

Pat J. Fitzpatrick*, Avichal Mehra, Jim Corbin, Joe McCaffrey, Tomasz Haupt,
 Purushotham Bangalore, Gregory Henley, Robert Moorhead, and John van der Zwaag
 Engineering Research Center, Mississippi State University
 Stennis Space Center, Mississippi

Matt Bettencourt
 Center for Higher Learning, University of Southern Mississippi
 Stennis Space Center, Mississippi

1. INTRODUCTION

The past decade has witnessed the proliferation of mature modeling packages as well as rapidly evolving computer technology which is making high-resolution modeling possible. In addition, the Internet phenomenon with constantly increasing network speed has driven technology revolutions, one of which is the development of web portals with easy user interfaces that hides the complexities of the underlying software and hardware. Currently, numerical models require considerable programming and UNIX knowledge, a familiarity with different computer platforms, and experience with serial, shared memory, and distributed memory hardware. Furthermore, usually modelers must use a clunky shell script interface instead of a Graphical User Interface (GUI), and currently no models can be run through a web portal.

The goal of the Distributed Marine Environment Forecast System (DMEFS) is to develop a web portal infrastructure which supports research and operational modeling needs, with "pop-up" user-friendly GUI interfaces, and hides the complexities of running a numerical model. Since DMEFS is a new approach to modeling, the software is setup in a generalized fashion to allow for future developments or alterations as the software evolves. These packages will include a suite of METeorology and OCeanography (METOC) models --- initially those of Navy interest, but can be extended to any model. Although each model uses unique applications to process input and output data, such as quality control, data interpolation schemes, and graphics, from the DMEFS user's perspective, the same functions will be done the same way through a common graphical user interface. Thus enormous time is saved in training and indoctrination of new personnel or when introducing a new model. Eventually, this infrastructure will support easier model coupling by providing wrappers using C, C++, and FORTRAN code built around the input and output of all the models without rewriting legacy code. Another benefit is that this project supports genuine, synergistic collaboration among computer

specialists, model developers, and operational users. A description of DMEFS now follows.

2. OVERVIEW OF DMEFS AND ITS WEB PORTAL INFRASTRUCTURE

Industry has undertaken enormous efforts to develop easy user interfaces that hide the complexity of underlying systems. Through Web Portals the user has access to variety of services such as weather forecasts, stock market quotes and on-line trading, calendars, e-mail, auctions, air travel reservations and ticket purchasing, and many others yet to be imagined. It is the simplicity of the interface that hides all implementation details from the user that has contributed to the unprecedented success of the idea of a web browser.

In this work the notion of the web portal to provide web access to *remote computational resources* (that is, hardware, software and data), thereby simplifying currently difficult to comprehend and changing interfaces and emerging protocols. More specifically, the systems hides the complexity of the inherently heterogeneous and distributed computing environment, and through the familiar Web Browser interface, allows the user to specify the computational environment.

DMEFS is a "multitier" system (to be explained in a moment) that extends the user desktop by providing a seamless access to remote high performance computational resources and data, and allows for integrating existing commodity components into a single, user-friendly, web accessible system without compromising security and integrity of the computational resources. In addition to modeling functions, the following services will be included in the future: model validation; visualization packages; real-time input data for initialization, boundary conditions, and data assimilation; and grid generation services. Since ultimately DMEFS will be tested in a Navy High Performance Computing environment, a major challenge is integrating the components into a single, user-friendly, web accessible system without compromising security. Currently DMEFS is

* Corresponding author address: Pat J. Fitzpatrick,
 ERC, BLDG 1103-233, Stennis Space Center,
 MS, 39529-6000; e-mail: fitz@erc.msstate.edu.

integrating two security models: kerberos targeted towards DoD supercomputing facilities, and PKI (Public Key Infrastructure) for NSF supercomputing facilities.

3. DMEFS MULTITIER ARCHITECTURE

3.1 Front End

Overall, three different groups of DMEFS can be identified: researchers, operators and managers. DMEFS must provide an environment for METOC model developers to perform research and development, for Navy operational users to launch daily operations; and for managers to access information and support for making decisions. The needs vary from access to high-end performance machines for the model developers and operators to a portable personal computer for some users only interested in end products and information. All such needs are satisfied through a web-based GUI. By using a standard web browser to deploy the front end of the system, the user is in a familiar friendly environment to access remote resources, hiding the complexity of the rest of the system. The three main functions of the front end: research and development, operations and decision support map directly onto the three different types of DMEFS portal: researchers, operators and managers.

3.2 Middle Tier

It is important that the tasks defined by the user including one or multiple applications are expressed in a platform independent way. This is achieved by using the concept of "proxies" representing actual codes and data which are contained in an application. An application proxy includes: information on location of the executables and scripts; arguments and switches; expected input files and output files; and the location of a template batch script, if available. All this information is entered in the system through a formal registration process. This way integration of both legacy and new applications into the system is easy.

The middle tier of the system is implemented as a shared collection of persistent components and services using Enterprise JavaBeans (EJB) technology. The EJB model introduces two types of components (or "Beans" in java jargon): persistent entity beans and session beans. The entity beans are used to represent components such as users, projects, tasks, and application proxies. The session beans represent services which operate on these entity beans. They provide access to back end remote resources, i.e., the high-performance system, file systems, databases etc. through a platform independent "computational grid interface" (Globus metacomputing toolkit). This way, the middle tier provides a transparent mapping between task and resource specification, hiding the

complexity of the heterogeneous back end systems from the user at the front end.

The research support functions require the most flexibility and demand for real-time system configuration. The basic tasks for a research user can be broadly defined under four categories: new model development, transition to operations, environmental simulations and validation. The operational user is more focused on product generation and less on processes, but needs more automation and a tailored environment for routine run streams for both operations (to run a model or to derive products from archived model runs) and transitions. The management user focuses on accessing the final products, such as forecasts, rather than running models. The manager can be from research or operations, but can also represent project, program or sponsor management, or users from the public domain. These users are distinctly high-level users of the system and would use web access primarily for information only. This information could be used for generating reports, making briefs, monitoring the status of the project and/or searching for relevant documents and references.

4. DMEFS Web Interface

4.1 Login Procedure

DMEFS is accessed via a web browser and the interface presents a login prompt. A registered user can then login presenting his/her credentials (a digital certificate in case of PKI or a ticket in case of Kerberos). In addition, a group identity is selected from the following categories: customer, operator, analyst, developer, administrator, demonstrator, and visitor.

All logins have distinct user interfaces and a hierarchy of access privileges. The customer typically is only interested in viewing METOC modeling products and has access to different METOC end products. The operator, in addition to products, also has monitoring access to the status of various submitted, executing and completed runs. The analyst has interactive access to visualization products. A developer has all the privileges discussed so far and also has the additional responsibility of registering applications and testing and validating these applications. This role is discussed in more detail in section 4.3. A developer would be primarily interested in research and an analyst could be either an operator or a researcher.

The administrator's privileges are very similar to the UNIX "root" access with all the main responsibilities of managing user accounts, job schedulers and performing machine maintenance. A visitor is, as the name implies, would casually visit the DMEFS

web site with minimum or none of the privileges discussed so far, and a demonstrator role is chosen for performing demonstrations and presentations by assuming a specific role from any of the categories discussed above.

4.2 Projects, tasks and applications

Once logged in, the developer's workspace in DMEFS is organized under projects and registered applications. A project is a very general definition of a collection of tasks which have a common long-term goal. A research endeavor to achieve a common objective, or a field program for making detailed observations, or an operational run stream – all fit a definition of a project within DMEFS. Each project then consists of multiple tasks which is akin to the sub-division of a big long-term assignment (projects) into smaller sub-assignments (tasks). Each task in turn consists of one or multiple applications, where an application is a proxy object comprising the application descriptor defined during the registration process. A user can then configure each of these applications by using an auto-generated interface and/or by importing a complex and custom-made interface during the registration process.

A user can view, edit, configure, delete or add any of these registered applications to a chosen task. These applications then show under the chosen tasks description, where they can be viewed, linked with other applications, deleted and/or published as public applications. After completing the task description, a user can then submit the task and then monitor its progress.

4.3 Registering an application

An application can be any software package (codes and data), and can be registered as a private or as a public application. The former are restricted to

individual user's access, while the latter are available to all privileged users. These applications are installed on hardware platforms which themselves are registered within DMEFS. Installing (or registering) an application within DMEFS requires the developer to define its attributes by filling an application descriptor html form. Information on the location of its executables and script files, input data including switches and parameters, output data for archiving, visualization and for coupling with other applications, and all information needed to generate a runtime batch script on a targeted hardware are fed to the application descriptor. Once registered, information in the application descriptor is used to automatically generate a GUI, which then any authorized user can use to configure the application for submission. Alternatively, if the application is complex, a formal GUI can be registered in the application descriptor.

Recently a complex GUI for running the Wave Model (WAM) has been developed, including options for choosing domain size, nesting specifications, forecast length, grid spacing, etc. Another GUI example for running the Coupled Ocean Atmosphere Mesoscale Prediction System (COAMPS) is shown in Fig. 1. This GUI, which is more complex than the WAM GUI, includes menu options for map projections, grid settings, nesting options, forecast length, data assimilation options, output frequency, and a choice of surface, pressure, height, sigma, etc output parameters including a selection of interpolated vertical levels. An example of surface-derived output parameters is shown in Fig. 1. Future models to be incorporated in DMEFS include the Navy's Coastal Ocean Model (NCOM) and the Princeton Ocean Model (POM).

Acknowledgments. The Office of Naval Research under Contract No. N00014-00-1-0886 provided support for this work.

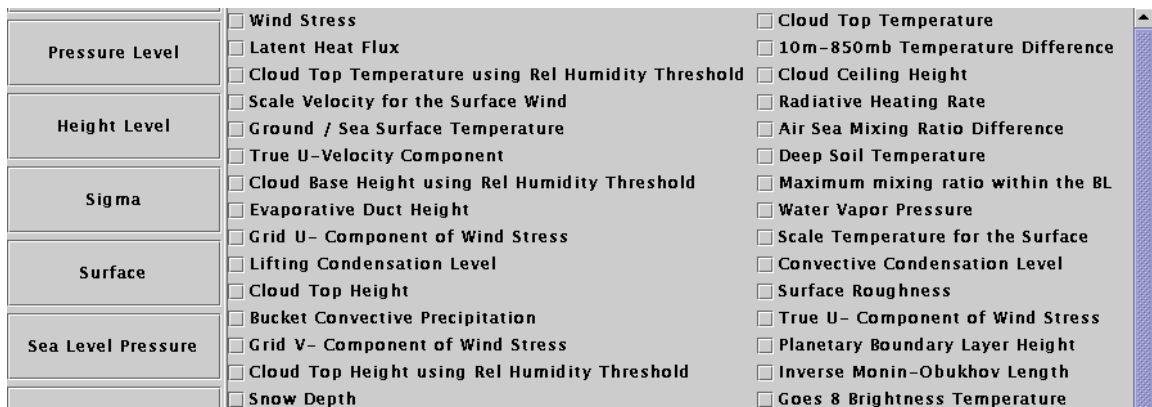


Figure 1. Example of the output selection portion of the COAMPS GUI. In this example, the user selects which surface-derived variables to output. Only a few of the 61 possible surface variables are shown. Other output options exist for different coordinate systems, such as pressure, height, and sigma.