The University of Osaka
Institutional Knowledge Archive

Title	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
Author(s)	石川, 武志; 山本, 哲男; 松下, 誠 他
Citation	情報処理学会研究報告. ソフトウェア工学研究会報告. 2001, 2001-SE-133(92), p. 23-30
Version Type	VoR
URL	https://hdl.handle.net/11094/50200
rights	ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

ソフトウェア工学 133－4
（2001．9．19）

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

石川武志　山本哲男　松下誠　井上克郎

大阪大学大学院基礎工学研究科
〒560-8531 大阪府豊中市待兼山町 1-3

近年，インターネットの発展に伴い，オープンソースソフトウェアと版管理システムを中心とする開発スタイルが注目されている．この種の開発では，開発者間のコミュニケーションを支援することが重要な課題の 1 つであるが，既存の環境は十分な支援を行っているとは言えない．本研究では，我々が現在開発を進めている開発支援環境の構成要素である，版管理システムの持つ情報を利用した開発者間コミュニケーション支援システムの提案を行う．本システムでは，過去のコミュニケーション履歴と版管理システムへ蓄積された開発履歴を統合したデータベースを用いて，コミュニケーション時に必要となる情報を提供する．これにより，開発者の負担を軽減することが期待できる．

Design of Communication Supporting System Cooperating with Revision Control System

Takeshi Ishikawa, Tetsuo Yamamoto,
Makoto Matsushita and Katsuro Inoue

Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama-cho, Toyonaka,
Osaka 560-8531, Japan

Open source software and its development style are recently hot topics of software development. Usually, a revision control system is used in development environments. Supporting communications between developers is one of important topics, however, existing environments do not support them well. In this paper, we propose a communication supporting system that is a component of our development-supporting environment. In this system, we build an integrated database that is based on revision history and communication history. This system enables developers to reduce communication costs and improve software quality.

1 はじめに

ソフトウェアの開発形態が多人数化，分散化して久しい．ある一つの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である [1]．また，ネットワーク環境の充実に伴い，各開発者がそれぞれ異なる拠点で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 (Open Source Software, OSS)[15] が注目されている．OSS では，開発中のソースコードを広く公開することにより，世界中の開発者が開発へ自由に参加することが可能である．オープンソースソフトウェアの開発 (OSS 開発) 作業では，ソースコードの管理のために版管理システムが広く用いられており，開発環境の基礎を成している．

版管理システムを用いた OSS 開発では，各地に分散した複数の開発者が，単一の版管理システムを利用して同時に作業を行う．また，多く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っていることが多い．このため，並列的に作業を行っている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，連絡や意志疎通が不足してしまう傾向がある．従って，開発者間のコミュニケーションを支援することが重要な課題である．

OSS 開発では，開発者間のコミュニケーションを支援するために，電子メールや会議システム等の CSCW ツールが広く用いられている．しかし，既存のツールは互いに独立したものであり，版管理システムとの連係も十分ではない．このため，開発者はそれぞれのツールを使い分け，必要な情報を収集する必要がある．また，開発者間で行われたコミュニケーションの記録は各ツールに分散して保存されるため，現在どのような話題が開発者間に存在し，どのような議論の状態であるかを把握することが非常に困難である．

我々は現在，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 OSS 開発を対象として，分散開発支援環境の構築について研究を行っている．本支援環境は，OSS 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．

本研究では，本支援環境における開発者のコミュニケーションを支援するシステムについて述べる．本システムでは，既存の版管理システムや電子メールのアーカイブ等と，それらが互いに分散して保持している各種の情報を，有機的に結合させた統合情報を管理するデータベースシステムから構成される．本システムを用いることにより，開発者が必要な情報を参照するための作業効率が良くなり，電子メール等による開発者間のコミュニケーションを円滑に行うことが可能になる．

提案するコミュニケーション支援システムを用いることにより，OSS 開発において，限られた時間を有効に利用したコミュニケーション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．

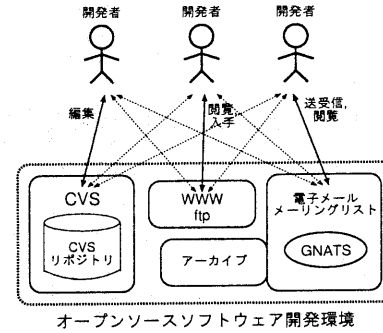

図 1: OSS 開発環境の構成の例

2 オープンソースソフトウェア

本節では，我々が研究の対象としているオープンソースソフトウェア開発について説明する．

2.1 オープンソースソフトウェア開発

OSS は，複数の開発者が並列的に開発作業を行うことができるように，開発中のソースコードを広く公開するソフトウェアである．OSS 開発には，世界中の開発者が，潜在的に，いつでも開発に参加することが可能である．実際に，FreeBSD や GNU Software，Linux といったソフトウェアの開発でもこの手法が用いられており，世界中の開発者が参加している．

OSS 開発環境の構成の例を図 1 に示す．OSS 開発環境は，多くのツールから構成されている．例えば，多数の開発者によって並列的にソースコードが扱われることが多いため，版管理システムの一つである CVS(Concurrent Versions System)[2][6] が広く用いられている．他にも，電子メールやメーリングリストとそのアーカイブ，GNATS(GNU Problem Report Management System) を用いたバグデータベース管理，WWW を用いた電子メールによる議論の内容検索，作成したファイルを配布するための ftp 等から構成される．

2.2 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，その開発過程を履歴として管理するシステムであり，開発プロセスの作成や再利用の際に役立つ [5]．

版管理システムは，リポジトリに格納されたファイルをリビジョン単位で管理する．このため，内部的にリビジョン情報を作成し，保持している．リビジョン情報として扱われる情報は，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，コメント等である．

版管理システムの実装は数多く存在し，また，実際に利用されている．多くの場合，CVS や RCS[18] 等のシステムが利用されている．また，ClearCase[14]，Visual SourceSafe[11] や PVCS[10] 等，商用の版管理システムも利用されている．また，版管理システムを用いて，ソフトウェア開発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 [16].

3 オープンソースソフトウェア開発の問題点

版管理システムを用いた OSS 開発では，分散した複数の開発者が同時に作業を行っている．それに伴い、現状ではいくつかの問題点を抱えている．以下では，本研究で対象とする問題点を挙げる．

3.1 開発者間のコミュニケーション支援

OSS 開発に参加している開発者は，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っていることが多い．このため，並列的に作業を行っている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，開発者間のコミュニケーションが不足してしまう傾向がある．従って，この種の開発においては，開発者間のコミュニケーションを支援することが重要な課題である．ここでは，問題点として，以下の二つを挙げる．

3.1.1 リビジョン情報の情報量不足

現状では，多くの開発者が版管理システムリポジトリから取得できるリビジョン情報を参照しながら，電子メール等を用いてコミュニケーションを行っている．しかし，版管理システムリポジトリから取得できるリビジョン情報には，コミュニケーション時の記録が含まれていないという問題がある．

例えば，あるリビジョンを更新する際に，t-yamamt，matusita, inoue という 3 人の開発者が電子メールを利用してその開発について議論した後，t-yamamt がリポジトリ内のリビジョンを更新したとする．この場合，議論に関らなかった t-ishikw が，後からこのリビジョン情報を参照しても，更新者として t-yamamt の名前が記録されるだけである．すなわち，そのリビジョン開発に深く関った開発者であるはずの matusita, inoue に関する情報は，電子メールのアーカイブを参照しなければわからない．

3.1.2 ツール／機能間の連係不足

OSS 開発において，開発管理や開発者間でのコミュニケーションを効率良く行うことを目的として，SourceForge[19] や SourceCast[3]，OSDL(Open Source Development Lab.)[13] 等のサービスが提供されている．これらのシステムは，電子メールや会議システム等の汎用的な CSCW ツール，WWW，CVS 等により，多くの機能を同時に提供している．しかし，提供されている各ツール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．また，行われたコミュニケーションは各ツールに分散して記録されているが，版管理システムとの連係は十分ではない．そのため，現状でどのような話題が開発者間に存在し，どういう議論の状態であるかを把握する際に，必要な情報を参照することが非常に困難である．

例えば，ある OSS 開発に t-ishikw が途中参加する場合を考える．この開発では，CVS，そのリビジョン情報を閲覧するために CVS 関連ツール bonsai[17]，電子メールとそのアーカイブを利用しているとする．t-ishikw は，これまでの開発状況を理解するために，アーカイブを参照した．「入出力の最適化」の議論が途中で放置されているため，t-ishikw は bonsai を用いて CVS のリビジョン情報を参照した．ところが，t-ishikw が思い当たるリビジョン情報にその内容は記載されていなかった．実際は，全く別のリビジョンで「入出力の最適化」が行われたのだが，t-ishikw が参照したアーカイブには，その内容が反映されていなかった．このように，電子メールのアーカイブと CVS の間で連係が取られていないために，開発者に対して負担が大きくなっている．

3.2 ソフトウェアの再利用

OSS 開発において，ソフトウェアの再利用を行う際の問題点には，以下のようなものがある．

3.2.1 再利用可能なソフトウェア部品の認識不足

開発者が再利用しようと思わなければ，ソフトウェア部品が再利用可能でも無用な存在になってしまう [20]．ソースコードの断片等の再利用可能な部品コンポーネントの存在を，開発者に認識させることが難しいという問題点が挙げられる．OSS 開発では，互いに分散した環境で作業が行われるため，他の作業者による開発プロダクトの存在を知ることが難しい．

例えば，t-ishikw が途中参加した OSS 開発では，あるシステムの入出力コンポーネントを過去に何回か作成していた．そのため，このコンポーネントは再利用可能な部品として実装されていた．t-ishikw は，ある部分の入出力を実装しようとしたが，開発全体の状況を把握しきれていないために，その存在を知らなかった．そのため，t-ishikw は，結果として余分な労力をその作業につぎこむことになった．

3.2.2 再利用可能なソフトウェア部品の単位

また，再利用可能なソフトウェア部品を，どの単位で扱うべきなのかという問題点が挙げられる．

オブジェクト指向プログラミング言語で記述されたソフトウェアでは，「クラス単位」，「メソッド単位」での再利用が一般的に行われている．しか

し，OSS 開発では，他者の開発プロダクトや，以前に開発されたソフトウェアの一部に手を加えて別の開発に役立てることが多い．そのため，「クラス単位」，「メソッド単位」のように構文的な単位で部品を再利用できるよりも，複数のクラスやメソッドの一部から機能的，あるいは，意味的な単位で部品を抽出して再利用できることが求められている．

例えば，t-ishikw が必要とする機能を持つ部品は，クラス A とクラス B の一部を組み合わせることにより実現できることがわかっているとする．この場合，クラス単位，あるいはメソッド単位での再利用ができたとしても，必要とする部分を部品として抽出できない．

4 OSS 開発向けの支援環境

我々は現在，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 OSS 開発を対象として，分散開発支援環境の構築について研究を行っている．本支援環境は，OSS 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．

4.1 支援環境の設計方針

本支援環境の設計方針を以下に示す．

- 既存の開発環境を壊さないようにする
  本支援環境では，現状の OSS 開発で用いられている版管理システムと開発者間でのコミュニケーションシステムを利用する．具体的には，CVS や，電子メールとそのアーカイブはそのまま変更することなく用いる．すなわち，開発者がこれまでに利用可能であった機能を，本支援環境により失うことはない．
- 既存のツールが保持する情報を有効に利用する
  例えば，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．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有効に利用することができる．

4.2 支援環境の構成

OSS 開発向けの支援環境の構成を，図 2 に示す．本支援環境は，コミュニケーション支援システムとソフトウェア再利用支援システムから構成される，この両システムは，統合情報データベースシステムを中心に構成され，CVS や電子メールのアーカイブと連係を持つ．

- コミュニケーション支援システム
  コミュニケーション支援システムは，統合情報データベースシステムと連係して，開発者に統合情報を提供することで，開発者間のコミュニケーションが円滑に行われるように支援する．
- ソフトウェア再利用支援システム
  ソフトウェア再利用支援システムは，CVS リポジトリを再利用コードリポジトリとして扱う．OSS 開発に適した新たな再利用手法により，統合情報データベースシステムと連係して，再利用可能なコンポーネント部品の管理を行う．

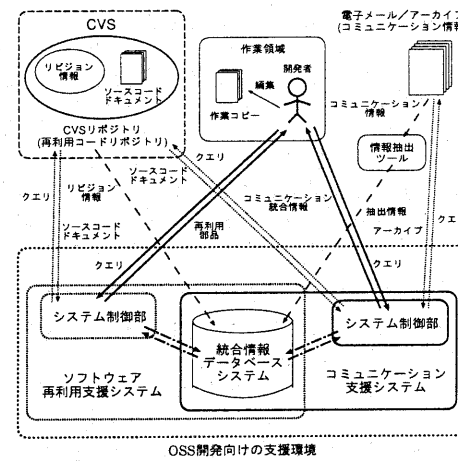

図 2: OSS 開発向けの支援環境の構成

- 統合情報データベースシステム
  統合情報データベースシステムは，CVS リポジトリと電子メールのアーカイブが互いに分散して保持している各種の情報を，有機的に結合させて統合情報として管理する．
- CVS
  CVS は，従来の版管理システムとしての役割の他に，ソフトウェアの再利用コードリポジトリとして使用される．
- 電子メール／アーカイブ
  電子メールとそのアーカイブは，開発者間でコミュニケーションが行われた記録を保持している．
- 情報抽出ツール
  情報抽出ツールは，電子メールとそのアーカイブから，有用な情報を抽出する．

5 コミュニケーション支援システム

ここでは，前節で述べた支援環境の構成要素の一つであるコミュニケーション支援システムについて述べる．

5.1 システムの構成

コミュニケーション支援システムのモデルを，図 3 に示す．本支援システムは，統合情報データベースシステムと，システム制御部から構成される．

- 統合情報データベースシステム
  統合情報データベースシステムは，本支援システムの根幹を成すシステムである．CVS が保持するリビジョン情報と，電子メールのアーカイブ等が保持する有用な情報を有機的に結合させて，統合情報として管理する．

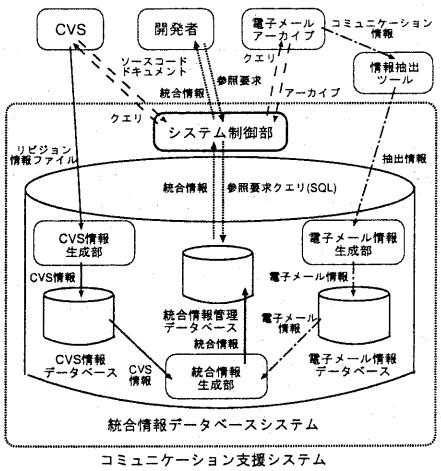

図 3: コミュニケーション支援システム

- システム制御部
  システム制御部は，統合情報データベースシステムと連係して，開発者から参照要求を受け付けて，その参照結果として統合情報を提供する．

5.2 統合情報データベースシステム

本データベースシステムは，CVS 情報生成部，CVS 情報データベース，電子メール情報生成部，電子メール情報データベース，統合情報生成部，統合情報管理データベースから構成される．

5.2.1 CVS 情報データベースの構築

CVS 情報生成部は，CVS から取得したリビジョン情報を保持するファイルから，CVS 情報を記録する．CVS 情報データベースは，生成部によって生成された CVS 情報を管理する．

CVS 情報生成部は，CVS リポジトリ内にあるリビジョン情報の中から，CVS 情報として以下の 6 つを抽出する．

- ファイルパス
- リビジョン番号
- 更新日時
- 更新作業者
- ログメッージ
- ブランチタグ情報

これらの情報は，CVS リポジトリ内の特定のファイルを解析して取得可能である．抽出された情報は，CVS 情報データベースに格納される．また，新たにリビジョンが更新された場合には，そのリビジョンの情報を自動的に取得して，CVS 情報データベースに格納する．

この中で，キーとなる要素は，ファイルパスとリビジョン番号である．そこで，処理を簡単化するため，これらの組合せに識別番号を対応させ，「識別番号」として CVS 情報の要素に追加する．

また，これらの CVS 情報の中に，「同じ更新作業者が，同時に更新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性が高い」という関係が成り立つ．そこで，これらのリビジョンに関する CVS 情報には，互いに高い関連性を持つリビジョンを ID 番号で記録するため，「関連識別番号」として CVS 情報の要素に追加する．

識別番号	c382
ファイルパス	test/hello.cpp,v
リビジョン番号	1.3
更新日時	2001/08/20 17:05:33
更新作業者	t-ishikw
ログメッセージ	optimized
ブランチタグ情報	(none)
関連識別番号	c383

識別番号	c383
ファイルパス	test/goodby.cpp,v
リビジョン番号	1.5
更新日時	2001/08/20 17:05:34
更新作業者	t-ishikw
ログメッセージ	optimized
ブランチタグ情報	(none)
関連識別番号	c381, c382

図 4: CVS 情報の例

例えば，互いに高い関連性を持つ二つのリビジョン情報から，図 4 に示す情報が本データベースに格納される．識別番号は，データベースにより内部的に管理された，c から始まる値が格納される．ファイルパス，リビジョン番号，更新日時，更新作業者，ログメッセージ，ブランチタグ情報はリビジョン情報の値が格納される．関連識別番号は，相互に高い関連を持つ相手のリビジョンの識別番号が，データベースにより格納される．

5.2.2 電子メール情報データベースの構築

開発者間で行われたコミュニケーションの記録である電子メールのアーカイブには，膨大な情報が含まれている．これらの情報の中から抽出した情報を，電子メール情報と呼ぶ．

電子メール情報生成部は，既存の形態素解析器等を情報抽出ツールとして利用して得られる抽出情報から，電子メール情報を生成する．電子メール情報データベースは，生成部によって生成された電子メール情報を管理する．

電子メール情報生成部は，抽出情報の中から，電子メール情報として以下の 6 つを抽出する．

- 「送信者」
- 「サブジェクト」
- 「送信日時」
- 「議論対象のファイルパス」
  あるリビジョンについて議論している電子メールの中には，議論の対象となっているファイル名，あるいは，ファイルパスが記述されている．その情報を抽出して，「ファイルパス」として電子メール情報の要素とする．
- 「キーワード」

あるリビジョンについて議論している電子メールの中で頻繁に出てくるファイル名やメソッド名等の単語は，その話題の中で重要な役割を果たす．そのような単語を抽出し，「キーワード」として電子メール情報の要素とする．

- 「議論者名」
  あるリビジョンについて議論している開発者の情報として，議論に参加した開発者の名前やメールアドレスといった情報を抽出して，電子メール情報の要素とする．CVS 情報として commit した開発者の名前が記録されるが，そのリビジョンに関しての議論に関った他の開発者の名前を抽出して記録しておくことで，開発作業を支援することができる．

この中で，キーとなる要素は，送信者とサブジェクト，送信日時である．そこで，CVS 情報と同様に，これらの組合せに識別番号を対応させ，「識別番号」として電子メール情報の要素に追加する．

また，これらの電子メール情報の中に，「同じ送信者が，同じサブジェクトで，一定の日時内に送信した電子メールが複数存在する場合，それらのリビジョンは関連性が高い」という関係が成り立つ．そこで，これらの電子メールに関する電子メール情報には，互いに高い関連性を持つ電子メールを ID 番号で記録するため，「関連識別番号」として電子メール情報の要素に追加する．

例えば，ある議論に関する電子メールのアーカイブから抽出された情報から，図 5 に示す情報が本データベースに格納される．識別番号は，データベースにより内部的に管理された，m から始まる値が格納される．送信者，サブジェクト，送信日時，ファイルパス，キーワード，議論者名は抽出情報から生成された値が格納される．関連識別番号は，相互に高い関連を持つ相手のリビジョンの識別番号が，データベースにより格納される．

5.2.3 統合情報管理データベースの構築

CVS 情報データベースと電子メール情報データベースから必要な情報を取り出し，両者を有機的に結合させたものを，統合情報と呼ぶ．

統合情報生成部は，CVS 情報と電子メール情報から情報を抽出し，両者を結合させて，統合情報を生成する．統合情報データベースは，生成部によって生成された統合情報を管理する．

統合情報生成部は，CVS 情報と電子メール情報の中から，識別番号と関連識別情報以外の要素について，情報を結合する．この際，重複するか，著しく類似した情報は一つにまとめる．また，情報の無い要素は削除する．すなわち，統合情報として，最大で，以下に示す 11 個の要素を持つことになる．

識別番号	m138
送信者	t-ishikw@ics.es.osaka-u.ac.jp
サブジェクト	optimize input methods
送信日時	2001/08/19 12:37:29
ファイルパス	test/hello.cpp,v
キーワード	optimize, input, output
議論者名	t-ishikw, t-yamamt, matusita
関連識別番号	m125, m128, m134, m141

識別番号	m141
送信者	t-ishikw@ics.es.osaka-u.ac.jp
サブジェクト	Re: optimize input methods
送信日時	2001/08/20 02:32:15
ファイルパス	test/hello.cpp,v
キーワード	debug, output, input
議論者名	t-ishikw, t-yamamt
関連識別番号	m128, m134, m138

図 5: 電子メール情報の例

- ファイルパス
- リビジョン番号
- 更新日時
- 更新作業者
- ログメッージ
- ブランチタグ情報
- 送信者
- サブジェクト
- 送信日時
- キーワード
- 議論者名

CVS 情報と電子メール情報が持っていた識別番号は，結合することにより生成された統合情報の「関連識別番号」として格納される．また，本データベース内での「識別番号」を新たに要素として追加する．これにより，統合情報の関連識別番号が，結合された CVS 情報と電子メール情報の識別番号に対する連係を残すことになる．

結合すべき CVS 情報と電子メール情報の間には，高い関連性が必要である．従って，結合する際には，以下の 3 つの要素を比較する．この判断基準により，両者が高い関連性を持つかどうか判断する．

- 「ファイルパス」
  まず，CVS 情報の「ファイルパス」と電子メール情報の「ファイルパス」に着目して，両者が同じファイルパスである場合には，高い関連性を持つ可能性があると考える．
- 「時間の流れ」
  また，CVS 情報の「更新日時」と電子メール情報の「送信日時」に着目して，両者がある一定の日時内である場合には，高い関連性を持つ可能性があると考える．
- 「開発者」
  さらに，リビジョンを更新した開発者は，電子メールで行われた議論にも参加している可能性が高い．そこで，CVS 情報の「更新作業者」と電子メール情報の「議論者名」に着目して，両者に同じ開発者が存在する場合には，高い関連性を持つ可能性があると考える．

識別番号	i78
ファイルパス	test/hello.cpp,v
リビジョン番号	1.3
更新日時	2001/08/20 17:05:33
更新作業者	t-ishikw
ログメッセージ	optimized
送信者	t-ishikw@ics.es.osaka-u.ac.jp
サブジェクト	optimize input methods
送信日時	2001/08/19 12:37:29 2001/08/20 02:32:15
キーワード	optimize, input, output, debug
議論者	t-ishikw, t-yamamt, matusita
関連識別番号	c382, m138, m141

図 6: 統合情報の例

これらの 3 つの要素が一致しているとき，CVS 情報と電子メール情報は互いに高い関連性を持つとする．例えば，図 4 の識別番号 c382 の CVS 情報と，図 5 の識別番号 m138，m141 の電子メール情報は，上記の判断基準を全て満たしている．このため，両者を実際に結合して生成された統合情報の例を図 6 に示す．識別番号は，データベースにより内部的に管理された，i から始まる値が格納される．関連識別番号は，結合した CVS 情報と電子メール情報の識別番号が格納される．その他の要素は，CVS 情報と電子メール情報の持つ情報が結合され，最適化されている．

5.2.4 統合情報データベースシステムの参照方法

最後に，本データベースシステムを利用した統合情報の参照方法について述べる．本システムは，検索キーによる参照と，特定要素検索による参照が可能である．

- 検索キーによる参照
  開発者は，参照したい語句を検索キーに指定する．データベースシステムは，この検索キーを全ての統合情報が持つ要素の値と比較する．一見，効率が悪いのだが，この統合情報は，関連識別番号から辿れる CVS 情報や電子メール情報から抽出した情報の集合であるため，その統合情報から全ての情報を辿った場合と同じ結果が得られ，非常に効率が良いことがわかる．例えば，「t-yamamt が開発に関与したリビジョンを知りたい」という参照がこれに当たり，検索キーは「t-yamamt」である．
- 特定要素検索による参照
  一方で，統合情報は，各要素の値として情報を保持しているため，その要素に条件を与える形での検索が可能である．例えば，「更新日時が 3 日以内のリビジョン情報と，それに高い関連性を持つ電子メールのアーカイブを取得したい」という参照がこれに当たる．

以下では，t-yamamt が開発に関与したリビジョンを知りたい場合について，その処理の流れを示す．

- 「t-yamamt」を検索キーとする．

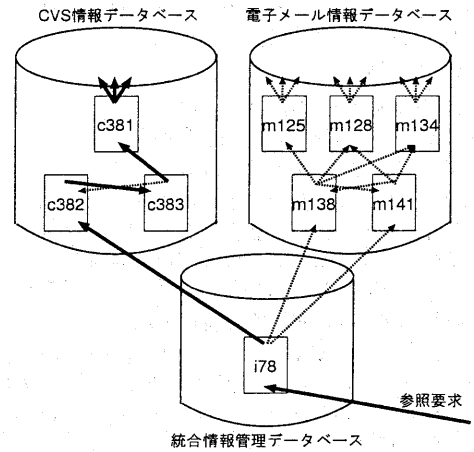

図 7: 再帰的参照の例

- まず，統合情報管理データベースに格納された統合情報の中から，「t-yamamt」という情報を持つ統合情報の識別番号を全て求めて，バッファ格納していく．
- 次に，この場合はリビジョンについて知りたいので，関連識別番号の中で CVS 情報 (頭文字が c) の c382 について，CVS 情報データベースから CVS 情報を取得する．
- 以降，再帰的に関連識別番号を辿る．(図 7)
- 最終的に，バッファに格納された識別番号が参照結果となる．

このような手順で処理が行われるため，場合により再帰的に行われる処理をある段階で打ち切る必要が生じる．そのため，再帰処理を行う深さはパラメータで指定できるものとする．

5.3 システム制御部

統合情報データベースシステムが提供する機能について，以下で説明する．

- 参照要求フォームの提供
  システム制御部は，GUI 等のインターフェイスを利用して開発者に参照要求フォームを提供し，開発者からの参照要求を受け付ける．
- 参照要求の解析
  受け付けられた要求フォームから開発者の参照要求を解析する．このとき，データベースの標準言語である SQL を用いて記述された参照クエリに変換する．これが，統合情報データベースシステムへの入力となる．
- 参照結果の解析
  データベースシステムから参照結果として統合情報が返ってくると，その結果を解析する．必要に応じて，CVS や電子メールのアーカイブ，あるいは，もう一度データベースシステムに対してクエリを送り，開発者の要求に対する参照

結果を取得する．得られた情報に整形処理等を施した後，開発者に提供する．

5.4 利用例

3.1節で述べた問題点が，本システムを用いた場合，どのように解決するのかを示す．

5.4.1 リビジョン情報の情報量不足

本システムを利用することにより，t-ishikwは，開発者名「matusita」を検索キーとしてCVS情報を参照要求することで，matusitaが開発に関与したリビジョンの情報を簡単に知ることができる．

5.4.2 ツール／機能間の連係不足

本システムを利用することにより，t-ishikwは，キーワード「input」を検索キーとしてCVS情報と電子メール情報を参照要求することで，「input」に関連したリビジョン情報と電子メールのアーカイブを同時に知ることができる．従って，電子メールのアーカイブとCVSの間での連係が補われており，既存のシステムと比較して，開発者が必要な情報を取得することが容易になる．

本システムを用いることにより，現状のOSS開発における問題点を解決し，電子メール等によるコミュニケーションを容易に行えるのではないかと考える．また，各開発者は，限られた時間を有効に利用したコミュニケーション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．

6 まとめと今後の課題

本稿では，既存の開発環境では支援が不十分な，開発者相互のコミュニケーションを支援するシステムを提案した．既存の版管理システムと電子メールのアーカイブ等が互いに分散して保持する各種の情報を，有機的に結合させた統合情報を管理するデータベースシステムの構築を行った．

今後は，システムの詳細な設計と共に，実装を進める．また，実際にシステムを利用することにより，本システムの有効性について検証を行いたいと考えている．

参考文献

[1] Asklund, U., Magnusson, B. and Persson, A., "Experiences: Distributed Development and Software Configuration Management", SCM-9, LNCS1675, pp.17–33, 1999.

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

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

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

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

[6] Fogel, K., "Open Source Development with CVS", The Coriolis Group, 2000.

[7] Hoek, A., Heimbigner, D. and Wolf, A. L., "A generic, peer-to-peer repository for distributed configuration management", Proceedings of ICSE-18, pp.298–307, Berlin, Germany, March 1996.

[8] Kilpi, T., "Product Management Requirement for SCM Discipline", SCM-7, LNCS 1235, pp.175–185, 1997.

[9] Lin, Y. J. and Reiss, S. P., "Configutarion Management with Logical Structures", Proceedings of ICSE-18, pp.298–307, Berlin, Germany, March 1996.

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

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

[12] Milewski. B., "Distributed Source Control System", SCM-7, LNCS 1235, pp98–107, 1997.

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

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

[15] Raymond, E. S., "The Cathedral & the Bazaar", O'REILLY, 1999.

[16] 寺口，松下，井上，"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.

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

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

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

[20] Ye, Y., Fischer, G. and Reeves, B., "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.