NAIST-IS-MT1051097

修士論文

リファクタリングの実施が欠陥混入に与える影響の調査を目的とした開発履歴分析

藤原 賢二

2012年 2月 2日

奈良先端科学技術大学院大学
情報科学研究科 情報システム学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士 (工学) 授与の要件として提出した修士論文である。

藤原 賢二

審査委員：

飯田 元 教授　　(主指導教員)
松本 健一 教授　(副指導教員)
安本 慶一 教授　(副指導教員)
吉田 則裕 助教　(副指導教員)

リファクタリングの実施が欠陥混入に与える影響の調査を目的とした開発履歴分析*

藤原 賢二

内容梗概

リファクタリングとは，ソフトウェアの外部的な振る舞いを変えることなく内部構造を改善することである．一般に，ソフトウェアの開発が進むと，リファクタリングが必要となる設計品質の低い箇所がソースコード中に増加していく．このような箇所をリファクタリングにより除去し，ソースコードを高い品質で維持して開発を行うことでソフトウェアの欠陥を予防できるといわれている．しかし，この効果を定量的に評価した研究は少なく，評価しているものについても，精度の低いリファクタリングの検出手法を利用しているという問題が指摘されている．本研究では，ソフトウェアの開発履歴を用いてリファクタリングが欠陥混入に与える影響を調査するための分析手法を提案する．提案手法では，開発履歴からリファクタリングの実施頻度，欠陥の混入頻度および欠陥の修正頻度を計測する．次に，提案手法をオープンソースソフトウェアである Columba に適用し，リファクタリングが欠陥混入に与える影響を調査した．調査の結果，Columba の開発においてリファクタリングが最も実施された期間以降は，欠陥の混入が減少傾向にあったことが観測された．また，リファクタリングの実施頻度が高くなった後に欠陥の修正頻度が高くなる傾向にあることがプロジェクト全体を通して観測された．

キーワード

リファクタリング，ソフトウェアリポジトリマイニング，版管理システム，バグ管

*奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 修士論文, NAIST-IS-MT1051097, 2012 年 2 月 2 日.

理システム，オープンソースソフトウェア

An Investigation into the Effect of Refactoring History on Defect Introduction*

Kenji Fujiwara

Abstract

Refactoring is a technique for improving software design. It is the process of changing the structure of program without changing its behavior. In general, fragments of source code that require refactorings are increased with the progress of a software development. These fragments can be improved by refactoring. Thus, frequent refactoring during software development is considered to prevent defect introductions because it keeps the design quality of source code. It is one of widely known refactoring effects. However, a few quantitative studies have been investigated the effect. In addition, refactoring detection techniques used by these researches have a problem of law recall. This research proposes a method to investigate the effect of refactoring on defect introductions, and applies the proposed method to an open source software project Columba. In order to investigate the relationship, we compute frequencies of refactoring, defect introductions and defect fixings by mining the development of Columba. The result shows defect introductions tend to decrease in the term after frequent refactoring, and defect fixings also tend to increase in the term after frequent refactoring.

Keywords:

*Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT1051097, February 2, 2012.

Refactoring, Mining Software Repositories, Version Control Systems, Bug Tracking Systems, Open Source Software

目次

図目次

表目次

リスト目次

関連発表論文

国際会議（査読無し）

1. Kenji Fujiwara, Kyohei Fushida, Norihiro Yoshida, Hajimu Iida, “An Approach to Investigating How a Lack of Software Refactoring Effects Defect Density,” Joint Workshop on Software Science and Engineering, In IEICE Technical Report, volume 111, number 107, pp.59–62, June 2011.

研究会・シンポジウム

1. 藤原 賢二, 伏田 享平, 吉田 則裕, 飯田 元, “オープンソースソフトウェアを対象としたリファクタリングが欠陥混入に与える影響の調査,” 日本ソフトウェア科学会 第 28 回大会 講演論文集, 2011 年 9 月.
2. 藤原 賢二, 伏田 享平, 吉田 則裕, 飯田 元, “ソースコード履歴情報に基づくリファクタリングと欠陥の関係分析,” 平成 22 年度 情報処理学会関西支部支部大会 講演論文集, 2010 年 10 月.

1. はじめに

情報化社会の成長とともにソフトウェアは社会の基盤を支える重要な構成要素となっている．それに伴い品質の高いソフトウェアを効率良く開発することの重要性が増している．加えて，近年のソフトウェア開発は変化の早い社会情勢に対応するために短期間でのリリースが求められることが多く，このような状況において高品質なソフトウェア開発することが課題となっている．

近年，ソフトウェアの品質を高める技術としてリファクタリングが注目されている．リファクタリングとはソフトウェアの外部的な振る舞いを変更することなく内部の構造を改善することをいう．Fowler が典型的なリファクタリングパターンをまとめている [6, 7]．これらのリファクタリングは対象となるソフトウェアがオブジェクト指向で設計，実装されていることを前提としている．広く知られているリファクタリングの一例としては「メソッドの引き上げ（Pull up Method）」，「メソッドの抽出（Extract Method）」が挙げられる．図 1 に示す「メソッドの引き上げ」リファクタリングでは，共通の基底クラス S を持つクラス A，B が持つメソッド a，b のコードが重複している場合に，基底クラスのメソッド s としてメソッドを集約する．このリファクタリングにより，修正前に重複していたコードに対応するコードの修正コストの削減が期待される．

一般にソフトウェアの開発が進むと，設計品質の低い箇所がソースコード中に増加していく．このような箇所は「コードの不吉な匂い」と呼ばれており，リファクタリングは不吉な匂いを除去することを目的として実施される．リファクタリング

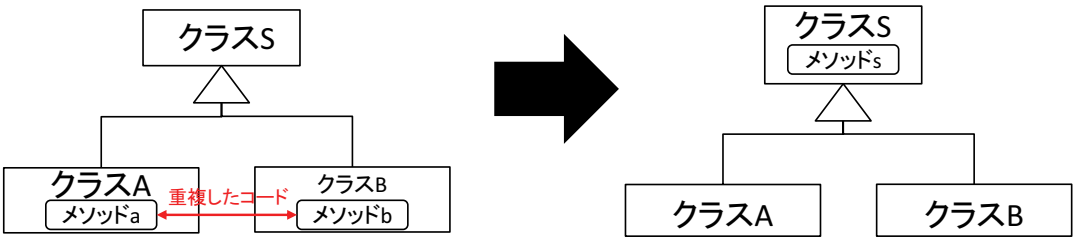

図 1　メソッド引き上げリファクタリングの例

を実施し設計品質の低いコードを除去することで，開発時における欠陥の混入を予防できるといわれている．しかし，欠陥の混入を予防するためにどの程度のリファクタリングをどのような頻度で行えば良いのかは明らかになっていない．そのため，ソフトウェア開発において，リファクタリングの実施を効果的に行うことは難しい．開発者がより効果的にリファクタリングを実施するためには，欠陥を予防するために最適なリファクタリングの種類や実施頻度を明らかにすることが望ましい．

本研究ではソフトウェア開発時の欠陥混入を予防する観点から，開発者がリファクタリングを効果的に実施可能になることを目的とする．そのために，まずリファクタリングが欠陥混入に与える影響を分析する．分析にあたり，リファクタリングの頻度，欠陥が混入された頻度，欠陥が修正された頻度の 3 種類の尺度を定義した．次に，一般的なソフトウェア開発プロジェクトにおいて用いられる版管理システム，バグ管理システムを用いてこれらの尺度を計測する手法を提案した．版管理システムとはソースコードの変更履歴を管理するためのシステム，バグ管理システムはソフトウェアの欠陥を集中管理するためのシステムである．提案手法をオープンソースソフトウェア Columba に適用し 3 種類の尺度の傾向を分析した．その結果，Columba の開発においてリファクタリングが最も実施された期間以降は，欠陥の混入が減少傾向にあることが観測された．また，リファクタリングの実施頻度が高くなった後に欠陥の修正頻度が高くなる傾向にあることがプロジェクト全体を通して観測された．本論文ではこれらの結果を踏まえ，リファクタリングが欠陥の混入に与える影響について考察する．

以降，2 章で本研究に関する諸研究について述べ，3 章で本研究において提案する手法について説明する．その後，4 章で提案手法をオープンソースソフトウェアに適用した実験について述べ，5 章で考察する．最後に，6 章で本論文をまとめる．

2. 関連研究

本章では本研究に関連する既存の研究について述べる．2.1 節では，提案手法で利用するソフトウェア開発履歴からリファクタリングに関する情報を抽出する研究について，2.2 節では同じく欠陥混入に関する情報を抽出する研究について述べる．2.3 節では，リファクタリングの効果を検証することを目的とした研究について紹介する．

2.1. リファクタリング検出に関する研究

リファクタリングの有効性や実施状況について分析する際は，「いつ」，「どのような」リファクタリングが行われたかを知る必要がある．この要求に対するアプローチとして，以下の 4 つのアプローチが研究されている [15].

(a) 版管理システムに記録されたコミットログを用いる:
開発者が版管理システムに変更をコミットする際にその変更内容を「コミットログ」として自然言語で記述，記録できる．コミットログに「refactor」や「extract」，「rename」など，リファクタリングに関する単語を含む場合，その変更でリファクタリングが行われたと判断する [19, 22].

(b) ソースコードの編集履歴を解析する:
異なるバージョン間におけるソースコードの差分を解析し，「変数名の変更」や「メソッドの抽出」などのリファクタリングを検出する．

(c) 開発者の行動を監視する:
開発者が「どのように」リファクタリングを行うかを，直接またはツールを使って監視する [3, 14, 17]．このアプローチは適用範囲が限られるが，必要な情報を詳細に収集することが可能である．

(d) リファクタリングツールの使用を記録する:
統合開発環境が提供するリファクタリング支援機能をユーザが「いつ」，「どのような場所に」使用したかを記録し，リファクタリングに関する情報を集める [4, 13, 20].

上記の手法の中で，(a)，(b) は既存の版管理されているソフトウェア開発プロジェクトに適用可能である．一方 (c)，(d) は事前に準備が必要なため，適用範囲が限られる．よって，本研究における提案手法では (a) または (b) のアプローチでリファクタリングが実施された箇所を検出するのが望ましい．しかし，Murphy-Hill らにより行われた調査ではコミット時に記録されるログメッセージはリファクタリングが行われたかどうかを判断するには信頼性が低いと結論づけている [16]．そこで，提案手法では (a) のアプローチより高精度にリファクタリングを検出できる (b) のアプローチを用いてリファクタリング検出を行う．以降，(b) のアプローチによるリファクタリング検出手法についての研究を紹介する．

Weißgerber と Diehl はソースコードの構文および識別子情報を用いた手法を提案している [23]．彼らの手法は，リファクタリングと思われるコードの変更箇所を，構文情報と識別子情報を用いて抽出する．そして，それらの変更箇所を，コードクローン検出手法 [10] を用いてランク付けし，リファクタリングを検出する．この手法で検出可能なリファクタリングは 10 種類である．

Xing らは任意のバージョン間におけるソースコードの設計情報の差分を求める UMLDiff アルゴリズム [26] を提案している．また，その差分情報を解析することでリファクタリングを検出する手法も提案している [27]．UMLDiff では比較対象となる 2 バージョン間のソースコードを解析し，表 1 に示した要素と親子関係で構成される木構造を構築する．次に，構築した 2 つの木構造間で同一の要素を表すものを同定する．同一かどうかの判定は名称が類似しているかどうか（名称の一致）と，対象となる要素周辺の親子関係が類似しているかどうか（構造の一致）を計算することで行う．そして，2 バージョン間における要素の一致度を調べることで，要素の追加，削除，（木構造内での）移動，名称変更を求める．文献 [27] の手法ではこれらの変更情報と各要素の属性についての変更情報を解析することでリファクタリングを検出する．この手法は検出可能なリファクタリングの種類が 33 と他の手法と比べて多いのが特徴である（付録 A）．Xing らは UMLDiff アルゴリズムおよびそれを用いたリファクタリング検出手法を統合開発環境である Eclipse[*1] のプラグインとして実装したツール JDEvAn を公開している [25, 28]．JDEvAn を用いることで，Java 言語で作成されたソースコード変更履歴からリファクタリング

[*1] http://www.eclipse.org/

表 1 UMLDiff が構築する木構造の要素と親子関係

要素の種類	要素が持つ子要素
Virtual root	パッケージ, 配列型
Package	クラスおよびインタフェース
Class	フィールド，メソッド，コンストラクタ，初期化子，内部クラス，クラスが実装するインタフェース
Interface	フィールド，メソッド，内部クラス，インタフェースが実装するインタフェース
Field	初期化子
Block	ローカルクラス，匿名クラス

が検出可能である．

Hayashi らは探索手法を利用したリファクタリングの検出手法を提案している [9]．先に紹介したリファクタリング検出手法は，2 バージョン間のプログラム間の差分を解析し，どのようなリファクタリングが行われたのかを推定する．これに対し，Hayashi らの手法では，変更適用前のバージョンのソースコードにどのようなリファクタリングを適用すると変更後のバージョンになるかを探索することで実施されたリファクタリングを推定する．この手法は，他の手法と比較して複数種類のリファクタリングが実施された変更に対して高精度な推定が可能となっている．

Prete らは論理プログラミング [8] の概念をリファクタリングの検出に導入した手法を提案している [12]．彼らの手法ではプログラムのソースコードと，リファクタリングを論理プログラミングで利用可能な表現に変換する．まず，2 バージョンのソースコードからそれぞれ Logic Fact を抽出し，データベースに登録する．Logic Fact はクラスやメソッドなどのプログラムの構成要素に関する情報と，それらの関係を基に抽出される．次に，リファクタリングが行われた場合に満たす条件を論理プログラミングにおけるルールとして記述しておき，データベースからルールを満たす解を求めることでリファクタリングを検出する．彼らはこれらの手法を実装したツール Ref-Finder を公開している [1, 18]．

2.2. 欠陥混入位置の特定に関する研究

ソフトウェアの欠陥に着目した研究では，多くの場合バグ管理システムと版管理システムに登録された情報を分析する．これらの情報はそれぞれ独立に管理されている場合が多いため，欠陥に関する情報とソースコードの修正に関する情報の対応をとるための手法がこれまでに多く提案されている．Fischer らの方法では，問題報告[*2]とソースコードの修正報告[*3]の双方にバグ ID や修正対象となるファイル名が含まれているかで，欠陥の発生情報とソースコードの修正情報の結び付けを行っている [5]．本研究で利用する Śliwerski らの SZZ アルゴリズムでは，「欠陥が一度でも修正されたとバグ票に記録されているか」「ソースコードの修正を行った開発者が，バグ票でその欠陥修正の担当者となっているか」といった情報を併用している [21]．Wu らが提案するアルゴリズムでは，さらにバグ修正に関するコミットからバグ票に修正されたと報告されるまでの時間や，コミットログとバグ票のコメントとのテキスト類似度を利用している [24]．また，上記のように欠陥の情報とソースコードの修正情報を自動的に関連づける手法の他に，これらの関連づけを手動で行うことを支援するツール LINKSTER が Bird らによって開発されている [2]．

2.3. リファクタリングの効果や役割を調査することを目的とした研究

リファクタリングと欠陥の関係を分析している研究として，Ratzinger らはコミットログを用いてリファクタリングを検出し，欠陥との関係を分析している [19]．彼らは，ある期間におけるリファクタリングの回数が多かった場合，後続する期間において欠陥の発生量が減少すると報告している．そして，リファクタリングは欠陥の混入を予防する効果があると述べている．しかし，彼らは時間に対する詳細な変化については分析していない．

Kim らは 3 つのオープンソースプロジェクトを対象に，ソフトウェア開発においてリファクタリングがどのような役割を果たしているのかを調査した [11]．彼女

[*2] 3.3 節でのバグ票に相当

[*3] 3.3 節でのコミットログに相当

らは，開発履歴からリファクタリングの履歴を復元し，リファクタリングの役割を次の 4 つの観点から評価している．

- リファクタリングの後にバグ修正が行われているか
- リファクタリングは開発者の生産性を向上するか
- リファクタリングはバグ修正を促すか
- ソフトウェアのメジャーリリース前にどの程度リファクタリングが行われているか

彼女らは分析の結果，Eclipse JDT プロジェクトにおいてリファクタリングが行われた 5 リビジョン以内に 26.1% の確率でバグ修正が行われていると報告している．また，リファクタリングによりバグ修正にかかる時間が減少するとも述べている．そして，リファクタリングと共にバグ修正が行われることが多いことと，メジャーリリース直前に多くのリファクタリングが行われていたことを報告している．彼女らは，バグ修正の観点からリファクタリングがソフトウェア開発の品質にどのような影響を与えるかを評価している．本研究では，欠陥の混入という観点からリファクタリングが与える影響を分析する手法を提案し，分析を行っている．

Murphy-Hill らはプログラマがどのようにしてリファクタリングを実施しているかを調査している [16]．彼らは 4 種類の異なる方法で収集されたリファクタリングの実施履歴を用いて 9 つの仮説を検証した．2.1 節でも述べたように，彼らは版管理システムのログメッセージはソフトウェア開発中に実施されたリファクタリングを検出するには信頼性が低いと結論づけている．

3. 提案手法

本研究ではリファクタリングが欠陥混入に与える影響を明らかにすることを目的として，ソフトウェア開発履歴からリファクタリングと欠陥に関する情報を抽出し，定量化する手法を提案する．具体的には，リファクタリングの実施履歴，欠陥の混入時期，欠陥の修正時期を開発履歴からそれぞれ抽出し，それらを用いてリファクタリング頻度，欠陥の混入頻度，欠陥の修正頻度を求める．

本章では 3.1 節において提案手法で扱うソフトウェア開発履歴と提案手法で測定する 3 つの尺度を定義する．3.2 節，3.3 節において各尺度の測定方法について説明する．

3.1. 諸定義

本節では提案手法を説明するにあたり必要となるソフトウェア開発履歴について概説し，版管理システムに記録される情報について形式的な定義を与える．また，提案手法において測定する 3 つの尺度についても定義する．

3.1.1. ソフトウェア開発履歴

ソフトウェア開発プロジェクトにでは，開発者の活動やソースコードなどのプロジェクトに関連する情報を記録する．これは情報を集中管理することで，開発者間で知識共有を促進することを目的としている．多くのソフトウェア開発プロジェクトにおいて記録される情報としては以下のものが挙げられる．

- ソフトウェアを構成するソースコードの変更情報
- 開発者が対応する必要のある活動（機能追加，欠陥修正）に関する情報
- 開発者間のコミュニケーション（E メール，会議の議事録など）

これらの情報の履歴を総称してソフトウェア開発履歴と呼ぶ．

これらの情報の記録を支援することを目的としたシステムはこれまでに多数開発されている．特に，ソースコードの変更情報を記録することに特化したシステムは版管理システムと呼ばれる．また，開発中に発見されたソフトウェアの欠陥（バグ）

に関する情報を記録するシステムはバグ管理システム[*4]と呼ばれる．

本研究で扱うソフトウェア開発プロジェクトは，版管理システム，及びバグ管理システムを用いてソフトウェア開発履歴を記録していることを前提とする．提案手法では版管理システムからリファクタリングの実施履歴を，版管理システムとバグ管理システムを用いて欠陥の混入時期，欠陥の修正時期を抽出する．

3.1.2. 版管理システム

版管理システムは，ソフトウェア開発プロジェクトにおいて，ソフトウェアのソースコードの変更履歴を管理するためのシステムである．版管理システムを用いることで，過去のバージョンの実装（ソースコード）を確認することや，開発者間で最新版のソースコードを共有することが可能である．代表的なものとして，CVS[*5]や Subversion[*6]，Git[*7]，Mercurial[*8]がある．

ソースコードに対して行った変更（行の追加，修正，削除）を版管理システムに記録する作業をコミットという．本研究が対象とするソフトウェア開発プロジェクトにおいては，以下に示す「リビジョン番号」と呼ばれる非負整数を用いて管理していることを想定している．リビジョン番号は，版管理システムにコミットが行われるごとに 1 ずつ加算されていく[*9]．開発者はこのリビジョン番号をもとに，任意の時点におけるソースコードのスナップショットを版管理システムから取得することができる．

本研究では，プロジェクト中のあるリビジョン r のソースコードのスナップショットを次のように定義する．

$$S_r = \{f_{r_1}, f_{r_2}, \ldots\}$$

ここで $f_{r_1}, f_{r_2}, \ldots$ は対応するファイルを表す（以降では特にリビジョン番号を指定してリビジョンを指す際には，S_r を「第 r リビジョン」と呼ぶ．）．次に，第 r リ

[*4] バグ管理システムは，欠陥に加えてソフトウェアに対する要件なども管理する場合には要件管理システムと呼ばれることもある．

[*5] http://www.nongnu.org/cvs/

[*6] http://subversion.apache.org/

[*7] http://git-scm.com/

[*8] http://mercurial.selenic.com/

[*9] 版管理システムに一度もコミットしていないとき，リビジョン番号は 0 である．

ビジョンのファイル f_{r_i} に対して施される変更（ソースコードの追加や削除，編集）を Δ_{r_i} と表す．

あるリビジョン r から次のリビジョン $r+1$ への変更（の集合）op_r を次のように定義する．

$$op_r = \{\Delta_{r_i}, \Delta_{r_{i+1}}, \dots\}$$

3.1.3. バグ管理システム

バグ管理システムは，ソフトウェア開発プロジェクトにおいてソフトウェアの欠陥を記録し，集中管理するためのシステムである．バグ管理システムを用いるプロジェクトでは，発見した欠陥は一度バグ管理システムに登録される．登録時に個々の欠陥には一意に識別するための ID（バグ ID）が割り当てられ，「バグ票」という形で登録される．バグ票には欠陥の発見時刻や発見者，欠陥に関する開発者からのコメントなどが記録されている．バグ管理システムの代表的なものとして，Bugzilla[*10]や Trac[*11]，Redmine[*12]がある．これらのバグ管理システムでは，版管理システムと連携して機能するものがある．ある欠陥の修正のための変更を版管理システムにコミットする際に，コミットログにバグ ID を記述することで，自動的に対応するバグ票の状態を変更することができる．3.3 節で説明する SZZ アルゴリズムでは，コミットログに記述されたバグ ID を手がかりに欠陥の修正を発見する．

3.1.4. リファクタリングと欠陥の関係を調査するための 3 つの尺度

提案手法では，リファクタリングと欠陥の関係を調査するために，リファクタリングの実施頻度，欠陥の混入頻度，欠陥の修正頻度の 3 つの尺度をソフトウェア開発履歴から測定する．版管理システムに記録されたソフトウェアの全リビジョンを V，最新のリビジョン番号を n とすると，$V = [S_1, S_2, ..., S_n]$ と表せる．次に 2 リビジョン間での差分に着目し，3 種類の尺度に対する定義を与える．

まず，op_i においてリファクタリングが行われたかを返す $r(op_i)$ を定義する．$r(op_i)$ は，op_i においてリファクタリングが実施された場合は 1，そうでなければ 0

[*10] http://www.bugzilla.org/
[*11] http://trac.edgewall.org/
[*12] http://www.redmine.org/

を返す．$r(op_i)$ を用いて，第 j リビジョンから第 k リビジョンにおけるリファクタリング頻度 $f_r(j,k)$ を次のように定義する．

$$r(op_i) = \begin{cases} 1 & (\text{if } op_i \text{ contains a refactoring}) \\ 0 & (\text{otherwise}) \end{cases}$$

$$f_r(j,k) = \frac{\sum_{i=j}^{k-1} r(op_i)}{k-j} \quad (j < k, \quad S_j, S_k \in V)$$

同様に，変更 op_i において欠陥が混入されたかを返す $d(op_i)$ と欠陥の混入頻度 $f_d(j,k)$，変更 op_i において欠陥が修正されたかを返す $f(op_i)$ と欠陥の修正頻度 $f_f(j,k)$ を次のように定義する．

$$d(op_i) = \begin{cases} 1 & (\text{if defects are introduced at } v_{i+1}) \\ 0 & (\text{otherwise}) \end{cases}$$

$$f_d(j,k) = \frac{\sum_{i=j}^{k-1} d(op_i)}{k-j} \quad (j < k, \quad S_j, S_k \in V)$$

$$f(op_i) = \begin{cases} 1 & (\text{if defects are fixed at } v_{i+1}) \\ 0 & (\text{otherwise}) \end{cases}$$

$$f_f(j,k) = \frac{\sum_{i=j}^{k-1} f(op_i)}{k-j} \quad (j < k, \quad S_j, S_k \in V)$$

本研究では，$f_r(j,k)$，$f_d(j,k)$，$f_f(j,k)$ の 3 種類の尺度を用いて，リファクタリングの実施と欠陥混入および欠陥修正の関係を示す．

3.2. リファクタリングの実施時期の特定

本研究では，リファクタリングの実施時期を特定するため，2.1 節で紹介した UMLDiff アルゴリズム [26] とそれを用いたリファクタリング検出手法 [27] を利用する．（以降，UMLDiff アルゴリズムと，それを用いたリファクタリング検出手法をまとめて UMLDiff と表記する）

本研究では Xing らのリファクタリング抽出手法と同様に UMLDiff アルゴリズムを用い，下記の手順でソフトウェア開発において実施されたリファクタリングを抽出し，それらの実施時期を特定する．

手順 1 全リビジョンを n リビジョン間隔で等分する．

手順 2 等分したリビジョン集合における最初のリビジョンと最後のリビジョンを入力として UMLDiff を適用する．

手順 3 検出されたリファクタリングの実施時期を版管理システムを用いて調べる．

上記の手順により，第 i リビジョンから第 $i+n$ リビジョンの間において，いつどのようなリファクタリングが行われたかを特定することができる．

3.3. 欠陥の修正・混入時期の特定

ソフトウェアの開発中に発見された欠陥の混入および修正時期を特定するため，本研究では SZZ アルゴリズム [21] を用いる．SZZ アルゴリズムは版管理システムに記録されたコミットログとバグ管理システムに記録された情報を利用する．

SZZ アルゴリズムは，以下の手順で版管理システムの各リビジョンのコミットログを分析し，欠陥の修正時期および混入時期を特定する．

手順 i コミットログにバグ ID の候補となるもの（数字）と修正を示すキーワード（例えば “fixes” や “closed” など）が含まれているかを確認する．含まれていない場合には，以降の手順はスキップして次のコミットログを分析する．

手順 ii i のコミットログに含まれていた番号のバグ ID のバグ票が，バグ管理システム上に登録されているか，またその欠陥が既に修正済みかを確認する．対象となる欠陥が実在し，修正も完了している場合，i のコミットログに対応するコミットが発生した時点で，その欠陥が修正されたとする．欠陥が実在しない，もしくは修正が完了していない場合は，以降の手順はスキップして次のコミットログを分析する．

手順 iii i のコミットログに対応するリビジョンで修正された箇所を，版管理システム上に記録された情報をもとに特定する．

手順 iv ii で確認した欠陥が報告された時点よりも前のコミットを版管理システム上で走査し，iii で特定した箇所が加えられた（すなわち，欠陥が混入した）リビジョンを特定する．

上記のように，手順 ii で欠陥の修正時期が，手順 iv で欠陥の混入時期がそれぞれ特定できる．

表 2 実験対象プロジェクトの概要

種類	開発期間	総リビジョン数	最終 LOC
メールクライアント	2006/7/9 〜 2011/7/11	458	192,941

4. 適用実験

オープンソースソフトウェア Columba[*13]を対象に，前節で述べた手法の適用実験を行った．Columba に関する情報を表 2 に示す．Columba では，版管理システムとして Subversion，バグ管理システムとして SourceForge.net[*14] が標準で提供するバグ管理機能を利用している．UMLDiff を実装した Eclipse プラグイン JDEvAn は，Java 言語を対象としている．また，分析の際には Eclipse のプロジェクトとしてソースコードを読み込む必要がある．Columba のソースコードは Java で書かれており，開発には Eclipse が使用されていため，UMLDiff を適用するのに適してる．

4.1. 実験手順

実験では Columba の開発履歴を対象に以下の手順で 3 章において示した 3 種類の尺度を測定し，リファクタリングと欠陥の混入および修正に関する傾向について調査した．

手順 a UMLDiff を用いてリファクタリングが実施された箇所および時期を特定した．

手順 b SZZ アルゴリズムを用いて欠陥が修正，および混入された箇所を特定した．

手順 c a および b で特定した情報をもとに，25 リビジョン間隔で 3 章において定義した尺度を算出する．

[*13] http://sourceforge.net/projects/columba/

[*14] http://sourceforge.net/

以下に手順 a および 手順 b について詳細を述べる．

手順 a: リファクタリング実施時期の特定

全リビジョンを分割せずに，全リビジョンにおける最初と最後のリビジョンを UMLDiff の入力として，すなわち，第 1 リビジョンと第 458 リビジョンを入力として，リファクタリング箇所の抽出を行った．その後，UMLDiff が抽出した全リファクタリングに対し，各リファクタリングがどの時点で行われたかを Subversion の履歴および差分閲覧機能を利用して手作業で確認した．この際，UMLDiff が誤検出したリファクタリング（例えば，Extract Method を行ったと検出されたが，名前が似ているのみでメソッドの定義自体は全く異なるものであった箇所など）をあわせて除去した．

手順 b: 欠陥修正および混入実施時期の特定

欠陥修正および混入実施時期の特定に関しては，実験用に開発した SZZ アルゴリズムを実装したツールを利用し，欠陥の修正，および混入が発生したリビジョンを特定した．Columba の開発プロジェクトではコミットログに対するルールが厳格に定められており，バグ修正に対応する変更には **[bug]** や **[fix]** という文字列がタグとして記述されている．そこで本実験では，3.3 節での手順 i として上記の文字列が含まれているコミットログのみを分析対象とした．このコミットログに対して，バグ ID が実際にバグ管理システム上に登録されており欠陥修正が対応しているかを確認した．そして確認がとれた場合，このコミットログに対応するコミットが行われた時点を欠陥修正がされた時期とした．さらに，この情報をもとに欠陥が混入した時期についても検出した．

4.2. 実験結果

UMLDiff による誤検出を除いた結果を表 3 に示す．Columba の開発において，13 種類のリファクタリング，計 77 個のリファクタリングを抽出することができた（付録 B）．UMLDiff を用いたリファクタリングの検出には約 7 時間，そこからリファクタリングの実施時期を特定し，誤検出を除く作業は約 12 時間要した．

SZZ アルゴリズムを適用した結果，修正を目的とした変更を 322 個抽出した（付

表 3 検出されたリファクタリング

リファクタリング	個数
Convert top level to inner	1
Die-hard/legacy classes	1
Downcast type parameter	3
Encapsulate field (get)	1
Extract class	2
Extract method	7
Extract subsystem/package	3
Generalize type (method)	9
Generalize type (parameter)	37
Information hiding	6
Inline subsystem/package	1
Pull-up method/field/behaivior	5
Push-down method/field/behaivior	1
合計	77

録 D)．さらに，この変更に関する情報をもとに，欠陥を混入した変更を 243 個抽出した（付録 C)．

これらの情報をもとに，リファクタリング実施頻度，欠陥の混入頻度および欠陥の修正頻度をプロットしたグラフを図 2 に示す．図 2 において横軸はリビジョン番号，縦軸は本研究で定義した 3 種類の尺度を表している．図中の頻度はそれぞれ 25 リビジョンを 1 間隔として計算している．一例として．図 2 において第 250 リビジョンから第 275 リビジョンの区間における欠陥の修正頻度は 0.2 である．

表 4 に UMLDiff アルゴリズムにより検出された Extract Method の一覧を示す．

表では左の列が Extract Method の対象となったソースコードを含んでいたメソッドを表し，右の列が Extract Method リファクタリングにより抽出されたメソッドを表している．一例として，1 行目は AbstractFolder クラスの add メソッドから，同クラスの getCacheStorage メソッドが抽出されたことを表している．

表の 1 行目から 7 行目に着目すると，AbstractFolder クラスにおける別々のメソッドが同一のメソッドとして抽出されたと UMLDiff アルゴリズムが検出していることが読み取れる．手作業による確認の結果，これらの変更は，Extract Method リファクタリングではなく，「自己カプセル化フィールド (Self Encapsulate Field)」リファクタリングであることが分かった．リスト 1，リスト 2 にそれぞれ第 24 リビジョンと第 458 リビジョンにおける AbstractFolder クラスの count メソッドを，リスト 3 に第 458 リビジョンにおける AbstractFolder クラスの getCacheStorage メソッドのソースコードを示す．なお，getCacheStorage メソッドは第 24 リビジョンには存在しない．リスト 1 の 2 行目において AbstractFolder クラスのフィールドである cacheStorage を参照していたコードが，リスト 2 では，getCacheStorage メソッドの呼び出しに変更されている．そして，getChaceStorage メソッドは cacheStorage フィールドを返す処理のみが記述されている．これらのことから，count メソッドの変更と getCacheStorage メソッドの追加は，フィールドのカプセ

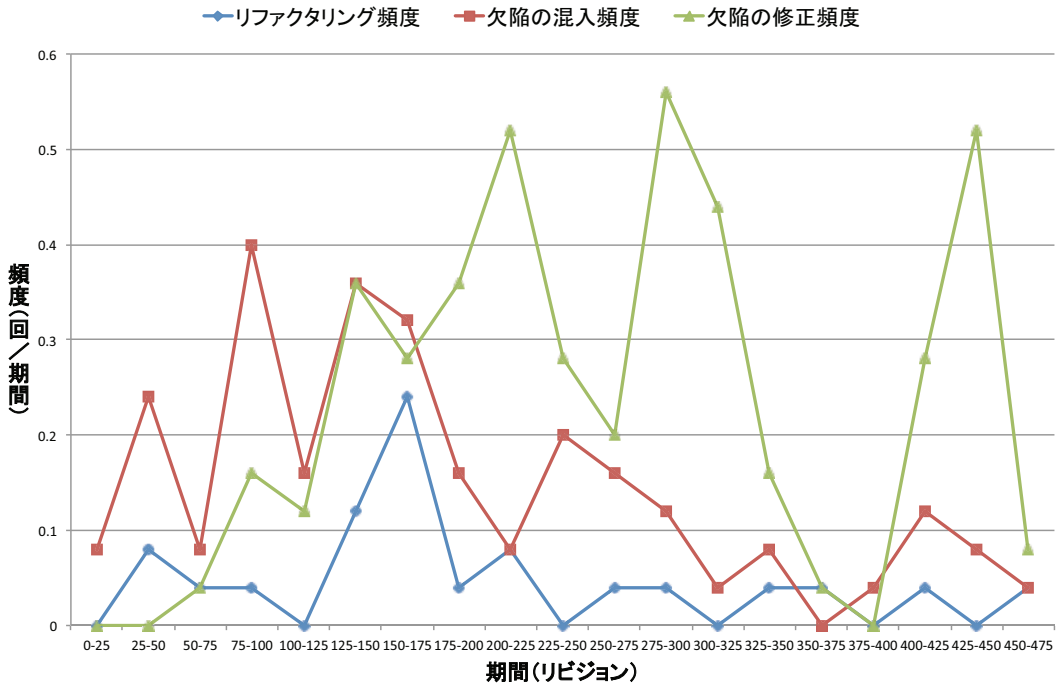

図 2　Columba におけるリファクタリング頻度と欠陥の混入・修正頻度

ル化を目的とした変更であると推測することができる．以上のようにして，表 4 の 1 行目が Extract Method リファクタリングではなく，自己カプセル化フィールドリファクタリングであると結論付けた．また，2 行目から 7 行目についても同様の結論が得られた．実験では，このような UMLDiff アルゴリズムが想定したリファクタリングが，手作業によって異なるリファクタリングであると確認された場合も誤検出として扱った．

次に，UMLDiff アルゴリズムによりリファクタリングを正しく検出できている例を紹介する．表 4 の 8 行目は，VCardParser クラスの read メソッドから，同クラスの fillContactModel メソッドが抽出されたことを表している．リスト 4 に第 24 リビジョンにおける VCardParser クラスのソースコード，リスト 5 に第 458 リビジョンのものを示す．第 24 リビジョンにおいて 100 行あった read メソッドの実装が，第 458 リビジョンでは 15 行になっている．また，第 458 リビジョンでは VCardParser クラスに fillContactModel メソッドが新たに追加されており，read メソッドから呼び出されている（リスト 5 の 186 行目）ことが分かる．以上のことから，第 24 リビジョンから第 458 リビジョンの間に Extract Method リファクタリングが実施されたことが分かる．実験では，このようにして UMLDiff アルゴリズムが検出したリファクタリングが，誤検出でないかを確認した上でリファクタリングが実施された時期の特定を行った．VCardParser クラスにおいては，第 142 リビジョンに fillContactModel が作成され，リファクタリングが実施されていたことが，VCardParser.java ファイルの変更履歴を遡ることで分かった．

リスト 1　AbstractFolder クラスの count メソッド（第 24 リビジョン）

```
1 public int count() throws StoreException {
2         return cacheStorage.count();
3 }
```

リスト 2　AbstractFolder クラスの count メソッド（第 458 リビジョン）

```
1 public int count() throws StoreException {
2         return getCacheStorage().count();
3 }
```

表 4 UMLDiff により検出されたリファクタリング（Extract Method）

リファクタリング対象メソッド	抽出されたメソッド
*[a].addressbook.folder.AbstractFolder.add(IContactModel)	*.addressbook.folder.AbstractFolder.getCacheStorage()
*.addressbook.folder.AbstractFolder.count()	*.addressbook.folder.AbstractFolder.getCacheStorage()
*.addressbook.folder.AbstractFolder.exists(String)	*.addressbook.folder.AbstractFolder.getCacheStorage()
*.addressbook.folder.AbstractFolder.getContactItemMap()	*.addressbook.folder.AbstractFolder.getCacheStorage()
*.addressbook.folder.AbstractFolder.getContactItemMap(String[])	*.addressbook.folder.AbstractFolder.getCacheStorage()
*.addressbook.folder.AbstractFolder.modify(String,IContactModel)	*.addressbook.folder.AbstractFolder.getCacheStorage()
*.addressbook.folder.AbstractFolder.remove(String)	*.addressbook.folder.AbstractFolder.getCacheStorage()
*.addressbook.parser.VCardParser.read(InputStream)	*.addressbook.parser.VCardParser.fillContactModel(Contact)
*.calendar.store.LocalCalendarStore.getComponentInfoList()	*.calendar.store.LocalCalendarStore.getComponentInfoList(String)
*.calendar.ui.calendar.MainCalendarController.viewToday()	*.calendar.ui.calendar.MainCalendarController.getWeekOfYear(Calendar)
*.mail.filter.plugins.AbstractFilterTst.tearDown()	*.mail.filter.plugins.AbstractFilterTst.recursiveDelete(File)
*.mail.folder.virtual.VirtualFolder.VirtualFolder(String,String,String)	*.mail.folder.virtual.VirtualFolder.VirtualFolder(String)
*.mail.folder.virtual.VirtualFolder.applySearch(AbstractMessageFolder,Filter)	*.mail.folder.virtual.VirtualFolder.addFilteredMessage(Object,IMailbox)
*.mail.gui.table.model.TableModelThreadedView.createHashmap(MessageNode)	*.mail.gui.table.model.TableModelThreadedView.addToHashmap(MessageNode)
*.mail.imap.IMAPServer.getLargestRemoteUid(IMAPFolder)	*.mail.imap.IMAPServer.getLargestRemoteUid(IMAPFolder,MailboxStatus)
*.mail.search.MailSearchProvider.query(String,String,int,int)	*.mail.search.MailSearchProvider.createFilterCriteria(String,String)
*.mail.search.SearchFolderFactory.prepareSearchFolder(FilterCriteria,IFolder)	*.mail.search.SearchFolderFactory.createVirtualFolder(IMailFolder)

[a]紙面の都合上，columba.org.columba を*と表記している

リスト 3 AbstractFolder クラスの getCacheStorage メソッド (第 458 リビジョン)

```
1 public ContactItemCacheStorage getCacheStorage() {
2         return cacheStorage;
3 }
```

リスト 4 VCardParser.java (第 24 リビジョン，一部抜粋)

```
1  public class VCardParser {
2
3    /**
4     * Write vcard contact to outpustream.
5     *
6     * @param c
7     * contact data
8     * @param out
9     * outputstream
10    */
11   public static void write(IContactModel c, OutputStream out) {
12           ...
13   }
14
15   /**
16    * Parse vCard contact data from inputstream.
17    *
18    * @param in
19    * inputstream to vCard data
20    * @return contact
21    */
22   public static IContactModel read(InputStream in) {
23         ContactIOFactory ciof = Pim.getContactIOFactory();
24         ContactUnmarshaller unmarshaller = ciof.createContactUnmarshaller();
25         unmarshaller.setEncoding("UTF-8");
26
27         net.wimpi.pim.contact.model.Contact importContact = unmarshaller
28                         .unmarshallContact(in);
29
30         ContactModel c = new ContactModel();
31
32         OrganizationalIdentity organisationalIdentity = importContact
33                         .getOrganizationalIdentity();
34
35
36         if (importContact.hasPersonalIdentity()) {
37                 PersonalIdentity identity = importContact.getPersonalIdentity();
38
39                 // sort-string
40                 c.setSortString(identity.getSortString());
41
42                 // list of nick names
43                 if (identity.getNicknameCount() > 0)
44                         c.setNickName(ParserUtil.getStringOfArray(identity
45                                         .listNicknames(), ","));
46
47                 // list of prefixes
48                 if (identity.listPrefixes().length > 0)
49                         c.setNamePrefix(ParserUtil.getStringOfArray(identity
50                                         .listPrefixes(), ","));
51
52                 c.setFamilyName(identity.getLastname());
53                 c.setGivenName(identity.getFirstname());
54
55                 // list of additional names (middle names)
56                 if (identity.listAdditionalNames().length > 0)
57                         c.setAdditionalNames(ParserUtil.getStringOfArray(identity
58                                         .listAdditionalNames(), ","));
59
```

```
60                  // list of suffices
61                  if (identity.listSuffixes().length > 0)
62                          c.setNameSuffix(ParserUtil.getStringOfArray(identity
63                                  .listSuffixes(), ","));
64
65                  // formatted name
66                  c.setFormattedName(identity.getFormattedName());
67
68                  // birthday
69                  Date birthday = importContact.getPersonalIdentity().getBirthDate();
70                  if ( birthday != null)
71                          c.setBirthday(birthday);
72
73          }
74
75          // url to website/homepage
76          c.setHomePage(importContact.getURL());
77
78          // email addresses
79          if (importContact.hasCommunications()) {
80                  Communications communications = importContact.getCommunications();
81
82                  Iterator it = communications.getEmailAddresses();
83                  while (it.hasNext()) {
84                          EmailAddress adr = (EmailAddress) it.next();
85                          c.addEmail(new EmailModel(adr.getAddress(),
86                                  EmailModel.TYPE_WORK));
87                  }
88          }
89
90          // address list
91          if ( importContact.listAddresses().length > 0 ) {
92                  // not that the editor ui only supports max of 3 addresses to edit
93                  Address[] addresses = importContact.listAddresses();
94                  for ( int i=0; i<addresses.length; i++) {
95                          Address a = addresses[i];
96
97                          int type = -1;
98                          if ( a.isHome())
99                                  type = AddressModel.TYPE_HOME;
100                         else if ( a.isWork())
101                                 type = AddressModel.TYPE_WORK;
102                         else
103                                 type = AddressModel.TYPE_OTHER;
104
105
106                         AddressModel adr = new AddressModel(a.getPostBox(), a.getStreet(),
                                    a.getCity(), a.getPostalCode(), a.getRegion(), a.getCountry(), a.
                                getLabel(), type );
107
108                         c.addAddress(adr);
109                 }
110         }
111
112         // name of organisation
113         c.setOrganisation(organisationalIdentity.getOrganization().getName());
114
115         c.setTitle(organisationalIdentity.getTitle());
116         c.setProfession(organisationalIdentity.getRole());
117
118         c.setNote(importContact.getNote());
119
120         return c;
121   }
122 }
```

リスト 5 VCardParser.java（第 458 リビジョン，一部抜粋）

```
1  /**
2   * Contact data parser for a vCard−standard compliant text/plain file.
3   * <p>
4   * It makes use of the jpim library. Its not really a wrapper. It only
5   * creates a mapping between the jpim data model and our data model.
6   *
7   * @author fdietz
8   */
9  public class VCardParser {
10
11     /**
12      * Write vcard contact to outpustream.
13      *
14      * @param c
15      * contact data
16      * @param out
17      * outputstream
18      */
19     public static void write(IContactModel c, OutputStream out) {
20             ...
21     }
22
23     /**
24      * Fill the contact model from the import contact
25      *
26      * @param importContact import contact
27      *
28      * @return contact model
29      *
30      */
31     public static ContactModel fillContactModel(net.wimpi.pim.contact.model.Contact
           importContact) {
32         ContactModel c = new ContactModel();
33
34         OrganizationalIdentity organisationalIdentity = importContact.
               getOrganizationalIdentity();
35
36
37         if (importContact.hasPersonalIdentity()) {
38             PersonalIdentity identity = importContact.getPersonalIdentity();
39
40             // sort−string
41             c.setSortString(identity.getSortString());
42
43             // list of nick names
44             if (identity.getNicknameCount() > 0) {
45                 c.setNickName(ParserUtil.getStringOfArray(identity.listNicknames(), ","));
46             }
47
48             // list of prefixes
49             if (identity.listPrefixes().length > 0) {
50                 c.setNamePrefix(ParserUtil.getStringOfArray(identity.listPrefixes(), ","));
51             }
52
53             c.setFamilyName(identity.getLastname());
54             c.setGivenName(identity.getFirstname());
55
56             // list of additional names (middle names)
57             if (identity.listAdditionalNames().length > 0) {
58                 c.setAdditionalNames(ParserUtil.getStringOfArray(identity.
                       listAdditionalNames(), ","));
59             }
60
61             // list of suffices
62             if (identity.listSuffixes().length > 0) {
63                 c.setNameSuffix(ParserUtil.getStringOfArray(identity.listSuffixes(), ","));
64             }
65
```

```
66              // formatted name
67              c.setFormattedName(identity.getFormattedName());
68
69              // birthday
70              Date birthday = importContact.getPersonalIdentity().getBirthDate();
71              if (birthday != null) {
72                  c.setBirthday(birthday);
73              }
74
75          }
76
77          // url to website/homepage
78          c.setHomePage(importContact.getURL());
79
80          // email addresses and phone numbers
81          if (importContact.hasCommunications()) {
82              Communications communications = importContact.getCommunications();
83
84              Iterator it = communications.getEmailAddresses();
85              while (it.hasNext()) {
86                  EmailAddress adr = (EmailAddress) it.next();
87
88                  int type = EmailModel.TYPE_WORK;
89                  if (adr.isType("HOME")) {
90                      type = EmailModel.TYPE_HOME;
91                  } else if (adr.isType("WORK")) {
92                      type = EmailModel.TYPE_WORK;
93                  } else if (adr.isType("OTHER")) {
94                      type = EmailModel.TYPE_OTHER;
95                  }
96
97                  c.addEmail(new EmailModel(adr.getAddress(),
98                          type));
99              }
100
101             it = communications.getPhoneNumbers();
102             while (it.hasNext()) {
103                 PhoneNumber phone = (PhoneNumber) it.next();
104
105                 int type = PhoneModel.TYPE_BUSINESS_PHONE;
106                 if (phone.isCar()) {
107                     type = PhoneModel.TYPE_CAR_PHONE;
108                 } else if (phone.isCellular()) {
109                     type = PhoneModel.TYPE_MOBILE_PHONE;
110                 } else if (phone.isFax()) {
111                     type = PhoneModel.TYPE_HOME_FAX;
112                 } else if (phone.isHome()) {
113                     type = PhoneModel.TYPE_HOME_PHONE;
114                 } else if (phone.isISDN()) {
115                     type = PhoneModel.TYPE_ISDN;
116                 } else if (phone.isPager()) {
117                     type = PhoneModel.TYPE_PAGER;
118                 } else if (phone.isWork()) {
119                     type = PhoneModel.TYPE_BUSINESS_PHONE;
120                 }
121
122                 c.addPhone(new PhoneModel(phone.getNumber(), type));
123             }
124         }
125
126         // address list
127         if (importContact.listAddresses().length > 0) {
128             // not that the editor ui only supports max of 3 addresses to edit
129             Address[] addresses = importContact.listAddresses();
130             for (int i = 0; i < addresses.length; i++) {
131                 Address a = addresses[i];
132
133                 int type = -1;
134                 if (a.isHome()) {
```

```
135                     type = AddressModel.TYPE_HOME;
136                 } else if (a.isWork()) {
137                     type = AddressModel.TYPE_WORK;
138                 } else {
139                     type = AddressModel.TYPE_OTHER;
140                 }
141
142
143                 AddressModel adr = new AddressModel(a.getPostBox(), a.getStreet(), a.
                        getCity(), a.getPostalCode(), a.getRegion(), a.getCountry(), a.getLabel(),
                         type);
144
145                 c.addAddress(adr);
146             }
147         }
148
149         // name of organisation
150         if (organisationalIdentity != null) {
151             if (organisationalIdentity.hasOrganization()) {
152                 c.setOrganisation(organisationalIdentity.getOrganization().getName());
153             }
154
155             c.setTitle(organisationalIdentity.getTitle());
156             c.setProfession(organisationalIdentity.getRole());
157         }
158
159         c.setNote(importContact.getNote());
160
161         // dummy address
162         if (!c.getEmailIterator().hasNext()) {
163             c.addEmail(new EmailModel("", EmailModel.TYPE_WORK));
164         }
165
166         return c;
167     }
168
169     /**
170      * Parse vCard contact data from inputstream.
171      *
172      * @param in
173      * inputstream to vCard data
174      * @return contact
175      */
176     public static IContactModel[] read(InputStream in) {
177         ContactIOFactory ciof = Pim.getContactIOFactory();
178         ContactUnmarshaller unmarshaller = ciof.createContactUnmarshaller();
179         unmarshaller.setEncoding("UTF-8");
180
181         net.wimpi.pim.contact.model.Contact[] importContacts = unmarshaller.
                unmarshallContacts(in);
182
183         ContactModel[] contacts = new ContactModel[importContacts.length];
184
185         for (int i = 0; i < importContacts.length; i++) {
186             contacts[i] = fillContactModel(importContacts[i]);
187         }
188
189         return contacts;
190     }
191 }
```

5. 考察

5.1. 実験結果からの考察

図 2 からは次に示す二つの傾向が見られる．

(a) リファクタリングが最も実施された第 150 リビジョンから第 175 リビジョンの区間以降は，欠陥の混入頻度が減少傾向にある．
(b) プロジェクト全体を通して，リファクタリングの実施頻度が高くなった後に，欠陥の修正頻度が高くなる傾向にある．

(a) に関して，第 0 リビジョンから第 175 リビジョンの区間における欠陥の混入頻度の平均が 0.23 であるのに対して，第 175 リビジョンから 第 458 リビジョンの区間においては 0.09 となっている．また，リファクタリング頻度が最も高い第 150 リビジョンから第 175 リビジョンの区間におけるコミットログを確認すると，第 151 リビジョンから第 156 リビジョンにおいて大規模なリファクタリングを実施したと記録されていた．このリファクタリングにより，ソースコードの品質が向上したため以降の欠陥混入が減少したのではないかと考える．しかし，初期バージョンの開発が完了し，プロジェクトが保守工程に移行したために，頻繁な機能追加が行われず，結果として欠陥の混入が減少したという見方も可能である．

(b) に関しては，リファクタリングを実施したためソースコードの可読性が向上し，それに伴い欠陥の発見・修正が容易になったため，欠陥の修正頻度も高くなったのではないかと考える．欠陥の修正を行うにあたり，開発者らがソースコードの変更を容易にするため，あらかじめリファクタリングを実施した後に欠陥の修正を行った可能性もある．一方，リファクタリングを実施したが失敗し，失敗箇所の修正を行ったため，結果としてリファクタリングの後に欠陥の修正が行われたという場合も考えられる．

5.2. 妥当性の検証

分析の結果から得られた傾向は，あくまで単一プロジェクトから得られた知見である．そのため，別のオープンソースソフトウェアプロジェクトでは別の傾向が得られる可能性がある．

今回の実験では，リファクタリングの実施時期を特定する際に，全リビジョンを分割せずに一度の UMLDiff でリファクタリングの抽出を試みた．このため，開発途中に作成されたファイルやクラスに対して実施されたリファクタリングや，最終バージョンでは削除されていたコードに対して実施されたリファクタリングなどは検出されていない可能性がある．また，実験結果は UMLDiff の適合率及び再現率の影響を強く受けていると考えられる．UMLDiff による設計情報の差分抽出については，適合率が 93.7%，再現率が 95.2% [*15] と共に高い値が報告されており，実験結果に与えている影響は小さいと考えられる [26]．リファクタリングの検出については文献 [27] においてオープンソースソフトウェアプロジェクトに適用した場合の検出数が報告されている．しかし，適合率，再現率の値については言及されていない．適合率に関しては，今回の実験においては誤検出を目視で排除したため影響は小さいと考えられる．一方で，UMLDiff によって抽出されたリファクタリングが網羅していない可能性がある．

対象プロジェクトにおける欠陥の修正，欠陥の混入が行われた時期を特定するにあたり利用した SZZ アルゴリズムは，一部の欠陥の修正，欠陥の混入を抽出できていない可能性がある．欠陥の修正に関しては，今回はコミットログをもとに特定したため，開発者らの記録の厳密さにその再現性は左右されている．また，欠陥の混入に関しては，SZZ アルゴリズムが欠陥修正の際に変更されたコードを版管理システムを用いて遡ることで混入時期を特定しているが，コードの削除のみが原因で欠陥の混入が行われた場合は混入時期を特定することができない．しかし，このような例は稀であると考えられるため，これにより今回得られた傾向が大きく変わるものではないと考える．

提案手法では 3 種類の頻度を求めるにあたり，各リビジョンにおいて，リファク

[*15] 再現率は文献 [26] における Table 5 中の *#Correct* と *#Reported* の比率として計算した．

タリング，欠陥の混入，欠陥の修正がそれぞれ行われたかどうかに着目している．このため，同一リビジョン内で複数種類のリファクタリングを実施した場合や，複数の欠陥を同時に修正した場合も，単一のリファクタリングや欠陥修正と同様に扱う．これは，提案手法で抽出可能なリファクタリングの粒度が大小様々であり，それら異なるリファクタリングの実施について，同じ 1 回のリファクタリングであると考えるのは不適切であると考えたため，リファクタリングを実施したかどうかのみに着目した．欠陥の混入・修正に関しても同様に，粒度の大小が存在するため欠陥の混入・修正が行われたかどうかのみに着目した．

6. おわりに

本研究では，ソフトウェア開発におけるリファクタリングが欠陥混入に与える影響を分析するために，リファクタリングの実施頻度，欠陥の混入頻度，欠陥の修正頻度を定義した．また，それぞれの尺度をソフトウェア開発履歴から収集する手法を提案した．そして，オープンソースソフトウェアを対象にリファクタリングの実施頻度，欠陥の混入・修正頻度を測定し，それらの関係を調査した．Columba からは，リファクタリングが最も行われた期間以降，欠陥の混入頻度が減少傾向にあったこと，プロジェクト全体を通してリファクタリング頻度が高くなった後に欠陥の修正頻度が高くなる傾向にあることが分かった．

今後の展開として，より多くのオープンソースソフトウェアに対して手法を適用するとともに，今回適用したプロジェクトよりも規模の大きなプロジェクトを対象に調査を行い，Columba に見られた傾向と比較することが考えられる．これにより，ソフトウェア開発において広く適用できる有用な知見を得ることができると期待される．しかし，提案手法を複数のプロジェクトに適用する際にはいくつか考慮しなければならない点がある．適用実験において，リファクタリングの実施時期を特定するのに約 19 時間を要している．UMLDiff によるリファクタリングの検出は利用しているツールの特性上，対象プロジェクトの規模が大きくなるほど時間を必要とする．また，今回の実験ではリファクタリングの実施時期を手作業で特定しているため，そのままでは他のソフトウェア開発プロジェクトに適用するのが難しい．更に，Columba より規模の大きいプロジェクトになると分析が現実的な時間で終わらなくなると考えられる．提案手法を複数のプロジェクトに適用するにあたっては，より短時間で 2 バージョン間からリファクタリングの検出を行うツールの開発と，リファクタリングの実施時期の特定を自動化することが求められる．

今回の調査では，リファクタリングや欠陥の混入・修正がどのファイルに実施されたかを考慮せずに，その頻度に着目して分析を行った．リファクタリングが欠陥混入に与える影響をより詳細に調査するには，リファクタリングが実施されたファイルが，リファクタリングによりその後の欠陥混入がなくなったかどうかなど，ファイルに着目して分析を行う必要がある．この際には，より正確にリファク

タリングの実施履歴を抽出することが求められるため，リファクタリングの実施時期特定におけるリビジョンの分割数を増やし，UMLDiff が抽出するリファクタリングの再現性を向上させる必要がある．

謝辞

本研究を進めるにあたり，多くの方々に御指導，御協力，御支援を頂きました．ここに謝意を添えて御名前を記させて頂きます．

奈良先端科学技術大学院大学 情報科学研究科 ソフトウェア設計学研究室 飯田 元 教授には，本研究の全過程において熱心な御指導を賜りました．研究方針だけではなく，研究に対する姿勢，論文執筆，発表方法についても多くの御助言を頂きました．心より厚く御礼を申し上げます．

奈良先端科学技術大学院大学 情報科学研究科 ソフトウェア工学研究室 松本 健一 教授には，様々な場面で本研究に対し貴重な御指導，御助言を賜りました．心より感謝申し上げます．

奈良先端科学技術大学院大学 情報科学研究科 ユビキタスコンピューティングシステム研究室 安本 慶一 教授には，様々な場面で本研究に対し貴重な御指導，御助言を賜りました．心より感謝申し上げます．

奈良先端科学技術大学院大学 情報科学研究科 ソフトウェア設計学研究室 吉田 則裕 助教には，本研究を進めるにあたり，広範囲かつ多大な御助力を頂きました．特に，学会発表や論文投稿時に貴重な御助言を頂戴いたしました．心より感謝申し上げます．

奈良先端科学技術大学院大学 情報科学研究科 ソフトウェア設計学研究室 伏田 享平 特任助教には，本研究を進めるにあたり，広範囲かつ多大な御協力を頂きました．また，入学前に大阪府立工業高等専門学校（現 大阪府立大学工業高等専門学校）において非常勤講師をなさっていた際は，私が本学へ入学しソフトウェア設計学研究室に所属するきっかけを与えて頂きました．心より感謝申し上げます．

奈良先端科学技術大学院大学 情報科学研究科 ソフトウェア設計学講座 川口 真司 助教（現 有人宇宙システム株式会社），名倉 正剛 特任助教（現 株式会社日立製作所）には，入学前に大阪府立工業高等専門学校からインターンシップで御世話になった際に，貴重な御指導を賜りました．インターンシップで得られた知識，経験は入学後の研究活動および大学生活を円滑に進めるにあたり非常に助けとなりました．心より感謝申し上げます．

奈良先端科学技術大学院大学 情報科学研究科 ソフトウェア設計学講座 大蔵 君治 博士（現 株式会社サイバード）には，本学在籍時に本研究を進めるにあたり，広範囲かつ多大な御助力を頂きました．また，大学生活への姿勢や後輩への接し方について多くのアドバイスを頂戴しました．心より感謝申し上げます．

奈良先端科学技術大学院大学 情報科学研究科 修了生である高田 純 氏，山科 隆伸 氏， 石田 響子 氏，奥村 哲也 氏，片山 真一 氏，西田 皓司 氏，福島 義彦 氏，水野 恵祐 氏，小山 貴和子 氏，山本 瑞起 氏にはインターンシップの際に大変御世話になりました．充実したインターンシップを送ることができたのは先輩方のおかげです．心より感謝申し上げます．

奈良先端科学技術大学院大学 情報科学研究科 修了生である木下 正喬 氏，オユンビレグ チングン 氏，大澤 直哉 氏，髙井 雄治 氏，藤田 将司 氏，佐々木 辰也 氏，坂東 祐司 氏，木村 直人 氏には先輩方が本学に在学されている頃から研究との関係の有無に拘わらず，様々な相談に乗って頂きました．また，常日頃より楽しい時間を過ごさせて頂きました．心より感謝申し上げます．

奈良先端科学技術大学院大学 川本 理恵 氏，呂 悠妃 氏には研究の遂行に必要な事務処理など，多岐にわたり御助力頂きました．心より感謝申し上げます．

奈良先端科学技術大学院大学情報科学研究科ソフトウェア設計学研究室，ならびにソフトウェア工学研究室，言語科学研究室の皆様には，日頃より多大な御協力と御助言を頂き，公私ともに支えて頂きました．ありがとうございました．

2010 年度 IT Spiral 高度ソフトウェア技術者育成プログラムの同期の方々には，プログラム受講時における活動のみならず，プログラム修了後も私生活で楽しい時間を過ごさせて頂きました．ありがとうございました．

最後に，大学院への進学に理解を示し，私を支えてくれた両親，普段から私の話に耳を傾けてくれた祖父，そしてこれまで共に過ごしてきた兄姉に心より深く感謝します．

参考文献

[1] Ref-Finder – A tool to detect refactorings between program versions. `https://webspace.utexas.edu/kp9746/www/reffinder/`.

[2] Christian Bird, Adrian Bachmann, Foyzur Rahman, and Abraham Bernstein. LINKSTER: Enabling Efficient Manual Inspection and Annotation of Mined Data. In *Proceedings of the 18th ACM SIGSOFT International Symposium on Foundations of Software Engineering(FSE 2010)*, pp. 369–370, 2010.

[3] Marat Boshernitsan, Susan L. Graham, and Marti A. Hearst. Aligning Development Tools with the Way Programmers Think About Code Changes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems(CHI 2007)*, pp. 567–576, 2007.

[4] Danny Dig, Kashif Manzoor, Ralph Johnson, and Tien N. Nguyen. Refactoring-Aware Configuration Management for Object-Oriented Programs. In *Proceedings of the 29th International Conference on Software Engineering(ICSE 2007)*, pp. 427–436, 2007.

[5] Michael Fischer, Martin Pinzger, and Harald Gall. Analyzing and Relating Bug Report Data for Feature Tracking. In *Proceedings of the 10th Working Conference on Reverse Engineering(WCRE 2003)*, pp. 90–99, 2003.

[6] Martin Fowler. Refactoring Home Page. `http://refactoring.com/`.

[7] Martin Fowler. *Refactoring: improving the design of exsiting code.* Addison Wesley, 1999.

[8] Michael Gelfond and Nicola Leone. Logic programming and knowledge representation–The A-Prolog perspective. *Artificial Intelligence*, Vol. 138, No. 1–2, pp. 3–38, 2002.

[9] Shinpei Hayashi, Yasuyuki Tsuda, and Motoshi Saeki. Search-Based Refactoring Detection from Source Code Revisions. *IEICE Transactions on Information and Systems*, Vol. E93-D, No. 4, pp. 754–762, 2010.

[10] Toshihiro Kamiya, Shinji Kusumoto, and Katsuro Inoue. Ccfinder: a multilinguistic token-based code clone detection system for large scale source code. *IEEE Transaction on Software Engineering*, Vol. 28, pp. 654–670, July 2002.

[11] Miryung Kim, Dongxiang Cai, and Sunghun Kim. An Empirical Investigation into the Role of API-Level Refactorings during Software Evolution. In *Proceedings of the 33rd International Conference on Software Engineering(ICSE 2011)*, pp. 151–160, 2011.

[12] Miryung Kim, Matthew Gee, Alex Loh, and Napol Rachatasumrit. RefFinder: A Refactoring Reconstruction Tool based on Logic Query Templates. In *Proceedings of the 18th ACM SIGSOFT International Symposium on Foundations of Software Engineering(FSE 2010)*, pp. 371–372, 2010.

[13] Gail C. Murphy, Mik Kersten, and Leah Findlater. How Are Java Software Developers Using the Eclipse IDE? *IEEE Software*, Vol. 23, pp. 76–83, July 2006.

[14] Emerson Murphy-Hill and Andrew P. Black. Breaking the Barriers to Successful Refactoring: Observations and Tools for Extract Method. In *Proceedings of the 30th International Conference on Software Engineering(ICSE 2008)*, pp. 421–430, 2008.

[15] Emerson Murphy-Hill, Andrew P. Black, Danny Dig, and Chris Parnin. Gathering Refactoring Data: a Comparison of Four Methods. In *Proceedings of the 2nd ACM Workshop on Refactoring Tools(WRT 2008)*, pp. 1–5, 2008.

[16] Emerson Murphy-Hill, Chris Parnin, and Andrew P. Black. How We Refactor, and How We Know it. In *Proceedings of the 31st International Conference on Software Engineering(ICSE 2009)*, pp. 287–297, 2009.

[17] Markus Pizka. Straightening Spaghetti-Code with Refactoring? In *Proceedings of the 2004 International Conference on Software Engineering Research and Practice(SERP 2004)*, pp. 846–852, 2004.

[18] Kyle Prete, Napol Rachatasumrit, Nikita Sudan, and Miryung Kim. Template-based Reconstruction of Complex Refactorings. In *Proceedings of the 2010 IEEE International Conference on Software Maintenance(ICSM 2010)*, pp. 1–10, 2010.

[19] Jacek Ratzinger, Thomas Sigmund, and Harald C. Gall. On the Relation of Refactorings and Software Defect Prediction. In *Proceedings of the 5th Working Conference on Mining Software Repositories(MSR 2008)*, pp. 35–38, 2008.

[20] Romain Robbes and Michele Lanza. SpyWare:A Change-Aware Development Toolset. In *Proceedings of the 30th International Conference on Software Engineering(ICSE 2008)*, pp. 847–850, 2008.

[21] Jacek Śliwerski, Thomas Zimmermann, and Andreas Zeller. When do changes induce fixes? In *Proceedings of the 2nd International Workshop on Mining Software Repositories(MSR 2005)*, pp. 1–5, 2005.

[22] Konstantinos Stroggylos and Diomidis Spinellis. Refactoring–Does It Improve Software Quality? In *Proceedings of the 5th International Workshop on Software Quality(WoSQ 2007)*, pp. 10–16, 2007.

[23] Peter Weißgerber and Stephan Diehl. Identifying Refactorings from Source-Code Changes. In *Proceedings of the 21st IEEE/ACM International Conference on Automated Software Engineering(ASE 2006)*, pp. 231–240, 2006.

[24] Rongxin Wu, Hongyu Zhang, Sunghun Kim, and Shing-Chi Cheung. ReLink: Recovering Links between Bugs and Changes. In *Proceedings of the 8th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering(ESEC/FSE 2011)*, pp. 15–25, 2011.

[25] Zhenchang Xing. JDEvAn. `http://webdocs.cs.ualberta.ca/~stroulia/Zhenchang_Xing_Old_Home/jdevan.html`.

[26] Zhenchang Xing and Eleni Stroulia. UMLDiff: an algorithm for object-oriented design differencing. In *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering(ASE 2005)*, pp.

54–65, 2005.

[27] Zhenchang Xing and Eleni Stroulia. Refactoring Detection based on UMLDiff Change-Facts Queries. In *Proceedings of the 13th Working Conference on Reverse Engineering(WCRE 2006)*, pp. 263–274, 2006.

[28] Zhenchang Xing and Eleni Stroulia. The JDEvAn Tool Suite in Support of Object-Oriented Evolutionary Development. In *Proceedings of Companion of the 30th International Conference on Software Engineering(ICSE Companion 2008)*, pp. 951–952, 2008.

付録

A. UMLDiff により検出可能なリファクタリング一覧

Convert inner type to top-level	Convert top-level to inner
Extract subsystem	Inline subsystem
Extract package	Inline package
Pull-up method/field	Push-down method/field
Pull-up behavior	Push-down behavior
Pull-up constructor body	Extract interface
Extract superclass	Extract subclass
Inline superclass	Inline subclass
Form template method	Replace inheritance with delegation
Replace delegation with inheritance	Extract class
Inline class	Die-hard/legacy classes
Convert anonymous class to nested	Move method/field
Move behavior	Deprecation + delegation
Information hiding	Generalize type
Downcast type	Introduce factory method
Introduce parameter object	Encapsulate field
Preserve whole object	

B. 適用実験により得られた Columba におけるリファクタリング

ファイル名	リビジョン番号
/columba/trunk/calendar/src/main/java/org/columba/calendar/store/LocalCalendarStore.java	128
/columba/trunk/calendar/src/main/java/org/columba/calendar/store/StoreEventDelegator.java	424
/columba/trunk/contact/src/main/java/org/columba/addressbook/folder/AbstractFolder.java	350
/columba/trunk/contact/src/main/java/org/columba/addressbook/folder/AbstractFolder.java	350
/columba/trunk/contact/src/main/java/org/columba/addressbook/folder/LocalFolder.java	350
/columba/trunk/contact/src/main/java/org/columba/addressbook/folder/XmlDataStorage.java	350
/columba/trunk/contact/src/main/java/org/columba/addressbook/parser/VCardParser.java	142
/columba/trunk/core/src/main/java/org/columba/core/facade/ServiceFacadeRegistry.java	193
/columba/trunk/core/src/main/java/org/columba/core/filter/AbstractFilter.java	151
/columba/trunk/core/src/main/java/org/columba/core/filter/AbstractFilterAction.java	151
/columba/trunk/core/src/main/java/org/columba/core/filter/Filter.java	151
/columba/trunk/core/src/main/java/org/columba/core/filter/FilterCriteria.java	154
/columba/trunk/core/src/main/java/org/columba/core/filter/FilterList.java	151,154
/columba/trunk/core/src/main/java/org/columba/core/filter/FilterRule.java	151,154
/columba/trunk/core/src/main/java/org/columba/core/gui/action/AbstractColumbaAction.java	210
/columba/trunk/core/src/main/java/org/columba/core/gui/profiles/ProfileManager.java	424
/columba/trunk/core/src/main/java/org/columba/core/services/Service.java	193
/columba/trunk/core/src/main/java/org/columba/core/services/ServiceRegistry.java	193
/columba/trunk/core-api/src/main/java/org/columa/core/config/IDefaultItem.java	153
/columba/trunk/core-api/src/main/java/org/columba/api/gui/IAbstractColumbaAction.java	210,211
/columba/trunk/core-api/src/main/java/org/columba/core/filter/IFilterCriteria.java	154
/columba/trunk/core-api/src/main/java/org/columba/core/filter/IFilterList.java	154
/columba/trunk/core-api/src/main/java/org/columba/core/filter/IFilterRule.java	154
/columba/trunk/core-api/src/main/java/org/columba/core/folder/api/IFolder.java	35
/columba/trunk/core-api/src/main/java/org/columba/core/folder/api/IFolderCommandReference.java	35

ファイル名	リビジョン番号
/columba/trunk/mail/src/main/java/org/columba/mail/filter/FilterCompoundCommand.java	151
/columba/trunk/mail/src/main/java/org/columba/mail/filter/MailFilterAction.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/MailFilterCriteria.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/AccountFilter.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/AddressbookFilter.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/BodyFilter.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/ColorFilter.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/ColorMessageFilterAction.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/CopyMessageAction.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/DateFilter.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/DeleteMessageAction.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/FlagsFilter.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/HeaderfieldFilter.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/MarkMessageAsReadAction.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/MatchAllFilter.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/MoveMessageAction.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/PriorityFilter.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/ScoreMessageFilterAction.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/SizeFilter.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/folder/AbstractMessageFolder.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/folder/search/DefaultSearchEngine.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/folder/search/DummyQueryEngine.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/folder/search/IMAPQueryEngine.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/folder/search/LuceneQueryEngine.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/folder/search/QueryEngine.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/folder/virtual/VirtualFolder.java	34
/columba/trunk/mail/src/main/java/org/columba/mail/folder/virtual/VirtualHeader.java	34
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/AbstractEditorController.java	262
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/ComposerController.java	262
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/html/HtmlEditorController2.java	262

ファイル名	リビジョン番号
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/html/HtmlEditorController3.java	262
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/text/TextEditorController.java	262
/columba/trunk/mail/src/main/java/org/columba/mail/gui/config/filter/plugins/DefaultActionRow.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/gui/config/filter/plugins/MarkActionRow.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/gui/table/model/TableModelThreadedView.java	292
/columba/trunk/mail/src/main/java/org/columba/mail/imap/IImapServer.java	155
/columba/trunk/mail/src/main/java/org/columba/mail/imap/IMAPServer.java	155,453
/columba/trunk/mail/src/main/java/org/columba/mail/search/MailSearchProvider.java	34,57,94
/columba/trunk/mail/src/main/java/org/columba/mail/search/SearchFolderFactory.java	34,57,134
/columba/trunk/mail/src/test/java/org/columba/mail/filter/FilterListTest.java	156
/columba/trunk/mail/src/test/java/org/columba/mail/filter/plugins/AbstractFilterTst.java	158
/columba/trunk/mail-api/src/main/java/org/columba/mail/folder/IMailbox.java	155
/columba/trunk/plugins/org.columba.mail.PlaySoundFilterAction/src/org/columba/mail/filter/plugins/PlaySoundFilterAction.java	156

C. 適用実験により得られた Columba における欠陥の混入

ファイル名	リビジョン番号
/mail/src/main/java/org/columba/mail/pgp/MultipartEncryptedRenderer.java	23
/dist/win32/columba_setup.iss	22
/mail/src/main/java/org/columba/mail/composer/MessageBuilderHelper.java	23
/mail/src/main/java/org/columba/mail/filter/plugins/CopyMessageAction.java	23
/core/src/main/java/org/columba/core/gui/themes/plugin/MacLookAndFeelPlugin.java	22
/mail/src/main/java/org/columba/mail/pop3/command/AddPOP3MessageCommand.java	23
/mail/src/test/java/org/columba/mail/folder/FolderTstHelper.java	23
/mail/src/main/java/org/columba/mail/smtp/SMTPServer.java	23
/mail/src/main/java/org/columba/mail/folder/command/CopyMessageCommand.java	23
/lib.properties	92
/plugins/org.columba.chat.altura/src/org/columba/chat/ui/action/AddContactAction.java	24
/contact/src/test/java/org/columba/addressbook/folder/GetHeaderItemListTest.java	22
/mail/src/main/java/org/columba/mail/gui/composer/util/ExternalEditor.java	23
/mail/src/main/java/org/columba/mail/gui/tree/command/CreateSubFolderCommand.java	23
/mail/src/main/java/org/columba/mail/gui/composer/html/action/UnderlineFormatAction.java	23
/core/src/main/resources/org/columba/core/i18n/dialog/tagging.properties	140
/contact/src/main/java/org/columba/addressbook/gui/dialog/group/EditGroupDialog.java	22
/plugins/org.columba.chat.altura/src/org/columba/chat/command/AddContactCommand.java	24
/mail/src/main/java/org/columba/mail/gui/composer/command/ForwardInlineCommand.java	23,329,408
/mail/src/main/java/org/columba/mail/folder/command/SaveMessageSourceAsCommand.java	23
/core/src/main/java/org/columba/core/gui/tagging/EditTagAction.java	124,130,157
/mail/src/main/java/org/columba/mail/gui/composer/SignatureView.java	23
/calendar/src/main/java/org/columba/calendar/command/AddEventCommand.java	22
/mail/src/main/java/org/columba/mail/gui/composer/command/RedirectCommand.java	23
/calendar/src/main/java/org/columba/calendar/ui/action/SaveAsAction.java	22

ファイル名	リビジョン番号
/mail/src/main/java/org/columba/mail/imap/IMAPServer.java	23
/contact/src/main/java/org/columba/addressbook/gui/table/model/FilterDecorator.java	22
/core/src/main/java/org/columba/core/help/HelpManager.java	22
/calendar/src/main/java/org/columba/calendar/command/CopyEventCommand.java	22
/mail/src/main/java/org/columba/mail/gui/config/account/IncomingServerPanel.java	23
/mail/src/main/java/org/columba/mail/gui/composer/contact/FolderComboBox.java	23
/mail/src/main/java/org/columba/mail/gui/table/TableView.java	23
/mail/src/main/java/org/columba/mail/composer/MessageComposer.java	23
/contact/src/test/java/org/columba/addressbook/folder/RemoveContactTest.java	22
/core/src/main/java/org/columba/core/desktop/ColumbaDesktop.java	22
/core/src/main/java/org/columba/core/print/cPrintVariable.java	22
/mail/src/main/java/org/columba/mail/gui/composer/command/ReplyToAllCommand.java	23
/mail/src/main/java/org/columba/mail/gui/message/command/ViewMessageCommand.java	23,34,95,169,201
/core/src/main/java/org/columba/core/main/ColumbaCmdLineParser.java	22
/mail/src/test/java/org/columba/mail/filter/plugins/AbstractFilterTst.java	23
/mail/src/main/java/org/columba/mail/gui/tagging/MailTagList.java	126,130,140,168
/core/src/main/java/org/columba/core/gui/themes/ThemeSwitcher.java	22
/calendar/src/main/java/org/columba/calendar/command/ActivityMovedCommand.java	22
/calendar/src/main/java/org/columba/calendar/command/DeleteEventCommand.java	22
/src.properties	24
/calendar/src/main/java/org/columba/calendar/store/LocalCalendarStore.java	22,128
/mail/src/main/java/org/columba/mail/folder/AbstractLocalFolder.java	28,35
/mail/src/main/java/org/columba/mail/pop3/POP3Store.java	23
/calendar/src/main/java/org/columba/calendar/ui/action/ImportCalendarAction.java	22
/mail/src/main/java/org/columba/mail/shutdown/ClearRecentFlagPlugin.java	23
/mail/src/main/java/org/columba/mail/gui/composer/util/SubjectDialog.java	23
/calendar/src/main/java/org/columba/calendar/ui/frame/CalendarFrameMediator.java	124
/mail/src/main/java/org/columba/mail/folder/command/SaveMessageBodyAsCommand.java	23
/core/src/main/java/org/columba/core/gui/dialog/DateChooserDialog.java	22
/contact/src/main/java/org/columba/addressbook/model/PhoneModel.java	22

ファイル名	リビジョン番号
/mail/src/main/java/org/columba/mail/filter/plugins/MoveMessageAction.java	23
/contact/src/main/java/org/columba/addressbook/gui/frame/AddressbookFrameController.java	124
/core/src/main/java/org/columba/core/gui/globalactions/PluginManagerAction.java	22
/contact/src/main/java/org/columba/addressbook/folder/importfilter/VCardImporter.java	22,142
/contact/src/main/java/org/columba/addressbook/gui/table/model/AddressbookTableModel.java	22
/calendar/src/main/resources/org/columba/calendar/i18n/global.properties	22
/core/src/main/java/org/columba/core/gui/docking/DockingPanel.java	22
/contact/src/test/java/org/columba/addressbook/folder/AddContactTest.java	22
/core/src/main/java/org/columba/core/gui/tagging/RemoveTagAction.java	124
/plugins/org.columba.chat.altura/src/org/columba/chat/ui/action/OpenConversationAction.java	24
/calendar/src/main/java/org/columba/calendar/facade/DialogFacade.java	173
/mail/src/test/java/org/columba/mail/filter/FilterListTest.java	23
/calendar/src/main/java/org/columba/calendar/parser/XCSDocumentParser.java	22
/core/src/main/java/org/columba/core/gui/tagging/EditTagDialog.java	132
/calendar/src/main/java/org/columba/calendar/ui/action/CopyActivityMenu.java	425
/mail/src/main/java/org/columba/mail/gui/table/model/TableModelThreadedView.java	23
/contact/src/main/java/org/columba/addressbook/folder/LocalRootFolder.java	22
/native/win32/Columba.lap	23
/mail/src/main/java/org/columba/mail/folder/FolderFactory.java	23
/core/src/main/java/org/columba/core/versioninfo/VersionInfo.java	22
/calendar/src/main/java/org/columba/calendar/ui/action/ExportCalendarAction.java	22
/core/src/main/java/org/columba/core/gui/docking/TitleBar.java	22
/mail/src/main/java/org/columba/mail/folder/mh/MHDataStorage.java	23
/contact/src/main/java/org/columba/addressbook/folder/AddressbookTreeNode.java	22
/mail/src/main/java/org/columba/mail/folder/command/ExportFolderCommand.java	23
/core/src/main/java/org/columba/core/desktop/JDICDesktop.java	22
/core/src/main/java/org/columba/core/gui/plugin/PluginManagerDialog.java	22
/calendar/src/main/java/org/columba/calendar/parser/DateParser.java	22
/calendar/src/main/java/org/columba/calendar/command/MoveEventCommand.java	22
/mail/src/main/java/org/columba/mail/gui/composer/ComposerModel.java	23,262,422

ファイル名	リビジョン番号
/core/src/main/java/org/columba/core/gui/base/ShadowBorder.java	22
/contact/src/main/java/org/columba/addressbook/folder/XmlDataStorage.java	22
/core/src/test/java/org/columba/core/associations/AssociationStoreCases.java	87
/mail/src/main/java/org/columba/mail/gui/config/folder/FolderOptionsDialog.java	23
/mail/src/main/java/org/columba/mail/gui/tree/TreeController.java	23
/mail/src/main/java/org/columba/mail/mailchecking/IMAPMailCheckingAction.java	23
/core/src/main/java/org/columba/core/gui/tagging/TagList.java	130,157
/core/src/test/java/org/columba/core/tagging/TagManagerTest.java	79
/mail/src/main/java/org/columba/mail/gui/tree/action/AbstractMoveFolderAction.java	23
/core/src/main/java/org/columba/core/gui/themes/plugin/PlasticLookAndFeelPlugin.java	22
/mail/src/main/java/org/columba/mail/gui/message/command/OpenAttachmentCommand.java	23
/mail/src/main/java/org/columba/mail/gui/composer/ComposerController.java	23,139,262,273
/contact/src/test/java/org/columba/addressbook/folder/GetContactTest.java	22
/mail/src/main/java/org/columba/mail/gui/composer/SubjectController.java	23
/contact/src/main/java/org/columba/addressbook/parser/VCardParser.java	22,142
/build.xml	24,79,82,83,86,93,102,180,244,278
/mail/src/main/java/org/columba/mail/main/MessageOptionParser.java	23
/mail/src/main/java/org/columba/mail/gui/composer/AccountController.java	23
/core/src/main/java/org/columba/core/xml/XmlNewIO.java	22
/mail/src/test/java/org/columba/mail/gui/util/AddressListRendererTest.java	293
/mail/src/test/java/org/columba/mail/filter/plugins/FlagsFilterTest.java	23
/mail/src/main/java/org/columba/mail/search/MailSearchProvider.java	23,34,35,103,94
/calendar/src/main/java/org/columba/calendar/command/SaveEventToFileCommand.java	22
/mail/src/main/resources/org/columba/mail/plugin/plugin.xml	23
/mail/src/main/java/org/columba/mail/gui/message/viewer/TextViewer.java	23,58
/mail/src/main/java/org/columba/mail/gui/composer/AccountView.java	23
/plugins/org.columba.chat.altura/src/org/columba/chat/ui/action/RemoveContactAction.java	24
/mail/src/test/java/org/columba/mail/parser/ListParserTest.java	23,399
/mail/src/main/java/org/columba/mail/gui/config/filter/plugins/DefaultCriteriaRow.java	23
/mail/src/main/java/org/columba/mail/gui/filtertoolbar/FilterToolbar.java	23,35

ファイル名	リビジョン番号
/contact/src/main/java/org/columba/addressbook/gui/table/model/ContactItemTableModel.java	22
/mail/src/main/java/org/columba/mail/gui/frame/ThreePaneMailFrameController.java	23,43,87,124,243,267,
/contact/src/main/java/org/columba/addressbook/model/GroupModel.java	22
/contact/src/test/java/org/columba/addressbook/parser/VCardParserTest.java	22
/mail/src/main/java/org/columba/mail/main/MailMain.java	23,155,179,193,202,242,254,317
/mail/src/main/java/org/columba/mail/folder/virtual/VirtualFolder.java	23,34,35,57,155,168,235
/mail/src/main/java/org/columba/mail/gui/tree/util/CreateFolderDialog.java	23
/mail/src/main/java/org/columba/mail/folder/headercache/BerkeleyDBHeaderList.java	28
/core/src/main/java/org/columba/core/desktop/MacDesktop.java	194
/contact/src/main/java/org/columba/addressbook/gui/table/TableMouseListener.java	22
/contact/src/main/java/org/columba/addressbook/folder/RemoteRootFolder.java	22
/core/src/main/java/org/columba/core/main/Bootstrap.java	22,77,
/mail/src/main/java/org/columba/mail/gui/util/AddressListRenderer.java	23
/mail/src/main/java/org/columba/mail/gui/table/SubjectTreeRenderer.java	23
/core/src/main/java/org/columba/core/gui/base/ComboMenu.java	22
/mail/src/main/java/org/columba/mail/gui/message/viewer/IMimePartViewer.java	23
/mail/src/main/java/org/columba/mail/gui/tree/action/RemoveFolderAction.java	23
/mail/src/main/java/org/columba/mail/pop3/POP3Server.java	28
/mail/src/main/java/org/columba/mail/gui/composer/html/HtmlToolbar.java	23
/mail/src/main/java/org/columba/mail/gui/config/filter/plugins/DateCriteriaRow.java	23
/mail/src/main/java/org/columba/mail/gui/config/general/MailOptionsDialog.java	23
/core/src/main/resources/org/columba/core/plugin/plugin.xml	22,29
/core/src/main/java/org/columba/core/gui/htmlviewer/HTMLViewerFactory.java	22
/mail/src/main/java/org/columba/mail/gui/tagging/TagFolderFactory.java	87
/contact/src/main/java/org/columba/addressbook/folder/AbstractFolder.java	22,45,289
/mail/src/main/java/org/columba/mail/facade/DialogFacade.java	23
/mail/src/main/java/org/columba/mail/parser/ListBuilder.java	23
/mail/src/test/java/org/columba/mail/imap/TestServer.java	23
/mail/src/main/java/org/columba/mail/gui/config/subscribe/SynchronizeFolderListCommand.java	23
/contact/src/main/java/org/columba/addressbook/gui/dialog/contact/ContactEditorDialog.java	22

ファイル名	リビジョン番号
/contact/src/main/java/org/columba/addressbook/gui/table/model/TableModelDecorator.java	22
/mail/src/main/java/org/columba/mail/smtp/command/SendMessageCommand.java	23
/core/src/main/java/org/columba/core/gui/tagging/AddTagAction.java	124,130
/mail/src/main/java/org/columba/mail/gui/composer/SubjectView.java	23
/plugins/org.columba.mail.PlaySoundFilterAction/src/org/columba/mail/filter/plugins/PlaySoundFilterAction.java	24
/mail/src/main/java/org/columba/mail/folder/command/AddSenderToAddressbookCommand.java	23
/mail/src/main/java/org/columba/mail/gui/message/command/SaveAttachmentAsCommand.java	23
/contact/src/main/java/org/columba/addressbook/folder/GroupFolder.java	22,326
/core/src/main/java/org/columba/core/gui/docking/DockableView.java	22
/contact/src/test/java/org/columba/addressbook/folder/ModifyContactTest.java	22
/plugins/org.columba.chat.altura/src/org/columba/chat/command/ConnectCommand.java	24
/core/src/main/java/org/columba/core/main/Main.java	101
/mail/src/test/java/org/columba/mail/folder/AbstractFolderTst.java	23
/core/src/main/java/org/columba/core/gui/plugin/PluginTreeTableModel.java	22
/mail/src/main/java/org/columba/mail/folder/search/DefaultSearchEngine.java	23
/calendar/src/main/java/org/columba/calendar/ui/action/EditActivityAction.java	22
/mail/src/main/java/org/columba/mail/gui/config/account/OutgoingServerPanel.java	23
/calendar/src/main/java/org/columba/calendar/ui/dialog/EditEventDialog.java	22,432
/core/src/main/java/org/columba/core/gui/util/StartUpFrame.java	22
/mail/src/main/java/org/columba/mail/gui/action/NewMessageAction.java	23
/mail/src/main/java/org/columba/mail/gui/table/TableController.java	23,145
/mail/src/main/java/org/columba/mail/search/SearchFolderFactory.java	134
/mail/src/main/java/org/columba/mail/parser/ListParser.java	23
/mail/src/main/java/org/columba/mail/gui/message/viewer/MessageParser.java	23
/plugins/org.columba.chat.altura/src/org/columba/chat/ui/roaster/RoasterTree.java	24
/mail/src/test/java/org/columba/mail/gui/composer/command/AbstractComposerTst.java	23
/mail/src/main/java/org/columba/mail/gui/composer/command/ReplyCommand.java	23,329,408
/calendar/src/main/java/org/columba/calendar/ui/box/CalendarBox.java	164
/mail/src/main/java/org/columba/mail/folder/imap/IMAPFolder.java	28,35,158
/mail/src/main/java/org/columba/mail/gui/table/model/HeaderTableModel.java	23

ファイル名	リビジョン番号
/core/src/main/java/org/columba/core/gui/tagging/TaggingMenu.java	126
/mail/src/main/java/org/columba/mail/folder/AbstractMessageFolder.java	23,158
/contact/src/main/java/org/columba/addressbook/folder/ContactItemCacheStorageImpl.java	22,233
/contact/src/main/java/org/columba/addressbook/model/ContactModelXMLFactory.java	22
/mail/src/main/java/org/columba/mail/gui/message/MessageController.java	23
/mail/src/main/java/org/columba/mail/gui/tree/action/CreateVirtualFolderAction.java	23
/mail/src/main/java/org/columba/mail/gui/tree/util/SelectSearchFolderDialog.java	451
/dist/webstart/columba.jnlp	22
/mail/src/main/java/org/columba/mail/gui/composer/PriorityView.java	23
/contact/src/main/java/org/columba/addressbook/gui/action/AddVCardAction.java	22,142
/mail/src/main/java/org/columba/mail/gui/tree/util/FolderTreeCellRenderer.java	23,35
/core-api/src/main/java/org/columba/core/filter/IFilterAction.java	154
/mail/src/main/java/org/columba/mail/gui/composer/action/SendAction.java	23
/mail/src/main/java/org/columba/mail/gui/composer/action/SendLaterAction.java	23
/mail/src/main/java/org/columba/mail/gui/composer/command/ReplyWithTemplateCommand.java	23
/core/src/main/java/org/columba/core/gui/util/AboutDialog.java	22
/mail/src/main/java/org/columba/mail/gui/composer/TextEditorPanel.java	23
/mail/src/main/java/org/columba/mail/config/AccountList.java	23
/mail/src/main/java/org/columba/mail/gui/composer/HeaderController.java	23
/mail/src/main/java/org/columba/mail/folder/mailboximport/AbstractMailboxImporter.java	23
/calendar/src/main/java/org/columba/calendar/command/ImportCalendarCommand.java	22
/contact/src/main/java/org/columba/addressbook/parser/ParserUtil.java	22
/mail/src/main/java/org/columba/mail/gui/contact/list/ContactDNDListView.java	23
/contact/src/main/java/org/columba/addressbook/folder/LocalFolder.java	22
/mail/src/main/java/org/columba/mail/parser/text/HtmlParser.java	23

D. 適用実験により得られた Columba における欠陥の修正

ファイル名	リビジョン番号
/columba/trunk/build.xml	80,81,110,123,180,182,246,278,279
/columba/trunk/calendar/src/main/java/org/columba/calendar/command/ActivityMovedCommand.java	199
/columba/trunk/calendar/src/main/java/org/columba/calendar/command/AddEventCommand.java	199
/columba/trunk/calendar/src/main/java/org/columba/calendar/command/CopyEventCommand.java	199
/columba/trunk/calendar/src/main/java/org/columba/calendar/command/DeleteEventCommand.java	199
/columba/trunk/calendar/src/main/java/org/columba/calendar/command/ImportCalendarCommand.java	199
/columba/trunk/calendar/src/main/java/org/columba/calendar/command/MoveEventCommand.java	199
/columba/trunk/calendar/src/main/java/org/columba/calendar/command/SaveEventToFileCommand.java	199
/columba/trunk/calendar/src/main/java/org/columba/calendar/facade/DialogFacade.java	199
/columba/trunk/calendar/src/main/java/org/columba/calendar/parser/DateParser.java	407
/columba/trunk/calendar/src/main/java/org/columba/calendar/parser/XCSDocumentParser.java	407
/columba/trunk/calendar/src/main/java/org/columba/calendar/store/LocalCalendarStore.java	264,429
/columba/trunk/calendar/src/main/java/org/columba/calendar/ui/action/CopyActivityMenu.java	430
/columba/trunk/calendar/src/main/java/org/columba/calendar/ui/action/EditActivityAction.java	199
/columba/trunk/calendar/src/main/java/org/columba/calendar/ui/action/ExportCalendarAction.java	199
/columba/trunk/calendar/src/main/java/org/columba/calendar/ui/action/ImportCalendarAction.java	199
/columba/trunk/calendar/src/main/java/org/columba/calendar/ui/action/SaveAsAction.java	199
/columba/trunk/calendar/src/main/java/org/columba/calendar/ui/box/CalendarBox.java	199
/columba/trunk/calendar/src/main/java/org/columba/calendar/ui/dialog/EditEventDialog.java	432,440
/columba/trunk/calendar/src/main/java/org/columba/calendar/ui/frame/CalendarFrameMediator.java	177
/columba/trunk/calendar/src/main/resources/org/columba/calendar/i18n/global.properties	189
/columba/trunk/contact/src/main/java/org/columba/addressbook/folder/AbstractFolder.java	289,350
/columba/trunk/contact/src/main/java/org/columba/addressbook/folder/AddressbookTreeNode.java	350
/columba/trunk/contact/src/main/java/org/columba/addressbook/folder/ContactItemCacheStorageImpl.java	289,350
/columba/trunk/contact/src/main/java/org/columba/addressbook/folder/GroupFolder.java	350

ファイル名	リビジョン番号
/columba/trunk/contact/src/main/java/org/columba/addressbook/folder/LocalFolder.java	350
/columba/trunk/contact/src/main/java/org/columba/addressbook/folder/LocalRootFolder.java	350
/columba/trunk/contact/src/main/java/org/columba/addressbook/folder/RemoteRootFolder.java	350
/columba/trunk/contact/src/main/java/org/columba/addressbook/folder/XmlDataStorage.java	350
/columba/trunk/contact/src/main/java/org/columba/addressbook/folder/importfilter/VCardImporter.java	142,350
/columba/trunk/contact/src/main/java/org/columba/addressbook/gui/action/AddVCardAction.java	142,350
/columba/trunk/contact/src/main/java/org/columba/addressbook/gui/dialog/contact/ContactEditorDialog.java	142
/columba/trunk/contact/src/main/java/org/columba/addressbook/gui/dialog/group/EditGroupDialog.java	199
/columba/trunk/contact/src/main/java/org/columba/addressbook/gui/frame/AddressbookFrameController.java	177
/columba/trunk/contact/src/main/java/org/columba/addressbook/gui/table/TableMouseListener.java	142
/columba/trunk/contact/src/main/java/org/columba/addressbook/gui/table/model/AddressbookTableModel.java	303
/columba/trunk/contact/src/main/java/org/columba/addressbook/gui/table/model/ContactItemTableModel.java	303
/columba/trunk/contact/src/main/java/org/columba/addressbook/gui/table/model/FilterDecorator.java	303
/columba/trunk/contact/src/main/java/org/columba/addressbook/gui/table/model/TableModelDecorator.java	303
/columba/trunk/contact/src/main/java/org/columba/addressbook/model/ContactModelXMLFactory.java	286
/columba/trunk/contact/src/main/java/org/columba/addressbook/model/GroupModel.java	284
/columba/trunk/contact/src/main/java/org/columba/addressbook/model/PhoneModel.java	142
/columba/trunk/contact/src/main/java/org/columba/addressbook/parser/ParserUtil.java	350
/columba/trunk/contact/src/main/java/org/columba/addressbook/parser/VCardParser.java	142,170
/columba/trunk/contact/src/test/java/org/columba/addressbook/folder/AddContactTest.java	158
/columba/trunk/contact/src/test/java/org/columba/addressbook/folder/GetContactTest.java	158
/columba/trunk/contact/src/test/java/org/columba/addressbook/folder/GetHeaderItemListTest.java	158
/columba/trunk/contact/src/test/java/org/columba/addressbook/folder/ModifyContactTest.java	158
/columba/trunk/contact/src/test/java/org/columba/addressbook/folder/RemoveContactTest.java	158
/columba/trunk/contact/src/test/java/org/columba/addressbook/parser/VCardParserTest.java	142
/columba/trunk/core-api/src/main/java/org/columba/core/filter/IFilterAction.java	199
/columba/trunk/core/src/main/java/org/columba/core/desktop/ColumbaDesktop.java	135
/columba/trunk/core/src/main/java/org/columba/core/desktop/JDICDesktop.java	199,255
/columba/trunk/core/src/main/java/org/columba/core/desktop/MacDesktop.java	196
/columba/trunk/core/src/main/java/org/columba/core/gui/base/ComboMenu.java	296

ファイル名	リビジョン番号
/columba/trunk/core/src/main/java/org/columba/core/gui/base/ShadowBorder.java	214
/columba/trunk/core/src/main/java/org/columba/core/gui/dialog/DateChooserDialog.java	200
/columba/trunk/core/src/main/java/org/columba/core/gui/docking/DockableView.java	214
/columba/trunk/core/src/main/java/org/columba/core/gui/docking/DockingPanel.java	214
/columba/trunk/core/src/main/java/org/columba/core/gui/docking/TitleBar.java	214
/columba/trunk/core/src/main/java/org/columba/core/gui/globalactions/PluginManagerAction.java	197
/columba/trunk/core/src/main/java/org/columba/core/gui/htmlviewer/HTMLViewerFactory.java	199
/columba/trunk/core/src/main/java/org/columba/core/gui/plugin/PluginManagerDialog.java	197
/columba/trunk/core/src/main/java/org/columba/core/gui/plugin/PluginTreeTableModel.java	220
/columba/trunk/core/src/main/java/org/columba/core/gui/tagging/AddTagAction.java	199
/columba/trunk/core/src/main/java/org/columba/core/gui/tagging/EditTagAction.java	157,199
/columba/trunk/core/src/main/java/org/columba/core/gui/tagging/EditTagDialog.java	133
/columba/trunk/core/src/main/java/org/columba/core/gui/tagging/RemoveTagAction.java	199
/columba/trunk/core/src/main/java/org/columba/core/gui/tagging/TagList.java	157,177
/columba/trunk/core/src/main/java/org/columba/core/gui/tagging/TaggingMenu.java	157
/columba/trunk/core/src/main/java/org/columba/core/gui/themes/ThemeSwitcher.java	199,228
/columba/trunk/core/src/main/java/org/columba/core/gui/themes/plugin/MacLookAndFeelPlugin.java	110
/columba/trunk/core/src/main/java/org/columba/core/gui/themes/plugin/PlasticLookAndFeelPlugin.java	159
/columba/trunk/core/src/main/java/org/columba/core/gui/util/AboutDialog.java	149
/columba/trunk/core/src/main/java/org/columba/core/gui/util/StartUpFrame.java	405
/columba/trunk/core/src/main/java/org/columba/core/help/HelpManager.java	199
/columba/trunk/core/src/main/java/org/columba/core/main/Bootstrap.java	110,114,202
/columba/trunk/core/src/main/java/org/columba/core/main/ColumbaCmdLineParser.java	202
/columba/trunk/core/src/main/java/org/columba/core/main/Main.java	110
/columba/trunk/core/src/main/java/org/columba/core/print/cPrintVariable.java	325
/columba/trunk/core/src/main/java/org/columba/core/versioninfo/VersionInfo.java	406
/columba/trunk/core/src/main/java/org/columba/core/xml/XmlNewIO.java	280
/columba/trunk/core/src/main/resources/org/columba/core/i18n/dialog/tagging.properties	157
/columba/trunk/core/src/main/resources/org/columba/core/plugin/plugin.xml	60
/columba/trunk/core/src/test/java/org/columba/core/associations/AssociationStoreCases.java	157

ファイル名	リビジョン番号
/columba/trunk/core/src/test/java/org/columba/core/tagging/TagManagerTest.java	157
/columba/trunk/dist/webstart/columba.jnlp	143
/columba/trunk/dist/win32/columba_setup.iss	80
/columba/trunk/lib.properties	231
/columba/trunk/mail/src/main/java/org/columba/mail/composer/MessageBuilderHelper.java	291,444
/columba/trunk/mail/src/main/java/org/columba/mail/composer/MessageComposer.java	421
/columba/trunk/mail/src/main/java/org/columba/mail/config/AccountList.java	227
/columba/trunk/mail/src/main/java/org/columba/mail/facade/DialogFacade.java	95
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/CopyMessageAction.java	199
/columba/trunk/mail/src/main/java/org/columba/mail/filter/plugins/MoveMessageAction.java	199
/columba/trunk/mail/src/main/java/org/columba/mail/folder/AbstractLocalFolder.java	158
/columba/trunk/mail/src/main/java/org/columba/mail/folder/AbstractMessageFolder.java	158,168,235
/columba/trunk/mail/src/main/java/org/columba/mail/folder/FolderFactory.java	235
/columba/trunk/mail/src/main/java/org/columba/mail/folder/command/AddSenderToAddressbookCommand.java	181,307
/columba/trunk/mail/src/main/java/org/columba/mail/folder/command/CopyMessageCommand.java	199
/columba/trunk/mail/src/main/java/org/columba/mail/folder/command/ExportFolderCommand.java	199
/columba/trunk/mail/src/main/java/org/columba/mail/folder/command/SaveMessageBodyAsCommand.java	199,213
/columba/trunk/mail/src/main/java/org/columba/mail/folder/command/SaveMessageSourceAsCommand.java	199
/columba/trunk/mail/src/main/java/org/columba/mail/folder/headercache/BerkeleyDBHeaderList.java	158
/columba/trunk/mail/src/main/java/org/columba/mail/folder/imap/IMAPFolder.java	158,446,449
/columba/trunk/mail/src/main/java/org/columba/mail/folder/mailboximport/AbstractMailboxImporter.java	199
/columba/trunk/mail/src/main/java/org/columba/mail/folder/mh/MHDataStorage.java	181
/columba/trunk/mail/src/main/java/org/columba/mail/folder/search/DefaultSearchEngine.java	442
/columba/trunk/mail/src/main/java/org/columba/mail/folder/virtual/VirtualFolder.java	168,235,320
/columba/trunk/mail/src/main/java/org/columba/mail/gui/action/NewMessageAction.java	137,199
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/AccountController.java	329
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/AccountView.java	329
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/ComposerController.java	135,136,273,329
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/ComposerModel.java	202,329,422,423
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/HeaderController.java	199,421

ファイル名	リビジョン番号
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/PriorityView.java	422
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/SignatureView.java	135
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/SubjectController.java	329
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/SubjectView.java	329
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/TextEditorPanel.java	135
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/action/SendAction.java	329
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/action/SendLaterAction.java	329
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/command/ForwardInlineCommand.java	329,408,444
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/command/RedirectCommand.java	329
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/command/ReplyCommand.java	329,408,444
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/command/ReplyToAllCommand.java	138,302
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/command/ReplyWithTemplateCommand.java	329
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/contact/FolderComboBox.java	181
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/html/HtmlToolbar.java	135
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/html/action/UnderlineFormatAction.java	135
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/util/ExternalEditor.java	199
/columba/trunk/mail/src/main/java/org/columba/mail/gui/composer/util/SubjectDialog.java	329
/columba/trunk/mail/src/main/java/org/columba/mail/gui/config/account/IncomingServerPanel.java	199,222
/columba/trunk/mail/src/main/java/org/columba/mail/gui/config/account/OutgoingServerPanel.java	199,222
/columba/trunk/mail/src/main/java/org/columba/mail/gui/config/filter/plugins/DateCriteriaRow.java	200
/columba/trunk/mail/src/main/java/org/columba/mail/gui/config/filter/plugins/DefaultCriteriaRow.java	296
/columba/trunk/mail/src/main/java/org/columba/mail/gui/config/folder/FolderOptionsDialog.java	450
/columba/trunk/mail/src/main/java/org/columba/mail/gui/config/general/MailOptionsDialog.java	221
/columba/trunk/mail/src/main/java/org/columba/mail/gui/config/subscribe/SynchronizeFolderListCommand.java	447
/columba/trunk/mail/src/main/java/org/columba/mail/gui/contact/list/ContactDNDListView.java	288
/columba/trunk/mail/src/main/java/org/columba/mail/gui/filtertoolbar/FilterToolbar.java	203
/columba/trunk/mail/src/main/java/org/columba/mail/gui/frame/ThreePaneMailFrameController.java	177,321
/columba/trunk/mail/src/main/java/org/columba/mail/gui/message/MessageController.java	439
/columba/trunk/mail/src/main/java/org/columba/mail/gui/message/command/OpenAttachmentCommand.java	199
/columba/trunk/mail/src/main/java/org/columba/mail/gui/message/command/SaveAttachmentAsCommand.java	199

ファイル名	リビジョン番号
/columba/trunk/mail/src/main/java/org/columba/mail/gui/message/command/ViewMessageCommand.java	169,199,201,217
/columba/trunk/mail/src/main/java/org/columba/mail/gui/message/viewer/IMimePartViewer.java	439
/columba/trunk/mail/src/main/java/org/columba/mail/gui/message/viewer/MessageParser.java	439,441
/columba/trunk/mail/src/main/java/org/columba/mail/gui/message/viewer/TextViewer.java	199,439
/columba/trunk/mail/src/main/java/org/columba/mail/gui/table/SubjectTreeRenderer.java	282
/columba/trunk/mail/src/main/java/org/columba/mail/gui/table/TableController.java	145,218
/columba/trunk/mail/src/main/java/org/columba/mail/gui/table/TableView.java	199,266
/columba/trunk/mail/src/main/java/org/columba/mail/gui/table/model/HeaderTableModel.java	292
/columba/trunk/mail/src/main/java/org/columba/mail/gui/table/model/TableModelThreadedView.java	283,292,331
/columba/trunk/mail/src/main/java/org/columba/mail/gui/tagging/MailTagList.java	235
/columba/trunk/mail/src/main/java/org/columba/mail/gui/tagging/TagFolderFactory.java	235
/columba/trunk/mail/src/main/java/org/columba/mail/gui/tree/TreeController.java	266
/columba/trunk/mail/src/main/java/org/columba/mail/gui/tree/action/AbstractMoveFolderAction.java	234
/columba/trunk/mail/src/main/java/org/columba/mail/gui/tree/action/CreateVirtualFolderAction.java	235
/columba/trunk/mail/src/main/java/org/columba/mail/gui/tree/action/RemoveFolderAction.java	199
/columba/trunk/mail/src/main/java/org/columba/mail/gui/tree/command/CreateSubFolderCommand.java	199
/columba/trunk/mail/src/main/java/org/columba/mail/gui/tree/util/CreateFolderDialog.java	199
/columba/trunk/mail/src/main/java/org/columba/mail/gui/tree/util/FolderTreeCellRenderer.java	272
/columba/trunk/mail/src/main/java/org/columba/mail/gui/tree/util/SelectSearchFolderDialog.java	452
/columba/trunk/mail/src/main/java/org/columba/mail/gui/util/AddressListRenderer.java	305
/columba/trunk/mail/src/main/java/org/columba/mail/imap/IMAPServer.java	199,449
/columba/trunk/mail/src/main/java/org/columba/mail/mailchecking/IMAPMailCheckingAction.java	446
/columba/trunk/mail/src/main/java/org/columba/mail/main/MailMain.java	202,317,322
/columba/trunk/mail/src/main/java/org/columba/mail/main/MessageOptionParser.java	202
/columba/trunk/mail/src/main/java/org/columba/mail/parser/ListBuilder.java	281
/columba/trunk/mail/src/main/java/org/columba/mail/parser/ListParser.java	421
/columba/trunk/mail/src/main/java/org/columba/mail/parser/text/HtmlParser.java	437
/columba/trunk/mail/src/main/java/org/columba/mail/pgp/MultipartEncryptedRenderer.java	199
/columba/trunk/mail/src/main/java/org/columba/mail/pop3/POP3Server.java	158
/columba/trunk/mail/src/main/java/org/columba/mail/pop3/POP3Store.java	199

ファイル名	リビジョン番号
/columba/trunk/mail/src/main/java/org/columba/mail/pop3/command/AddPOP3MessageCommand.java	204,300
/columba/trunk/mail/src/main/java/org/columba/mail/search/MailSearchProvider.java	94,225
/columba/trunk/mail/src/main/java/org/columba/mail/search/SearchFolderFactory.java	235
/columba/trunk/mail/src/main/java/org/columba/mail/shutdown/ClearRecentFlagPlugin.java	322
/columba/trunk/mail/src/main/java/org/columba/mail/smtp/SMTPServer.java	199
/columba/trunk/mail/src/main/java/org/columba/mail/smtp/command/SendMessageCommand.java	199
/columba/trunk/mail/src/main/resources/org/columba/mail/plugin/plugin.xml	296
/columba/trunk/mail/src/test/java/org/columba/mail/filter/FilterListTest.java	156
/columba/trunk/mail/src/test/java/org/columba/mail/filter/plugins/AbstractFilterTst.java	158
/columba/trunk/mail/src/test/java/org/columba/mail/filter/plugins/FlagsFilterTest.java	158
/columba/trunk/mail/src/test/java/org/columba/mail/folder/AbstractFolderTst.java	158
/columba/trunk/mail/src/test/java/org/columba/mail/folder/FolderTstHelper.java	158
/columba/trunk/mail/src/test/java/org/columba/mail/gui/composer/command/AbstractComposerTst.java	158
/columba/trunk/mail/src/test/java/org/columba/mail/gui/util/AddressListRendererTest.java	305
/columba/trunk/mail/src/test/java/org/columba/mail/imap/TestServer.java	156
/columba/trunk/mail/src/test/java/org/columba/mail/parser/ListParserTest.java	421
/columba/trunk/native/win32/Columba.lap	318
/columba/trunk/plugins/org.columba.chat.altura/src/org/columba/chat/command/AddContactCommand.java	199
/columba/trunk/plugins/org.columba.chat.altura/src/org/columba/chat/command/ConnectCommand.java	199
/columba/trunk/plugins/org.columba.chat.altura/src/org/columba/chat/ui/action/AddContactAction.java	199
/columba/trunk/plugins/org.columba.chat.altura/src/org/columba/chat/ui/action/OpenConversationAction.java	199