一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
SS2021-57(2022-03)

プロジェクトコーパスを用いたソフトウェア概要推測手法の実験的評価

寺川　航平†　陳　思楠†　中村　匡秀†,††

† 神戸大学　〒657–8501 神戸市灘区六甲台町 1–1
†† 理化学研究所・革新知能統合研究センター〒103-0027 東京都中央区日本橋 1-4-1
E-mail: †{odajin,chensinan}@ws.cs.kobe-u.ac.jp, ††masa-n@cs.kobe-u.ac.jp

あらまし　過去に開発されたソフトウェアは時間とともに次第に利用されなくなる．しかし，そうしたソフトウェアのソースコードには再利用可能かつ資産的価値の高い情報も数多く含まれているため，既存ソフトウェアの特徴を把握することで新しいソフトウェアの開発を行う際に活かすことができる．先行研究では，プロジェクトコーパス（ $corpus_{Pj}$ ）を用いた既存システムの概要推測手法を提案し，実現可能性を検証する予備実験を行った．その結果．$corpus_{Pj}$ からはシステムの目的や機能，使用技術に関する推測が可能であると判明した．本稿では，先行研究で検討されていない視点から $corpus_{Pj}$ の有効性を確かめる．検証項目を 3 つ設定し，$corpus_{Pj}$ を用いてシステムのもつ機能，システムの利用技術，システムの概要の推測を行う実験を実施する．実験結果から，プロジェクトの推測は 2 つの要因に左右されることが確認できた．1 つ目は $corpus_{Pj}$ がシステムの持つ情報を正しく反映していること，2 つ目は回答者がコーパス内の単語に関する知識を備えていることである．

キーワード　ソフトウェア開発, 技術的負債, アップサイクリング, コーパス

Experimental Evaluation of Inferring Software Description Using Project Corpus

Kohei TERAKAWA†, Sinan CHEN†, and Masahide NAKAMURA†,††

† Kobe University　Rokkodai-cho 1–1，Nada-ku，Kobe，Hyogo，657–8501 Japan
†† Riken AIP, 1-4-1 Nihon-bashi, Chuo-ku, Tokyo 103-0027
E-mail: †{odajin,chensinan}@ws.cs.kobe-u.ac.jp, ††masa-n@cs.kobe-u.ac.jp

Abstract　Software developed in the past is gradually discontinued over time. However, the source code of such software contains a lot of information that can be reused and has high asset value, so understanding the characteristics of existing software can be utilized when developing new software. In a previous study, we proposed a method for inferring the outline of an existing system using a project corpus ($corpus_{Pj}$), and conducted preliminary experiments to verify its feasibility. The results are as follows. As a result, it was found that $corpus_{Pj}$ can be used to infer the purpose, function, and technology of a system. In this paper, we confirm the validity of $corpus_{Pj}$ from a viewpoint that has not been examined in previous studies. We set three verification items and conduct an experiment in which we use $corpus_{Pj}$ to infer the function of the system, the technology used in the system, and the outline of the system. From the results of the experiment, we confirmed that the guessing of the project depends on two factors: first, the $corpus_{Pj}$ correctly reflects the information of the system, and second, the respondents' knowledge of the words in the corpus.

Key words　Software Development, Technical debt, Upcycling, Corpus

1. はじめに

過去に開発されたソフトウェアは環境の変化や技術的負債 [1] の蓄積などの要因により，次第に利用されなくなる．しかし，そうしたソフトウェアのソースコードには再利用可能かつ資産的価値の高い情報も数多く含まれているため，既存ソフトウェアの特徴を把握することで新しいソフトウェアの開発を行う際に活かすことができる．

既存のソフトウェアプロダクトの再利用を行うため，我々はソフトウェアアップサイクリングという考え方について検討を

This article is a technical report without peer review, and its polished and/or extended version may be published elsewhere.
Copyright ©2022 by IEICE

進めている．ソフトウェアアップサイクリングでは既存プロジェクトの目的・機能を把握し，再利用可能なアイデアや設計を見つけ出すことが重要である．

そこで我々は，既存プロジェクトの概要把握に関する先行研究としてプロジェクトコーパス, $corpus_{Pj}$ を用いたシステムの目的・機能の推測手法を提案した．これを用いて予備実験を行い，有効性を検証した．

しかし，**予備実験には課題点がいくつか残されている．**この先行研究では 3 名の被験者に対して予備実験を行ったが，**さらに被験者を増やして実験を実施**し，提案手法の有効性を検討する必要がある．また，実験方法にも改善の余地が存在する．予備実験により $corpus_{Pj}$ から技術的な手がかりが得られることが判明した．そのため，**使用技術の把握という観点**からもプロジェクトコーパスの有効性を検証する必要がある．

そこで本研究では, 先行研究に加えて新たな視点から $corpus_{Pj}$ の有効性を検証することを目的とする．そのため，以下のように RQ を設定した．

RQ1: $corpus_{Pj}$ によってシステムのもつ機能を把握できるか

RQ2: $corpus_{Pj}$ によってシステムが利用している技術を把握できるか

RQ3: $corpus_{Pj}$ によってシステムの概要を把握できるか

さらに，先行研究と比較してより多くの被験者に対して調査を実施し，結果の分析を行う．

実験に用いる $corpus_{Pj}$ は，先行研究での提案手法を用いて作成する．$corpus_{Pj}$ とは，ソフトウェアの文脈におけるコーパスを指す言葉として，提案手法の中で定義した用語である．コーパスとは一般に言語学の分野で用いられ，テキストや発話を大規模に集めてデータベース化した言語資料を指す．すなわち，ソフトウェアにおけるコーパスとはプロジェクトに含まれる情報から作成される，プロジェクトを要約した資料を意味する．

先行研究における提案手法では，キーアイデアとしてクラス名構成単語に着目した．クラス名はプロジェクトの目的や機能に深く関連すると予想し，クラス名構成単語を抽出する．得られた単語群を抽出元のプロジェクトの性質を反映しているとみなし，これを参考にプロジェクトの概要推測を行う．

実験は以下の 5 つのステップで構成される．

Step1: プロジェクト，被験者の選定

Step2: コーパスの準備

Step3: 質問の設計

Step4: 被験者による回答

Step5: 回答の採点

実験の結果，正答率の高い問題，低い問題が存在することが判明した．分析の結果，プロジェクト概要の把握は大きく分けて 2 つの要因に左右されることが確認できた．

まず，$corpus_{Pj}$ がシステムを正しく要約できているかという点が重要である．さらに，推測には既存技術に関する知識やソフトウェア工学の経験がある程度必要であると判明した．

2. 先行研究: クラス名構成単語に基づく既存プロジェクトの目的・機能の推測 [2]

2.1 着目する課題

ソフトウェアの再利用を行う手法として，ソフトウェアアップサイクリング [3] がある．ソフトウェアアプサイクリングとは既存のソフトウェアプロダクトの設計や実装に着目し，新たに手を加えることで価値あるソフトウェア資産（コード，サービス等）に転換することを指す．よって，ソフトウェアアップサイクリングでは既存プロジェクトの目的・機能を把握し再利用可能なアイデアや設計を見つけ出すことが重要である．

しかし, 組織や会社などの現場において全てのソフトウェアプロダクトが資産的に管理されているわけではない．*README* といったプロダクトにまつわる説明書が欠けている，または不十分である等のケースも存在する．アップサイクリングを促進させるには，説明書の有無に関わらずプロジェクトの概要把握を行うことが課題となる．

2.2 提案手法

先行研究では，既存プロジェクトを README などの説明書に頼らず概要把握を行う手法を提案している．クラス名構成単語から $corpus_{Pj}$ を作成し，これによりシステムの目的や機能の推測を行う．

概要を図 1 で示す．MP3PlayService というプロジェクトについて提案手法を適用した例を示している．STEP1 ではリポジトリマイニングを行い，存在するクラスのクラスファイル名をすべて取得する．STEP2 では，取得したクラス名を単語に分割する処理を行い，得られた単語群を $corpus_{Pj}$ とする．STEP3 では，STEP2 で得られた単語群を辞書順に並べ替え，単語の出現頻度による重みでタグクラウド [4] 画像を生成する．STEP4 では 2 種類で表現した $corpus_{Pj}$ を用いて，元のプロジェクトの目的・機能を推察する．

2.3 プロジェクトコーパス

提案手法では，クラス名分割処理によって得られた単語群を $corpus_{Pj}$ と定義した．コーパスとは一般に言語学の分野で用いられ，テキストや発話を大規模に集めてデータベース化した言語資料を指す．

つまり，$corpus_{Pj}$ とはソフトウェアの文脈におけるコーパスを意味する．具体的には，プロジェクトに含まれるクラス名を収集し単語に分割する．得られた単語群は抽出元のプロジェクトの性質を反映しているとみなし，これを参考にプロジェクトの概要を推測できれば，README が存在しないプロジェクトについても概要把握が容易に行える．

2.4 予備実験の結果

提案手法を用いて，三名の被験者に対して予備実験を行った．全員が半数以上の問題に正答し，クラス名構成単語がプロジェクト概要の把握に一定の効果を持つことが明らかとなった．

しかし，課題点もいくつか存在する．この先行研究では 3 名の被験者に対して予備実験を行った．さらに被験者を増やして実験を実施し，提案手法の有効性を検討する必要がある．

また，予備実験の考察により，$corpus_{Pj}$ から技術的な手がか

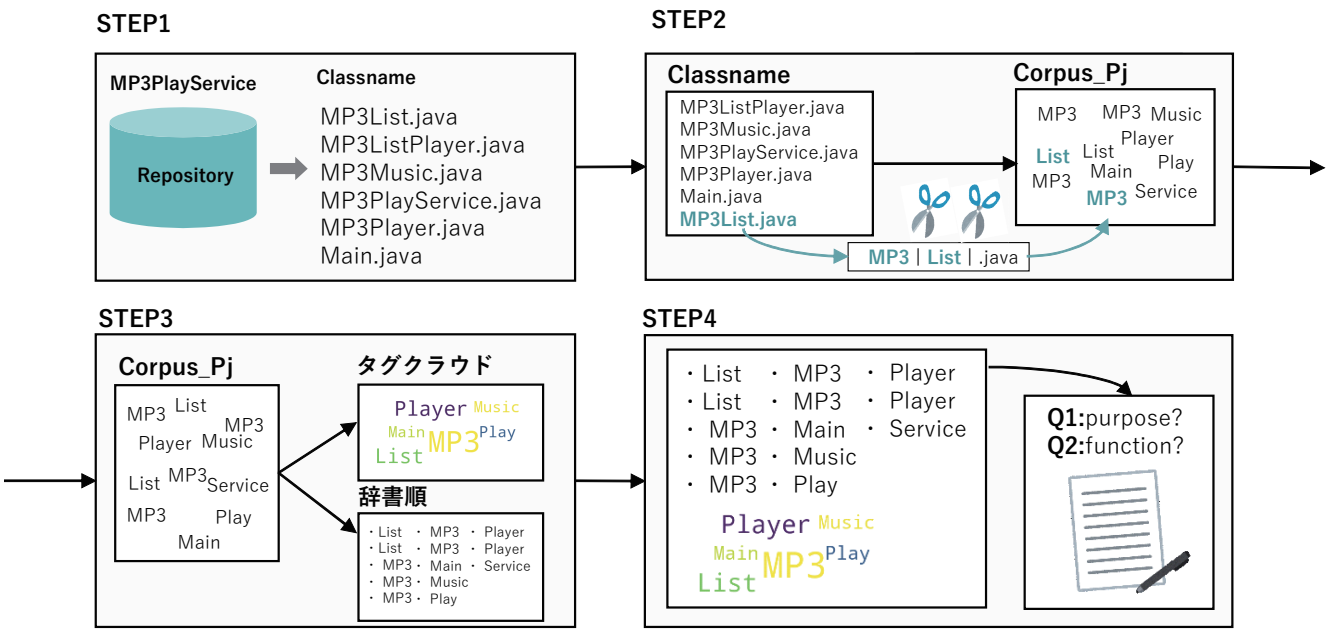


図 1　先行研究での提案手法の流れ

りが得られることが判明した．回答者によっては機能と使用技術に関する記述が混在しており，どの程度技術的な事項を把握しているのかを切り分けることができなかった．そのため，使用技術の把握という観点からも $corpus_{Pj}$ の有効性を検証する必要がある．

3. 評 価 実 験

3.1 実験の目的

本研究での目的は，先行研究で検討されていない視点から提案手法の有効性を検証することである．そのため，以下のように RQ を設定する

RQ1: $corpus_{Pj}$ によって，システムのもつ機能を把握できるか

RQ2: $corpus_{Pj}$ によって，システムが利用している技術を把握できるか

RQ3: $corpus_{Pj}$ によって，システムの概要を把握できるか

3.2 実験の概要

RQ への回答を行うため，実験を行う．実験は以下の 5 つのステップで構成される．

Step1 プロジェクト，被験者の選定: 自研究室で管理する 154 個のプロジェクトを対象とする．

Step2 コーパスの作成: 先行研究における提案手法により，$corpus_{Pj}$ を作成する．

Step3 質問の設計: RQ への回答のために，3 つの質問を設定する．

Step4 被験者による回答: google フォームを用いて，自然言語で回答を入力してもらう

Step5 回答の採点: 各プロジェクトを理解している人が採点を行い，評価を行う．

3.3 STEP1: プロジェクト，被験者の選定

本実験に使用するプロジェクトは，研究室で管理する Gitlab に含まれている Java プロジェクトを利用する．

ここから 10 個のプロジェクトを無作為に抽出し，これを実験用データとする．実験に用いたプロジェクトの詳細を，表 1 に示す．本実験では，著者の所属する研究室に所属する 12 名の学生を対象に実験を行う．

3.4 STEP2: コーパスの作成

先行研究の提案手法により，$corpus_{Pj}$ を作成する．実験用の 10 個のプロジェクトに対してリポジトリマイニングを行い，クラス名構成単語を抽出する．具体的には GitlabAPI を利用してマイニングを行い，得られたデータを Python を用いてクラス名分割処理を行う．これにより，10 個のプロジェクトのそれぞれについて $corpus_{Pj}$ を作成する．

得られた $corpus_{Pj}$ を辞書順に並び替え，単語の出現頻度を重みとしてタグクラウド画像を作成する．表 2 に，作成した $corpus_{Pj}$ とタグクラウド画像を示す．

3.5 STEP3: 質問の設計

プロジェクトへの理解度を図るため，以下の 3 つの質問を設計する．

(Q1)：このシステムにはどんな機能があるか，思いつくだけ箇条書きで書いてください

(Q2)：このシステムはどのような技術や仕組みで動作しているか，思いつくだけ箇条書きで書いてください

(Q3)：$corpus_{Pj}$ から推察される，システムの概要を 1〜2 行程度で記述してください．

Q1, Q2, Q3 は RQ1, RQ2, RQ3 に対応している．各質問に対する分析を第 4 章で行い，これを RQ への回答とする．

3.6 STEP4: 被験者による回答

作成した $corpus_{Pj}$ を google フォーム上で提示し，被験者が

表 1　実験に用いたプロジェクトの詳細

問題番号	概要	機能	使用技術
1	**NewLINEAgentService** LINE Botを使って，LINEメッセージを送信するサービス	・LINEメッセージを送信する ・ユーザの友達イベントを取得し，プロフィールを登録する ・メッセージのログを保存する	Webhook, LINE Messaging API, Jersey, MongoDB, NextCloud
2	**KobeDemographicAPI** 神戸市の各町丁目の人口データを取得するAPI	・細粒度な人口データを取得する ・町丁目から町コードを取得する	Jersey MySQL
3	**LifeActivityMailer** 環境センサに変化があった際にメールで通知するサービス	・環境センサボックスの変化点検知サービスを見張る ・変化が起こった際に，センサの所有者（管理者に）メールで通知する	JavaMail センサボックス変化点検知サービス(SBCPDService)(SBCPD = SensorBox Change Point Detection)
4	**BLEAdapter** BLE（Bluetooth Low Energy）のデータを整形し，すれ違いフレームワークに適合させるアダプタ すれ違いフレームワークとは，様々な近接通信技術を駆使して，ユーザ同士のすれ違いを検知する枠組み 本実装は，BLEによるすれ違いをフレームワークに適合させるアダプタ	・BLEの生データを解析し，標準的なIDに変換して，すれ違いDBにポストする ・与えられた場所を，すれ違いDBにポストする	Java Webサービス XML
5	**enishi_mock** Enishi（縁，エニシ）のモバイル・アプリケーションのモック Enishiとはすれ違いフレームワークを活用して，街頭で求人マッチングを行うサービス 求人側と求職側の好みをクラウドで管理し，好みが近しい両者が街頭で近接した際に，通知を送りマッチン行う	・Androidモバイル・アプリケーションのモック モックなので画面だけ．何も行わない	AndroidアプリSDK
6	**ENISHI** Enishiのバックエンドサービス モック用に機能がかなり絞られている	・ユーザ管理機能 ・求人情報取得 ・求職者情報取得	Jersey Firebase MySQL
7	**MP3PlayService** 与えられたURLに基づき，MP3を再生するサービス 再生リストによる再生も可能	・MP3を再生する ・MP3のリストを再生する ・再生を停止する ・再生中のURLを取得する ・次の曲へ進める	Apache Axis2 Webサービス JLayer(MP3再生ライブラリ)
8	**virtual-care-giver** ヴァーチャルエージェントを用いてケアシナリオを実行するWebアプリケーション 声掛け，忘れ物リマインド，予定管理，YouTube再生等を行う	・ケアシナリオの実行 ・声掛け・問診 ・Youtubeの再生 ・利用者情報の管理	Java WebSocket REST Webサービス JavaScript
9	**thin-cas** ECAルールによるコンテキストアウェアサービス(CAS)を実行する軽量なコマンドラインアプリケーション	・ECAルールの登録 ・ECAルールの評価．アクションの実行	Pure Java
10	**StateCacheService** アプリの状態をキャッシュするWebサービス 任意の状態・情報を，識別子（キー）とJson（バリュー）のペアで保持	・状態の記憶 (set state) ・状態の取得 (get state) ・全識別子の取得 (get all IDs) ・全状態の取得 (get all states)	Jersey

回答を入力する．$corpus_{Pj}$ については Step2 で用意した 2 通りの表現を用いる．回答に要した時間を計測するため，回答開始時，回答終了時の時間を記録してもらう．

3.7　STEP5: 回答の採点

Step4 で得られた回答への採点を行う．実験対象のプロジェクトの詳細について把握している者が採点を行う．採点基準を表 3 に示す．ここでは以下の 5 段階によって大まかに評価を行う．

4.　実験結果と考察

4.1　結果の概要

実験の結果，12 人の被験者の全員から回答を得た．ここで，本実験における正解率を定義する．◎，○の採点は概要把握に成功したとみなし，これらの 2 つの合計が占める割合を計算する．以下に計算式を示す．

$$\text{正解率 (\%)} = (\text{◎と○の合計}) \div (\text{総回答数}) \times 100 \qquad (1)$$

各問題の 3 つの質問 Q1, Q2, Q3 に対する正解率を以下の表 4 に示す．問題 7 が Q1 Q3 を通して最も正解率が高く，反対に問題 5 は正解率がすべて 0 となった．

4.2　RQ1 に対する分析

図 2 に Q1 に対する各問題の採点結果を表す．12 名が◎，○，△，×，？の 5 段階の評価を行ったため，各問題につき合計で 12 個の評価が回答として得られた．このグラフは各問題における採点結果の内訳を示しており，表示が濃いほど良い評価を表している．グラフにおいて問題 3, 7 は◎、○が 8 割以上を占めており，正答率が高いとわかる．一方，問題 4, 5 については正答率が低く，採点結果は×，？が 7 割以上を占めている．.

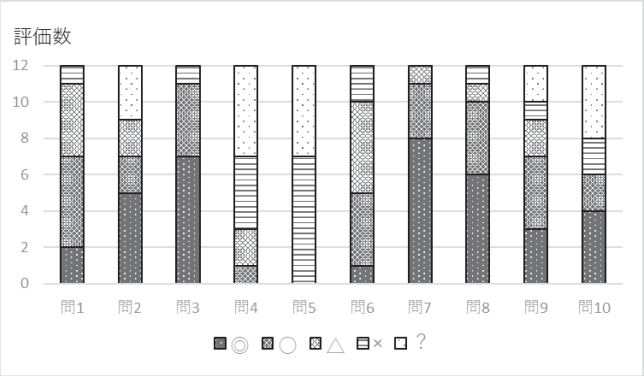


図 2　Q1 に対する採点結果の内訳

正解率が高くなった原因を考察するため，問題 3 の LifeActivityMailer に着目する．概要に示すように，プロジェクトはセンサーボックスの値を検知し，変化が起こった際にセンサー

表 2　本実験で作成したコーパス

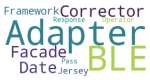
問題番号	$corpus_{Pj}$	image
1	Beacon Cloud Content Controller DAO DAO DAO DAO Date Drive Event Event Event Event Event Event Event Event Event Event Follow Handler Handler Handler Handler Handler Handler Handler Handler Http Http Info Info Join LINE LINE LINE LINE LINE LINE LINEBOT LINEBOT LINEBOT Leave Listener Listener Listener Log Log Message Message Message Method Mongo Next Postback Profile Profile Response Service Test Time Unfollow Unknown User User Utils Utils Utils Utils Webhook Webhook Wrapper	
2	1 45 5 Area Connector Demography Demography Error Gson Hello List List Message My Population Rest Rest SQL Test Town Town Town Town bin connector jar mysql	
3	Box Detection Listener Location Mailer Mailer Mailer Mailer Manager Model Notify SBCPD SBCPD Score Sensor Service Service Solver Stream Task	
4	Adapter BLE By Corrector Date Facade Framework Jersey Operator Pass Response	
5	Activity Activity Adapter App Application Base Base Component Example Example Fragment Fragment Instrumented Item List Main Main Module My My Sample Test Test Unit Util	
6	1 1 39 39 5 5 Account Account Account Account Activity Activity Adapter Adapter App Application Application Base Base Client Component Controller Controller Controller Detector Enishi Example Facade Facade Firebase Fragment Fragment Fragment Fragment Fragment Fragment Info Info Info Info Info Info Info Info Info Info Info Info Info Info Info Info Info Info Item Jersey List List List List List List List List Main Main Map Matching Messaging Module My My My My Mysql News Offerer Offerer Offerer Offerer Offerer Offerer Parser Passby Passby Passby Sample Seeker Seeker Seeker Seeker Seeker Seeker Service Service Service Test Test Unit User User User User User Util View Web Xml bin bin connector connector jar jar mysql mysql	
7	List List MP3 MP3 MP3 MP3 MP3 Main Music Play Player Player Service	
8	Admin Agent Answer Bean Bean Bean Button Care Care Care Check Controller DB Display Giver Helper Http Image Image List Main Manager Manager Page Page Patient Question Question Question Queue Resource Result Router Schedule Schedule Schedule Schedule Search Search Server Server Single Subtitle Test Utils Virtual Virtual WS Youtube	
9	Client Context Context Context Context Context DAO DAO DAO DAO DAO DAO Endpoint Endpoint Endpoint Endpoint Engine Engine Engine Evaluation Evaluator Evaluator Event Event Facade Facade Facade Facade Facade Facade Helper Http Http Http Main Mock Mock Model Model Model Morphia Observer Resource Router Rule Rule Rule Rule Rule Rule Test Test Test Test Test Test Test Util	
10	Cache State	

表 3　実験における評価基準

評価	基準
◎	正解
○	概ね正解
△	一部正解，一部不正解含む
x	間違っている
?	わからない

の所有者にメールで通知するという 2 つの機能をもつ．これらについて 12 人中 7 名が正しく記述しており，◎の採点結果となった．ここで，表 2 中で示した問題 4 に関するコーパスを以下に列挙する．

- Box Detection Listener Location Mailer Mailer Mailer Mailer Manager Model Notify SBCPD SBCPD Score Sensor Service Service Solver Stream Task

ここには，LifeActivityMailer の持つ 2 つの機能に関連する単語が含まれていることがわかる．以下に機能と関連する単語を示す．

（機能 1: センサーボックスの値を検知）Sensor, Box, Score, Detection

（機能 2: センサーの所有者にメールで通知）Mailer, Notify

一般的に，センサーがなにかの値を検知するという振る舞いは明らかであるため，回答者の多くが $corpus_{Pj}$ から Sensor と Score，Detection といった単語に着目したと考えられる．以上より，回答者が意味的に関連性のある単語同士を結びつけることによって，システムのもつ機能を推測していることが判明した．

正解率の低かった原因を考察するため，問題 5 に着目する．表 1 の示すように，このプロジェクトの実態は別プロジェクトのための画面設計である．Android SDK によってプロジェクトが作成されたため，コーパスには Activity, Fragment といった Android SDK 特有のコンポーネント名が含まれている．これらは Android のアプリ開発で用いられる概念であり，具体的な動作とは関わりがない．以上より，コーパス内に予測に必要な情報が十分に含まれていない場合，機能の推測は困難であることが明らかとなった．

4.3　RQ2 に対する分析

表 4　正　答　率

問題番号	Q1	Q2	Q3
1	58%	83%	75%
2	58%	92%	58%
3	92%	67%	67%
4	8%	42%	0%
5	0%	0%	0%
6	42%	83%	25%
7	92%	75%	100%
8	83%	83%	58%
9	58%	42%	42%
10	50%	17%	50%

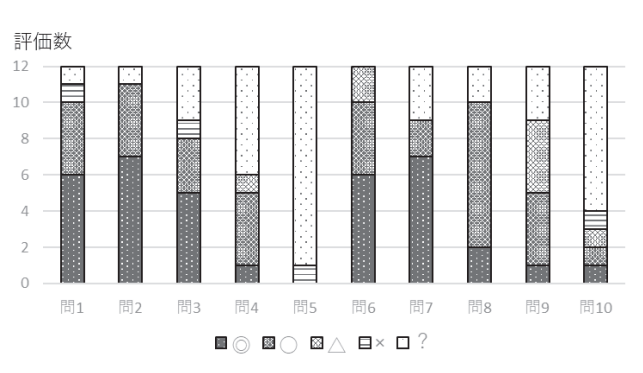


図 3　Q2 に対する採点結果の内訳

図 3 は，Q2 に対する各問題の採点結果を表している．問題 1, 2, 6, 7, 8 については採点結果の 7 割以上が○、◎を占めた

一方，問題 4,5,10 については正答率が低く，×または？が目立った

正解となった回答では，コーパス内の単語をそのまま記述する例が多く見られた．具体的に問題 6 を例に上げる．表 1 に示すように，問題 6 の機能はユーザ情報を DB とやりとりすることである．これは MySQL を用いて実現しており，コーパス内には MySQL という単語が含まれていることが確認できる．回答では 12 人中 10 人が使用技術に MySQL を記述し，同時に Q1 への回答として CRUD にまつわる記述を行っていた．MySQL という単語の存在から，システムに DB への CRUD を行う機能が存在すると解釈したと考えられる．以上より，コーパス内に使用技術に関わる単語が存在する場合，回答者は技術用語を用いて規定の動作を推測することが分かった．

一方で，回答者によって解釈が分かれた例も存在する．問題 4 では，半分以上の回答で？（わからない，無記名）という回答であった．表 1 に示すように，問題 4 の機能は BLE で得たデータを DB にポストすることである．表 2 中の問題 4 の $corpus_{Pj}$ を確認すると，コーパス内に BLE という単語が含まれていることがわかる．BLE とは、Bluetooth の拡張仕様の一つである．極低電力で通信が可能であり，これを生かして各種センサーに用いられる．しかし，BLE を使用技術として記述し，BLE のセンサーとしての働きを読み取った例は 3 名しかいなかった．以上より，使用技術から読み取れる情報量は回答者によって異なることが判明した．

4.4 RQ3 に対する分析

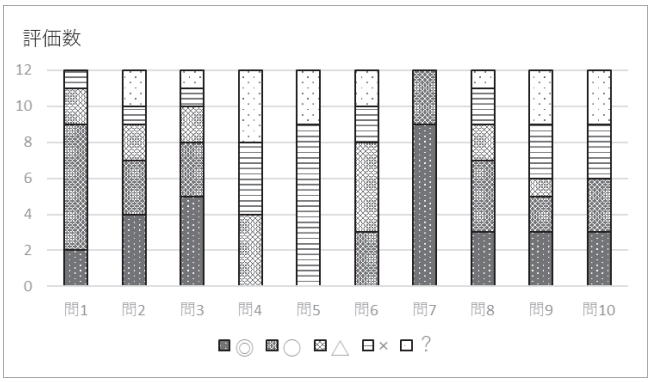


図 4　Q3 に対する採点結果の内訳

図 4 は，Q3 に対する各問題の採点結果を表している．RQ1 のグラフと比較して，正答率の高い問題，低い問題が共通している．しかし，問題 3，問題 6，問題 8，問題 9 に関しては正答率が Q1 と比較して低下している事がわかる．

Q1 と比較して最も正答率の下がった問題 8 の virtual-caregiver に着目して，その原因を考察する．表 1 に示すように，このプロジェクトは様々な種類のサービスを備えており，それらをヴァーチャルエージェントが提供するというシステムである．12 人すべての回答者はシステムの備える機能である動画の再生，質問を通した会話などのいずれかを記述していた．特に，スケジュール管理を行う機能に関しては 12 人中 11 人が把握していた．一方で，Q3 においてヴァーチャルエージェントが様々な種類のケアを提供するという概要を正しく把握した回答は 12 人中 3 つにとどまった．不正解となった回答では，動画再生やスケジュール管理のシステムであるという回答が多い．このように，一部機能をシステム全体の主な概要として捉えている例が多く見られた．以上より，概要把握のためにはシステムがもつ複数の機能の把握と，それらを要約してシステム全体の概要を推測する必要があると明らかになった．

4.5 全体の考察

RQ1, 2, 3 に対する分析より，プロジェクトの推測は 2 つの要因に左右されることが確認できた．1 つ目は，$corpus_{Pj}$ がシステムの持つ情報をどれだけ正しく反映しているかという点である．RQ1 の分析では，$corpus_{Pj}$ にプロジェクトの機能に関わる情報が欠如している例を挙げた．$corpus_{Pj}$ はクラス名構成単語から作成するため，抽出元のクラス名はシステムの特徴を表している必要がある．

2 つ目は，回答者がコーパス内の単語に関する知識の有無である．RQ2 の分析では，コーパス内に使用技術に関わる単語が含まれる例を確認した．こうした既存技術を正しく解釈するためには回答者の知識が必要であり，その結果単語から得られる情報量は各人で異なる，また，RQ3 ではシステムの機能や使用技術の把握が必ずしもプロジェクト全体の正しい概要推測につながらないことを確認した．システムは様々な機能を備えるが，それらを総合してシステム全体としては何を目的として動作するのかを推測する必要がある．そして，与えられた情報から正しく推測を働かせるためには．ソフトウェア工学の知識や経験が必要である．

4.6 今後の課題

今回の提案手法に関する課題として，クラス内に含まれる情報を含めてコーパスを構造化することが挙げられる．一つのクラスの中にどれだけの機能や役割を持たせるかは開発者によって異なる．複数のメソッドを持つクラスに対して提案手法を用いた場合，そのクラスの持つ機能や目的を取りこぼしてしまう恐れがある．よって，ファイル内部のメソッド名・メンバ変数などを取得することでクラスの保つ役割をより正確に $corpus_{Pj}$ へ反映できると期待される．

また，今回の実験では回答者ごとの性質の違いを考慮しなかった．RQ に対する分析で，$corpus_{Pj}$ を用いた推測には回答者の知識や経験が影響することが判明した．今後は回答者の経験や知識の有無に加え，着目した単語や読み取った情報などを調査して考察に取り入れたい．

5. ま と め

本研究では，先行研究における提案手法の有効性を詳細に分析するための実験を行った．

具体的には，質問項目を再考し，$corpus_{Pj}$ によって得られる情報をプロジェクトの機能・使用技術・概要の 3 つの軸から評価した．さらに，先行研究と比較してより多くの被験者に対して調査を実施し，結果の分析を行った．

実験の結果，12 人から回答を得た．これにより，プロジェクト概要の把握は $corpus_{Pj}$ がシステムを正しく要約できている

か，回答者が推測できるだけの知識を持つかの 2 点に左右されることが判明した．

今後の課題としては，クラス内に含まれるメソッド名やメンバ変数を抽出し，より多くの情報をコーパスに加えて構造化することを考えている．また，実験により問題の正誤には回答者にも要因があるとわかった．よって，推測に用いた情報や手順を記録することで， 回答者の基礎知識が結果に及ぼす影響を調査したい．

謝辞 本研究の一部は JSPS 科研費 JP19H01138, JP18H03242, JP18H03342, JP19H04154, JP19K02973, JP20K11059, JP20H04014, JP20H05706 および，立石科学技術振興財団の研究助成を受けて行われている．

文　献

[1] P. Kruchten, R.L. Nord, and I. Ozkaya, “Technical debt: From metaphor to theory and practice,” Ieee software, vol.29, no.6, pp.18–21, 2012.

[2] 寺川航平，陳　思楠，中村匡秀，“クラス名構成単語に基づく既存プロジェクトの目 的・機能の推測手法に関する予備的検討，” 電子情報通信学会技術研究報告，第 121 巻，pp.1–6，Jan. 2022. 長崎県建設総合会館.

[3] B. Bridgens, M. Powell, G. Farmer, C. Walsh, E. Reed, M. Royapoor, P. Gosling, J. Hall, and O. Heidrich, “Creative upcycling: Reconnecting people, materials and place through making,” Journal of Cleaner Production, vol.189, pp.145–154, 2018.

[4] S. Bateman, C. Gutwin, and M. Nacenta, “Seeing things in the clouds: the effect of visual features on tag cloud selections,” Proceedings of the nineteenth ACM conference on Hypertext and hypermedia, pp.193–202, 2008.