

TwitterAPI1.0 の廃止によるアプリケーションへの影響の分析

2010SE078 梶田裕紀 2010SE220 高井亮輔 2010SE226 高須健

指導教員：横森励士

1 はじめに

近年のソフトウェア開発では、第三者が提供するコードや API を利用してソフトウェアを開発することは珍しくない。ソフトウェアの保守が長期にわたる場合、開発の間にフレームワークが新しいバージョンに更新されることも考慮する必要がある。また、API などを通じて提供されるサービスの内容が変わったり不具合に対して対策を行った結果、その変化に対応することが必要となる場合も考えられる。これらの更新や変更に対して、実際の開発プロジェクトがどのように対応しているかを分析することは、ノウハウの蓄積の観点から有益であると考えられる。

本研究では TwitterAPI1.0 が廃止された事例を題材に、実際にその API を利用しているオープンソースアプリケーションがどう対応したかを調査する。どの程度の量のプロジェクトが対処を行ったか、対処を行ったプロジェクトに対してはどのように対応を行ったかを調査する。これらの調査を通じて、フレームワークや API によって提供される機能が廃止された時に、開発者側がどのようなことを考慮して対応を行うべきか考察する。これらの知見を得ることで、オープンソースのプロジェクトを扱うときに考慮しなければならないこと、一般的な保守作業において考慮すべきことを提案し、近年のソフトウェア開発や保守における知見を得る。

2 背景技術

2.1 TwitterAPI

TwitterAPI[1] とは Twitter 社が提供している Twitter[2] における API の総称である。Twitter でのツイートやフォローなどといった機能がまとめられている。Twitter 社は開発者に向けて API を公開しており、各プロジェクトはこれに沿って開発されている。TwitterAPI は web 経由でサービスにアクセスする方式を採用しており、web アプリケーションとの親和性が高い。開発者の視点から見ると、各 API が提供するサービスを利用したアプリケーションの開発が容易である。公開された TwitterAPI を利用して Twitter のほとんどの機能呼び出すことができるが、そのためには様々なプログラミング言語にあったライブラリが必要になる。このライブラリを自分で組み合わせる TwitterAPI を呼び出すのは困難であるが、オープンソースなどでたくさんのライブラリ API が公開されている。

2.2 関連研究

Xia の研究 [3] では、zlib、libpng などなどのライブラリとして提供されているソースコードを題材に、それらが

どのようにコピーペーストされて利用されているか、ライブラリ側の更新があった場合にどのように対応しているかを、ファイル名やリポジトリの更新履歴の分析に基づいて調査を行った。

その結果、130 個ほどのオープンソースプロジェクトがコピーによる利用を行っており、そのうち zlib で 3 割、libpng では 9 割のオープンソースプロジェクトが何かの問題を抱えているライブラリのバージョンをコピーして利用していることがわかった。全体の 8 割ほどのプロジェクトがライブラリのコードをそのまま取り込んでいる一方で、残りの 2 割ではコピーペースト後に修正を加えていることがわかった。全体では、15% ほどのプロジェクトはしっかりとライブラリ側のアップデートにそってアップデートされている一方で、約 70% のプロジェクトは取り込んだソースコードの更新を行っていないことがわかった。これらの結果について、開発者は機能の実現に優先的に関心を持っており、セキュリティ面において問題が起こる頻度は低いと考えられることからあまり関心がないのではないかと考察を行っていた。

3 研究内容

3.1 動機

API やライブラリの移行という事例に対して、分析を行っている研究はあまり多くなく、先行研究 [3] で述べた論文くらいである。実際には、重大なバグや不具合やサービス提供側の変化によって対応を余儀なくされることが想定される。このような更新や変更に対して、実際の開発プロジェクトがどのように対応しているかを分析することは、プロジェクト開発におけるノウハウの蓄積の観点から有益であると考えられる。

3.2 目的

オープンソース開発において、今まで提供されていた API が機能しなくなった場合にどのような対応が取られるのかを調査する。また、API が機能しなくなった場合に、新たに利用可能な API への対応がなされているか、なされている場合にはどのような対応がなされているのかを調査する。オープンソース開発において API が機能しなくなった場合に、開発者がどのように対応を取るのかを調査することで、オープンソース開発におけるフレームワークや API 利用の特徴が得られると考える。

3.3 アプローチ

TwitterAPI が 1.0 から 1.1 に完全移行した事例を題材に、TwitterAPI を利用しているオープンソースアプリケーションを列挙し、開発履歴やリリースノート、実際の

成果物をもとにする。以下の項目について調査する。

Q1：どれだけ TwitterAPI の移行に対応しているソフトウェアが存在するか。

Q2：API1.1 へ対応しているアプリケーションが存在している場合に、どのように対応しているか、また何か特別な処置を施しているか。

これらの事例からオープンソース開発プロジェクトの特色や、API の移行に関して考慮すべきことなどを考察する。

4 TwitterAPI の移行

4.1 現在までの流れ

2006 年に TwitterAPI のサービスが開始されてからしばらくの間 API1.0 が継続して使われていたが、API1.0 が廃止されることが 2012 年 8 月 16 日に発表され、最終的に 2013 年 6 月 12 日には API1.1 へ変更された。2012 年 8 月にバージョン 1.0 の廃止が告知されてから約 1 年間の移行期間が存在した。現在はバージョン 1.1 に完全移行しており、API1.0 は全く機能していない。公式の発表より主要な変更内容を列挙したものを以下に示す。これらの変更は全て適応保守に分類される [5]。

4.2 変更内容

API1.0 対応のエンドポイントの廃止

REST API を呼び出す際のエンドポイントは `http(s)://api.twitter.com/1/*` だったが、API 1.1 で `http(s)://api.twitter.com/1.1/*` へ変更された。検索 API 呼び出し用のエンドポイントは `http://search.twitter.com/search.json` だったが、API1.1 では REST API と統合され `https://api.twitter.com/1.1/search/tweets.json` へ変更された。

レートリミットの改良

サードパーティ製クライアントの API1.1 のメイン TL が 15 分で利用できる回数は 15 回である。公式クライアントの API1.1 のメイン TL が 15 分に利用できる回数は 180 回ととても大きな差が生まれ、サードパーティ製クライアントでの API 呼び出し回数には非常に大きい制限がかけられた。

取得可能データの変更

Twitter 社は、プラットフォーム間で共有される JSON フォーマットを支持することを選択した。API1.0 では、「XML」「Atom」「RSS」でも取得可能だったが API1.1 では利用することができなくなった。

すべてのエンドポイントで認証は必須

API1.1 では、アプリケーションすべてのリクエストは OAuth 1.0a で認証することを要求している。API1.0 では、認証を通すことなく利用できたアプリケーションも存在していたが、悪質な行為を防止するための措置が行われた。この変更により、アプリケーションご

とにどのような API が利用されているかを調査できるようになった。

開発者ディスプレイ要件

以前のサードパーティ製クライアントでは開発者独自の見た目を自由に生かしたものになっていた。しかし、Twitter 社は API を通じて取得したコンテンツを「Display Guidelines」で決められたフォーマットに従って表示するよう開発者に要請をした。いままでは強制することはなかったが、今回からアプリケーションは必ずそのフォーマットに従ってツイートなどの表示をしなければいけなくなった。この制約を守らずに API を利用していた場合は、アプリケーションの動作は不可能となる。

新しい Twitter クライアントポリシー

API1.1 ではユーザー 1 人 1 人にトークンが割り振られ、そのトークンにより認証を行うが、サードパーティ製クライアントには 10 万人ユーザーの制限がつけられた。上限を超えてしまったアプリケーションでは、インストールをすることはできてもユーザー認証を行うことができなくなる。こちらはクライアントのみ適応されるもので、Twitter と連携するサービスなどでは関係を持たない。

5 調査

5.1 プロジェクトの抽出と分類

SourceForge[4] で公開されている Twitter に関連するオープンソースアプリケーションを中心に、TwitterAPI を利用しているものを対象として抽出し、分析した。それぞれのアプリケーションについて開発履歴、リリースノート、最終更新日、実際の成果物の情報を集めた。さらにアプリケーションをファイルの有無、リリースノートの有無、で分類を行う。

総数	238個
成果物あり	171個
リリースノートあり	29個
成果物なし	67個

図1 プロジェクトの分類

238 個のオープンソースプロジェクトが分析対象としてリストアップされたが、その中で成果物が存在したオープンソースプロジェクトは 171 個であった。また成果物があるプロジェクトのうち、リリースノートが残されていたオープンソースプロジェクトは 29 個であった。

以下では最終更新日が TwitterAPI1.0 廃止の発表が行われた 2012 年 8 月以降かつ API の移行に対応している

ソフトウェアはどのくらいあるのかを調査し、アプリケーションが API1.1 に対応がなされていた場合には、対応した内容と特別な処置を施しているかを確認する。

5.2 対応状況の調査

抽出できたオープンソースプロジェクト 171 個のうち、1 種類の成果物のみしか公開されておらず、保守のフェーズに含まれないオープンソースプロジェクトが 63 個存在した。そのうち API1.0 廃止後に新規公開されたオープンソースプロジェクトは 7 つであった。

残りの 111 個は 2 種類のバージョン以上の成果物が公開されていた。その中で API1.0 廃止後に新たに公開されたオープンソースプロジェクトは 7 つであった。残りの 104 個は、プロジェクト開発中に API1.0 の廃止がアナウンスされたプロジェクトであると考えられることができる。

『プロジェクト開発中に API1.0 の廃止がアナウンスされたプロジェクトである』と考えられる 100 個に対し分析を行った結果、11 個のプロジェクトがアナウンス後に更新を行っており、そのうち 6 つのプロジェクトが API1.1 への対応を行っていた。以下ではこの 6 つのプロジェクトにおける API1.1 への対応方法を説明する。

総数	238個
成果物なし	67個
成果物が1個のみあり API1.0廃止前で更新が停止	56個
成果物が1個のみあり API1.0廃止後に開発	7個
成果物が2個以上ありAPI1.0廃止前 で更新が停止	93個
成果物が2個以上ありAPI1.0とAPI1.1 の両方の期間に更新	11個
成果物が2個以上あり API1.0廃止後に開発	7個

図 2 プロジェクトの分類

5.3 対応内容の調査

成果物が 2 つ以上あり、API1.0 と API1.1 の両方の期間に更新があったアプリケーションで、API1.0 から API1.1 への変更に対応していたのは「Twichi」「characterbot」「OpenTween」「MusicTweetMessenger」「TwinLite」「Twitter4J」の 6 つのプロジェクトであった。

「Twitter4J」とは、TwitterAPI の Java ラッパであり、JSON や HTTP に詳しくなくても容易に TwitterAPI を使用するアプリケーションを作成する際に利用される。旧 API である API1.0 に対応するバージョンとして ver2.2.x までの開発が行われていたが、API1.0 廃止後の更新では、ver3.0.x としてメジャーバージョンを切り替えて API1.1 への対応が表現されていた。「Twitter4J」での、API1.0 から API1.1 への変更への対応は以下に示す。

- 「xml」から「JSON」への取得データの変更

- さまざまなエンドポイントの変更
- OAuth 認証必要化の対応

これらの変更とともに、「Twitter4J」における変更点として、Java1.4X サポートの廃止といった点がアナウンスされており、利用者はこれらの対応も求められた。

この「Twitter4J」を利用したアプリケーションとして「Twitch」と「characterbot」の 2 つのアプリケーションがある。これらの 2 つのプロジェクトは、「Twitter4J」が提供する jar ファイルを、ver2.0 台の jar ファイルから Ver3.0 台の jar ファイルへ利用する jar ファイルを変更することで、API1.0 から API1.1 への移行に対応していた。

残りの 3 つのプロジェクトは直接 TwitterAPI を呼び出す形で利用していた。「OpenTween」においては TwitterAPI1.1 への移行期間中に更新が行われており、「OpenTween」内でも移行期間中は TwitterAPI1.0 と API1.1 の利用コードが両方存在し、TwitterAPI1.0 の廃止後には API1.0 向けのコードが廃止されていた。

一方で「TwinLite」の API1.1 への対応は、API1.0 の廃止がアナウンスされてから 4 ヶ月ほどで対応しているが、API1.0 向けのファイルを直接書き換えられ、API1.0 向けのファイルはこの時点で無くなっており API1.0 と API1.1 の両方を利用できる期間は設けられていなかった。

また、「MusicTweetMessenger」は API1.1 への対応が API1.0 が廃止されて 4 日ほど経ってからの 2013 年 6 月に、API1.1 対応の Ver.1.0.3.3 がリリースされており、こちらも「TwinLite」と同様に、API1.0 と API1.1 の両方を利用できる期間は設けられていなかった。いずれの 3 つのプロジェクトも、取得データの変更とエンドポイントの廃止という 2 種類を変更し、API1.0 から API1.1 への変更に対応していた。

6 考察

前節の調査をもとに、調査項目として挙げられた 2 つの項目について考察を行う。

6.1 Q1: どれだけ TwitterAPI の移行に対応しているソフトウェアが存在するか

TwitterAPI1.0 の廃止が告知された時点で 104 個のオープンソースプロジェクトが保守の段階、もしくは保守後の休眠状態にあったと考えられ、これらのプロジェクトは、移行への対応がなされる可能性があったと考える。それらのプロジェクトのうち 6 つのプロジェクトが必要な対応を行っており、全体では約 6 % のプロジェクトが対応を行っていたことがわかる。関連研究 [3] では、セキュリティなどに問題を含むソースコードを取り込んだプロジェクトのうち、15 % がライブラリ側の更新に合わせて更新を行っていたという報告がある。今回与えられた結果は、API1.1 へ移行しないと機能が引き続き利用できないなど移行を強く促すものであったにも関わらず、この事例よりも低い値の対応率であった。値が低くなった原因として以下のこと

が考えられる。

- Twitter を利用したプロジェクトは、zlib などを利用するプロジェクトよりもコミュニティができていく可能性がある。「OpenTween」のようなユーザの多い Twitter クライアントについては、コミュニティがしっかりと形成され、きちんと対応がなされている。
- 一方で公開されていたプロジェクトを見てみると、プログラムを作ることを中心に行っているというよりは、『こういう機能を作成したいので人を集めている』などのプロジェクトがよく見られた。このように現状プログラム開発コミュニティが成り立っているプロジェクトが多くないということを感じた。
- TwitterAPI1.0 の廃止から約半年しか経過しておらず、まだプロジェクトの中に移行しきっていないものがあるかもしれない。
- 今後休眠していたプロジェクトについてユーザが定着し、保守が再び行われるようになれば、実際の対応率はもう少し高いモノになるかもしれない。

6.2 Q2: どのように対応しているか、また何か特別な処置を施しているか

実際に対応がなされた事例を考察すると、対応時に考慮すべき事項として、次のような方針が考えられる。

1. 「Twitter4J」のように外部から利用されることを想定したアプリケーションは、明確にバージョンを区切るなどの対応をもって利用者への周知を行うことが必要であると考えられる。
2. 「Twitchi」「characterbot」のような外部のラップクラスなどを利用している場合には、ラップクラスにおける対応も十分に理解した上で、新しく置き換えることが必要である。
3. 移行期間中に更新を行う場合は、新旧両方に対応できる形で対応しておく、移行についての問題が整理しやすくなるなどメリットがあると考えられる。
4. 旧バージョンが廃止された後に更新を行う場合は、すでに旧コードは意味を成さないもので、いきなり新コードに切り替わっても大きな問題にはならない。今後更新されるプロジェクトは単に切り替えるだけの対応で十分だと思われる。

6.3 オープンソースプロジェクトについての考察

オープンソースプロジェクトの場合、コミュニティが成立して開発が活発に行われているプロジェクトがある一方で、ソースコードが公開されているだけのプロジェクトも多かった。そのようなオープンソースのプロジェクトの場合は、利用者がメンテナンスを行う者になることが求められる。新しい機能を持つソフトウェア開発を行う際には、新規にプロジェクトを立ち上げてから行うか、既存のプロジェクトを利用して行うかを検討する必要がある。そのと

きに既存のオープンソースのソフトウェアを利用する際には、利用するソフトウェアがどの程度保守されているか、ソフトウェアの生存期間中にどの程度までメンテナンスする必要があるかを検討することがコストの見積もりとして必要不可欠であると考えられる。

6.4 保守作業において考慮すべきこと

ソフトウェアを保守する際には、ソフトウェアが置かれている立場に応じて考慮しなければならないことが生じることがある。例えば、そのソフトウェアを利用して他のソフトウェアが実現されている場合は、そのソフトウェアの利用者に対してアナウンスすることが重要な作業の1つとなる。例えば「Twitter4J」では廃止されるメソッドなどが告知されており、利用者が新バージョンに移行するためのガイドラインなどで合わせた告知が行われている。これらの告知は、利用者のスムーズな移行に必要な不可欠なものである。別な立場として、利用するライブラリが更新されるときに移行期間が設けられることがある。その移行期間中に更新を行うときには、問題の所在などを明らかにしやすくするために、「OpenTween」のように新しいライブラリに対応する部分と古いライブラリに対応する部分で共存させ、切り替えることができるようになることに価値がある場合もあると考えられる。ただし、これらの変更には設計変更などのコストが生じる可能性も高いので、どう変更すべきかを検討する必要があると考えられる。

7 おわりに

本研究では TwitterAPI1.0 が廃止された事例を題材に、どれだけプロジェクトが移行を行ったか、どのように移行が行われたかを調査した。調査した結果として、移行が行われたプロジェクトが約6%であるなど、関連研究での移行率より低い値が得られた。また、それぞれのプロジェクトごとに移行時のアプローチの仕方が異なるなどを確認することができた。オープンソースのプロジェクトを扱うときに考慮しなければならないこと、一般的な保守作業において考慮すべきことを提案し、近年のソフトウェア開発や保守における知見を得る。今後の課題としてプロジェクトへの聞き取り調査などを行い、なぜこのような結果になったかを詳細に分析することなどが挙げられる。

参考文献

- [1] 山本裕介:『TwitterAPI ポケットリファレンス』。技術評論社, 2011.
- [2] “Twitter”, <https://twitter.com/>
- [3] Pei Xia, “An Empirical Study of Out-dated Third-party Code in Open Source Software”, 大阪大学修士論文, 2013.
- [4] “SourceForge.JP”, <http://sourceforge.jp/>
- [5] S. L. プリーガー:『ソフトウェア工学』。ピアソン・エデュケーション, 2001.