NITE - Numerical Weather Prediction Information Technology Environment

Final Report by the Developmental Testbed Center - 3/31/2015

Ligia Bernardet
NOAA /ESRL Global Systems Division, Boulder, CO
CU Cooperative Institute for Research in the Environmental Sciences, Boulder, CO
ligia.bernardet@noaa.gov

Laurie Carson
National Center for Atmospheric Research, Boulder, CO
carson@ucar.edu

1. Executive summary

In response to recommendations from the DTC Science Advisory Board and the UCACN, and discussions with NCEP leadership, the DTC undertook the task of creating a design for an infrastructure to facilitate development of NCEP numerical models by scientists both within and outside of EMC. Requirements for this NWP Information Technology Environment, or NITE, are based on a survey of potential NITE users, information obtained during site visits to EMC, UKMO, and ECMWF, discussions with focus groups, and reviews of various existing model development systems. The NITE design put forth to address these requirements includes the following elements:

- **Data management and experiment database.** The data management element provides scientists with access to input datasets (model and observations), a mechanism for storing selected output from all experiments and tools for browsing, interrogating, subsetting, and easily retrieving data. An important aspect of this element is establishing standards for storing model and observation datasets, as well as their metadata. Another important aspect of the data management element is capturing all metadata pertaining to an experiment. To facilitate sharing of information, NITE needs to record information on key aspects of the experiment setup, such as provenance of source code and scripts, configuration files, and namelist parameters, in a searchable database.
- **Source code management and build systems.** Single SVN code repositories or distributed Git repositories need to be available for all workflow components run within NITE. This will support code unification and collaboration between developers. Code repositories need to be available outside of the NCEP firewall, where the community can access them. Fast, parallel build systems should be implemented to efficiently build all workflow components of a suite before experiments are conducted.
- **Suite definition and configuration tools.** All configurable aspects of a suite are abstracted to files that can be edited to create the experiments. No aspects of the directory structure, batch system, namelist parameters etc. are hardcoded in the scripts or source code. Predefined suites are provided as a starting point for creating experiments, with scientists also having the option to compose their own suites.
- **Scripts.** The scripting for NITE is such that each workflow component within NITE (e.g., GSI) is associated with a single script, regardless of which suite is being run (e.g., NAM or RAP). Standardization of scripts reduces the overall maintenance costs for NCEP's multiple suites and helps scientists familiar with one suite quickly learn another.
- **Workflow management system.** The workflow management system handles all job submission activity. Hence, the scripts used to run workflow components within NITE do not contain job submission commands. To meet long-term plans for NITE, it will be important for this workflow management system to be available for use outside of WCOSS.
- **Documentation and training.** Documentation and training on all workflow components and suites available through NITE, as well as on NITE itself, are readily available through electronic means.

In addition to the elements above, standardized tools for data visualization and forecast verification need to be available to all scientists as part of NITE.

We recognize that substantial resources will be required for initial and ongoing development of NITE. However, it should be pointed out that NOAA already has several tools that can be used as a starting point for various NITE elements (e.g., MADIS, NOMADS, Rocoto, HWRF object-oriented scripts, VLab, and WRF Portal). While the initial deployment of NITE will cause some disruption to model development, we are confident that this infrastructure will facilitate developing and running NCEP suites and will make transition of new research and development to operations more efficient and effective.

2. Introduction

NCEP employs a number of numerical weather prediction models to provide operational guidance to its service centers and to the NWS field offices. The GSM is used for the global applications and three other models are used for limited area applications: WRF-NMM, WRF-ARW, and NMMB. Here we consider WRF-ARW and WRF-NMM as two distinct models since they use different dynamic cores and physics interfaces, meaning that not all physics parameterizations available in the WRF package are available to both WRF-ARW and WRF-NMM. Note that while ARW and NMM employ the WRF framework (which defines the I/O, timing, etc.), GSM and NMMB employ the completely distinct NEMS framework. In addition to the number of models and frameworks used, NCEP runs a large number of modeling suites, defined here as NWP systems with multiple workflow components, assembled for specific applications (Fig. 1). Global suites include the CFS, GFS, and GEFS, while limited area suites include NAM, HRW, RAP, HRRR, GFDL Hurricane model, HWRF, and SREF. Suites have various workflow components besides the atmospheric forecast models, for instance, pre-processing of analyses and forecasts from parent models, observation pre-processing, data assimilation, coupled workflow components (such as oceanic and hydrologic models), postprocessing, and product generation tools. In addition, each suite depends on a myriad of operations related to retrieving and staging input files, creating output directories, archiving output, purging disks, etc. This complex set of NWP tools needs to be maintained and improved over the years so that daring NOAA goals in raising forecast accuracy can be achieved. The topic of this report is a recommendation for the creation of a robust NWP Information Technology Environment (NITE) at NCEP to support and accelerate the development and improvement of modeling suites.

The research leading to this report was conducted by the DTC, an organization whose mission is to establish a bridge between the research and operational communities working in NWP (Bernardet et al. 2008 and Bernardet et al. 2014). Over the years, the DTC has put in place several mechanisms to facilitate the use of operational models by the general community, mostly by supporting operational codes (WRF-ARW, WRF-NMM, HWRF, GSI, UPP, among others) and organizing workshops and tutorials. By stimulating the use of operational codes by the research community, composed of universities, NCAR, and government laboratories, several new NWP developments have been transitioned to NCEP operations. However, in spite of the relative success of the DTC, there are still significant gaps in the collaboration between the research and operational groups.

The DTC SAB has documented that the research community finds it difficult to use and run operational suites, and that this is a deterrent to conducting research and development using the NCEP systems. Instead, researchers tend to use the friendliest NWP systems available, which are also the ones with the largest community of users where peer-to-peer support is widely available. The DTC SAB is not alone in identifying that the collaboration between the academic community and NCEP is sub-optimal. Similar conclusions have been reached by the UCACN, which explicitly recommended that interactions with the community should be enhanced, and that NOAA needs to devote resources to the creation of a modeling infrastructure to facilitate the use of operational suites by the research community (UCACN 2014).

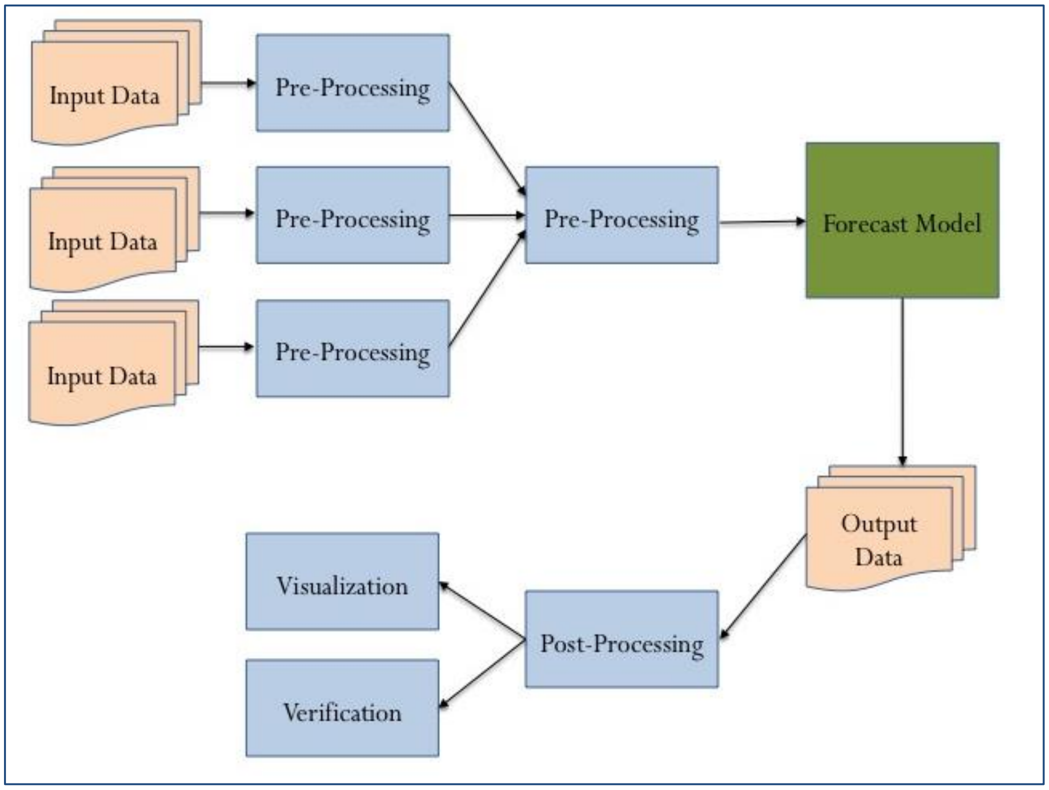

Figure 1. Example of a suite containing various workflow components.

It is important to note that while many think of NCEP as the operational entity in NOAA NWP, NCEP actually has a very broad mission. NCO runs the NWP suites in production mode, while EMC is responsible for model development and improvement. EMC centralizes the development of certain models and suites (for example, the NMMB and the NAM suite) while other models and suites are mostly developed by collaborators and transitioned to EMC (e.g., WRF-ARW and RAP). Since EMC engages in model development, scientific discovery, and model testing ranging from case studies to multi-year pre-implementation testing, EMC is an important customer for NITE. In fact, funding for the NITE design was only secured upon agreement that the system would be targeted for model development both within EMC and by collaborators.

The approach taken to reach the NITE design engaged large segments of the national and international NWP community and is described in Section 2. Section 3 describes the proposed design for NITE, while Section 4 has conclusions and possible next steps for making NITE a reality at NCEP. Note that all acronyms are defined in Appendix 1.

3. Process and survey results

Two basic approaches were used to reach the NITE design: a survey of the potential NITE user community to understand the main problems in running experiments with NCEP operational systems, and an assessment of selected existing NWP infrastructure systems, which included review of available documentation, site visits, and focus groups with NWP teams.

3.1 Focus groups and the NITE survey

The NITE survey was conducted electronically. It was created by the DTC in collaboration with EMC, and distributed to over 100 potential NITE users. Forty responses were obtained: 19 from EMC staff, 11 from NOAA research laboratories (ESRL and AOML), 9 from NCAR, 12 from universities, and 2 from ARL (Fig. 2). All of the responders already had some experience running NCEP operational suites, mostly for research and testing purposes. In addition to the NITE survey, focus groups about NITE were conducted at EMC and NOAA ESRL.

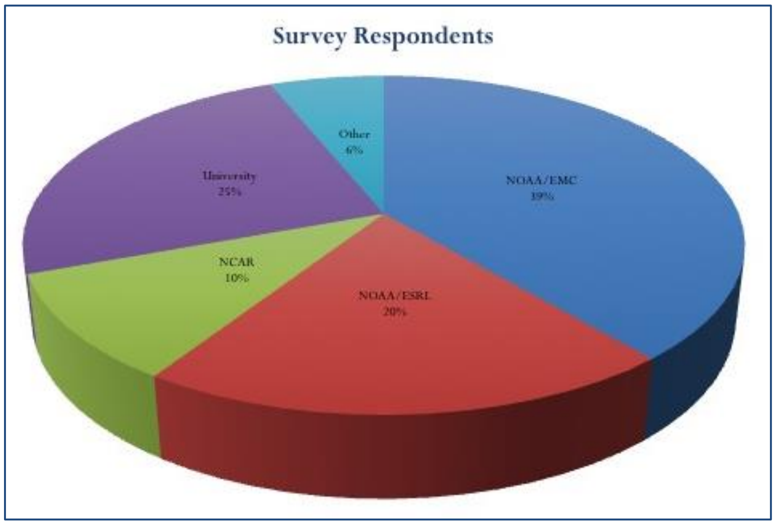

Figure 2. Partition of the NITE survey respondents among various institutions: NOAA ESRL (red), NOAA EMC (dark blue), universities (purple), NCAR (green), and other (light blue).

The issues below, listed in order of significance, were identified by the survey and focus groups participants as obstacles in running NCEP models.

- Lack of a friendly way to configure the modeling suite. The configurations are buried in various namelists, configuration files, and scripts, with too many hardcoded aspects.
- Difficult access to input datasets, especially to observations. This is particularly critical for model users without access to the NOAA computational platforms.
- Insufficient source code management, including limited access to code used in operations and to code used by other researchers and developers.
- Absence of a systematic way to keep track of the provenance of the source codes, configurations, and inputs used in an experiment conducted by a researcher or his/her peers. This makes it difficult to reproduce experiments and prevents EMC's staff from understanding how an experiment was conducted, and, therefore, from accepting its results as a demonstration of new development.
- Absence of documentation and training opportunities on the scientific aspects of the modeling suite, as well as on practical aspects of how to run it.
- Difficult access to automation tools, making it cumbersome to run experiments and limiting one's ability to run a large number of cases, especially when experiments involve cycled data assimilation procedures.
- Lack of standardized data visualization and forecast verification workflow components to be used by the group.

The overwhelming majority of participants emphasized that NITE would be helpful to address the issues above, and stated that NITE could be used throughout the life cycle of development, from the initial formulation of an idea, all the way to pre-implementation testing. There were a few areas in which less consensus was found among the participants. One of them was the number of versions of a modeling suite that should be supported by NITE. While EMC respondents placed larger emphasis on the support for the latest version of the codes, stating there is no point in conducting development on a model version that has been phased out, academic respondents saw the value of supporting older versions so that research projects and papers can be finalized. A lack of consensus was also noted in the type of datasets that need to be made available. The answers varied from a few case studies all the way to massive retrospective datasets and realtime data feeds, clearly reflecting the variety of potential uses for NITE.

The questions also addressed issues in computational platform. In additional to the operational platform, EMC developers have access to several NOAA high-performance computing platforms, all of which connect with the NOAA high-performance storage system. On the other hand, NCAR and university researchers often do not have access to NOAA computational platforms and have, therefore, limited or no access to the input datasets and tools available to EMC staff. The results indicate that NITE should be usable in generic platforms, but that the top priority is the *Zeus* (or *Theia* or subsequent replacement) NOAA research platform. Until NITE is fully available in other platforms, outside collaborators may need to be granted access to the NOAA high-performance computing systems.

3.2 Review of NWP infrastructure systems

The research for the NITE design involved the review of several existing NWP infrastructure systems. While far from comprehensive, the review covered various operational, research, and cross-cutting NWP systems, both in the US and internationally. A careful comparison of attributes of the infrastructure systems was a valuable process to gain in-depth understanding of each system. For this review the authors performed site visits to EMC, ECMWF, and the UKMO. A comprehensive overview of these systems is outside the scope of this report, and, therefore, only a brief description of each is provided below.

- **NCEP NAM infrastructure.** This suite, which employs the NMMB model, currently has few users outside of NCEP. The first Community NMMB tutorial is planned for April 2015. Some of the NAM development is done using the NEMS launcher, a framework that facilitates configuring and running cold-start experiments using the NMMB model. In operations, the NAM is run using the ECMWF ecFlow workflow management system, while in research mode the NAM is run using cascading scripts that submit the jobs.
- **NCEP NAMRR infrastructure**. This suite is under development at EMC for possible replacement of the NAM suite in 2016. A one-hour cycled data assimilation strategy is employed, and the Rocoto workflow management system is used to drive the ksh scripts.
- **NCEP HWRF infrastructure.** This suite, which employs the WRF-NMM coupled with an ocean model, has a history of community support through the DTC, has a mature code management system, and employs a set of object-oriented Python scripts that allow the system to be run with multiple workflow management systems (Trahan et al. 2015).

- **ECMWF IFS infrastructure.** The GUI PrepIFS is used to configure, launch, and record experiments with the IFS. Code management is done with perForce and FCM is used to streamline the build system. Access to the HPSS for storage of outputs and retrieval of inputs and output datasets is facilitated with MARS. Both the SMS and ecFlow workflow management systems are employed.
- **UKMO UM infrastructure.** A single atmospheric model, the UM, is used for both global and limited area model suites. The Cylc workflow management system is used in research and operations, and a set of Python-based tools named Rose are used to configure the suites. Robust experiment database, issue tracking, and software management tools are employed.
- **NCAR CESM infrastructure.** Various workflow components, the atmosphere, land, ocean, and ice models, are coupled to compose this community climate model. Because of its prominent role in generating information for the IPCC report, recording provenance of model runs is of paramount importance. The CESM has a powerful experiment database for recording and retrieving the experiment configurations.
- **WRF Portal.** This tool, developed by NOAA ESRL, can be used to configure and launch WRF and WPS runs. The WRF Portal is not associated with any particular suite, and it does not support data assimilation or cycling. It includes a database of experiment metadata, but it does not track the provenance of source code.

4 NITE design

NITE is an infrastructure aimed at facilitating and enhancing the development of NCEP operational models. It has been designed to make the work of EMC staff and their collaborators more effective. NITE was not designed to work exclusively for a certain model or suite; instead it should transcend the current NWP suites and be applicable to the expected evolution of NCEP systems. Due to its large scope and complexity, it is likely that NITE has to be built incrementally, and therefore a few gradual NITE implementation scenarios are described at the end of this section.

It is not anticipated that NITE will have its own HPC platform. Instead, NITE must work on the platforms currently available to the target users. It is also important that NITE works on the NCEP operational platform, so that the codes for operational implementation are obtained directly from NITE, and all emergency fixes implemented by NCO are readily available through NITE. The top priority platforms for NITE's initial deployment are the NOAA supercomputers WCOSS (operational) and *Zeus* (research). Note that while we refer to *Zeus* in this report, as time goes on and *Zeus* is phased out, this should refer to the new equivalent research platform (*Theia* and beyond*)*.

For those working in the NOAA platforms, access to code repositories and datasets will be more straightforward. However, granting access to NOAA platforms to all scientists that want to use NITE may be costly and infeasible. Therefore, it is ideal for NITE to be available in platforms outside the NOAA firewall as well. For that reason, in later deployments, NITE should be extended to additional NOAA research platforms, the NCAR supercomputer, and, finally, to any generic platform (e.g., university Linux clusters).

4.1 Design goals

The overarching goals for NITE are summarized in the list below.

- A single model development infrastructure is available both to EMC staff and their external collaborators.
- Experiments can be conducted on a variety of computational platforms (within and outside of NOAA), with multiple operating systems and schedulers.
- The development and testing conducted within the NITE framework is relevant for operational consideration and results can be seamlessly transitioned to NCO.
- Scientists can easily create their own suites by assembling workflow components in the desirable configuration. Additionally, predefined suites are available to scientists as a starting point for conducting research experiments.
- Metadata of past experiments can be browsed, making it possible to reproduce previous runs. This includes the ability to reproduce operational, pre-implementation, and research runs.
- Datasets needed as inputs to NWP suites, such as observations, analyses, and forecasts, are readily available. Output from numerical experiments can easily be stored and retrieved. All datasets use documented standards so new workflow components can be readily plugged in.
- Standardized tools for basic data visualization and forecast verification are available and can be added to any suite.
- The infrastructure is easy to use and well documented.
- The infrastructure does not pertain to a specific NWP model or suite. Instead, it is general and can be applied to NCEP's various suites, including those that are under development for future implementations.
- The infrastructure is modular so that it can be incrementally implemented.

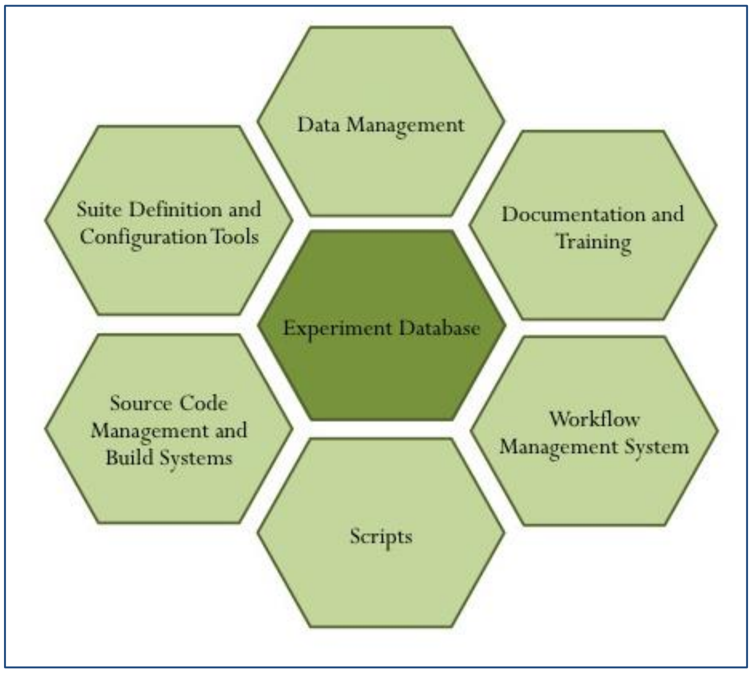
Data Management
Suite Definition and Configuration Tools
Documentation and Training
Experiment Database
Source Code Management and Build Systems
Workflow Management System
Scripts

Figure 3. NITE seven elements: data management, source code management and build systems, suite definition and configuration tools, scripts, workflow management system, experiment database, and documentation and training.

4.2 NITE elements

In order to fully support model use and development, NITE will have the following elements, also described in Fig. 3: data management, source code management and build systems, suite definition and configuration tools, scripts, workflow management system, experiment database, and documentation and training.

4.2.1 Data Management

Typical NWP experiments require a variety of input datasets and produce numerous output files. With the increase in computer power in the last few years, and the consequent ability to run models in higher resolution and/or ensemble mode, the volume of I/O data is ballooning and can only be expected to keep growing. The NITE data management element contains two basic aspects: storage itself and tools for efficient archival and retrieval of information. In the NITE concept there is no difference between the storage of inputs and outputs. For example, outputs from GFS operational runs can be inputs to limited area model runs.

4.2.1.1 Analysis and forecasts for input

The top priority is to provide scientists with easy access to the analyses and forecasts from the operational GFS for use as initial and lateral boundary conditions for their experiments. As a second priority, analyses and forecasts from the operational GEFS (to initialize LAM ensemble systems) should be provided.

Analyses and forecasts from GFS reforecasts would also be useful to initialize LAM reforecasts. While some development teams do not use GFS reforecasts, other teams rely extensively on them. For instance, the EMC Hurricane Team typically tests HWRF with GFS reforecasts to provide input to GFS operational implementations.

4.2.1.2 Observations

For most applications, access to global quality controlled observations, such as those contained in the prepBUFR files associated with the GFS operational runs, is sufficient. However, certain types of research, such as data assimilation and observation system experiments, may require additional inputs and tools. The 2014 report of the DTC Science Advisory Board emphasizes the need for tools that allow researchers to reproduce the QC process used at NCEP to create the prepBUFR files, and to make changes, and potential improvements, to this process.

While this cannot be considered for the first implementation, later implementations of NITE could include a system similar to the ODB used at ECMWF and UKMO. ODB converts observational datasets, which in their raw form are in various standards and file formats, to standardized data schema. For example, one dataset may have a *longitude* variable, while another has *lon.* Likewise, one dataset may have longitude ranging from 0 to 360, and another from -180 to +180. Standardization and publication of such standards is of paramount importance to facilitate use and make sure all workflow components can ingest the datasets. CF conventions should be considered for all datasets.

To avoid loss of information, it is preferable for all observations to be stored in their raw form and converted to the proper standards at retrieval time. By providing scientists with standardized files, it can be guaranteed that the observations can be used seamlessly by various workflow components (plotting and data assimilation, among others).

One important aspect of a well-constructed observation database is that it optimizes observation retrieval for high-performance computing platforms, which is important because a substantial portion of the data assimilation process is spent on data retrieval.

The NOAA MADIS provides some of the capabilities described in this section. MADIS ingests files from NOAA data sources and non-NOAA data providers, decodes the data and then encodes all of the observations into a common format with uniform observation units and time stamps. Quality control checks are conducted and the integrated data sets are stored in the MADIS database with a series of flags indicating the quality of the observation from a variety of perspectives (e.g., temporal consistency and spatial consistency), or more precisely, a series of flags indicating the results of various QC checks.

4.2.1.3 Timeline of datasets

Regarding the amount and timeline of data, the top priority is to have four months of retrospective data, one for each season, in order to provide enough diversity of meteorological scenarios for experiments. Additionally, datasets to initialize challenging meteorological cases (forecast *dropouts*) are a top interest.

In time, it is recommended that the retrospective datasets be extended to an entire year, and later to multiple years, to allow testing with a larger sample size. It is also important to start realtime data feeds, as NOAA has a growing practice of conducting realtime research.

Note that for certain weather phenomena, such as tropical cyclones, data needs to be concentrated on specific seasons and cover multiple years so statistical significance of the results can be achieved.

4.2.1.4 Accessibility of datasets

The NOAA Environmental Security Computing Center, contains a HPSS in support of NOAA's research and development. This HPSS contains many of the input datasets needed for NITE. However, the NOAA HPSS is not accessible by those without NOAA accounts. Given that the process to provide scientists with NOAA accounts is costly and time consuming, as it requires a background check, it may be more practical to have the higher-priority datasets available outside of the NOAA firewall.

For the first instantiation of NITE, data could be provided only in the NOAA HPSS. Later, the most important datasets could be staged in selected computational platforms, for example, NCAR’s HPSS. The ultimate goal would be to have all NITE datasets available through a data service and usable by any community member.

The NOAA NCDC houses NOMADS, which provides access to both realtime and retrospective model datasets. The need for data dissemination can be fulfilled by serving additional datasets through NOMADS or another system based on standard protocols for data exchanges, such as OPeNDAP.

Scientists also need storage space for the output of their experiments, preferably in a location accessible to all to facilitate collaboration. As with input datasets, the first step is to have archival of outputs in the NOAA HPSS. The second step is to archive experiments on locations accessible to the entire community, so that peers can share results.

As an example, most of the UKMO's archives are on-site and behind the firewall. However, climate scientist use an additional off-site 7 PB super-data-cluster (JASMIN) to store and share outputs from their experiments. The concept of federated datasets could be considered for NITE in order to alleviate the requirement of having all the data housed in a single place.

4.2.1.5 Data formats for storage
In addition to the access to storage for experiment inputs and outputs, some consideration must be given to data formats. For NITE to work well, formats need to be standardized so the data visualization and forecast verification tools can read outputs from all models. At ECMWF HPSS storage is limited to files in BUFR and GRIB format, with secondary on-disk storage for miscellaneous files. Similarly, UKMO stores its files in BUFR and PP format (PP is the UKMO proprietary data format), with a foreseen transition to BUFR and NetCDF format in the future. In this sense, storage of other formats, such as native binary output, is discouraged.

4.2.1.6 Storage of native versus postprocessed information
Another consideration regarding storage of model results refers to saving native model output versus postprocessed files. Since there is loss of information through postprocessing, it is recommended that native variables on native model levels be stored. Since native datasets are potentially larger than derived datasets, it is important that the archive resources be commensurate with the needs.

4.2.1.7 Metadata for stored datasets
Scientists must be able to quickly browse the archives using a variety of search parameters, a capability that requires the existence of substantial metadata about the files in the archive. Categories of metadata include, but are not limited to: date, operational or experimental suite used to generate it, person/institution that generated the data, areal coverage, project keywords and description. A GUI or web page is recommended for browsing the archive contents.

4.2.1.8 Retrieval of information
Scientists need to be able to easily retrieve datasets, in their entirety or subsets. Possible subsets are based on data, areal coverage, vertical levels, and meteorological variables. For example, the ECMWF MARS system allows retrieval based on the following query: *500-hPa temperature for the 3-day IFS forecasts over Europe from January 1 through February 28, 2010.*

In addition, it is important to allow retrieval of postprocessed information to reduce the need to transfer large datasets. Scientists should be able to obtain plots of the raw data as well as be able to perform basic operations on the data and retrieve the results. For example, one could retrieve only temperature means instead of entire temperature fields, or a single vorticity field instead of two fields for wind components.

4.2.2 Source code management and build systems
Source code for NITE itself should be placed in a code management server, such as SVN or Git, and its development should follow best practices in software management. However, the topic of this section is not source code for NITE itself but for the workflow components of NWP suites supported by NITE.

Scientists should have access to code management tools to obtain the source code for the workflow components and keep track of their development. A primary goal is that all development should have a path toward potential operational implementation. Therefore, scientists should be strongly encouraged to place all code changes in developmental branches of the code repository. This is the easiest way to keep track and recover code, and is a prerequisite to prevent code from aging off or scientist code diverging from the main development. Use of a code repository for development is particularly important to allow frequent updates of experimental code to make sure scientists are using the most relevant code for pre-implementation testing. While this ability may not be necessary for university scientists, it is indispensable for EMC staff working on rapid development for implementation.

Placement of developmental code in a branch should be encouraged but not mandatory. For small code development exercises (a few lines of code changed), scientists should be given the option of simply recording the code changes in the experiment database.

Since suites are composed of various workflow components with large code bases, NITE should incorporate powerful tools for expediting parallel builds, keeping careful track of dependencies between codes, using tools such as the *Flexible Configuration Management system* developed by the UKMO. Build systems should be flexible in their choice of compiler and compiler options, and allow for plenty of debugging opportunities, including support for using interactive debuggers such as TotalView. The NEMS AppBuilder is an example of NOAA tool that is already in place for facilitating building NEMS-based applications, and which should be considered foruse in NITE.

Code integration with code repositories, such as SVN, is a lot easier if all development is conducted in a single repository. The duplication of SVN code repositories, and the resulting need for mirroring and synchronization increase cost and vulnerability to error. Therefore, we recommend a single code repository for each component, located in a place accessible to all NITE users. It is not necessary that all code used in a suite be housed in the same code repository, or even by the same institution, as NITE can easily pull code from various places. We recognize that this recommendation poses some security questions. If code repositories are located within the NCEP firewall, access will have to be granted to outsiders. One possible solution is for NCEP to place its codes outside its firewall and create a frequent backup inside the firewall for added security. This solution goes a long way, but does not address the issue of community codes, or codes whose main development is housed somewhere in the community. One important example is the WRF model, whose code repository is located at NCAR. For these cases, NCEP and NITE users should use the established community repositories, again making a backup inside the NCEP firewall.

An alternative is the use of multiple connected repositories using Git (being employed in NOAA's VLab), which would allow distributed development in various repositories. Of course this would involve a transition of EMC/NCO's codes from SVN to Git.

It is recommended that all code repositories use issue tracking tools such as Trac, which is already used by NCEP and also employed at UKMO. Issue tracking is a tool that enhances collaboration as it allows scientists to record problems and bugs encountered in the code, as well as the solutions and decisions made about such issues. In addition, tools such as Trac Roadmap provide a planning framework for upcoming implementation and releases, allowing record of what needs to be done, by whom, and what has been completed with what outcome. Another very useful tool for expediting and enhancing development is a code browser, which should be made available as part of NITE in order to facilitate code inspection and development. The code browser would be best considered in the context of an IDE. For example, the NEMS group is using the Cupid custom plug-in for the widely used Eclipse IDE. If the chosen solution were to keep the codes in Git repositories, GitHub could be used as the server, as it provides an IDE and browser.

In addition to providing access to source codes in code repositories, NITE should support the use of pre-compiled executables on disk. Those should primarily be standard builds, for example, corresponding to an operational implementation or community release. Using code pre-compiled outside of the NITE system by a scientist or his/her collaborator should be discouraged as there will be no way to record the provenance of the code used in the experiment.

4.2.3 Suite definition and configuration

A suite is defined as a set of workflow components that is run using a certain configuration and workflow. One example (simplified for conciseness) is the operational NAM suite, composed of the following workflow components: NPS, GSI, NMMB, and UPP. Likewise, the HWRF suite is composed of WPS, prep_hybrid, GSI, vortex relocation, WRF, coupler, ocean model, and UPP. It can readily be seen that a single workflow component (in this case, UPP) can be part of multiple suites.

4.2.3.1 Predefined suites

NITE uses the concept of predefined suite, which is simply a suite that is supported out-of-the-box. Scientists can use a predefined suite as a starting point for conducting their runs or developing their own suite. Or they can completely bypass the predefined suites and build their own suite from scratch.

Predefined suites are nothing more than working sets of workflow components, configured in a specific way, deemed to be relevant for a group of scientists. Two important predefined suites, identified by EMC as priorities for the first NITE deployment, are NAM and NAMRR. Examples of predefined suites that could be added later are NGGPS, HWRF, RAP, and HRRR. Predefined suites do not need to be tied to operational systems. There could be pre-defined suites for any major scientific project or development effort, such as, NARRE, HWRF coupled with a different ocean model, HWRF ensemble etc. Note that predefined suites are referred to as *Component Sets*, or *compsets*, by the CESM community, which regards them as a very useful way of setting up CESM experiments.

Predefined suites are a means to help scientists make faster progress, and should not constitute a limitation to the types of suites that can be constructed by individual researchers. Through the NITE survey, potential NITE users identified the types of changes a scientist might want to make over a predefined suite. Those changes include sources code of a workflow component, physical parameterization namelist options, domain size, grid spacing, forecast length, output fields, etc. It is clear that these desired changes will only be possible if the workflow components themselves allow them. For example, for the domain size and grid spacing of the NAM suite to be changeable, it is necessary that these quantities not be hardcoded. Therefore, the more flexible the workflow components themselves are, the more flexibility scientists will have in setting up their experiments.

4.2.3.2 Configuring suites

The configurable parameters in a suite should be abstracted to a set of configuration files. NITE can include a GUI to assist in editing the configuration files. However, one requirement clearly expressed by EMC leadership is that a GUI for this purpose is not a top priority and should not be the exclusive way of composing the configuration files. Scientists should be able to configure suites by simply editing the configuration files.

One important aspect of suite composition is that there are limited degrees of freedom. Once a user makes a choice for a certain parameter, there are other parameters that depend on it. For example, when choosing a certain boundary layer parameterization, a compatible surface layer parameterization may need to be chosen. Or, when reducing the grid spacing, the timestep may need to be reduced accordingly. NITE should include a tool to check if the choices made in the configuration files are internally consistent. Additionally, NITE could include tools that assist scientists in making adequate choices. The latter could be implemented in the context of a GUI, in which changing one variable automatically prompts the user to adjust additional variables. While creating a tool to perform such checks is not difficult, documenting all the dependencies and internal constraints of each workflow component is no easy task. However, providing scientists with information about these constraints, and with tools to help make the proper choices and check for inconsistencies, will make the NCEP suites much easier to run by those with less experience. In fact, this is a recurring theme at every Annual WRF Workshop organized by NCAR: users want more guidance about how to make choices among the many degrees of freedom available in the WRF namelist.

4.2.4 Scripts

In addition to executing the workflow components, a myriad of small tasks need to be accomplished when running a suite, for example, creating working directories, staging the input datasets, creating the namelists using the information in the configuration files, invoking the executables, moving the files produced to delivery areas on disk, etc. At NCEP, each suite has many scripts to accomplish these tasks; most of them in ksh, with some exceptions, one of them being HWRF, which is now run using a set of object-oriented Python scripts.

A NWP suite typically involves a complex decision-making process. For example, the HWRF vortex relocation is composed of several executables, which are selectively called depending on characteristics of the observed tropical cyclone and on the availability of certain input datasets. In this example, the HWRF scripts control the decision-making process. It would be possible to rewrite the HWRF vortex relocation code in such a way that the decisions are made inside the Fortran code, which would reduce the complexity of the scripts. Alternatively, it would be possible to use an ESMF workflow approach in which the components and all the functionality are contained in a single executable. There is no absolute right or wrong way to partition the amount of logic in the scripts versus in the components codes themselves, it is a case-by-case decision. However, scripts need to be organized enough for NCEP collaborators to understand and make contributions.

4.2.4.1 Requirements

One important requirement to reach the NITE portability goal is that the scripts do not contain any automation features. All automation will be controlled by the workflow management system, described in the next section. This separation of automation from the suite control scripts allows for the scripts to be used with a variety of automation systems (or none at all). Operational forecast systems may use one automation system, while a graduate student working on a thesis might not need any automation to run a single case study.

We put forth here a few underlying principles for the scripts to be used by NCEP suites supported through NITE.

- Each workflow component in a suite has its own scripts, which contain only the code necessary to execute the workflow component. For example, WRF-ARW has a script and it is used by all suites that invoke WRF-ARW.
- Workflow component scripts do not contain automation features (no job submission/monitoring/polling and no batch system options)
- Each script accurately checks for its own success or failure and report either so the workflow management system can take appropriate action.
- Each workflow component script is idempotent, that is, the same output is obtained when a script is run multiple times. One requirement for achieving this is that scripts do not change/rename outputs from previous scripts and tasks in a suite.
- All platform-specific aspects of scripts are abstracted such that they can be specified or calculated at runtime.
- All parameters in a script are configurable at runtime. Values that could change for any reason should not be hardcoded. The location of input files, work directory, and output directory are configurable at runtime.

4.2.4.2 Standardization

NCEP has various suites that use the same workflow components. For example, both SREF and RAP use ARW, and both HWRF and NAM use UPP. However, currently the scripts to run these workflow components are not the same in each suite. For example, the script used to execute UPP in HWRF is completely different than the one used in the NAM. This diversity of scripts is an impediment to model development, as scientists trained in one suite cannot easily execute another. Additionally, this diversity makes it very difficult to make NITE expandable. If NITE is deployed initially for NAMRR (using its scripts), expanding it to another model, such as HWRF, would imply a completely new way of defining the suites.

While we recognize the cost of this effort, we recommend a consolidation of the scripting systems used in the NCEP suites. Each workflow component should have a single script to run it, configurable using configuration files. This can be implemented in a phased way, with the NWP systems progressively migrating onto the standardized protocols over a period of time.

4.2.4.3 Modularity and language

To achieve standardization, and avoid code duplication at all cost, we recommend that scripts be created in a modular way. This can be achieved by having a library of functions that perform small tasks, which can be assembled to compose the various scripts.

Using a scripting language that provides object-oriented functionality is a good choice for creating the functions and scripts, as it allows for easy refactoring of code. Using an object-oriented approach also allows for basic building blocks (e.g., run UPP) to be common between suites, but also allows their customization for a particular suite. (i.e., run UPP for HWRF). In this example, the common steps needed to run UPP are shared, and any specific HWRF needs are added to the HWRF instance. In fact, the HWRF Python scripts, with some modifications, could provide a solid starting point for a revised set of NCEP scripts.

4.2.5 Workflow management system

Since NCEP suites are complex and contain many tasks, it is generally not feasible for scientists to conduct experiments by submitting jobs individually. This situation is exacerbated for suites representing ensemble prediction systems. For this reason, NITE must use a workflow management system, a software system to manage complex collections of tasks that need to be carried out in a certain way, with complex interdependencies and requirements.

In addition to launching tasks based on dependencies with intelligence associated with throttling, workflow management systems can resubmit tasks when they fail, providing a fault tolerance. All this is done through interactions with the batch system, which provides information on job completion status.

At NCEP, the ecFlow workflow management system (developed by ECMWF) is used by NCO to run the operational suites. Since ecFlow is not available in the NOAA developmental machines, EMC scientists have historically used a system of *cascading scripts* to execute the suites in the research platforms. The *cascading scripts* provide a process for progressing through the scripts, with one script submitting the next. However, there are several issues with this method of launching jobs. One is that the *cascading scripts* do not provide fault tolerance, and, therefore, more manual intervention is required to run large sets of retrospective forecasts. Additionally, it is difficult to re-try a step that has failed (which is common during development), or to start a suite from a mid-point. The cascading scripts approach also ties the scripts to a particular batch system, and hinders portability between systems.

The limitations of the *cascading scripts* have led two EMC teams to start utilizing the Rocoto workflow management system developed at ESRL/GSD. Rocoto is available in all NOAA research supercomputers and can be easily ported to other platforms and batch systems. Both the HWRF and the NAMRR development are now utilizing Rocoto. Rocoto provides similar functionalities to ecFlow and to Cylc, the workflow management system developed at NIWA and employed at the UKMO. The most important difference between Rocoto and the other two systems is that Rocoto does not rely on a daemon process that runs continuously. Instead, it is invoked, performs its actions, and quits. Another significant difference is that Rocoto runs on the same platform as the NWP suite, while ecFlow and Cylc are generally installed on a separate server. These aspects make Rocoto easy to install by users, as it does not depend on the system administrators.

Since it is unlikely that ecFlow will be usable outside of NCEP operations, our recommendation is for the running scripts to be agnostic of the workflow management system. By eliminating all automation from the scripts, they should be executable by any workflow management system. This is the case with HWRF, which runs with Rocoto on *jet* and with ecFlow on WCOSS.

4.2.6 Experiment database

NITE contains a database for storage and retrieval of experiment metadata. The goal is to record provenance of codes, scripts, configuration files, and inputs related to an experiment so that the experiment can be reviewed and reproduced (Ma et al. 2014). This will substantially enhance the level of confidence that EMC staff has of experiments conducted by outside partners. It is important for collaborators to see each other's experiment configuration but it is also important to preserve the privacy of those who do not want to share their setup. Therefore, the first instantiation of NITE should have a central database with permissions setting for determining sharing properties. In subsequent instantiations, federated databases could be considered.

The review of modeling infrastructures revealed two distinct philosophies regarding recording experiment metadata. The first approach involves the scientist using a GUI, such as the ECMWF's prepIFS, to enter the experiment configuration. The configuration is then saved in the database and the experiment is launched. The second approach involves the scientist configuring the experiment by editing files on disk. After the experiment is configured, the scientist captures the setup in a database, and launches the experiment, as is done in the NCAR CESM Experiment Database. While both approaches are possible, we recommend that NITE adopt the second approach (metadata capture after the experiment is configured). One reason for this recommendation is that model configuration GUIs are difficult to maintain, especially in an environment of rapid development, where model namelist variables are changing. Additionally, there is a perception among scientists that a model configuration GUI could slow down the development process. The second reason for the recommendation is that this process is more reliable and allows for fewer discrepancies between what is stored in the database and what the scientist ultimately ran.

Consideration must be given to what metadata should be archived. The list of metadata will vary among modeling suites, since they have different workflow components. For that reason, it is important that a specific list of metadata be attached to each workflow component supported by NITE.

List of metadata that needs to be recorded

- Workflow components in the suite.
- Origin of executables, including whether they were pre-compiled or compiled as part of the experiment. In the latter case, information on provenance of source code and on any changes made to it. Source code provenance should be recorded as a tag or revision number in a code repository. Compiler options should also be saved.
- Origin of scripts and any changes made to them. Should refer to a tag or revision number in a code repository.
- Scientist name, institution, and project, with keywords and experiment description. The amount of description should be determined by each scientist, but plenty of tools should be provided for those that want to describe experiments in detail. In the UKMO, the experiment database connects to a wiki, in which scientists can upload papers and figures and get comments from collaborators.
- Values of variables in the namelists and configuration files.
- Source of all input files, including time dependent and static (*fix*) files, not generated within the suite.
- Date(s) in which the experiment was performed and platform(s) used.
- Description of archived files (if any).

Most experiment databases do not store all the provenance information but instead store the differences between a customized experiment and a pre-existing one. This reduces the volume of information needed and makes searches more effective. Careful consideration must be given to structuring the database so that searches are as fast and efficient as possible.

4.2.7 Documentation and training

Documentation and training on both the NWP systems and NITE itself are of paramount importance for the success of this effort. Every workflow component in NITE needs to have a Users' Guide. A scientific documentation is also recommended, when applicable. This information needs to be available electronically so it can easily reach a distributed audience. Some of the types of information that need to be conveyed to scientists are listed below.

- Functionality of the workflow component.
- Inputs and outputs to the workflow component.
- What aspects of the workflow component are configurable, and how to customize them.
- How to run the workflow component.

In addition to general information about the workflow component, it is necessary to provide information about how each workflow component is configured within the predefined suites. For example, the workflow component GSI is configured in different ways for NAM and HWRF. The GSI configuration encompasses which datasets are used, how many outer loops are employed, whether the hybrid capability is invoked, etc. For scientists to be productive using and improving upon predefined suites, they need to know how they are configured.

Training on using NITE will be extremely important, especially in the early years. If the system is well documented and easy to use, scientists will be more likely to adopt it. Survey respondents identified online tutorials as an important way to get training, with in-person tutorials being of secondary importance.

As evidenced above, NITE will contain a plethora of documentation. Early consideration should be given to standardizing how it will be provided.

4.3 Incremental NITE implementation

The development and implementation of NITE will require a substantial commitment of time and resources from the NWS. For this reason, an incremental implementation is recommended. There are multiple ways in which an incremental implementation could be devised and a final decision will have to be made based on the priorities and resources available. A possible progression is summarized in Table 1, with a goal of creating an initial capability in the NOAA platforms.

The capabilities within Phase I are focused on the NAM and NAMRR predefined suites and variations thereof. With all configuration aspects of these suites well-documented and abstracted to configuration files, scientists are able to create variations of these predefined suites to conduct their experiments, which are recorded in a basic experiment database. Analyses, forecasts, and observations associated with GFS operational runs for four months (one in each season) are available for conducting the runs with the Rocoto workflow management system on *Zeus* and WCOSS, and with the ecFlow workflow management system on WCOSS. Initial standardized visualization, forecast verification, and archival capability system are available workflow components for any suite.

In Phase II, the capability is expanded to the NGGPS suite. At this moment we do not know which dynamic core will be selected for NGGPS, but we do know that the model will be a component within NEMS, so we already have significant amount of information available to start connecting it to NITE. For NGGPS initialization, dataset availability is extended to realtime. Other expanded capabilities of Phase II are the inclusion of GEFS data, enabling the generation of NAM and NAMRR limited-area ensembles using IC diversity created by downscaling GEFS. In Phase II the relational experiment database is fully functional, and the possible platforms for running the experiment are expanded to additional NOAA research and NCAR supercomputers. The standardized tools for visualization and forecast verification reach maturity during this phase.

Table 1. Proposed incremental implementation of NITE. With each phase, additional capability is added.

		Phase I	Phase II	Phase III
Data Management	Analyses/Forecasts	GFS operational	GEFS operational	GFS reforecasts
	Observations	GFS PrepBUFR		Raw observations and processing tool
	Timeline	4 months retrospective plus case studies	Realtime data available in NOAA HPSS	Multi-year retrospective
	Access	NOAA HPSS	NCAR HPSS	Data service
	Formats	No change – various		Standardized
	Level of processing	No change – postprocessed		Native
	Metadata	Initial capability		Metadata database
	Retrieval	No change – manual		Retrieval by subsets and of processed data
	Archive Results	Postprocessed files to NOAA HPSS	NCAR HPSS	Data service
Additional aspects	Computational Platform	*Zeus* and WCOSS	Yellowstone and other NOAA research	Generic
	Workflow Management Systems	Rocoto and ecFlow		
	Predefined suites	NAM and NAM-RR	NGGPS	Others (such as HWRF, HRRRE etc.)
	Suite composition and scripts	All configuration abstracted to config files		
	Experiment Metadata	Initial implementation	Completed relational database	Expanded search/query/summarize tools
	Visualization and Verification	Initial capability	Fully implemented	
	Documentation and training	Basic, for friendly users	Complete, including tutorials	Expanded

Phase III is marked by the ability to run NITE on generic platforms, with support for multiple batch systems. Input datasets, with metadata browsing and intelligent retrieval capability, now include multi-year model and raw observations (with processing tool), and are available through data services such as NOMADS and MADIS, making them easily retrievable by community users. The data archives are mature, distributed, and allow storage of raw files, which are postprocessed upon retrieval. Additional suites, such as HWRF, are added to the system. The experiment database is enhanced with additional tools for querying and retrieving information.

As stressed above, there is plenty of flexibility for making changes to the incremental implementation, depending on resources. It is important that each phase provide additional capabilities, allow additional types of experiments, and be aligned with NOAA's priorities.

5 Conclusions and recommendations

The NCEP models have become progressively more complex over the years, posing a challenge for scientists inside and outside EMC to configure, launch, and keep track of experiments. This report outlined the design for NITE, an infrastructure intended to facilitate model development and testing, with the goal of expediting the transition of NWP research to operations. While this design was prepared by the DTC, a group that has historically focused exclusively on LAMs, the design is generic and can apply to both global and LAMs. In fact, NITE can be a very useful tool for development of the NOAA NGGPS, and synergies with that program are encouraged.

NITE will have several elements: data management to facilitate access to both input and output datasets, code management and build systems to centralize and keep track of code development, suite definition and configuration tools, scripts, workflow management system, experiment database and documentation. The recommendations for each element are described in Section 3 and synthesized in the Executive Summary. Figure 4 contains an example of how a scientist might employ several aspects of NITE when conducting an experiment.

NITE can be implemented incrementally, as suggested in Table 1, with the understanding that the phases suggested in Table 1 can be altered to best match NOAA priorities and funding. In the suggested scenarios, NITE's first deployment is focused on the NAM and NAMRR suites for scientists with access to NOAA platforms. In later deployments, more suites become available and the system is accessible from additional, non-NOAA, platforms.

Funding for developing NITE has yet to be identified, and it should be stressed that NITE cannot be developed as a side project by the already busy EMC scientists. It is very important that EMC, DTC, and community scientists remain involved in the NITE development, but very qualified software engineers will be needed for the majority of the work. It should be stressed that NITE will not be developed without some intrusion onto EMC daily work. Scripts will need to be redeveloped, potentially using more modern languages, and suites will have to be transitioned from current systems to NITE, requiring staff development.

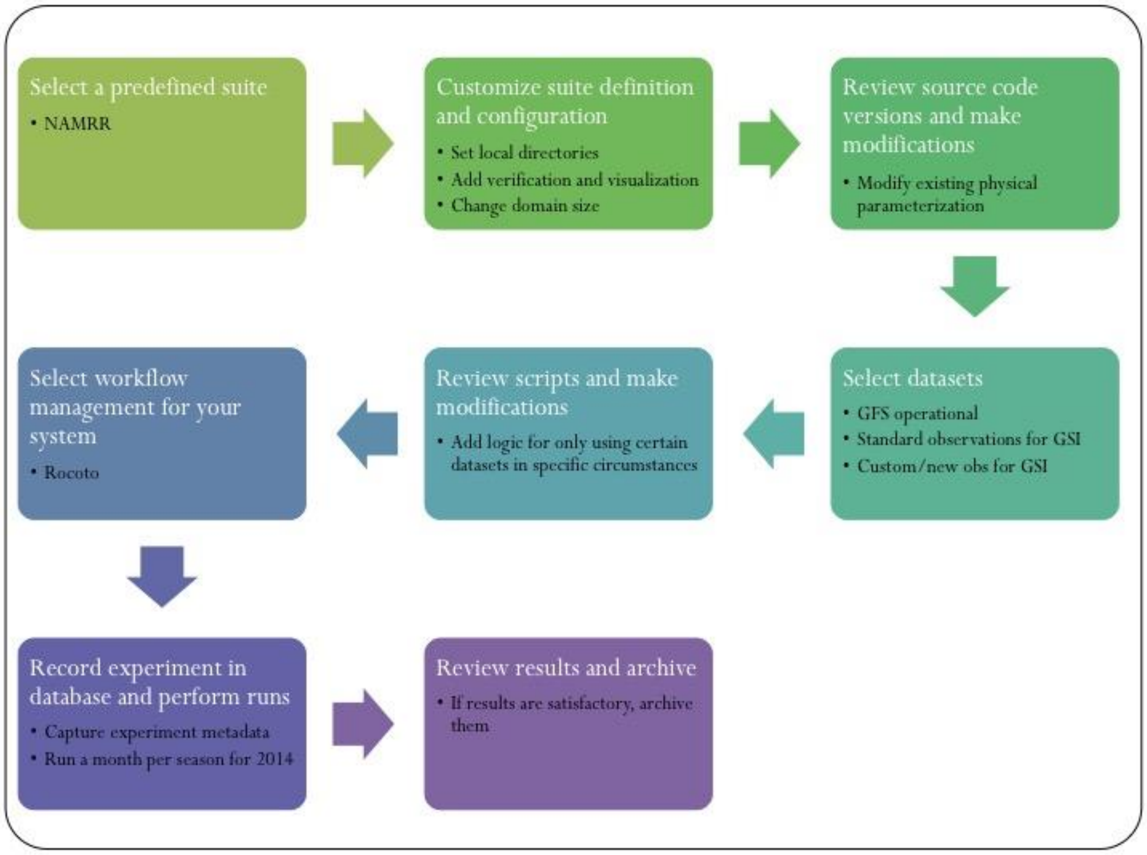
Select a predefined suite
• NAMRR
Customize suite definition and configuration
• Set local directories
• Add verification and visualization
• Change domain size
Review source code versions and make modifications
• Modify existing physical parameterization
Select datasets
• GFS operational
• Standard observations for GSI
• Custom/new obs for GSI
Review scripts and make modifications
• Add logic for only using certain datasets in specific circumstances
Select workflow management for your system
• Rocoto
Record experiment in database and perform runs
• Capture experiment metadata
• Run a month per season for 2014
Review results and archive
• If results are satisfactory, archive them

Figure 4. Example of how a scientist might use NITE to conduct an experiment.

NOAA already has several tools that can be used as a starting point for NITE. In particular, NOMADS can be expanded for model data management, MADIS data standards can be utilized for standardizing observations, the Rocoto workflow management system can be used for launching jobs in non-operational settings, the HWRF object-oriented Python scripts can be used as a prototype, the VLab experience can be leveraged for sharing code through Git, and the experiment database used in the WRF Portal can be expanded to capture more complete experiment metadata.

The NITE development will involve a comprehensive effort that will require large initial investment, continuous growth and training, with the outcome of extensively facilitating how researchers interact with NCEP operational models. Finally, the NITE code management and experiment database will provide accurate recording of provenance of experimental results, making them relevant for consideration by EMC, enhancing the number of developments made available for potential operational implementation.

6 Acknowledgements

The DTC sincerely thanks all scientists that participated in the survey and focus groups. Additionally, we thank our colleagues at NCAR, ESRL, EMC, ECMWF and UKMO for the valuable input. The DTC is funded by NOAA, AFWA, NSF, and NCAR.

7 References

Bernardet, L., S. Koch, E. Szoke, A. Loughe, J. L. Mahoney, L. Nance, M. Demirtas, T. Fowler, R. Gall, H.-Y. Chuang, and M. Pyle, 2008: The Developmental Testbed Center and its Winter Forecasting Experiment. *Bull. Amer. Meteor. Soc.*, **89**, 611–627. doi: http://dx.doi.org/10.1175/BAMS-89-5-611.

Bernardet, L., V. Tallapragada, S. Bao, S. Trahan, Y. Kwon, Q. Liu, M. Tong, M. Biswas, T. Brown, D. Stark, L. Carson, R. Yablonsky, E. Uhlhorn, S. Gopalakrishnan, X. Zhang, T. Marchok, B. Kuo, R. Gall, 2014. *Bull. Amer. Meteor. Early release available at* doi: http://dx.doi.org/10.1175/BAMS-D-13-00093.1

Ma, X., P. Fox, C. Tilmes, K. Jacobs, and A. Waple, 2014. Capturing provenance of global change information. *Nature Climate Change,* **4**, 409–413, doi:10.1038/nclimate2141.

Trahan, S. G., T. P. Brown, T.-L. Hsiao, B. Thomas, C. Holt, L. Bernardet, V. Tallapragada, H. Tolman, B. Kyger, and W. M. Lapenta, 2015. Modernizing the Operational Workflow and Automation of the NCEP Hurricane Weather Research and Forecast (HWRF) Modeling System using Python and Rocoto. 5th AMS Simp. on Advances of in Modeling and Anal. Using Python. San Diego, Jan 5-8.

UCAR, 2014. Annual Report of the UCAR Community Advisory Committee for NCEP Available at http://www.vsp.ucar.edu/UCACN/NCEP%20Final%20Reports/UCACN_Report_Jan2014_v10-22Sep2014.pdf.

8 Appendix 1 – List of acronyms

AFWA – Air Force Weather Agency
AOML – NOAA Atlantic Oceanographic and Atmospheric Laboratory
ARL – Army Research Laboratory
ARW – Advanced Research Weather Research and Forecasting model
BUFR - Binary Universal Form for the Representation of meteorological data
CESM – Community Earth Modeling System
CompSets – Predefined suites in CESM
CF – Climate and Forecast (a standard for metadata)
CFS – Climate Forecast System
DTC - Developmental Testbed Center
ecFlow – ECMWF workflow management system
ECMWF - European Centre for Medium-Range Weather Forecasting
EMC – NOAA Environmental Modeling Center
ESMF - Earth System Modeling Framework
ESRL – NOAA Earth System Research Laboratory
FCM - Flexible Configuration Management
GFDL – NOAA Geophysical Fluid Dynamics Laboratory
GRIB - Gridded Binary data format
GEFS – Global Ensemble Forecast System
GFS – Global Forecast System
GSI - Gridpoint Statistical Interpolator
GSM – Global Spectral Model
GUI - Graphical User Interface
HPSS - High-Performance Storage System
HRRR - High-Resolution Rapid Refresh modeling suite
HRRRE – HRRR Ensemble
HRW – High-Resolution Windows modeling suite
HWRF - Hurricane WRF modeling suite
IC – Initial Conditions
IDE – Integrated Development Environment
IFS - Integrated Forecast System
IPCC – Intergovernmental Panel on Climate Change
LAM – Limited Area Model
MADIS - Meteorological Assimilation data Format System
MARS - Meteorological Archival and Retrieval System
NAM – North American Mesoscale modeling suite
NAMRR – North American Mesoscale – Rapid Refresh modeling suite
NCAR – National Center for Atmospheric Research
NCDC – NOAA National Climatic Data Center
NCO – NCEP Central Operations
NEMS – NOAA Environmental Modeling System
netCDF – Network Common Data Form
NGGPS – NOAA Next-Generation Global Prediction System
NITE - NWP Information technology Environment
NIWA - National Institute of Water and Atmospheric Research
NCEP - National Centers for Atmospheric Prediction
NMM – Non-Hydrostatic Mesoscale Model

NMMB	– Non-hydrostatic Multi-scale Model
NOAA	- National Oceanic and Atmospheric Administration
NOMADS	- NOAA National Operational Model Archive and Distribution System
NPS	– NEMS Preprocessing System
NSF	– National Science Foundation
NWP	- Numerical Weather Prediction
ODB	- Observational database
OPeNDAP	- Open-source Project for a Network Data Access Protocol
PrepBUFR	– Quality controlled observations in BUFR format
PrepIFS	– GUI to configure the IFS
PP	– Post Processing UKMO data format
QC	– Quality Control
RAP	- Rapid Update Cycle modeling suite
SAB	– Science Advisory Board
SMS	- Supervisor Monitor Scheduler
SREF	- Short-Range Ensemble Forecast suite
SVN	- SubVersion
UKMO	- United Kingdom Meteorological Office
UCACN	- UCAR Community Advisory Committee for NCEP
UM	- Unified Model
UPP	– Unified Post-Processor
VLab	– NOAA Virtual Laboratory
WCOSS	– Weather and Climate Operational Supercomputing System
WPS	– WRF Preprocessing System
WRF	- Weather Research and Forecasting model