Constantin Berzan
exit3219@gmail.com
exit on IRC

Standalone Mail Composer in KDEPIM

Having a standalone mail composer (instead of it being a part of KMail) will significantly reduce code duplication between the KDE email applications, KMail and Mailody. In addition, it will offer non-email applications an easy way to present the user with a send-email dialog. The proposed implementation focuses on porting the mail composer in KMail from the antiquated mimelib to the modern KMime, hardening it with unit tests, and separating it from KMail so that other applications can use it too.

Background / Motivation

Currently KDE comes with two major email applications: the tried-and-tested KMail and the new-and-lightweight Mailody. Each of these applications come with their own mail composer component. Mailody uses the modern KMime library from kdepimlibs, while KMail still uses the old mimelib library (unmaintained for at least the past two years). There is unnecessary duplication on three levels: the MIME library, the message builder, and the composer GUI. Having a standalone mail composer, split into a message builder and a composer GUI, would solve this duplication problem, with the added benefit of porting KMail to KMime (at least partially; the reader component can be ported and separated from KMail later).

Nowadays, when a regular non-email application wants to send an email, it has no way of doing so without loading the entire weight of KMail. (use case: when Amarok crashes, it offers to submit a backtrace by email). This proposal is the first step towards fixing this problem. The remainder of the solution is to have a global outbox via Akonadi.

Over the years KMail has become a huge and difficult to maintain heap of code. This proposal fits into the plan of breaking down KMail into reusable, manageable parts. An important goal of this proposal is to create solid code, easy to test and maintain. This will be achieved partly by using KMime, and partly by writing thorough unit tests.

Implementation Details

The main goals of this proposal are to:

- Separate the composer component from KMail, and put it in a library where it would be accessible to all applications. Separate the message builder and GUI parts of the composer.
- Write extensive unit tests for the composer component.
- Port the composer to KMime.

- Perform surgery on KMail to allow it to use this new composer.

These changes must not result in a loss of functionality from the current composer that KMail uses.

Some optional goals of this proposal are to:

- Modify Mailody to use this new composer.
- Separate the mail reader component of KMail and port it to KMime as well.
- Finish the mail dispatcher agent that Ingo Klöcker has started in playground/pim, using the mailtransport library. Port KMail to use this for sending mail. This would create a global outbox that all applications can use.
- Create an Akonadi agent for POP3 based on the code in kio_pop3. This is another step towards porting KMail to use Akonadi.

A subset of these goals will be achieved, depending on the remaining time available after implementing the main goals.

Tentative Timeline

Now – May 22	Study KMail and Mailody code to come up with an initial plan for what the composer should have. Discuss with mentor and KMail developers to decide whether unit testing should be attempted before or after the separation of the composer from KMail.
May 23 – July 6	Separate the message builder from the composer GUI in KMail. Bombard the builder with unit tests. Separate the message builder and the composer GUI from KMail and put them in a library.
July 7 – August 10	Make KMail use the newly separated composer. Port the message builder to KMime. Weigh the remaining time and implement some of the optional goals.
August 11 – August 17	Improve unit tests and documentation. Clean up code and discuss merging it into the main branch.

About Me

I am studying Computer Science at Tufts University, near Boston, in the United States, but I will be going home to Moldova for the summer. I will be in EDT (UTC-4) up until May 10th, and in EEST (UTC+3) afterwards, until September. I have no other obligations during the summer, and I am planning to devote to this project full-time.

I have been using GNU/Linux and KDE since 2005. I have contributed to KDE sporadically since 2007, my involvement mostly limited to summers because of the amount of schoolwork in the rest of the year. Since 2008 I am maintaining KCharSelect, where I have fixed some bugs and implemented some minor features.

I am the developer of Konvert2Ogg[1], a simple sound conversion front end for audio book enthusiasts. I have also started a port of Filelight[2] to KDE4, but the project has stalled due to my lack of time and obsessive tinkering with the design. Recently Martin Sandsmark released a port of his own in playground/utils. I am very well-versed in C++ and fairly well acquainted with Qt and kdelibs, as well as the open source development process. A KDE SoC project would be my largest contribution so far, and I am looking forward to spending my summer developing KDE.

I have submitted another application for a KDE SoC (window tiling in KWin), but this project (standalone mail composer) is my *first* choice. I am very excited to work on it if it gets accepted.

[1] http://konvert2ogg.sourceforge.net/

[2] http://gitorious.org/projects/filelight-kde4port/