Requirements document

SQUID

Helsinki 22nd February 2005

Software Engineering Project

UNIVERSITY OF HELSINKI
Department of Computer Science

Course

581260 Software Engineering Project (6 cr)

Project Group

Mikko Jormalainen
Samuli Kaipiainen
Aki Korpua
Esko Luontola
Aki Sysmäläinen

Client

Lauri J. Pesonen

Project Masters

Juha Taina
Jenni Valorinta

Homepage

`http://www.cs.helsinki.fi/group/squid/`

Change Log

Version	Date	Modifications
0.1	9.2.2005	First version (Aki Sysmäläinen)

Contents

1 Introduction 1

1.1 Glossary 1

2 Overview 1

3 Use cases 1

3.1 Measuring 1

3.2 File formats 3

3.3 Functionality 4

3.4 AF sequences 4

4 User requirements 5

4.1 Functional requirements 5

4.2 Quality requirements 7

4.3 Environment 7

4.4 Maintainability 7

4.5 Restrictions 7

5 System requirements / Functions 7

5.1 System restrictions 7

6 User interface 7

6.1 Goal derived use cases 8

7 Architecture overview 8

8 External interfaces 8

8.1 Hardware control protoocls 8

8.1.1 Handler control protocol 8

8.1.2 Measurement reading protocol 8

8.1.3 Demagnetizer control protocol 8

9 Validation 8

1 Introduction

This document describes client requirements and system requirements for a SQUID magnetometer program that will be designed and implemeted as a software engineering student project at University of Helsinki at the Computer Science Department. The client is the Department of Geophysics.

This document serves as a contract between client and us..

Expected readership of this document here..

1.1 Glossary

Technical terms here..

2 Overview

A brief overview of the problem domain..

3 Use cases

Describes planned use cases for the program. Derived from user interface prototype and requiremets. All use cases are made by "the user" in program main screen, unless otherwise noted.

3.1 Measuring

Do single step measuring without demagnetization

Enter as next AF demagnetization step "0" or empty (default for new projects), meaning no demagnetization, and click "Single step".

Precondition: Open AF project, sample in sample holder.
Postcondition: Sample measured, results on screen.
Error condition: The program shall let the user know if something went wrong.

Do single step measuring with demagnetization

Enter as next AF demag step anything greater than zero, and click "Single step".

Precondition: Open AF project, sample in sample holder.
Postcondition: Sample demagnetized (possibly ruined) and measured, results on screen.

Error condition: The program shall let the user know if something went wrong, and, should the demagnetization field not be coming down, warn user with an alarm sound :)

Do automatic demagnetization-measuring sequence

Enter the AF sequence (see 3.4 for ways to enter it) and click "Measure".

Precondition: Open AF project, sample in sample holder.
Postcondition: Sample demagnetized according to entered AF sequence (possibly ruined) and measured after each demagnetization, results on screen.
Error condition: The program shall let the user know if something went wrong, and, should the demagnetization field be uncalm, warn user with an alarm sound x)

- Pause automatic measuring sequence

  While measure sequence is running, click "Pause".

  Precondition: Ongoing measure sequence.
  Postcondition: Measure sequence halts after current step is done, results on screen.
  Error condition: Program tells if sequence can't be paused (and something has gone terribly wrong).

- Abort automatic measuring sequence

  While measure sequence is running or paused, click "Stop immediately".

  Precondition: Ongoing or paused measure sequence.
  Postcondition: Measure sequence halts immediately [and program enters "fully manual" mode?]
  Error condition: Program tells if sequence can't be aborted (and something has gone terribly wrong).

Do thellier measuring

Click "Single step". (Temperature can be entered later, as it won't affect measuring.)

Precondition: Open TH project, sample in sample holder.
Postcondition: Sample measured, results on screen.
Error condition: As usual.

Do thermal measuring

[Exactly the same as thellier?]

Click "Single step". (Temperature can be entered later, as it won't affect measuring.)

Precondition: Open TH project, sample in sample holder.
Postcondition: Sample measured, results on screen.
Error condition: As usual.

Measure magnetometer ground noise

Click "Noise" and "Calibrate".

Precondition: None.
Postcondition: [Sample holder at home?] Ground noise measured, results on screen.
Error condition: As usual.

Measure empty sample holder noise

[2005-02-21 In current UI prototype, "Noise" actually does sample holder noise measuring.]

Click "Holder noise" and "Calibrate".

Precondition: Empty sample holder.
Postcondition: [Sample holder at home?] Holder noise measured, results on screen.
Error condition: As usual.

Fully manual measuring

Click any of the manual control components [2005-02-21 at right third of UI proto].

Precondition: None.
Postcondition: Manual action done, result on screen.
Error condition: As usual.

- Move sample handler to desired position
- Rotate sample handler to desired angle
- Measure in current position
- Demagnetize in current position

3.2 File formats

Automatically save all measurement cycles in project (.dat?) file

Save standard sample measurement results in .std file

Export (thellier) results into .tdt file

Export single measurement details into .srm file

Print measurement results

Print graph sheet (with 7 different graphs; described elsewhere)

3.3 Functionality

Create new project (.dat file?)

Load project (.dat file?)

Append measurement results to project (.dat file?)

Panic abort operation instantly

When any measuring action, click "Stop immediately".

Precondition: Single step measuring or ongoing measure sequence.
Postcondition: All demagnetization and measuring halts immediately [and program enters "fully manual" mode?]
Error condition: Program tells if measuring can’t be aborted, meaning something has gone terribly wrong.

3.4 AF sequences

As in automatic demagnetization-measuring sequences, or Alternating Field sequences

Insert AF sequence with start-step-stop values

Load AF sequence

Save AF sequence

Edit AF sequence on-the-fly

Edit stored AF sequences

Rename stored AF sequence

Delete stored AF sequence

4 User requirements

Goals of the software set by client..

4.1 Functional requirements

Identifier: R1
Name: Basic
Description: Able to control Squid-magnetometer and make measurements with it.
Priority: 1

Identifier: R2
Name: Saving
Description: Measurements can be saved in .dat, .dtd and .srm files.
Priority: 1

Identifier: R3
Name: Auto saving
Description: Program will save information after every measurement.
Priority: 1

Identifier: R4
Name: Loading
Description: Saved information can be loaded into program.
Priority: 2

Identifier: R5
Name: Filemanagement
Description: New data can be added to existing datafiles.
Priority: 2

Identifier: R6
Name: Numeric presentation of data
Description: Program shows measurement data in numbers.

Priority: 1

Identifier: R7
Name: Graphic presentation of data
Description: Program draws graphs from measurement data.
Priority: 3

Identifier: R8
Name: Editing
Description: Ability edit data afterwards.
Priority: 2

Identifier: R9
Name: Recalculation
Description: Recalculate based on changed data.
Priority: 2

Identifier: R10
Name: Control
Description: Ability to stop any action immediately.
Priority: 1

Identifier: R11
Name: Sequence
Description: Able to create measuring sequences with several different sized steps.
Priority: 2

Identifier: R12
Name: Sequence control
Description: Able to stop sequence after current step.
Priority: 2

Identifier: R13
Name: Hotkeys
Description: Possibility to create and change hotkeys.
Priority: 4

Identifier: RXX
Name:
Description:
Priority:

4.2 Quality requirements

Identifier: QR1
Name: Ease of use
Description: Program should be easy to use for first time users.
Priority: 1

Identifier: QR2
Name: Help pages
Description: Program should have good help pages.
Priority: 3

4.3 Environment

4.4 Maintainability

4.5 Restrictions

Program will be used in normal PC which is connected to magnetometer. Taking into account rapid phase of computer evolution it is possible that computer in which program is used can change, accordingly the program should be able to be installed by outsiders. We need not prepare to changing of magnetometer, as new magnenetometer will probably have its own program.

5 System requirements / Functions

Specific explanation of the functions to be implemented

5.1 System restrictions

6 User interface

Overview of UI described here..

6.1 Goal derived use cases

7 Architecture overview

8 External interfaces

Interface to existing software and use of it described here..

8.1 Hardware control protoocls

8.1.1 Handler control protocol

8.1.2 Measurement reading protocol

8.1.3 Demagnetizer control protocol

9 Validation

Description of how to validate the set requirements.