Design Document 1.0

Group Mavis

Helsinki 3.5.2006
Software Engineering Project
UNIVERSITY OF HELSINKI
Department of Computer Science

Course

581260 Software Engineering Project (6 cr)

Project Group

Kimmo Holm

Juho Julkunen

Rami Järvinen

Janne Laukkarinen

Joel Linden

Jan Wagner

Client

Fabio Donadini

Tomas Kohout

Project Masters

Juha Taina

Joni Salmi

Project Home Page

http://www.cs.helsinki.fi/group/mavis/

Table of Contents

1 Introduction

This document describes planned architecture for the analyzing software that will be implemented as a software engineering student project at University of Helsinki, Department of Computer Science. The clients are Lauri Pesonen with his assistants Fabio Donadini and Tomas Kohout from Department of Geophysics.

The document serves as an internal guide to the project team for aiding implementation phase, and describes the software at about level of accuracy which allows to implement the software based on this document and requirements document.

1.1 Structure of the document

Section 1 (this section) describes the meaning and structure of the document.

Section 2 describes coding conventions used by the project group in this software.

Section 3 describes the software and architecture at high abstraction.

Section 4 describes planned architecture at lower abstraction, including class diagrams and a short description of each subsystem.

2 Code Conventions

We will follow the Code Conventions for the Java Programming Language set by Sun, and the code conventions used by the previous group.

3 Overview of the system

The Mavis analysis software facilitates displaying of demagnetization results in visual form and isolation of separate magnetic components from sample data. This is all done through a graphical user interface.

The software is split into two sections in this document. The first section describes the classes and interfaces used for project-management and data flow in the software. The second section documents all the graphical user interface classes.

4 Architecture Description

In this section we describe each subsystem shortly, and present a class diagram of each, as well as the whole system divided roughly into data and GUI parts.

4.1 Data Classes and interfaces:

Data classes are in packages mavis and mavis.util.

4.1.1 Project

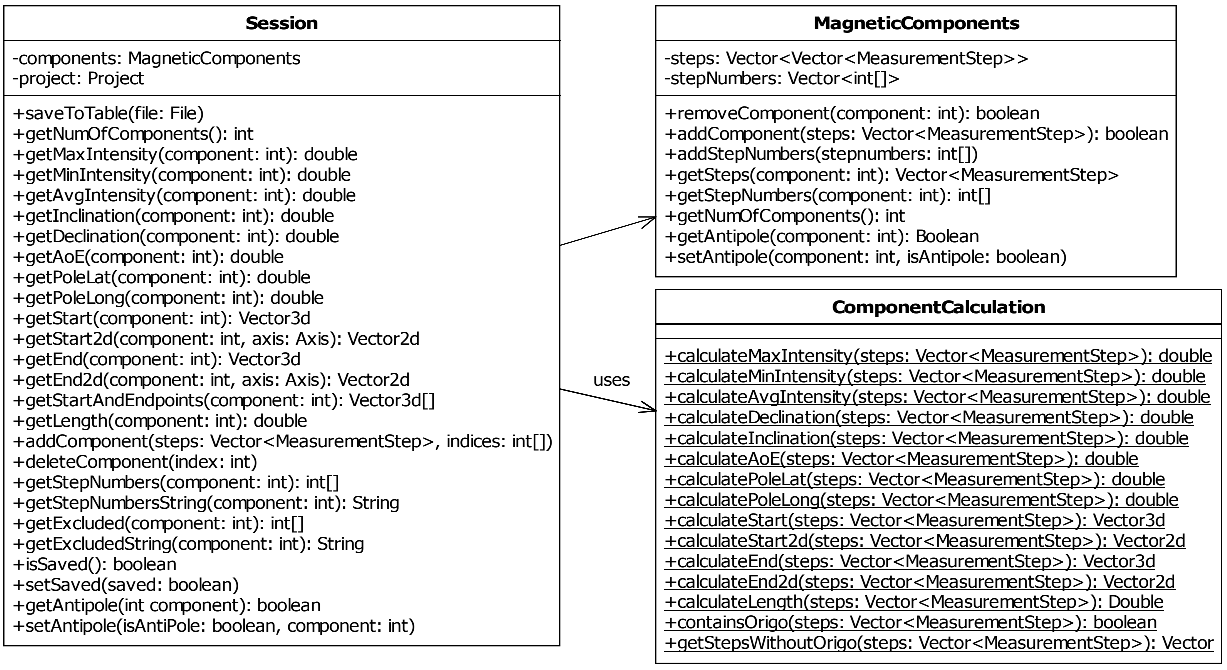
Ikayaki's Project class, but many methods that aren't used in Mavis may have to be stripped.

4.1.2 MeasurementSequence, MeasurementStep, MeasurementResult

From Ikayaki with minor modifications.

4.1.3 MeasurementValue

Has new methods for calculating difference vectors and values for the ZijderveldPlot.


Session
-components: MagneticComponents
-project: Project
+saveToTable(file: File)
+getNumOfComponents(): int
+getMaxIntensity(component: int): double
+getMinIntensity(component: int): double
+getAvgIntensity(component: int): double
+getInclination(component: int): double
+getDeclination(component: int): double
+getAoE(component: int): double
+getPoleLat(component: int): double
+getPoleLong(component: int): double
+getStart(component: int): Vector3d
+getStart2d(component: int, axis: Axis): Vector2d
+getEnd(component: int): Vector3d
+getEnd2d(component: int, axis: Axis): Vector2d
+getStartAndEndpoints(component: int): Vector3d[]
+getLength(component: int): double
+addComponent(steps: Vector<MeasurementStep>, indices: int[])
+deleteComponent(index: int)
+getStepNumbers(component: int): int[]
+getStepNumbersString(component: int): String
+getExcluded(component: int): int[]
+getExcludedString(component: int): String
+isSaved(): boolean
+setSaved(saved: boolean)
+getAntipole(int component): boolean
+setAntipole(isAntiPole: boolean, component: int)
uses
MagneticComponents
-steps: Vector<Vector<MeasurementStep>>
-stepNumbers: Vector<int[]>
+removeComponent(component: int): boolean
+addComponent(steps: Vector<MeasurementStep>): boolean
+addStepNumbers(stepnumbers: int[])
+getSteps(component: int): Vector<MeasurementStep>
+getStepNumbers(component: int): int[]
+getNumOfComponents(): int
+getAntipole(component: int): Boolean
+setAntipole(component: int, isAntipole: boolean)
ComponentCalculation
+calculateMaxIntensity(steps: Vector<MeasurementStep>): double
+calculateMinIntensity(steps: Vector<MeasurementStep>): double
+calculateAvgIntensity(steps: Vector<MeasurementStep>): double
+calculateDeclination(steps: Vector<MeasurementStep>): double
+calculateInclination(steps: Vector<MeasurementStep>): double
+calculateAoE(steps: Vector<MeasurementStep>): double
+calculatePoleLat(steps: Vector<MeasurementStep>): double
+calculatePoleLong(steps: Vector<MeasurementStep>): double
+calculateStart(steps: Vector<MeasurementStep>): Vector3d
+calculateStart2d(steps: Vector<MeasurementStep>): Vector2d
+calculateEnd(steps: Vector<MeasurementStep>): Vector3d
+calculateEnd2d(steps: Vector<MeasurementStep>): Vector2d
+calculateLength(steps: Vector<MeasurementStep>): Double
+containsOrigo(steps: Vector<MeasurementStep>): boolean
+getStepsWithoutOrigo(steps: Vector<MeasurementStep>): Vector

4.1.4 MagneticComponents

A new class used to store the steps of magnetic components.

4.1.5 ComponentCalculation

A new class used to calculate the properties of magnetic components.

4.1.6 Session

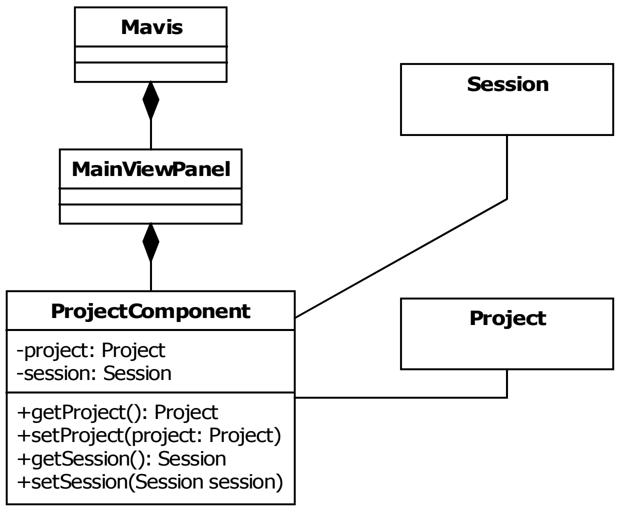
A new class that stores session information, such as opened project and found components.

4.1.7 Settings

Holds the file browser directory history and saves it on the hard drive in a history file.

4.2 GUI classes and interfaces:

GUI classes are in the package mavis.gui.



4.2.1 Mavis

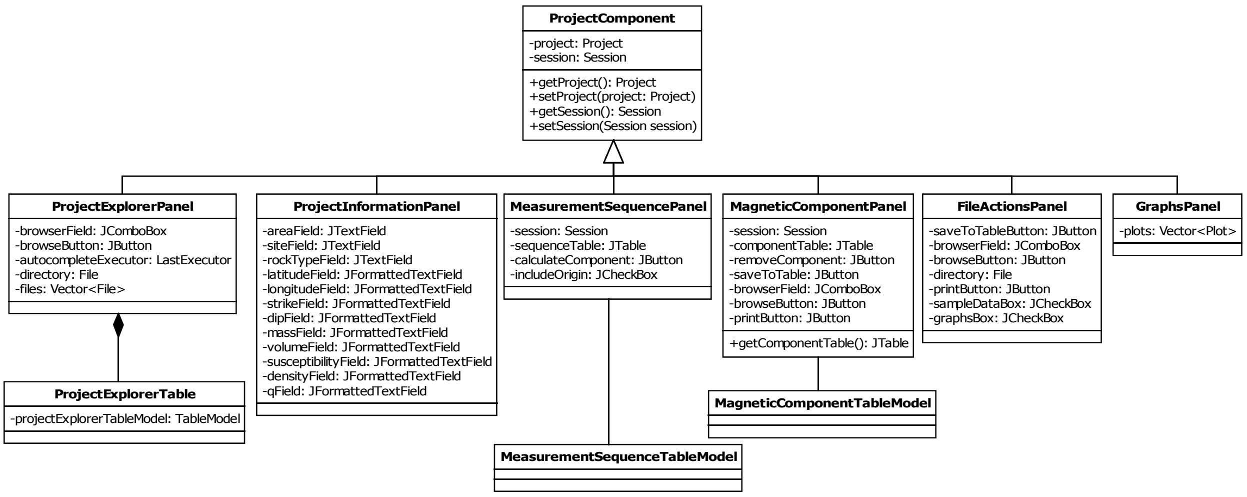
The Main program.

4.2.2 DataPrintPanel

Creates layout from measurement values to be printed. PrintedPanel is preview of print and there are controls to print or cancel. Derived from Ikayaki's PrintPanel class.

4.2.3 GraphPrintPanel

Creates layout from Plots to be printed. PrintedPanel is preview of print and there are controls to print or cancel. Derived from Ikayaki's PrintPanel class.


ProjectComponent
-project: Project
-session: Session
+getProject(): Project
+setProject(project: Project)
+getSession(): Session
+setSession(Session session)
ProjectExplorerPanel
-browserField: JComboBox
-browseButton: JButton
-autocompleteExecutor: LastExecutor
-directory: File
-files: Vector<File>
ProjectExplorerTable
-projectExplorerTableModel: TableModel
ProjectInformationPanel
-areaField: JTextField
-siteField: JTextField
-rockTypeField: JTextField
-latitudeField: JFormattedTextField
-longitudeField: JFormattedTextField
-strikeField: JFormattedTextField
-dipField: JFormattedTextField
-massField: JFormattedTextField
-volumeField: JFormattedTextField
-susceptibilityField: JFormattedTextField
-densityField: JFormattedTextField
-qField: JFormattedTextField
MeasurementSequencePanel
-session: Session
-sequenceTable: JTable
-calculateComponent: JButton
-includeOrigin: JCheckBox
MeasurementSequenceTableModel
MagneticComponentPanel
-session: Session
-componentTable: JTable
-removeComponent: JButton
-saveToTable: JButton
-browserField: JComboBox
-browseButton: JButton
-printButton: JButton
+getComponentTable(): JTable
MagneticComponentTableModel
FileActionsPanel
-saveToTableButton: JButton
-browserField: JComboBox
-browseButton: JButton
-directory: File
-printButton: JButton
-sampleDataBox: JCheckBox
-graphsBox: JCheckBox
GraphsPanel
-plots: Vector<Plot>

4.2.4 ProjectComponent

Almost the same as the one in Ikayaki, but also includes the session field, as well as getters and setters for this field.

4.2.5 ProjectExplorerPanel, ProjectExplorerTable

Similar to the ones in Ikayaki, but without the components/methods used to create new projects.

4.2.6 ProjectInformationPanel

Contains editable text fields for: Specimen title, Site, Latitude, Longitude, Strike, Dip, Volume, Mass and Rock type (+ susceptibility in room temp). Also contains uneditable text fields for Density and Q. Density is calculated based on volume and mass, Q is calculated based on NRM and susceptibility. Density and Q are updated automatically. Changes made to the editable properties are saved in the .ika project file.

4.2.7 MeasurementSequencePanel, MeasurementSequenceTableModel

Contains the sequence table similar to the one in Ikayaki, but not the controls to create new sequence steps. Deleting steps and adding new steps should not be possible. Added in the sequenceTable is an extra step that represents the origin.

Also contains a button to start the component calculation for the selected steps.

4.2.8 MagneticComponentPanel, MagneticComponentTableModel

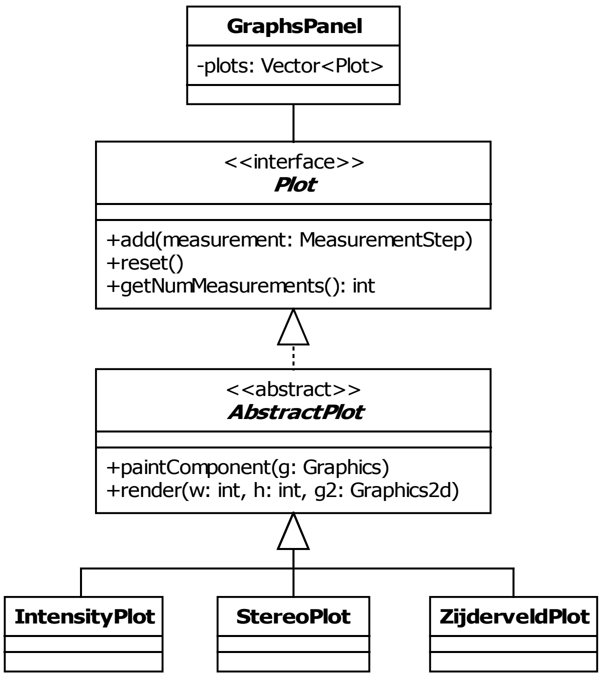
Contains a component table, which lists all the found magnetic components and their properties. The properties are: Steps, (Intensity) Range, Declination, Inclination, Angle of error, Median Intensity, Longitude of pole and Latitude of pole.

A button to remove a selected component is also included.

4.2.9 FileActionsPanel

There is a drop down combobox to select a file to save to and a button to save the components to the selected file as well as a "Save as"-button to open a JFileChooser-dialog that can be used to save the file.

There are also checkboxes to select whether to print the measurement data, the graphs or both and a print button, that prints the selected data as well as a print preview button.


GraphsPanel
-plots: Vector<Plot>
<<interface>>
Plot
+add(measurement: MeasurementStep)
+reset()
+getNumMeasurements(): int
<<abstract>>
AbstractPlot
+paintComponent(g: Graphics)
+render(w: int, h: int, g2: Graphics2d)
IntensityPlot
StereoPlot
ZijderveldPlot

4.2.10 GraphsPanel

Is the same class as the one in Ikayaki.

4.2.11 Plot, AbstractPlot, IntensityPlot, StereoPlot

No notable changes to these classes.

4.2.12 ZijderveldPlot

Draws plots based on declination, inclination or both. Magnetic components are drawn in the Zijderveld plot. Also supports zooming in on parts of the plot and returning to the normal zoom level.