
Functional Requirements Specification

for the

Model Manager

Version: 1.0

Prepared by: Elena Schuler

April 7th, 2006

Table of Contents

- 1. Introduction.....1**
 - 1.1 Purpose1
 - 1.2 Document Conventions.....1
 - 1.3 Intended Audience.....2
 - 1.4 Project Scope.....2
 - 1.5 References.....2
- 2. Overall Description.....2**
 - 2.1 Product Perspective.....2
 - 2.2 Product Features.....3
 - 2.3 User Classes and Their Characteristics.....3
- 3. System Features.....4**
 - 3.1 Set up and submit a model job.....4
 - 3.2 Set up and submit a “post-processing” job.....8
 - 3.3 Submit a 'By-hand' job.....9
 - 3.4 Load a job configuration from a file and submit the job.....9
 - 3.5 Retrieve and run a previously saved job configuration.....10
 - 3.6 View scheduled, running and old jobs.....10

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This document defines the functional or high-level user requirements for the Model Manager software tool. It describes the main functionality of the tool from the user's perspective. This document will serve as input for a potential Software Requirement Specifications (SRS) document.

1.2 Document Conventions

“MM” - Model Manager

“4DWX OTM” (short: OTM) - “4DWX On the Move”

“model back end system” – a term describing configuration files, data input, scripts and executables that are needed to run model jobs

“model job” - a collection of processes computing a weather or climate model over a given domain and for a given time period

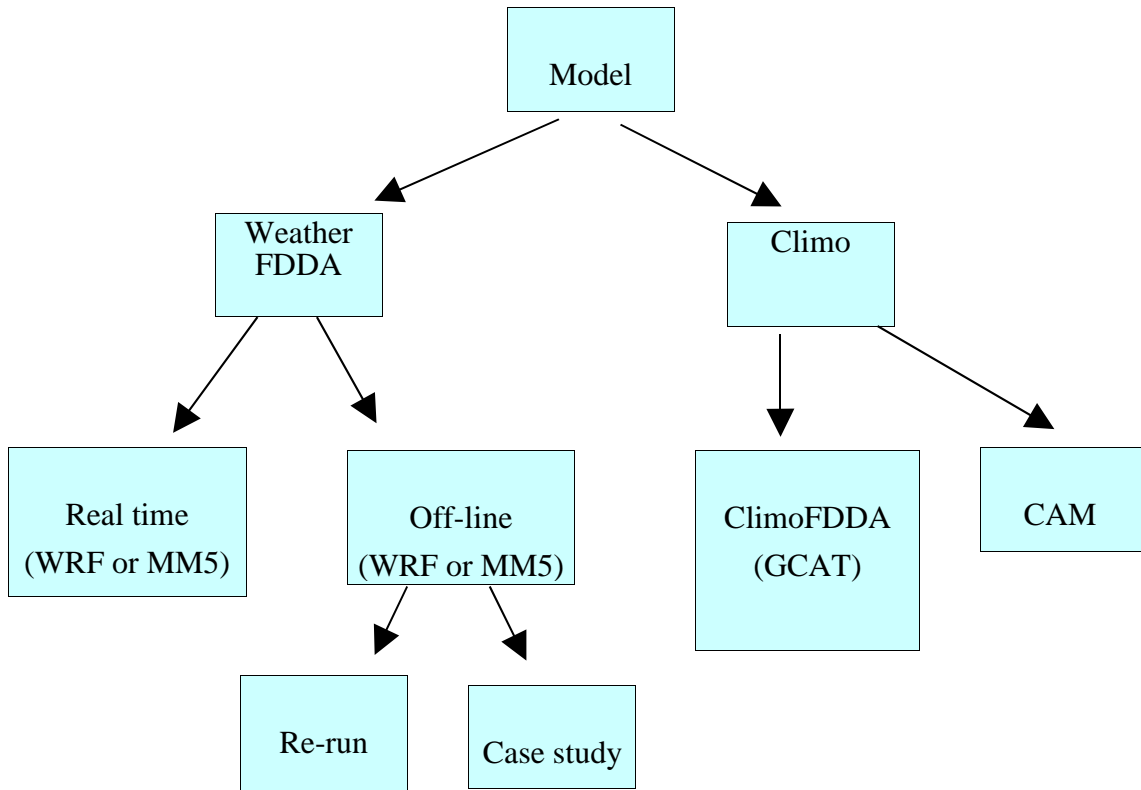
“climo job” - either a “ClimoFDDA”-job (GCAT) or a CAM job

“GMOD job” - the current operational RTFDDA job using MM5 or WRF

“RTFDDA job” - a GMOD job or a RTFDDA ensemble

“off-line FDDA job” - a re-run of a previous RTFDDA cycle or a case study.

Using the above conventions, we can categorize model jobs for the MM that as follows:



"Post-processing' job" - a set of processes that take model output files as input, perform calculations or data conversions on the input and possibly produce output files in a different data format

"by-hand' job" - an existing set of scripts and executables that were set up by the user on a cluster in advance. A 'by hand' job can run any executable.

1.3 Intended Audience

The intended audience for this document is the users of the system (see 2.3), scientists and software engineers at RAL who will be developing the system, NSAP project sponsors and management.

1.4 Project Scope

The Model Manager is a software tool that will allow the user to configure, schedule, run, monitor, and stop (and re-start /resume) model jobs. The primary goal of this project is to extend the current model back end system and to automate the setup, running and monitoring of model jobs. This will include a more automated node management system.

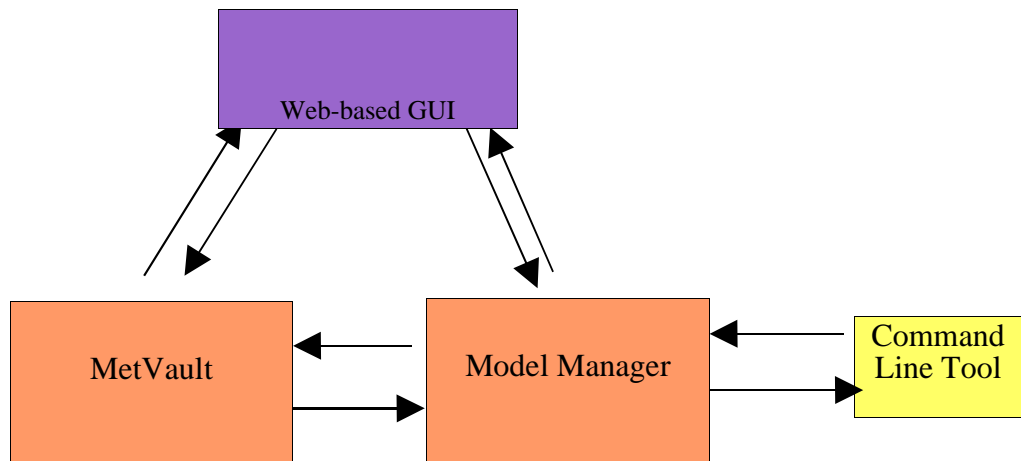
1.5 References

[1] "4DWX on the Move (4DWX OTM)", Scott Swerdlin, concept paper:
http://sdg.rap.ucar.edu/wiki/index.php/Model_Manager#Documents

2. Overall Description

2.1 Product Perspective

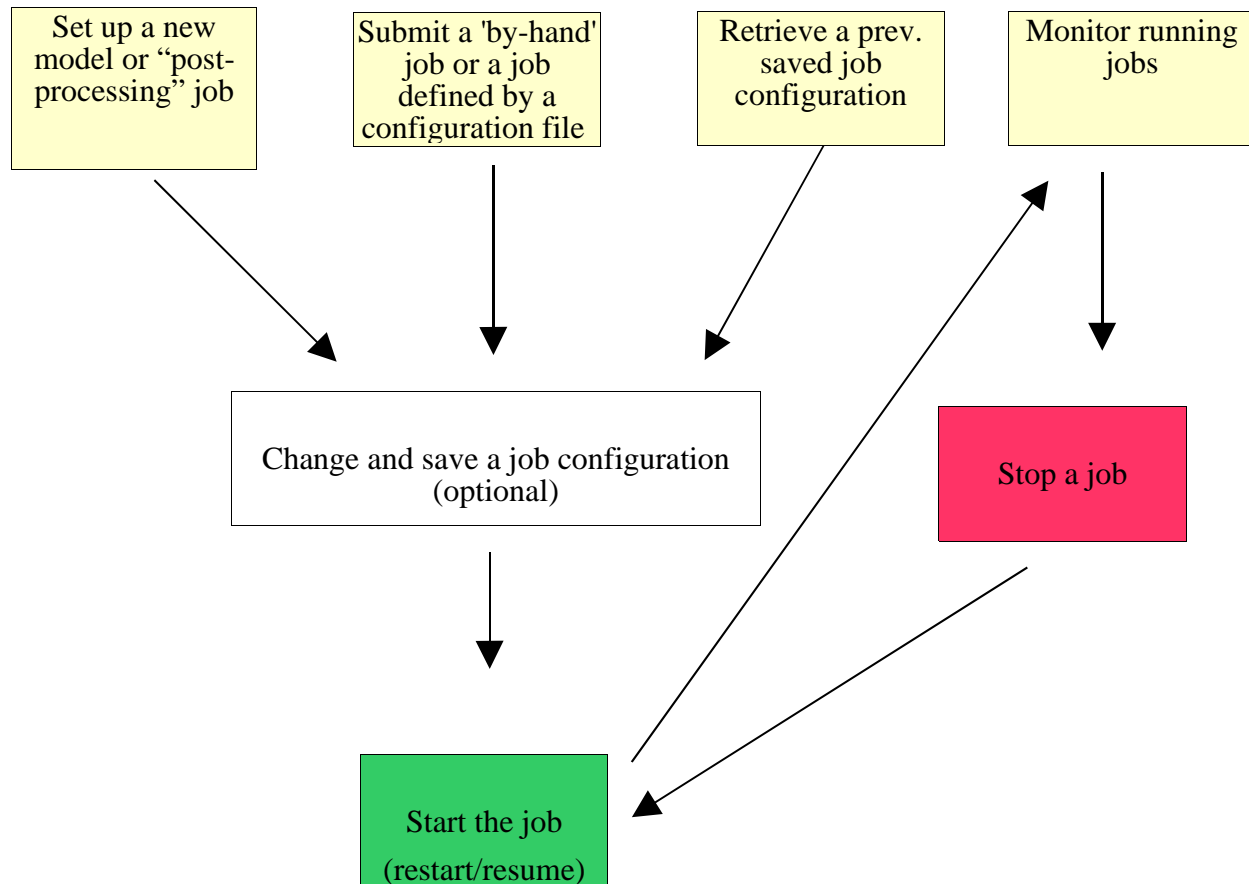
With the increase in NSAP projects, staff and hardware, a more automated and procedural, and in some cases, easier way of setting up and running a model job has become necessary. The MM will provide this extension and enhancement to the current model back end. The MM is a stand-alone software tool, which will also be part of the 4DWX OTM system (see [1]).



The MM can run in conjunction with the MetVault, and it can also run as a stand-alone application with no knowledge about a MetVault. The MM will be accessible from a web-based GUI and a command line tool.

2.2 Product Features

The main functions of the system are described in the diagram below. Users of the system will be able to:



2.3 User Classes and Their Characteristics

There are three main user classes of the system. First, scientists and software engineers at NSAP, who will be using the system for setting up and maintaining operational model runs. These are users who are experienced and familiar with the model back end. A second group of users consists of scientists at RAL (and also outside of RAL), who will be using the system for running model jobs for research purposes. This group is experienced and familiar with the model setup. Users in this group may want to setup and run customized model jobs, e.g., provide their own input data and own input data processors. A third group consists of users who are less familiar with model setups. Users in this group might want view the status of operational model runs, stop and restart existing jobs or set up and start a 'standard' model job.

3. System Features

This chapter describes the use cases for the MM in more detail.

The MM will manage jobs on one or more clusters. Node allocation, management and utilization will be provided by the MM (SRS note: The MM provides centralized control over a collection of clusters.). For a user, this has the following implications: A user may not want or need to know how many and which clusters a MM is managing. S/he will submit a job to the MM and the MM will make the decision on what cluster(s) the job will be running. On the other hand, a user might need to submit customized jobs to a specific cluster. In this case, the user will need to have the option to specify what cluster his/her job should run on. Managing more than one cluster, will also allow RTFDDA and climoFDDA ensembles to run across several clusters.

Once a user is logged in into the system, s/he will be presented with the features in 2.2:

- Submit a new job
- Retrieve saved job configurations
- View the job queue/monitor running jobs

A new job can be submitted to the MM four different ways:

- The user can set up a new model or 'post-processing' job through the Job-Setup module and submit it (3.1 and 3.2).
- The user can submit a 'by-hand' job (3.3).
- The user can submit a job by supplying a job configuration file to the MM (3.4).
- The user can retrieve a job configuration that was previously saved with the MM, change it and re-submit that job (3.5).

If the user chooses to submit a new job, s/he will be given the following options to choose from:

- "Setup a new model job"
- "Setup a new 'post-processing' job"
- "Submit a 'by-hand' job"
- "Submit a job configuration file".

Note that the above terminology is subject to change.

3.1 Set up and submit a model job

This feature allows the user to set up and schedule a model job using the MM's Job-Setup module. Suppose the user is logged on to the system and has made the following choices "Submit a new job" -> "Set up a new model job". S/he is then presented with two more options: "Weather FDDA" and "Climo".

Use Case "WeatherFDDA": Setting up a real time or off-line FDDA job

The objective of this feature is to automate the set up of new real time and off-line FDDA jobs. This use case describes the set up of GMOD jobs, re-runs and case studies. At this point, RTFDDA ensemble jobs will be submitted to the MM as 'by hand' jobs or through a job configuration file. The MM will provide a default GMOD job configuration, which can be changed by the user (TBD: Define the defaults for a GMOD job). It is important to note that the model manager will accept and run "custom" GMOD jobs. These are jobs that are also set up through the Setup-module, but do not use the default GMOD configuration. For a "custom" GMOD job, e.g., the user may choose to supply his/her own input data, own pre-processors or a customized version of a MM5 executable.

MM's Job-Setup module will allow the user to substitute the default configuration, but it is the user's responsibility to make sure that these scripts, executables, etc. reside on the cluster or clusters where the job will be running on. Submitting a "custom" GMOD job through the Job-Setup module will allow the user to save the job's configuration with the MM.

Primary actor: Meteorologist, software engineer

Goal: Set up and run a real time and off-line GMOD job

Action Sequence:

1. User chooses to "Set up a Weather FDDA Job".
2. User may choose a cluster where the job should run on.
3. User decides what model should be used: MM5 or WRF.
4. User defines a JOBID.
5. User determines domains (TBD):
 - creates own domains (Note: This may only apply to MM5 jobs. From earlier discussions: creating domain files for WRF takes a long time.) or
 - chooses between a number of pre-defined domains or
 - submits own TERRAIN files
6. User defines when a job is to be run and/or what cycle to run. If the cycle time is in the past, then the user is prompted to specify whether the job is a "case study" or "re-run".
7. User supplies other job specific information, such as, cycle interval, forecast length and other applicable information.
8. User can specify whether to write restart files and the frequency of how often they are to be written (TBD: Is this a correct statement and does frequency only apply to WRF?).
9. User can choose between predefined sigma-level configurations or supply own sigma-level configuration
10. User has the option to specify the number of nodes to use.
11. User can choose to receive email notification upon start, end and termination of the job.
12. User chooses between standard or custom IC/BC data sources:
 - standard: ETA, AVNFTP (GFS), GFS004
 - or
 - custom: provide data source (e.g., host:Full_Path_to_Dir)
 - or
 - For off-line jobs, the user must specify the data source, i.e., location (MetVault or a directory) and time period. Important note for re-runs, if the input data is to obtain from the MetVault, then MetVault returns the data that was available and used in that cycle.
 - TBD: Determine standard IC/BC data source and standard pre-processors.
11. Depending on the choice above, user can provide custom IC/BC pre-processor or choose the standard:
 - standard processing
 - or
 - provide own pre-processing script

12. User is given the option to run additional pre-processors for the IC/BC data, such as, LDAS or supply own custom pre-processor or skip this option.
13. User chooses between standard and/or custom obs data sources and processing:
 - Standard: WMO, SAMS, MADIS, GTS, RAWs, okmeso, SatWinds, ACARS, etc. and/or
 - provide custom obs source1 and custom obs processor1
 - provide custom obs source2 and custom obs processor2
 - etc.or
 - For off-line jobs, the user must specify the data source, i.e., location (MetVault or a directory) and time period. Important note for re-runs, if the input data is to obtain from the MetVault, then MetVault returns the data that was available and used in that cycle.
 - **TBD**: Define all standard observational data sources and identify their processing scripts.
14. Depending on the choice of the model different options are given to the user:
 - MM5: The domain size and number of nodes for this job was determined earlier. Based on both choices, the user is presented with different MM5-executables to choose from. These executables have been compiled in advanced. The MM will be able to retrieve the compile info about the executables, e.g., domain size, number of nodes, number of sigma levels, etc. These few executables are standard executables (**TBD**: determine what 'standard MM5 executable' means). Or, the user can also supply own executable, e.g., its location on the cluster
 - WRF: User defines model options (**TBD**: determine possible model options)
15. User chooses whether or not to run Final Analysis. This may only apply to re-runs and case studies.
16. User chooses whether or not to run Prelim. Analysis. This may only apply to re-runs and case studies.
17. User chooses whether or not run additional processing on the model output. (**TBD**: What exactly are the options for additional processing? Bias Correction?)
18. User can choose to save the model output in MetVault. If 'yes', user must specify what output file is to be sent to the MetVault.
19. User is given the option to save and submit the job now. Submitting now, would run IC/BC-data and obs processing and the model, no post-processing.
20. User chooses whether or not to run post-processing. If 'yes', then s/he will go through the action sequence in 3.4. (**TBD**: How customizable should post-processing be? In GCAT, e.g., the user can specify locations for pseudo-soundings, cross sections, pseudo-obs, etc. Do MM-users at RAL need that level of post-processing customization? Do MM-users at the ranges need that level of post-processing customization, or would be a set pre-configured post-processing options be sufficient?)
21. User can save the above job configuration. Job configurations can be saved to a file.
22. User submits the job.

Use Case "ClimoFDDA": Setting up a ClimoFDDA job

The objective of this feature is to integrate the GCAT functionalities within the MM. Suppose the user is logged on to the system and has made the following choices “Submit a new job” -> “Set up a new model job”. S/he is then presented with two more options: “Weather FDDA” and “Climo”.

Primary actor: Meteorologist, software engineer

Goal: Set up and run a ClimoFDDA job. At this point, the MM only supports “ClimoFDDA”.

Action Sequence:

1. Chooses to “Set up a Climo Job”.
2. User defines a JOBID.
3. User supplies other job parameters. For details, see GCAT tool
4. User determines the domain location.
5. User can specify locations for pseudo-obs and custom cross sections. (TBD: Is this configuration information that is only used during post-processing?)
6. User picks a pre-configured MM5 setup
7. User sets the time line for the job: start time, end time and what years
8. User sets the ensemble options:
 - hourly (TBD: Are there other options? Daily?)
 - min, max, mean, standard deviation,....
 - diurnal cycle
 - typical moment
9. User can request the number of nodes this job should run on.
10. User chooses whether or not run additional processing on the model output. (TBD: Is 10. applicable to ClimoFDDA jobs?)
11. User can choose to save the model output in MetVault. If 'yes', user must specify what output, e.g., member and/or ensemble output is to be send to the MetVault.
12. User is given the option to save and submit the job now. Submitting now, would run the member models, the ensemble, and possible model output processing, no post-processing.
13. User chooses whether or not to run post-processing on the ensemble output. If “yes”, then s/he will be presented with the following options:
 - Plots (NCL or RIP)
 - NAPS
 - MDV
 - Sites
 - MEDOC (1 – 4) (?)
 - Raster
 - PRF
 - Wind Roses
 - ...

For each output product, the user is prompted to supply a destination host and location, where the output files should be copied.

14. The user can also specify whether the members' output should be post-processed. S/he can specify which year-output to post-process and what post-processor(s) to use and where the output files should be copied.
15. The user can specify whether another process (coupled app) should be run on the post-process output products. **This needs further clarification.**
16. User can save the above job configuration. Job configurations can be saved to a file.
17. User submits the job.

Use Case "CAM": Setting up a CAM job

TBD

3.2 Set up and submit a "post-processing" job

The objective of this feature is to provide the ability to only run "post-processing" on an existing model output file. It will also provide the post-processing part of the use case "WeatherFDDA" in 3.1. Suppose the user is logged on to the system and has made the following choice, "Submit a new job".

Primary actor: Meteorologist, software engineer

Goal: Run post-processing on model output

Action Sequence:

1. User selects "Set up a new 'post-processing' job".
2. User is prompted to supply location and name of the model output file:
 - If the model output file will be produced by a running or scheduled FDDA-job, then the user supplies JOBID & cycle time
 - If the model output file already exists, then the user supplies its location.
3. User chooses the type of post-processing:
 - Plots (NCL or RIP)
 - NAPS
 - MDV
 - Sites
 - MEDOC (1 – 4) (?)
 - Stereo
 - Verification
4. For each of the options chosen in 3., the user can supply a custom configuration file (if this is applicable) or use the default configuration file.
5. User must supply destination location for the output products.
6. User can specify the number of nodes this job should run on (**TBD**: Is this a correct requirement?)
7. User can save this job's setting. Job configurations can be saved to a file.

8. User submits job.

3.3 Submit a 'By-hand' job

The objective of this feature is to accommodate the current GMOD-framework. It will also give the user the ability to run customized jobs. In order to submit a custom job to the MM, the user must first identify the cluster(s), where his/her job should run on. Then, log on to this machine, perform operations that are necessary for setting up the job and then register the job with the MM (see Action Sequence below). It is important to note that since a 'by-hand' job wasn't set up through MM's Job-Setup module, the MM doesn't know what the job is actually doing. In order for the MM to accept the job, the user will have to provide certain mandatory information about the job. Suppose the user is logged on to the system and has made the following choice, "Submit a new job".

Primary actor: Meteorologist, software engineer

Goal: Register a custom model job with the MM

Action Sequence:

1. User selects "Submit a 'By Hand' Job"
2. User supplies:
 - Mandatory:
 - job id (GMUAE, GWDPG,...)
 - location of the script (host:/full_path_to_script)
 - time when to run the script
 - estimated time of how long the script will run
 - name(s) of executable(s)
 - max. runtime for the executable(s)
 - number of nodes (??)
 - location of output products (such as: host:/dir_path)
 - Optional:
 - frequency of how often the script should run
 - job type
 - any additional information
3. User can save the job's settings. Job configurations can be saved to a file.
4. User submits job.
5. User can view his/her job in the job queue (see 3.1.).
6. User can receive an email notification when job is started, finished or killed.
(SRS Note: 'custom' job will need to notify the MM when done.)

3.4 Load a job configuration from a file and submit the job

The objective of this feature is to provide the ability to load a job configuration into MM from a file. Suppose the user is logged on to the system and has made the following choice, "Submit a new job".

Primary actor: Meteorologist, software engineer

Action Sequence:

1. User selects "Submit a job configuration file".

2. User supplies file name to load.
3. User can make changes to the configuration.
4. User can save the changed configuration.
5. User submits the job.

3.5 Retrieve and run a previously saved job configuration

The objective of this feature is to provide the ability to retrieve a previous job configuration and to re-run this job or change its settings and run it again.

Primary actor: Meteorologist, software engineer

Goal: Run a previously configured job.

Action Sequence:

1. User logs on with user id/password.
2. User chooses to look at his/her previously saved job configurations. A table of saved jobs may include the following attributes:
 - Job id
 - Job type
 - Cycle time that was run last (if applicable)
 - time this job was run last
 - ...
3. User selects a job. User can change or delete this job configuration.
4. User changes the job configuration.
5. User can save the changed job configuration.
6. User submits job.

3.6 View scheduled, running and old jobs

The objective of this feature is to facilitate monitoring of running jobs, viewing the job queue and viewing jobs that ran in the past.

Primary actor: Meteorologist, software engineer

Goal: Monitor a running job and view scheduled and old jobs

Action Sequence:

1. User logs on with user id/password.
2. User chooses to look at all running jobs, all scheduled jobs (the job queue), past jobs or all jobs (running, scheduled, old). User selects one of the four options.
3. User is presented with a job table. Depending on the user's choice in 2, the job table may present the following attributes to the user:
 - user id – the 'owner' of the job

- Job id
 - Job type – such as: GMOD, climoFDDA, FDDA-re-run, case study, custom,....
 - Job priority (TBD: How should jobs be prioritized?)
 - start time
 - remaining time (estimated) or time it took to run the job
 - cycle (if applicable)
 - stage (Pre-processing, F-analysis, Prelim. Analysis,...)
 - status (SCHEDULED, RUNNING, status in % - if applicable, DONE)
 - cluster and nodes (for a running job)
 - number of processors
 - and others
4. User can select a job and receive more detailed information (TBD: What exactly are the details?)
 5. User can look at a jobs log files.
 6. User can delete his/her jobs from the job queue. A “super user” can delete any job from the job queue.
 7. User can stop his/her running job. A “super user” can stop any running job.
 8. User can re-start his/her job. A “super user” can re-start any job.
 9. User can resume his/her stopped job. A “super user” can resume any stopped job.