

BornAgain - Feature #1408

Feature # 964 (Rejected): === GUI ===

GUI: refactor/beautify/finalize FitWidget

14 Apr 2016 15:53 - pospelov

Status:	Archived	Start date:	14 Apr 2016
Priority:	Normal	Due date:	
Assignee:	pospelov	% Done:	50%
Category:		Estimated time:	160.00 hours
Target version:	Sprint 31		
Description			
<p>For the moment, underlying functionality (model side) is implemented by David. Fitting views are in prototyping state. This item specifies what still has to be done to include fitting in the release. There will be lots of accompanying refactoring on the way.</p>			
<ul style="list-style-type: none">• create selector widget for an experimental data modify MaskEditor to be able to specify region_of_interest take care of GUI->Python script generation in the case of masked rectangular detector create icon 2 days• refactor JobView and introduce new type of activity: Fit Activity this will require new docks: fit parameters dock, fit progress dock 2 days• create FitParametersSetting widget Implement all logic to create fit parameters out of real time parameters 4 days• create minimizer settings widget 1 day• create RunFit widget 1 day• create FitLog widget 1 day• create FitPresenter widget real data, simulated data and chi2 3 days• Implement transform to/from Core/GUI/Python domains for all fit machinery 4 days• Implement Core/GUI/Python functional test for fitting			

2 days

History

#1 - 15 Apr 2016 16:53 - wuttke

- Parent task set to #964

#2 - 02 May 2016 17:55 - pospelov

- Description updated

- % Done changed from 0 to 30

#3 - 18 May 2016 16:33 - pospelov

- Description updated

- % Done changed from 30 to 50

#4 - 29 Jun 2016 17:33 - pospelov

- Status changed from *Sprint* to *Resolved*

For the moment it's good enough to start showing to the users. Still lots of things has to be done, but it will be a subject of other issues.

Just for history, I put below all subitems I was addressing, or was having in mind during this "beautify/finalize" issue. Total amount of new code for fitting widgets is about 8000k lines with comments.

And thanks a lot to David Li, who has made crucial refactoring of our model structure, has made a few key suggestions about fitting logic, and has created a working fitting prototype.

Fitting Widget - the Plan

- + Fit Parameters widget and fitting logic
- + Import experimental data widget
- + Fit progress widgets
- + Minimizer settings, transformations

- + was done
- was rejected
> postponed

Fit Parameters widget and fitting logic

- > Refactor JobView for new type of activity
- ++ refactor JobView and introduce new type of activity: Fit Activity
- ++ this will require new docks: fit parameters dock, fit progress dock
- ++ Extract JobViewPrivate into separate class
- ++ JobModel: remove signal selectionChanged
- ++ Refactor JobSelector
- ++ Introduce JobViewProgressBar
- ++ Introduce JobViewActions, JobViewFlags
- ++ Refactor JobOutputDataWidget to rely on StackPresenter
- ++ Refactor JobRealTimeWidget to rely on StackPresenter
- ++ Introduce common update, which will work on JobFinished
- Introduce common setActivity, if necessary
- ++ Hide JobQueueData from everybody except JobModel
- ++ Refactor FitSuiteWidget to not to rely on JobPresenter
- ++ Remove one-pixel frame in docks
- ++ Get rid from JobPresenter ?
- Think of not setting JobItem to hidden docks
- ++ Check write/read on disk
- ++ Get rid from onModelLoaded, if necessary
- Refactor JobView so the decision which View to show/hide would be delegated to views itself
- Make JobView remember default positions of splitter
- >> Introduce two types of Actions for JobSelectorWidget (for dark and light backgrounds)

- >> Repair IntensityDataItem equalization
- >> Make F1, F2, F3 hot keys for activities.

- > Fitting logic
 - ++ Make SelectRealData combo remember previous state
 - ++ Make FitActivity respect whether JobItem is suitable for fitting (do not create an empty widget)
 - ++ Make SimulationSettingAssistant which will make sure that RealData is suitable for JobItem
 - + Introduce JobItemAssistant which will create all necessary containers for JobItem with fit
 - Repair JobItem::P_WITH_FITTING
 - ++ block ParameterTuningWidget on start fitting
 - ++ Block drag-and-drop when there is a fitting running
 - ++ Introduce fitting activities and match drag/drop allowance in ParamTuningWidget and FitParameterWidget

- > FitActivityPanel
 - ++ Refactor to rely on ItemStackedWidget
 - ++ connect slider
 - ++ place label with number of iterations
 - ++ Change initial size of activity panel
 - ++ Smaller sizes of RunFit button
 - Make FitControlWidget a sunken panel
 - + Make RealTime activity remember column sizes in QTreeView
 - ++ Fix geometry of just-opened-panel.

- > Fit Parameter Model
 - Repair fit parameter model, QAbstractProxyModel
 - ++ Make design decision: SessionModel, AbstractModel or ProxyModel ?
 - ++ It was SessionModel: Redefine index function, make selection working
 - ++ onAddRow, onRemoveRow, dataChanged (proxy <—> source)
 - ++ modelTuningDelegate for FitParameterModel
 - ++ internal dragAndDrop
 - + cross-selection
 - ++ Make tooltips for headers of FitParameterItem's tree

- > FitParameterItem and its FitParametersWidget
 - ++ Invoke delete button for fit parameters removal
 - ++ Remove fit parameter link
 - ++ Creation of new fit parameter should respect limits
 - >> Introduce current value of fit parameter
 - >> Provide update of current value on iteration change
 - >> Introduce error/step of fit parameter
 - ++ Implement remove FitParameterItem on right mouse button
 - + Add special delegate for AttLimits
 - + Implement Fixed status and corresponding check box
 - >> Make a FitParametersItem tree view remember columns positions
 - ++ Implement dragAndDrop from modelTuningWidget
 - ++ Create TuningModelProxy FilterPropertyProxy
 - >> Propagate FitParameterItem links status to ParameterItem's icon in ParameterTuningWidget

- > Fit Failure
 - ++ Add WarningSignWidget to "Run fit Console"
 - >> Provide WarningSignWidget with animation to stress it's reappearance
 - >> WarningSignController which will extract logic out of WarningSignWidget
 - >> Provide possibility to change WarningSignWidget header
 - >> Add TextEditor to WarningSignWidget for long error logs
 - >> When switch to another job (without fitting support) WarningSign should disappear
 - ++ Propagate failure message to JobMessagePanel

-
- Import experimental data widget
-

- > > ImportDataView, RealDataEditor
 - ++ create selector widget for an experimental data
 - ++ create set of icons for selector widget
 - ++ create main icon for ImportDataView
 - ++ Implement AppSvc with common access to ProjectDocument
 - ++ Provide load of data file
 - Implement DocumentItem which will remember all general session settings
 - ++ Refactor ProjectDocument: move save/load on board of application models
 - ++ Provide saving of RealDataModel together with all intensity data
 - ++ Make fitting running using the data from RealDataModel
 - ++ BUG: can't import *.gz file under the MAC
 - ++ IMPORTANT: Make one pixel splitter
 - >> How to make axes adjustment to the instrument
 - >> Provide right mouse button context menu for SelectorWidget

- >> Provide pictogram generation for intensitydataview and RealDataItem's
- >> Refactor InstrumentView and ImportDataView to have same structure
- >> Repair mask editor
- ++ Provide tooltips for SampleView
- >> ApplicationModel and SessionModel: move methods related to resetModel on board of corresponding model
- ++ Implement ColorMapCanvas into ImportDataView widget
- ++ Make automatic ! propagation of axes from instrument to ImportDataView

Fit progress and results widgets

- > Fit Progress
- ++ Introduce FITTING status for JobItem
- Block global progress bar, add additional sign for "Fit Running"
- Block local progress bar, introduce special status/color for Fit is Running

- > Fit Results Widget
- ++ Fit Log Console / Job Message Service
- ++ Make update frequency of plots/log/fit parameter consistent
- Make special printing functions in RunFitManager
- ++ Change of ownership of the data flying.
- ++ Add color of text to addMessage
- ++ Provide the bar with closing button similar to SampleView
- >> Tune default size, borders
- >> Where to present final fit parameter values, errors
- >> Fitting status (converged)
- >> Make corresponding tab lighting on fit completed

- > ColorMap's for fitting data
- ++ Implement ItemComboWidget, ItemComboToolBar
- ++ Refactor JobOutputDataWidget to rely on new ComboWidget
- ++ Implement FitComparison Widget
- ++ Provide GUI way of calculation of relative difference map
- ++ Validate if relativeDifferenceMap is correct: call of plot on last iteration?
- ++ remove unnecessary signal chiSquaredMap

- > ColorMapPlot speedup
- ++ replace ColorMapPlot::plotItem -> getAxesBinIndices
- ++ Introduce initItem which will take care about settings of CustomPlot wrt IntensityDataItem
- ++ Make only data plotting inside ColorMapPlot::plotItem
- ++ Make setting of zero item to disable everything
- ++ setItem second time should not lead to replote of item
- ++ Introduce UpdateTimer to schedule replete of ColorMapPlot
- ++ Implement ColorMap, future replacement of ColorMapPlot.
- ++ Implement StatusLabel with respectance to external layout
- ++ Implement ColorMapCanvas with decoupled status bar
- ++ Implement ColorMapCanvasEvent with auto font adjustment
- Provide enable/disable axes label
- ++ Fix bug in CustomPlot with non-updating bottom margin
- ++ Refactor ColorMapPlot for projections: cross-on-mouse should not be the part of ColorMapPlot
- Provide ColorMap plot title (too ugly, whatever way I choose (CustomPlot or ColorMapCanvas)
- ++ Provide back up of axes labels in the case of FitComparisonWidget
- ++ Provide Size of the QCustomPlot font adjustable wrt the size of widget (ColorMapCanvasEvent)
- >> ColorMapCanvasActions: provide right mouse button context menu: SavePlot, RsetView, PropertyDialog
- >> Create ColorMapPropertyDialog

- > Fighting for performance
- ++ make component editor remember top item, to not to replot !!!
- >> JobView::onFocusRequest -> Do I need to reselect already selected item?
- >> Probably not, but wait for ComponentEditor fix
- ++ QtTreePropertyBrowserPrivate::updateItem setfirstcolumnspanned IMPORTANT Commented
- >> Implement SessionItemWidget + facility to deactivate when it is not shown
- ++ BUG: when parameter values is unphysical - all widgets disappears
- >> Implement ColorMapCanvas in JobView
- >> Make IntensityDataWidget separate entry on JobPresenter
- ++ Make JobItem setJobResults work on setRawBuffer

- > Some visual niceties
- ++ Add reset button to FitComparisonWidget
- Add synchronous zoom
- Add synchronous colormap scale
- ++ IMPORTANT MACOS: make zero bound of MinimizerSettingsWidget
- >> Make width of column of ComponentEditor equal

Minimizer settings, transformations - 2 days

- > Minimiser Settings Widget
- ++ Implement global minimiser setting
- ++ Implement Minuit2/Migrad Minimizer
- ++ Implement Genetic minimiser
- >> Make equal width for ComponentEditor
- >> Refactor Core minimisers in the part of Minimizer/Algorithms splitting
- ++ Refactor MinimizerFactory + Catalogue
- ++ Refactor MinimizerItemCatalogue to serve as a translator
- >> Refactor the way all options are set to minimizers

- > FitParameterLink
- >> Implement possibility to add several fit patterns to single FitParameterLink
- >> Make FitParameterItem generate corresponding FitParameterLink

Overall improvements

- + Implement permanent SessionModelView
- > Generalise Splitter logic everywhere: auto adjustment on first show
- > ComponentEditor::onDataChanged should not be invoked when it is hidden
- > Change initial width of widgets (position of splitter) so they would look nice under MacOS
- > Revise width of all styled toolbars. Why they are different?

- > Take care of GUI std::out warnings
- QColor::setRgb: RGB parameters out of range
- QWidget::setMinimumSize: (dockWidget5/QDockWidget) Negative sizes (0,-1) are not possible
- QPainter::begin: Paint device returned engine == 0, type: 2
- QPainter::begin: Paint device returned engine == 0, type: 2
- QPainter::begin: Paint device returned engine == 0, type: 2
- qt.network.ssl: Error receiving trust for a CA certificate
- bool QCPLLayoutGrid::addElement(int, int, QCPLLayoutElement *) There is already an element in the specified row/column: 0 1

- + Refactor NodeEditor/PortConnections
- > Restore icons of CoreShellParticle, ParticleComposition and DistributedParticle
- + What about blue frame which suppose to emphasise currently active widget?
- > Implement metric for real time view
- > Icons of IParticleView doesn't update immediately when type of the particle is changed

- > Exception thrown in Prism3 doesn't contain explanation, what is going wrong

- > Refactor DetectorItem -> DetectorContainerItem
- > SphericalDetectorItem and RectangularDetectorItem should be derived from common DetectorItem
- > MaskContainer should be moved to from DetectorContainerItem to Spherical/Rectangular

- > take care of GUI->Python script generation in the case of masked rectangular detector
- > for the moment masks are correctly processed only for spherical detector. In the case of rectangular masks will be still in degrees instead of mm.
- > modify MaskEditor to be able to specify region_of_interest

#5 - 04 Jul 2016 11:23 - herck

- Status changed from Resolved to Archived