

Aufgaben

Pascal Huber

April 23, 2014

Contents

1	TODO Create new branch	2
2	TODO Determine which files have to be modified	2
2.1	Files for Potential-tab	2
2.2	Files for Datafile-tab	2
2.3	Description of parametertextedit.h and parametertextedit.cpp	2
2.3.1	Description of textEditStatusBar class	3
2.3.2	Description of parameterTextEdit class	3

- Note taken on *2014-04-16 Wed 15:57*
Besprechung mit Christian um *2014-04-16 Wed 15:57*

Zwei Aufgabenziele:

1. Kurzfristiges Ziel: Überarbeitung des “potentials” tabs
 - Übernahme Elemente von “Datafile” tab
 - Obere 2/3 des Tabs: Editor-Fenster in das ein Potentials-file geladen werden soll
 - * der Editor soll die Funktionen “open” and “save” bereitstellen
 - * open-button:
 - soll erweiterbar sein (d.h. es sollen noch mögliche “Suchfunktionen” eingebunden werden können -> handler)
 - vorerst: einen beliebigen Ordner öffnen
 - nach .potential files filtern

- nachdem der Nutzer ein file ausgewählt hat, soll eine Kopie des files in den Projekt-Ordner gelegt werden mit dem Namen "PROJECTNAME.potentials"
- falls nach Sicherung eines .potential-files ein weiteres file geöffnet wird, soll eine Warnung zum "Datenverlust" abgegeben werden.
- * save-button:
 - speichert die Änderungen im .potentials file.
- Stelle sicher, dass immer ein .validates-file erzeugt wird (notfalls auch leer (default)).
- 2. Langfristiges Ziel:
 - Verifizieren von Ensembles
 - etwas mit Monte-Carlo Methoden (siehe folgende Links)
 - * Combining molecular dynamics and Monte Carlo methods
 - * Uniform-acceptance force-bias Monte Carlo method

1 TODO Create new branch

2 TODO Determine which files have to be modified

2.1 Files for Potential-tab

potential.cpp potential.h potentialparameter_data.cpp potentialparameter_data.h
 potentialparametertable.cpp potentialparametertable.h potentialparameterui.cpp
 potentialparameterui.h potentialparameteruisuper.cpp potentialparameteruisuper.h
 potentialparameterwidget.cpp potentialparameterwidget.h

2.2 Files for Datafile-tab

parametertextedit.h parametertextedit.cpp

2.3 Description of parametertextedit.h and parametertextedit.cpp

- Two classes
 1. class textEditStatusBar : public QWidget
 2. class parameterTextEdit : public QMainWindow

2.3.1 Description of `textEditStatusBar` class

Implements the status bar in the “Datafile”-tab (just at the lower end of the window).

The status bar consists of 4 different frames (`QFrame`) containing each a `QLabel` with

- the current position of the cursor
- a indicator that shows if the file has been changed
- the type of the file
- the name of the file

The layout is managed by a `QHBoxLayout` for the statusbar itself and for different `GridLayouts` for each of the frames.

The following slots are implemented:

- `changeCurrentPos`
- `setFilename`
- `setEditMode`
- `languageChange`

2.3.2 Description of `parameterTextEdit` class

Implements a texteditor for datafiles in the “Datafile”-tab. It is essentially a `QMainWindow` with a `QTextEdit`. Moreover the class has a `textEditStatusBar` as child object.

The class contains the following children:

- `QTextEdit` for the texteditor window
- `QToolBar` a toolbar for the texteditor
- `QActions` for all buttons on the toolbar
- `QStrings` to save the parameter file and a suffix
- `textEditStatusBar`
- bools for different indications

The layout is managed by only one `QVBoxLayout`.
There are only a few method implemented in the class:

- `bool isSaved() const` // returns true if the file is saved
- `void setSavedStatus(bool status)` // sets the status of the file
- `void fileLoad(QString File)` // loads the file that is supposed to be edited
- `void renewTextEdit2()` // ?

Slots:

- `void fileSave()` // saves the file
- `void reloadfile(QString parameterFileIn)` // reload the file
- `void clearBuffer(bool ask=true)` // ?
- `void setLoadingfileStatus(bool status)` // ?
- `void enableFileSaveAction(void)` // ?
- `void changeCurrentPos (void)` // ?
- `void fileOpen()` // open new file
- `fileSaveAs()` // save file as
- `textEditChanged` // action if textEdit has been changed
- `virtual void languageChange()` // set language

Signals:

- `void newSaveState()` // is send if the save-state of the file changes