

プロセスマイニングとソースコード解析を用いた マイクロサービス分割

横井 一輝^{1,a)}

概要：モノリシックシステムからマイクロサービスアーキテクチャへ移行する際、マイクロサービスとして分割する境界の特定から始める。この分割境界を特定する作業は難しいため、この作業を支援する手法が提案されている。しかし、従来手法ではビジネスとシステムの両方の観点を考慮した分割境界の特定ができていない。マイクロサービスの分割境界の特定には、ビジネスとシステムの両方の観点が重要である。そこで本研究では、ビジネスとシステムの両方を考慮に入れた分割境界の特定手法を提案する。具体的には、プロセスマイニングで実行ログを解析し、業務プロセスをグラフ化する。次にソースコード解析でメソッド呼び出しとデータベースアクセスを解析し、プログラムの依存関係をグラフ化する。最後にそれら2つのグラフを用いた協調的クラスタリングにより、マイクロサービスの分割境界を特定する。

キーワード：マイクロサービス分割、マイクロサービス移行、プロセスマイニング、ソースコード解析

1. はじめに

モノリシックシステムとはすべてのプログラムが単一のシステムにまとめられたアーキテクチャである。モノリシックシステムは機能同士の疎結合性が意識されておらず、影響範囲が不明瞭な場合が多い。その結果システムの保守性が低下し、影響確認にかかる作業コストの肥大化につながる。したがって、機能同士の疎結合に改善することは、保守性向上のために重要である。

システムを疎結合に保つためのアーキテクチャとして、マイクロサービスアーキテクチャがある。マイクロサービスアーキテクチャとは、比較的小規模で独立した複数のサービスで構成されるアーキテクチャである。モノリシックシステムをマイクロサービスアーキテクチャに移行することで、機能間が疎結合になり保守性が向上する。したがって、保守性向上を動機としたマイクロサービス移行に近年注目が集まっている。

モノリシックシステムをマイクロサービスアーキテクチャに移行する際、マイクロサービス分割と呼ばれる分割境界を特定する作業からはじめる。マイクロサービス分割の際、ビジネスとシステムの両方の観点が重要になる。ビジネスの変化を簡単かつ迅速に反映するため、システム観点だけでなくビジネス観点でも疎結合にする必要がある。

したがってビジネスとシステムの両方の理解が求められるため、マイクロサービス分割はマイクロサービス移行の中でも難しい。そのためマイクロサービス分割は専門家の経験則によって手作業で実行されることが多く、マイクロサービス分割を支援する手法を求める開発者は多い。

既存研究にて、マイクロサービス分割を支援する手法はいくつか提案されている [1], [2], [3], [4]。しかしながら、ビジネスもしくはシステムいずれかの観点を考慮した手法はあっても、両方を考慮した手法は少ない。さらに両方の観点を考慮しつつ自動的にマイクロサービス分割できる手法はない。そこで本研究では、ビジネスとシステムの両方の観点を組み合わせた自動的なマイクロサービス分割手法を提案する。

2. 提案手法

本研究では、ビジネスとシステムの両方を考慮したマイクロサービス分割手法を提案する。具体的にはプロセスマイニングとソースコード解析の結果を組み合わせることで、両方の観点を考慮した分割境界の特定を行う。プロセスマイニングとは、ログデータを解析し業務プロセスをグラフ構造として表現する手法である。なお、本研究では Java で実装された WEB アプリケーションを対象とする。

2.1 プロセスマイニングによる業務プロセスのグラフ化

最初に、提案手法ではログ出力を埋め込む。このステッ

¹ 株式会社 NTT データ
NTT DATA Corporation

^{a)} Kazuki11.Yokoi@nttdata.com

プではアプリケーション内のすべての Java ソースコードに対し、メソッドの実行開始と終了、そしてデータベースへのアクセスログを出力するソースコードを埋め込む。なお、ひとつのセッションを構成するログの系列を特定するため、セッション ID とタイムスタンプをログに含める。

次に、アプリケーションを一定期間実行してログを収集する。収集したログに対してプロセスマイニングを用いることで、業務プロセスを表現するグラフを構築する。このグラフのノードはメソッドまたはデータベーステーブルに対応する。また、グラフのエッジは実行時に呼び出されたメソッドまたはテーブルをつなぎ、呼び出し頻度に応じて重みづけをおこなう。このようにプロセスマイニングを用いて頻繁に実行される業務プロセスをグラフ化することで、ビジネス的なつながりを表現する狙いがある。

2.2 ソースコード解析によるプログラム依存関係のグラフ化

次はソースコード解析により、プログラム依存関係を明らかにする。ソースコード解析ではプログラムのメソッド呼び出しとデータベースへのアクセス情報を解析し、プログラムの依存関係を表現したグラフを構築する。グラフのノードは呼び出し関係にあるメソッドとデータベーステーブルに対応する。また、グラフのエッジは呼び出し関係にあるメソッドまたはテーブルをつなぎ、更新や参照など呼び出しの種類に応じてヒューリスティックに重みづけを行う。前節で説明した業務プロセスのグラフ表現とノードは同じになるが、エッジの重みが異なる。このようにソースコード解析を用いてプログラム依存関係をグラフ化することで、システム的なつながりを表現する狙いがある。

2.3 分割境界の特定

最後に、前手順で構築した2つのグラフに対して協調的クラスタリングを行い、分割境界を自動的に特定する。協調的クラスタリングとは、古典的な凝集型階層クラスタリングを拡張した手法である [2]。協調的クラスタリングを用いることで複数のグラフ構造を考慮したクラスタリングができる。提案手法では、業務プロセスグラフとプログラム依存関係グラフを協調させながらクラスタリングを行う。これにより、マイクロサービス分割において重要なビジネスとシステムの両方を考慮でき、より適切な分割境界の特定を狙いとしている。

3. 関連研究

Daoud らは業務プロセスの情報を用いたマイクロサービス分割手法を提案した [2]。業務プロセスに着目した点で提案手法と似ているが、Daoud らの手法は手作業で業務プロセスを定義する。提案手法はプロセスマイニングを用いて業務プロセスを定義する点で、Daoud らの手法とは異なる。

なる。

Taibi らはプロセスマイニングを用いてマイクロサービス分割するフレームワークを提案した [3]。プロセスマイニングを用いる点で提案手法と似ているが、Taibi らの手法は分割のための手順を示したフレームワークであり、分割境界を自動的に特定できるわけではない。提案手法は分割境界を自動的に特定する点で、Taibi らの手法とは異なる。

Krause らはドメイン駆動設計とシステム解析の結果を組み合わせたマイクロサービス分割手法を提案した [4]。ビジネスとシステムの両方を考慮する点で提案手法と似ている。しかしドメイン駆動設計は実施が難しく、専門家による手作業が必要である。提案手法はプロセスマイニングを用いてビジネス観点の解析ができる点で、Krause らの手法とは異なる。

4. まとめと今後の課題

本研究では、プロセスマイニングとソースコード解析を用いたマイクロサービス分割手法を提案した。提案手法を用いることで、ビジネスとシステムの両方の観点を考慮に入れたマイクロサービス分割が可能となる。

今後の課題として、以下の3点が挙げられる。

提案手法の評価 現時点では提案手法の評価まで実施できていない。提案手法を定量的に評価し、提案手法の有用性を示すことは今後の課題である。

グラフ化対象の拡張 本研究では業務プロセスと、メソッドとデータベースの呼び出し関係のみをグラフ化した。しかし提案手法が用いる協調的クラスタリングは、3つ以上のグラフ表現もクラスタリングが可能である。セッション共有やトランザクション境界など、グラフ化対象の拡張は今後の課題である。

COBOL などの他言語への拡張 本研究では Java アプリケーションを対象とした。レガシーシステムでは、Java 以外にも COBOL など他言語の利用が多い。他言語への拡張は今後の課題である。

参考文献

- [1] J. Fritzsche, J. Bogner, A. Zimmermann, and S. Wagner. From monolith to microservices: A classification of refactoring approaches. In *Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, pp. 128–141, 2019.
- [2] M. Daoud, A. El Mezouari, N. Faci, D. Benslimane, Z. Maamar, and A. El Fazziki. A multi-model based microservices identification approach. *Journal of Systems Architecture*, Vol. 118, p. 102200, 2021.
- [3] D. Taibi and K. Systä. From monolithic systems to microservices: A decomposition framework based on process mining. In *CLOSER*, 2019.
- [4] A. Krause, C. Zirkelbach, W. Hasselbring, S. Lenga, and D. Kröger. Microservice decomposition via static and dynamic analysis of the monolith. *ICSA-C*, pp. 9–16, 2020.