

Impact of Opportunistic Reuse Practices to Technical Debt

Rafael Capilla , Tommi Mikkonen , Carlos Carrillo
Francesca Arcelli Fontana, Ilaria Pigazzini , Valentina Lenarduzzi

2021 IEEE/ACM International Conference on Technical Debt (TechDebt)

紹介者: 西山 知弥

Opportunistic Reuse

- 機会的再利用, 日和見的再利用
- GitHub, Bitbucket, Gitlabなどの主要なリポジトリで再利用可能なコンポーネントを探す再利用のこと

*日和見的・・・状況に応じて自分に利益のある行動を選択する

背景

- 技術的負債の推定に影響を与える要因の一つに,サードパーティ製コンポーネントの再利用や既存システムへの統合がある
- 近年,システムの的な再利用ではなく,機会的な再利用が増加している
- ソフトウェアを機会的に再利用することは,品質の低下を招きTDを誘発する可能性があるが,TDにおける機会的な再利用の影響を調査した研究はない

目的・アプローチ

- 目的

機会的再利用がソフトウェアアーキテクチャにどのような影響を与えるのかを調査する

- アプローチ

SonarQube[1] と Arcan [9] ツールを使用して, 3 種類のアプリケーションのコードアーキテクチャの負債比率を分析

[1] <https://docs.sonarqube.org/latest/user-guide/metric-definitions/>

[9] F. Arcelli Fontana, I. Pigazzini, R. Roveda, D. A. Tamburri, M. Zanoni, and E. D. Nitto, "Arcan: A tool for architectural smells detection," in *Int'l Conf. Software Architecture (ICSA)*, 2017, pp. 282–285.

リサーチクエスチョン

RQ1 : 再利用されたコンポーネントの統合において, 最も複雑な問題は?

RQ2 : 再利用可能な資産を統合した後, コードデットはどのように影響を受けるか?

RQ3 : オープンソースソフトウェアの再利用によって, アーキテクチャ負債はどのように影響を受けるか?

調査するプロジェクト

BikeApp ・ ・ ・ マドリッド工科大学(UPM)で2020年2月～5月にかけて開発された自転車レンタルシステム

TEAMMATES 4 ・ ・ ・ 学生のチームプロジェクトにおける, 機密性の高い相互評価を管理するための無料のオンラインツール

Kurki5 ・ ・ ・ 講座に参加している学生の管理, 講座の達成度の管理, 成績などの情報を提供するために必要な操作を行うWebアプリケーション

データの収集と分析

- (1). 3つのプロジェクトについて SonarQube 8.5 を実行し, 技術的負債比率, **code smell** や問題の数を収集
- (2). Arcan を使って, アーキテクチャ負債指数 (ADI) を計算し, アーキテクチャの**smell**を検出
- (3). 拡張するための新機能を 4つのリポジトリで検索. 検索と統合の労力は, 著者のうち2人が手動で計算した.

分析結果

$$TD\ ratio = \frac{TD\ cost}{(Cost\ to\ develop\ 1\ line\ of\ code * Number\ of\ lines\ of\ code)}$$

(1) プロジェクトのチームが潜在的な顧客の要求に基づいてそのプロジェクトの機能を拡張したいというシナリオをシミュレート

(2) 新しい要望のリストを作成

(3) それらをリポジトリで検索

AS . . . アーキテクチャsmell

ADI . . . アーキテクチャ負債指数

q(ADI) . . . ADIの定量化

Table I
RESULTS FROM SONARQUBE AND ARCAN BEFORE REUSE

SonarQube				Arcan		
Case studies	TD ratio	Smells	Issues	AS	ADI	q(ADI)
BikeApp	5.2	248	253	0	-	-
TEAMMATES	0.6	2087	2905	7	8.0	5
Kurki	1.0	504	536	3	5.0	5

Table II
NUMBER OF REUSABLE COMPONENTS FOUND IN OPEN-SOURCE REPOSITORIES

Case studies		Maven	GitHub	SourceForge	GitLab
BikeApp	Encode data	115	170	9	13
	Display geolocation	7867	92	5	0
	Calendar	1648	111	8	2
TEAMMATES	Voting system	17	14	92	9
	Event manager	561	80	25	78
Kurki	Event manager	561	80	25	78

(i) サーバーとクライアント間のデータをエンコードする

(ii) 自転車の地理的位置を特定する新機能を提供する

(iii) カレンダーを搭載する

コンポーネント検索

(1)コンポーネントの機能の説明

(2)ソフトウェアと再利用コンポーネントのライセンスの互換性

(3)ソースコードへのアクセス

(4)再利用元のバージョンと既存プロジェクトの互換性

(5)コンポーネントのインポートおよび設定の機能

(6)見つかったコンポーネントの使用方法を説明するJavadocファイルの存在に基づいて検索

Table III
REUSABLE COMPONENTS SELECTED IN OPEN-SOURCE REPOSITORIES

Case studies	Maven	GitHub	SourceForge	GitLab
BikeApp	Base64	—	Base64	—
	GoogleMaps JDatePicker	GoogleMaps JDatePicker	— —	— —
TEAMMATES	Voting- Reward	Voting-Reward, Voting-System	—	—
	—	FullCalendar	—	—
Kurki	—	FullCalendar	—	—

Table IV
SEARCH AND INTEGRATION EFFORTS OF THE REUSED ASSETS (HOURS)

Case studies	assets	Search effort	Integration effort
BikeApp	Base64	4.5	1.7
	GoogleMaps	5	5.5
	JDatePicker	6	1.1
TEAMMATES	VotingSystem	6.5	1.4
	FullCalendar	6	2
Kurki	FullCalendar	0	2.2

Bike Appの構造

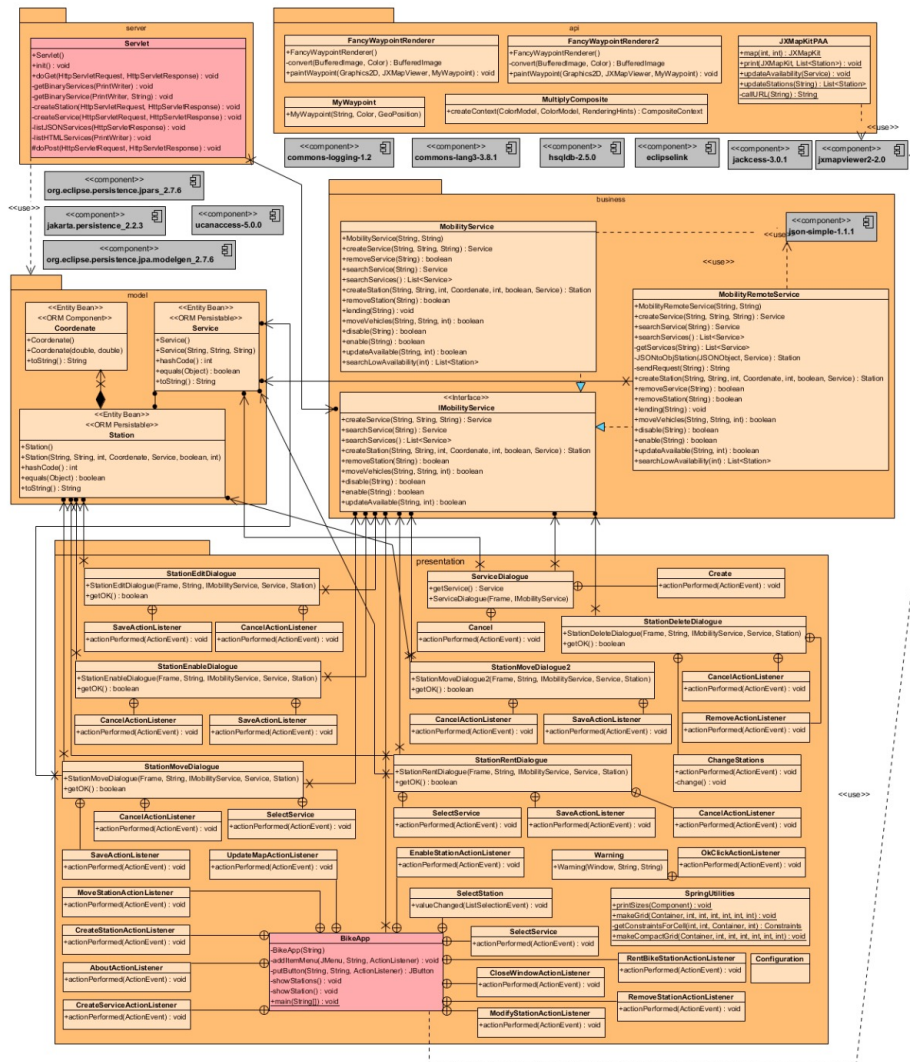


Figure 1. Reversed architecture of the BikeApp software

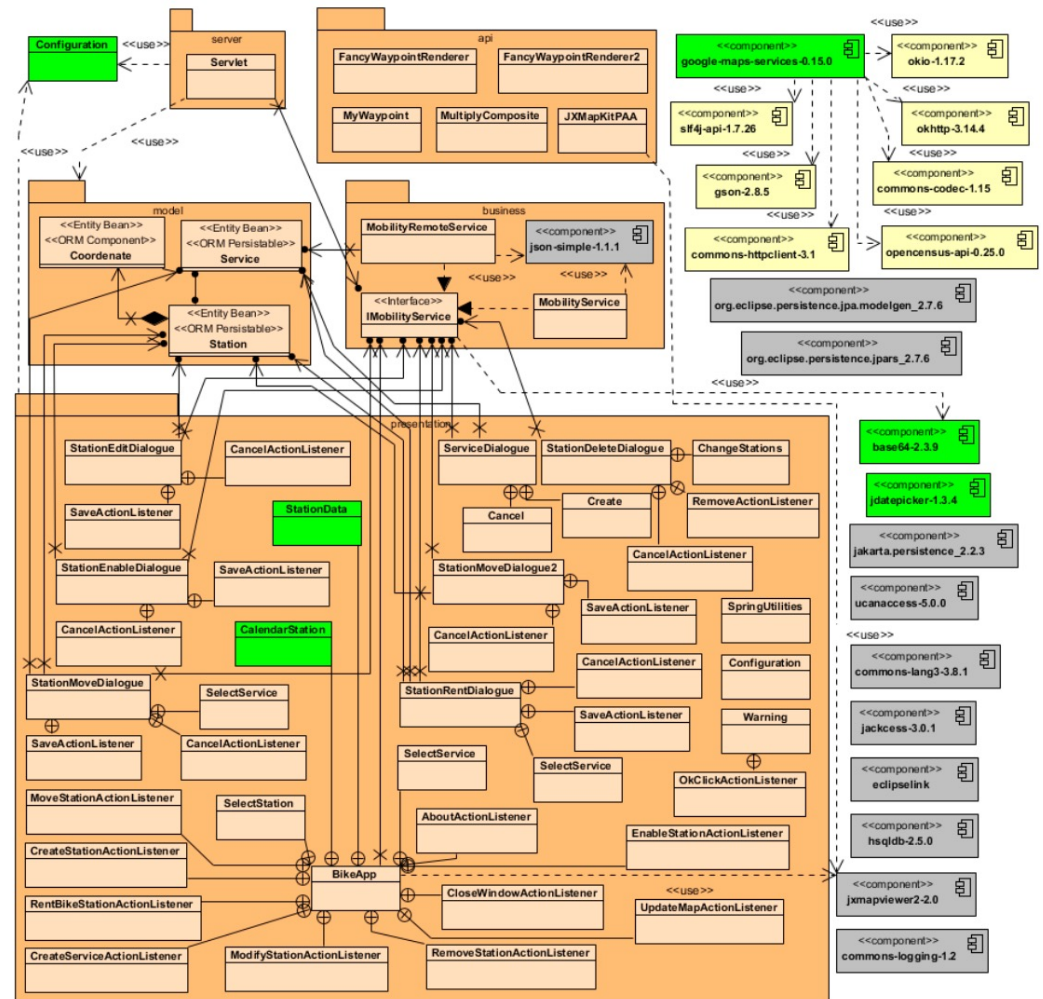


Figure 2. Final version of the reversed architecture of the BikeApp software

RQ1:再利用されたコンポーネントの統合において,最も複雑な問題は?

- 対象となるプロジェクトのどの要素を修正する必要があるかを特定すること
- ターゲットプロジェクトで使用されているアーキテクスタイルとテクノロジーに応じて,適切なコンポーネントを探すこと

RQ2:再利用可能な資産を統合した後, コードデットはどのように影響を受けるか?

- Google Mapsのような大規模なコンポーネントを再利用した場合, TD比率が減少していた
- コンポーネントのサイズが小さい場合は, TD比率はほとんど変化しない
- ほとんどのケースでは, smellやIssuesの数が増加
- 1つのケース(投票システムコンポーネント)では, Issuesの数が少し減少

Table I
RESULTS FROM SONARQUBE AND ARCAN BEFORE REUSE

Case studies	SonarQube			Arcan		
	TD ratio	Smells	Issues	AS	ADI	q(ADI)
BikeApp	5.2	248	253	0	-	-
TEAMMATES	0.6	2087	2905	7	8.0	5
Kurki	1.0	504	536	3	5.0	5



Table V
TECHNICAL DEBT AND ARCHITECTURAL DEBT RATIOS AFTER REUSE

Case studies	assets	SonarQube			Arcan		
		TD ratio	Smells	Issues	AS	ADI	q(ADI)
BikeApp	Base64	4.5	365	383	30	1.0	3
	GoogleMaps	1.7	3494	3701	64	11.0	5
	JDatePicker	1.7	3551	3759	66	10.0	5
TEAMMATES	VotingSystem	0.6	2136	2977	7	12.0	5
	FullCalendar	0.6	2136	2979	7	12.0	5
Kurki	FullCalendar	0.6	735	678	3	5.0	5

RQ3:オープンソースソフトウェアの再利用によって、アーキテクチャ負債はどのように影響を受けるか？

- 表Vの通り, 再利用がアーキテクチャのsmellや負債に影響を与えることがわかった
- アーキテクチャの負債は再利用後に増加.
特に大規模なコンポーネントを再利用した場合に増加しました.
- **TEAMMATES**の場合, 再利用後もASの数は変わらず, 負債も増加

まとめ

- 異なるリポジトリでソフトウェアコンポーネントを機会的に再利用することが, TDに与える影響を初めて検証

今後の課題

- より多くのプロジェクトやリポジトリで評価
- Pythonなどの他のプログラミング言語で実装されたプログラムと比較すること
- 数だけではなく、新しいsmellや問題の質や深刻さについても調査する必要がある

所感

- Opportunistic Reuse という言葉を初めて知った
- 大規模なコンポーネントを再利用するとTD比率が小さくなるのは,意外であった