

グラフ表現学習を用いた Web API エコシステム分析方法の 提案と評価

加納 辰真^{†1*} 青山 幹雄^{†2}

概要：近年，Web API を通したネットワークで形成される Web API エコシステムが拡大し，その構造が複雑化している．本研究では，プロパティグラフを用いた Web API エコシステムグラフモデルを提案し，提案グラフモデルに対してグラフ深層学習(Graph CNN)を適用することにより Web API エコシステムグラフの特徴量を特定し，PageRank と併用することによる分析方法を提案する．提案方法を 2005 年から 2019 年までの 15 年間の ProgrammableWeb 上の Web API と GitHub 上のマッシュアップサービスのデータに適用し，Web API エコシステムの進化構造を分析した．分析結果から自然エコシステムで現れる特性が Web API エコシステムでも現れることを明らかにした．この結果，突然変異の出現により Web API エコシステムの構造の分散と自己組織化が繰り返されることを明らかにした．

キーワード：Web API エコシステム，ソフトウェア進化，自己組織化，グラフ表現学習，機械学習ソフトウェア工学

An Analysis Method of Web API Ecosystem Using Graph Representational Learning and its Evaluation

TATSUMA KANO^{†1*} MIKIO AOYAMA^{†2}

1. 研究の背景

近年，Web API を用いたソフトウェア開発が広がっている．そのため，Web API を通したネットワークで形成される Web API エコシステムが発展し，複雑化している．Web API エコシステムとは，Web API を介して Web サービスを連携し，価値を創出するシステムである．図 1 に Web API を利用し，2018 年 1 月 1 日から 12 月 31 日の間に提供されたマッシュアップサービスの Web エコシステムを示す．図は中心に近いほど，その PageRank [14][13]が高くなるように配置している．Google の主たる Web API を中心とするエコシステムが形成されていることが分かる．このような Web API エコシステムの構造やその特性を分析することは Web ソフトウェアアーキテクチャに加え，ビジネスアーキテクチャの構造分析につながり，新たな研究課題となっている

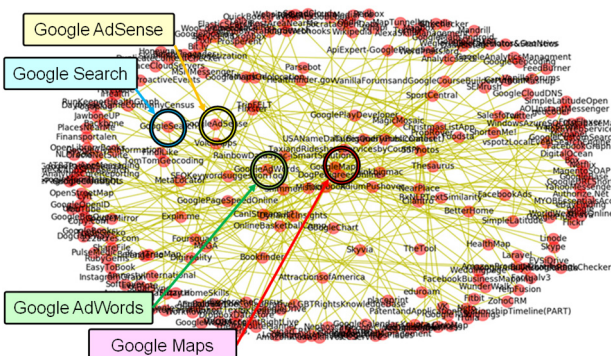


図 1 Web API のエコシステム
Figure 1 An Ecosystem around Some Google Web APIs

[1][3].

Web API エコシステムをグラフ理論に基づきグラフモデルを定義する研究は行われている．しかし，グラフモデルの深層表現学習を用いて Web API エコシステムの大域的特性を分析する研究は少ない．

本研究では，プロパティグラフモデル[17]を用いて Web API エコシステムを Web API エコシステムグラフモデルとして定義し，深層表現学習と PageRank を用いたグラフの構造化に基づく Web API エコシステムの進化特性を分析する方法を提案する．

2. 研究課題

本稿では，以下の 3 点を研究課題として設定する．

- (1) Web API エコシステムグラフモデルの提案
- (2) 深層表現学習を用いて Web API エコシステムの特徴量を獲得し，PageRank とあわせて構造を分析する方法の提案
- (3) 実際の Web API とマッシュアップサービスに適用し，提案方法の有効性と妥当性の評価

3. 関連研究

3.1 エコシステムの構造と進化

3.1.1 Web API エコシステムの構造化

Web API とそれを利用するマッシュアップサービスの関係をグラフモデルを定義し，Web API エコシステムを可視化する研究がある[6][24][27]．このモデルに対してグラフの構造分析を行うことにより局所的特性を分析している．しかし，大規模な

^{†1} 南山大学大学院 理工学研究科 ソフトウェア工学専攻
Graduate Program of Software Engineering, Nanzan University

^{†2} 南山大学 理工学部 ソフトウェア工学科
Dep. of Software Engineering, Nanzan University

*現在，株式会社ヤブリ勤務

Web エコシステム全体の特性を分析するまでには至っていない。

3.1.2 ソフトウェアエコシステムの構造と進化

ソフトウェアエコシステムとは API とそれを利用するアプリケーションのネットワークから構成されるエコシステムである[6]。ソフトウェアエコシステムは API を利用するアプリケーションの進化と共に進化することが示されている[21][24]。エコシステムの進化の源泉は相互作用にある。ソフトウェアエコシステムにおいてもソフトウェア間の相互作用により進化する[1]。

3.1.3 Web API エコシステム上で代替性を持つ Web API の推論

Web API とそれを利用するマッシュアップサービスは常時変化することから、ソフトウェアエコシステムに比べ変化が早いことが知られている。例えば、Web API の利用停止や類似性をもつ他の Web API の誕生により構造が動的に変化する。このため代替性を持つ Web API を推論する方法が提案されている。深層学習を用いて類似性を持つ Web API の推薦方法[9][8]やグラフデータベースを用いる研究[26]、統計的方法を用いる研究[23]がある。

3.1.4 ビジネスエコシステムの構造と進化

ビジネスエコシステムはその基盤となるソフトウェアエコシステム、さらには、Web API エコシステムと一体となっていることから、ビジネスエコシステムの研究は Web API/ソフトウェアエコシステムの研究と強い関連がある[4]。そのため、Web API エコシステムとビジネスエコシステムとの構造分析と進化に関する研究が行われている[3]。しかし、これらの研究は Web API エコシステムの構造分析とその応用に留まり、進化の構造分析までには至っていない。

3.2 グラフ深層表現学習

3.2.1 グラフ表現学習(Graph Representation Learning)

表現学習はデータの特徴量(Feature)を分散表現[10]として抽象化する方法である。グラフ表現学習とはグラフ構造からノードの分散表現を獲得する方法である[2][5]。このアルゴリズムとして graph2vec[11]などが提案されている。グラフの分散表現からラベル推定や分類タスクを高い精度で実行できる。例えば、グラフ表現学習を用いた OSS エコシステムの進化構造分析手法の提案がある[7]。大規模グラフの可視化へも応用されている[18]。

3.2.2 GCN (Graph Convolutional Networks)[19]

GCN はグラフ構造に対する畳み込み演算を定義し、CNN と類似したタスクを実行する[19][28]。Kipf らの提案した GCN の構造モデルが広く参照されている[8]。GCN は近傍ノードの信号を集約して重ね合わせ、畳み込み演算を行う。グラフを入力とし、1 層が 1 ホップ隣接したノードの情報の畳み込みを行う。柔軟な特徴抽出を可能とし、ノードの分類予測、ランキング、グラフ分類、リンク予測等のタスクにおいて従来の方法よりも高い精度を得ている[28]。

しかし、GCN をソフトウェア工学の問題へ応用する研究は、著者の知る限りない。

4. アプローチ

4.1 自然エコシステムから Web API エコシステムへ

本研究では、自然エコシステムで見られる特性が Web API エコシステムでも見られると仮定している。そのため、自然エコシステムの特性に着目した研究のアプローチをとる。

4.1.1 自然エコシステムの特性

自然エコシステムでは次の特性が知られている[1]。

(1) 個体間の変化がもたらすフィードバック機構

フィードバックには、ポジティブフィードバックとネガティブフィードバックがあり、自然エコシステムではポジティブフィードバックにより進化を加速する。

(2) 自己組織化

個体群は個体間の局所的結合、あるいは適合によって組織を構成する。

(3) 複雑系循環

自然エコシステムの進化は、漸進的変化と突然変異を組み合わせた複雑系循環で構成される。

(4) 共通化と共通適応

共通化は Web API エコシステムとビジネスシステムが共に進化することである。共通適応は相互に適応することである。個体間や個体群間で発現する。

4.1.2 自然エコシステムの進化構造

自然エコシステムには、次の 4 つの進化フェーズが見られることが知られている。このフェーズと進化の構造を明らかにすることで、Web API エコシステムの成熟度を分析することが可能である[20][21]。

(1) 誕生

自然エコシステムが誕生するフェーズ

(2) 成長

誕生した自然エコシステムが構成群と協調し、成長するフェーズ

(3) 覇権の交代

重要な構成群に代わって、新たな構成群が発現し、エコシステムの覇権交代となるフェーズ

(4) 再生

エコシステム内外の脅威から構成群は自らを変化させ、進化することで再生するフェーズ

4.2 Web API エコシステムの構造進化分析へのアプローチ

本研究では、エコシステムの構造が相互作用のグラフとしてモデル化できること[13]に基づき Web API エコシステムの構造をグラフ構造でモデル化し、その進化の特性を明らかにするため、Web API エコシステムのグラフ表現の動的な構造変化を分析する。グラフモデルとしてはノードとエッジに属性が定義できるプロパティグラフ[17]を基礎とする。

Web API エコシステムの構築期間を一定期間に分けて、Web API エコシステムグラフモデル(以下、エコシステムグラフモデルと略記)を構成し、その特徴量を獲得する。この特徴量は Web

エコシステムの大域的特性を表すと考えられる。それによって、エコシステムグラフモデルの構造を推定可能となる。あわせて、PageRank によりエコシステムグラフモデルを Web API を中心に構造化する。この二つの技術を組み合わせることにより、エコシステムグラフモデルの構造進化を視覚化し、その特性を評価可能にする。さらに、Web API エコシステムにおける重要なノード、Web API エコシステムの成熟度を評価することが期待できる。

このようなアイデアに基づくアプローチを図 2 に示す。

ここで、Web API は世界最大の Web API キュレーションサイトで事実上の標準である ProgrammableWeb[16]から取得する。マッシュアップサービスは GitHub から取得する。

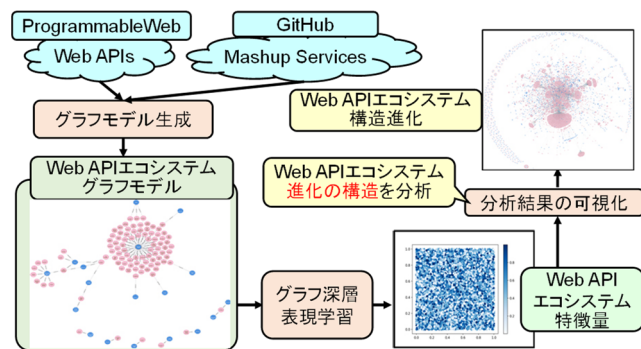


図 2 アプローチ
 Figure 2 Approach

5. 提案分析方法

5.1 Web API エコシステムモデルグラフの定義

Web API エコシステムを分析するためにプロパティグラフ[17]を基礎としたエコシステムグラフモデルを定義する。Web API エコシステムグラフモデルのメタモデルを図 3 に示す。

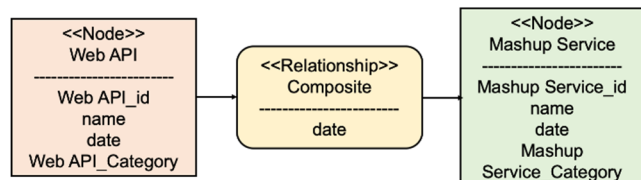


図 3 Web API エコシステムモデルグラフのメタモデル
 Figure 3 Metamodel of Web API Ecosystem Graph Model

5.1.1 ノードの定義

エコシステムグラフモデルでは、Web API とマッシュアップサービスをノードとする。表 1 にノードの定義を示す。

表 1 ノードの定義
 Table 1 Definition of Nodes

ノード	定義	プロパティ
Web API	Web API	Web API_id, name, date, Web API_Category
Mashup Service	マッシュアップサービス	Mashup Service_id, name, date, Mashup Service_Category

5.1.2 エッジの定義

エッジはマッシュアップサービスノードから利用する Web API ノードへのリンクを表す。表 2 にエッジの定義を示す。

表 2 エッジの定義
 Table 2 Definition of Edges

エッジ	定義	プロパティ
Composite	マッシュアップサービスに使用する Web API	date

5.1.3 プロパティの定義

エコシステムグラフモデルの各ノードとエッジにプロパティを定義する。任意のノードとエッジには、1 個以上のプロパティのセットが関連付けられる。プロパティの定義を表 3 に示す。

表 3 プロパティの定義
 Table 3 Definition of Properties

プロパティ	定義
Web API_id	Web API の ID
name	名前
date	日付
Mashup Service_id	マッシュアップサービスの ID
Web API_Category	Web API のカテゴリ
Mashup Service_Category	マッシュアップサービスのカテゴリ

5.2 Web API エコシステム分析方法のプロセス

本稿で提案する Web API エコシステム分析方法のプロセスを図 4 に示す。

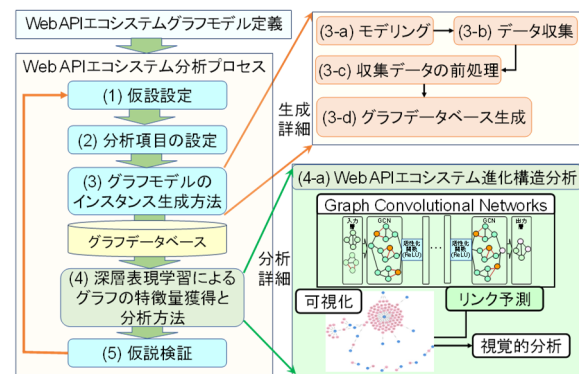


図 4 分析方法のプロセス
 Figure 4 Analysis Method Process

(1) 仮設定

本研究では Web API エコシステムの進化特性が自然エコシステムの進化特性に従うと仮定する。この仮定に基づき Web API エコシステムの進化が推進されると仮説を設定した。設定した仮説は検証プロセスの結果に基づき必要があれば変更する。

(2) 分析項目の設定

設定した仮説に基づいて、仮説ごとに明らかにすべき項目を列挙する。列挙した項目を統合して、分析項目を設定する。

(3) グラフモデルのインスタンス生成方法

(3-a) モデリング

分析対象の Web API エコシステムに対して 5.1 節で定義したエコシステムグラフモデルに必要なノードとエッジがある場合はエコシステムグラフモデルを拡張する。

(3-b) データの収集

データ収集は ProgrammableWeb からスクレイピングを用いて Web API を、GitHub から GitHub API を利用してマッシュアップサービスのデータを収集する。

(3-c) 収集データの前処理

収集したデータには、ノイズデータが含まれる可能性がある。

そこでグラフデータベースへデータを入力する前に必要があればノイズデータの除去などのデータクレンジングを行う。

(3-d) グラフデータベース(グラフ DB)の生成

グラフ DB 上にエコシステムグラフモデルのインスタンスを生成し、前処理されたデータを入力する。

(4) 深層表現学習によるグラフの特徴量獲得と分析方法

(4-a) Web API エコシステム進化構造分析

グラフ DB 上に生成したエコシステムグラフモデルのインスタンスに対して、GCNの畳み込み演算によって、グラフのノードに対する特徴量を抽出する。抽出された特徴量を用いて、グラフ上でのリンク予測を行い、Web API とマッシュアップサービスの関係を分析する。Web API とマッシュアップサービスの関係を期間ごとに分析することで Web API エコシステムの進化特性と後述する重要度を分析する。

(5) 仮説検証

分析で得られた結果から、設定した仮説を検証する。必要に応じ、得られた結果に基づいて、仮説の変更を行い、はじめからの一連のプロセスを繰り返す。

また、本研究ではこのプロセスにより設定された仮説の検証がグラフ深層表現学習によって実現できたかを評価する。

5.3 深層表現学習によるグラフの特徴量獲得と分析方法

生成されたエコシステムグラフモデルのインスタンスを隣接行列に変換する。一方、エコシステムグラフモデルのインスタンスを訓練データとテストデータに分割する。この訓練データとテストデータによって、学習モデルの評価を行う。

学習方法は、畳み込み演算を用いてノードに対する特徴量を獲得する。獲得した特徴量を視覚的に分析することによって Web API エコシステムの進化特性を分析する。図 5 に本研究で使用する GCN の構造を示す。隣接ノード自体も、前の層ではターゲットノードとしてその隣接ノードに畳み込みしているため広範囲の情報が畳み込まれる。

GCN は次の 3 ステップで実行する。

- (1) 畳み込みの中心となるターゲットノードを決定。任意の隣接関係にあるノードの情報を畳み込む
 - (2) 上の(1)で畳み込めない非隣接ノードの畳み込み
 - (3) さらに(2)の畳み込みで残ったノードに対する畳み込み
- 隠れ層の数は、(1)(2)(3), それぞれ, 1,430, 16, 7 と設定した。

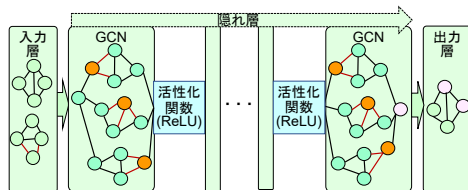


図 5 GCN のネットワーク構造
 Figure 5 GCN Networks Structure

6. プロトタイプの実装

6.1 プロトタイプの目的と構成

提案方法により自然エコシステムの特徴が Web API エコシ

テムでも発現するか評価するためにプロトタイプを用いる。

本研究で提案する分析方法を支援するプロトタイプの構成を図 6 に示す。

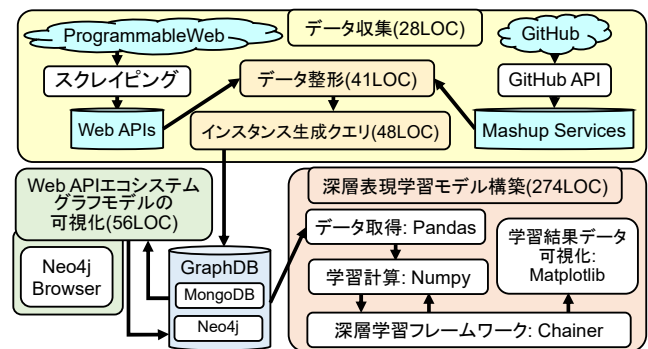


図 6 プロトタイプの構成

Figure 6 Prototype Configuration

6.2 実装環境

表 4, 表 5 にプロトタイプを実装するための環境と深層表現学習の構築に使用した環境を示す。

表 4 実装環境

Table 4 Implementation Environment

コンポーネント名	名称	Version
OS	Ubuntu	18.04
Graph Data Base	Neo4j [12][17]	3.5.9
	MongoDB	3.6.14
可視化ツール	Neo4j Browser [12]	3.5.9
データ収集	Python	3.7.4
利用する外部 API	GitHub API	V4

表 5 深層表現学習に用いた実装環境

Table 5 Implementation Environment Used for Deep Representational Learning

コンポーネント名	名称	版
OS	Ubuntu	18.04
実装言語	Python	3.7.4
深層学習フレームワーク	Chainer [15][22]	6.4.0
学習結果データの可視化	Matplotlib	3.1.1
データ加工ツール	Pandas	0.25.1
学習計算	Numpy	1.17.0

7. Web API エコシステムデータへの適用と評価

7.1 適用の方法

Web API エコシステムの特徴を獲得するために提案分析方法を適用する。適用するデータを訓練データ、テストデータに分けて、正解率と誤差を算出する。正解率の算出式は式[1]に示す。誤差は、深層学習で式[2]と式[3]に示す Softmax 関数と Cross-Entropy 関数から導出された式[4]に示す Softmax-Cross-Entropy 関数を用いる。Softmax-Cross-Entropy 関数は、一般的に CNN, GCN など広く使われている誤差関数である。算出された値で深層表現学習の学習モデルを評価する。

$$\text{正解率} = \frac{\text{正解したデータ数}}{\text{総データ数}} \quad [1]$$

$$\text{Softmax} = \frac{e^{x_{ij}}}{\sum_{k=1}^n e^{x_{ik}}} \quad [2]$$

$$\text{Cross-Entropy} = -\frac{1}{m} \sum_{i,j} t_{ij} \log y_{ij} \quad [3]$$

$$\begin{aligned} \text{Softmax-Cross-Entropy} &= -\frac{1}{m} \sum_{i,j} t_{ij} x_{ij} \\ &+ \frac{1}{m} \sum_{i,j} t_{ij} \log \left(-\frac{1}{m} e^{x_{ij}} \right) \end{aligned} \quad [4]$$

7.2 Web API エコシステムデータ

分析対象期間を表 6 に示す。2005 年 9 月 1 日から 2019 年 8 月 31 日の間、ProgrammableWeb[16]と GitHub 上で公開された Web API とマッシュアップサービスを対象として分析を行う。

表 6 分析対象データ

Table 6 Analysis Target Data

分析期間	Web API	マッシュアップサービス
期間	2005/09/01~2019/08/31	2005/09/01~2019/08/31
ノード数	19,793	6,401

7.3 GCN の実行

表 6 に示した分析対象データを用いて GCN で学習するために、表 7 に示すように訓練データとテストデータに分割した。次節で説明するように epoch 数は正解率と誤差の推移から 200 に設定した。

これらの学習により、2019 年における Web API エコシステムの構造を予測し、Web API エコシステムの構造進化分析の評価対象として用いている。

表 7 GCN のデータ利用

Table 6 Data Usage for GCN

利用形態	訓練データ	テストデータ
データの範囲	2005/09/01~2017/12/31	2018/01/01~2019/08/31

7.4 GCN の評価

GCN による Web API エコシステム深層表現学習モデルの有効性を評価するために正解率と誤差の推移と 200 epoch における学習モデルの評価結果を表 8 と図 7 と図 8 に示す。図 7 より訓練とテストの正解率が共に 1.0 に近い値で収束していることから、この深層表現学習はリンク予測の学習が適切にできていると判断できる。また、図 8 より汎化誤差も低い値で収束しているため、過学習とはなっていないと判断できる。

表 8 200epoch における学習結果

Table 8 Learning Results at 200 Epoch

エポック数	訓練 正解率	訓練 誤差	テスト 正解率	汎化 誤差
200	0.95	0.36	0.82	0.66

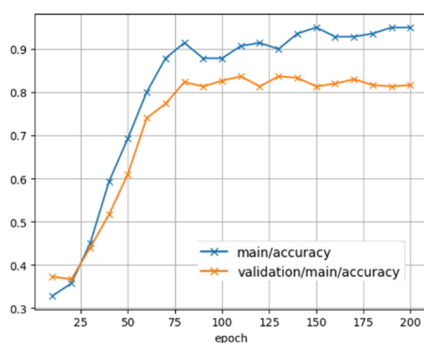


図 7 正解率の推移

Figure 7 Trend of Accuracy

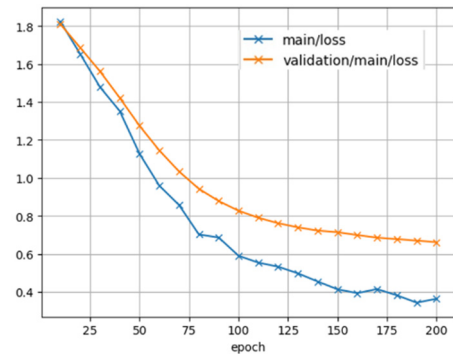


図 8 誤差の推移

Figure 8 Trend of Loss

7.5 Web API エコシステムの視覚的分析結果

Web API エコシステムデータを Web API エコシステムグラフモデルに基づき、一定期間毎に分割し、視覚的分析を行った。図 9 に Web API エコシステム構造の進化を示す。

表示するグラフの配置は、PageRank[14][13]によって重要度を算出し、中央に近いほど重要度が高くなるようにしている。また、青色のノードは Web API、ピンク色のノードはマッシュアップサービスを表す。

Web API エコシステムは誕生からある期間毎に組織化と分散を繰り返している。組織化は同一カテゴリ内での連鎖、分散は突然変異の出現が挙げられる。図 10, 11 に組織化と突然変異の例を示す。Web API エコシステム進化の構造を図 12 に示す。

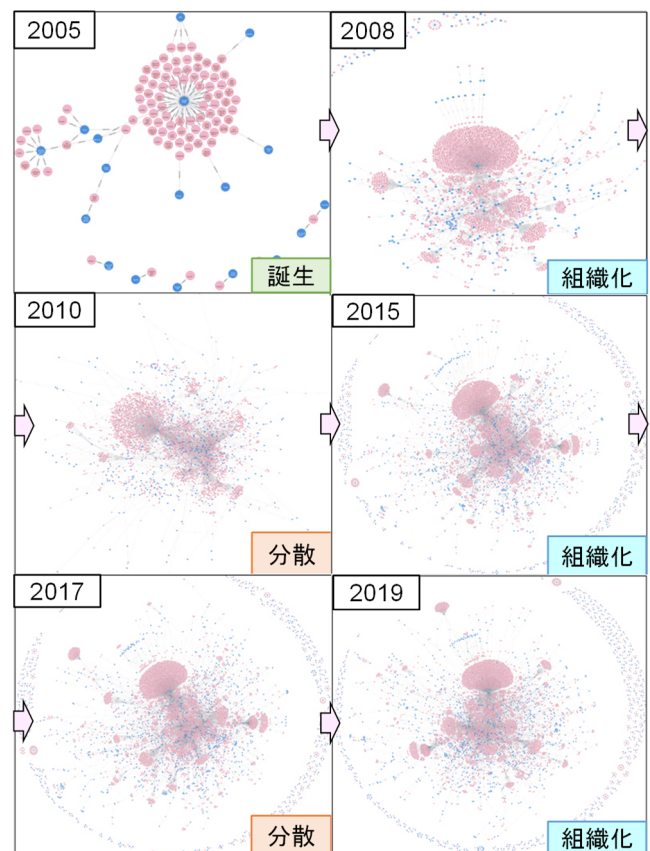


図 9 Web API エコシステム構造の進化

Figure 9 Evolution of the Web API Ecosystem Structure

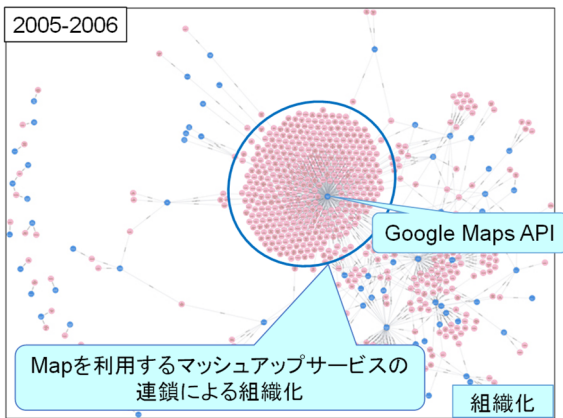


図 10 Web API エコシステムの組織化の例
 Figure 10 Examples of Organizing the Web API Ecosystem

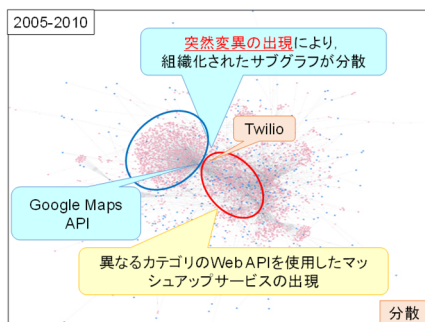


図 11 Web API エコシステムの組織化の例
 Figure 11 Examples of Mutations in the Web API Ecosystem

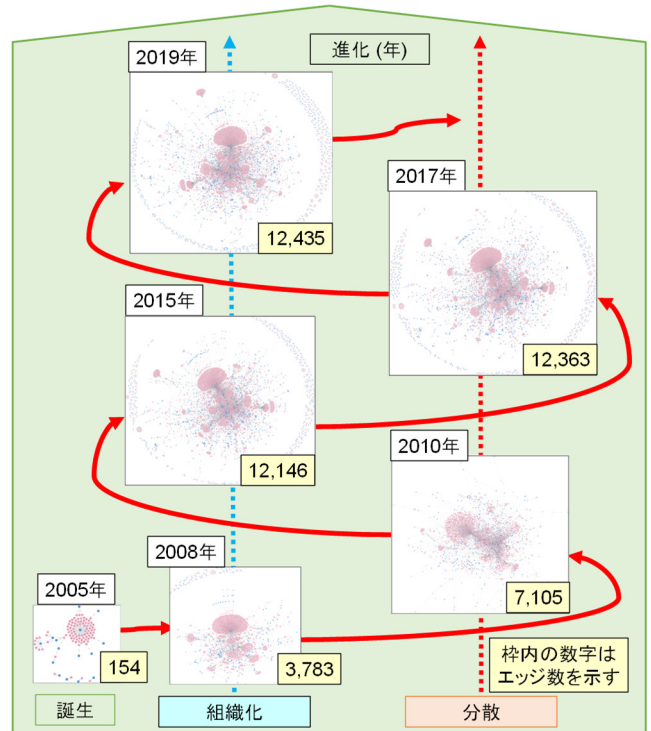


図 12 Web API エコシステム進化の構造
 Figure 12 Structure of Web API Ecosystem Evolution

8. 考察

8.1 エコシステムの進化特性

Web API エコシステムの視覚的分析により、自然エコシステムで見られる特性が Web API エコシステムでも明らかであるか考察する。

8.1.1 ポジティブフィードバックによる共進化

Web API エコシステムの組織化から Web API エコシステムでは、Web API の開発による Web API の進化がマッシュアップサービスの進化の連鎖となり、この相互作用により共進化していることが明らかになった。例として、図 13 を示す。Web API とマッシュアップサービスそれぞれの増加を表 9 に示す。「Google Maps API」の進化が「Map」に関するマッシュアップサービスの進化を促している。逆に、マッシュアップサービスの進化が Web API の進化を促している。その関係を図 14 に示す。

表 9 Web API とマッシュアップサービスの共進化

Table 9 Statistics of Coevolution of Web APIs and Mashup Services

分析対象	2005	2006
「Mapping」に関する Web API 数	6	13
「Mapping」を利用するマッシュアップサービス数	88	522

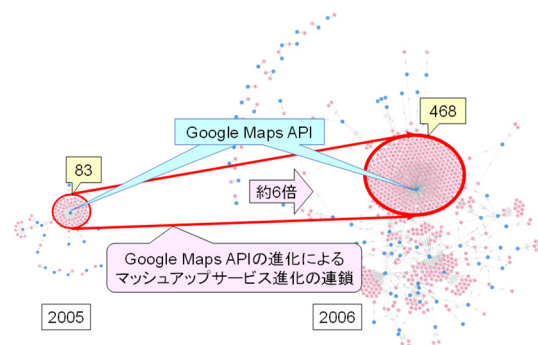


図 13 Web API エコシステムの共進化の例
 Figure 13 Examples of Coevolution of the Web API Ecosystem

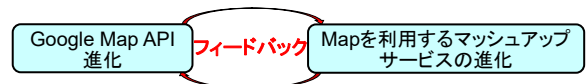


図 14 Web API エコシステムの共進化の関係
 Figure 14 Relationship of Coevolution of the Web API Ecosystem

8.1.2 自己組織化

組織化の構造から Web API エコシステムでは同じカテゴリで分類される Web API とマッシュアップサービスの連鎖によって組織を形成することが明らかになった。例として、「Twilio」が誕生し、「Telephony」に関するマッシュアップサービスも誕生し、組織化の連鎖によって、Web API エコシステムが進化している。その様子を図 15 に示す。

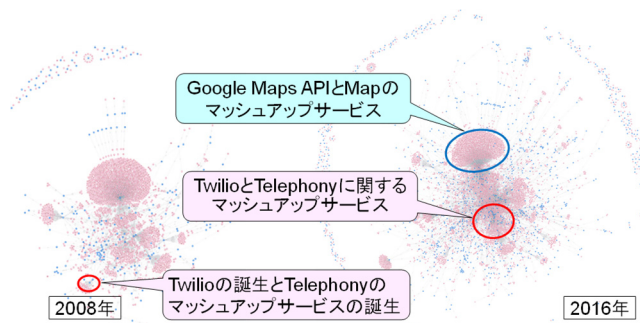


図 15 自己組織化の例
 Figure 15 Example of Self-Organization

8.1.3 複雑系循環

図 12 より, 組織化と突然変異が出現して分散へと循環することによって Web API エコシステムが進化することが明らかになった。分散は突然変異が出現することで, 組織化で起こった, Web API エコシステム内のサブグラフが発散することである。Web API エコシステムの突然変異として, 異なるカテゴリの Web API で構成されるマッシュアップサービスの出現が観測された。突然変異によって, 分散している例を図 16 に示す。

8.1.4 共通化と共通適応

組織化から Web API エコシステムでは, Web API の仕様変更とマッシュアップサービスに使用する Web API の変化との間に依存関係があることが明らかとなった。例として, 図 17 に「Instagram API」の仕様変更によって, 「Social」に関するマッシュアップサービスが組織化から分散へと変化することを示す。

共通化はエコシステム内の自己再組織化とそれをビジネスシステムに適応することとの 2 つの進化パターンが確認できた。

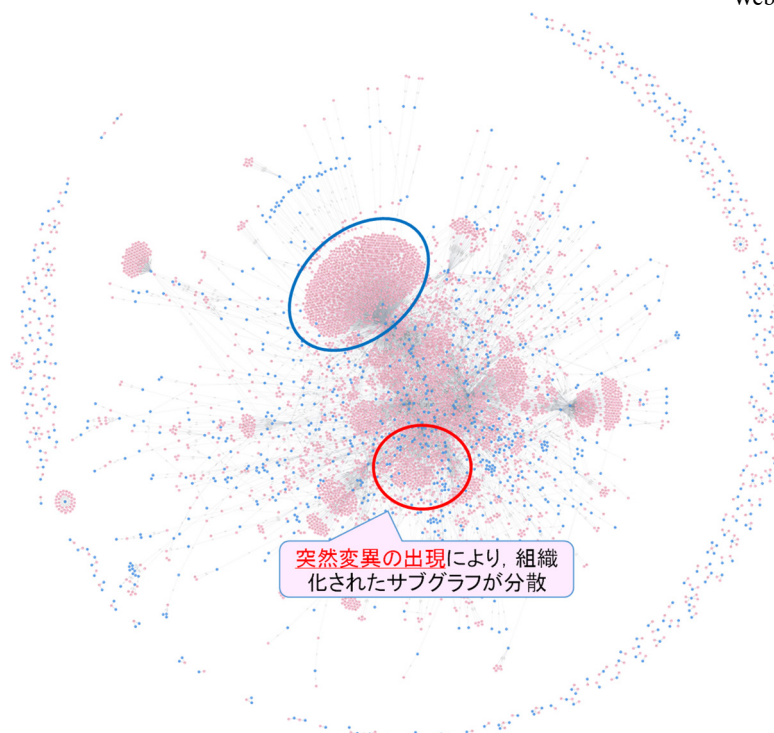


図 16 突然変異の出現による分散の構造
 Figure 16 Structure of Variance Due to the Appearance of Mutations

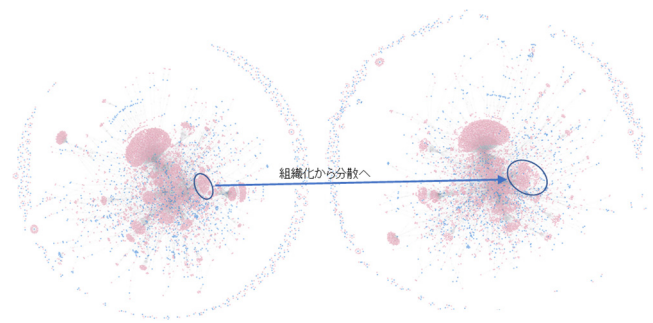


図 17 Web API の仕様変更による構造の進化
 Figure 17 Structural Evolution due to Changes in Web API Specifications

8.2 Web API エコシステム進化の構造

図 12 より Web API エコシステム進化の構造が明らかになった。Web API エコシステム進化の構造には組織化と分散を循環している。組織化と分散は循環することで, Web API エコシステムが進化しているが, 組織化と分散の構造がそれぞれ進化することで, Web API エコシステムもしているかを考察する。

自然エコシステムでは, 進化のフェーズとして, 「誕生」, 「成長」, 「覇権の交代」, 「再生」の 4 つがある。それぞれが Web API エコシステムでは組織化と分散を循環する構造に適応させて考察することで, Web API エコシステムの成熟度を分析することが可能と考える。

8.2.1 Web API エコシステムの誕生期

Web API エコシステムの誕生期では, Web API が出現し, それによってマッシュアップサービスが出現する。Web API とマッシュアップサービスによって Web API エコシステムが誕生する期間である

8.2.2 Web API エコシステムの成長期

Web API エコシステムの成長期では, 8.2.1 と同様に Web API とマッシュアップサービスが出現する。それらが, 8.1.1 のポジティブフィードバックによる共進化と 8.1.2 の自己組織化によって, Web API エコシステム内のサブグラフが構築され, 拡大している。Web API エコシステム内で重要なノードは Web API エコシステムグラフの中央に位置し, そうではないノードは Web API エコシステムグラフの円周に位置する。この成長期では円周に位置するノードも増加していることが明らかになった。

8.2.3 Web API エコシステムの覇権の交代期

Web API エコシステムの覇権の交代期では, 8.2.1 や 8.2.2 と同様に Web API とマッシュアップサービスが出現する。しかし, 突然変異が出現することで成長期に構築されたサブグラフが分散する。また, 8.2.2 の成長期で Web API エコシステムグラフの円周に位置するノードが Web API エコシステムグラフの円の中央に位置が変わっている。これは, 分散することでノードの重要性が変化していることを意味している。

8.2.4 Web API エコシステム再生期

Web API エコシステムの再生期では, 8.2.3 で覇権の交代が起こり, ノードの重要性が変わって, 分散し

たサブグラフが共通により、自己組織化することでサブグラフを再構築している。また、将来、重要となる Web API エコシステムの円周に位置するノード群もこのフェーズにおいて増加していることが明らかとなった。この再生期は成長期と類似して、ポジティブフィードバックによる共進化や自己組織化も同様に行われていることが明らかとなった。

8.3 提案方法の妥当性と有効性

8.1 節より、自然エコシステムで見られる進化特性が Web API エコシステムでも発現することが明らかになった。従って、自然エコシステム進化の構造と Web API エコシステム進化の構造には類似性があるといえる。この結果は、Web API を利用するアプリケーション開発の進化特性を明らかにしたといえ、提案方法は有効であるといえる。また、一般に知られている自然エコシステムの特性が Web API エコシステムでも発現することが確認されたことから、本研究の提案方法は妥当性であるといえる。

8.4 関連研究との比較に基づく考察

グラフ表現学習を用いた Web API エコシステムの進化構造分析は新しいアプローチである。関連研究[7]では、OSS のコミュニティにおける開発活動の構造進化分析に同様のアプローチをしている。本稿ではこのアプローチを発展させ、Web API のエコシステムの構造進化の分析方法を提案している。

また、グラフ表現学習は機械学習において近年急速に発展している技術である[28]。ソフトウェア工学の対象はグラフで表現できることが多いことから、グラフ表現学習の適用は有用であると期待できる。これまで、ソフトウェア工学におけるグラフ表現学習の適用はほとんどない。本稿の成果はソフトウェア工学におけるグラフ表現学習の有用性を示している点で意義があるといえる。

9. 今後の課題

- (1) GCN の構造とアルゴリズムの検討
- (2) Web API エコシステムの構造進化分析方法の改善
- (3) 分析結果の評価方法の改善

10. まとめ

本稿では、Web API エコシステムの動的な構造変化に着目してグラフ表現学習を用いた Web API エコシステム分析方法を提案した。提案方法により、グラフ表現学習モデルの一つである GCN を用いて Web API エコシステムの特徴量を獲得した。提案方法を ProgrammableWeb の Web API を利用するマッシュアップサービスの Web API エコシステムに適用した、この結果から進化の構造を可視化し、自然エコシステムで見られる進化特性が Web API エコシステムでも見られることを明らかにした。

提案方法は GCN をソフトウェア工学に応用した点で新たな成果といえる。また、本成果は Web API エコシステムの進化構造を分析し、視覚化する新たな方法として有用性が期待できる。

謝辞: 本研究は JSPS 科研費 JP18K11251 の助成を受けたものです。

参考文献

- [1] 青山 幹雄, サービス生態系工学への扉を開く, 情報処理学会ソフトウェア工学研究会ウィンターワークショップ 2008・イン・道後 論文集, Vol. 2008, No. 3, Jan. 2008, pp. 79-80.
- [2] 浅谷 公威, ネットワークの表現学習, 人工知能, Vol. 31, No. 4, 2016, pp. 587-593.
- [3] R. C. Basole, et al., Understanding Business Ecosystem Dynamics: A Data-Driven Approach, ACM Trans. Management Information System, Vol. 6, No. 2, Article 6, May 2015, pp. 1-32.
- [4] R. C. Basole, Accelerating Digital Transformation: Visual Insights from the API Ecosystem, IEEE IT Professional, Vol. 18, No. 6, Nov.-Dec. 2016, pp. 20-25.
- [5] W. L. Hamilton, et al., Representation Learning on Graphs: Methods and Applications, IEEE Data Engineering, Vol. 40, No. 3, Sep. 2017, pp. 52-74 [arXiv:1709.05584].
- [6] S. Jansen, et al., A Sense of Community: A Research Agenda for Software Ecosystems, Proc. of ICSE 2009, IEEE, May 2009, pp. 187-190.
- [7] 加藤 聖也, ほか, 機械学習によるグラフモデル OSS 開発コミュニティ構造の特徴量分析方法の提案と評価, 情報処理学会第 81 回全国大会講演論文集, Vol. 1, No. 6N-03, Mar. 2019, pp. 281-282.
- [8] T. N. Kipf and M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, Proc. of ICLR 2017, Apr. 2017, pp. 1-14 [arXiv:1609.02907].
- [9] H. Labbaci, et al., A Deep Learning Approach for Web Service Interactions, Proc. of WI 2017, ACM, Aug. 2017, pp. 848-854.
- [10] T. Mikolov, et al., Distributed Representations of Words and Phrases and Their Compositionality, Proc. of NIPS 2013, Dec. 2013, pp. 3111-3119.
- [11] A. Narayanan, et al., graph2vec: Learning Distributed Representations of Graphs, Proc. of MLG 2017, Aug. 2017, pp. 1-8 [arXiv: 1707.05005].
- [12] Neo Technology, Neo4j, <http://neo4j.com/>.
- [13] M. A. Nowak, Evolutionary Dynamics, Belknap Press/Harvard University Press, 2006.
- [14] L. Page, et al., The PageRank Citation Ranking, Tech. Rep., Stanford InfoLab, 1999, pp. 1-17.
- [15] Preferred Networks Inc., Chainer, <http://chainer.org/>.
- [16] ProgrammableWeb, <https://programmableweb.com/>.
- [17] I. Robinson, et al., Graph Database, 2nd ed, O'Reilly, 2015.
- [18] M. Russell, Mining the Social Web, 2nd ed., O'Reilly, 2014.
- [19] M. Schlichtkrull, et al., Modeling Relational Data with Graph Convolutional Networks, Proc. of ESWC 2018, LNCS Vol. 10843, Springer, Jun. 2018, pp. 593-607.
- [20] J. Teixeira, et al., Lessons Learned from Applying Social Network Analysis in an Industrial Free/Libre/Open Source Software Ecosystem, J. of Internet Services and Applications, Vol. 6, No. 14, Springer, Jul. 2015, pp. 1-27.
- [21] J. Teixeira, et al., How Do Software Ecosystems Co-Evolve?, Proc. of ICSOB 2017, LNBIP Vol. 304, Springer, Aug. 2017, pp. 115-130.
- [22] S. Tokui, et al., Chainer: A Deep Learning Framework for Accelerating the Research Cycle, Proc. of KDD 2019, ACM, Aug. 2019, pp. 2002-2011.
- [23] X. Wang, et al., Mashup-Oriented API Recommendation via Random Walk on Knowledge Graph, IEEE Access, Vol. 7, 2018, pp.7651-7662.
- [24] M. Weiss, et al., Modeling the Mashup Ecosystem: Structure and Growth, R&D Management, Vol. 40. No. 1, Jan. 2010, pp. 40-49.
- [25] M. White, et al., Toward Deep Learning Software Repositories, Proc. of MSR 2015, IEEE, May 2015, pp. 334-345.
- [26] E. Wittern, et al., API Harmony: Graph-Based Search and Selection of APIs in the Cloud, IBM J. of Research and Development, Vol. 60, Mar. 2016, pp. 1-11.
- [27] S. Yu, et al., Innovation in the Programmable Web: Characterizing the Mashup Ecosystem, Proc. of ICSOC 2008, LNCS Vol. 5472, Springer, Dec. 2008, pp. 136-147.
- [28] Z. Wu, et al., A Comprehensive Survey on Graph Neural Networks, IEEE Trans. on Neural Networks and Learning Systems, Mar. 2020, pp. 1-21 [arXiv:1901.00596].