_Analysis_V5_2_9_Pub_Int... | serviceInvoked | Service Invoked |

FIG. 1. XBay workflow composer in the LEAD portal, shown in event monitoring mode for a USDA crop disease forecasting project. A user can select a pre-assembled workflow from a repository or build his/her own by selecting web service-wrapped applications or sources of data from a palette, dragging-and-dropping them into the workflow graph, then “wiring” them together in a desired sequence. The color of the boxes indicates the status of the computational task, with components now executing shown in green, those completed in dark grey, and those waiting to be started in orange. In the case depicted above, an ADAS analysis/assimilation of the meteorological input data is underway to prepare initial conditions for a WRF model simulation (upper right box). A description of each web service is provided in the lower left window while helpful status messages appear in the table below the workflow graph.

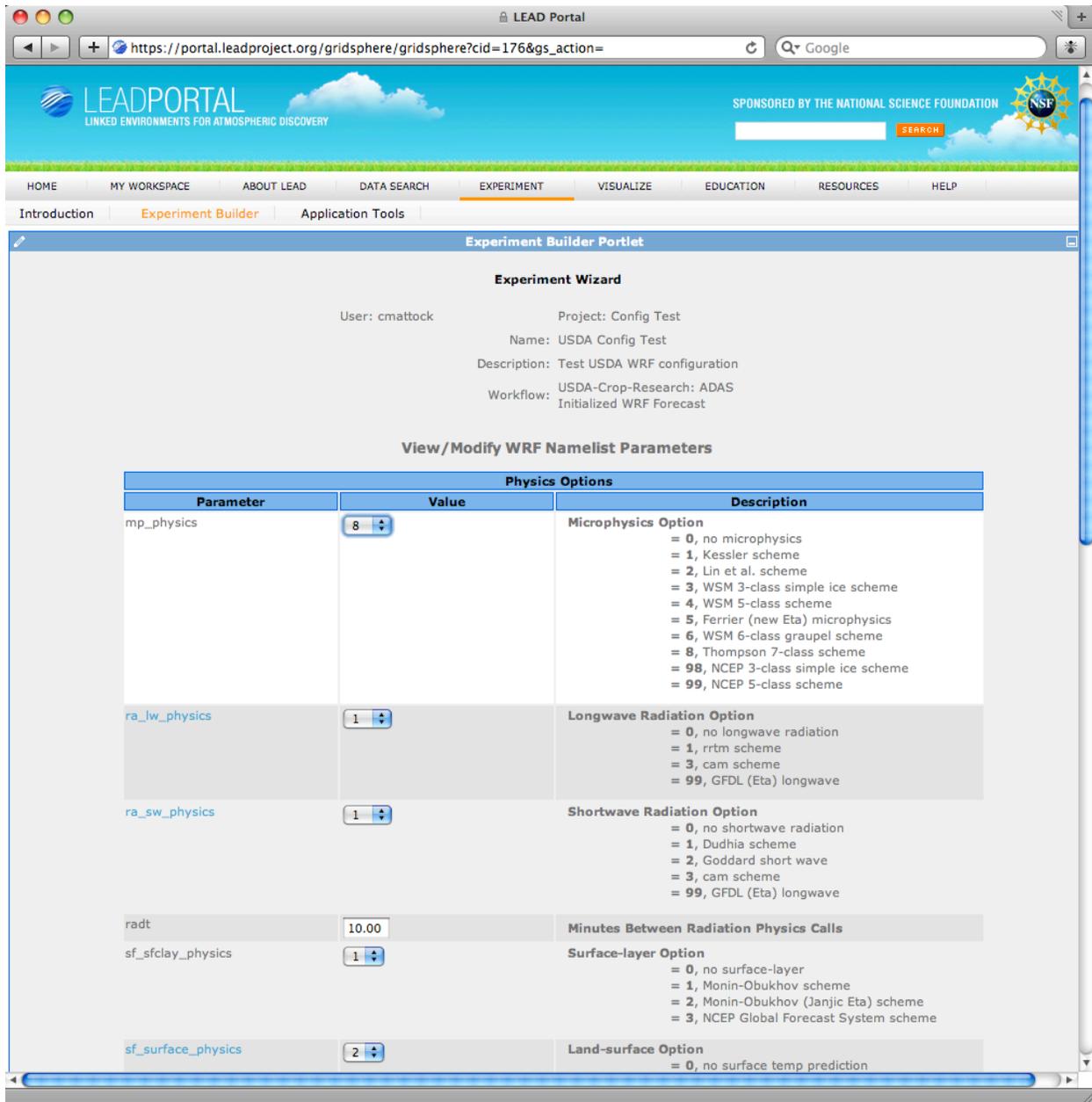


FIG. 2. WRF namelist editor in the Experiment Wizard of the LEAD portal. This tool, which operates on a small subset of physics parameters that are commonly modified in operational forecast ensembles, is ideal for users who prefer a lightweight web interface and only need to make a few simple changes to a namelist.input file. Dynamic highlighting (white color) draws the user's attention to the parameter that is currently being edited.

WRF Domain Wizard for LEAD

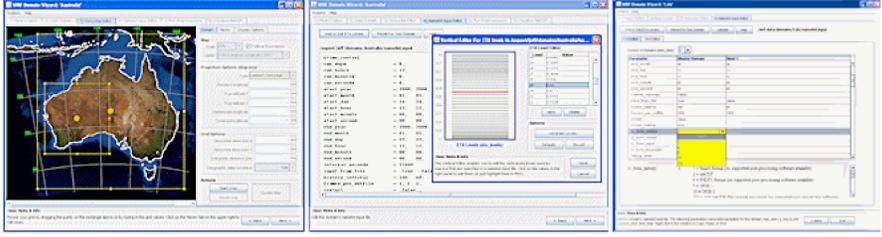
http://wrfportal.org/DomainWizardForLEAD.html

WRF Portal Home | Contact Us

WRF Domain Wizard for LEAD

WRF Domain Wizard for LEAD
 GUI for editing namelist.wps and namelist.input
 Version 1.33 for Linux, Mac, and Windows - released December 4, 2008

Home
 WRF Portal
 Domain Wizard
 Tutorials (HTML)
 Tutorials (Video)
 F.A.Q.
 About



WRF Domain Wizard is a graphical user interface (GUI) for editing namelist.wps and namelist.input. It enables users to easily define and localize domains (cases) by selecting a region of the Earth and choosing a map projection. Users can also define nests using the nests editor, and edit the file, namelist.input.

Source Code is available [here](#).
 What's [new in version 1.33](#)

Run WRF Domain Wizard (for LEAD) using Java Web Start

[Click here to launch WRF Domain Wizard version 1.33](#)
 Requires Java 5 or later (go [here](#) if you can only run Java 1.4).

Help! WRF Domain Wizard doesn't launch! Here are some [trouble-shooting tips](#).

What is Java Web Start?

Java Web Start is an application-deployment technology that automatically downloads your software then launches (runs) the application. You do not need to set up directories, run installation programs, or configure anything. Just click the link and the program runs. The first time you click on the link above, there is a delay while the software downloads. The next time you click the link, Web Start checks to see if an update is available. If one is, then it downloads the update. Otherwise it runs the local copy (avoiding the delay of downloading it again).

NOAA
 Disclaimer
 Privacy Policy
 NOAA website
 ESRL website
 FSL website
 Accessibility statement

Hyperlink to JNLP file

FIG. 3. New website at NOAA/ESRL/GSD created to serve and support the LEAD version of the WRF Domain Wizard.

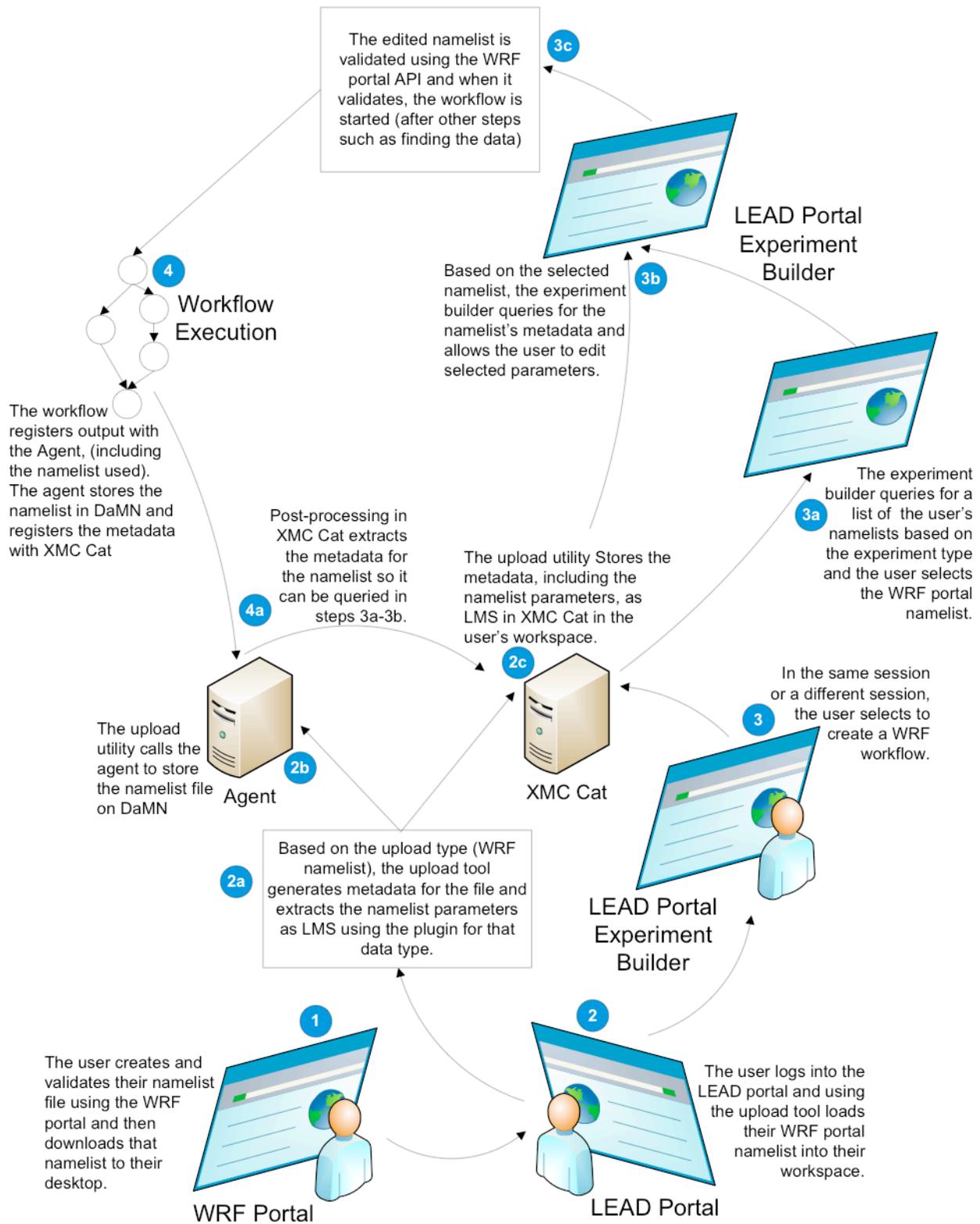


FIG. 4. Architectural changes required to implement the WRF Domain Wizard in the LEAD Experiment Builder (figure courtesy of computer scientist Scott Jensen, Indiana University).

TestNG: Unit Test

file:///Users/cmattcock/Desktop/test_validate/test-output/emailable-report.h

| Test | Methods Passed | Scenarios Passed | # skipped | # failed | Total Time | Included Groups | Excluded Groups |
|---------------------|----------------|------------------|-----------|----------|-------------|--------------------------------------|-----------------|
| Namelist Validation | 17 | 17 | 0 | 1 | 0.0 seconds | tests.string tests.date tests.number | |

| Class | Method | # of Scenarios | Time (Msecs) |
|------------------------------|---|----------------|--------------|
| Namelist Validation — failed | | | |
| tests.StringTest | testTrimAll (tests.string) | 1 | 2 |
| Namelist Validation — passed | | | |
| tests.DateTest | testIsValidYear (tests.date) | 1 | 1 |
| | testIsValidDate (tests.date) | 1 | 0 |
| | testIsValidDay (tests.date) | 1 | 17 |
| | testIsValidMonth (tests.date) | 1 | 1 |
| | testIsValidSimulationTime (tests.date) | 1 | 0 |
| | testIsValidTime (tests.date) | 1 | 0 |
| tests.NumberTest | testIsEven (tests.number) | 1 | 6 |
| | testIsInteger (tests.number) | 1 | 1 |
| | testIsNumber (tests.number) | 1 | 0 |
| | testIsOdd (tests.number) | 1 | 1 |
| | testIsWholeNumber (tests.number) | 1 | 1 |
| | testIsWithinCorrectRange (tests.number) | 1 | 0 |
| | testNotEqual (tests.number) | 1 | 1 |
| tests.StringTest | testIsValidLength (tests.string) | 1 | 0 |
| | testIsEmpty (tests.string) | 1 | 0 |
| | testIsNotEmpty (tests.string) | 1 | 0 |
| | testIsValidText (tests.string) | 1 | 0 |

Namelist Validation

tests.StringTest:testTrimAll

```

expected:<true> but was:<false>

org.testng.Assert.fail(Assert.java:84)
at org.testng.Assert.failNotEquals(Assert.java:438)
at org.testng.Assert.assertTrue(Assert.java:32)
at org.testng.Assert.assertTrue(Assert.java:42)
at tests.StringTest.testTrimAll(StringTest.java:33)
23 lines not shown

```

FIG. 5. Test report (in HTML format) for the low-level namelist validation functions, automatically generated by the TestNG unit testing framework. Unit tests that passed (failed) are indicated in green (red) color.

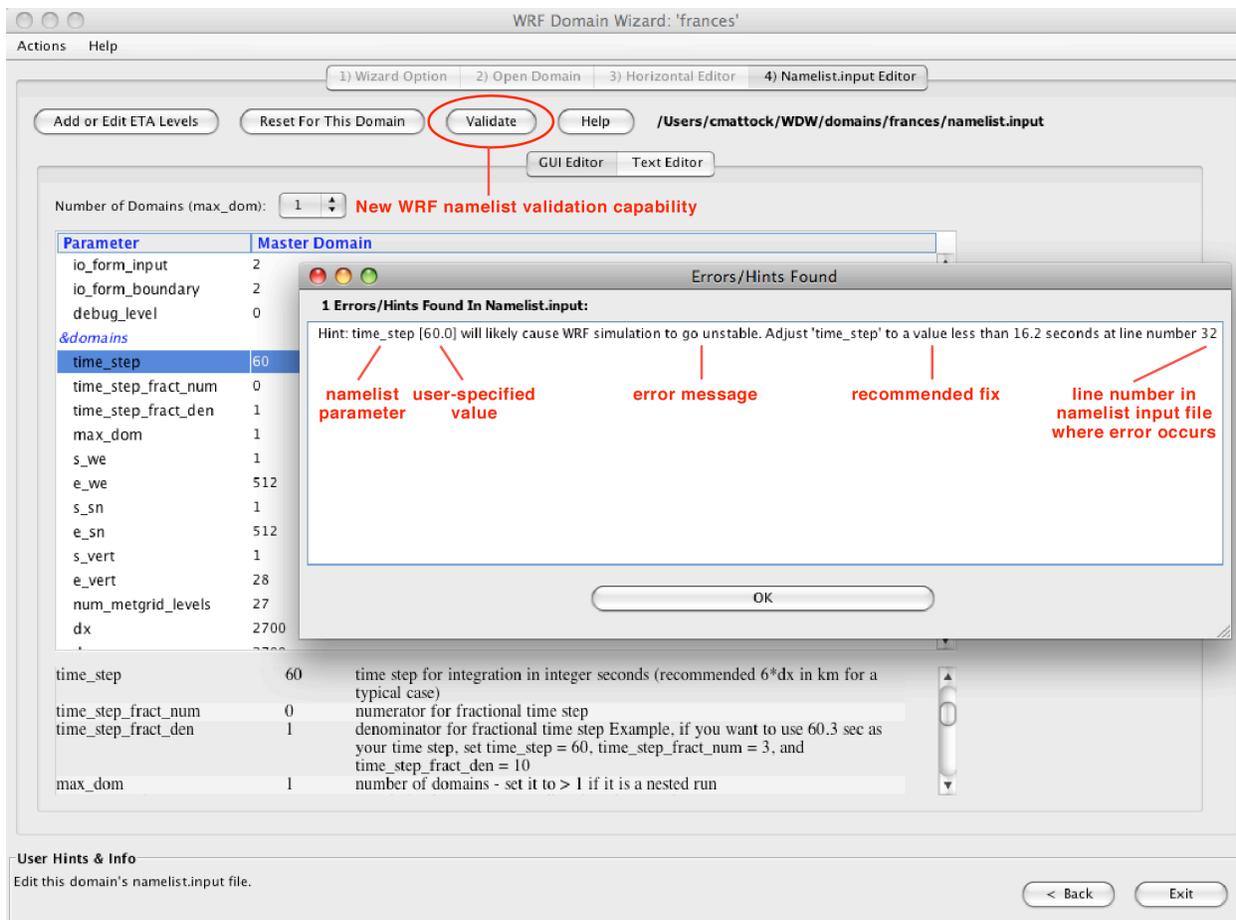


FIG. 6. Namelist Editor in the LEAD version of the WRF Domain Wizard, running independently of both WPS and an SSH connection to a remote supercomputing cluster. Note the new “Validate” button in the user interface. In the depiction above, the numerical stability checker has detected a user error in the specification of the time step and has suggested a new value that will prevent the WRF model simulation from going unstable.

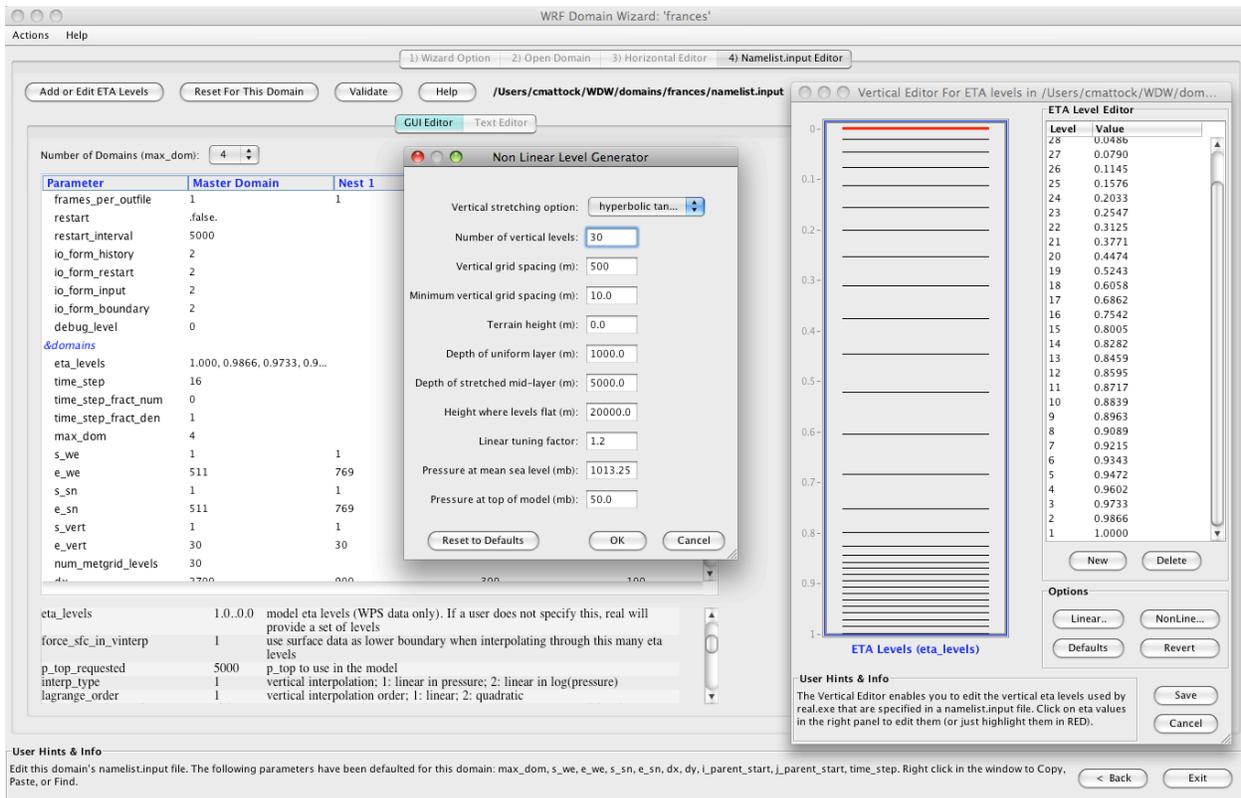


FIG. 7. The University of Oklahoma-CAPS ARPS model vertical grid stretching algorithm, adapted for use with the WRF NWP model, running in the WRF Domain Wizard to configure the σ_p (eta) levels in a quadruply nested WRF simulation of Hurricane Frances (2004). The user has selected the hyperbolic tangential stretching option in the Non-Linear Level Generator (left) to pack multiple uniformly spaced layers in the planetary boundary layer. Editable levels and their values are displayed in the Vertical Editor for ETA Levels (right). Cubic and linear vertical stretching options are also available in the pulldown menu.

Table 1. AFWA JME Member Configurations<http://www.mmm.ucar.edu/wrf/users/workshops/WS2008/abstracts/7-01.pdf>

| Member | Surface | Microphysics | PBL | Cumulus | Longwave | Shortwave |
|--------|---------|--------------|-----|---------|----------|-----------|
| 1 | Thermal | WSM3 | MRF | Grell | CAM | Dudhia |
| 2 | Thermal | Ferrier | YSU | Grell | CAM | CAM |
| 3 | Thermal | WSM6 | MYJ | KF | RRTM | CAM |
| 4 | Noah | Lin | MRF | KF | RRTM | CAM |
| 5 | Noah | WSM5 | YSU | KF | RRTM | Dudhia |
| 6 | Noah | WSM5 | MYJ | Grell | RRTM | Dudhia |
| 7 | RUC | Lin | YSU | BM | CAM | Dudhia |
| 8 | RUC | Ferrier | MYJ | KF | RRTM | Dudhia |
| 9 | RUC | Ferrier | YSU | BM | RRTM | CAM |
| 10 | RUC | WSM6 | YSU | Grell | CAM | CAM |

Table 2. LEAD/CAPS 2008 Spring Real-time Storm Scale Forecast Ensemble<http://www.mmm.ucar.edu/wrf/users/workshops/WS2008/presentations/7-3.pdf>

| Member | Surface | Microphysics | PBL | Cumulus | Longwave | Shortwave |
|----------|---------|--------------|-----|---------|----------|-----------|
| 1 (Cntl) | Noah | Thompson | MYJ | none | RRTM | Goddard |
| 2 (C0) | Noah | Thompson | MYJ | none | RRTM | Goddard |
| 3 (n1) | Noah | Ferrier | YSU | none | RRTM | Goddard |
| 4 (p1) | Noah | WSM6 | MYJ | none | RRTM | Dudhia |
| 5 (n2) | Noah | Thompson | MYJ | none | RRTM | Goddard |
| 6 (p2) | Noah | WSM6 | YSU | none | RRTM | Dudhia |
| 7 (n3) | Noah | Thompson | YSU | none | RRTM | Dudhia |
| 8 (p3) | Noah | Ferrier | MYJ | none | RRTM | Dudhia |
| 9 (n4) | Noah | WSM6 | MYJ | none | RRTM | Goddard |
| 10 (p4) | Noah | Thompson | YSU | none | RRTM | Goddard |