

## 科学研究費助成事業（学術研究助成基金助成金）研究成果報告書

平成 25 年 6 月 4 日現在

機関番号：32619

研究種目：若手研究(B)

研究期間：2011～2012

課題番号：23700039

研究課題名（和文） RIA 開発に適したアスペクト指向言語に関する研究

研究課題名（英文） A study on aspect-oriented language for RIA development

研究代表者

福田 浩章 (FUKUDA HIROAKI)

芝浦工業大学・工学部・准教授

研究者番号：30383946

研究成果の概要（和文）：本研究では、次世代型ウェブアプリケーションである RIA(リッチインターネットアプリケーション)を対象に、その開発を支援するアスペクト指向言語とウィーバの研究を行った。近年の RIA 実行環境は JavaScript+HTML の組み合わせと、ActionScript3+MXML の組み合わせという 2 種類が存在する。本研究では両者を視野に入れ、それぞれ環境でアスペクト指向言語とウィーバを実現した。まず、JavaScript+HTML の環境では、RIA の典型である動的コンテンツ、特にアニメーションの実行に着目し、その実行に必要なコードをアスペクトとして 1 つにまとめることができる言語の定義と、定義したアスペクトを既存のプログラムに合成するウィーバを実現した。一方、ActionScript3+MXML の環境では、頻繁に変更されるデザインと、イベントを処理するプログラムの混在に起因する開発時の問題点に着目し、それらを完全に分離して記述可能な言語拡張と、アプリケーション実行前にそれらを合成するウィーバを実現した。

研究成果の概要（英文）：In this research, we designed and developed Aspect-Oriented Language and the weaver for RIA(Rich Internet Application) which is one of the web applications for next generation. This research aimed at two RIA platform such as JavaScript+HTML and ActionScript3+MXML. First, we focused on dynamic contents especially animations, and provided an Aspect-Oriented Language, which enables developers to write all code in one aspect and its weaver for JavaScript+HTML environment. In addition, we also focused on the tangled code that makes it difficult to develop real applications in ActionScript3+MXML environment. Then we also provided an Aspect-Oriented Language, which enables developers and designers to work in completely separation manner, and its weaver for ActionScript3+MXML environment.

交付決定額

(金額単位：円)

|       | 直接経費      | 間接経費    | 合計        |
|-------|-----------|---------|-----------|
| 交付決定額 | 2,300,000 | 690,000 | 2,990,000 |

研究分野：総合領域

科研費の分科・細目：情報学・ソフトウェア

キーワード：プログラミング言語, アスペクト指向

## 1. 研究開始当初の背景

近年、計算機の性能向上とネットワークの高速化を背景に、Ajax(Asynchronous Javascript and XML)を代表とする Web2.0 系の技術を用いた RIA[1]がウェブアプリケ

ーションの主流になりつつある。従来型ウェブアプリケーションと比較すると、RIA では洗練されたデザインや操作性が重要になるため、RIA 開発ではロジックを実装する開発者と画面を設計するデザイナーとの協業が必

要になる。

RIA を含むウェブアプリケーション開発では、一般にデザイナーが専用ツール (Photoshop や Fireworks) を用いた画面の設計や、インタラクションの設計をした後、開発者がビジネスロジックやインタラクションの実装を行う。特に RIA では、“ボタンをクリックする” といったイベントを処理して作業を進める (イベントドリブン) ため、開発者はデザイナーが作成した画面に手を加え、イベントハンドラの定義やハンドラの呼出しを記述する必要がある。したがって、アプリケーション開発中、または運用中に画面の変更要求があると、デザイナーの作業だけではなく、ロジックの再実装や動作確認といった開発者の作業も必要になり、作業量が増加する。本研究は筆者が実現した RIA 開発環境の一つである Flex では、アプリケーション開発のために MXML (Macromedia Flex Markup Language), ActionScript3 という 2 種類の言語を備えており、デザイナーは主に MXML を利用して画面を作成し、開発者が ActionScript3 でロジックを実装していく。前述したように、Flex の開発では、開発者はデザイナーが作成した画面のソース (MXML) に手を加え、イベントハンドラの定義やハンドラの呼び出しを記述する必要がある。その結果変更に伴う改修が困難になる。さらに、開発の規模が大きくなるほど変更起因する作業は増大する。

これらの問題に対し、Dependency Injection と Convention over Configuration の導入、さらにイベントモデルを利用することによって、イベントハンドラとイベントの関連付けをソースコードで直接記述するのではなく、命名規約にもとづき実行時に関連付けることによって画面とロジックのソースレベルでの分離を可能にしているフレームワークも存在する。これらのフレームワークを利用した開発では、画面用コンポーネントに重複しない ID を付与し、命名規約に従ったイベントハンドラの定義だけでよいため、デザイナーの負担も少なく、容易に導入することができる。しかし、実用的なアプリケーション開発に導入すると、不十分な点も多い。例えば AspectFX では、イベントを処理する専用のクラス (Handler と呼ぶ) を用意し、画面で発生する各イベントに対してイベントハンドラを定義する必要がある。そして、画面と Handler は 1 対 1 で対応することを前提としている。この前提は管理アプリケーションなど、画面構成が単純な場合には問題とはならないが、一般ユーザを対象とし、操作性やデザインを重視する画面構成が複雑なアプリケーションでは厳しい制約とである。このように画面構成が複雑なアプリケーションでは、一般に MXML 自身をコンポーネ

ント化し、MXML の入れ子構造で 1 つの画面を構成する。そして、より内側の MXML で発生したイベントを外側の MXML に伝播させてまとめて処理することも多い。Flex 自身はこの仕組みを簡単に実現する機能を備えているが、AspectFX では MXML の入れ子構造を想定していないため、イベント通知の仕組みを開発者自身が実装する必要がある。また、イベントハンドラとイベントの関連付けを命名規約にもとづいて実行時に行うため、タイプミスなどによる規約違反が実行時にならないと判明せず、混乱の原因となることも判明した。そして、本来静的に決定できるイベントハンドラとイベントを実行時に解析して関連付けるため、実用上問題にはならないがオーバーヘッドがかかってしまう。

## 2. 研究の目的

- (1) AspectJ を参考に ActionScript3 の言語拡張を行い、AspectJ が持つ機能に加えてイベントの発生をジョインポイントとする拡張を行うことでこれまでの問題を解決し、アプリケーション実行前に 1 つにまとめ上げる手段を実現する。
- (2) 複雑な画面構成など、RIA 特有の特徴を考慮したアスペクト定義を行うための言語拡張を行う。

## 3. 研究の方法

本研究は 4 つのステップを踏んで進める

- (1) ActionScript3 の文法をコンパイラコンパイラを利用して定義し、プログラムの構文解析ができるパーザを実装す ActionScript3 は国際規格である ECMA-262 に準拠しているものの、オブジェクト指向への対応 (クラスやインタフェースの定義が可能) や、メタタグなど独自の拡張が行われている。ActionScript3 の正確な文法は公開されていないものの、Java 言語 (Java1.4) に類似している。そして、Java1.4 の文法は Sun Microsystems から公開され、コンパイラコンパイラで文法定義されているため、それをもとに拡張していく。また、Flex のコンパイラは Java で実装されており、将来的にはバイナリである SWF ファイルそのものを出力することを考慮して、本研究では Java で利用できる、Java1.4 の構文が提供されている字句解析に JFlex、構文解析に byaccj を利用する。また、完全な文法ではないものの、ANTLR で記述された文法が公開されているため、適宜参照して利用する。

- (2) 定義した文法をもとに、AspectJ の文法拡張を参考にしながら ActionScript3 をアスペクトの定義が可能な言語に拡張する。この拡張では、AspectJ を参考にするものの、研

究期間を考慮して AspectJ で可能な定義をすべて記述可能にするのではなく、必要なものだけを取舍選択を行う予定である。反対に、AspectJ では考慮されていない、イベントハンドラとイベントの織り込みや、MXML の入れ子構造に対応した本研究で新規となるアスペクト定義はを可能にする。また、前述したように、アスペクトとなり得る RIA 特有の事情を探るため、勉強会に参加して実際に現場で開発を行っている開発者に意見を求め、言語拡張に反映させる。そして、言語拡張後の文法に従ったプログラムを含め、プログラムから抽象構文木を作成するまでの実装を行う。

(3). アスペクトの織り込みを行うウィーバの実装を行う。具体的には、通常のプログラムとアスペクトのプログラムを抽象構文木に変換し、アスペクト中に記述されるポイントカット、アドバイスの定義をもとに構文木のノードをマージする。そして最後に構文木から織り込み後のソースプログラムを出力する。同様に、画面 (MXML) に対する織り込みも、MXML を XML の木構造に、アスペクトを抽象構文木に変換し、XML の木構造を操作してイベントハンドラを呼び出すプログラムを挿入することで実現し、最後にソースプログラムを出力する。

(4). 実装したウィーバを利用し、実際に簡単なシステムを実装して動作の確認、および不具合の修正を行う。さらに、勉強会に参加して成果物を紹介し、実際に開発を行っている開発者の生の意見を得たいと考えている。本研究の成果としては、織り込みを行う時間といった定量的な評価をとることはもちろん、どの程度複雑なアプリケーションに対応できるかという、定性的な評価を行う。

#### 4. 研究成果

本研究の成果物は、(1)Flex (ActionScript3) を対象としたウィーバ、(2)JavaScript を対象とした動的コンテンツの再生に必要な HTML, CSS, Javascript を 1 つのアスペクトファイルまとめ、それらを既存のプログラムに織り込むウィーバ、(3)JavaScript ライブラリの 1 つである jQuery におけるメソッドチェーンの織り込みを可能にしたウィーバとランタイムライブラリ、である。また、これらに関する国内会議での発表 2 件と、国際会議論文誌 2 編の発表を行った。

本研究の計画段階では RIA の実行環境として Flash (Flex) を想定していたが、スマートフォンの普及、特に iPhone のブラウザで Flash が動作しないこと、HTML5 や WebSocket といった Web の新しい技術の出現を考慮し、HTML+Javascript での RIA も対象に研究を進

めた。そして、JavaScript を用いた RIA の典型である動的コンテンツ、特にアニメーションの実行に注目し、その実行に必要なコードをアスペクトとして 1 つにまとめるための実装を行った。具体的には、動的コンテンツの再生に必要な HTML, CSS, Javascript を 1 つのアスペクトファイルにまとめて記述できる言語拡張を行い、Javascript, HTML, CSS をそれぞれ解析し、抽象構文木を生成してアスペクトと既存のプログラムを合成するウィーバを実装した。本実装によって、従来は 1 つのアニメーションを実行するために、HTML, CSS, Javascript の 3 箇所に別々に記述していたプログラムコードを、1 つのアスペクトとして記述できるため、モジュール性が高まり、アニメーションの追加、変更、削除に伴うプログラムの修正コストを抑えることができる。また、織り込みにかかる時間も十分実用可能な範囲であることを実験によって確認している。またアプリケーション実行前に織り込むため、実行速度に与える影響はない。さらに、この仕組みを発展させ、動的なアニメーションを作成するときによく利用される jQuery に対し、アスペクトを織り込む機構 (Aspect-jQuery) を実現した。jQuery を用いた開発ではメソッドチェーンを多用し、それらのメソッドは順番に実行される。開発者はこの特徴を利用し、動的なアニメーションを容易に作成できるが、振る舞いを変更するには呼び出すメソッドや順序を変更する必要がある。既存の JavaScript 用 AOP フレームワークでは、メソッドの呼び出し順序をポイントカットに指定する手段が提供されていないため、アニメーションの一部の振る舞いをアスペクトとして分離することが難しかった。Aspect-jQuery では、新たに between ポイントカットを導入してメソッドチェーンに容易にアスペクトを織り込む機構を実現した。

一方、当初予定していた Flex 環境においても、ActionScript3 の言語拡張と、ウィーバの開発を行った。Flex では JavaScript と異なり、GUI コンポーネントが予め用意されている。そのため、画面 (GUI) のコードとロジックを実現するコードを分離して記述することができるが、RIA はイベントドリブンで動作するため、イベントを捉えるコードを GUI のコードに追加する必要があり、完全に分離する事は難しい。この問題に対し、様々なフレームワークが提供されているが、これらは全て実行時の情報を利用して DI (Dependency Injection) を行う。そのため、遅延生成されるオブジェクトには DI することができないという問題、また他のフレームワークと干渉するなどの問題が発生する。そこで本研究では、静的な情報を利用し、コンパイル時に GUI とロジックを関連付ける機構

を実現した。具体的には、GUI とビジネスロジックを分離するため、“eventcut” というポイントカットを用意し、イベントを処理するプログラムにイベントの処理対象である GUI 部品の指定を可能にすることで、GUI とビジネスロジックの完全分離を実現した。また、多層的な構造を持つ GUI 部品に対応するため、inner アスペクトという独自の機構を設け、現実的なアプリケーションでは多用され、従来システムでは対応が難しかったアプリケーションの構造であっても GUI とビジネスロジックを分離することを可能にした。

## 5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[学会発表] (計 4 件)

- ① Kohei Nagashima, Hiroaki Fukuda and Shingo Takada, Aspect-jQuery: An Aspect-Oriented Framework for jQuery, A0Asia/Pacific2013: 8<sup>th</sup> International Workshop on Advanced Modularization Technique, Fukuoka, Japan, Mar. 25 2013.
- ② Hiroaki Fukuda, AOP based language extension for web development, 12<sup>th</sup> IASTED International Conference on Software Engineering, Innsbruck, Austria, Feb. 13 2013.
- ③ 永島康平, 福田浩章, 山本喜一, DCWeaver: 動的な Web コンテンツを考慮したアスペクト指向言語の提案と実装, 日本ソフトウェア科学会ソフトウェア工学基礎ワークショップ, 青森, 2011 年 11 月 25 日.
- ④ 永島康平, 福田浩章, 山本喜一, 動的コンテンツを考慮したアスペクト指向言語の提案, 日本ソフトウェア科学会 第 28 回全国大会, 沖縄, 2011 年 9 月 27 日

## 6. 研究組織

### (1) 研究代表者

福田 浩章 (FUKUDA HIROAKI)  
芝浦工業大学・工学部・准教授  
研究者番号: 30383946