Project Plan: Designing a User Interface for the Innovative E-mail Client Framework

Student: Alexandra Burns
Supervising Assistants: Stephanie Balzer, Joseph N. Ruskiewicz
Supervising Professor: Betrand Meyer

December 2005

1 Project Description

1.1 Project Overview

This project aims at developing the graphical user interface of an email client offering a solution to the information overload problem. The term *information overload* refers to the time consuming issue of keeping up with large amounts of incoming and stored email. Users often have to deal with this problem on a daily basis and therefore benefit from an email client that allows them to efficiently search, display and store their email. The project builds upon an object-oriented framework by Andrea Rezzonico [5]. A first part of this project focusses on identifying the problems that users encounter when working with existing email clients. This information will then be compiled into use cases and scenarios. The *use cases* will summarize the experience of a user with a specific functionality of the existing solution - for instance, there will be a use case focusing on search mechanisms. The *use scenarios* will state what the user should experience with the specific functionality. While the *use case* describes the situation as it is, the *use scenario* focuses on the situation as it should be. The second part of the project will be to implement a graphical user interface that satisfies the use scenarios.

1.2 Project Goals

A first goal is to identify and study information overload problems of existing email clients. For instance, is it possible to find an email with a specific date and if yes, how easily and how fast is it retrieved? What kinds of viewing options does the user have? We will then gather the results to form use cases that are problematic when applied to existing email clients. The aim is then to design and implement a graphical email client interface that supports the use scenarios corresponding to the use cases. The graphical interface should reuse the elements of the framework [5], in a way that provides the users with easy, concise, fast and flexible access to their email. We will not evaluate the resulting system as a prototype, but as an extensible basis for a new email client solution. Upon completion of the development we will test the system using an 4GB email archive. Together with the underlying framework the client interface should result in a tool that supports users by combining functionalities that are not offered in the same way by existing tools. Many email clients are designed according to the inofficial standard, i.e. they provide the functions that the other email clients offer. One of the reasons may be that users are reluctant to adapt to a completely new way of doing things, especially with applications like email clients that are crucial to everyday life. The aim of this project is to explore what is possible outside this standard, without having to think of business rules.

2 Project Management

2.1 Objectives

- **Background Material** : The first subgoal is to analyse the background material for the semester project.
- **Use scenarios**: The problems when using existing email clients are to be identified. The information is then combined to form use cases and use scenarios.
- **Implementation** : The implementation will focuss on solving the information overload problems that were encountered in the use scenarios.

2.2 Criteria for Success

The goal of this thesis is to find a solution to the information overload problem. This will be achieved by first identifying the use cases that are problematic for existing solutions. The success of the project will be measured by how the proposed solution satisfies the use scenarios developed in the second stage of the project.

The quality of the implementation part of the thesis will be ensured by adhering to the following guidelines: The **Design by Contract** principle entails pre- and postconditions, class invariants and loop variants and invariants. To ensure **Good Design** we will use design patterns and careful abstraction. The implementation must also be extendible and reusable. Furthermore we will follow the core principles of **Object Oriented Software Construction OOSC** [3] , namely Small Interfaces, uniform access, information hiding, operand and command query separation principle.

2.3 Process

We will monitor the progress of the project with the following measures: The student Alexandra Burns will send weekly reports to both supervisors describing the progress made on the tasks, problems encountered (if applicable) and an analysis of the reading material. The supervisors Stephanie Balzer and Joseph Ruskiewicz will review these weekly reports and provide feedback and corrections where applicable. If the supervisors or the student encounter a problem concerning the project, a meeting will be called to discuss the matter.

2.4 Final Deliverables

The final deliverables consist of the solution implementation and the thesis report. The thesis report incorporates the weekly reports and contains an analysis of the existing technologies, a review of the reading material, the use cases and scenarios and a documentation of the implementation.

3 Proposed Milestones

3.1 Project Steps

Milestone	**Objective**
M1	Reading
M2	Analysis of existing solutions
M3	Develop use cases and scenarios
M4	Implementation
M5	Documentation

3.2 Deadlines

Milestone	**Deadline**
M1	1 week
M2	2 weeks
M3	4 weeks
M4	3 weeks
M5	2 week

3.3 Tentative Schedule

This is the tentative schedule for the project. The duration of each task is estimated in weeks.

Task	1	2	3	4	5	6	7	8	9	10
Reading										
	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Analysis of existing solutions										
	▨	▨								
Use Cases and Scenarios										
			▨	▨	▨	▨				
Implementation										
							▨	▨	▨	
Documentation										
	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

4 Bibliography

References

[1] Thomas B. Hodel, Roger Hacmac, and Klaus R. Dittrich. Using text editing creation time meta data for document management. In *CAiSE*, pages 105–118, 2005.

[2] David D. Lewis and K. A. Knowles. Threading electronic mail - a preliminary study. *Inf. Process. Manage.*, 33(2):209–217, 1997.

[3] Bertrand Meyer. *Object-Oriented Software Construction.* Prentice Hall, 2nd edition, 1997.

[4] Gail C. Murphy, Mik Kersten, Martin P. Robillard, and Davor Cubranic. The emergent structure of development tasks. In *ECOOP*, pages 33–48, 2005.

[5] Andrea Rezzonico. Master thesis: Designing an innovative email client, 2005.

[6] Hironori Washizaki and Yoshiaki Fukazawa. A model-view separation architecture for gui application components. In *ITCC (2)*, pages 359–364, 2005.

[7] Steve Whittaker and Candace L. Sidner. Email overload: Exploring personal information management of email. In *CHI*, pages 276–283, 1996.