Webアプリケーション開発のための分散JavaScript言語

東京工業大学 情報理工学研究科
数理・計算科学専攻
学籍番号 09M37117　加藤 真人

指導教員　脇田 建 准教授

平成24年1月11日

目 次

図 目 次

第1章　はじめに

1.1　背景

Webアプリケーションとは、Webブラウザ上で動作するアプリケーションのことである。古くはHTMLのフォーム機能とサーバーサイドでのHTML生成を用い、ユーザーの入力によってページを動的に切り替えるものであり、単純なインターネット掲示板などがそれに該当する。これらのWebアプリケーションでは、ユーザーのフォームへの文字入力、及びボタン入力によって表示されるWebページの内容を変えることができる。そして最近ではブラウザ上で動作す言語であるJavaScriptの使用の活発化、及びHTML5を始めとするWebブラウザをとりまく環境の高度化により、実用的で視覚的にもリッチでかつインタラクティブなWebアプリケーションが多数登場している。その中にはWebブラウザ上でメールの送受信を行える高機能メーラであるGmailや、様々なデバイスで記録した情報を一挙に管理できるEvernoteなど、Webの特性を活かすことで従来のデスクトップアプリケーションを置き換えうるようなものもある。これらの新しいWebアプリケーションの特徴として、クライアントであるブラウザ上でJavaScriptのプログラムが動作し、それがユーザーのボタン入力などをトリガとしてサーバーのプログラムと動的に通信を行い、得られた結果を用いて動的にビューであるHTMLを書き換えるということがあげられる。これによって、より従来のデスクトップアプリケーションに近い、ユーザーの操作にすぐに反応し、必要な情報を瞬時に表示できるアプリケーションとなっている。また、FacebookやTwitterなどの、複数のユーザーが関与するアプリケーションもある。これらのアプリケーションでは、あるユーザーの入力が他のユーザーの視覚に影響すし、ユーザーのコミュニケーションを助ける。こういったリアルタイム性のあるアプリケーションも、昨今のWebアプリケーション事情の変化がもたらしたものであると考えられる。このようなWebアプリケーション事情はますます複雑高度化していくものであると考えられる。

本研究では、これらWebアプリケーションの開発に注目をする。昨今のWebアプリケーションは上述のとおり、ブラウザ上で動作するJavaScriptのプログラムと、サーバーで動作する他のプログラムがアプリケーションの実行に伴い動的に通信を行いながら動作する。そのため、その開発もクライアント用にJavaScriptでプログラムを記述し、サーバー用にまた別の言語でプログラムを記述する必要がある。さらにそれらを協調動作させるために、通信を意識したコーディングをする必要もある。このコーディングは本来作りたいアプリケーションのロジックとは異なる。Webアプリケーション開発には、このように面倒である点が存在し、プログラマの効率的

な開発の妨げとなる。そこで、これからの Web アプリケーション事情が更に発展していうく上でより効率的な開発をするための開発言語が必要であると考え、本研究では分散 JavaScript 言語 *OctopuScript* を提案する。

1.2 貢献

1.3 論文の構成

本論文では以下の内容を述べる. n 章では …

第2章　Webアプリケーションとその開発

この章では

2.1　Webアプリケーションとは

この節ではまず、Web アプリケーションとはどういうものか、実在する例を用いて解説する。

2.1.1　最も単純なWebアプリケーション

最も基本的な Web アプリケーションの開発形態は、サーバーサイドプログラムを用い、ユーザーの入力に応じて動的に Web ページを書き換えるものである。

2.1.2　動的に通信を行うWebアプリケーション

ブラウザ上で JavaScript がユーザー入力に応じて動的にサーバーサイドのプログラムと通信、ページ書き換えを行うことでよりインタラクティブな Web アプリケーションを作成することができる。

2.2　開発支援

2.2.1　フレームワークを用いたWebアプリケーション

前節の形態の Web アプリケーションが現在最も主流であるが、見てきたようにその開発は少々手間である。これらの手間を解決するために、大規模な開発ではフレームワークを用いるのが主である。ここではその内の幾つかを紹介する。

2.2.2 サーバーサイド JavaScript node.js

2.2.3 WebSocket

2.2.4 多言語から JavaScript への変換

第3章　分散JavaScript言語 OctopuScript

この章では

3.1　設計思想

3.2　概観

OctopuScript は、処理系 Octopus の上で動作する JavaScript のサブセットであり、Web アプリケーションを記述するために用いる。プログラマは OctpuScript を用いて、従来のようにサーバー用のプログラムと、ブラウザ用のコードを記述する。従来と大きく違うのは、それぞれのコードから他方のコードを自然に参照することができることである。例えば、以下のようなコーディングが可能になる。

コード 3.1: サーバー用の OctopuScript コード

```
1 var logging = function(logstr) {
2   console.log(logstr);
3 };
```

コード 3.2: ブラウザ用の OctopuScript コード

```
1 server.logging("Hello from client!");
```

サーバー用のコードはサーバー起動時に、ブラウザ用のコードは各ブラウザがサーバーにアクセスした時に実行される。このアプリケーションを動作させると、ブラウザがサーバーにアクセスするたびに、サーバーのログに”Hello, client!” という文字列が表示される。ここで、ブラウザ用コード中の server 変数はサーバー側の (logging 関数が定義されている) トップレベルを指しているとする。この例では、ブラウザからサーバーに定義された関数に、ブラウザで生成された文字列を引数として渡して実行しているが、

3.2.1　プログラム例

3.3　仕様

3.3.1　言語を構成する要素

3.3.2　式

3.3.3　構文

3.4　ライブラリ関数

第4章　分散JavaScript言語処理系 Octopus

この章では分散 JavaScript をどのように既存のブラウザ・サーバー上に実現するかについて論じる。

4.1　構成

4.2　コード変換

4.2.1　Rhino Ast ノード

4.2.2　標準形分散 JavaScript

4.2.3　Continuation Passing Style

4.3　補助ライブラリ

4.3.1　リモートオブジェクト・関数

4.3.2　制御構文関数

4.4　最適化

第5章　評価

この章では

5.1　主観評価

5.2　実行速度評価

第6章　関連研究

第7章　今後の課題

この章では