

メッセージのソース表示可能な Android 用電子メールアプリケーション開発

2010SE203 正坂 佳也

指導教員：後藤 邦夫

1 はじめに

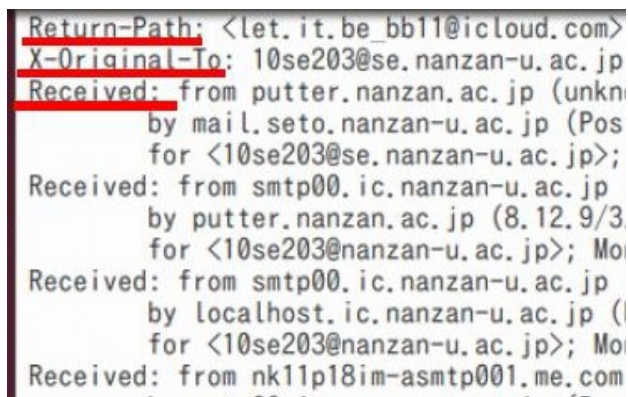
近年スマートフォンの普及率が急激に増加しており、日本では約 2 人に 1 人が所持している。“小さな PC”とも呼ばれ、大変便利でなくてはならない存在になっている人も多い。その分危険とも隣合わせであり、多くの事件・事故が起きている。

トラブルを未然に防ぐ一つの策として、メッセージのソース表示を可能にすることである。PC のアプリケーションには当たり前にある機能だが、スマートフォンのアプリケーションには数が少ない。

そこで本研究では、K-9 Mail のオープンソースを使い、メッセージビューの外観が定義されたプログラムにメッセージソース情報を追加し、レイアウトを変更する。本研究を実現することによって、送信元を特定でき迷惑メール対策に有効的である。

2 メッセージのソース情報

インターネットでやりとりされるメールには、メッセージに関するさまざまな情報が「ヘッダ」として添付されている。



```
Return-Path: <let.it.be_bb11@icloud.com>  
X-Original-To: 10se203@se.nanzan-u.ac.jp  
Received: from putter.nanzan.ac.jp (unknown  
by mail.seto.nanzan-u.ac.jp (Postfix)  
for <10se203@se.nanzan-u.ac.jp>);  
Received: from smtp00.ic.nanzan-u.ac.jp  
by putter.nanzan.ac.jp (8.12.9/3.1.1)  
for <10se203@nanzan-u.ac.jp>; Mon, 12  
Received: from smtp00.ic.nanzan-u.ac.jp  
by localhost.ic.nanzan-u.ac.jp (Postfix)  
for <10se203@nanzan-u.ac.jp>; Mon, 12  
Received: from nk11p18im-asmt001.me.com
```

図 1 PC で見た電子メールのメッセージソース

スマートフォンでは、送信者のアドレスしか表示することができない。PC では、図 1 で示したように多くの情報が表示することができる。

• Received

ヘッダ情報の中でもっとも重要なのが Received である。経由したサーバの IP アドレスやドメイン名、送信元の IP アドレスやドメイン名などが記録される。経由したサーバの数だけ Received が記録される。基本的にヘッダ番号の下から順に Received が送信元で、一番上の Received が自分が利用しているメールサーバであり、下から順に経由したサーバ

が記録されている。経由したサーバがわかるので怪しいか判断できる。迷惑メールを通報をする場合に必要となる。

```
Received: from 送信した端末名またはホスト名など (右記の IP アドレスのホスト名 [送信した端末の IP アドレス])  
by 受け取ったサーバ名 with SMTP などの転送方法 id 転送時の ID 番号  
for 宛先のメールアドレス; 処理時刻
```

3 システムの実現

このアプリケーションを設計する上で追加する機能を挙げ、それを基に設計方針を述べる。

3.1 追加機能

1. メッセージのソース表示可能
前述で述べたスマートフォンで表示できないソースを表示する。
2. ユーザーの使いやすい・見やすい表示方法
このアプリケーションを実装しても、ユーザーが解りづらい表示では意味がない。わかりやすい表示方法にする。

3.2 設計方針と概要

前述で述べた機能を満たすために、次の設計方針を考えた。

- Android SDK をインストール [3]
Android アプリを開発する際に必要なツール。eclipse[2] を含む。
- オープンソースの K-9 Mail[1] を使用し設計する。
K-9 Mail は複数候補あるオープンソースの中でも多機能であり、メッセージソース表示を実現することによって、ユーザーが求める電子メールソフトウェアとなる。
ファイル数 3042

本研究で主要となるディレクトリ階層を図 2 に示す。

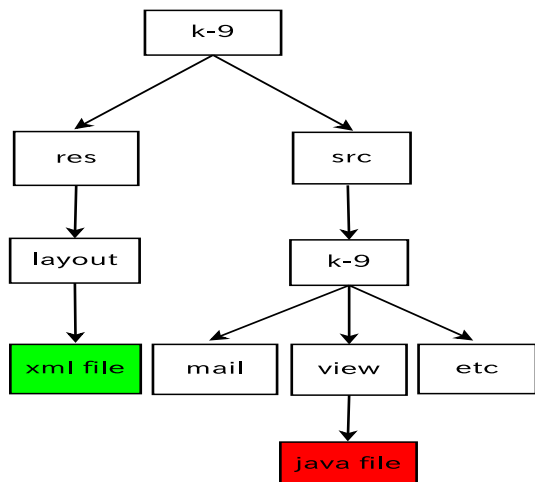


図2 追加・修正するディレクトリ階層

xml ファイルは、外観を定義する。その xml ファイルを追加・修正し、画面の構成をする。java ファイルは、ヘッダーを取得するプログラムである。java ファイルを修正し、ヘッダーの取得・表示を可能にする。

3.3 機能追加プログラム

layout/message.xml にメッセージビューの外観が定義されている。

MessageView の画面構成の中にヘッダーが表示されるように include message_view_header.xml を新たに定義する。

message_view_header.xml に表示されるヘッダーのレイアウトを新たに定義する。(206 行)

- message_view_header.xml の構造の一部分を以下に示す。

additional_headers_view が追加ヘッダーである。

```

MessageHeader
View (from)
View (to_label)
View (to)
View (cc_label)
View (cc)
View (date)
CheckBox (flagged)
View (additional_headers_view)
  
```

ヘッダーの取得、反映するように MessageHeader.java を継承し、一部変更する。(46 行)
修正・追加部分の一部を以下に示す。

```

view/MassegeHeader.java を継承する .
mAdditionalHeadersView = &(View)
    findViewById(R.id.additional_headers_view);
additional_headers_view を取得して ,
mAdditionalHeadersView に設定する .
showAdditionalHeaders(); を定義する .
  
```

4 実験と評価

実装より、期待通りメッセージソース図 3 を表示することができた。

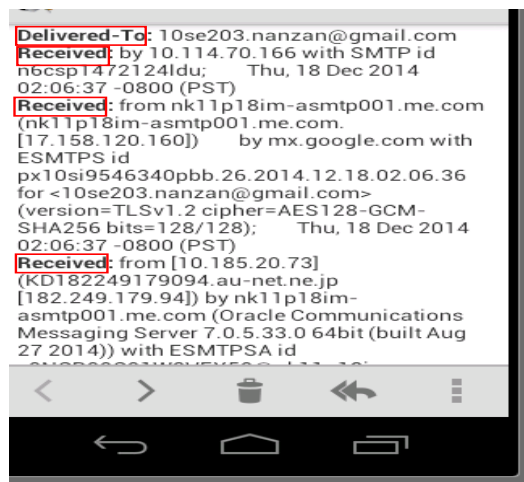


図3 Android メッセージソース画面

実機 (Android4.4) での表示も確認することができた。

5 おわりに

本研究の結果から、メッセージソースの表示を実現することができた。このことにより、迷惑メール対策などユーザーがより安全に安心して電子メールを使用することができると考えられる。そして、今後の研究課題として、以下の 2 点が挙げられる。

- セキュリティ面の強化 (メールフィルターの追加)
- 品質の向上, 評価, 改善

上記の研究課題を完成させることにより、ユーザーにとって使いやすいものに近づくと考えられる。

参考文献

- [1] GitHub: K-9 Mail (accessed: Jan. 2015). <https://github.com/k9mail/k-9>.
- [2] MergeDog Project: Dynamic AOP Translation Container for Eclipse (accessed: Jan. 2015). <http://mergedoc.sourceforge.jp/>.
- [3] 山田祥寛: はじめての Android アプリ開発 Android4 対応版, 秀和システム, pp. 1-45 (2012).