Vol.2013-SE-181 No.5
2013/7/17

ソフトウェアリポジトリマイニング教育における探索的データ分析ツールの効用

大平 雅雄[1,a)]　中野 大輔[2,b)]　松本 健一[2,c)]

概要：ソフトウェアリポジトリマイニング (MSR) 技術は，ソフトウェア開発の生産性・品質を改善するための有力な手段の１つである．多くの研究者および実務者が，新たなマイニング手法の提案あるいは既存手法の改良に取り組んでいる．しかしながら，MSR 技術に対して馴染みのない学生および実務者にとって，実際のプロジェクトを対象として MSR 技術を適用する事は容易ではない．MSR 技術を適用するためには，対象プロジェクトのドメイン知識，統計学やデータマイニングに関する知識，データクリーニング手法など，広範囲の知識やスキルを必要とするためである．本稿では，MSR 技術を適用できるようになるまでに必要となる教育を「リポジトリマイニング教育」と位置付け，探索的データ分析ツールの利用について検討をする．また，MSR 技術をこれから学ぼうとする初学者を対象としておこなった被験者実験の結果について報告する．

1. はじめに

ソフトウェアリポジトリマイニング技術（以降，MSR 技術とする）は，ソフトウェア開発の生産性・品質を改善するための有力な手段の１つである．現在多くの研究者および実務者が，ソフトウェア開発に有益な情報や知見をソフトウェアリポジトリから抽出するための新たな手法，あるいは，既存手法の改良に取り組んでいる．MSR (Mining Software Repository) 分野において提案された各種マイニング手法は，すでにソフトウェア開発支援ツールとして組み込まれているもの存在する [1, 2].

しかしながら，MSR 技術に対して馴染みのない実務者にとって，実際のプロジェクトを対象として MSR 技術を適用する事は容易ではない．主な理由には，(1) データ収集コスト，(2) データクリーニングに要するコスト，(3) データマイニングアルゴリズムおよびデータマイニングツール活用するための学習コストなど，多くのコストを必要とすることが挙げられる [3, 4]．MSR 分野の研究者には現在，MSR 技術を研究開発するだけではなく，実務者が MSR 技術を利用する際の障壁をできるだけ低くするための工夫が求められている [3].

[1] 和歌山大学
Wakayama University
[2] 奈良先端科学技術大学院大学
Nara Institute of Science and Technology
[a)] masao@sys.wakayama-u.ac.jp
[b)] daisuke-n@is.naist.jp
[c)] matumoto@is.naist.jp

本稿では，先に挙げた MSR 技術利用に関する諸課題とは別に，MSR 技術をこれから習得しようとする学生および実務者の観点から，MSR 技術の学習の困難さをとりあげる．**図 1** は，Xie らが IEEE Software において紹介している，ソフトウェアリポジトリマイニングの一般的な手順 [4] を示したものである．文献 [4] において Xie らは，ソフトウェアリポジトリマイニングの初期段階について次のように述べている．

> *software engineers can start with either a problem-driven approach (knowing what SE task to assist) or a data-driven approach (knowing what SE data to mine), but in practice they commonly adopt a mixture of the first two steps: collecting/investigating data to mine and determining the SE task to assist.* [4, p.56]

すなわち，ソフトウェアリポジトリマイニングを実施する初期段階では，支援すべき／できる対象 (SE task) はマイニングする対象 (SE data) に依存するため，支援対象を決めてからマイニングを行う (problem-driven approach) のか，マイニング対象を選定してから支援対象を決定するのか (data-driven approach) については通常定かではなく，両方を同時に進めながらおこなう場合が多いことを示唆している．我々のこれまでの経験からも，ソフトウェアリポジトリマイニング初期段階におけるこの曖昧さ（mixture of the first two step）は，MSR 技術をこれから習得しようとする学生および実務者にとっての大きな障壁となってい

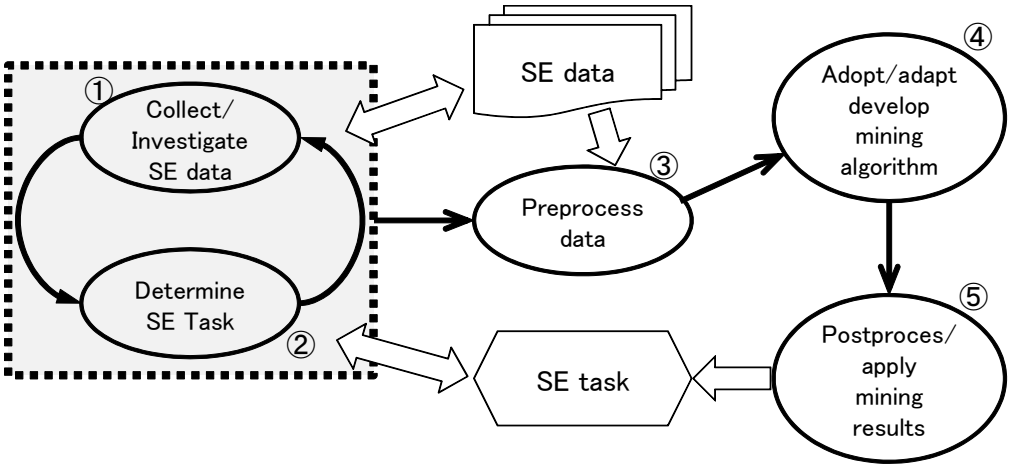

図 1 ソフトウェアリポジトリマイニングの一般的手順
Fig. 1 A general procedure of mining software engineering data

ると思われる．

そこで本稿では，まず続く 2 章において，MSR 技術を習得しようとする初学者にとって障壁となる要素についてまず議論する．次に 3 章では，データマイニングあるいは分析の目的が明確でない段階においても，初学者がデータを探索的に分析し，SE task を明確にするための仮説生成がおこなえるよう支援する方法について述べる．4 章では，メリーランド大学の研究者らによって開発された探索的データ分析ツール HCE (Hierarchical Clustering Explore) を紹介する．5 章では，ソフトウェアリポジトリマイニング教育への探索的データ分析ツールの効果を調べるためにおこなった実験について報告する．6 章で実験結果を考察し，最後にまとめと今後の課題について述べる．

2. ソフトウェアリポジトリマイニング教育

2.1 ソフトウェアリポジトリマイニングにおける課題

本稿冒頭で述べたように，ソフトウェアリポジトリマイニングを実際の開発現場で用いる際，統計学やデータマイニング手法に精通していなければ，

(1) データ収集コスト（どのリポジトリを対象に，どの種類のデータを，どのような形式で取得するべきかを決定し，データ収集のためのスクリプトを作成する必要がある，など）
(2) データクリーニングに要するコスト（膨大なデータからマイニング結果に悪影響を及ぼすノイズとなるデータを除去する必要がある，など）
(3) データマイニングアルゴリズムおよびデータマイニングツール活用のための学習コスト（統計学や機械学習など，データマイニングを実施するにあたって学習する必要があり，また，Weka [5] などのデータマイニングツールを活用する際にも適切なパラメータを選択するためにデータマイニング手法についての学習が必要になる，など）

など，様々なコストが必要となることが多い．

これらに加え，これから MSR 技術を習得しようとする学生あるいは実務者には，

- どのタスクを MSR 技術によって支援すべきか (what SE task to assist)
- どのソフトウェアリポジトリをマイニング対象とすべきか (what SE data to mine)

を決定することは通常困難を極める．

特に，ソフトウェア開発に従事した経験のない学生にとっては，ソフトウェア開発組織での支援ニーズがあらかじめ分かっていないばかりか，開発ドメインに対する知識も不足していることが多いため，「何を支援すべきか」を決めることは難しい．

そのため，まず，散布図やヒストグラムなどを用いてデータの傾向を把握することから始めて，探索的に支援対象を絞り込んでいく必要がある（図 1 における Step2 から Step1 の流れ）．ただし，対象とするソフトウェアリポジトリから取得するデータは通常膨大な量であるため，データ全体の傾向を把握する目的で利用する散布図やヒストグラムのみでは，重要な特徴や問題を発見できないことも多い．

一方，ソフトウェア開発に従事する実務者であっても，統計学やデータマイニングに精通していなければ，「何をマイニングすべきか（特に，どのソフトウェアリポジトリを対象にして，どのような統計手法やデータマイニング手法を用いるべきか）」については必ずしも決められない場合がある（図 1 における Step1 から Step2 の流れ）．

そのため，「何を支援すべきか」について十分理解しているとしても，MSR 技術を使いこなすことができない場合がしばしば生じる．Hassan 氏と Xie 氏らもそれぞれの論文 [3, 4] において指摘しているように，ソフトウェア工学の専門家と統計学／データマイニングの専門家との協働が，MSR 技術を活用するための今後の重要課題となっている．

2.2 ソフトウェアリポジトリマイニング教育における課題

本研究では，MSR 技術に対して馴染みのない学生および実務者を初学者と定義する．また，MSR 技術を適用できるようになるまでに必要となる教育を「リポジトリマイニング教育」と位置付ける．

初学者にとっては，PROMISE データセット [6] のような容易に入手できるデータセットを用いたとしても，図図 1 の Step1 および Step2 を実施することは難しい．実際に，図図 1 のソフトウェアリポジトリマイニングの一連の流れを実施できるようになるためには，事前に十分時間をかけて学習をおこなう必要がある．

例えば，大学の研究室であれば，毎年数名の学部生／大学院生を受け入れることになるが，MSR 技術に関して初学者である学生が，MSR 技術を適用したり新たな手法について研究できるようになるまでには，多くの労力を必要とする．

まず，ソフトウェアリポジトリ（svs/subversion/git や bugzilla/redmine/trac など）の利用方法について学び，リポジトリに蓄積される情報と取得可能なメトリクスにはどのようなものがあるかを理解する必要がある．次に，ソフトウェアリポジトリマイニングに必要な統計手法やデータマイニング手法について多くの時間を割いて学ぶ必要がある．次に，支援対象となるドメイン（我々の場合はオープンソースソフトウェア開発）について学び，解決すべき問題 (what SE task to assist) について，文献調査などをおこないながら理解する必要がある．さらに，膨大なデータからノイズとなるデータを除去するためのデータクリーニング手法なども学ぶ必要がある．

このように，初学者が MSR 技術を習得するためには多大なコストを必要とするため，学習意欲を失ってしまうことも少なくない．次節では，初学者が学習意欲を失わずに段階的に MSR 技術を学ぶための環境について考察する．

2.3 初学者のための環境

初学者が図 1 の Step 3, 4, 5 へと進むためには，Step 1, 2 において散布図やヒストグラムなどを用いてデータの傾向や特徴を把握し，「何を支援すべきか (what SE task to assist)」に結びつく仮説を生成する必要がある．このプロセスは初学者に特有のものではなく，近年ソフトウェア工学分野の研究方法論の主流となっている Resarch Question を設定する作業に相当する．ただし，経験豊富な研究者ではなく初学者が Resarch Question を設定することは容易ではないため，「何度も試行錯誤」できることが初学者の仮説生成プロセスには重要であると思われる．

また，前節で述べたような学習を一通り終えていない段階でも，実際のデータを分析して自分なりに仮説生成できることが，学習意欲の継続という観点からは望ましい．自分で立てた仮説が正しいかどうかを検証しようする中で，統計手法やデータマイニング手法を学ぶ動機付けにもなるからである．そこで本研究では，探索的データ分析 [7] をおこないながら初学者自身で仮説を生成／検証できる環境が，ソフトウェアリポジトリマイニング教育には必要であると考えた．

探索的データ分析とは，明確な分析目的や検証すべきモデルを持たない状態で，与えられたデータに潜むモデルや傾向を多角的に分析することを指す．散布図やヒストグラム，箱ひげ図等は，探索的データ分析をおこなうためのツールとしてしばしば利用される．近年では，探索的データ分析のためのソフトウェアが数多く登場しており，その多くはデータや分析結果をグラフィカルに表示し，ユーザが「試行錯誤」しながらデータを理解したりモデルを構築するのを助ける．

本研究では，MSR 技術を学ぶ初学者にとって，「試行錯誤」のプロセスを支援する探索的データ分析ツールの利用が，「何を支援すべきか (what SE task to assist)」の発見につながる仮説生成を支援できると考えた．本稿では，探索的データ分析ツールの 1 つである HCE (Hierarchical Clustering Explore) を使った実験をおこない，リポジトリマイニング教育への探索的データ分析ツールの効用について調べる．

3. 探索的データ分析と支援ツール

3.1 ソフトウェアリポジトリマイニングにおける探索的データ分析

探索的データ分析は，大規模で複雑な多次元データに隠れたパターンや傾向をつかんだり理解したりする際に有用な分析方法である．ソフトウェアリポジトリマイニングにおいても，対象となるデータは多次元（多変量）であることが一般的である．例えば，fault-prome モジュールを予測するには，コード行数や複雑度をはじめとする様々なメトリクスを変数として計測し，予測モデルを構築する．その際，探索的データ分析は，データセットの特徴を把握するため散布図やヒストグラム，箱ひげ図といった統計ツールの利用により，モデル構築に寄与すると思われる変数の選択を支援することができる．

3.2 HCE (Hierarchical Clustering Explore) [8]

3.2.1 概要

HCE (Hierarchical Clustering Explore) [8] は，メリーランド大学の研究者らによって開発された探索的データ分析のためのツールである．遺伝子解析の分野をはじめとする様々な分野に適用されている．**図 2** は，HCE ツールの外観を示すスクリーンショットである．

HCE ツールはまず，与えられた入力データを階層的ク

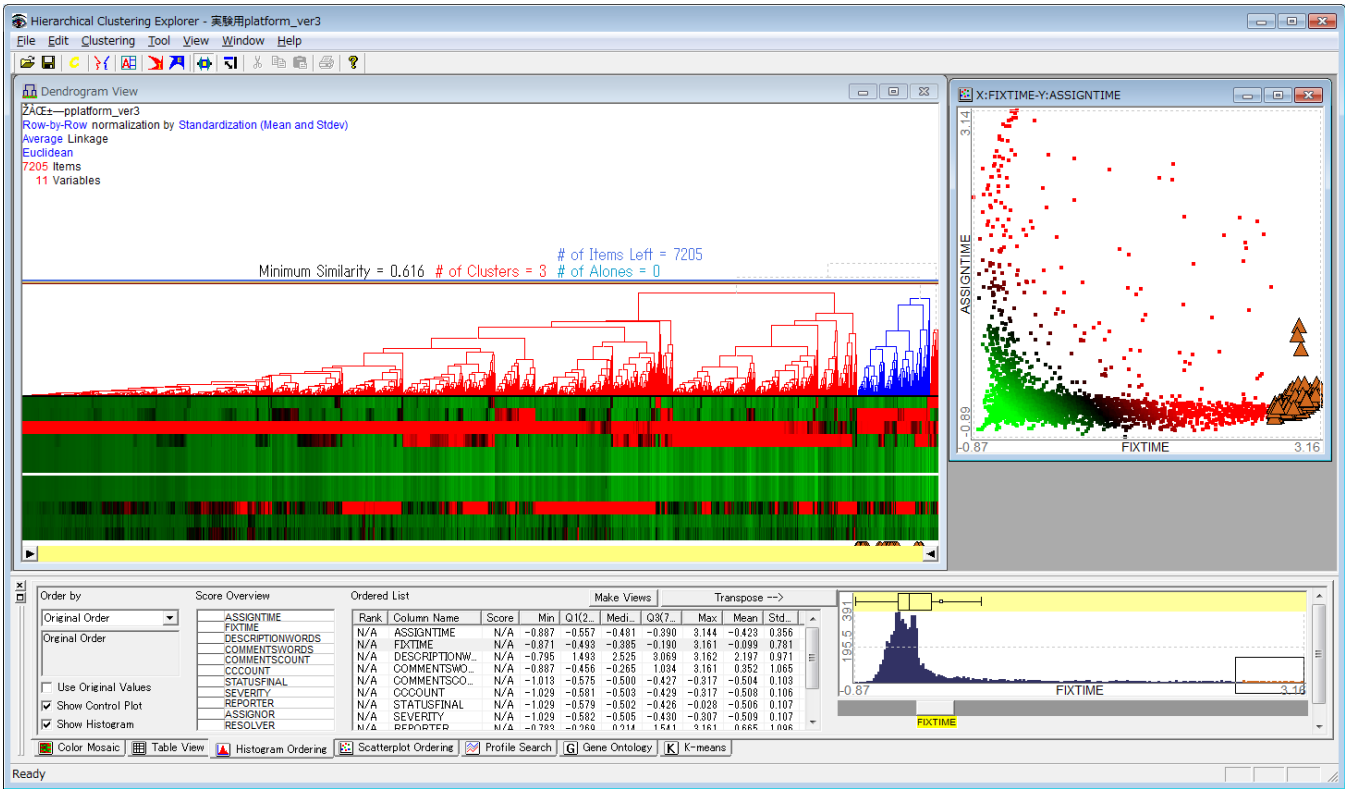

図 2 HCE ツールのスクリーンショット
Fig. 2 A Screenshot of Hierarchical Clustering Explorer [8]

ラスタリングした結果を視覚的に表示する（図 2 左の最も大きな領域）．ユーザが興味のあるクラスタを 1 つ選択すると，選択したクラスタに関連する情報が図 2 下部に表示される．図 2 下部では，ヒストグラムや平行座標プロットなど，探索的データ分析をおこなうためのツールが利用可能である．図 2 右部には，選択したクラスタがデータ全体の中でどの位置に分布しているのかがすぐに見て取れるよう可視化される．

3.2.2 利用方法

図 2 は，Eclipse Platform プロジェクトの約 7,000 件の不具合データを入力として解析結果を表示したものである．Minimum Similarity Bar を用いて調整することで，クラスタリング結果のしきい値（クラスタをいくつ作成するか）を画面上で自由に調整することが出来る．階層クラスタリングの結果から興味のあるクラスタを選択する（図中，青色で表示されているクラスタ）ことで，そのクラスタに関連する情報が画面下部に表示されている．ユーザが選ぶ基準で各変数のスコアを算出／並び替え，各変数を一次元データとして表示する図中の Histgram Ordering タブや，任意の 2 変数間の相関係数がグラフィカルに一望できる Scatterplot Orddering タブを用いながら，変数間の関係や各変数を 1 つ 1 つ確認することができる．また図右部では，それらをグラフィカルに表示する．図 2 右部では，変数 AssiginTime と FixTime の関係を散布図として表示しており，選択したクラスタ（三角形の集合）が全体のどの位置に分布しているかが見て取れる．

	A	B	C	D	E	F	G	H	I
1	BugID	Status	Severity	Reporter	Assignor	Resolver	AssignTime	FixTime	Descript
2	fieldtype	CATEGORICAL	CATEGOR	CATEGOR	CATEGOR	CATEGOR	INTEGER	INTEGER	REAL
3	67972	RESOLVED	normal	daniel_meg	t.s.maeder	Platform-U	8.02	177.25	2
4	67975	RESOLVED	normal	andre_wein	Darin_Swar	platform-c	0.21	1.08	3
5	67999	VERFIED	normal	sonia_dimi	Tod_Creas	douglas.pol	0.05	1.95	3
6	68002	RESOLVED	normal	kellrman	dejan	mfaraj	0.35	0.85	5
7	68013	CLOSED	enhancem	john_arthor	Tod_Creas	platform-in	0.01	1.01	11
8	68021	RESOLVED	normal	michaelvan	dirk_baeum	platform-s	0.59	958.4	8

図 3 HCE ツールの入力ファイルフォーマット
Fig. 3 Input file format for HCE

3.2.3 入力データ形式

データ分析における試行錯誤のプロセスをインタラクティブに支援できる点が，ソフトウェアリポジトリマイニング教育，特に初学者自身による仮説生成に有益であると考え，被験者実験で使用するツールとして採用した．また，**図 3** に示したように，HCE ツールの入力データ形式が非常にシンプルなことも採用理由の 1 つである．

HCE ツールでは，ユーザがデータセットを準備するのを助けるために，タブ区切り／コンマ区切りのテキストファイルの他，MS Excel ファイルを入力ファイルとして利用できる．図 3 のように，一行目には変数名，二行目にはデータの型を記述し，入力ファイルを作成すればすぐに分析が可能となる．

表 1 実験で使用した変数（メトリクス）
Table 1 Variables used in the experiments

変数名	HCE ツールでのデータ型	説明
Resolution	Categorical	不具合報告の最終状態 (RESOLVED, VERIFIED, CLOSED など)
Component	Categorical	不具合の対象コンポーネント名
Severity	Categorical	報告された不具合の重要度
Time to assign	Integer	不具合が報告されてから修正者に不具合修正を依頼するまでに要した時間
Time to fix	Integer	不具合修正を依頼されてから不具合が修正されるまでに要した時間
Reporter	Categorical	不具合報告者の email アドレス（報告者を特定する情報として利用）
Triager	Categorical	管理者の email アドレス（管理者を特定する情報として利用）
Fixer	Categorical	修正者の email アドレス（修正者を特定する情報として利用）
Description words	Real	不具合報告中の “descriptions” 欄に記述された単語の数
Comments	Real	不具合報告に対するコメントの数
Comments words	Real	不具合報告に対するコメントに含まれる単語の数
CC	Real	不具合報告に関係する開発者の数（同報通知数）

4. 被験者実験

本章では，ソフトウェアリポジトリマイニング教育に対する探索的データ分析ツールの効用を調べるためにおこなった被験者実験について述べる．本実験の主な目的は，探索的データ分析ツールを用いることで，初学者が

- 与えられたデータセットから変数間の関係を導きだせるかどうか
- さらにマイニングするに値する仮説を生成できるかどうか

を確かめることである．

4.1 データセット

被験者に与えるデータセットとして，Eclipse Platform プロジェクトから Eclipse 3.2 に関連する不具合票を約 7,000 件収集した．なお，計測可能なメトリクスをすべて被験者に与えると，分析すべき対象が多くなりすぎることが懸念されたため，12 件のメトリクスを実験者が選び，最終的なデータセットとして与えている（12 メトリクス ×7,000 件のデータセット）．**表 1** は，実験で使用したメトリクスの一覧である．

4.2 被験者

オープンソース開発に関する知識（すなわちドメイン知識）がほとんどない学部生 3 名と，OSS プロジェクトの不具合報告データを対象としてデータマイニングした経験のある大学院生 3 名に，実験の被験者として参加してもらった．前者の学部生 3 名は情報科学分野で 3 年以上学んだ学生であり，講義の 1 つとしてソフトウェア工学の履修経験がある．

後者の大学院生 3 名は，エンピリカルソフトウェア工学の分野で研究を少なくとも 1 年以上おこなった経験があり，オープンソース開発における不具合報告データについてはある程度の理解がある．大学院生 3 名のうち 1 名は特に，Eclipse Bugzilla を対象としてソフトウェアリポジトリマイニングを過去におこなった経験がある．

これら計 6 名の被験者は，あらゆる統計手法やデータマイニング手法について精通している訳ではなく，程度の差はあるものの本研究に置ける「初学者」と見なすことができる．

4.3 実験手順とタスク

実験に先立ち，被験者には HCE ツールの利用方法と使用するメトリクスについて説明した後，食品に含まれる栄養素をまとめたサンプルデータセット（本実験には無関係のドメインにおけるデータ）を与え，HCE ツールを試用してもらった．HCE ツールの利用に大きな支障がないことを確認した後，前述したデータセットを与えて実験を開始した．実験時間は 1 時間とした．

被験者には，データセットに含まれる興味深い関係や傾向を，何を支援すべきか (what SE task to assist) につながるよう，仮説として抽出するよう指示した．また，仮説を思いついた時点でメモとして記録しておくよう依頼した．

4.4 実験結果

実験結果を**表 2** に示す．表中の，Su は学部生を，Sg は大学院生であることを示している．また，**太字**は仮説生成に使われたメトリクス（変数）名を，下 線は「何を支援すべきか (what SE task to assist)」につながる仮説であることを示している．[**BA**] および [**QR**] はそれぞれ，不具合割り当て (Bug Assignment) 方法改善の必要性を示唆するもの，不具合報告の品質 (Quality of bug Reports) の改善の必要性を示唆するものである．

4.4.1 生成した仮説の数

学部生 3 名の被験者が生成した仮説は，合計 8 件（平均：約 2,7 件）であった．大学院生 3 名の被験者が生成した仮

表 2 実験結果
Table 2 Results of our experiment

被験者	生成した仮説の数	具体例
Su1	3	・重要度 (**severity**) の高い不具合は多くの **CC** を持つ傾向にある．レビューされる回数が多い不具合ほどプロジェクトによって重要な不具合かもしれない． ・不具合が修正者に割り当てられるまでの時間 (**Time to assign**) は，管理者 (**Triager**) によって異なる．修正依頼までに要する時間を短縮しようとしているプロジェクトとって，管理者の選出は非常に重要な要素となる [**BA**] のでは．
Su2	3	多くのコメントが付けられた (**Comments**) 不具合報告は，(**Resolution**) が **RESOLVED** や **VERIFIED** となっている気がする．もし報告者が不具合を本当に直してほしいのであれば，開発者がコメントの内容を理解できるよう明瞭に不具合の内容を記述すべきである [**QR**].
Su3	2	・たくさんの **CC** をもつ不具合は，コメント (**Comments**) がたくさん付けられるようだ．コメントは他の開発者にも見てもらいたくて付けられているのでは？
Sg1	8	・**Time to assign** が短いときは，**Time to fix** にばらつきがあるが，**Time to assign** が長ければ **Time to fix** が短くなっている．慎重に不具合を割り当てることがプロダクトの品質改善につながるの [**BA**] では． ・**Time to assign** が長いほど，コメント数 (**Comments**) と **CC** が多くなっている．これらの不具合は開発者間で十分な議論がおこなわれているのでは． ・重要度 (**severity**) が **critical** となっている不具合は，より早く修正依頼され (**Time to assign**)，修正も早い (**Time to fix**)．これらの不具合には特定のエキスパート開発者を割り当てるようなルールが存在するのではないか．
Sg2	5	・修正依頼時間 (**Time to assign**) と修正時間 (**Time to fix**)．がともに長いクラスタに着目すると，descriptions の単語数 (**Description words**) が多い傾向にある．不具合をより正確に報告することは，不具合の修正時間を改善するのに役立つ [**QR**] のでは． ・管理者 (**Triager**) のヒストグラムを見ると，少数の管理者が不具合を多数割り当てていることが分かる．また，それらは修正されるのが遅くなる (**Time to fix**) 傾向がある．こうした事態を防ぐためには，適度な負荷分散をおこない不具合修正プロセスを改善する必要がある [**BA**] のではないか．
Sg3	5	・重要度 (**severity**)**critical** の不具合は，タスク割り当ても (**Time to assign**) 修正も早い (**Time to fix**)． ・コメント数 (**Comments**) と descriptions の単語数 (**Description words**) の散布図から，ほとんどコメントがなく単語数も少ないクラスタに分布する不具合は，コメント回数／単語ともに多い不具合に比べて，修正されるのが遅いことが分かった．不具合報告の情報は，（不具合修正プロセスを改善するために）重要である [**QR**] と思われる．

説は，合計 18 件（平均：6 件）であった．実験前に予期していた通り，ドメイン知識に乏しい学部生の方が仮説生成数は少ない結果となった．

4.4.2 仮説の質

仮説生成数は学部生の方が少ない結果となったが，学部生 3 名のうち 2 名は，大学院生 3 名が生成した仮説に近い，あるいは，同等の仮説を生成できていることが分かった．

例えば被験者 Su1, Sg1, Sg2 はともに，「不具合修正プロセスを改善するためには慎重なタスク割り当て [**BA**] が必要である」という趣旨の指摘をおこなっている．また，Su2, Sg2, Sg3 はともに，「不具合報告の品質 [**QR**] が，不具合修正時間を短縮化する上で重要な要因である」という趣旨の指摘をおこなっている．

これらの指摘は，ソフトウェアリポジトリマイニング分野の先行研究においてもすでに確認されており，支援手法もすでに多数提案されているもの [9–13] である．したがって，探索的データ分析ツールを用いることで，初学者であっても「何を支援すべきか (what SE task to assist)」につながる重要な仮説を生成することが可能であると言うことができる．

こうした重要な仮説は，学部生 3 名の被験者で合計 2 件，大学院生 3 名で合計 4 件が生成された．どちらのグループにおいても，生成できたすべての仮説のうち約四分の一は，「何を支援すべきか (what SE task to assist)」につながる重要な仮説であることが分かった．

5. 考察

5.1 探索的データ分析の効用

本稿では，ソフトウェアリポジトリマイニング教育における探索的データ分析ツールの利用とその効用について調べるためにおこなった実験の結果を示した．実験の結果，ドメイン知識の違いによって生成できた仮説の数は，学部

生と大学院生とで大きな差が見られたが，仮説の質という観点では，どちらも重要な仮説を生成できていることを確認した．

探索的データ分析ツールを利用させるという我々のアプローチは，シンプルかつ構造化されたデータセットを用いる場合には有効であるが，変数を数十件含む様なより複雑なデータセットや，テキストマイニングで利用されるような非構造化データセットを用いるには限界がある．そうした限界を除けば，初学者があまり馴染みのないドメインについて具体的に考える材料を提供できるという点で，探索的データ分析ツールの利用は教育／研究の観点からも有益であると考える．実際，今回の実験においても，「なぜこの変数がこの変数と関係が高いのか？」を，段階的かつ具体的に考えられることが「何を支援すべきか (what SE task to assist)」を明らかにする上で重要であり，初学者が複雑なデータを分析する際の動機を高めていることがうかがい知れた．

今回の実験結果は，実験内容とメトリクスの説明および HCE ツールの試用を含めて合計 2 時間のツール試用経験で得られたものである．被験者は HCE ツールの機能すべてに熟知していた訳ではないが，我々の予想に反して多くの仮説生成をおこなえた．HCE ツールの利用方法そのものに習熟すれば，さらに多くの仮説を生成できるものと思われる．

5.2 制約

今回の実験では，実験者が選んだ 12 種類のメトリクスからなる 7,000 件分のデータセットを用いた比較的小規模な実験であった．そのため，得られた知見の一般性および妥当性は高くないものと考えている．また，HCE ツールを用いないで，表計算ソフトウェアのみを用いて分析した結果との比較をおこなっていないため，探索的データ分析ツールそのものの効果がどれくらい存在するのかについても厳密には定かではない．今後は，Excel 等の表計算ソフトウェアのみを用いた実験をおこない，生成できる仮説の量と質の面から探索的データ分析ツールの効用をより正確に明らかにしたい．

また，今回の実験では，HCE ツールの利用方法については被験者に 1 時間程度試用してもらったのみで実験を開始した．そのため，被験者毎に分析の手順や利用する機能にばらつきがあり，生成できた仮説の数にも違いが現れていると考えている．今後は，探索的データ分析ツールを用いて効率よく仮説生成するための手順あるいは枠組みを構築する必要があると考えている．

6. まとめと今後の課題

本稿では，ソフトウェアリポジトリマイニング教育における探索的データ分析ツールの利用とその効用について調べる実験をおこなった．HCE ツールを用いて実験をおこなった結果，生成できた仮説の数は，学部生よりも大学院生の方が多かったが，仮説の質に関しては，どちらの被験者も今後さらに検証するに値する重要な仮説をいくつか生成できていた．これらの結果から，ソフトウェアリポジトリマイニング教育における初期段階において，探索的データ分析の利用は，さらに詳細に仮説を検証するために今後必要となる統計手法やデータマイニング手法を，初学者が学ぶ動機付けにつながるものと期待できる．

本研究は，ソフトウェアリポジトリマイニング教育において初学者の学習を段階的に支援することが最終的な目的ではあるが，現時点ではまだ十分とは言い難い．今後以下の様な課題に取り組む必要があると考えている．

- ソフトウェア工学データを探索的データ分析ツールを用いて体系立てて分析する枠組みを構築すること
- より大きな規模（被験者数，データセットのサイズ，使用する変数の数）の実験をおこない，実験結果の一般性および妥当性を向上させること
- Bugzilla から取得した不具合報告データのみではなく，CVS や subversion といったバージョン管理システムから取得したデータを使った実験をおこなうこと
- 経験豊富な実務者と初学者が探索的データ分析ツールを用いてデータ分析をおこなた場合の比較をおこなうこと

謝辞 本研究の被験者実験に参加して下さった，奈良先端科学技術大学院大学情報科学研究科の皆様，和歌山大学システム工学部の皆様に深く感謝します．本研究の一部は，文部科学省科学研究補助金（基盤 (B):23300009）および（基盤 (C):24500041）による助成を受けた．

参考文献

[1] Coverity, Inc.: http://www.coverity.com/.
[2] Pattern Insight Inc.: http://patterninsight.com/.
[3] Hassan, A. E.: The road ahead for Mining Software Repositories," Frontiers of Software Maintenance, *Frontiers of Software Maintenance (FoSM)*, pp. 48–57 (2008).
[4] Xie, T., Thummalapenta, S., Lo, D. and Liu, C.: Data Mining for Software Engineering, *Computer*, Vol. 42, No. 8, pp. 55–62 (2009).
[5] Weka: Data Mining Software in Java: http://www.cs.waikato.ac.nz/ml/weka/.
[6] PROMISE Software Engineering Repository: http://promise.site.uottawa.ca/SERepository/.
[7] Tukey, J. W.: *Exploratory Data Analysis*, Addison-Wesley (1977).
[8] Seo, J. and Shneiderman, B.: Interactively Exploring Hierarchical Clustering Results, *IEEE Computer*, Vol. 35, No. 7, pp. 80–86 (2002).
[9] Ohira, M., Hassan, A. E., Osawa, N. and Matsumoto, K.: The Impact on Bug Management Patterns on Bug Fixing: A Case Study of Eclipse Projects, *Proceedings of*

the 2012 International Conference on Software Maintenance (ICSM2012), pp. 264–273 (2012).

[10] Anvik, J., Hiew, L. and Murphy, G. C.: Who should fix this bug?, *Proceedings of the 28th international conference on Software engineering (ICSE '06)*, pp. 361–370 (2006).

[11] Jeong, G., Kim, S. and Zimmermann, T.: Improving bug triage with bug tossing graphs, *Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering (ESEC/FSE '09)*, pp. 111–120 (2009).

[12] Zimmermann, T., Premraj, R., Bettenburg, N., Just, S., Schroter, A. and Weiss, C.: What Makes a Good Bug Report?, *IEEE Transactions on Software Engineering (TSE)*, Vol. 36, No. 5, pp. 618–643 (2010).

[13] Breu, S., Premraj, R., Sillito, J. and Zimmermann, T.: Information needs in bug reports: improving cooperation between developers and users, *Proceedings of the 2010 ACM conference on Computer supported cooperative work (CSCW '10)*, pp. 301–310 (2010).