

SES'13 WS ソーシャルソフトウェア工学とその支援ツール (2013/09/11)

担当者: 大森 隆行 (takayuki@fse.cs.ritsumei.ac.jp)

Using Information Fragments to Answer the Questions Developers Ask

Thomas Fritz, Gail C. Murphy

概要

開発者が普段出くわすソフトウェア開発に関する「質問(question)」には、答えるのが難しいものがある。

例えば、解釈が一通りでないものや、複数種類の情報を考慮しないと
いけないものなど。

既存手法では、複数のビューを開いて、それらを見比べる必要がある。
クエリ言語を使う場合、学習コストが高い。

本論文では、ソフトウェア開発に関する「質問」への答えを助ける手法を
提案。複数種類の情報を統合した **information fragment** を生成。

貢献

- 78個の質問
- Information fragment modelの導入
ソフトウェア開発に関する情報をグラフとして表現。
- ケーススタディ
→18名の開発者が提案モデルを容易に使うことができた。

(1) 78個の質問

■ 実験方法

■ 事前準備

- パイロットとして、開発中に必要となる情報を聞いておく
- あらかじめツールのプロトタイプを見せておく

■ 11名の開発者にオープンインタビュー

- 15~60分の面接
- 78個の「質問」を識別
- 「質問」は8つのドメイン(=情報の種類)に属する
- 大半の「質問」は1人の開発者が述べたもの

8つのドメイン
 SC: ソースコード
 CHS: 変更セット(change sets)
 T: 開発メンバー
 WI: ワークアイテム(e.g. issue)
 WW: ウェブサイト、Wiki
 CO: WIのコメント、メール
 ST: 例外スタックトレース
 TC: テストケース

Tab.1

Question	Operator	Source Code	Change Sets	Teams	Work Items	Comments	Web/Wiki	Stack Traces	Test Cases
<i>Who is working on what (people specific)</i>									
1. Who is working on what?*	id	X	X	X	X				
2. What are they [coworkers] working on right now?*	id	X	X	X	X				
3. How much work [have] people done?*	id	X	X	X	X				
4. Who changed this [code], focused on person?*	id	X	X	X	X				
5. Who to assign a code review to? / Who has the knowledge to do the code review?*	id	X	X	X	X				
6. What have people been working on?*	id				X	X			
7. Which code reviews have been assigned to which person?*	id			X	X				

→ ドメイン

似たような「質問」でも、関連するドメインが異なるものがある。

* フルセットは <http://www.cs.ubc.ca/labs/spl/allQuestions.pdf>

(2) Information fragment model

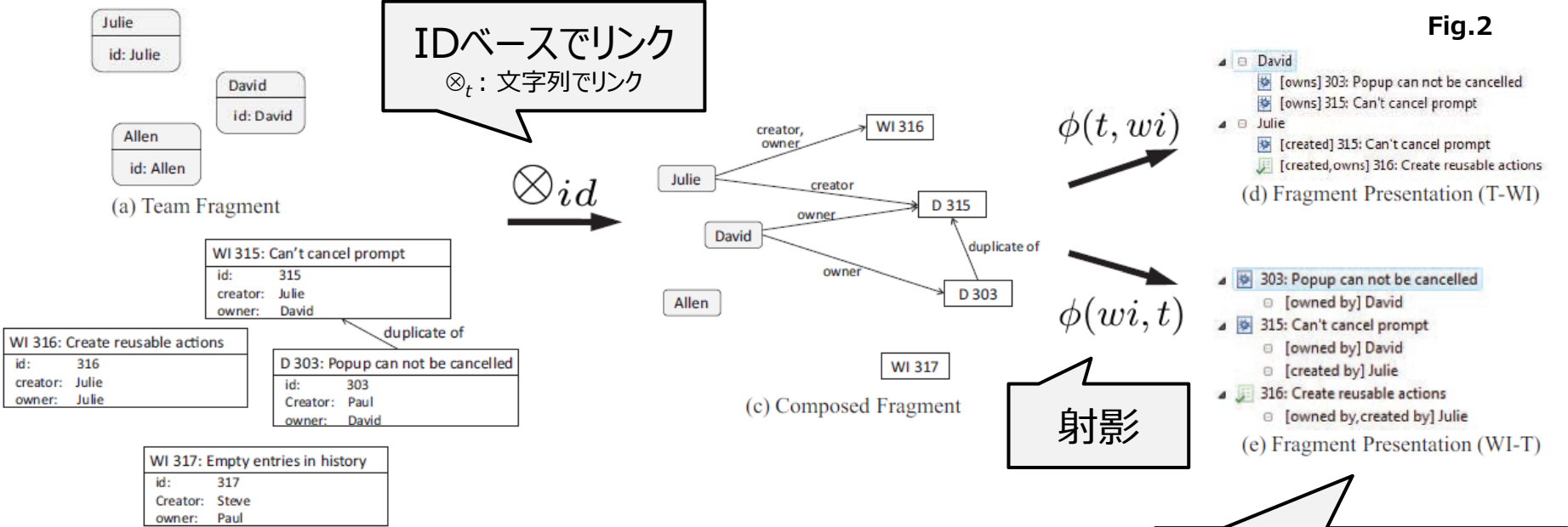


Fig.2

IDベースでリンク
⊗_t: 文字列でリンク

⊗_{id}

射影

大元の情報断片

各ノードは、属性として、他の情報への参照を保持

- 8つのドメイン(再掲)
- SC: ソースコード
- CHS: 変更セット(change sets)
- T: 開発メンバー
- WI: ワークアイテム(e.g. issue)
- WW: ウェブサイト、Wiki
- CO: WIのコメント、メール
- ST: 例外スタックトレース
- TC: テストケース

射影に基づく
ツリー形式で情報表示

(3) ケーススタディ

Question (orig. dev. question)	Time (in minutes) and Success per Question for each Developer																			Mean
	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>S4</i>	<i>S5</i>	<i>S6</i>	<i>S7</i>	<i>S8</i>	<i>S9</i>	<i>S10</i>	<i>T1</i>	<i>T2</i>	<i>T3</i>	<i>T4</i>	<i>T5</i>	<i>T6</i>	<i>T7</i>	<i>T8</i>		
Q1 (21)	1.1✓	0.7✓	2.2✓	1.8✓	1.9✓	1.1✓	3.2✓	1.4✓	1.1✓	0.6✓	1.1✓	0.6✓	1.5✓	1.9✓	0.6✓	2.6✓	5.7✓	1.1✓	1.7	
Q2 (15)	2.9✓	0.6✓	1.0✓	2.1✓	1.2✓	1.7✓	2.1✓	0.7✓	1.3✓	0.8✓	5.1✓	0.7✓	1.6✓	2.5✓	0.5✓	2.5✓	3.9✓	4.9✓	2.0	
Q3 (14)	2.0✓	1.5✓	2.2✓	2.1✓	1.3✓	5.9✓	1.0✓	3.0✓	2.3✓	1.5✓	0.9✓	1.7✓	4.0✓	1.9✓	1.0✓	1.5✓	6.6✓	5.4✓	2.5	
Q4 (6)	1.1✓	0.8✓	0.6✓	0.4✓	0.8✓	0.7✓	0.8✓	1.0✓	1.3✓	1.0✓	0.8✓	0.8✓	1.7✓	2.1✓	1.0✓	1.7✓	1.3✓	0.4✓	1.0	
Q5 (11)	1.5✓	5.6✓	3.4✓	4.2*	6.1✓	3.3✓	5.8✓	1.8✓	5.0✓	1.7✓	7.2✓	6.8*	7.2✓	5.5✓	2.8✓	7.0✓	6.5*	1.4✓	4.4	
Q6 (23)	1.0✓	0.8✓	0.9✓	0.7✓	0.7✓	1.8✓	3.0✓	0.6✓	5.0✓	1.2✓	1.7✓	0.6✓	2.9✓	1.6✓	0.8✓	1.5✓	1.6✓	3.5✓	1.7	
Q7 (19)	2.7✓	2.6✓	3.2✓	2.6✓	4.9✓	3.4✓	1.3✓	1.5✓	2.2✓	2.0✓	2.3✓	3.5✓	2.9✓	2.5✓	1.8✓	5.2✓	1.9✓	1.6✓	2.7	
Q8 (26)	1.7✓	2.1✓	4.3✓	3.1✓	- ■	5.3✓	8.1*	1.2✓	3.3✓	2.0✓	6.5*	- ■	5.4✓	5.3✓	2.6✓	7.8*	- ■	2.7✓	3.3	
Mean	1.8	1.8	2.2	1.8	2.4	2.9	2.5	1.4	2.7	1.3	2.7	1.3	3.4	2.9	1.4	3.1	3.5	2.6	2.3	

Tab.3

実験

- 8つの「質問」を選択 (全ドメインをカバーできていない)
- 被験者(18名のプロ開発者)には予め10~15分のチュートリアルを

結果

- 135/144 (94%) はヒントなしで正しく解答 平均2.3分
- 23/144 (16%) は5分以上要した
- 9/144 (6%) はヒントを与えた

SES'13 WS ソーシャルソフトウェア工学とその支援ツール (2013/09/11)

担当者: 大森 隆行 (takayuki@fse.cs.ritsumei.ac.jp)

Peer Interaction Effectively, yet Infrequently, Enables Programmers to Discover New Tools

Emerson Murphy-Hill, Gail C. Murphy

概要

- 開発者にとって有用なツールを見つけるのは困難。
本研究では、social solutionsに着目して、インタビュー調査；
主にペアプログラミングにおける開発者間のツールの知識の伝達に着目。
- どの発見方法(discovery mode) (e.g., ペアプロで助言、Twitter経由等)が一番多いか？ どの発見方法が最も有効か？

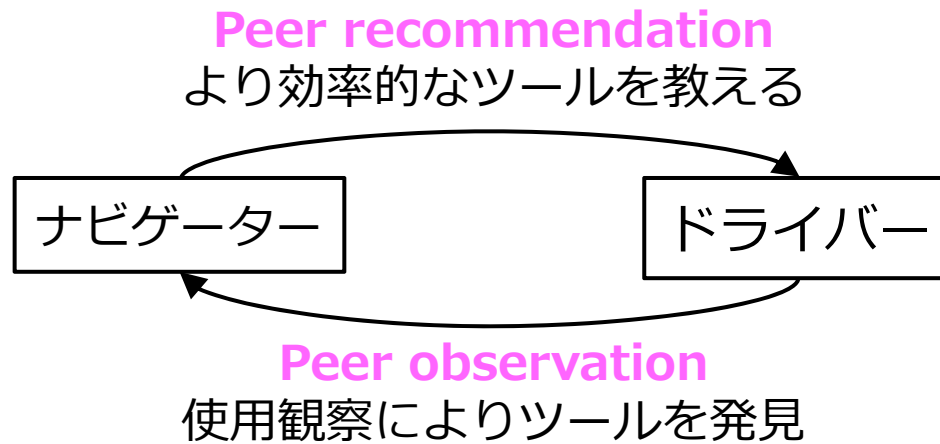
貢献

ピアインタラクション：開発において他の開発者を通してツールを発見するというツール発見方法。

peer observation, peer recommendationがある。

- ピアインタラクションの特徴。
- ピアインタラクションが、あまり出現しないものの、最もツールの学習に効率が良いであろうことを示している点。
- 将来的にピアインタラクションの頻度を高めるための考察。

ピアインタラクション



実験

- 被験者は18名の開発者 (16名はOpen Source Bridge、2名は企業から)
- 1対1のインタビュー
- semi-structured あらかじめ何を聞くか決めてある
- 電話 or インスタントメッセージ
- 一人あたり約1時間

実験結果

■ ピアインタラクションの発生状況

■ peer observation

- ペアプログラミング、リモートペアプログラミング、偶然のやり取り、変更の通知

■ peer recommendation

- ペアプログラミング、偶然のやり取り、助け舟、リモートで助け舟、Eメール

■ ピアインタラクションの頻度

- peer observation/recommendationは、tool encounter(IDEのメニューから見つける)よりは少ない (Fig.2)

■ ピアインタラクションの効率

- peer observation/recommendationは、最もツールの学習(実際にそのツールを使うかという点で)に関して効率的だと考えられている (Fig.3)

- Twitter/RSSの長所
フォローしている人の意見は信頼できる
多くの意見を一度に集められる
- Twitter/RSSの短所
関係ないツイートが多い
一番人気のあるツールを推薦する(有用なものでなく)
情報ソースが信用できない
まるで広告のようなツイートもある

→各々の詳細はTab.3

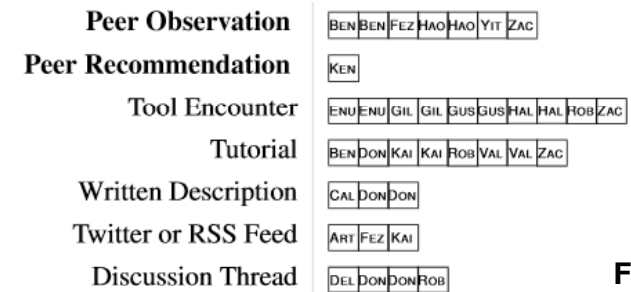


Fig.2

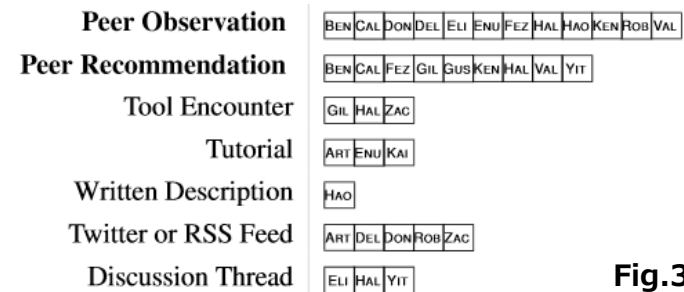


Fig.3

実験結果(続)

- **ピアインタラクションの障壁**
 - 物理的に離れている
 - ペアの人異なる開発環境を使用している
 - いったんペアの人の開発環境を見て慣れば、その後新たに学習する機会がない
 - 会社のポリシー
 - 納期に追われている
 - 開発者自身が知識の共有を喜ばない
 - まだ本人が十分に習熟していないため、有用性に自信が持てない / 単に新しいツールに興味がない / 既存ツールでうまくいっている
- **ピアインタラクションの流れ**
 - 熟練者・指導者から新人・被指導者へ。逆もある。

既存ツール改善のための考察

- **ツールの使用やその効果が目に見えるようにする**
 - ホットキーの使用は目に見えない / EclipseのOrganize Importsは画面外が変化
- **遠隔ペアプログラミング環境の整備**
 - 特に目に見える効果が大事、便利で正確なコミュニケーション手段が必要

SES'13 WS ソーシャルソフトウェア工学とその支援ツール (2013/09/11)

担当者: 大森 隆行 (takayuki@fse.cs.ritsumei.ac.jp)

Blogging Developer Knowledge: Motivations, Challenges, and Future Directions

Chris Parnin, Christoph Treude,
Margaret-Anne Storey

概要

- ソフトウェア開発に関する記事をブログに書くことの利点、問題、必要となるツールは？
- 本論文では、ソフトウェア開発ブログ投稿(blogging)のモチベーション、問題を理解するための研究アジェンダを提供。
- 開発者へのインタビュー、ブログ記事のマイニング、分析を行った。

背景

- ソーシャルメディア技術(ブログ、Wiki、Stack Overflow等)により、既存の開発者の貢献と知識共有の方法が変化[1]。
- jQueryを利用する開発者のブログで88%のAPIがカバーされている[2]。
- Android APIに関するSO上の議論を分析した結果、公式ドキュメントよりも説明とコード例が豊富[3]。
- Eclipse, GNOME, PostgreSQL, Pythonに関するブロガーの分析：1.8%のブログ記事(blog post)がコードを含む[4]。

実験

- IDEプラグイン開発 (Eclipse, Visual Studio)、モバイル開発 (Android, iPhone)、Web開発 (Django, jQuery) の分野に絞って調査。
- 各分野に関して、3つの検索キーワード (3*6) を生成。
 - e.g., "jquery div id", "jquery ajax", "jquery json"
- Google Blog Searchで検索。上位3~4件を選択。55のブログサイト (6726記事、32188コメント) を取得。
- 300記事をランダムに選択して調査。
- ブログ上での インタラクション、社会的構造 を調査するため、各ブログ著者に関して、コメントの頻繁な記事、頻繁でない記事を選択。全部で93記事の435コメントを分析。
- 55ブログの著者に直接連絡を取って調査。30人返答があった。

RQs & 知見

■ ソフトウェア開発者はブログで何をしているのか？

- ブログ記事はコードに関連しているが、多くの観点(ドキュメンテーション、進捗、技術議論等)を含む

■ ソフトウェア開発者はなぜブログをするのか？

- 個人ブランディング、evangelism and recruitment (自社製品に関する教育など)、個人的な覚え書き(knowledge repository)、フィードバック依頼

■ ソフトウェア開発ブログはどの程度インタラクティブか？

- 52%の記事にコメントあり。10/32は解決していない質問あり。
- 多くが質問と感謝。コメント欄で技術的議論になることも。

■ ブログをする開発者にとっての問題は何か？

- ブログの管理とモチベーション維持。気がついたらスパムだらけ。
- 読者対応、コミュニティへの貢献。質問が多すぎる。忘れた頃に質問。
- ブログは時間がかかる。コードを貼ったり、例を作ったり。
- 企業だと職場で書けないことも → 有用性を制限。

