ソフトウェア開発記録の多次元データ分析に向けた可視化方式 Treemap Forestの設計と実証的評価



中川 尊雄※　伊原 彰紀※　松本 健一※

ソフトウェア品質の第三者評価を行う分析者は，開発に従事していない者であることが妥当であり，客観的な視点から探索的解析を行うことが期待されている．しかし，ソフトウェア開発データに含まれる様々な要素（成果物，課題票，組織形態）の間にある関連性を考慮しながら解析を行うには工数を要する．本論文では，データ間の関連性に基づき，開発データの俯瞰を次々提示することで多次元データ分析を実現する新たな可視化方式Treemap Forestを提案する．Treemap Forestでは，データ間の関係性を明示化するため，開発データを関係データベース形式で表現し，開発データに対する探索的データ解析を実現する．有用性の評価実験を実施し，提案手法の利用は，従来よく用いられてきたExcelに比べ，約38%の時間でデータを解析できることを示した．

Treemap Forest: An Exploratory Data Visualization Approach for Software Development Project Datasets

Takao Nakagawa, Akinori Ihara, Kenichi Matsumoto

Software quality analysts in an independent evaluation organization should not be members of the software development team, because they are expected to perform evaluations from a neutral perspective. However exploratory data analysis targeting several kinds of software development datasets (e.g., products, issues, and members) is not easy to understand for analysts. In this study, we propose Treemap Forest, which is an exploratory data visualization approach for software development project datasets, and develop a prototype system with Treemap Forest. In order to evaluate the approach, we compare exploratory data analysis using Treemap Forest with traditional approaches. The Treemap approach can conduct tasks in 38% of the time taken by traditional approaches.

1　はじめに

ソフトウェア開発ベンダが製品・システムを提供する場合，利用者に対して安全性や信頼性をはじめとする品質についての説明を付することが求められる．しかし開発者が自ら評価することは妥当ではなく，「専門知識を有する中立的立場の第三者」が客観的に品質評価を行い，専門知識を持たない利用者にも理解できる説明を提示する仕組みが必要である［IPA］．

※奈良先端科学技術大学院大学, Nara Institute of Science and Technology

我々は，ソフトウェア品質の第三者評価の技術基盤の確立に向けて，ソフトウェアやその品質が実現される過程を解析・可視化するための概念「ソフトウェアプロジェクトトモグラフィ」を提唱・構築してきた[IPA2012][IPA2013]．ソフトウェアプロジェクトトモグラフィでは，ソフトウェア開発プロジェクトを，多様なプロジェクトデータとその解析結果から成るスナップショットの系列で表現する．スナップショットの系列から開発体制や開発速度などの俯瞰的な解析，また，特定のスナップショットから開発中に発生したイベントの解析を支援する．

本論文では，ソフトウェアプロジェクトトモグラフィを構成する要素技術である「ソフトウェア開発データの客観的な解析・可視化」を実現するための新たな可視化方式“Treemap Forest”を提案し，プロトタイプシステムを開発した．Treemap Forestの効果について被験者実験を行った．続く２節ではソフトウェア開発データに対する探索的データ解析の課題を，３節では課題を解決する可視化方式の設計を示す．４節でそのプロトタイプ実装について述べ，５節で実験設定を示し，その結果を６節に示す．

2　探索的データ解析

2.1　ソフトウェア品質評価のための探索的データ解析

昨今のソフトウェア開発では，版管理システム，課題管理システムなどを活用した開発データ（ソースコードの変更履歴，既知の欠陥情報など）が習慣的に記録されるようになりつつある．これらの開発データは膨大であり，その品質評価は第三者にとって工数のかかる作業である．従来研究では，膨大な開発データに対して，明確な分析目的を持たない状態で，データに潜むモデルや傾向を多角的に分析・評価する手法である探索的データ解析[Tukey]の有効性が評価されている．

ソフトウェア品質の第三者評価のために実施される探索的データ解析の主たる目的は，文献[IPA]における第三者評価のスコープのうち，①プロセス実施，②採用規格・技術の妥当性に対する検証的データ解析の前提となるモデルや基準の提供である．開発データは多変量データであることが多く，従来研究では，ソフトウェア開発データの多変量データから統計的に関連の強い変数を発見し，探索的データ解析を支援するツールHCE（Hierarchical Clustering Explorer）[Seo]の有効性が明らかにされている[大平]．

2.2　ソフトウェア開発の多次元データ解析

ソフトウェア開発データを解析する場合，課題，作業，成果物，人員など，異なる視点を同時に調査することが多い．具体的には図１のように成果物に関する「ソースコードの解析結果」や課題に関する「課題の優先順位や担当者」など，複数の二次元表で表される多変量データが解析対象となる．従って，開発データは関係データベース（RDB）のように，複数の表が共通に保持する要素（キー）で紐づけて取り扱うことが望ましい．しかし，多くの分析ツールや検証環境では複数の表を人手によって紐づけ，要素間の関係に潜む課題や特徴を解析しており，その分布や関係性の俯瞰的な把握は容易ではない．

本論文では，多変量データを俯瞰的に解析し，複数の多変量データの関係を探索的に解析する可視化方式を提案する．

ファイル名	行数	複雑度
Foo.java	241	31
Bar.java	122	15
Fizz.java	31	2
⋮	⋮	⋮

(a)成果物の表

課題ID	担当者	優先度
#1	Alice	高い
#2	Bob	低い
#3	Carol	高い
⋮	⋮	⋮

(b)課題の表

図１　課題と成果物に関する二次元表の例

3　可視化方式の設計

3.1　概要

本研究では，ソフトウェアトモグラフィが取り扱う多変量データについて，互いに関連する複数の多変量データを同時に可視化することで俯瞰的な可視化を実現する可視化方式を検討する．これは情報可視化における従来の原則“Overview first, zoom and filter, then details on demand”を実現する可視化方式であり，分析者がソフトウェア開発データを多角的な視点から効率的に解析を行う助けとなる．

本研究では，Treemap［Jenifer］と呼ばれる多次元データに対する可視化手法が持つ機能を，ソフトウェア開発データが持つ多変量・多次元データに合わせて拡張する可視化方式Treemap Forestを提案する．具体的には，以下の要件に基づいて，ソフトウェア開発データに対する新たな可視化方式の設計を行う．

(1)要素に含まれる特徴量の俯瞰的な提示

(2)要素間関係に基づく探索的データ解析の支援

3.2　要素に含まれる特徴量の俯瞰

二次元表で表されるソフトウェア開発データにおいて，特定の要素を構成する個々の項目が全体に占める割合や偏

りを解析することで，当該要素の特異点が発見される．

提案手法で利用するTreemapは，数値の比を面積の比として可視化する手法である．例えば図２は，Eclipseプロジェクトを対象に，モジュール別の課題数の比を可視化したもので，個々の矩形は各モジュールを，その面積は課題数を表している．以降，個々の矩形を「項目」，面積を決める量的属性を「尺」と呼ぶ．加えて，項目のソフトウェア開発データにおける分類（モジュールならば，成果物）を「要素」と呼ぶ．この表記に従うと図２は，「要素：成果物，項目：モジュール，尺：課題数」についてのTreemap可視化図である．

Treemapでは，同じ「成果物」の要素であっても異なる項目（例えば，ファイル，クラス），異なる尺（複雑度，コード行数）を指定できる．

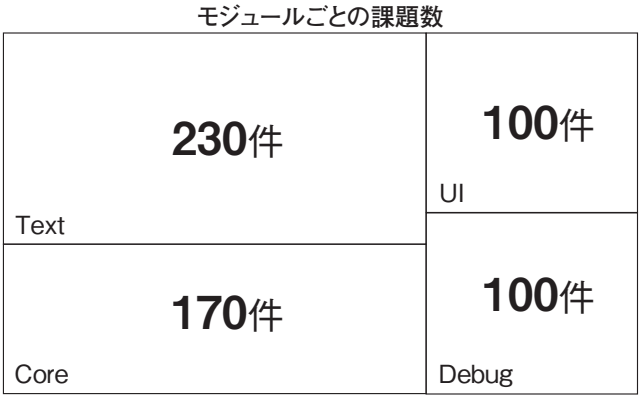

図２　Treemapの概略図

3.3　要素間関係に基づいた探索的データ解析

ソフトウェア開発データ中に出現する要素は，ほかの要素と関係している．例えば「開発者」は「課題を担当する」と「成果物を編集する」といったように，ほかの要素（課題，成果物）と関係する．従来，このような関連づけの作業は人手で複数の表を参照して行われてきたが，表の数が増えるにつれ現実的な方法ではなくなる．

Treemap Forestでは，RDBにおけるキーの機能を用いてこれらの関係を表現し，関係性に基づいて画面上に複数のTreemapを展開することで，作業の負担を軽減する．複数のTreemapを，データの関係に基づいて同時に展開する実際のイメージを，図３に示す．

図３上図は，各開発者が担当したプロジェクトの数についてのTreemapである（要素：組織，項目：開発者，尺：担当プロジェクト数）．

ここで，例えば分析者が開発者Aに注目して分析を行いたい場合，Treemap Forestでは開発者Aを表す矩形の内部に，開発者Aに関する新たなTreemapを入れ子状に展開できる．

図３下の各図は，それぞれ，開発者Aと関連する各要素について新たなTreemapを展開した際の図である．図中の赤枠部分の内側に新たなTreemapが展開されており，注目されていない開発者B・Cについては以前のままの要素・項目・尺度の組が選択されている．

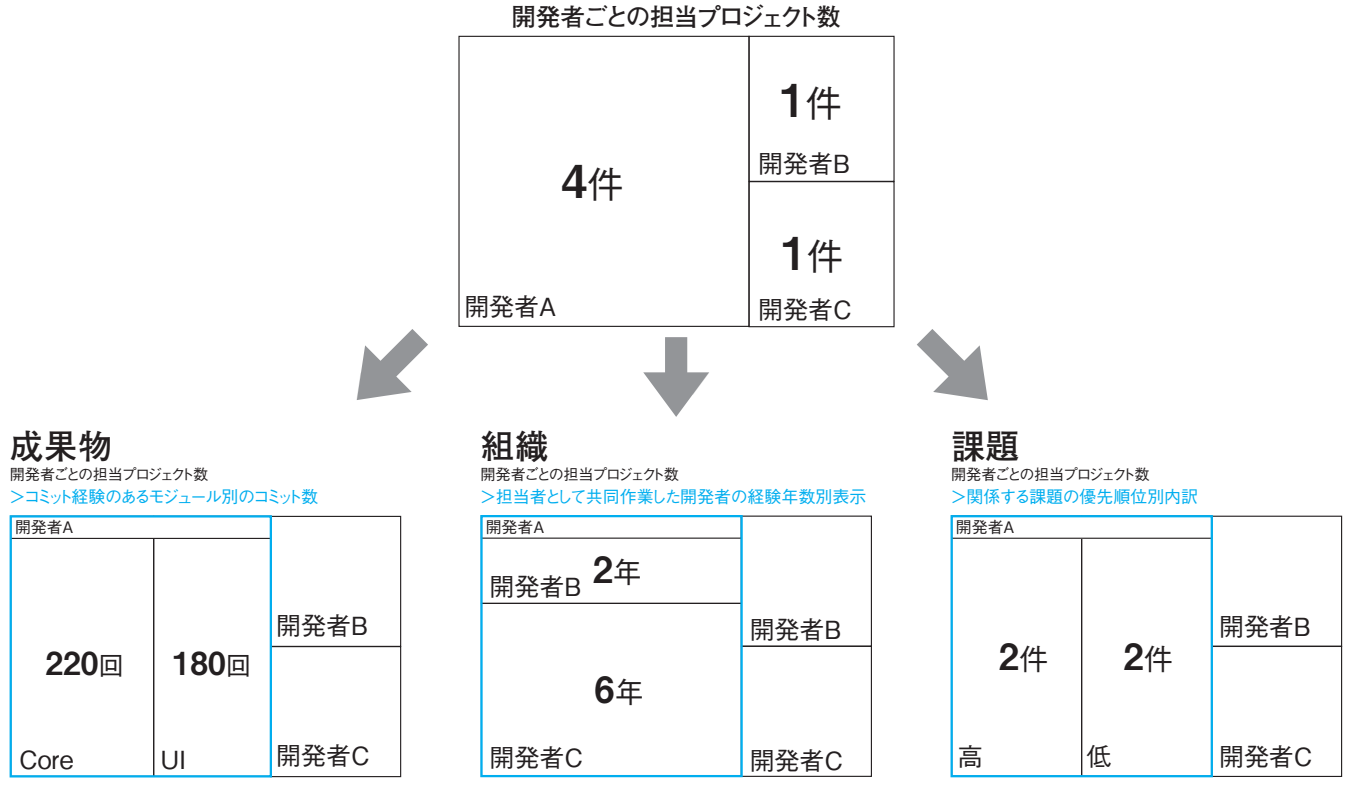

図３　外部キーを利用したTreemapの入れ子

図３下の各図で展開された新たなTreemapを構成する要素，項目，尺の組を次に示す．

- 左　：要素（成果物），項目（モジュール），尺（コミット数）の場合
- 中央：要素（組織），項目（共同作業を行った開発者），尺（経験年数）
- 右　：要素（課題），項目（優先順位），尺（課題数）

入れ子にしたTreemapを更に展開していくと，際限なく要素間の関係を可視化できるが，一方でひとつのTreemapの面積は小さくなり，視認性が下がると考えられる．そこで，Treemap Forestでは，ひとつのTreemapを選択して，画面全体へ拡大する機能（ズームアップ機能）を提供することで，この影響を排除する．

4　プロトタイプシステムの実装

我々は，設計した方式の有効性を評価するため，提案するソフトウェア開発データ可視化方式Treemap Forestのプロトタイプシステムを開発し，その有用性を確認した．本節では，Treemap Forestのプロトタイプシステムの概要を述べる．

4.1　システムの構成要素

本プロトタイプシステムは，可視化方式Treemap Forestを実装したものであり，ソフトウェア開発中に記録される要素（成果物，組織，課題，作業）に関する二次元表データの特徴を俯瞰的に提示し，更に，おのおのの二次元表データ間に紐づけられた情報を用いて，ほかの要素との関係，分布を提示する．分析者は，容易に複数の要素間の関係や特徴を俯瞰的に把握し，探索的データ解析を実現する．

4.2　システムの構成要素

プロトタイプの構成要素を図４に示す．それぞれの構成要素を述べる．

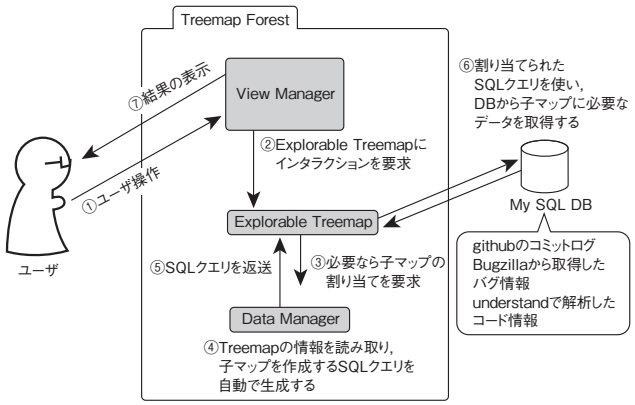

図４　Treemap Forestの構成

ViewManager：分析者の入力(要素,項目)を受け付け，ExplorableTreemapによって生成されたTreemapを提示する．

ExplorableTreemap：分析者の入力をDataManagerに受け渡して得たSQLクエリを用いてDBに問い合わせ，返却された値によってTreemapを構成する．

Treemapの構成には，Ben Fryが開発したTreemapライブラリを用いている．同ライブラリは，Martin Wattenberg，Ben Bedersonらが開発したJava用のライブラリをProcessing用に改造したもので，MPLライセンスの下配布されている．

DataManager：ExplorableTreemapから受け付けた入力情報をもとにSQLクエリを生成する．

DB：ソフトウェア開発データを関係データベース管理システムMySQLで一元管理する．

4.3　システムの操作と特徴

可視化方式Treemap Forest，及び，開発したプロトタイプの操作と特徴を述べる．

分析者は，評価対象プロジェクトの詳細，また，プロジェクトから収集された開発データの詳細を把握しておらず，品質評価において調査すべき事柄を知る手がかりがない．従って，Treemap Forestのプロトタイプでは，起動後に要素選択を行うと，項目，尺はランダムに決定され可視化結果が出力される．もちろん可視化後に，手動で項目，尺を変更することが可能である．図5は起動後に項目（課題）を選択後の出力結果で，優先順位別の課題数が表示されている．ここで，P１からP５は課題の優先順位のレベルを示す（P１が最も高くP５が最も低い）．

分析者がある項目とほかの要素の関係を解析する場合，新たに展開するTreemapの要素を選択する必要がある．プロトタイプシステムでは，項目をクリックすることで，各要素を表す扇形のボタン（上：組織，右：課題，左：成果物）を組み合わせた円形の要素セレクタを用意した．要素が選択されると，項目内に新たな Treemapが展開される．例えば図７は，図６に示されるセレクタで要素（成果物）を選択し，項目としてモジュール，尺として課題数が選ばれた場合に得られた出力であり，優先順位３の課題すべてについて，その課題数をモジュール別に表示したものと解釈できる．

プロジェクトの内容について俯瞰的な解析を繰り返した後も，ランダム性の機能は偶発的発見のために役立つ．もし分析者が望まない項目，尺を選択した場合は，項目，尺を選択しなおすための機能を用いて解決することができる．

最終的に，評価者は複数の要素間に見られる関係性を次々と可視化していく過程で，「最も複雑度の高いソースコードを担当した開発者」のような多要素にまたがる探索的データ解析を，従来手法（Excelなど）に比べて短時間で実現できる．

図5　優先順位別の課題数

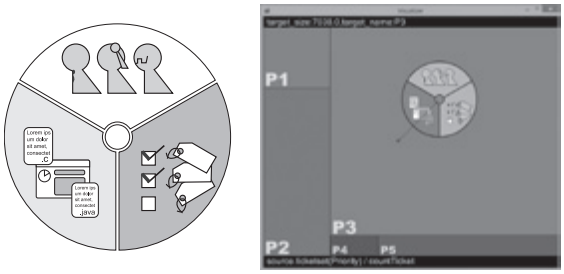

図6　要素セレクタ（左）と表示例（右）

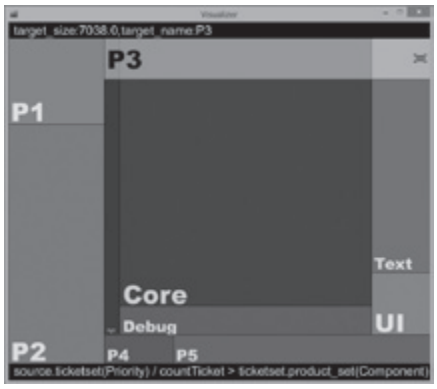

図7　優先順位３の課題のモジュール別課題数

5　被験者実験

5.1　概要と目的

ソフトウェアトモグラフィ可視化方式Treemap Forestを利用することで，ソフトウェア品質第三者評価の初期段階における，効率的な探索的データ解析の実現を評価するために被験者実験を行った．

実験では，分析対象のプロジェクトに対する深い知識を持たず，Treemap Forest，及び，そのプロトタイプの操作を初めて行う被験者が，(1) Treemap Forest，及び，プロトタイプを利用することで，プロジェクトデータの概要を短時間で探索できるか，(2) Treemap Forestを利用することで，プロジェクトの特徴，複数の要素に関係する知見を得ることができるか，を調べるため，二種類の実験を行った．

被験者を二群に分け，ある群ではTreemap Forestを，もう一方の群ではExcel 2013を用いて同内容の実験を行う．

実験１では，「最も関連チケット数の多いコンポーネント名を答えよ」のような，プロジェクトデータについての質問に回答するタスクを計６個用意し，それぞれのタスクにかかる時間を計測する．

実験２では，分析対象のソフトウェア開発に直接精通していない，ソフトウェア工学研究者が解析することを想定してプロジェクトデータを自由に探索してもらい，有用と思われる知見を発見してもらう．

実験結果の分析は，タスク完了時間，発見した知見の数・性質を比較する．

表１　実験に用いたプロジェクトデータ
（主キー，外部キーは他表との関係を表す）

要素	表	項目	尺
課　題	課題表	課題ID（主キー）	ー
		担当開発者名（外部キー）	関係チケット数
		報告開発者名（外部キー）	関係チケット数
		モジュールID（外部キー）	関係チケット数
		進捗状態	チケット数
		解決状態	チケット数
		チケット種別	チケット数
		優先順位	チケット数
組　織	開発者表	開発者名（主キー）	被割当てチケット数
成果物	ファイル表	ファイルID（主キー）	行数
			合計複雑度
			最大複雑度
			平均複雑度
			コメント割合
		モジュールID（外部キー）	所属ファイル数
	モジュール表	モジュールID（主キー）	総行数
			平均複雑度
			最大複雑度
			コメント割合
			継承ツリーの深さ
作　業	コミット表	コミットID（主キー）	ー
		モジュールID（外部キー）	コミット数
		開発者名（外部キー）	コミット数

5.2　対象データ

可視化対象の開発データはオープンソースソフトウェアであるEclipse JDTプロジェクトから収集した．各データは課題追跡システム，版追跡システム，そしてテクマトリックス社のソースコード解析ツール「Understand Ver.2.6※」から別々に得られた計５つの表から成る．表１に，対象データの構成を表す．本実験データでは，要素（作業）に紐づく項目・尺が存在しないため，これを可視化の対象としない．ただし，作業に関する表は利用される．

※Understand：http://www.techmatrix.co.jp/quality/understand/

表２　実験１における質問の一覧

Q１	チケットのPriorityについて，最もありふれたものはどれか
Q２	UIコンポーネントで最も複雑度の高いファイルの複雑度は幾つか
Q３	Enhancementチケットは全部でいくつあるか
Q４	Debugコンポーネントに多くコミットした主要な開発者の名前を５名挙げよ
Q５	UIコンポーネントに多くコミットした５人の開発者の中に，ひとり，ほとんどのチケットが解決できていない開発者が居る．特定せよ
Q６	最もチケット数の多い３つのコンポーネントについて，チケットの進捗状況に違いがあればそれを述べよ

5.3　被験者

被験者は，ソフトウェア工学，データマイニングに関連する研究に取り組む大学院生８名（修士課程７名，博士課程１名）である．それぞれの被験者は，プロジェクトに含まれるデータに対して，個人差はあるものの一定以上の知識を持つものである．また，被験者はいずれもExcelを用いたデータ分析，関数などの利用経験がある．一方で，本実験以前にTreemap Forestの使用経験は一度もない．

5.4　実験１

実験１では，5.1節で述べた(1)並びに(2)の一部を検証するため，プロジェクトデータに関する質問（計６個）を与え，解答までにかかった時間を測定する．Excelを利用する被験者は実験中にヘルプを参照することができ，Treemap Forestを利用する被験者は操作法に関するガイドを読むことができる．実験で用いた質問を表２に示す．なお，一問に15分以上かかった場合は解決不能とみなし，次のタスクに進んでもらう．

６つの質問のうち，Q１～Q３はTreemap Forestでは単一のツリーマップを見るだけで答えられる．Excelにおいても，一つの表を操作して答えられる．Q４～Q６は，Treemap Forestではツリーマップを何段か入れ子にする必要があり，Excelにおいては，複数の表を操作して答える必要がある．

5.5　実験２

実験２は5.1節で述べた目的(2)を検証するために，対象プロジェクトの開発データを自由に探索してもらい，有用と思われる知見を発見してもらうものである．本実験における知見は，研究者による第三者評価を想定しているため，プロジェクトの成功・失敗に関する状態や特徴的な要素について述べるものである．知見の例として実験開始前に被験者に提示したものを次に挙げる．

- UIコンポーネントに属するファイル群は，全体的に複雑度が均質であり，ファイルの切り分けがうまく行われたのではないか
- APTコンポーネントにおいて，優先順位の高いチケットはほとんど解決済みであり，優先順位づけがうまく働いているようだ

探索を行う時間は15分とし，Treemap Forestを用いる被験者は生成した知見をスクリーンショットと共に，Excelを用いる被験者は表から取得したデータと共に，記録してもらう．

5.6　実験手順

被験者は順に部屋に呼ばれ，Treemap Forest若しくはExcelについて20分程度の解説を受ける．次に，対象データに含まれるデータ数や，どのような表・要素が用意されているかの解説を受ける．Excelを用いる被験者には，操作法に関する知識の多寡が実験結果に影響を与えないよう，フィルタ機能の利用方法について教えた上で，ヘルプの使用が認められていることを伝えた．

その後，実験１，実験２の順番で実験を行う．

6　結果と考察

6.1　実験１の結果

実験１の質問／ツール別の解答時間を図８に示す．図中の各点は，質問ごとの被験者の解答時間（縦軸）を表す．また，制限時間を超えた場合，15分としてプロットしている．各被験者の詳細な解答時間と回答失敗数は，表４，５に示す．

表３，４から，Treemap Forestを使ったプロトタイプを利用した被験者群は，Excelを利用した被験者群の38%程度の時間で解答できており，Treemap Forestによって短時間でソフトウェア開発データを参照できることが分かる．

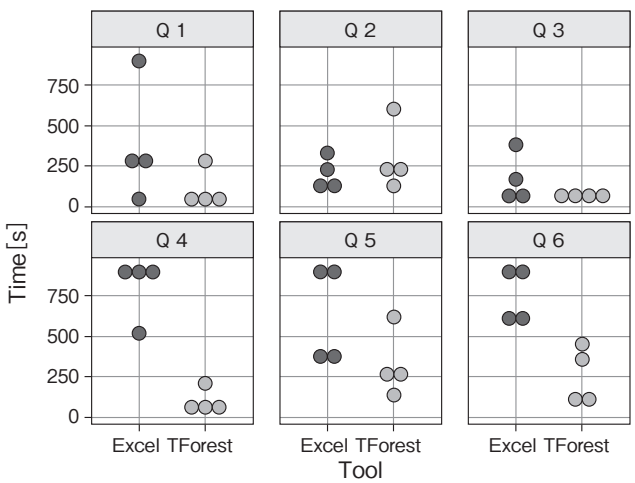

図８　各質問のツール別解答時間

表3　Treemap Forestによる解答時間（分：秒）

	A	B	C	D	平均
Q 1	05 : 21	00 : 43	00 : 28	00 : 46	01 : 49
Q 2	10 : 02	04 : 11	02 : 12	03 : 14	04 : 55
Q 3	01 : 34	00 : 27	00 : 27	00 : 48	00 : 49
Q 4	03 : 29	01 : 00	00 : 59	01 : 04	01 : 38
Q 5	04 : 14	10 : 21	04 : 38	02 : 18	05 : 23
Q 6	05 : 58	01 : 43	01 : 59	07 : 33	04 : 18
合計	30 : 38	18 : 25	10 : 43	15 : 43	18 : 52
回答失敗	0	0	0	0	0

※背景青は最小，グレーは最大.

表4　Excelによる解答時間（分：秒）

	E	F	G	H	平均
Q 1	–	01 : 03	04 : 03	04 : 36	>06 : 10
Q 2	05 : 32	04 : 24	01 : 35	02 : 39	03 : 33
Q 3	06 : 21	01 : 43	00 : 42	02 : 48	02 : 53
Q 4	–	08 : 39	–	–	>13 : 25
Q 5	–	06 : 13	–	06 : 20	>10 : 38
Q 6	–	–	10 : 39	09 : 46	>12 : 36
合計	71 : 53	37 : 02	46 : 59	41 : 09	49 : 16
回答失敗	4	1	2	1	2.25

※背景青は最小，グレーは最大.

表5　Treemap可視化図のみによる解答時間（分：秒）

	E	F	G	H	平均
Q 1	01 : 58	02 : 11	01 : 53	00 : 29	1 : 38
Q 3	03 : 23	01 : 35	01 : 48	01 : 39	1 : 57

※背景青は最小，グレーは最大.

正答数に着目すると，Treemap Forestを利用した全被験者は，すべての質問を時間内に正答していた．一方で，Excelを利用した被験者は全問正解者が居なかった．とくに，単一の表を見れば答えが分かるQ 1からQ 3においては被験者Eの一問不正解を除いて正答であったが，複数の表を閲覧しなくてはならないQ 4からQ 6では全体で7件失敗していた．このことから，Excelの場合，開発データの俯瞰や調査に時間がかかり，場合によっては正しい情報を得られないことが明らかになった．

6.2　実験2の結果

実験2において報告された知見はTreemap Forestで平均2.5件，Excelで2件と，大きな差が見られなかった．生成された知見と，知見を発見するために参照された要素を表6及び7に示す．

システムによって生成される知見の質や，言及される領域が異なるかに着目すると，Treemap Forestでは同一の被験者でも知見2や3のように課題や開発者，コンポーネントといった多様な要素に言及している一方，Excelを利用している被験者は知見2や7など，すべての被験者が単一の要素（課題表）から得られる知見のみを報告していた．

課題表は，実験データ中で最も属性の数が多い表であるため，Excelを利用した群はこの表から有用な知見を得られると感じた可能性がある．こうした分析は，要素間の紐づけにかかる手間を省き，仮説の生成数を上昇させる反面，Excelを利用した被験者にとって，成果物や作業といった要素に紐づいた量的属性，あるいは要素間の関係は見落としやすい部分であることを示唆している．

6.3　考察

実験1の解答時間・正答について質問別に見ると，ほぼすべての質問でExcelよりTreemap Forestの最小・最大・平均

表6　Treemap Forestで生成された知見（項目・表は関連する要素・表の頭文字, 例：モジュール表なら「モ」）

	生成された知見	要素	表
1	10人程度の比較的積極的な開発者によって運営されている	組/作	コ
2	各開発者が特定のモジュールに集中して，役割分担が行われている	組/成/作	コ/モ
3	コミット数が多いが，課題の割り当て数が少ない開発者が居る	組/課/作	課/コ
4	複雑なモジュールには未解決の課題が多い	成/課	モ/課
5	課題の割当が特定の開発者に集中しており，割り当てに問題がある	組/課	課
6	課題数最大のUIモジュール以外では，未解決課題が多い	成/課	モ/課
7	高優先順位な課題は解決済であり，品質が高い	課	課
8	多くの課題を割り当てられ，ほとんど解決済の，熟練した開発者が居る	組/課	課
9	機能拡張に関する課題は優先順位が低い	課	課

表7　Excelで生成された知見（項目・表は関連する要素・表の頭文字, 例：モジュール表なら「モ」）

	生成された知見	要素	表
1	Major課題の半分弱は優先順位が3以下であり，優先順位の割当てに問題がある	課	課
2	不具合修正課題の半分以上が解決済であり，品質向上意識が高い	課	課
3	機能拡張課題は全体の20%しか解決しておらず，やや消極的である	課	課
4	1832件も課題を抱えている開発者がおり，負担がかかっている	課/組	課
5	Coreに関する優先順位3以上の課題は解決済であり，優先順位が上手に働いている	課/成	課
6	特定の3人の開発者は600以上の課題を割り当てられているが，報告課題数が0であり，活動的ではない	課/組	課
7	UIモジュールは課題報告の重複が多い	課	課
8	UIを除くモジュールでは，課題修正が滞っており，開発者が足りていない	課/組	課

解答時間が短いが，唯一Q２においてはExcelのほうが短時間で解答できている．原因として，Q２の解答には各領域のサイズを把握し，最大の要素を見つける必要があるにもかかわらず，解答候補となるファイルが複数あり，Treemap表現による視覚的な比較がうまく働かなかった可能性が考えられる．

このことはTreemapが，全体に対する構成要素の中で，とくに大きな割合を占める特異的な構成要素を発見することに向いている反面，値のよく似た複数の構成要素の厳密な比較に不向きであることに由来する．ただし，本手法は厳密な検証を実施する前に簡易なモデルや仮説を生成することを目的とした探索的データ解析であり，意図的に捨象している部分であるとも言える．

また，本手法の有効性が，Treemapによる表現と，データ間の関連づけやユーザ操作のどちらに由来するかを調査するため，単一の表から生成可能な（データ間の関係を考慮しない）可視化図のみを用いた追加実験を実施した．結果を表５に示す．

実験の結果，Q １では平均１分38秒と，提案手法より素早く解答できることがわかった．また，解答時間の最小値には１秒しか差がない一方，最大値には３分以上の差が見られた．一方，Q ３では平均・最小・最大解答時間共にTreemap Forestのほうが短かった．これらの結果は，Treemap Forestは提示できるデータや可能な操作が多く，ユーザによっては習熟に時間がかかることや，操作に慣れるにつれて素早く情報を提示できる可能性を示唆する．

実験２で生成された知見に注目すると，Treemap Forestを利用した開発者に比べて，Excelを利用した開発者は特定の項目（例えば，UIモジュール）に注目した分析を行い，その厳密な数値について述べる傾向があることが分かった．

また，Treemap Forestで生成された仮説と関係する最大要素数は３，表数は２であった．本制約には，実験の時間的制約や，被験者の習熟度，システムの見やすさなどが影響すると考えられる．

ただし，本実験で用いたデータセットは開発記録のごく一部であり，実環境においては要素数や表数が変動する可能性があり，結果の一般化は難しい．そのため，ツールの可用範囲や，生成される仮説にかかる制約を明確化することは，今後の検討課題となる．

7　おわりに

本論文では，「ソフトウェアプロジェクトトモグラフィ」を構成する要素技術である「ソフトウェア開発データの量的属性の探索的可視化」を実現するため，新たな可視化方式であるTreemap Forestを提案し，その効果についての被験者実験を行った．

オープンソース開発データ（Eclipse JDT）についての質問に解答する実験１の結果，提案手法はExcelと比較して38%程度の時間で質問に解答できることがわかった．一方，Excelでは，制限時間内に解答できないケースも多く，ソフトウェア開発データの探索的データ解析におけるTreemap Forestの有用性が示された．

データを自由に探索し得られた知見を報告する実験２の結果，Treemap Forestでは複数の表が参照された反面，Excelでは課題表のデータしか参照されておらず，Treemap Forestによって広範なデータから知見を集められることが示された．

本研究の制約として，実験の可視化対象データにオープンソースプロジェクトから取得したものを使っていることや，厳密な値の比較に不向きということがある．また，HCEなどほかの探索的データ解析ツールには，データに対して統計処理を行うものが存在するが，Treemap Forestは値の分布を示すのみで，高度な統計解析が行えないことにも留意する必要がある．

ただし，これらの点を考慮したとしても，多面的なデータに対する俯瞰を短時間で実施でき，また要素間の関係に基づく知見を得ることができるTreemap Forestには有用性があると考えられる．

今後の課題として，前述した制約の解決に加え，オープンソースプロジェクト以外の開発データを対象としたTreemap Forestの適用や，学生以外の被験者による評価について検討する余地がある．

謝辞

本研究の一部は，独立行政法人情報処理推進機構（IPA）「2013 年度ソフトウェア工学分野の先導的研究支援事業」の委託に基づいて行われた．

【参考文献】

[IPA] 情報処理推進機構, “製品・システムにおけるソフトウェアの信頼性・安全性等に関する品質説明力強化のための制度構築ガイドライン”（平25-6）.
[IPA2012] 2012 年度ソフトウェア工学分野の先導的研究支援事業「ソフトウェア品質の第三者評価のための基盤技術－ソフトウェアプロジェクトトモグラフィの開発－」成果報告書, http://www.ipa.go.jp/files/000026806.pdf
[IPA2013] 2013 年度ソフトウェア工学分野の先導的研究支援事業「ソフトウェア品質の第三者評価のための基盤技術－ソフトウェアプロジェクトトモグラフィ技術の高度化－」成果報告書, http://www.ipa.go.jp/files/000045268.pdf
[Tukey] J.W.Tukey, “Exploratory Data Analysis,” Addison-Wesley, 1977.
[大平] 大平雅雄, 伊原彰紀, 中野大輔, 松本健一, “ソフトウェア品質の第三者評価における探索的データ解析ツールの利用とその効果：OSSデータを対象とした検証実験”, SEC journal, Vol.9, No.4, 2014.
[Jenifer] Jenifer Tidwell, “Designing Interfaces,” O’Reilly Media, 2011.
[Seo] J. Seo, B. Shneiderman, “Interactively Exploring Hierarchical Clustering Results,” IEEE Computer, Volume 35, Number 7, pp. 80-86, 2002.