Requirements document

SQUID

Helsinki 24th 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

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 user requiremets. All use cases are made by "the user" in program main screen, unless otherwise noted.

Format for uce cases:

UC0: Uce case number and title
Scenario a: First scenario for doing the use case
Scenario b: Second scenario for doing the use case
Precondition: Preconditions for use case
Postcondition: Postconditions for use case
Error condition: Error handling, mainly if anything special needs to be done

3.1 Measuring

UC1: Do single step measuring without demagnetization
Scenario a: 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.

UC2: Do single step measuring with demagnetization
Scenario a: 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 :)

UC3: Do automatic demagnetization-measuring sequence
Scenario a: 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)

UC4: Pause automatic measuring sequence
Scenario a: 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).

UC5: Abort automatic measuring sequence
Scenario a: 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).

UC6: Do thellier measuring
Scenario a: 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.

UC7: Do thermal measuring
[Exactly the same as thellier?]
Scenario a: 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.

UC8: Measure magnetometer ground noise
[2005-02-23 not in current UI proto, nor planned for implementation]
Scenario a: Click "Ground noise" and "Calibrate".
Precondition: None.
Postcondition: Ground noise measured, results on screen.

UC9: Measure empty sample holder noise
[2005-02-21 In current UI prototype, "Noise" actually does sample holder noise measuring.]
Scenario a: Click "Holder noise" and "Calibrate".
Precondition: Empty sample holder.
Postcondition: Holder noise measured, results on screen.

UC10: Fully manual measuring
- Demagnetize X, Y or Z axis
- item Measure X, Y or Z axis

Scenario a: Click any of the manual control components [2005-02-21 at right third of UI proto].
Precondition: Manual mode enabled.
Postcondition: Manual action done, result on screen.

UC11: Enable manual mode
Scenario a: Click "Manual" checkbox above the manual control components.
Precondition: No ongoing measurement.
Postcondition: Manual mode enabled.

3.2 File formats

UC12: Automatically save all measurement cycles in project [.dat?] file
Scenario a: Make any measurement action.
Precondition: Open project file.
Postcondition: After measurement is done, new results appended to project file.

UC13: Save standard sample measurement results in .std file
Scenario a: Click "Standard sample", "Calibrate".
Precondition: No ongoing measurement.

Postcondition: After standard sample measurement is done, new results appended to pre-defined .std file.

UC14: Export project data into .dat file
Scenario a: In project explorer file list, right-click on desired project file, click "Export .dat in current directory".
Scenario b: In project explorer file list, right-click on desired project file, click "Export .dat to disk drive A:".
Scenario c: In project explorer file list, right-click on desired project file, click "Export .dat...", chooce directory and filename to export.
Precondition: At least 1 project file in current (selected) directory.
Postcondition: Project data exported to .dat file.
Error condition: Notify if file error occurs (such as no disk in A: drive).

UC15: Export (thellier) project data into .tdt file
Scenario a: In project explorer file list, right-click on desired project file, click "Export .tdt in current directory".
Scenario b: In project explorer file list, right-click on desired project file, click "Export .tdt to disk drive A:".
Scenario c: In project explorer file list, right-click on desired project file, click "Export .tdt...", chooce directory and filename to export.
Precondition: At least 1 project file in current (selected) directory.
Postcondition: Project data exported to .tdt file.
Error condition: Notify if file error occurs (such as no disk in A: drive).

UC16: Export single measurement details into .srm file
Scenario a: In measurement result table, right-click on desired measurement line, click "Export .srm in current directory".
Scenario b: In measurement result table, right-click on desired measurement line, click "Export .srm to disk drive A:".
Scenario c: In measurement result table, right-click on desired measurement line, click "Export...", chooce directory and filename to export.
Precondition: At least 1 measurement result in current project.
Postcondition: Measurement details exported to .srm file.
Error condition: Notify if file error occurs (such as no disk in A: drive).

UC17: Print measurement results
Scenario a: Click "Print...", "Measurement results". [2005-02-23 propably gone in current UI prototype; implementation priority low]
Precondition: Open project.
Postcondition: Measurement results printed via [Java] standard printing window.
Error condition: Let know if printing error occurs.

UC18: Print graph sheet (with 7 different graphs; described elsewhere)
Scenario a: Click "Print...", "Grap sheet". [2005-02-23 propably gone in current UI prototype; implementation priority low]
Precondition: Open project.
Postcondition: Measurement results printed via [Java] standard printing window.
Error condition: Let know if printing error occurs.

3.3 Functionality

UC19: Create new project [.dat file?]

UC20: Load project [.dat file?]

UC21: Append measurement results to project [.dat file?]

UC22: 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

UC23: Insert AF sequence with start-step-stop values

UC24: Load AF sequence

UC25: Save AF sequence

UC26: Edit AF sequence on-the-fly

UC27: Edit stored AF sequences

UC28: Rename stored AF sequence

UC29: 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 mesurement data after every measurement step.
Priority: 1

Identifier: R4
Name: Loading
Description: Saved mesurement date 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: R14
Name: Manual
Description: Able to operate magnetometer manually.
Priority: 2

ReqIdR15 **Name:** Calibration
Description: Magnetometer must be calibrated every 24h..

Priority: 2

ReqIdRXX **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

The computer that will run this program will be equal or better to 1GHz CPU, 256MB RAM, 1280x1024 resolution. The program must run under Windows XP. The hardware and communication with it is described in the section "External interfaces".

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

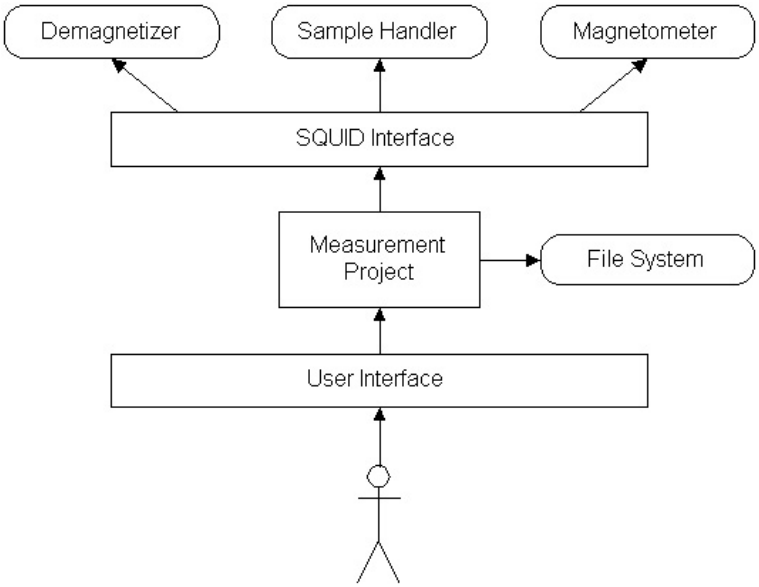

Figure 1: Architecture overview

5.1 System restrictions

6 User interface

Overview of UI described here..

6.1 Goal derived use cases

7 Architecture overview

8 External interfaces

Interfaces to existing software and hardware are described here.

8.1 Existing program

The existing software for using the SQUID is "2G Enterprises Data Acquisition". We have access to the source code for version 2.99.3 of the program. From the old source code we will reuse basically only the SerialIO component. We will build an interface for communicating with the SQUID hardware by using Java and JNI (Java Native Interface).

8.2 Hardware control protocols

The SQUID consists of three independent units:

- Automated Sample Handler System (MODEL 2G800)
- Automatic Sample Degaussing System (MODEL 2G600)
- Superconducting Rock Magnetometer (MODEL 755R or 760R)

Automated Sample Handler System controls the movement and rotation of the sample holder. Its protocol is described in Appendix 1.

Automatic Sample Degaussing System controls the demagnetizer. Its protocol is described in Appendix 2.

Superconducting Rock Magnetometer reads the measurements from the magnetometer. Its protocol is described in Appendix 3.

9 Validation

Description of how to validate the set requirements.

Appendix 1. Automated Sample Handler System Protocol

Korvaa tämä sivu tiedostolla "Automated Sample Handler System - Protocol.pdf"

Appendix 2. Automatic Sample Degaussing System Protocol

Korvaa tämä sivu tiedostolla "Automatic Sample Degaussing System - Protocol.pdf"

Appendix 3. Superconducting Rock Magnetometer Protocol

Korvaa tämä sivu tiedostolla "Superconducting Rock Magnetometer - Protocol.pdf"