修士学位論文

題目

オープンソース開発支援のための
ソースコード及びメールの履歴対応表示システム

指導教官

井上 克郎 教授

報告者

石川 武志

平成 14 年 2 月 13 日

大阪大学 大学院基礎工学研究科

情報数理系専攻 ソフトウェア科学分野

平成 13 年度 修士学位論文

オープンソース開発支援のための
ソースコード及びメールの履歴対応表示システム

石川 武志

内容梗概

インターネットに代表されるネットワーク環境の発展に伴い，分散した多くの開発者が協調しながら開発が行われる「オープンソースソフトウェア」が，近年，注目を集めている．オープンソースソフトウェアの開発では，開発中のプロダクトが広く公開されているため，世界中に分散している開発者が自由に参加できる．また，プロダクトは版管理システムや電子メール等を用いたオープンソース開発環境によって管理されるため，多くの開発者が並列的に開発作業を行える．

しかし，現状のオープンソース開発環境では，各システムがそれぞれ固有の形式で開発情報を蓄積しているため，「システム情報が利用困難」という問題がある．また，開発環境を構成する各システムが互いに連係を持たず，開発情報が分散して存在しているため，「システム間の連係不足」という問題がある．

そこで，本研究では，これらの問題を解決することを目的として，オープンソース開発支援のためのソースコード及びメールの履歴対応表示システム (SPxR) を設計し，その実装を行った．本システムは，履歴データベースシステムと履歴データ表示システムから構成される．履歴データベースシステムは，既存の版管理システムと電子メールアーカイブから取得した開発情報と，それらの中で相互に関連を持つ情報から生成する統合情報を，データベース化して管理する．履歴データ表示システムは，開発者と履歴データベースシステムを結ぶインターフェースとしての役割を果たし，履歴データベースシステムと連係して，開発者が必要とする情報を提供する．本システムを利用することにより，開発者が各システムからそれぞれ開発情報を取得するために大きな負担を強いられることなく，開発者に必要な情報を的確に提供することが可能となるため，開発作業を支援することができる．

また，実際のオープンソース開発で用いられているデータを用いて，本システムの適用実験を行った．その結果，本システムを用いることにより，開発者が必要とする情報を効率良く提供することが可能となり，現状のオープンソース開発環境における問題点が解決されることを確認した．

主な用語

オープンソース開発 (Open Source Development)
CVS(Concurrent Versions System)
電子メール (Electric Mail)
ソフトウェアプロダクト (Software Products)

目 次

1 はじめに **5**

2 オープンソース開発 **7**

2.1 オープンソース開発環境 7

2.1.1 版管理システム 8

2.1.2 電子メール 9

2.2 オープンソース開発の問題点 10

2.2.1 システム固有の情報蓄積 10

2.2.2 システム間の連係不足 12

3 オープンソース開発向けの開発支援環境 **13**

3.1 開発支援環境の設計方針 13

3.2 開発支援環境の構成 13

4 オープンソース開発支援用ソースコード・メールの履歴対応表示システム (SPxR) **16**

4.1 履歴データベースシステム 17

4.1.1 CVS 情報管理部 17

4.1.2 電子メール情報管理部 20

4.1.3 統合情報管理部 23

4.2 履歴データ表示システム 25

4.3 システムの実装 26

4.3.1 履歴データベースシステム 26

4.3.2 履歴データ表示システム 30

5 実験 **34**

5.1 実験対象 34

5.2 実験結果・考察 35

5.2.1 情報の取得・生成時間 35

5.2.2 統合情報の妥当性 38

5.2.3 オープンソース開発環境の問題点 40

6 関連研究 **42**

7 まとめ **43**

謝辞 44

参考文献 45

付録 49

1 RCS ファイル **50**

2 電子メール **53**

3 設定ファイル (.spxrconf) **56**

1　はじめに

ソフトウェアの開発規模が増大するのに伴い，その開発形態は多人数化，分散化している．ある一つの大規模なソフトウェアを開発する場合，複数の開発者が，互いにソースコードを共有しながら同時に一つの開発作業に携わることが一般的になりつつある [2]．また，インターネットやイントラネット等のネットワーク環境の発展に伴い，分散した多くの開発者がそれぞれ異なる拠点で開発作業を行うことも珍しくなくなってきた．

近年では，オープンソースソフトウェアにおけるソフトウェアの開発が注目されている [27][28][32]．オープンソース開発は，開発中のプロダクトを広く公開することにより，世界中に分散した開発者がいつでも自由に参加できる．公開されるソフトウェアプロダクトはオープンソース開発環境で管理されるため，多くの開発者が並列的に開発作業を行うことが可能となる．オープンソースソフトウェアは，このような開発環境を背景として，短時間で多くの作業を行えることから，高品質で多機能なソフトウェアとして広く利用されている．オープンソース開発環境では，公開するプロダクトを管理するために，版管理システムや電子メール等，多くの既存システムが用いられる．

しかし，現状のオープンソース開発環境にはいくつかの問題点が存在している．例えば，多くの既存システムがそれぞれ固有の形式で開発情報を蓄積しており，開発者がそれらの情報を利用するコストが大きい．あるいは，多くのシステムをまとめて提供されるが，各システムが連係を持たないために，開発情報が分散して存在する．

本研究では，これらの問題を解決するために，オープンソース開発支援のためのソースコード及びメールの履歴対応表示システム (SPxR) の設計と，その実装を行った．本システムは，履歴データベースシステムと履歴データ表示システムから構成される．履歴データベースシステムは，既存の版管理システムと電子メールアーカイブから取得した開発情報や，それらの中で関連を持つ情報から生成する統合情報を，それぞれデータベース化して管理する．履歴データ表示システムは，開発者と履歴データベースシステムを結ぶインターフェースとしての役割を果たし，履歴データベースシステムと連係して，開発者が必要とする情報を提供する．本システムを利用することにより，開発者が各システムからそれぞれ開発情報を取得するために大きな負担を強いられることなく，開発者に必要な情報を的確に提供することが可能となるため，開発作業を支援することができる．

また，実際のオープンソース開発で用いられているデータを用いて，本システムの適用実験を行った．情報に取得にかかる時間を測定し，生成した統合情報の妥当性についての検証を行った．その結果，本システムは十分に実用的な時間で動作することを確認した．また，開発者が必要とする情報を提供することが可能となり，さらに，現状のオープンソース開発環境における問題点が解決できることを確認した．

オープンソース開発向けの開発支援環境を構築することにより，限られた時間を有効に利用して開発作業を行うことが可能となり，開発者の負担を軽減し，より質の高いソフトウェアを開発するための基礎とすることができると考えられる．

以降，2 節では，オープンソース開発とその開発環境について説明し，その問題点について述べる．3 節では，我々が研究を行っているオープンソース向けの開発支援環境の提案を行う．4 節では，オープンソース開発支援用ソースコード・メールの履歴対応表示システムの設計と実装について述べる．5 節では，そのシステムに対して適応実験を行い，評価と考察を行う．最後に，6 節で本研究のまとめと今後の課題について述べる．

2　オープンソース開発

オープンソース (Open Source) 開発とは，複数の開発者が並列的にソフトウェアの開発作業を行うために，開発中のソースコードやドキュメント等のプロダクトを広く公開する開発手法である [27][28][32]．従って，オープンソース開発には，世界中に分散した各開発者が，潜在的に，いつでも自由に開発作業に参加することが可能である．また，このような手法で開発されたソフトウェアは，オープンソースソフトウェア (Open Source Software, OSS) と呼ばれている．実際に，FreeBSD[36] や Linux[22]，GNU[13]，Apache[1] 等のソフトウェア開発でもこの手法が用いられており，世界中の開発者が開発作業に参加している．これらのオープンソースソフトウェアは，高品質で多機能なソフトウェアとして世界中で利用されており，オープンソース開発はその開発手法として，特に注目を集めている．

本節では，オープンソース開発の現状と，その問題点について説明する．

2.1　オープンソース開発環境

オープンソース開発では，各開発者がそれぞれ分散して並列的に開発作業を行うことが可能である．その一方で，開発中のソースコードやドキュメント等のプロダクトを広く公開するため，それらの管理を行う必要がある．そこで，オープンソース開発に参加する開発者は，オープンソース開発環境と呼ばれる環境の中でプロダクトの管理を行う．

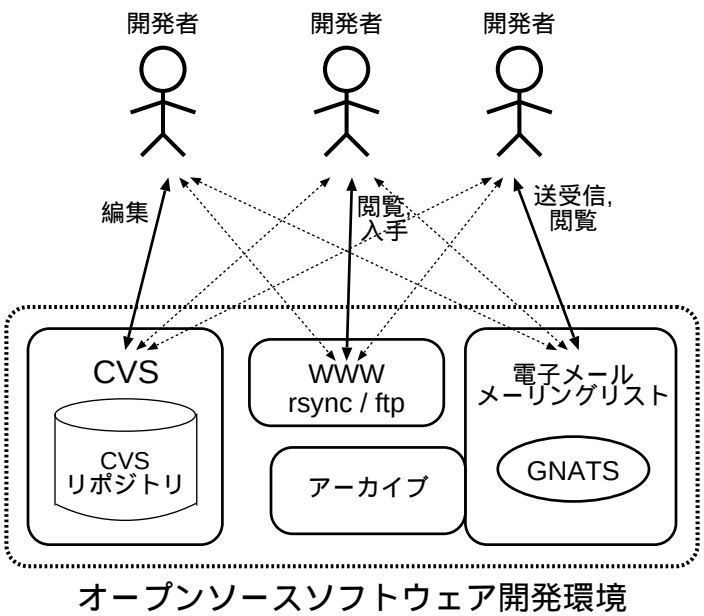

図 1: オープンソース開発環境の構成例

オープンソース開発環境の構成例を図 1 に示す．オープンソース開発環境は，一般に複数の既存システムから構成されている．図 1 の構成例の場合，ソースコードやドキュメント等のプロダクトは，版管理システム [9] の一つである CVS(Concurrent Versions System)[4][12][20][29] を用いて管理される．それらのプロダクトは，rsync や ftp を利用して，各開発者に複製，配布される．また，開発者間で相互に行われる意志疎通の手段として，電子メールやメーリングリストが用いられる．その内容はアーカイブとして保存され，WWW(World Wide Web) を用いた検索エンジンによって自由に検索や閲覧が可能である．開発者からのバグ報告等フィードバックは，GNATS(GNU Problem Report Management System) を用いたバグデータベースによって管理される．

以下では，これらのシステムの中から，オープンソース開発で広く用いられている版管理システムと電子メールについて説明する．

2.1.1 版管理システム

版管理システムとは，ソフトウェア開発の際に行われる各プロダクトに対する作業履歴を蓄積し，管理するシステムである．また，蓄積された作業履歴を開発者に提供することにより，プロダクトの修正作業を開発者間で調整することができるため，間違いのない修正を行うことが可能となる [3][10]．あるいは，ソフトウェアの再利用や開発プロセスの作成を行う際にも役立つ [11]．

版管理システムは，各プロダクトに対する作業履歴を，リポジトリ (repository) にファイル単位で蓄積する．開発者がリポジトリにプロダクトを格納することをコミット (commit) と呼ぶ．リポジトリの内部では，プロダクトのある時点における状態であるリビジョン (revision) 単位で管理する．そのため，内部的にリビジョン番号，リビジョン間の差分情報，更新日時，更新者，更新時のコメント等をリビジョン情報として保持する．

版管理システムは数多く存在し，実際にさまざまな開発において利用されている．また，ローカルな環境で利用するシステムもあれば，ネットワークを介してグローバルな環境で利用するシステムもある [14][17][26]．例えば，FreeBSD や Linux 等の UNIX 系オペレーティングシステム上では，RCS(Revision Control System)[39] や CVS が標準的に利用可能である．あるいは，ClearCase[31]，Visual SourceSafe[25] や PVCS[24] 等，商用の版管理システムも数多く発売されている．

また，版管理システムを用いて，ソフトウェアの開発作業や保守を支援する試みも行われている．例えば，ソースコードの差分情報を利用した，ソフトウェア開発時のデバッグ支援システムがある [35]．

オープンソース開発では，版管理システムとして CVS が広く用いられている．CVS は，UNIX 上で動作する標準的な版管理システム RCS を改良して設計されており，以下のよう

な RCS が持つ問題点を解決している．

- プロジェクトがファイル単位で管理される

  RCS では，プロジェクトがファイル単位でのみ管理されるため，複数のファイルをディレクトリ構造として扱うことが困難である．

  CVS では，ディレクトリ構造を認識するため，プロジェクトをディレクトリ単位で管理することが可能である．

- 開発者がファイルをロックする方式である

  RCS では，開発者がファイルをロックして修正を行うため，他の開発者が並列的に開発作業を行うことが困難である．

  CVS では，開発者がリポジトリから作業用コピーを取得して修正を行い，同時に加えられた変更を必要に応じてマージすることができるため，ファイルをロックすることなく修正を行うことが可能である．

- ネットワーク環境に対応していない

  RCS では，各ファイルの履歴情報がそれぞれのマシンに散在する一方で，ネットワーク環境に対応していないため，開発者が分散して開発作業を行うことが困難である．

  CVS では，ネットワーク環境に対応しているため，インターネット上のどこからでもアクセスすることができる．

以上のような RCS が持つ問題点を解決している CVS は，オープンソース開発の版管理システムとして非常に適している．

CVS は，GUI や Web インターフェース，エディタ等のさまざまな目的に応じて，その関連ツールが数多く開発されており，実際に利用されている．例えば，CGI を利用した CVS の Web インターフェースとして cvsweb[7] がある．cvsweb を利用することにより，リポジトリ内に存在するファイル一覧や，各リビジョンのデータ，リビジョン間の差分等を既存の Web ブラウザで閲覧することが可能である．同様のシステムとして，bonsai[38] 等もある．

2.1.2 電子メール

電子メールとは，インターネットやイントラネット等のネットワークを通じて、文書や画像等のデータをやりとりするためのシステムである．今日では，ネットワーク環境の充実に伴い，容易に利用することが可能となり，単に「メール」と称されることも多い．

オープンソース開発では，開発者が世界中に分散して存在し，開発作業を行うことが多い．そのため，互いの意志疎通のための手段として，インターネットを介した電子メールが一般

的に利用されている．オープンソース開発では，開発者間での電子メールのやりとりを一括して管理するために，メーリングリスト (Mailing List) と呼ばれるシステムが利用されることも多い．メーリングリストでは，例えば，ある参加者がメーリングリスト宛に電子メールを送信すると，同じ電子メールを参加者全員に配信される．また，誰かがその電子メールに対して返信すると，その電子メールも参加者全員に配信される．このため，他の開発者間での議論内容を，各開発者が容易に捕捉することが可能となる．さらに，これらの電子メールが膨大な量になると，アーカイブ (archive) として管理される．また，WWW を用いて，アーカイブの中から電子メールによる議論の内容を検索するシステムも存在する．

これらのシステムを利用することにより，分散している開発者の間でさまざまな情報を共有することが可能となり，開発作業の促進につながる．

2.2 オープンソース開発の問題点

オープンソース開発では，分散した複数の開発者が自由に各々の開発作業を行う．これらの開発者は，普段から開発作業に専念しているのではなく，個人的な時間を利用して作業を行うことが多い．従って，並列的に作業を行う他の開発者の進捗状況を綿密に追跡する時間が十分に確保できず，その把握が難しい．あるいは，開発者間での意志疎通が不足してしまう傾向がある．すなわち，この種の開発においては，開発者間での意志疎通の支援が不十分であることが，重要な問題点の一つである．

以下では，本研究において，この問題点を解決する上で着目すべき要因について説明する．

2.2.1 システム固有の情報蓄積

現状のオープンソース開発は，複数のシステムを組み合わせた開発環境で作業が行われる．これらのシステムは，その目的に応じて，開発に関する情報をシステム内部に蓄積し，開発者に提供する．ところが，各システムから提供された情報は，あくまでもそのシステム固有の情報でしかない．従って，それらの情報が開発者に対して個々に提供された場合には，開発者が必要とする情報が含まれているとは限らない．そのため，開発者の立場から見ると，各システムの情報量が不足していると考えられてしまう．

例えば，多くの開発者は，版管理システムのリポジトリからリビジョン情報を取得し，参照する．ところが，版管理システムのリビジョン情報は，ファイル単位でリビジョンが管理されている．このため，ファイル単位での履歴を取得することは容易に可能であるが，その他の視点から履歴を取得することは困難である．例えば，ある開発者がこれまでに行った開発作業の履歴を取得したり，特定の日時に行われた更新作業の履歴を取得することは非常に難しい．

```
% rlog ln.c,v

RCS file: ln.c,v
Working file: ln.c
            .
            .
         ( 中略 )
            .
            .
----------------------------
revision 1.20
date: 2001/04/26 17:22:48;  author: sobomax;  state: Exp;  lines: +1 -1
Previous commit should read:

style(9) Reviewed by:   bde
----------------------------
revision 1.19
date: 2001/04/26 17:15:57;  author: sobomax;  state: Exp;  lines: +28 -9
Bring in ‘-h’ compatability option and its alias ‘-n’ to match NetBSD and GNU
semantics.

style(9) Reviewed by:
Obtained from:  NetBSD
----------------------------
            .
            .
         ( 以下略 )
            .
            .
```

図 2: ファイル単位のリビジョン情報

具体的には，以下のような問題が生じる．

CVS リポジトリ内にある，src/bin/ln/ln.c,v というファイルについて，ファイル単位でリビジョン情報を取得するのは容易である．例えば，CVS コマンドの一つである rlog コマンドを利用することで，図 2 に示すように，ファイル単位でリビジョン情報を簡単に取得可能である．

しかし，リビジョン 1.19 の更新作業者 sobomax がこれまでに行った開発作業の履歴を取得する方法や，リビジョン 1.19 の更新日時 2001/04/26 17:15:57 と同時に行われた更新作業の履歴を取得することは非常に難しい．

2.2.2 システム間の連係不足

オープンソース開発では，開発管理を効率良く行うことを目的として，SourceForge[40] や SourceCast[6]，OSDL(Open Source Development Lab.)[30] 等のサービスが提供されている．これらのサービスでは，電子メールや会議システム等の汎用的な CSCW ツールや，その内容を記録したアーカイブ，WWW，版管理システム等，多くのシステムをまとめて開発者に提供している．しかし，提供されている各システムは互いに独立したものであり，単一の環境として何らかの連係を持っているわけではない．すなわち，電子メールで行われた意志疎通の内容はそのシステムの内部にアーカイブ化されて記録されているが，それらの情報は版管理システム CVS の情報と連係を持つことはない．そのため，CVS リポジトリを参照しながら，それに関連した電子メール上での話題を取得したい場合，あるいは，電子メールを参照しながら，それに関連した CVS リポジトリの情報を取得したい場合には，開発者が二つのシステムから個々に情報を取得して，それらを結びつける必要がある．しかし，そのように関連付けを開発者が行う場合のコストは，開発規模が増大するにつれて非常に大きくなる．

具体的には，以下に示すような問題が生じる．

あるオープンソース開発では，版管理システム CVS と，電子メールとそのアーカイブを利用しているとする．ある開発者が，CVS リポジトリ内の `src/bin/ln/ln.c,v` というファイルのあるリビジョン `1.19` の情報を参照した際に，それに関連した電子メール上での話題を取得したい場合には，ファイルパスやリビジョン番号，ログメッセージからいくつかのキーワードを抽出して，電子メールの全文検索を行うなどの方法を利用する必要がある．あるいは，ある電子メールを参照しながら，それに関連した CVS リポジトリのリビジョン情報を取得したい場合には，電子メールのサブジェクトや本文からいくつかのキーワードを抽出して，それを手がかりにして CVS リポジトリを検索しなければならない．

3　オープンソース開発向けの開発支援環境

我々の研究グループでは，版管理システムを用いた分散ソフトウェア開発の一例であるオープンソース開発を対象として，開発支援環境の構築を目指している [18][23]．この開発支援環境は，オープンソース開発の生産性を高めることを目的として，開発者に対する作業支援を行う．

本節では，オープンソース開発向けの開発支援環境を提案する．

3.1　開発支援環境の設計方針

オープンソース開発が実際のソフトウェア開発の手法として取り入れられてから，既に数年が経過している．この数年間で，数多くのソフトウェアが開発されており，それらの膨大なプロダクトが蓄積されている．我々の研究では，既に蓄積されているこれらのプロダクトを，今後のオープンソース開発に役立てることを主眼に置いて，研究を行っている．

我々が提案する，オープンソース開発向けの開発支援環境の設計方針を以下に示す．

- 既存のシステムをそのまま利用する

  本開発支援環境では，現状のオープンソース開発で用いられている既存のシステムをそのまま利用する．具体的には，CVS や，電子メールとそのアーカイブを現状のまま変更することなく用いる．すなわち，開発者が現状でこれらのシステムによって利用可能である機能を，本開発支援環境を導入することにより失うことはない．

- 既存のシステムが蓄積した情報を有効に利用する

  本開発支援環境では，既存のシステムがこれまでに蓄積した膨大なプロダクトに関する情報を，有効に利用する．現状の CVS リポジトリや電子メールのアーカイブは，情報を蓄積することに主眼が置かれている．その一方で，蓄積された情報は，CVS コマンドや全文検索等，直接的な手段でしか参照できない．しかし，これらの情報の中に含まれる相互関係を抽出することにより，開発者はそれらの情報を結びつけて，有効に利用することができる．

3.2　開発支援環境の構成

オープンソース開発向けの開発支援環境の構成を，図 3 に示す．

本開発支援環境は，現状のオープンソース開発で用いられる既存のシステムと，これらのシステムが蓄積したプロダクトを有効に活用するためのシステムから構成されている．

これらのシステムについて，以下で簡単に説明する．

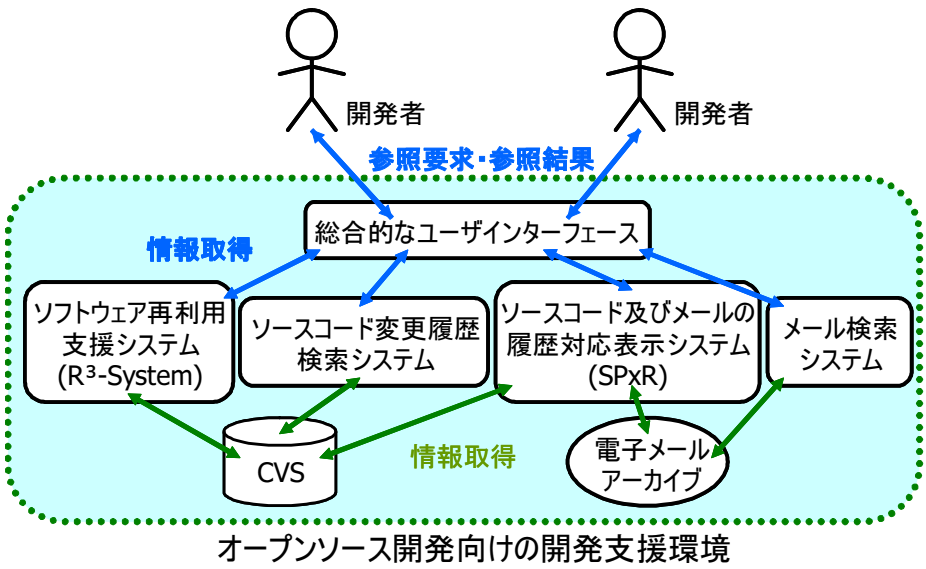

図 3: オープンソース開発向けの開発支援環境の構成

- CVS

  CVS は，従来の版管理システムの機能を提供し，ソースコードやドキュメント等のソフトウェアプロダクトリポジトリとしての役割を果たす．

- 電子メール

  電子メールやメーリングリストとそのアーカイブは，開発者間に意志疎通の機能を提供し，その記録を保持する役割を果たす．

- ソフトウェア再利用支援システム (R^3-System)[15]

  ソフトウェア再利用支援システム (R^3-System) は，CVS リポジトリに蓄積された複数のソースコードファイル間の利用関係と類似度を用いて，ソースコードファイルの相対的再利用性 (Relative Reusability) を評価し，順位付け (Ranking) を行う．

  開発者が，過去に開発されたソースコードから再利用可能な部品を検索する場合，検索キーワードによる検索結果を相対的再利用性の順に表示することが可能である．

  これにより，ソフトウェアの再利用を行う開発者を支援する.

- プログラム変更履歴検索システム (CoDS)[33]

  ソースコード変更履歴検索システム (CoDS) は，CVS リポジトリに蓄積されたソースコードの変更履歴をデータベース化する．それらの情報を，ソースコード片を用いて検索することが可能である．

開発者が，手持ちのソースコード片を用いて検索を行うことで，開発者のソースコード修正作業を支援する．

- ソースコード・メールの履歴対応表示システム (SPxR)

  ソースコード・メールの履歴対応表示システム (SPxR) は，CVS のリビジョン情報と電子メールから情報を取得し，管理する．また，それらの情報から統合情報を生成し，管理する．

  これらの情報を開発者に提供することで，開発者の作業を支援する．

  このシステムの詳細については，本稿の次節で説明する．

- メール検索システム [34]

  メール検索システムは，オープンソース開発における意志疎通の重要な手段である電子メール，メーリングリストのアーカイブから情報を取得し，管理する．それらの情報を用いて，過去にやりとりされたメールの中から，開発者が興味を持つ内容のメールを効率良く検索することができる．

  開発者が，議論に使われたキーワード，ファイルパス，共同議論者等から電子メールの検索を容易に行うことで，開発者間の意志疎通を支援する．

4　オープンソース開発支援用ソースコード･メールの履歴対応表示システム (SPxR)

本節では，オープンソース開発向けの開発支援環境の構成要素の一つである，オープンソース開発支援用ソースコード・メールの履歴対応表示システム (Software Products CROSS Reference system, SPxR) について詳しく説明する．

本システムは，版管理システム CVS と，電子メールとそのアーカイブから情報を取得し，それらの統合情報を生成して，管理する．また，それらの情報を開発者に提供することで，開発者の作業を支援する．

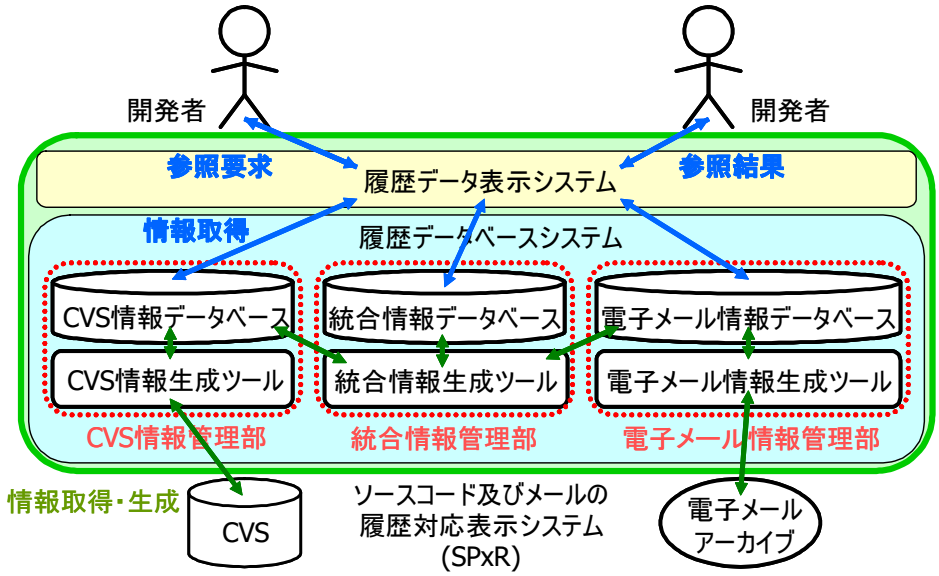

図 4: ソースコード・メールの履歴対応表示システム (SPxR) の構成

本システムの構成を，図 4 に示す．本システムは，履歴データベースシステムと，履歴データ表示システムという，2 つのサブシステムから構成される．

なお，本研究では，オープンソースソフトウェアの代表例である FreeBSD[36] のオープンソース開発を研究対象とする．FreeBSD のオープンソース開発環境では，版管理システム CVS を用いてソースコードを管理している．その CVS リポジトリは，CVS 関連ツール cvsweb を利用して，`http://www.freebsd.org/cgi/cvsweb.cgi/` で公開されている．また，開発作業に関連した開発者間の意志疎通は電子メールを用いて行なわれており，メーリングリストを利用して全開発者に送信される．それらのアーカイブは，WWW を利用して，`http://docs.freebsd.org/mail/` で公開されており，その内容を閲覧したり，検索することが可能である．

4.1 履歴データベースシステム

履歴データベースシステムは，本システムの根幹を成すサブシステムである．CVS リポジトリから取得したリビジョン情報と，電子メールのアーカイブから取得した意志疎通情報を取得し，管理する．また，両者の情報を結合して，統合情報を生成し，管理する．

履歴データベースシステムは，CVS 情報管理部，電子メール情報管理部，統合情報管理部という，3 つの部分から構成される．以下では，これらの部分について説明する．

4.1.1 CVS 情報管理部

CVS 情報管理部は，CVS 情報生成ツールと CVS 情報データベースから構成される．

- CVS 情報生成ツール

  CVS 情報生成ツールは，CVS リポジトリからリビジョン情報を取得し，CVS 情報を作成する．

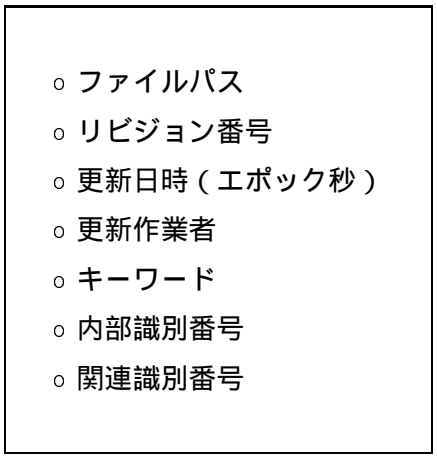

図 5: CVS 情報のデータ構造

CVS 情報のデータ構造を，図 5 に示す．

CVS 情報は，ファイルパス，リビジョン番号，更新日時，更新作業者，キーワード，内部識別番号，関連識別番号の 7 つのデータを持つ．

ファイルパス，リビジョン番号，更新日時，更新作業者は，CVS リポジトリ内の RCS ファイル形式のリビジョン情報を解析することにより取得可能である．また，リビジョン情報に含まれている更新時のログメッセージを解析して，出現頻度が高い単語やファ

イルパスをキーワードとして取得する．なお，RCS ファイルの書式とその具体例を，巻末の付録 1 に添付しておく．

複数ある CVS 情報の中から特定の CVS 情報を指定する場合には，ファイルパスとリビジョン番号を指定すれば一意に指定可能である．そこで，CVS 情報の処理を簡単化するために，これらの組み合わせに対して内部的に識別番号を対応させて，内部識別番号として CVS 情報の要素に追加する．内部識別番号は，CVS 情報であることを表す，"c" から始まる値が格納される．

また，CVS 情報の中で，「同一の更新日時に，同一の更新作業者が更新作業を行ったリビジョンが複数存在する場合，それらのリビジョンは関連を持つ」という関係が成り立つ．そこで，そのような関係がある複数のリビジョンに関する CVS 情報には，互いに関連を持つリビジョンの内部識別番号を記録するため，関連識別番号として CVS 情報の要素に追加する．これにより，相互に関連を持つ CVS 情報の間にリンク情報を付加することができるため，関連する CVS 情報の検索が容易になる．

ファイルパス	src/bin/ln/ln.1,v
リビジョン番号	1.15
更新日時 (エポック秒)	2001/04/26 17:15:57 (988305357)
更新作業者	sobomax
キーワード	NetBSD
内部識別番号	c260678
関連識別番号	c260708

ファイルパス	src/bin/ln/ln.c,v
リビジョン番号	1.19
更新日時 (エポック秒)	2001/04/26 17:15:57 (988305357)
更新作業者	sobomax
キーワード	NetBSD
内部識別番号	c260708
関連識別番号	c260678

図 6: CVS 情報

例えば，同一の更新日時 (2001/04/26 17:15:57) に，同一の更新作業者 (sobomax) によって更新作業が行われて，互いに関連を持つ 2 つのリビジョン情報から，図 6 に示

すCVS情報がそれぞれ作成されて，CVS情報データベースに登録される．

- CVS情報データベース

  CVS情報データベースは，CVS情報生成ツールによって作成されたCVS情報を，データベース化して管理する．

  一般的に，CVSリポジトリに新たなリビジョン情報が追加される際には，過去のリビジョン情報に変更が加えられることはない．

  また、本研究が対象としているFreeBSDのオープンソース開発は，CVSリポジトリ内に存在するファイルの総数は約100,000個，リビジョンの総数で約700,000個と，非常に膨大な情報量である．

  そのため，CVSリポジトリから取得したリビジョン情報をデータベース化することにより，CVS関連ツールとして有名なcvswebやbonsai等の既存システムと比較した場合，以下に示す利点がある．

  - リビジョン情報を取得するコストが大幅に削減できる

    既存のシステムでは，利用者の要求に応じて，毎回必要なリビジョン情報をCVSリポジトリから取得し，リビジョン情報を提供する．従って，これらのシステムを利用する場合，過去に取得した情報と全く同じ情報を再度取得することが多く，そのコストは無視できない．

    それに対して，CVS情報をデータベース化することにより，一度取得した情報をCVSリポジトリから再度取得することなく，有効に再利用できる．また，新たに追加されたリビジョンがある場合には，その差分情報だけを取得すればよいため，リビジョン情報を取得するコストが大幅に削減できる．
  - リビジョン情報に必要な情報を付加して管理することが可能となる

    既存のシステムでは，毎回必要なリビジョン情報をCVSリポジトリから取得するため，リビジョン情報に必要な情報を付加して管理することは困難である．

    それに対して，リビジョン情報をデータベース化することにより，リビジョン情報に必要な情報を付加して管理することが可能となる
  - リビジョン情報の検索能力が向上する

    既存のシステムでは，毎回必要なリビジョン情報をCVSリポジトリから取得するため，CVSリポジトリ内のリビジョン情報に対する検索を行うことは可能であるが，そのコストは非常に大きい．

    それに対して，リビジョン情報をデータベース化することにより，その検索能力は飛躍的に向上し，利用者の要求に応じた情報を提供することが可能となる．

4.1.2 電子メール情報管理部

オープンソース開発では，電子メールを用いて開発作業に関連した開発者間の意志疎通が行われている．このため，電子メールは開発者間の意志疎通手段として，重要な役割を果たしている．また，これらの電子メールを一括して管理するために，メーリングリストが利用されることも多い．さらに，それらの電子メールが膨大な量になると，アーカイブとして管理される．

電子メールは一般的な用途にも広く用いられており，その本文内容は多種多彩である．ところが，オープンソース開発を目的とした電子メールの本文内容は，以下に示す「コミットメール」と「議論メール」の 2 つに大きく分けられる．なお，コミットメールと議論メールの具体例を，巻末の付録 2 に添付しておく．

- コミットメール

  コミットメールとは，CVS リポジトリのリビジョン情報が更新された際に，CVS から送信される通知メッセージである．

  コミットメールは，更新されたファイルパスとリビジョン番号，更新作業者，更新日時，更新時のログメッセージ等の更新情報を，開発者に通知する．これにより，更新されたリビジョンに関する情報を，開発者が容易に取得することができる．

- 議論メール

  議論メールとは，コミットメール以外の全ての電子メールである．コミットメールに対する返信の電子メールも，議論メールに含まれる．

  議論メールには，開発作業に関連する一般的な話題が記述されており，それに返信することで相互に議論が行われる．

電子メール情報管理部は，電子メール情報生成ツールと電子メール情報データベースから構成される．

- 電子メール情報生成ツール

  電子メール情報生成ツールは，電子メールあるいはそのアーカイブから，過去に開発者間で行われた意志疎通に関する情報を取得し，電子メール情報を作成する．

  電子メール情報のデータ構造を，図 7 に示す．

  電子メール情報は，メッセージ ID，送信者とその電子メールアドレス，送信日時，サブジェクト，キーワード，内部識別番号，関連識別番号の 7 つのデータを持つ．さらに，コミットメールの場合には，更新作業者，更新日時，更新ファイルパスとリビジョン番号の 3 つのデータが追加される．

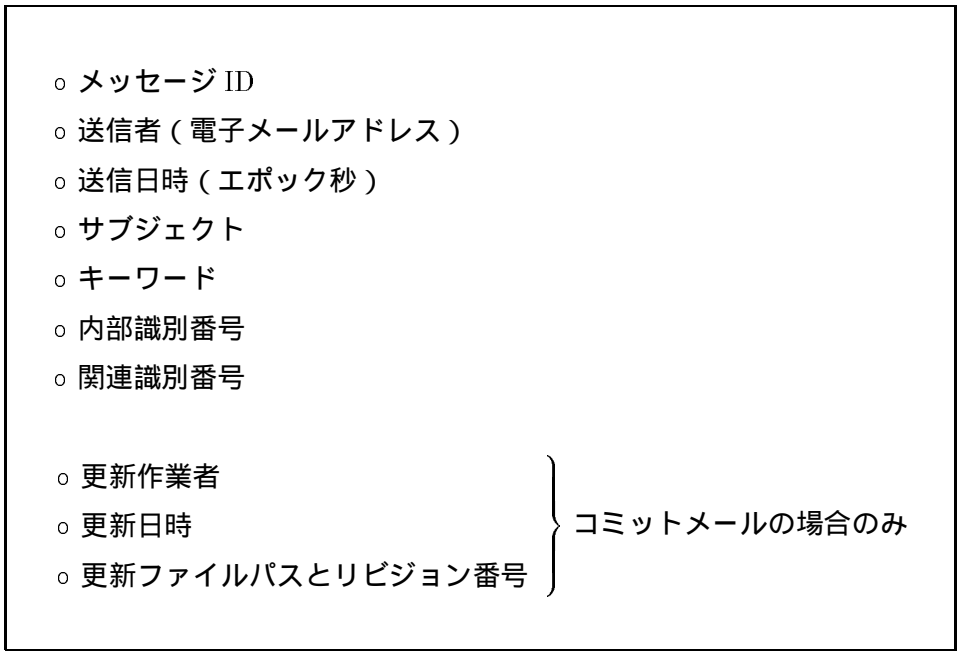

図 7: 電子メール情報のデータ構造

メッセージ ID，送信者とその電子メールアドレス，送信日時，サブジェクトは，電子メールのヘッダ部分を解析することにより取得可能である．また，電子メールの本文内容を解析して，出現頻度が高い単語やファイルパスをキーワードとして取得する．

複数ある電子メール情報の中から特定の電子メール情報を指定する場合には，メッセージ ID を指定すれば一意に指定可能である．そこで，電子メール情報の処理を簡単化するために，このメッセージ ID に対して内部的に識別番号を対応させて，内部識別番号として電子メール情報の要素に追加する．内部識別番号は，電子メール情報であることを表す，”m”から始まる値が格納される．

また，電子メール情報の中で，「ある電子メールと，それに対して返信された電子メールが存在する場合，それらの電子メールは関連を持つ」という関係が成り立つ．そこで，そのような関係がある複数の電子メールに関する電子メール情報には，互いに関連を持つ電子メールの内部識別番号を記録するため，関連識別番号として電子メール情報の要素に追加する．これにより，相互に関連を持つ電子メール情報の間にリンク情報を付加することができるため，関連する電子メール情報の検索が容易になる．

例えば，巻末の付録 2 に添付した，あるリビジョンが更新された際のコミットメールと，それに対して返信された議論メールから，図 8 に示す電子メール情報がそれぞれ作成されて，電子メール情報データベースに登録される．

メッセージ ID	200104261715.f3QHFvs90663@freefall.freebsd.org
送信者 (電子メールアドレス)	Maxim Sobolev (sobomax@FreeBSD.org)
サブジェクト	cvs commit: src/bin/ln ln.1 ln.c
送信日時 (エポック秒)	Thu, 26 Apr 2001 10:15:57 -0700 (PDT) (988305357)
キーワード	NetBSD
識別番号	m16118
関連識別番号	m16128,m16143,m16158,m16168,m16259
更新作業者	sobomax
更新日時	2001/04/26 10:15:57 PDT
更新ファイルパス	src/bin/ln/ln.1; 1.15 src/bin/ln/ln.c; 1.19

メッセージ ID	20010426203418.A72921@sunbay.com
送信者 (電子メールアドレス)	Ruslan Ermilov (ru@FreeBSD.org)
サブジェクト	Re: cvs commit: src/bin/ln ln.1 ln.c
送信日時 (エポック秒)	Thu, 26 Apr 2001 20:34:18 +0300 (988306458)
キーワード	NetBSD
識別番号	m16128
関連識別番号	m16118,m16143,m16158,m16168,m16259

図 8: 電子メール情報

- 電子メール情報データベース

  電子メール情報データベースは，電子メール情報生成ツールによって作成された電子メール情報を管理する．

  電子メールの場合も，CVS リポジトリのリビジョン情報と同様の特徴として，新たな電子メールがアーカイブに追加される際には，過去の電子メールに変更が加えられることはない．

  また、本研究が対象としている FreeBSD のオープンソース開発は，メーリングリスト内で一週間につき平均約 3000 通以上の電子メールのやりとりがあり，非常に膨大な情報量である．

  そのため，電子メール情報をデータベース化することにより，CVS 情報をデータベース化した場合と同様の利点がある．

4.1.3 統合情報管理部

統合情報管理部は，統合情報生成ツールと統合情報データベースから構成される．

- 統合情報生成ツール

  統合情報生成ツールは，CVS 情報データベースから CVS 情報を，電子メール情報データベースから電子メール情報を取得して，両者のデータを比較する．その結果，両者が相互に関連を持つ情報である場合にはそれらを結合して，統合情報を生成する．

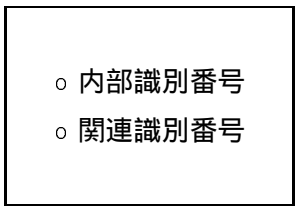

図 9: 統合情報のデータ構造

作成される統合情報のデータ構造を，図 9 に示す．

統合情報は，内部識別番号，関連識別番号の 2 つのデータを持つ．

統合情報を作成する場合，CVS 情報と電子メール情報を比較する必要がある．

まず，CVS 情報，電子メール情報のどちらか一方から，情報を取得する．この情報の内部識別番号を，統合情報の内部識別番号とする．また，この情報の関連識別番号を，統合情報の関連識別番号とする．

次に，最初に取得した情報が CVS 情報の場合には電子メール情報データベースから，電子メール情報の場合には CVS 情報データベースから情報を取得し，両者を比較して結合する．

このとき，CVS 情報と電子メール情報を結合する場合には，両者が相互に関連を持つ必要がある．そこで，以下の判断基準によって両者の情報を比較して，結合すべき情報であるかどうか判断する．

1. 「ファイルパス」と「リビジョン番号」が一致する

   コミットメールから取得された電子メール情報の場合には「更新ファイルパスとリビジョン番号」のデータを持っているため，CVS 情報の「ファイルパス」「リビジョン番号」と比較することが可能である．

   両者が同じファイルパス，同じリビジョン番号である場合には，CVS 情報と電子メール情報が相互に関連を持つと判断する．

2. 「キーワード」が一致する

   CVS 情報と電子メール情報はどちらも「キーワード」のデータを持っており，両者の「キーワード」を比較することが可能である．

   両者が同じキーワードを持つ場合には，CVS 情報と電子メール情報が相互に関連を持つと判断する．

   あるいは，CVS 情報の「キーワード」にファイルパスが含まれている場合には，電子メール情報の「更新ファイルパス」と，電子メール情報の「キーワード」にファイルパスが含まれている場合には，CVS 情報の「ファイルパス」とそれぞれ比較することも可能である．

   両者が同じファイルパスである場合には，CVS 情報と電子メール情報が相互に関連を持つと判断する．

両者を比較した結果がこれらのいずれかに当てはまる場合には，CVS 情報と電子メール情報が相互に関連を持つと判断して，両者を結合する．この際，結合した CVS 情報，あるいは電子メール情報が持っていた内部識別番号は，統合情報の「関連識別番号」として格納される．これにより，統合情報の関連識別番号に，結合された CVS 情報と電子メール情報の内部識別番号に対するリンク情報を付加することができるため，関連する CVS 情報，電子メール情報の検索が容易になる．

例えば，図 6 (18 ページ) の内部識別番号 c260678 ，c260708 の CVS 情報と，図 8 (22 ページ) の内部識別番号 m16118 の電子メール情報は，「ファイルパス」と「リビジョン番号」が一致するため，上記の判断基準 1. を満たす．従って、c260678 と m16118 ，c260708 と m16118 をそれぞれ結合して，統合情報を生成する．

また，図 6 の内部識別番号 c260678 ，c260708 の CVS 情報と，図 8 の内部識別番号 m16128 の電子メール情報は，「キーワード」が一致するため，上記の判断基準 2. を満たす．従って、c260678 と m16128 ，c260708 と m16128 をそれぞれ結合して，統合情報を生成する．

これらの情報を実際に結合して生成された統合情報の例を，図 10 に示す．

- 統合情報データベース

  統合情報データベースは，統合情報生成ツールによって作成された統合情報を管理する．

  統合情報は，CVS 情報と電子メール情報を結合して生成される情報である．そのため，統合情報をデータベース化することにより，CVS 情報，電子メール情報をデータベース化した場合と同様の利点がある．

図 10: 統合情報

4.2 履歴データ表示システム

履歴データ表示システムは，開発者と履歴データベースシステムを結ぶインターフェースとしての役割を果たす．開発者から参照要求を受け付け，履歴データベースシステムと連係し，その参照結果を表示して，開発者が必要とする情報を提供する．

履歴データ表示システムが提供する機能について，以下で説明する．

- CVS 情報，電子メール情報の表示と検索

  GUI 等のインターフェイスを利用して CVS 情報，電子メール情報を表示することにより，開発者に情報を提供する．また，CVS 情報，電子メール情報の検索を行い，開発者が必要とする情報を提供する．

- 統合情報の表示と検索

  CVS 情報，電子メール情報を表示した上で，統合情報を利用して，それらの間で相互に存在する関連を表示することにより，開発者に統合情報を提供する．また，統合情報の検索を行い，開発者が必要とする情報を提供する．

これらの情報の検索方法を，図 11 に示す．

本システムを利用することにより，開発者は CVS 情報と電子メール情報の関連を把握することが容易に可能となる．これにより，開発者のコストを軽減し，開発作業が促進される．

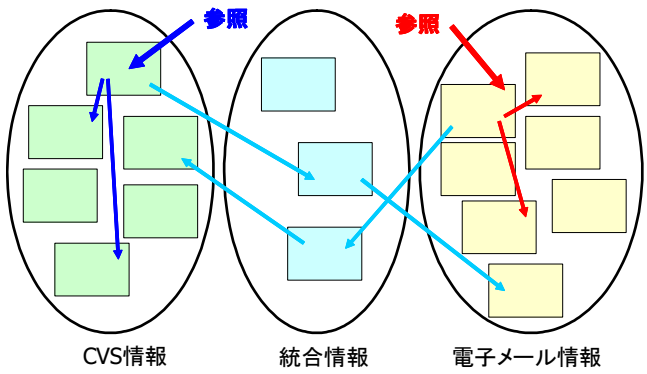

図 11: 情報の検索方法

4.3 システムの実装

本研究では，オープンソース開発支援用ソースコード・メールの履歴対応表示システム (SPxR) の実装を行った．

開発環境は，以下の通りである．

- CPU : PentiumIII 1GHz
- RAM : 256MB
- OS : Linux version 2.4.17
- 言語 : perl 5.6.1
- データベース : PostgreSQL 7.1.3 , BerkeleyDB (perl5)
- WEB サーバ : Apache 1.3.22

コードサイズは，全体で約 3600 行となった．内訳としては，履歴データベースシステムが約 2100 行，履歴データ表示システムが約 1500 行となっている．

4.3.1 履歴データベースシステム

履歴データベースシステムは，CVS 情報管理部，電子メール情報管理部，統合情報管理部という，3 つの部分から構成される．これらは，それぞれ独立して動作することが可能である．また，共通の設定ファイルによって，設定等の情報を一括して管理する．設定ファイル (`.spxrconf`) の例を，巻末の付録 3 に添付しておく．

CVS 情報管理部

CVS 情報管理部は，CVS 情報生成ツールと CVS 情報データベースから構成される．

- CVS 情報生成ツール (cvsinfogen.pl)

  CVS 情報生成ツールの処理概要を，以下に示す．

  – 設定ファイル (.spxrconf) に指定された CVS リポジトリのルートディレクトリ等の設定情報を読み込む．

  – ルートディレクトリから，CVS リポジトリ内のディレクトリ構造を再帰的に辿って，各 RCS ファイルの解析を行い，リビジョン情報を取得する．

  – 取得したリビジョン情報から CVS 情報を生成して，CVS 情報データベースに登録する．

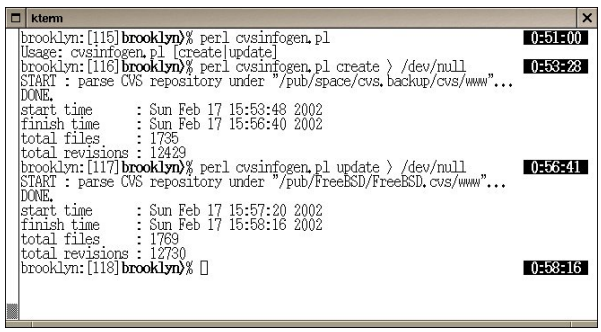

図 12: CVS 情報生成ツール (cvsinfogen.pl)

CVS 情報生成ツール (cvsinfogen.pl) の CUI(Command User Interface) 画面を，図 12 に示す．

CVS 情報生成ツールの起動書式は，以下の通りである．

```
% cvsinfogen.pl [create|update]
```

CVS 情報の取得方法として，引数に新規作成 create と更新 update を指定する必要がある．

create が指定された場合には，CVS リポジトリから全てのリビジョン情報を取得して，CVS 情報を生成し，CVS 情報データベースに登録する．

update が指定された場合には，前回の情報取得時から後に更新作業が行われたリビジョン情報だけを取得して，CVS 情報を生成し，CVS 情報データベースに登録する．

- CVS 情報データベース

  CVS 情報データベースには，PostgreSQL と，perl5 の BerkeleyDB を利用した．

  CVS リポジトリから取得した CVS 情報は，PostgreSQL のデータベース cvsinfo に登録される．しかし，登録される CVS 情報が膨大な量になる場合，頻繁に参照される情報を扱うためには，PostgreSQL のデータベースにアクセスするよりも，perl5 の BerkeleyDB によるハッシュ構造を利用する方が効率が良い．そこで，両者を組み合わせることにより，効率良く CVS 情報の取得や検索を行うことができる．

電子メール情報管理部

電子メール情報管理部は，電子メール情報生成ツールと電子メール情報データベースから構成される．

- 電子メール情報生成ツール (mailinfogen.pl)

  電子メール情報生成ツールの処理概要を，以下に示す．

  – 設定ファイル (.spxrconf) に指定された電子メールアーカイブのルートディレクトリ等の設定情報を読み込む．
  – ルートディレクトリから，電子メールアーカイブのディレクトリ構造を再帰的に辿って，各電子メールの解析を行う．
  – 電子メール情報を生成して，電子メール情報データベースに登録する．

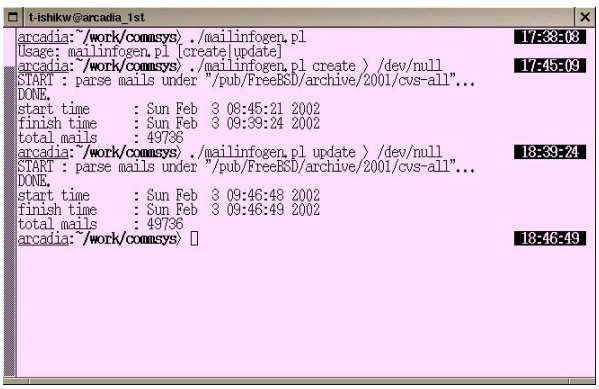

図 13: 電子メール情報生成ツール (mailinfogen.pl)

電子メール情報生成ツール (mailinfogen.pl) の CUI 画面を，図 13 に示す．

電子メール情報生成ツールの起動書式は，以下の通りである．

```
% mailinfogen.pl [create|update]
```

電子メール情報の取得方法として，CVS 情報の場合と同様に，引数に新規作成 create と更新 update を指定する必要がある．

create が指定された場合には，電子メールアーカイブから全ての電子メールに関する情報を取得して，電子メール情報を生成し，電子メール情報データベースに登録する．

update が指定された場合には，前回の情報取得時から後に送信された電子メールに関する情報だけを取得して，電子メール情報を生成し，電子メール情報データベースに登録する．

- 電子メール情報データベース

  電子メール情報データベースには，CVS 情報データベースと同様に，PostgreSQL と，perl5 の BerkeleyDB を利用した．

  電子メールアーカイブから取得した電子メール情報は，PostgreSQL のデータベース mailinfo に登録される．しかし，登録される電子メール情報が膨大な量になる場合，頻繁に参照される情報を扱うためには，PostgreSQL を利用するよりも，perl5 の BerkeleyDB によるハッシュ構造を利用する方が効率が良い．そこで，両者を組み合わせることにより，効率良く電子メール情報の取得や検索を行うことができる．

統合情報管理部

統合情報管理部は，統合情報生成ツールと統合情報データベースから構成される．

- 統合情報生成ツール (integinfogen.pl)

  統合情報生成ツールの処理概要を，以下に示す．

  - 設定ファイル (.spxrconf) から，統合情報に関する設定情報を読み込む．
  - CVS 情報データベースから CVS 情報を，電子メール情報データベースから電子メール情報を取得して，両者のデータを比較する．
  - 相互に関連を持つ情報である場合には，両者を結合して統合情報を生成し，統合情報データベースに登録する．

  統合情報生成ツール (integinfogen.pl) の CUI 画面を，図 14 に示す．

  統合情報生成ツールの起動書式は，以下の通りである，

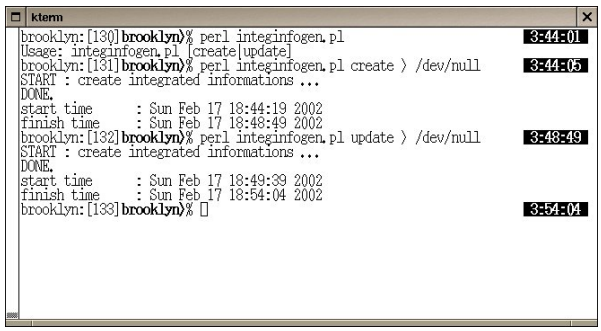

図 14: 統合情報生成ツール (integinfogen.pl)

```
% integinfogen.pl [create|update]
```

統合情報の生成方法としては，CVS 情報，電子メール情報の場合と同様に，新規作成 create と更新 update を指定する必要がある．

create が指定された場合には，全ての CVS 情報，電子メール情報が，統合情報を生成する対象となる．

update が指定された場合には，前回の統合情報生成時から後に追加された CVS 情報，電子メール情報が，統合情報を生成する対象となる．

- 統合情報データベース

  統合情報データベースには，perl5 の BerkeleyDB を利用した．両者を結合して生成された統合情報は，perl の BerkeleyDB によるハッシュ構造を利用したデータベースに登録される．これにより，効率良く統合情報の取得や検索を行うことができる．

4.3.2 履歴データ表示システム

履歴データ表示システム (spxr.cgi) は，開発者から参照要求を受け付けて，履歴データベースシステムと連係し，その参照結果を表示することにより，開発者が必要とする情報を提供する．

履歴データ表示システムは，perl による CGI で記述されており，開発者と履歴データベースシステムを結ぶインターフェースとしての役割を果たす．従って，既存の WEB ブラウザを通して情報の表示が可能である．これは，本システムを実際のオープンソース開発環境に組み込むことを視野に入れているため，一般的に広く利用されている GUI(Graphical User Interface) の一つである，CGI による WEB ブラウザ表示という手法を選択した．

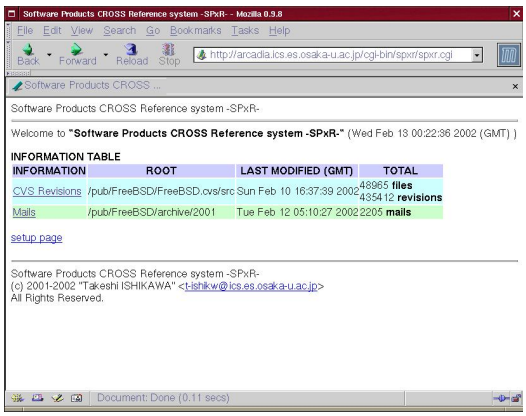

図 15: 履歴データ表示システム (spxr.cgi)

WEB ブラウザによる履歴データ表示システム (spxr.cgi) の表示画面を，図 15 に示す．履歴データ表示システムが提供する表示機能について，以下で説明する．

- CVS 情報の表示と検索

  CVS 情報を表示することにより，開発者に CVS 情報を提供する．

  CVS 情報は，基本的に CVS リポジトリのディレクトリ構造で表示される．開発者が各ディレクトリを自由に辿ることにより，CVS 情報を取得することが可能である．

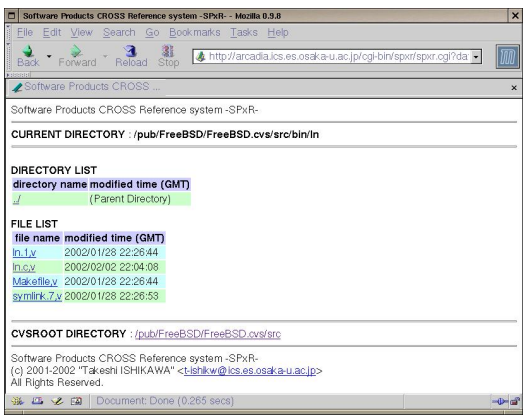

図 16: CVS 情報の表示画面

CVS リポジトリのディレクトリ構造による CVS 情報の表示画面を，図 16 に示す．

また，CVS 情報の検索を行い，開発者が必要とする情報を提供する．本システムでは，更新日時，更新作業者，キーワードによる CVS 情報の検索が可能である．

- 電子メール情報の表示と検索

  電子メール情報を表示することにより，開発者に電子メール情報を提供する．

  電子メール情報は，基本的に送信日時でソートされて．リスト表示される．開発者がこのリストを自由に辿ることにより，電子メール情報を取得することが可能である．

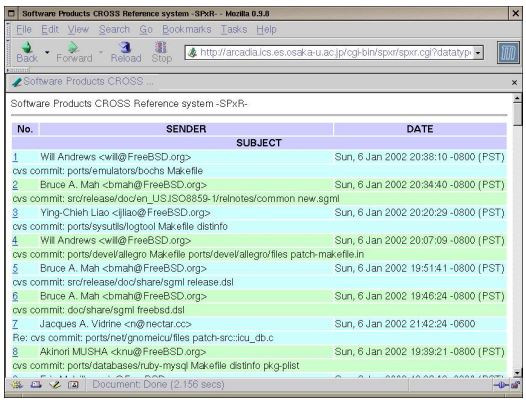

図 17: 電子メール情報の表示画面

電子メール情報の表示画面を，図 17 に示す．

また，電子メール情報の検索を行い，開発者が必要とする情報を提供する．本システムでは，送信者，サブジェクト，キーワードによる電子メール情報の検索が可能である．

- 統合情報の表示と検索

  上記のように，CVS 情報や電子メール情報を表示した上で，統合情報データベースから統合情報を検索して，それらの間で相互に存在する関連を表示することにより，開発者に統合情報を提供する．

CVS 情報を基準にした統合情報の表示画面を図 18 に，電子メール情報を基準にした統合情報の表示画面を図 19 にそれぞれ示す．

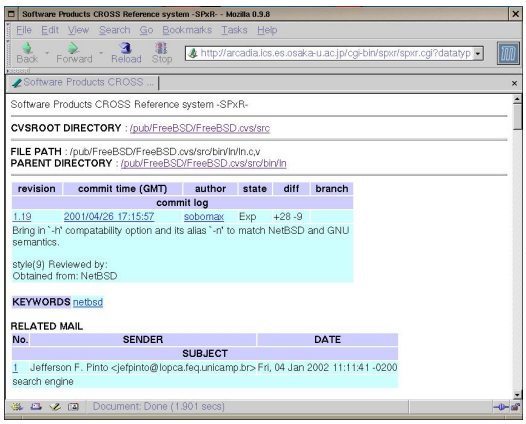

図 18: CVS 情報を基準にした統合情報の表示画面

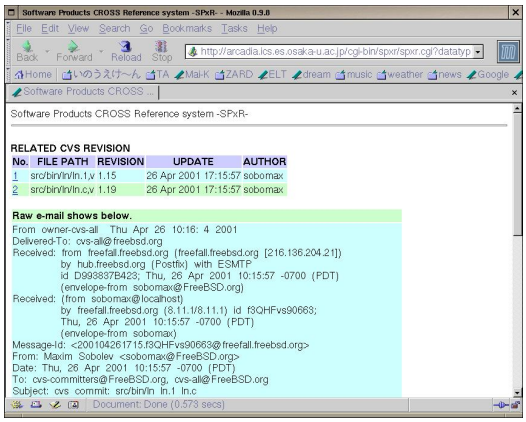

図 19: 電子メール情報を基準とした統合情報の表示画面

5 実験

本研究で実装を行った，オープンソース開発支援用ソースコード・メールの履歴対応表示システム (SPxR) について適用実験を行い，その評価と考察を行う．

5.1 実験対象

本システムの適用実験を行うために，実際のオープンソース開発の中から，仮想的なオープンソース開発のモデルを用意する．

具体的には，オープンソースソフトウェアの代表例である FreeBSD[36] のオープンソース開発環境から以下に示すデータを取得して，仮想的なオープンソース開発のモデルとする．

- CVS リポジトリ

  FreeBSD の CVS リポジトリをミラーリングして，そこから必要なリビジョン情報を取得する．

  FreeBSD の CVS リポジトリ内に存在するファイルの総数は約 100,000 個，リビジョンの総数で約 700,000 個，データサイズは約 1.5GB である．

- 電子メール

  FreeBSD のメーリングリストアーカイブをミラーリングして，そこから必要な電子メール情報を取得する．

  FreeBSD のメーリングリストの場合，一週間で平均約 3000 通以上の電子メールのやりとりがある．また，実際にメーリングリストが利用されるようになった 1994 年から 2001 年末までの間にやりとりされた電子メールの総数は約 1,050,000 通，データサイズは約 2.6GB である．

なお、実験環境は，以下の通りである．

- CPU : Pentium4 2GHz
- RAM : 1GB
- OS : FreeBSD 4.5-STABLE
- データベース : PostgreSQL 7.1.3 , BerkeleyDB (perl5)
- WEB サーバ : Apache 1.3.22

5.2 実験結果・考察

本システムを，仮想的なオープンソース開発モデルに適用して，実際に各種情報の取得と検索を行った．

5.2.1 情報の取得・生成時間

本システムは，まず，CVS リポジトリから CVS 情報を，電子メールアーカイブから電子メール情報を取得する．次に，これらの情報が相互に関連を持つ場合には両者を結合させて，統合情報が生成される．

- CVS 情報の取得

  本システムは，CVS リポジトリからリビジョン情報を取得する際に対象とするディレクトリを，設定ファイル（.spxrconf）の中で指定することが可能である．そこで，異なる規模のディレクトリを指定して，CVS 情報を取得する際にかかる実行時間をそれぞれ計測した．

新規作成（create）

ファイル総数	リビジョン総数	データサイズ	取得時間	指定したディレクトリ
76	499	826KB	3s	projects/
184	694	616KB	5s	distrib/
39	1651	24.2MB	8s	CVSROOT/
1735	12429	27.9MB	1m42s	www/
4375	38275	134MB	12m13s	doc/
50419	215355	292MB	47m13s	ports/
48129	432978	1.00GB	1h57m11s	src/
104957	701881	1.48GB	5h47m51s	./

更新（update）

ファイル総数	リビジョン総数	データサイズ	取得時間	指定したディレクトリ
76	499	826KB	2s	projects/
184	694	616KB	3s	distrib/
39	1679	24.6MB	4s	CVSROOT/
1763	12672	29.0MB	43s	www/
4407	38375	126MB	3m04s	doc/
50697	216794	289MB	14m21s	ports/
48961	435395	1.00GB	30m03s	src/
105409	704420	1.47GB	56m23s	./

図 20: CVS 情報の取得時間

計測結果を，図 20 に示す．

CVS 情報を新規作成する際には，更新する際よりも多くの時間がかかることがわかる．

ところが，CVS 情報を新規作成するのは，本システムを最初に使用する際のみである．一旦，CVS 情報を作成してしまえば，以降は追加された CVS 情報を更新するだけでよく，効率が良いことがわかる．本システムによる CVS 情報の更新は，新たに追加されたリビジョンの差分情報だけを取得して処理するため，新規作成する場合に比べてがコストが小さくなる．

また，CVS 情報の取得にかかる時間は，ファイル総数やリビジョン総数，CVS リポジトリのデータサイズにほぼ比例していることがわかる．特に相関が高いのは，リビジョン総数であると考えられる．

CVS 情報の取得時間とリビジョン総数との関係を，図 21 に示す．

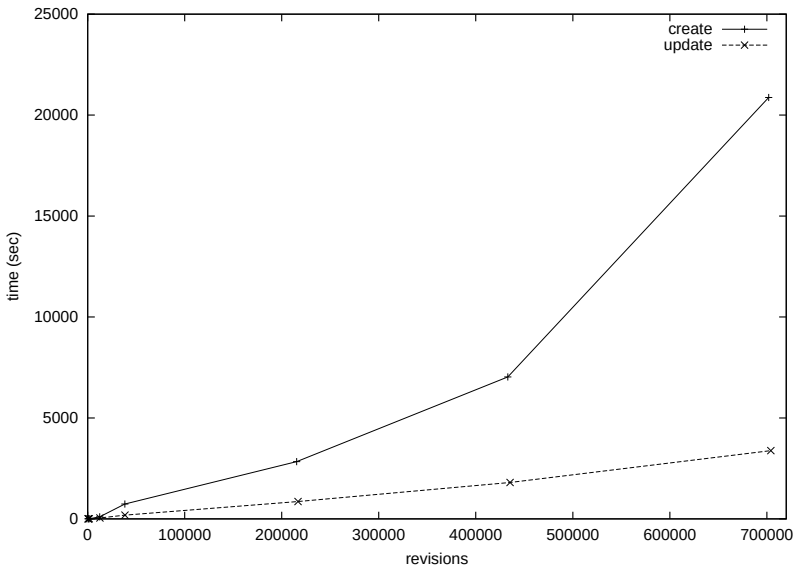

図 21: CVS 情報の取得時間とリビジョン総数の関係

また，新規作成，更新のいずれの場合にも，実時間で処理が行われることがわかる．今回の実験対象である FreeBSD の CVS リポジトリは，オープンソース開発の中でも大規模なものであるため，一般的な規模のオープンソース開発にも十分に適用可能であると考える．

- 電子メール情報の取得

  本システムは，電子メールアーカイブから電子メール情報を取得する際に対象とするアーカイブを，設定ファイル（`.spxrconf`）の中で指定することが可能である．そこ

で，異なる規模のアーカイブを指定して，電子メール情報を取得する際にかかる実行時間をそれぞれ計測した．

計測結果を，図 22 に示す．

電子メール総数	データサイズ	時間	指定したディレクトリ
2205	2MB	50s	1994/
69169	121MB	32m49s	1995/
99727	192MB	47m22s	1996/
119059	231MB	56m11s	1997/
174113	393MB	1h19m49s	1998/
190129	456MB	1h32m18s	1999/
216023	534MB	1h46m56s	2000/
242827	688MB	2h06m58s	2001/

図 22: 電子メール情報の取得時間 新規作成 (create)

電子メール情報の取得にかかる時間は，電子メール総数や電子メールアーカイブのデータサイズにほぼ比例していることがわかる．

また，CVS 情報の場合と同様に，電子メール情報を新規作成する際には，更新する際よりも多くの時間がかかる．ところが，電子メール情報を新規作成するのは，本システムを最初に使用する際のみである．一旦，電子メール情報を作成してしまえば，以降は追加された電子メール情報を更新するだけでよく，効率が良いことがわかる．本システムによる電子メール情報の更新は，新たに追加された電子メールの差分情報だけを取得して処理するため，新規作成する場合に比べてがコストが小さくなる．

また，新規作成，更新のいずれの場合にも，実時間で処理が行われることがわかる．今回の実験対象である FreeBSD の電子メールアーカイブは，オープンソース開発の中でも大規模なものであるため，一般的な規模のオープンソース開発にも十分に適用可能であると考える．

- 統合情報の生成

  異なる規模の CVS 情報と電子メール情報から，統合情報を生成する際にかかる実行時間をそれぞれ計測した．

  計測結果を，図 23 に示す．

  統合情報の生成にかかる時間は，CVS リビジョン総数や電子メール総数にほぼ比例していることがわかる．

  また，CVS 情報や電子メール情報の場合と同様に，統合情報を新規作成する際には，

CVS リビジョン総数	電子メール総数	時間
38275	7115	1m22s
215355	7115	1m45s
432978	7115	2m03s
38275	435395	1h12m14s
215355	435395	1h43m55s
432978	435395	2h24m23s

図 23: 統合情報の生成時間 新規作成 (create)

更新する際よりも多くの時間がかかる．ところが，統合情報を新規作成するのは，本システムを最初に使用する際のみである．一旦，統合情報を作成してしまえば，以降は追加された統合情報を更新するだけでよく，効率が良いことがわかる．本システムによる統合情報の更新は，新たに追加された CVS リビジョンや電子メールの差分情報だけを取得して処理するため，新規作成する場合に比べてがコストが小さくなる．

統合情報を生成するためには，CVS 情報と電子メール情報を総当たりで比較する手法が考えられる．しかし，この手法では，取得する情報の規模が大きくなると，処理にかかる時間が飛躍的に増大してしまう．そこで，本システムの実装では，取得した CVS 情報をハッシュ化する手法を用いて，その解決を目指している．ハッシュ化することにより，処理にかかる時間を線形増加で押さえることに成功している．このため，統合情報の生成にかかる時間はハッシュ化した CVS 情報の規模によらずほぼ一定であり，電子メール情報の規模に大きく依存していることがわかる．

また，新規作成，更新のいずれの場合にも，実時間で処理が行われることがわかる．従って，一般的なオープンソース開発にも十分に適用可能であると考える．

5.2.2 統合情報の妥当性

次に，本システムによって取得・生成された情報が，実際に妥当な関係を持つかどうか検証する．

CVS 情報と電子メール情報について，本システムが両者を結合して生成した統合情報 (以降，これをシステム統合情報と呼ぶ) と，開発者が実際に手作業で両者を結合して生成した情報 (以降，これを開発者統合情報と呼ぶ) を比較する．

ある一定の期間にやりとりされた電子メールの内容から，それと関連を持つ CVS のリビジョンを抽出する．そこから抽出されたシステム統合情報と開発者統合情報を比較して，システム統合情報と開発者統合情報に含まれるもの (a_1)，システム統合情報に含まれるが開発者統合情報には含まれないもの (a_2)，システム統合情報には含まれないが開発者統合情報

には含まれるもの (a_3) に分類して，それぞれの情報を解析する．その包含関係を，図 24 に示す．

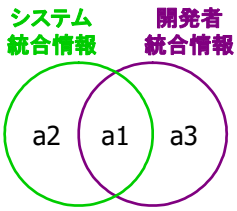

図 24: システム統合情報と開発者統合情報

ここでは，ある期間にやりとりされた電子メール 100 通を対象に実験を行った．
実験結果を，図 25 に示す．

メーリングリストの種類	a_1	a_2	a_3
cvs-all 1	90	2	8
cvs-all 2	82	4	14
cvs-all 3	85	3	12
freebsd-questions 1	35	12	53
freebsd-questions 2	41	10	49
freebsd-questions 3	37	14	49

図 25: 関連情報の妥当性

メーリングリスト cvs-all に流れる電子メールは，主にコミットメールが多い．コミットメールの場合には，更新されたファイルパスとリビジョン番号が記述されているため，関連を持つ CVS 情報が特定しやすい．従って，システム統合情報と開発者統合情報に含まれるものが多くなると考えられる．

これに対して，メーリングリスト freebsd-questions に流れる電子メールは，ほとんどが議論メールである．議論メールの場合には，本文中からキーワードを抽出して，CVS 情報のキーワードと比較するため，コミットメールの場合と比べると，関連を持つ CVS 情報が特定しにくい．従って，システム統合情報と開発者統合情報に含まれるものが少なくなり，一方に含まれるものが多くなると考えられる．

今後は，議論メールの関連を抽出する精度を改善する必要がある．

5.2.3 オープンソース開発環境の問題点

最後に，履歴データ表示システムを用いて取得・生成された情報を表示，検索することにより，2.2 節 (10 ページ) で説明した，オープンソース開発の問題点を解決していることを確認する．

- システム固有の情報蓄積

  CVS リポジトリのリビジョン情報は，ファイル単位でリビジョンが管理されているため，ファイル単位での履歴を取得することは容易に可能であるが，その他の視点から履歴を取得することは困難であった．例えば，ある開発者がこれまでに行った開発作業の履歴を取得したり，特定の日時に行われた更新作業の履歴を取得することは非常に難しいという問題点があった．

  ところが，本システムを利用することにより，ファイル単位だけでなく，更新日時単位，更新作業者単位，キーワード単位での履歴を取得することができる．従って，開発者が必要とする情報を容易に提供することが可能となり，先述の問題点を解決している．

  具体的には，以下のように問題点が解決される．

  本システムを利用することで，CVS リポジトリ内にある src/bin/ln/ln.c,v というファイルについて，ファイル単位でリビジョン情報を容易に取得可能できる．しかも，このファイルのあるリビジョン 1.19 について，更新作業者 sobomax がこれまでに行った開発作業の履歴や，更新日時 2001/04/26 17:15:57 と同時に行われた更新作業の履歴や，キーワード NetBSD に関連がある他のリビジョン情報を取得することが容易に可能である．

- システム間の連係不足

  CVS リポジトリを参照しながら，それに関連した電子メール上での話題を取得したい場合，あるいは，電子メールを参照しながら，それに関連した CVS リポジトリの情報を取得したい場合には，開発者が複数のシステムから個々に情報を取得して，それらを結びつける必要があった．しかし，そのように関連付けを開発者が行う場合のコストは，開発規模が増大するにつれて非常に大きくなるという問題点があった．

  ところが，本システムを利用することにより，CVS 情報と電子メール情報を結合して生成された統合情報を利用して，情報の検索を行うことが可能である．従って，開発者が複数の情報を関連付けするコストが削減できることになり，先述の問題点を解決している．

具体的には，以下のように問題点が解決される．

本システムを利用することで，CVS リポジトリ内の src/bin/ln/ln.c,v というファイルのあるリビジョン 1.19 の情報を参照した際に，それに関連した電子メール上での話題を取得したい場合には，ファイルパスやリビジョン番号，ログメッセージから抽出したキーワードから，そのリビジョン情報と関連を持つ電子メールを容易に取得することができる．あるいは，ある電子メールを参照しながら，それに関連した CVS リポジトリのリビジョン情報を取得したい場合には，電子メールのサブジェクトや本文から抽出したキーワードから，その電子メールと関連を持つリビジョン情報を容易に取得することができる．

6 関連研究

近年では，オープンソース開発で利用されるシステムが蓄積した情報をデータベース化して，開発者に情報を提供するさまざまな研究が行われている．

例えば，CVSSearch[5] や FreshSource[37]，FreshPorts[8] 等のシステムが存在する．

CVSSearch は，CVS リポジトリの RCS ファイルを解析して，各リビジョンの更新ログメッセージと，リビジョン間でソースコードの差分情報を取得してデータベース化し，両者の間で関連付けを行って，開発者にその情報を提供するシステムである．

FreshSource は，FreeBSD のメーリングリストが管理する電子メールアーカイブを XML データベース化することにより，膨大な情報を効率良く管理して，開発者が必要とする情報を提供するシステムである．

FreshPorts は，CVS リポジトリの中から最新の更新情報を取得して表示することにより，開発者が容易に開発作業の流れを把握できるシステムである．

しかし，これらのシステムは，ある一つのシステムが蓄積した情報を対象としているものであり，本研究のように複数のシステムを対象としているものは少ない．従って，現状のオープンソース開発環境の問題点の一つである「システム固有の情報蓄積」を解決するものであり，「システム間の連係不足」を解決するには至っていないと考えられる．

また，我々の研究グループでは，複数の電子メールから関連のある電子メールを検索するシステムに関する研究 [34] が行われている．この研究では，従来のサブジェクトやメールヘッダ情報に基づいた電子メールのスレッド化ではなく，メール本文に含まれるキーワードを抽出し，それに基づいてメール相互間に関連の強弱を持たせてスレッドの構築を行っている．本システムの電子メール情報生成ツールにこの手法を組み込むことで，複数の情報間に存在する関連生成の大幅な改善が期待できる．

7　まとめ

本研究では，オープンソース開発を研究対象として，まず，現状のオープンソース開発環境における二つの問題点を指摘した．各システムがそれぞれ固有の形式で開発情報を蓄積する「システム固有の情報蓄積」の問題と，各システムが互いに連係を持たず，開発情報が分散して存在する「システム間の連係不足」の問題により，開発者が大きな負担を強いられていると考える．

そこで，これらの問題点を解決し，数年間のオープンソース開発で蓄積された膨大なソフトウェアプロダクトを開発者が有効に利用することを目的とした，オープンソース開発向けの開発支援環境を提案した．

さらに，その構成要素の一つであるオープンソース開発支援用ソースコード・メールの履歴対応表示システム (SPxR) を設計し，その実装を行った．本システムは，「履歴データベースシステム」と「履歴データ表示システム」という二つのサブシステムから構成される．履歴データベースシステムは，perl と PostgreSQL データベースを利用しており，CVS 情報と電子メール情報を取得し，それらの中で相互に関連を持つ情報から統合情報を生成して，それぞれデータベース化して管理する．履歴データ表示システムは，perl による CGI を利用しており，開発者と履歴データベースシステムを結ぶインターフェースとしての役割を果たし，履歴データベースシステムと連係して，開発者が必要とする情報を提供する．本システムを利用することにより，開発者が各システムからそれぞれ開発情報を取得するために大きな負担を強いられることなく，開発者に必要な情報を的確に提供することが可能となるため，開発作業を支援することができる．

また，実際のオープンソース開発で用いられているデータを用いて，本システムの適用実験を行った．その結果，本システムは大規模なオープンソース開発に適用した場合でも実時間で動作することを確認した．さらに，本システムを用いることにより，開発者が必要とする情報を提供することが可能となること，現状のオープンソース開発環境における問題点が解決されることを確認した．

本システムを実際のオープンソース開発に適用するためには，実行速度の向上と，複数の情報間に存在する関連生成の改善が望まれる．実行速度の向上については，データベースの構築やテキスト処理を工夫することが可能であり，改善の余地が残されていると考える．関連生成の改善については，例えば，関連の有無だけではなく，関連に強弱を持たせてスコア付けを行う手法を利用することにより，さらなる向上が見込まれる．

また，本システムを実際にオープンソース開発に適用することで，実際にオープンソース開発に携わる開発者からのフィードバックを得られる．これにより，さらなる成熟と信頼性の向上を経て，本システムが現状のオープンソース開発環境に貢献することを目指す．

謝辞

本論文を作成するにあたり，常に適切な御指導を賜わりました大阪大学大学院基礎工学研究科情報数理系専攻 井上 克郎 教授に心より深く感謝致します．

本論文の作成にあたり，適切な御指導および御助言を頂きました大阪大学大学院基礎工学研究科情報数理系専攻 楠本 真二 助教授に深く感謝致します．

本論文の作成にあたり，適切な御指導および御助言を頂きました大阪大学大学院基礎工学研究科情報数理系専攻 松下 誠 助手に深く感謝致します．

本論文の作成において，適切な御助言を頂きました大阪大学大学院基礎工学研究科情報数理系専攻 山本 哲男 氏に深く感謝致します．

本論文の作成において，ご協力を頂きました大阪大学基礎工学部情報科学科 高尾 祐治 君に深く感謝致します．

最後に，その他様々な御指導，御助言等を頂いた大阪大学大学院基礎工学研究科情報数理系専攻井上研究室の皆様に深く感謝致します．

参考文献

[1] The Apache Software Foundation, Apache Projects, http://www.apache.org/.

[2] Ulf Asklund, Boris Magnusson, and Annita Persson, "Experiences: Distributed Development and Software Configuration Management", In Proceedings of SCM-9, Ninth International Symposium on System Configuration Management, J. Estublier (Ed.), LNCS1675, pp.17–33, Toulouse, France, 1999.

[3] Wayne A. Babich, "Software Configuration Management", Addison-Wesley, Reading, Massachusetts, 1986.

[4] Brian Berliner, "CVS II:Parallelizing Software Development", In USENIX Association, editor, Proceedings of the Winter 1990 USENIX Conference, pages 341–352, Berkeley, CA, USA, 1990.

[5] Annie Chen, Eric Chou, Joshua Wong, Andrew Y. Yao, Qing Zhang, Shao Zhang, Amir Michail, "CVSSearch: Searching through Source Code using CVS Comments", International Conference on Software Maintenance, pp.364–373, 2001.

[6] Collab. Net, Inc., SourceCast, http://www.collab.net/products/sourcecast/.

[7] CVSWeb, http://stud.fh-heilbronn.de/ zeller/cgi/cvsweb.cgi/.

[8] DVL Software Limited., FreshPorts, http://freshports.org/.

[9] Jacky Estublier, "Software Configuration Management: A Roadmap". The Future of Software Engineering in 22nd ICSE, pp.281–289, 2000.

[10] Jacky Estublier and Rubby Casallas, "The Adele Configuration Manager", In Walter Tichy, editor, Configuration Management, pp.99–133, John Wiley and Sons, Ltd., Baffins Lane, Chichester, West Sussex PO19 1UD, England, 1994.

[11] Peter H. Feiler, "Configuration Management Models in Commercial Environments", CMU/SEI-91-TR-7 ESD-9-TR-7, March, 1991.

[12] Karl Fogel, “Open Source Development with CVS”, The Coriolis Group, 2000.

[13] Free Software Foundation, Inc., The GNU Project,
http://www.gnu.org/.

[14] Peter Fröhlich and Woflgang Nejdl, “WebRC Configuration Management for a Cooperation Tool”, SCM-7, LNCS 1235, pp.175–185, 1997.

[15] 藤原晃, “類似度を用いたプログラムの再利用性尺度の提案と実現”, 大阪大学大学院基礎工学研究科修士学位論文, 2002.

[16] André van der Hoek, Dennis Heimbigner, and Alexander L. Wolf, “A generic, peer-to-peer repository for distributed configuration management”, Proceedings of ICSE-18, pp.298–307, Berlin, Germany, March 1996.

[17] James J. Hunt, Frank Lamers, Jürgen Reuter, and Walter F. Tichy, “Distributed Configuration Management via Java and the World Wide Web”, ICSE’97, SCM-7, LNCS1235, pp.161–174, 1997.

[18] 石川武志, 山本哲男, 松下誠, 井上克郎, “ソフトウェア開発時における版管理システムを利用したコミュニケーション支援システムの提案”, 情報処理学会研究報告, 2001-SE-133, Vol.2001, No.92, pp.23–30, 2001.

[19] Tapani Kilpi, “Product Management Requirement for SCM Discipline”, SCM-7, LNCS 1235, pp.175–185, 1997.

[20] 鯉江英隆, 西本卓也, 馬場肇, “バージョン管理システム (CVS) の導入と活用”, SOFT BANK, December, 2000.

[21] Yi-Jing Lin and Steven P.Reiss, “Configutarion Management with Logical Structures”, Proceedings of ICSE-18, pp.298–307, Berlin, Germany, March 1996.

[22] Linux Online Inc., The Linux Home Page,
http://www.linux.org/.

[23] 松下誠, 井上克郎, “自由な開発形態を支援するソフトウェア開発環境”, ソフトウェアシンポジウム 2000 論文集, pp.236–242, 2000.

[24] Merant, Inc., PVCS Home Page,
http://www.merant.com/pvcs/.

[25] Microsoft Corporation, Microsoft Visual SourceSafe, http://msdn.microsoft.com/ssafe/.

[26] Bartosz Milewski, “Distributed Source Control System”, SCM-7, LNCS 1235, pp98–107, 1997.

[27] 落水浩一郎，“分散共同ソフトウェア開発に対するソフトウェアプロセスモデルに関する基礎考察”, 電子情報通信学会技術研究報告，SS2000-48(2001-01), pp.49–56, 2001.

[28] Koichiro Ochimizu, Hiroyuki Murakoshi, Kazuhiro Fujieda, and Mitsunori Fujita, “Sharing Instability of a Distributed Cooperative Work”, ISPSE2000, pp.33–42, 2000.

[29] 大月美佳, “入門 CVS Concurrent Versions System”, SHUWA SYSTEM Co., Ltd, 2001.

[30] Open Source Development Lab, Inc., Open Source Development Lab, http://www.osdlab.org/.

[31] Rational Software Corporation, Software configuration management and effective team development with Rational ClearCase, http://www.rational.com/products/clearcase/.

[32] Eric S. Raymond, “The Cathedral & the Bazaar”, O’REILLY, 1999.

[33] 田原靖太, “既存ソフトウェアの変更履歴を利用したソースコード修正支援システム”, 大阪大学大学院基礎工学研究科修士学位論文, 2002.

[34] 高尾祐治, “オープンソース開発支援用メール検索システムの試作”, 大阪大学基礎工学部情報科学科特別研究報告, 2002.

[35] 寺口正義, 松下誠, 井上克郎, “バージョン間の差分を利用したデバグ手法の提案”, 電子情報通信学会技術研究報告, SS99-52, pp.17–24, 2000.

[36] The FreeBSD Project, The FreeBSD Project, http://www.freebsd.org/.

[37] The FreshSource project, FreshSource, http://freshsource.org/.

[38] The Mozilla Organization, bonsai, http://www.mozilla.org/bonsai.html.

[39] Walter F. Tichy, "RCS - A System for Version Control", SOFTWARE - PRACTICE AND EXPERIENCE, VOL.15(7), pp.637–654, 1985.

[40] VA Linux Systems, Inc., SourceForge, http://sourceforge.net/.

[41] Yunwen Ye, Gerhard Fischer, and Brent Reeves, "Integrating Active Information Delivery and Reuse Repository Systems", Proceeding of the ACM SIGSOFT Eighth International Symposium on the Foundations of Software Engineering (FSE-8), pp.60–68, 2000.

付録

1. RCS ファイル
2. 電子メール
3. 設定ファイル (.spxrconf)

1 RCS ファイル

- RCS ファイルの書式 (rcsfile(5) の一部を抜粋引用)

```
rcstext ::=  admin {delta}* desc {deltatext}*
admin ::=  head {num};
           { branch {num}; }
           access {id}*;
           symbols {sym : num}*;
           locks {id : num}*; {strict  ;}
           { comment {string}; }
           { expand {string}; }
           { newphrase }*
delta ::=  num
           date num;
           author id;
           state {id};
           branches {num}*;
           next {num};
           { newphrase }*
desc ::=  desc string
deltatext ::=  num
               log string
               { newphrase }*
               text string
num ::=  {digit | .}+
digit ::=  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
id ::=  {num} idchar {idchar | num}*
sym ::=  {digit}* idchar {idchar | digit}*
idchar ::=  special を除く任意の可視文字 (visible graphic character)
special ::=  $ | , | . | : | ; | @
string ::=  @{任意の文字、ただし @ は 2 重化される}*@
newphrase ::=  id word* ;
word ::=  id | num | string | :
```

- RCS ファイルの具体例 (src/bin/ln/ln.c,v の一部を抜粋引用)

```
head    1.22;
access;
symbols
        RELENG_4_5_0_RELEASE:1.15.2.3
           .
           .
        ( 中略 )
           .
           .
        CSRG:1.1.1;
locks; strict;
comment @ * @;
           .
           .
        ( 中略 )
           .
           .
1.20
date    2001.04.26.17.22.48;    author sobomax; state Exp;
branches;
next    1.19;

1.19
date    2001.04.26.17.15.57;    author sobomax; state Exp;
branches;
next    1.18;
           .
           .
        ( 中略 )
           .
           .
desc
@@
           .
           .
        ( 中略 )
           .
           .
1.20
log
@Previous commit should read:
```

```
style(9) Reviewed by:   bde
@
text
@d45 1
a45 1
  "$FreeBSD: src/bin/ln/ln.c,v 1.19 2001/04/26 17:15:57 sobomax Exp $";
d67 1
a67 1
int     linkit __P((char *, char *, int));
d77 1
a78 1
        char *p, *sourcedir;
d158 1
a158 1
        char *target, *source;
d162 2
a164 1
        char *p, path[MAXPATHLEN];
@


1.19
log
@Bring in ‘-h’ compatability option and its alias ‘-n’ to match NetBSD and GNU
semantics.

style(9) Reviewed by:
Obtained from:  NetBSD
@
text
@d45 1
a45 1
  "$FreeBSD: src/bin/ln/ln.c,v 1.18 2000/08/17 16:08:06 sheldonh Exp $";
@
```

```
           .
           .
        (以下略)
           .
           .
```

2 電子メール

- コミットメールの具体例

```
From owner-cvs-all  Thu Apr 26 10:16: 4 2001
Delivered-To: cvs-all@freebsd.org
Received: from freefall.freebsd.org (freefall.freebsd.org [216.136.204.21])
        by hub.freebsd.org (Postfix) with ESMTP
        id D993837B423; Thu, 26 Apr 2001 10:15:57 -0700 (PDT)
        (envelope-from sobomax@FreeBSD.org)
Received: (from sobomax@localhost)
        by freefall.freebsd.org (8.11.1/8.11.1) id f3QHFvs90663;
        Thu, 26 Apr 2001 10:15:57 -0700 (PDT)
        (envelope-from sobomax)
Message-Id: <200104261715.f3QHFvs90663@freefall.freebsd.org>
From: Maxim Sobolev <sobomax@FreeBSD.org>
Date: Thu, 26 Apr 2001 10:15:57 -0700 (PDT)
To: cvs-committers@FreeBSD.org, cvs-all@FreeBSD.org
Subject: cvs commit: src/bin/ln ln.1 ln.c
X-FreeBSD-CVS-Branch: HEAD
Sender: owner-cvs-all@FreeBSD.ORG
Precedence: bulk
X-Loop: FreeBSD.ORG

sobomax     2001/04/26 10:15:57 PDT

  Modified files:
    bin/ln               ln.1 ln.c
  Log:
  Bring in ‘-h’ compatability option and its alias ‘-n’ to match NetBSD and GNU
  semantics.

  style(9) Reviewed by:
  Obtained from:        NetBSD

  Revision  Changes    Path
  1.15      +27 -7     src/bin/ln/ln.1
  1.19      +28 -9     src/bin/ln/ln.c


To Unsubscribe: send mail to majordomo@FreeBSD.org
with "unsubscribe cvs-all" in the body of the message
```

- 議論メールの具体例

```
From owner-cvs-all  Thu Apr 26 10:34:28 2001
Delivered-To: cvs-all@freebsd.org
Received: from whale.sunbay.crimea.ua (whale.sunbay.crimea.ua [212.110.138.65])
        by hub.freebsd.org (Postfix) with ESMTP
        id 45AE437B424; Thu, 26 Apr 2001 10:34:21 -0700 (PDT)
        (envelope-from ru@whale.sunbay.crimea.ua)
Received: (from ru@localhost)
        by whale.sunbay.crimea.ua (8.11.2/8.11.2) id f3QHYIh73055;
        Thu, 26 Apr 2001 20:34:18 +0300 (EEST)
        (envelope-from ru)
Date: Thu, 26 Apr 2001 20:34:18 +0300
From: Ruslan Ermilov <ru@FreeBSD.org>
To: Maxim Sobolev <sobomax@FreeBSD.org>
Cc: cvs-committers@FreeBSD.org, cvs-all@FreeBSD.org
Subject: Re: cvs commit: src/bin/ln ln.1 ln.c
Message-ID: <20010426203418.A72921@sunbay.com>
Mail-Followup-To: Maxim Sobolev <sobomax@FreeBSD.org>,
        cvs-committers@FreeBSD.org, cvs-all@FreeBSD.org
References: <200104261715.f3QHFvs90663@freefall.freebsd.org>
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Disposition: inline
User-Agent: Mutt/1.2.5i
In-Reply-To: <200104261715.f3QHFvs90663@freefall.freebsd.org>;
        from sobomax@FreeBSD.org on Thu, Apr 26, 2001 at 10:15:57AM -0700
Sender: owner-cvs-all@FreeBSD.ORG
Precedence: bulk
X-Loop: FreeBSD.ORG

On Thu, Apr 26, 2001 at 10:15:57AM -0700, Maxim Sobolev wrote:
> sobomax     2001/04/26 10:15:57 PDT
>
>   Modified files:
>     bin/ln               ln.1 ln.c
>   Log:
>   Bring in ‘-h’ compatability option and its alias ‘-n’ to match NetBSD
>   and GNU semantics.
>
>   style(9) Reviewed by:
>   Obtained from:      NetBSD
>
>   Revision  Changes    Path
```

```
>   1.15      +27 -7     src/bin/ln/ln.1
>   1.19      +28 -9     src/bin/ln/ln.c
>
These options should be documented as non-standard under the COMPATIBILITY
manpage section.


Cheers,
--
Ruslan Ermilov          Oracle Developer/DBA,
ru@sunbay.com           Sunbay Software AG,
ru@FreeBSD.org          FreeBSD committer,
+380.652.512.251        Simferopol, Ukraine

http://www.FreeBSD.org  The Power To Serve
http://www.oracle.com   Enabling The Information Age

To Unsubscribe: send mail to majordomo@FreeBSD.org
with "unsubscribe cvs-all" in the body of the message
```

3 設定ファイル(.spxrconf)

```
### CVS config

CVS ROOT :: /pub/FreeBSD/FreeBSD.cvs
CVS PARSE ROOT :: /pub/FreeBSD/FreeBSD.cvs
CVS TOTAL FILES :: 104737
CVS TOTAL REVISIONS :: 701798
CVS LAST MODIFIED :: Mon Jan 28 00:39:29 2002 (1012178369)


### MAIL config

MAIL ROOT :: /pub/FreeBSD/archive/
MAIL PARSE ROOT :: /pub/FreeBSD/archive/2001/cvs-all
MAIL TOTAL COUNTS :: 49736
MAIL LAST MODIFIED :: Tue Jan 29 19:21:18 2002 (1012332078)


### INTEG config

INTEG CVS TOTAL COUNTS :: 701798
INTEG MAIL TOTAL COUNTS :: 49736
INTEG LAST MODIFIED :: Wed Jan 30 03:06:54 2002 (1012360014)
```