

コンテキストの推定による 開発支援システム間の情報統合

谷 宗一郎^{†1} 上野 秀剛^{†1}

バグ管理システムや、版管理システムを用いて開発に関する情報を管理しているプロジェクトは多い。しかし、それらに格納されている情報は、不具合履歴やソースコードの変更履歴といった情報の種類ごとに管理されており、開発中の話題のまとめ（コンテキスト）で管理されていない。そのため、ある一つの話題について調べる場合、複数のシステムを検索しなければならず、情報共有を難しくしている。本稿では複数のシステムにまたがった情報を統合し、同じコンテキストに属する情報の閲覧を容易にするシステムについて提案する。提案システムはプロキシとして動作することで、既に利用している外部システムを利用しながら、情報を統合できる。

Information Integration between Development Support Systems by Estimate of Context

SOUICHIRO TANI^{†1} and HIDETAKE UWANO^{†1}

Many software projects use support system such as bug tracking system or version control system to manage information on development. Such support systems collect the information based on type, not based on topics of the development, hence, a string of information which belong to same development context spread through the systems. Therefore, developer must search more than one system, so it makes information sharing difficult. This paper proposes a system to integrate the information located different support systems for efficient collection which belong to same context. The system runs as a proxy server; developers can utilize existing systems and stored information.

^{†1} 奈良工業高等専門学校
Nara National College of Technology

1. はじめに

ソフトウェア開発の外部委託やオープンソースソフトウェア (Open Source Software: OSS) プロジェクトの増加に伴い、地理的に離れた場所にいる開発者間で情報を共有する需要が増加している。外部委託においては、海外の開発組織に委託するオフショア開発が増加していることもあり¹⁾、タイムゾーンの異なる遠隔地との情報交換の重要性も高まっている。

このような環境では、移動時間や移動費用のかかるオフラインミーティングや、スケジュール調整が必要な電話会議と併用して、オンラインによる非同期な情報交換が良く用いられている^{2),3)}。ソフトウェア開発における、代表的なオンライン、非同期コミュニケーションのツールとしてメーリングリスト (Mailing List: ML) やバグ管理システム (Bug Tracking System: BTS)、版管理システム (Version Control System: VCS) がある (以下、支援システム)。

これらのシステムは、ソフトウェア開発を行っている企業においては独自に開発・運営されていることが多く、複数のシステムを組み合わせた統合システムも存在する^{4),5)}。個別のシステムを併用する場合と比べ、1つのシステムに統合することで検出された不具合と、それに関するコードの変更履歴、MLでの議論といった開発における文脈 (開発コンテキスト) の理解が容易になり、効率的に情報を収集することができる。

一方で、小規模な OSS プロジェクトではシステムの導入と維持に必要なコストの支払いが難しいことから、無料で利用可能なレンタルサービス⁶⁾⁻⁸⁾ が多く用いられている。レンタルサービスを用いる場合、個々のシステムを変更・改造するのが難しく、必要な情報を各システムから手動で調査しなければならない。複数のシステムから必要な情報を取得するためには、各開発コンテキストにおいてそれぞれの支援システムがどれだけ利用されたかを理解している必要があるが、そのような情報は支援システムには記録されていない。したがって、プロジェクトに途中から参加した開発者による収集や、古いコンテキストに関する情報の収集を行った場合、漏れが発生する可能性がある。統合システムを無料で提供しているサービスも存在するが、既に個別のシステムを利用している場合、各システムに格納されたデータの移行が難しい、プロジェクトに必要な機能が統合システムで提供されていない可能性がある、といった問題がある。

本稿では、既に利用している支援システムを変更することなく、複数のシステムに記録された一連のコンテキストに属する情報を合わせて表示するシステムについて提案を行う。提案システムは、支援システムを閲覧する際のプロキシとして存在する。開発者が閲覧しよう

とした支援システム内の情報からコンテキストを推定し、他の支援システムに格納された、同一のコンテキストに属すると思われる情報へのリンクを追加する。

以下、2章ではソフトウェア開発に用いられる支援システムの概要と問題点について説明する。3章では提案システムについて説明し、現状の実装方法とプロジェクトに適用した際の動作例を示す。4章では従来研究との違いを述べ、本研究の位置づけを明らかにする。最後に5章では、まとめと今後の課題を述べる。

2. 開発支援システム

2.1 概要

開発支援システムとは、ソースコードの変更履歴やバグ（不具合）の検出から除去までの履歴、メールやBBSでの議論など、開発に関する様々な情報を共有するためのシステムである。開発システムの多くはサーバ上でCGIやサーバソフトウェアとして動作し、Webブラウザなどを經由して利用される。

支援システムは共有したい情報の種類ごとに異なるシステムが開発されている。BTSは、バグを検出してからバグ位置の特定、修正方針の決定、除去までの履歴を管理する。また、これらの作業に伴って行われる議論の履歴も含まれる。VCSは、ソースコードや仕様書・マニュアルといったドキュメントの変更履歴を管理している。MLでは、調査内容の報告やスケジュール調整といったプロジェクト内の情報交換のログが蓄積される。ソフトウェアテストの結果報告なども、MLで行われる場合がある。

支援システムの利用者は、これらのシステムをそれぞれ参照し、開発やサポートに必要な情報を収集することができる。利便性向上のために、複数の支援システムを統合したシステムも存在する^{4),5),9)}。

ソフトウェア開発を行っている企業では、独自に支援システムを開発し、社内ネットワークで運用・管理が行われている。一方で、小規模なOSSプロジェクトでは開発のための予算を持たず、サーバの用意・運用にかかる費用を負担できないことも多い。また、OSSプロジェクトに参加する開発者はボランティアであることが一般的であり、参加者は流動的に変化するため、サーバを適切に管理できる人員を安定的に確保できないことがある。そのため、Google Code⁶⁾ や Source Forge⁷⁾ , Launchpad⁸⁾ のような無償のレンタルサービスが利用されている。

2.2 問題点

個別の支援システムや統合されたシステムを用いることで、開発者間の情報共有が容易に

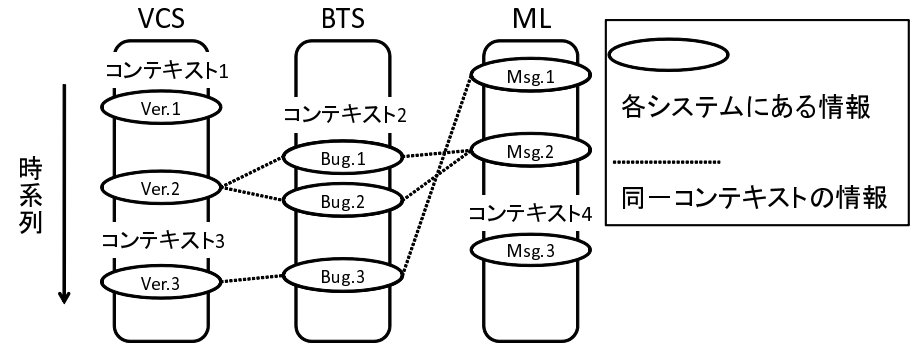


図1 各システムが持つ開発情報のつながり
Fig. 1 Relation of development information stored in each system.

なるが、以下の問題がある。

- 複数の支援システムにまたがって検索をしなければいけない
- 開発コンテキストが支援システムに記録されない
- 個別のシステムから統合システムへのデータ移行が困難
- 外部システムと他のシステムとの結合が困難
- 使用できる支援システムに制限がある

これらの問題点について、1) 複数の支援システムを併用した際の問題、2) 統合システムを用いた際の問題の2つの観点から説明する。

2.2.1 複数の支援システムの併用による問題

複数の支援システムを用いて、1つのプロジェクトに関する情報を管理している場合、開発者はある1つの事柄を調査するために、手で複数のシステムから情報を収集する必要がある。このとき、調査している事柄（例えば、ある不具合がどのように議論・修正されたか）を示す開発コンテキストを理解する必要がある。

図1に複数の支援システムを用いた開発におけるコンテキストの様子を示す。3つの長方形はVCS・BTS・MLの各支援システム、楕円はそれぞれのシステムに保存された情報、楕円を繋ぐ点線は同一コンテキストに属する情報のつながりを表している。それぞれのコンテキストに含まれる情報は、その種類ごとに異なるシステムに分散して保存されている。ある情報に関係する情報を探したいとき、点線で示されるつながりをたどることになるが、これをたどる情報は支援システムに存在せず、開発者の知識に頼っているのが現状である。

このような状況において、開発者が調査を行う際には以下のような問題点がある。

問題 1 複数のシステムにまたがった検索

複数の支援システムを併用している場合、1つの開発コンテキストに関する情報を収集するために、開発者が手動で複数のシステムを調査する必要がある。このとき、各システムがもつ、同じ不具合に関する情報は互いにリンクされておらず、必要な情報を見落としてしまう可能性がある。BTS や VCS に記録される不具合情報・履歴情報には、内容の理解や検索に用いるためのコメントや不具合 ID などが入力されるが、システム間が統合されていない場合、それらを元に手動で検索を行う必要がある。

問題 2 開発コンテキストの不足

複数の支援システムから、必要な一連の情報を探すためには、開発時のコンテキストを理解していなければならない。例えば BTS に記録されたある不具合の情報を収集する時、不具合についてどのような議論や修正が行われてきたのかを知らなければ VCS や ML の検索が難しい。このような開発コンテキストは支援システムには保存されておらず、新規にプロジェクトに参加した開発者はコンテキストの把握ができない。また、以前から参加している開発者であっても、過去のコンテキストを完全に覚えているわけではなく、記憶に頼っているのが現状である。

2.2.2 統合システムを用いる際の問題

前節で述べた問題を解消するために、複数のシステムを統合したシステムが提案・実装されている^{(4),(5),(9)}。これらの統合システムでは BTS や VCS, ML の他に Wiki などを統合することで、プロジェクトの管理に必要な情報を 1つのシステム内で管理できる。従来の統合システムには、既存の支援システムを拡張したものと、統合システムとしてあらかじめ用意されたシステムを利用するものがある。しかし、これらの統合システムをプロジェクトに導入するためには以下の問題点がある。

問題 3 既存支援システムのデータ利用が困難

プロジェクトが既に支援システムを導入している場合、これまでに支援システムに蓄積された情報を新規に導入する統合システムで使用することが難しい。一部の統合システムでは他の支援システムからのデータインポートをサポートしているが、全ての情報が適切に変換されるわけではない。

問題 4 外部システムを統合できない

既に運用している支援システムが存在する場合、システムを拡張することで蓄積さ

れた情報をそのまま統合システムを利用することができる。しかし、レンタルサービスの外部で提供された支援システムを用いている場合、システムの変更ができないため、統合することができない。

問題 5 使用できる支援システムに制限がある

個別の支援システムを用いる場合、それぞれの情報の種類に応じて、プロジェクトに必要な機能を持ったシステムを選択する事ができる。しかし、統合システムを用いる場合、各統合システムが提供する機能しか利用できず、個別の支援システムを組み合わせた場合に比べて制限がある。

3. 提案システム

3.1 概要

本稿では、2.2 節で述べた問題点に対応するために、既に利用している外部の支援システムを変更することなく、開発情報を統合して表示するシステムを提案する。提案システムの構成及び動作例を図 2 に示す。なお、図中では説明を容易にするために、開発者がアクセスする支援システムとして BTS を想定している。本システムは proxy サーバとして動作し、開発者の支援システムに対するアクセスを検出すると自動的に動作する。なお、支援システム以外へのアクセスでは、本システムは基本的なプロキシの動作以外何も行わない。この処理はアクセス判定部で行われる(図 2(1))。

支援システムへのアクセスを検知すると、本システムはその参照結果を処理にかける(2)。まず、参照結果から推定に利用する情報を抽出収集する。これはテキスト解析によって行われる。収集する情報は、BTS では投稿ごとに振られる固有番号、投稿者名、投稿時刻、投稿されたバグに関する文章など、VCS ではリビジョン番号、変更者名、コミットメッセージ、コミットファイル名などである。収集した情報は、システム内部のキャッシュデータベースに保存される。

情報収集の後、収集した情報に対して同一コンテキスト情報の推定が行われる(3)。収集した情報は、キャッシュデータベースに保存された他システムの情報一つ一つと関連性が推定される。この際の推定方法は特に提案しない。現在の実装では、単語の出現頻度で推定を行っている(3.3 で詳述)。推定精度を高める方法については今後の検討課題である。

推定処理の後、その結果はそれぞれの支援システムに合わせて整形され、情報収集の対象となった支援システムの参照結果の中に組み入れ、支援システムを閲覧しようとした開発者に提示される(4)。

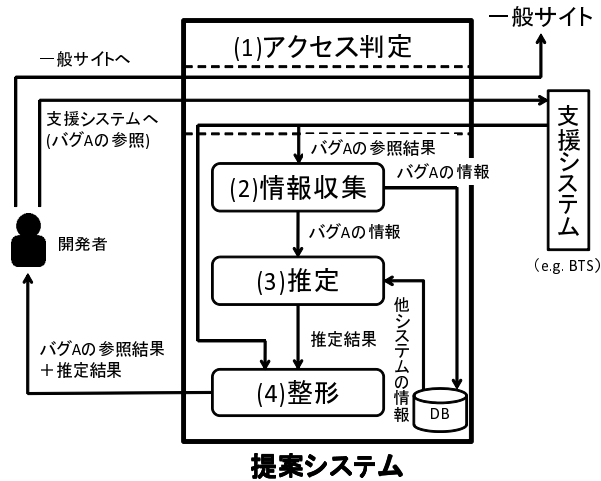


図2 提案システムの構成と動作例

Fig. 2 Structure and workflow of a proposed system.

このように、本システムの一連の動作は支援システムへのアクセスをトリガとして開始され、支援システムの出力を改変し、開発者に渡して終わる。このときの開発者（支援システムの閲覧に利用したブラウザ）と提案システム、各支援システムの間を、図3に示す。本システムは、それぞれの支援システムのホストとは独立し、開発者の支援システムへのアクセスに従って情報を自動的に取得するため、開発者は本システムの存在を意識することなく、他のシステムに保存された関連情報を見ることができる。

3.2 問題点との対応

提案システムを用いて、単体で動作している支援システムにアクセスすると、自動的に他の支援システムに保存されている関連の情報を抽出し、表示する。これによって、問題1（複数のシステムにまたがった検索）が解決される。このとき、複数のシステムに保存された情報間の関連性を推定し、開発コンテキストを知らない開発者に一連の情報を示す事で、問題2（開発コンテキストの不足）に対応している。

問題3（既存支援システムのデータ利用が困難）及び問題4（外部システムを統合できない）については、既存のシステムがあらかじめ持っているインタフェース（例えばHTTP）を介して通信を行い、システムを変更することなく、蓄積された情報をそのまま利用するこ

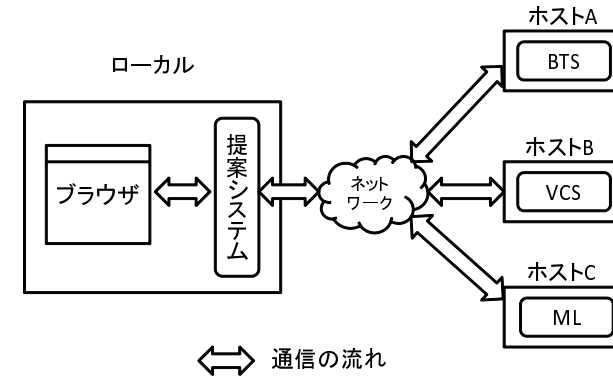


図3 システムの実装のハードウェア構成

Fig. 3 Hardware structure of the system implementation.

とで対応している。また、上記の点に加え、支援システムとの通信モジュールの新規実装を容易にするために、インタフェースクラスを用意している。プロジェクトで利用したい支援システムと提案システムの結合を容易にすることで、問題5（使用できる支援システムに制限がある）に対応している。

3.3 実装

提案システムが適切に動作するか確認するために、実装を行った。現在、man-atubbs(BTS)¹⁰⁾とGoogle Code⁶⁾(VCS)に対応したシステムの試作を行っている。本実装は、C#で記述されており、.NET Framework上で動作する。ソースコードは約2000行、約8個のクラスで構成される。本実装では各支援システムへの対応（情報の抽出、推定を行うための共通フォーマットへの整形、提示情報の整形など）を、共通処理を除き外部ライブラリとして実装しているため、他の支援システムへの対応も容易に行うことができる。

現在の実装では、推定は以下の手順で行っている。

- (1) 本システムを利用するプロジェクトで話題のキーとなる単語や特徴的な単語を設定する。3.4節に示す動作例では、特徴的な単語としてBTSのサマリにおける頻出単語61個を利用した。
- (2) 現在閲覧中の情報と、本システム内にキャッシュされたすべての情報について、設定した単語の出現回数をカウントする。
- (3) それぞれの情報について、各単語の出現回数を、その情報での単語の総出現回数で割

@510 (#1615) 日付選択にて初期値を設定できるようにしてほしい - (2010-05-22 03:07)

【症状】日付選択(カレンダーダイアログによる入力)では、必ず初期値が今日になる。
【再現方法】
【要望】日付選択において、初期値設定ができるようにしたい。(入力後の再修正や、既にある日付を修正する用途でりようしたい)
【バージョン】確認したバージョン 1.5328

システムの出力

このページの単語(Windows7 2,マニュアル 1,命令 1,設定 6,ダイアログ 4)

ManatsuBBS: 465(0.287,設定 3,リファレンス 1,サンプル 3,部品 1)
81(0.296,ワード 1,命令 1,設定 1)
155(0.305,HTTP 2,設定 3,サンプル 1)
447(0.308,母艦 2,TEデータ 5,設定 22,ファイル 7,パネル 3,タブ 1,自身 1)
238(0.313,ボタン 2,設定 2,サンプル 1)

GoogleCodeSVN: 61(0.193,文字列 1,命令 1,設定 3)
234(0.201,Windows7 1,命令 1,設定 1)
126(0.296,命令 1,設定 1,部品 1)
170(0.296,マニュアル 1,設定 1)
98(0.332,DLL 1,プラグイン 1,設定 1,dnako 1)

↑ (#1619) r234で修正 - (2010-05-24 01:31) 中 盛 源 待 ち

報告ありがとうございました。
下記の通り修正しました。
- 「日付選択」命令で初期値を設定できない不具合とWindows7カレンダーが欠ける問題を修正(r234) (@510)。

図 4 提案システムを介したバグ管理システムの出力

Fig. 4 Screenshot of bug tracking system through the proposal system.

り、単語の出現割合とする。

- (4) 現在閲覧中の情報と、本システム内にキャッシュされたすべての情報それぞれの間で、単語ごとに出現割合の差の二乗を取り、各情報の組で総和を求める。この値を現在閲覧中の情報と、本システム内にキャッシュされた情報のひとつとの推定スコアとする。
- (5) 推定スコアが小さい組のシステム内の情報ほど、現在閲覧中の情報と同一コンテキストに属している可能性が高いと見なす。

現在の実装では、同一コンテキストに即している可能性が高い情報のうち上位 5 件を提示している。

Project Home Downloads Source

Checkout | Browse | Changes | Search Trunk

Revision: r234 <r233 r234 r235>

Author: [hshirano](#)
Date: May 23, 2010
Review scores: No one has yet scored this revision.

Log message

- 「日付選択」命令で初期値を設定できない不具合とWindows7カレンダーが欠ける問題を修正(r234) (@510)。

システムの出力

このページの単語(Windows7 1,命令 1,設定 1)

ManatsuBBS: 81(0.222,ワード 1,命令 1,設定 1)
329(0.333,マニュアル 1,DLL 2,命令 1,Delphi 1,設定 1)
201(0.375,命令 2,配列 1,引数 1)
365(0.375,DLL 1,命令 2,ネット 1)
465(0.396,設定 3,リファレンス 1,サンプル 3,部品 1)
GoogleCodeSVN: 234(0,Windows7 1,命令 1,設定 1)
126(0.222,命令 1,設定 1,部品 1)
61(0.24,文字列 1,命令 1,設定 3)
48(0.347,FTP 2,命令 1,設定 1,ファイル 1)
173(0.375,なでしこTEデータ 1,命令 2,タブ 1)

Affected files expand all collapse all

- Ⓜ Modify /trunk/History.txt diff
- Ⓜ Modify /trunk/vnako_unit/frmCalendarU.dfm diff
- Ⓜ Modify /trunk/vnako_unit/frmCalendarU.pas diff

図 5 提案システムを介した版管理システムの出力

Fig. 5 Screenshot of version control system through the proposal system.

図 4 と図 5 に、本システムを介して manatubbs および Google Code にアクセスした時の出力例を示す。図の中央付近「システムの出力」の下(点線内)に、本システムが出力した結果が表示されている。次の節で提案システムの動作例について説明する。

3.4 動作例

図 4 と図 5 は日本語プログラミング言語「なでしこ」の開発プロジェクト*1で実際に使われている manatubbs(BTS) と Google Code(VCS) に、提案システムを介してアクセスした

*1 <http://nadesi.com/>

様子を示している。出力の一行目は現在表示しているページに含まれる特徴的な単語 (Windows7, マニュアル, 命令, 設定, ダイアログ) と, その出現回数を示している。2 行目以降には, それぞれの支援システムについて, このページで得られた単語から推定されたページへのリンクが 5 つずつ提示されている。ここでの表記のフォーマットは「ページ番号 (スコア, 各単語の数...)」となっている。スコアは推定によって求めた推定スコアであり, この値が 0 に近いほど, 現在のページと同一のコンテキストに含まれる可能性が高い。各単語の数の表記は, 一行目と同一である。なお, この例では, 話題の特徴を表す単語はなでしこプロジェクトに参加している著者の経験に基づいて主観的に決定した。

図 4 に示した BTS のページには, なでしこに含まれる日付選択用のダイアログボックスについての要望と対応結果 (ID:@510) が記述されている。また, 図 5 には同じ要望に対する変更履歴 (ID:r234) が記録されている。このプロジェクトでは, 開発者が各システムへの入力を行う際に, 関連する情報の ID を手動で入力することで, 他のシステムとの関連性を管理している (図 4 下部, および図 5 上部)。この動作例の場合, 開発者は変更履歴 r234 と @510 が関連しているとして, BTS, VCS それぞれにコメントで記録している。これらの履歴に対して, 提案システムの BTS への出力に r234 へのリンクが, VCS への出力に @510 へのリンクが含まれており, 適切なリンクを出力できていることを示している。

4. 関連研究

関連研究として先に挙げた二つのシステムについて述べる。

Trac⁴⁾ は版管理システムと連動するバグ管理システムである。Trac の利用者はバグ管理システムに保存された不具合に割り当てられた ID を, 版管理システムへの投稿に含めることで, 関連づけを行う。しかし, このシステムでは関連づけを開発者の手動で行っており, 開発者が VCS の情報と BTS の関連性を把握していなければならない。また, 既存プロジェクトで他の BTS を利用している場合, 既存のシステムに登録された情報を Trac に移す必要があるが, 手動で再投稿を行う場合, 多くの労力が必要な上, 開発コンテキストの理解に重要な情報である投稿日時が失われてしまう。既存システムのデータベースを Trac で用いられているデータベースに変換する事もできるが, 両者のデータベース構造に関する知識が必要となり, 容易に情報を移行できるものではない。

石川らは ML と VCS を組み合わせた検索システムについて提案している⁹⁾。このシステムはそれぞれの支援システムから情報を取得し, 投稿者や投稿時間, メッセージから抽出したキーワードなどの要素から情報を関連づけている。既存のシステムをそのまま取り込む方

表 1 既存の統合システムと提案システムの比較
Table 1 Comparison between the proposal system and other integration systems.

	Trac ⁴⁾	石川ら ⁹⁾	大平ら ⁵⁾	提案システム
複数種類の情報の格納				
既存システムの活用	×			
ホストマシン	必要	必要	必要	不要
関連情報の結合	手動	自動	プロジェクト単位で集計	自動
任意のシステムの追加	×	×		

式のため, 移行の問題は解決されるが, ホストとなるマシン上にシステムを構築するため, マシンが用意できない場合や, レンタルサービスを利用している場合など, 適用できないケースが存在する。

大平らは複数の支援システムの情報を収集し, プロジェクトのリアルタイムな管理を行うシステム EPM(Empirical Project Monitor) を提案している⁵⁾。EPM は CVS や Mailman, GNATS など 6 つの支援システムに対応しており, 未対応のシステムに対しても Ruby のスクリプトを作成することで比較的容易に対応することができる。プロジェクトで既利用している支援システムをそのまま利用でき, 提案システムと類似している。

しかし, EPM はプロジェクトの定量的な分析を目的に設計されている。そのため, ソースコードの規模推移や, 不具合数の推移などを容易に把握できるが, 個々の不具合に対する議論や修正作業のようなコンテキストの把握には適していない。また, 石川らのシステムと同様にホストとなるマシンを用意しなければならず, 外部システムの結合も対応していない。

これらのシステムと, 本稿で提案したシステムの比較を表 1 に示す。Trac は種類の異なる情報をひとつのシステムで扱うということができるが, その中で関連する情報を自動で結合させることはできず, 既存システムの情報も活用できない。石川らや大平らは関連情報の結合や既存システムの活用が可能であるが, その運用にホストとなるマシンを必要とする。また, 石川らは任意のシステムを追加することができず, 大平らは任意のシステムを追加することはできるが, 外部システムの追加ができないという制限がある。本稿の提案システムはローカル環境で proxy サーバとして動作するためホストとなるマシンを必要とせず, 任意のシステムについて, 既に蓄積された情報を活用することができる。さらに, 情報間の関連性の推定を行うことで, 関連情報の結合にも対応している。

5. おわりに

本稿では, 複数の支援システム間にまたがった検索における, 同一コンテキストの情報に

スムーズにアクセスできないという問題を解決するために、支援システムを変更することなく統合するシステムについて提案を行った。提案システムはプロキシして動作し、その通信の過程で得た情報により開発コンテキストを推定することで、対象とする支援システムを選ばず、スムーズに同一コンテキストの情報へのアクセスができる。また、提案システムが適切に動作するかを確認するために、実際の OSS プロジェクトで用いられている 2 つの支援システムに提案システムを適用し、動作の確認を行った。

本稿では、2 つの支援システムを提案システムで結合し、開発者が手動で入力した関連情報をシステムが出力していることが確認できたが、他の情報については確認を行っていない。今後、複数の支援システムを利用している開発者にシステムを使用してもらい、操作履歴やアンケートから開発におけるコンテキストを正しく推定できているか確認する。情報間の関連について推定する方法についても、精度を高めるための方法を検討する。現状の推定方法は簡易なものであるが、今後、TF-IDF 方を用いた推定手法の適用を検討している。

また、開発のコンテキストを示す重要な要素である、日時についても推定に用いることで精度が高まると考えられる。1 つのコンテキストに属する一連の情報であっても、時間の経過と共に主題が変化していくことがある。例えばあるバグに関する議論で、最初はバグの再現性や挙動についての議論をしていても、議論が進むにつれて、不具合位置の特定や修正方法についての議論に話題が遷移する事がある。このとき、話題の主題は同じバグであっても、再現性と修正方法という異なる内容が含まれるため、特徴付けるキーワードが変化すると考えられる。このような、時系列の変化をとらえることは開発のコンテキストを理解するためには重要である。今後の研究では、一連の話題から短期間のイベントを抽出して話題推移の把握を支援する手法¹¹⁾をシステムに導入することで、より精度の高い推定が行えると考えられる。

参 考 文 献

- 1) 情報処理推進機構: IT 人材白書 2010, <http://www.ipa.go.jp/about/press/20100407.html> (2010).
- 2) Bikram Sengupta, Satish Chandra, Vibha Sinha: A research agenda for distributed software development, In proc. the 28th international conference on Software engineering (ICSE), pp.731-740 (2006).
- 3) Carl Gutwin, Reagan Penner, Kevin Schneider: Group awareness in distributed software development, In proc. the 2004 ACM conference on Computer supported cooperative work, pp.72-81 (2004).

- 4) Edgwall Software: The Trac Project!, <http://trac.edgwall.org/>, (2010).
- 5) 大平雅雄, 横森励士, 阪井 誠, 岩村 聡, 小野英治, 新海 平, 横川智教: ソフトウェア開発プロジェクトのリアルタイム管理を目的とした支援システム, 電子情報通信学会論文誌 D-I, Vol.J88-D-I, No.2, pp.228-239 (2005).
- 6) Google Project Hosting: Project Hosting on Google Code, <http://code.google.com/hosting/>, (2010).
- 7) Geeknet, Inc.: SourceForge.net: Find and Develop Open Source Software, <http://sourceforge.net/>, (2010).
- 8) Canonical Ltd.: Launchpad, <https://launchpad.net/>, (2010).
- 9) 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).
- 10) 酒徳峰章: manatubbs - くじらはんどラボ, <http://d.aoikujira.com/labo/index.php?manatubbs>, (2010).
- 11) 菊池匡晃, 岡本昌之, 山崎智弘: 階層型クラスタリングを用いた時系列テキスト集合からの話題推移抽出, 日本データベース学会論文誌, Vol.7, No.1, pp.85-90 (2008).