3L-5

Web アプリケーションのフロー記述フレームワーク

富田裕紀子† 段 希倫† 前田敦司‡
筑波大学大学院システム情報工学研究科† 筑波大学システム情報系‡

1. 目的

Web アプリケーションの一般化・高度化により，Ajax という技術が欠かせないものになっている．

しかし，Ajax を用いた Web アプリケーションには，サーバサイドのプログラムを記述した流れ通りに実行できないなどの問題点が存在する．本研究は，これらの問題を解決するために，開発者を支援するためのフレームワークを提案することを目的とする．サーバサイドのプログラムを分かりやすくするために，セッション全体をひとまとまりの処理として扱い，そのセッション中の処理を直感的に実際の流れ通りに書けるようにすることを目指す．本稿では，この流れ通りに書けることをフロー記述と呼ぶ．

2. 研究の背景

本研究に関連する既存の研究として，Ajax アプリケーションをサーバサイドのプログラミングのみで Java GUI を作成する感覚で記述することができる研究，GUI アプリケーションにおいて，一連のイベント処理をフロー記述できるようにする研究[2]，クライアント側の JavaScript を自動生成し，サーバ側のみのフロー記述によって Ajax アプリケーションを作成する研究[3] などがある．本研究では，Akka Actor の機能を用いるために，Scala をベースにした Play Framework でのフレームワークを実現する．

3. Web システムの問題点とフロー記述

Web アプリケーションでは，HTTP 通信の特性により，クライアントから見て一連の流れを持つ同一セッションの処理でも，基本的に，クライアントからのリクエスト毎に接続と切断を繰り返す点である．図 1のように，サーバとクライアントの通信は全て独立していて，サーバサイドではどのクライアントから何回目のリクエストを受け取ったかという情報は持たない．

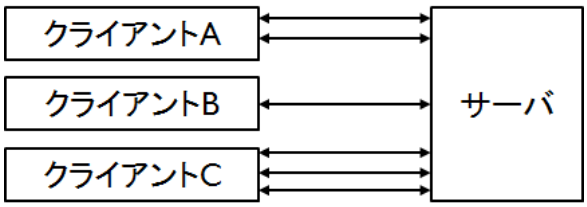

図 1 HTTP 通信

これを解決するためには，セッションごとに状態を管理し，セッションの状態によってアクセスするプログラムを振り分け，プログラムを複数箇所に記述することになるが，これでは全体の流れが掴みにくい(図 2)．

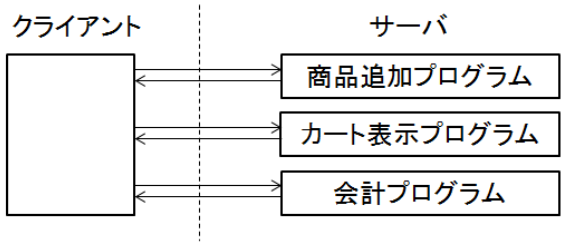

図 2 従来の Web アプリケーションプログラミング

そこで，プログラムの計算の続きをデータとして保存しておき，後からその情報を呼び出して，続きから処理を再開できるようにすることで，例えば図 3のようなプログラミングが可能となる．

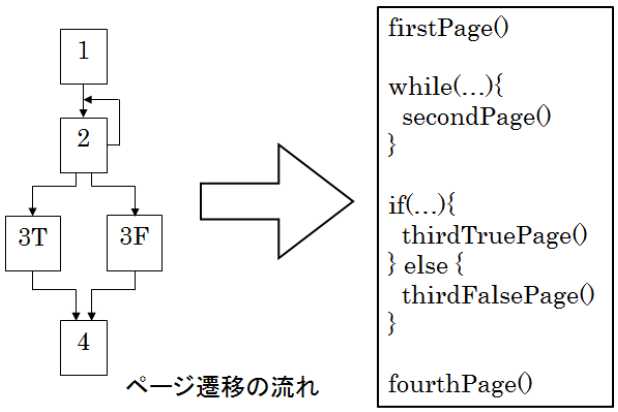

図 3 フロー記述のプログラムの例

4. Play Framework と Akka

Play Framework(以下 Play)は，Scala を扱うこ

Flow-based Framework for Web Applications
†Graduate School of System and Information Engineering, University of Tsukuba
‡Faculty of Engineering, Information and Systems, University of Tsukuba

とができる Web アプリケーションフレームワークである．最新版の Play2.2 では，Java，Scala，そして Akka をネイティブにサポートしている．Akka は Scala に統合されたフレームワークであり，耐障害性や負荷分散機能を持つ．Akka には，Akka Actor という機能が搭載されている．Akka Actor の最大の特徴はメッセージパッシングである．Akka Actor はそれぞれにメッセージキューを持ち，受け取ったメッセージの蓄積や，まとめての処理が可能である．

5. 提案フレームワークとフロー記述実装

本研究のフレームワークの概要図は図 4のようになる．ここでのワークフローとは，Web アプリケーションの処理が流れ通りに書いてあるプログラム本体のことである．まず，サーバサイドでクライアントからのリクエストとセッション情報を受け取ると，クライアントごとに Akka Actor を生成し，受け取った情報を対応する Akka Actor のメッセージキューに格納する．ワークフローが実行される際に，メッセージキューの中身を適宜取り出して処理を行う．そして，その処理の結果からクライアントサイドへと返すプログラムを生成する．

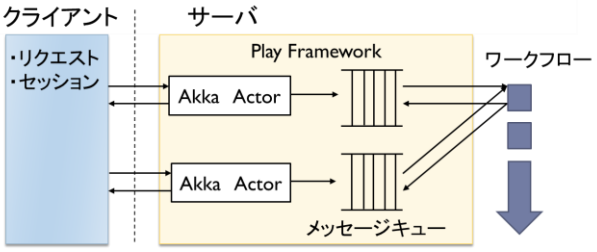

図 4　フレームワークの概要

フロー記述の実装にあたって，実際の内部構造は図 5のようになった．クライアントごとに生成した Akka Actor を制御するための Akka Actor を追加した．ここでは，クライアントごとの Akka Actor を seqActor，seqActor を制御する Akka Actor を callActor，ワークフローを Seq としている．

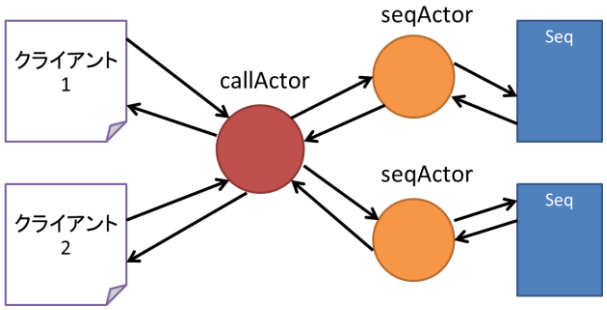

図 5　フレームワークの内部構造

フロー記述の実現のために，文の集合体である seq 文，ifThenElse 文，while の役割を持つ seqWhile 文を用いて制御構文を作成する．具体的には，seq，ifThenElse，seqWhile の各クラスと，ページを更新する文を格納するための show クラスを作成し，これらのクラスに exec メソッドを持たせる．exec メソッドは実行時に呼び出される．これらのクラスはすべて Stmt 型であり，その要素も Stmt 型とする．

6. 評価

設定ファイルなどを除いた，提案手法の実装の際に記述したコード量と，開発者が記述するコード量を比較した結果を表 1に，提案手法の実装の際に書き換えたファイル数と，開発者が書き換えるファイル数を表 2に示す．

表 1　コード量(Byte)

提案手法		従来手法	
総サイズ	記述サイズ	総サイズ	記述サイズ
9947	2103	5892	5150

表 2　ファイル数

提案手法		従来手法	
総ファイル数	記述ファイル数	総ファイル数	記述ファイル数
3	2	7	7

Intel Core i7 870 (4GB RAM)上の Windows7 Professiional で，サーバ側でリクエストを返す際の実行時間を表 3に示す．

表 3　実行時間

(msec)	提案手法	従来手法
ログイン→ログイン	6.26	2.52
ログイン→商品選択	7.00	2.51
商品選択→購入確認	7.13	1.55
商品選択→購入完了	7.82	0.84
購入確認→購入完了	6.50	1.55
平均	6.94	1.80

実用的な範囲での実行時間の増加と引き換えに，記述量の大幅な削減が得られている．

参考文献

[1] 石井 和仁，前田敦司，山口喜教，Ajax 実装のための JavaServlet 上のライブラリの提案．第60回情報処理学会 PRO 研究会．2006

[2] 小澤 史和．フロー記述のできる GUI イベント処理方式の提案．筑波大学大学院 システム情報工学研究科 CS 専攻 修士論文．2010．

[3] 藤原 佳己．Ajax のサーバ側制御構造を実現する Java ライブラリ．筑波大学大学院 システム情報工学研究科 CS 専攻 修士論文．2013