

機関番号：32612

研究種目：若手研究（B）

研究期間：2009～2010

課題番号：21700040

研究課題名（和文） デザイナーと開発者の協業を考慮したアプリケーション開発を支援するフレームワーク

研究課題名（英文） A framework for supporting collaborations between designers and developers.

研究代表者

福田 浩章（HUKUDA HIROAKI）

慶應義塾大学・理工学部・助教

研究者番号：30383946

研究成果の概要（和文）：本研究では、近年普及しつつあるリッチインターネットアプリケーション(RIA)開発において、デザインとロジックをソースコードレベルで完全に分離して記述することができるフレームワークの開発を行った。本フレームワークは、デザイナーと開発者の協業がより重要となる RIA 開発において、デザインとロジックの関連を疎結合とすることでお互いの並行作業を可能とし、デザインや仕様変更に伴うテストや修正を最小化することで開発効率の向上に貢献する。

研究成果の概要（英文）：In this research, we developed a framework which separates view from logic in source code level in RIA development. The framework is able to make the relationship between design and logic loosely coupled for RIAs where the collaborative works are more important than traditional web applications. By introducing this framework, developers and designers are able to concentrate on their tasks and the development/maintenance cost of RIAs will be reduced.

交付決定額

（金額単位：円）

	直接経費	間接経費	合計
2009 年度	1,200,000	360,000	1,560,000
2010 年度	900,000	270,000	1,170,000
年度			
年度			
年度			
総計	2,100,000	630,000	2,730,000

研究分野：アスペクト指向ソフトウェア工学

科研費の分科・細目：

キーワード：フレームワーク，Dependency Injection

1. 研究開始当初の背景

インターネット、特にウェブアプリケーションは我々の生活に必要な不可欠な要素の1つになりつつあり、特に近年はウェブアプリケーションの利便性を保持しつつ、デスクトップアプリケーションの操作性を実現するリッチインターネットアプリケーション(RIA)が普及し始めている。従来のウェブアプリケーション開発では、仕様に従った機能を実現することに時間を費やしてきたが、それに加

え RIA 開発ではデザインの要素や操作性が重要視されるため、プログラムを記述する開発者と、画面を設計するデザイナーとの協業がより重要視されている。RIA 開発では、アプリケーションの画面とロジックを関連付ける必要があるが、そのためにはデザイナーが作成した画面に開発者が手を加える必要があり、画面や仕様の変更に伴う修正やテストによって著しく開発効率が低下する。この原因は、開発者とデザイナーという専門分野が異なる

る両者が、画面のソースコードをお互いの目的のために修正せざるを得ないという製薬があるためである。

2. 研究の目的

(1) 画面とロジックの関連をソースコードで記述するという直接的な方法ではなく、間接的な方法で行うことによって疎結合にする。具体的には、Dependency Injection と、命名規則にもとづく規約を導入し、ソースコードレベルで画面とロジックを完全に分離して記述することを可能とする。

(2) ソースコードレベルで分離した画面とロジックを実行時に結合し、1つのアプリケーションとしてまとめ上げる機能をフレームワークとして開発者とデザイナーに提供する。

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) RIA 全般およびアスペクト指向に関する技術的な調査を行う。特に、本研究で対象とする AIR や Flex, Flash に特有の処理や、Flash をコンポーネントとして利用する場合の扱いについて詳細に調査し、フレームワーク全体の設計を行う。また、アスペクト指向に関しては、さまざまな文献や、国際会議の論文などを参考に、最近の動向や技術を調査し、フレームワークの設計に役立てる。RIA 開発プロセスで実際に発生している問題点の調査には、Flex や AIR の勉強会や、Flash の勉強会に参加し、実際に開発を行っている開発者の意見を取り入れる。なお、本研究では対象外としているが、Silverlight や WPF の開発でも画面の設計を XAML と呼ぶマークアップ言語で記述し、プログラミング言語でロジックを実装するという Flex や AIR と酷似した開発スタイルを採用しているため、勉強会に参加して技術的な動向や開発時の問題点を調査する。

(2) 調査結果をもとに、主要な機能だけを実装したプロトタイプシステムの開発を行う。実際に実装してみると、設計時には発見できなかった問題点や、仕様変更せざるを得ない事態が起こるが、一度実装してみることでそれらの問題点を洗い出し、フレームワークとして動作する状態まで仕上げる。また、本システムが有効か否かを判別するためには、複数の開発者やデザイナーに利用してもらう必要があるため、オープンソースとして公開し、広く意見を集める。そして、簡単なアプリケーション開発に利用して適用範囲を明確にする。

(3) プロトタイプをアプリケーション開発に適用した結果をもとに、不足した機能や仕

様を見直し、本格的なフレームワークの設計と実装を行う。実装においてはプロトタイプで実現した機能に加え、Flash コンポーネントと連携する機能や、本格的なアプリケーション開発では必須である検証機能、MVC アーキテクチャでの開発を支援する機構を導入し、本格的なアプリケーション開発に耐えるものに仕上げる。

そして、Adobe 社で提供されている Flex の実用的なアプリケーションに対して本フレームワークを適用し、フレームワークの適用可能性を検証する。

(4) 本フレームワークの適用結果をもとに評価を行う。評価のポイントは、定性的な評価として仕様変更やデザインの変更に伴うプログラム変更の有無や、デザイナーと開発者の作業の独立性である。従来のアプリケーション開発では、画面のデザインが決まらなければ開発者がロジックを実装できない場合があるが、本フレームワークを利用することで仕様が確定した段階でお互いが完全に独立して作業を進めることができると予想している。また、定量的な評価として、本フレームワークを導入した場合に生じる実行時のオーバーヘッドや、デザインや仕様変更に伴い変更が必要なソースコードの量である。

最後に、本研究の成果物としてフレームワークを公開するとともに、フレームワーク自身や得られた知見を国際会議や論文誌に投稿して研究成果を公開する。

4. 研究成果

本研究の成果物は、主に実際に利用できる Flex を対象とした DI コンテナライブラリ、および開発した DI コンテナの新規性、有効性を記した国内論文誌 2 編、国際会議論文誌 3 編である。現在、Java 言語を中心に多くの DI コンテナが開発され、公開されているが、それらは開発するアプリケーションの設計と実装を分離することを主たる目的としている。また、Flex を対象とした DI コンテナも幾つか開発されているが、先の例と同様に多くは設計と実装の分離を目的としている。それらに対し、本研究で実現した DI コンテナはデザイナーと開発者の協業がより重要となる RIA 開発において、設計と実装の分離ではなくデザインとアプリケーションのロジックを分離することを主たる目的としており、その点が既存システムとは大きく異なる。この特徴を主張した国際会議での発表を評価された結果、SES2009 では招待講演を依頼され、本研究で取り組む課題や成果を産業界や研究者に広く発信することができた。

また、技術的な視点からの本研究の貢献は次の 2 点に集約される。

まず、従来は RIA 開発における画面のソー

スコードに手を加え、イベントハンドラの記述を直接ソースコードで記述していたが、本研究では命名規則などの規約を導入し、それらの関連付けを実行時に行うことで画面とイベントハンドラをソースコードレベルで完全に分離することを可能にした。また、実用的なアプリケーション開発を通じ、実行前に静的に決定できる画面とイベントハンドラの関係は完全に分離できるものの、実行時にしか決定できない場合には適用できないこと、そして実用的なアプリケーションではこのような状況が十分にありうることを明らかにした。本研究では開発者に API を提供することでこの問題を解決している。

次に、前述したアプリケーション開発を通じ、RIA ではよく利用されるアニメーションの開始や終了など、アニメーション制御用のメソッド呼出しが至る所に散財し、アニメーション制御の追加や修正の負担となっていることを発見した。そこで本研究では、アニメーションの制御がイベントの発生や終了に深く関わることに着目し、アニメーション制御に新たな命名規約を導入した。そして、命名規約に基づく制御用メソッドを、フレームワークから呼び出す事によってメソッド呼出しをソースコードから取り除くことによってこの問題を解決している。

一方、本研究で開発した DI コンテナは、実行時にデザインとアプリケーションのロジックを結合させるため、アプリケーションの構造解析やイベントとイベントハンドラの間接付けにオーバーヘッドがかかる。そこで、評価アプリケーションを作成し依存関係にかかる時間の測定を行った。この測定では、デザインである画面に配置するコンポーネントの個数を 5 個から 30 個まで 5 個ずつ増やし、DI コンテナを利用する場合と利用しない場合でアプリケーションの起動に要する時間を測定した。その結果を図 1 に示す。図 1 に示すように、DI コンテナを利用するか否かにかかわらず、コンポーネントの個数が増加するにつれて起動時間も増加している。また、コンポーネントが 5 個の場合には起動時間の差が約 50(ms)であるのに対し、30 個の場合では約 130(ms)と、配置するコンポーネントの個数が増加するにつれて起動時間も増加している。これは、アプリケーションのロジックであるイベントハンドラの解析に要する時間であり、イベントハンドラが増加するほどその差が顕著に現れる。しかし、DI コンテナを利用せず、コンポーネントを 1 つも配置しないアプリケーションであっても起動に 1,400(ms)程度時間がかかることを確認しているため、計算機の高速度という近年の状況を考慮すると、DI コンテナの導入によるオーバーヘッドは実用上問題にならないと考えている。しかしながら、1 つの画面に多

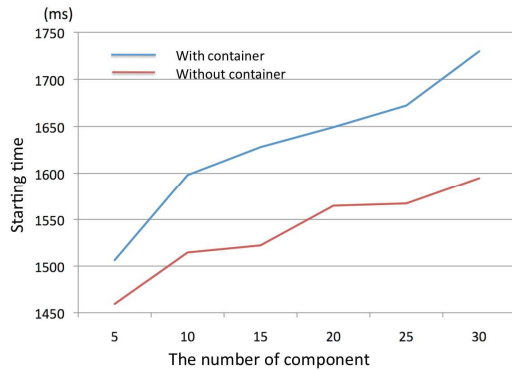


図 1 コンポーネントの個数と起動時間の比較

数のコンポーネントが配置されるアプリケーションに DI コンテナ利用する場合には、画面を分割するなどして起動時のオーバーヘッドを削減する工夫が必要となる。なお、起動後は通常のアプリケーションとして動作するため、DI コンテナがアプリケーションに与えるオーバーヘッドはない。

また、本研究で新たな問題も明らかになった。本研究ではイベントとイベントハンドラは 1 対 1 で対応させ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なった。このように画面構成が複雑なアプリケーションでは、一般に画面の一部をコンポーネント化し、それらの入れ子構造で 1 つの画面を構成している。そして、内部のコンポーネントで発生したイベントを外側のコンポーネントに伝播させ、まとめて処理することも多い。本研究ではこのような状況を想定できておらず、これまでの方針では対応が難しい。また、命名規約にもとづき、実行時にイベントとイベントハンドラを関連付けるため、タイプミスなどによる規約違反を実行前に判定することが出来ず、混乱の原因となることもある。

そこで、今後は RIA の特徴を踏まえた上、アスペクト指向技術を取り入れたアスペクト指向言語を開発し、これらの問題に対応していく予定である。

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)

[雑誌論文](計 2 件)

- ・ 福田浩章, 山本喜一, モジュールの独立性を考慮した RIA 開発フレームワーク, 日本ソフトウェア科学会コンピュータソフトウェア, 査読有, 2010, pp. 97-113.
- ・ 福田浩章, 山本喜一, AspectFX: アスペ

クト指向による RIA 開発での協業を支援するフレームワーク,電子情報通信学会論文誌 D,査読有,Vol. J93-D, No. 7, 2010, pp. 1165-1178.

〔学会発表〕(計4件)

Hiroaki Fukuda, Yoshikazu Yamamoto, A framework for supporting collaborative works in RIA by aspect oriented approach, In Proceedings of 12th International Conference on Enterprise Information Systems, 10/6/2010, Portugal.

Hiroaki Fukuda, Yoshikazu Yamamoto, Yet another aspect in Rich Internet Applications, In Proceedings of Asian Workshop on Aspect-Oriented Software Development, 16/11/2009, New Zealand.

Hiroaki Fukuda, Yoshikazu Yamamoto, Modularity oriented framework for Rich Internet Application, In Proceedings of IASTED International Conference on Software Engineering and Applications, 3/11/2009, U.S.A.
福田浩章,デザインとの協業を考慮した RIA 開発フレームワーク,情報処理学会ソフトウェアエンジニアシンポジウム,2009年9月9日,東京

〔図書〕(計0件)

〔産業財産権〕

出願状況(計0件)

名称:
発明者:
権利者:
種類:
番号:
出願年月日:
国内外の別:

取得状況(計0件)

名称:
発明者:
権利者:
種類:
番号:
取得年月日:
国内外の別:

〔その他〕

ホームページ等

6. 研究組織

(1) 研究代表者

福田 浩章 (HUKUDA HIROAKI)
慶應義塾大学・理工学部・助教
研究者番号: 30383946

(2) 研究分担者

該当なし

(3) 連携研究者

該当なし