

Designing an Innovative E-Mail Client

PROJECT PLAN

Project Period: April 1st 2005 - October 7th 2005
Student Name: Andrea Rezzonico
Student-No: 00-920-793
E-Mail Address: rezzo@student.ethz.ch
Supervisor Assistant: Joseph N. Ruskiewicz
Supervising Professor: Prof. Bertrand Meyer

1 Project Description

1.1 Overview

Electronic mail is an essential form of communication in today's society. The ability to have instant communication with friends, family, and colleagues has made it substantially important for the average person.

As more and more individuals are using email, the mailboxes are getting fuller and fuller over time. As a user progresses from an average user to a power user, they are faced with the problem of organizing messages in such a way that they can find and recall them for later reference.

The current set of tools available for the power user's are not able to keep up with this ever increasing load of email on their machine. Even the most advanced tools for storing and retrieving email are unable to filter and adjust email according to the user's preferences. Thus the user is required to manage their own way of organizing email into separate "folders" or repositories.

With advanced power users, say over 10 Gig. of email data, this is a large problem. The ability of the user to keep up with the incoming email and have them sorting has lead to an *information overload*.

This information overload is known in the email tool domain (e.g. Gmail [1] and Lookout [2]), but these tools have only started to lead the way to solving this problem. The email tool must allow for dynamic relationships between emails, automatically storing, and most importantly, quick and responsive searching across all the repositories.

This work proposes to provide a solution for the power users of email. It will provide an object-oriented framework in which to build a powerful email tool upon with emphasis on the storage and efficient retrieval of emails based on users preferences.

1.2 Scope of the Work

This project aims to provide an applicable solution to the information overloading that power users of email are experiencing. To solve this problem the tool must use advanced software engineering techniques and have efficient algorithms for handling the storage and retrieval of emails.

The use of advanced object-oriented techniques will provide a foundation for further developers to create and deploy advanced versions of the tool. The foundation provided by the development of this project will be a step way for creating many semester projects each focusing on one specialized part of the email tool.

So the tool will be responsive and adaptive to the user's needs, efficient algorithms in the domain of machine learning [11, 7] will need to be developed and deployed. The proposed solution will use modern techniques of machine learning to "learn" what the user is intending to do and be able to react to it in an automatic fashion.

Extending upon the work of the machine learning, it will also be quite important that the solution will provide an advanced query language [4] that will allow the user to rapidly find the important emails that they are searching for in an easy and efficient manner.

1.3 Intended Results

The goal of this project is to implement and deploy a foundation for future semester projects which should add functionality to the core system in order to develop a complete email client. Possible semester projects may be:

- Protocol handlers (POP, IMAP, SMTP)
- Implement a GUI
- Dealing with storage issues
- Spam filtering (intelligent filters)

To facilitate the future potential semester projects, the solution will need to provide a solid foundation for the students to work on. This foundation should use proven object-oriented techniques as given in object-oriented construction [9] and should provide an extensible and reusable framework that will allow the students to continue extending and enhancing the email tool.

The solution shall also provide the essential algorithms and data structures that will implement features required by power users. These algorithms and data structures will implement the indexing, searching, and storage features of the email tool. The storage features will include message filtering that will adapt to the user's manual filtering criteria, eventually learn and automate the process.

2 Background Material

Reading list

- Virtual Folders: Database Support for Electronic Messages Classification [3]
- A Query Language for Retrieving Information in Electronic Mail Environments [4]
- Issues when designing filters in messaging systems [10]
- Analyzing an Email Collection Using Formal Concept Analysis [6]
- Threading Electronic Mail: A Preliminary Study [8]
- Dynamics of Incremental Learning in an email Classifier [11]
- ifile: An Application of Machine Learning to email Filtering [7]
- Ishmail: Immediate Identification Of Important Information [5]

3 Project Management

3.1 Project Phases

Analysis of the existing email clients During this phase of the project current solution should be studied and reviewed in order to achieve a solid knowledge of the “state of the art”.

Define the object model In this phase an object model (OM) should be developed. This OM should consists in a set of deferred classes which defines the main components of the system (i. e. the abstraction of an email message, protocols handlers, . . .).

Define the data structures and the algorithms Since the product should deals with a huge amount of data, particular attention should be payed to performance issues.

During this phase the data structures and the algorithms should be defined in order to achieve the needed performance.

3.2 Objectives and Priorities

Objective	Priority
Reading	2
Evaluating existing technologies	2
Define the object model	1
Define the data structures and the algorithms	1
Developer manual	2
GUI	3
Thesis report	1

1 means highest priority, 3 means lowest priority.

3.3 Criteria for Success

The goal of this project is to design and develop a framework of an email client which is able to handle a huge amount of email messages in order to avoid the information overloading problem.

3.3.1 Guidelines

The criteria for success of this project are:

- Use of Design by Contract
 - Pre- and postcondition
 - Class invariants
 - Loop variants and invariants

- Good design
 - Use of design patterns
 - Extendibility
 - Reusability
 - Careful abstraction
- Core principles of OOSC2 [9]
 - Command/query separation
 - Simple interfaces
 - Uniform access
 - Information hiding

3.3.2 Key Points

The key facilities of such an email client are:

- Index email messages using a database
- Develop a query language to retrieve email in a fast way
- Filter incoming email messages automatically using intelligent filters and manually

3.4 Quality Management

Quality will be ensured by:

- Weekly progress reports sent to the supervisor, if necessary a meeting with the supervisor will take place
- Detailed progress report for each milestone
- Review of each milestone by the supervisor concluded by a meeting
- Validation of each milestone after review (see Sec. Validation Steps below)
- Documentation (see Sec. Documentation below)

3.5 Documentation

- Progress reports
 - Short weekly progress reports consisting of the main task completed
 - Detailed reports for each milestone consisting of:
 - * The main tasks
 - * Eventual encountered difficulties

* Achieved results

- Developers report: This manual documents the software architecture and its limitations, the difficulties encountered during the implementation, explains how the software could be extended.
- Thesis report: The thesis report consists of the final developer manual and a theoretical part describing the technologies used.

3.6 Validation Steps

The validation steps of each milestone comprises:

- Report: Sending a detailed report and the relevant parts of the work to the supervisor for review
- Meeting: Organizing a meeting with the supervisor for presentation and discussion of the conducted work
- Revision: Revising parts of all the work for this milestone, depending on the conclusion of the supervisor

4 Plan With Milestones

4.1 Project steps

Milestone	Objective
M1	Acquire knowledge about the state of the art
M2	Definition of the object model
M3	Implementation of the data structures and algorithms
M4	Developer manual
M5	Thesis report

4.2 Deadlines

Milestone	Deadline
M1	27 May 2005
M2	24 Jun 2005
M3	19 Aug 2005
M4	16 Sep 2005
M5	7 Oct 2005

4.3 Tentative Schedule

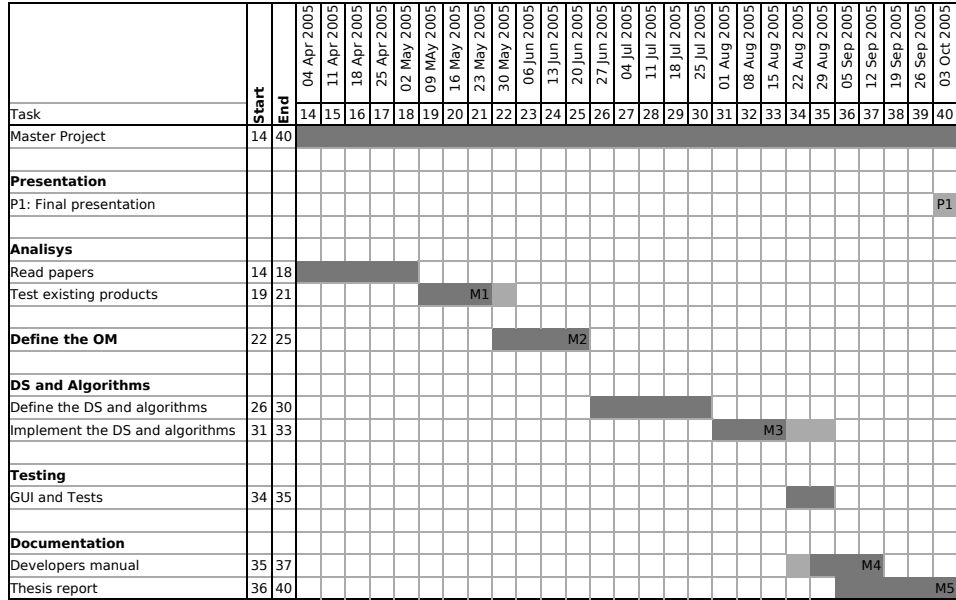


Figure 1: Tentative schedule

5 Bibliography

References

- [1] Gmail. <http://gmail.google.com>.
- [2] Lookout. <http://www.lookoutsoft.com/Lookout>.
- [3] Karin Becker and Simone Nunes Ferreira. Virtual folders: Database support for electronic messages classification. In *CODAS*, pages 163–170, 1996.
- [4] S. Ferreira. A query language for retrieving information in electronic mail environments, 1997.
- [5] J. Helfman and C. Isbell. Ishmail: Immediate identification of important information, 1995.
- [6] Richard J. Cole II and Peter W. Eklund. Analyzing an email collection using formal concept analysis. In *Principles of Data Mining and Knowledge Discovery*, pages 309–315, 1999.
- [7] R. Jason. ifile: An application of machine learning to e-mail filtering, 1998.
- [8] David D. Lewis and K. A. Knowles. Threading electronic mail - a preliminary study. *Information Processing and Management*, 33(2):209–217, 1997.
- [9] Bertrand Meyer. *Object-Oriented Software Construction*. Prentice Hall, 2nd edition, 1997.
- [10] J. Palme, J. Karlgren, and D. Pargman. Issues when designing filters in messaging systems, 1993.
- [11] Richard B. Segal and Jeffrey O. Kephart. Dynamics of incremental learning in an e-mail classifier.