

Requirements for User Interface in Data Fitting Service

MAJOR REVISIONS

Inception draft(April 4, 2007)

Paul Kienzle and Wenwu Chen

First draft

SCOPE

Implement a generic user interface for the users to access the services provided by PARK (data fitting service server).

REQUIREMENTS

- Service request management
 - Project management
 - * start and stop the project, add and remove the users of the project
 - * display the associated information of the project, e.g., the name, the starting time, the resource limitation, such as the CPU time, memory resource and storage resource limitation, the users that are working on, and its currently running and finished jobs
 - * display the associated information of the user, e.g., the projects that it can work on and it is working on, the resource limitation, the jobs that it has submitted
 - * specify the dataset for the project
 - * show the usage of the server
 - Job management
 - * Submit the job request
 - * Kill the job that has been submitted
 - * Show the current status of the submitted jobs, such as whether it is running, waiting or finished, the running priority, the names of the user and project that it belongs to, the arrival time, start to running, and the CPU time that has used
 - * Change the status of the job, such as its priority and the resource limitation
 - * Register as the viewer for the specific submitted jobs
 - * Show the details of the job, such as the information about the model, fitting parameters and the constraints
- Data operations
 - Selecting Data
 - * show the available data resources
 - * data may live on local machine
 - * data may live on facility depot
 - * share authentication for access to compute resources and data
 - Viewing Data
 - * show raw data
 - * show meta data
 - * show the application specific data views, e.g. Q or θ as x-axis
 - Transforming Data
 - * exclude bad points from fit

- * limit fit to a range of data
- * use reduction functions specific to application
- * show the reduced data when associated with the predefined dataset and reduction functions
- * transform raw data into a form suitable for fitting
- * transform theory into a form suitable for fitting
- * deal with the resolution
 - fixed resolution ($M(x) = f(p; x) \otimes G(x)$)
 - varying resolution ($M_i(x) = f(p; x) \otimes G_i(x)$)

- Fit View

- user can select a different predefined optimizer
- user can modify optimizer specific control parameters, e.g. population size for GA
- user must specify the fitting parameters, and can specify whether the fitting parameter should be the integer or float-point number
- user can provide the range and the initial values for the fitting parameters
- user can express the dependency among the fitting parameters, such as the symbolic expressions
- user can define the constraints for the fitting parameters, such as the equality and inequality constraints, linear and quadratic constraints, and maybe the non-linear constraints

- File transport

The user needs to cooperate with PARK when transferring large files

- External Interface

The service UI will be implemented in Python, so it can:

- be callable from python
- be callable as a pyre component
- support rich client applications for monitoring fit results (GUI)

The python client API can be used by non-DANSE applications, such as ISAW and DAVE, to communicate with PARK, via a Python bridge, or applications can communicate directly over TCP/IP.

- Deployment

This software is easy to install and manage:

- Support OSX, Unix/Linux, and Windows clients
- The python client APIs are in the package to be used as a library
- The GUI is an executable program, which can download and use in a conventional manner for 3rd party applicaionts on the system.
- Email gateway and AJAX interface may be more difficult

- Opening Topics

- The security issues are not discussed in this requirement, and should be addressed elsewhere; the UI will need to support authentication in some way.
 - The advance features to view the results from the server, such as to generate high-quality and publishable graphics, are not addressed here.
- Licensing
- Please see the Licensing section in "Requirements for Optimization".

FIRST IMPLEMENTATION ITERATION REQUIREMENTS

The University of Maryland will design the service UI framework for DANSE in consultation with the science subprojects.

- Design interface for GUI and script UI.
- Implement the prototype for GUI
- Implement the simple viewers for data results from the server.

RISK MITIGATION PLAN

No significant risk in first implementation iteration.

ITERATION PLANS

- Refine design and implementation until good enough to meet the needs of the DANSE community.
- Generate only as many planning and specification documents as required to communicate needs.