

Ajax アプリケーション保守容易性のソフトウェアメトリクス

平山祐資^{†1} 小野康一^{†2} 深澤良彰^{†3}

我々は Ajax アプリケーションの保守容易性を測るソフトウェアメトリクスを定義するための研究を行っている。通常の Web アプリケーションを対象とする保守容易性ソフトウェアメトリクスは既に存在するが、Ajax の特徴を考慮したものはまだない。我々は実際に Ajax アプリケーションを作成し、保守を繰り返すことで保守容易性が変化する経過を観察し具体的データを得た。このデータをもとに回帰分析により保守容易性を測るための指標を求めた。

Towards software metrics to evaluate maintainability for measuring Ajax applications

YUSUKE HIRAYAMA^{†1} KOUICHI ONO^{†2} YOSHIAKI FUKAZAWA^{†3}

We are working on software metrics to measure maintainability of Ajax applications. There are a number of previous studies about software maintainability metrics of classical Web applications. However, they do not consider characteristic properties of Ajax applications. We obtained concrete maintenance data by observing and recording the progress of iterative modifications on an actual Ajax application through the software development process. It is expected that the software maintainability changes through the iterative modifications. We defined a suite of software maintainability metrics by a regression analysis based on those observed data.

1. はじめに

Web アプリケーションは大規模/複雑化の一途をたどっており、近年はページの一部を非同期通信で更新する Ajax (asynchronous JavaScript[™]+XML) が注目されている。

一般的に保守のコストはソフトウェア開発/保守の大部分を占め、アプリケーションの複雑度が高いほどコストも増す。また稼働直後は高い保守容易性 (Maintainability) をもっていたアプリケーションも、度かさなる保守によって、保守容易性が悪化することはよく知られている。そのためある時点におけるアプリケーションがどの程度の保守容易性を保っているかをソフトウェアメトリクス (以下メトリクス) を用いて示すことは重要である。メトリクスを用いて保守容易性を測る研究は昔から行われているが、Ajax を用いた Web アプリケーションを対象にしたものに関してはほとんど見られない。

本研究ではメトリクスを用いて Ajax アプリケーションの保守容易性計測のモデルを提案する。なお保守容易性を実効行 100 行あたりの保守時間と定義した。

2. 関連研究

前述したようにメトリクスに関する研究は昔から行われているが、Web アプリケーションに特化したものは少

なく、保守容易性を測る研究はさらに限定される。

Ghoshch らは UML[™] を Web アプリケーションの特徴に着目し、拡張したモデル (Conallen のモデル) からメトリクスを用いて保守容易性を測っている [1]。Cheh らは Web アプリケーションに CK メトリクスを適用、保守コストと相関がないことを示した [2]。以上述べた研究は全て Ajax を考慮しない Web アプリケーションのため本研究の方向性と異なる。

加賀谷らは Ajax アプリケーションに対してメトリクスを提案し、バグの数を複雑度と置き換え相関を出している [3]。本研究では保守容易性とメトリクスの関係式を導き出すことを目的とするため主旨が異なる。

3. Ajax と保守

以下では本論文で対象とする Ajax アプリケーションについて述べ、従来の Web アプリケーションとの違いを論ずる。その上で本研究が対象とする Ajax アプリケーションの保守について述べる。

3.1 Ajax

Ajax は JavaScript の HTTP 通信機能を使い非同期でサーバーと XML 形式 (プレーンテキストも可) のデータを通信できる。従来の Web アプリケーションはサーバーと通信している間 Web ブラウザ側は一切の操作ができなかった上、結果を得るためにはページ全体をリロードし直さなければならなかった。非同期で通信をすることによってシームレスな Web アプリケーションを作ることができる。Ajax を用いたアプリケーションの例として Google Maps[™] があげられる。

^{†1} 早稲田大学

Waseda University

^{†2} 日本アイ・ピー・エム株式会社東京基礎研究所

IBM Research - Tokyo

^{†3} 早稲田大学

Waseda University

3.2 Ajax アプリケーションの保守

Ajax は従来と違う機能を持つため実装方法も異なる。従来の実装方法ではサーバー側で記述する量が多く保守もサーバー側に変更の重きが置かれる。対して Ajax はクライアント側で実装する量が増え、必然的に保守もクライアント側で記述される量が増える。このため Ajax アプリケーションの保守容易性を測るには従来から用いられていたメトリクスのみでは不十分であり Ajax の特性を考慮する必要がある。具体的にはサーバー側のメトリクスとは別にクライアント側のメトリクスを考える必要がある。また Ajax は通信データの形式が XML かプレーンテキストにわかる。本研究ではプレーンテキスト形式のデータ通信を実装した Ajax アプリケーションに焦点を当て保守容易性を計測する。

4. 保守容易性メトリクス提案

本章では 2 つの事項について述べる。まず Ajax の特性を考慮したメトリクス及び従来から使われていたメトリクスで、本研究で用いるメトリクスについても述べる。次に実験として Ajax アプリケーションに保守を施して本研究で用いるメトリクスを適用、その結果から回帰分析を行い保守容易性計測の式を導く。以下にその流れを述べる。

4.1 考慮する指標

本研究では Ajax の特性を考慮するために保守容易性は「XMLHttpRequest オブジェクトの数」「非同期通信によって返ってきたデータ数」「イベントリスナの数」に依存するものとする。Ajax の処理の特徴は非同期通信によってページの一部を書き換えることである。その処理の中でのデータの流れを追うことでこれらのメトリクスを用いるに至った。またこれらのメトリクスのみで保守容易性を測ることは難しいため、従来から用いられているメトリクスとしてクライアント側の「ウィジェットの数」「Cyclomatic complexity」に加え、サーバーサイドに関しても「変数の数」「メソッドの数」「Cyclomatic complexity」を選んだ。ここでは混乱を避けるためサーバー側の「Cyclomatic complexity」を「Cyclomatic complexity2」と呼ぶこととする。

4.2 回帰分析による係数決定

保守容易性計測指標の式を提案するため、まず社員登録機能を持つ Ajax アプリケーションを作り、それに対して繰り返し保守を行い、4.1 で定義した指標でデータをとる。それらのデータをもとに回帰分析を行い、式を導出する。以下に導き出された式を示す。

$$-191.8 * (\text{XMLHttpRequest オブジェクトの数}) - 11.08 * (\text{非同期通信で返ってきたデータ数}) + 130.78 * (\text{イベントリスナの数}) + 20.17 * (\text{ウィジェットの数}) - 126.17 * (\text{Cyclomatic complexity}) - 34.85 * (\text{メソッド数}) + 60.95 * (\text{変数の数}) + 768.35 * (\text{Cyclomatic complexity 2}) - 2364.13.$$

現状データが少なく有意水準 (0.95) 以下で回帰式が受け入れられていない。今後さらなる保守を繰り返してデータを増やすことで有意水準を満たす式を導き出す必要がある。

5. 評価方針

現時点で想定している評価の方針について述べる。まず式の妥当性を確かめるために社員登録システムとは別に作成した社員検索システムに対して回帰式を適用、実際の保守時間と式で導かれた保守時間との相関を出す。また大規模システムに適用できるか確認するためオープンソースのアプリケーションにも適用する。実際の式をアプリケーション全体に適用してしまうと保守時間が爆発してしまうため変更があったファイルの中のメトリクスのみ適用する。

6. おわりに

Ajax アプリケーションの保守容易性を計測するために Ajax の特性に特化したメトリクスを用いて、保守容易性を保守時間で測ることが本研究の新規性である。しかし現状データが少なく回帰式は有意水準を満たしておらず式の妥当性はまだ検証できていない。

今後はデータを増やし回帰式を導く。同時に妥当性を検証及び考察を行う。その後導きだした式で考慮されなかった部分について見直し、式の精度を上げていきたいと考えている。また今回の実験においてクライアント側のメトリクスはすべて手作業で測り、式の計算も手動で行っていたためこれらをツールで測れるようにすることも今後の目標としたい。

7. 参考文献

- 1) Ghosheh, E. Black, S. and Qaddour, J. : Design metrics for Web application maintainability measurement, In proc. of ACS/IEEE International Conference on Computer Systems and Applications (AICCSA), IEEE Computer Society Press, pp. 778-784. (2008).
- 2) DiLucca, G. Fasolino, A. Tramontana, P. and Visaggio, C. : Towards the definition of a maintainability model for Web applications. In Proc. of the 8th European Conference on Software Maintenance and Reengineering. IEEE Computer Society Press. pp.279-287. (2004).
- 3) 加賀谷有紀, 海尻賢二, 海谷治彦 : Ajax のための Web 保守容易性メトリクスの提案, 電子情報通信学会技術研究報告. Vol. 111, No. 282. pp. 91-96. (2011)

商標

本稿中の会社名、製品名およびサービス名等はそれぞれ各社の商標である。