

氏名	松塚 貴英
授与学位	博士(工学)
学位授与年月日	令和3年9月27日
学位授与の根拠法規	学位規則第4条第1項
研究科専攻	秋田県立大学大学院システム科学技術研究科 博士後期課程総合システム科学専攻
学位論文題目	
指導教員	教授 <u>堂坂 浩二</u>
論文審査委員	主査 教授 <u>堂坂 浩二</u> 副査 教授 <u>西垣 正勝</u> (静岡大学 創造科学技術大学院) 准教授 <u>草苺 良至</u>

論文内容要旨 ※(20ポイント)

※用紙体裁：A4 縦、左右余白 20mm、上下余白 30mm、1 行 45 文字、40 行

※文字：MS 明朝 10.5 ポイント

※文字数：4,500 文字程度

近年、グローバル化をはじめとてますます激しくなるビジネス環境の変化により、企業を取り巻く不確実性が増している。そのような環境において、変化に対応していくことは企業にとっての重要課題である。特に、Web アプリケーションは顧客や従業員などの利用者によって直接利用されるため、特にこのような変化の影響を受けるソフトウェアである。具体的には、近年のスマートフォンをはじめとしたデバイスの普及により、クライアントの種類も動作環境もきわめて多様になっており、動作中の環境変化も激しくなっている。これに対し、従来の Web アプリケーションではサーバーが事前に設定した対応しかできないため、様々な環境変化に対応しきれないという問題がある。これらの変化に対応するには、ソフトウェアコードに手を入れることなく、実行中に機動的に対応できる仕組みが必要である。一方で、実行中の対応には限界があり、大きな変化に対してはソフトウェアを修正して対応することになるため、ソフトウェアコードが高い保守性を持つ必要がある。従来 Web アプリケーションはサーバーとクライアントでフレームワークが異なっており、開発者は両方のフレームワークを習得しなければならないだけでなく、クライアントとサーバーの連携をフレームワークの組み合わせごとに定義しなければならない。するとクライアントの変化に対応するための連携機能も個別実装になり保守性が大きく悪化してしまうという問題がある。

これらの問題に対し、本研究においては、従来サーバーおよびクライアントにて別々に実装されているフレームワークの構造をモデル化して統合し、同じ概念で双方のフレームワークを実装できるよう

にする。これによりクライアントとサーバーの連携方法をはじめとして一貫性が確保され、保守性と柔軟性を高めることができる。また、クライアントからの環境情報に合わせ、実行中にサーバーが対応を変化させられる仕組みを導入する。このとき、環境変化が当初予期していた範囲を超えていた場合でも、サーバーがその変化に対してより抽象度の高い解釈をして対応することができることが特徴である。また、大きな対応が必要な場合はソフトウェアコードの修正を行う必要があるため、実行中に対応を変化できる仕組みをフレームワークに組み入れることで保守性を高める。

本論文は、次の6章で構成される。

第1章では、背景を述べ、Webアプリケーションが抱える問題と、それに対する課題を議論したうえで、本研究の目的を述べる。ソフトウェアが環境や要件の変化に対応するには、ソフトウェアを一度止め、ソースコードを変更する静的対応と、実行中に動作の設定や構造を変えることで動作を変更する動的対応の2種類に大別される。Webアプリケーションにおいては、静的対応としてフレームワーク技術が利用されるが、クライアントとサーバーという2種類の問題領域が存在するため、それぞれ個別にフレームワークが開発されている。すると開発時には開発者がそれぞれに対して熟達しなければならず、学習コストや保守コストが高くなることが問題である。そのため、これらのフレームワークを統合することを1つ目の課題とする。また、動的対応においては自己適応技術が知られているが、開発者がフレームワークとは全く別の技術として取り扱わなければいけないという問題がある。このことから、Webのフレームワークに動的対応を追加することを2つ目の課題とする。本研究の目的は、これら2つの課題を解決するために、フレームワークモデルの提案をすることである。

第2章では、関連研究について検討し、本研究の位置づけを述べる。まず、静的対応においては、要求レベル、仕様レベル、コードレベルで様々な研究がある中で、Webアプリケーションにおいては静的対応と動的対応を統合するという観点から、主にコードレベルに着目することについて述べる。次に、動的対応の技術として自己適応技術について検討し、自己適応技術は管理対象サブシステム、管理サブシステムから構成されること、変更のメカニズムがパラメーター変更と構造変更で大別されることを述べる。本研究では、静的対応と動的対応の両方に対応するために、Webアプリケーションフレームワークへの自己適応技術の統合を行うこととした。

第3章では、フレームワークモデルの提案を行う。Webアプリケーションはフレームワークとの親和性が高いため、静的対応・動的対応を一つのフレームワークとして統合できることが望ましいが、現実にはサーバーとクライアントでは利用されるプログラミング言語や動くプラットフォームが異なり、完全に同一な実装のフレームワークをサーバーおよびクライアントの双方で動作させることはできない。このことから、クライアントおよびサーバーにおけるフレームワークを一段抽象的な概念でモデル化し、同じモデルからサーバーとクライアント双方のフレームワークを実装できるようにするアプローチを取る。

まずサーバーフレームワークとクライアントフレームワークを統合するために、双方のメカニズムの共通点から処理の流れを7段階のステップに整理する。各ステップはデータ入力、データ抽象化、データチェック、ロジック選択、ロジック実行、画面作成、画面表示から構成される。ロジック選択ステップでは、呼出判別表という定義体を利用してクライアントのリクエストをロジック呼び出しにマッピングしている。処理を行う際の入出力データとその処理の相互依存関係を解消し、画面表現と入出力オブジェクトの組み合わせを部品としてアプリケーション内で再利用ができるようにする。

次に、自己適応技術の統合を行う。Web アプリケーションにおいてはクライアントからの要求に対してどのようなロジックを実行するかが、もっとも変化が多く本質的な部分となるため、ロジックの選択を動的に変えることができれば、様々な環境の変化に対応することができるようになる。そのため、管理対象サブシステムとして条件項目を追加した呼出判別表を利用し、管理サブシステムとして学習・更新モジュールを追加することで自己適応技術に対応させた。

また、入出力オブジェクトにクラス階層を導入する。これは、変化する状況下においてあらかじめ想定した処理では対応できない場合に、段階的に縮退することを可能にするためである。入出力オブジェクトは、スーパークラスにより一般的な内容が設定され、サブクラスにより個別の内容が設定される。クライアントからの情報が個別の情報であればあるほど、特定の状況に合わせたロジックを呼び出すように設定することができる。しかし、クライアントからの情報がサーバー側で想定していた値の範囲に収まらなかったり、値が欠損していたりした場合は、その特定の状況に合わせたロジックを選択することができない。代わりに、そのような場合はスーパークラスに対応したロジックを呼び出す、すなわちサーバーがより抽象度の高い対応をすることで、クライアントの状況に対して次善の策を取ることができる。このスーパークラスへの縮退は再帰的に行われるため、最終的にどのクラスでの対応も不可能であった時も、最終的に全てにマッチするエントリーを呼出判別表に用意することで例外対応をすることができる。

本モデルに対する考察として、以下について議論する。保守性に対しては、コード量、関心の分離、再利用性という観点に関して効果があることを示す。次に本フレームワークモデルで導入している呼出判別表は、プログラミング言語のメソッド呼び出しに相当するが、条件指定とオーバーロードという二点において拡張されていることを示す。また、動的対応のレベルを自己適応技術に照らし合わせて議論する。本モデルにおいては、Foreseen, Foreseeable という 2 種類の変化について対応することができる。さらに、本モデルの Web アプリケーション以外への適用可能性を検討する。フレームワークのコア部分はリアクティブな動作メカニズムであることから、Web アプリケーション以外のユーザインタフェースを実装するフレームワークにも適応可能であるほか、マルチエージェントシステムへの応用に可能性がある。

第 4 章では、モデルによるフレームワークの実装例として、サーバー型フレームワーク、クライアント型フレームワーク、適応型フレームワークの 3 種について実装とその評価を行う。

サーバー型フレームワークにおいては、Servlet/JSP 環境を利用した実装フレームワークについて述べる。本フレームワークモデルに従って実装することで、画面の部品化、再利用ができるようになった。また、ロジックの実装量が大幅に削減されるとともに関心の分離を達成し、開発効率及び保守性を上げることができた。

クライアント型フレームワークにおいては、HTML/JavaScript 環境における実装フレームワークを示す。クライアント型においては画面動作をきめ細かくコントロールする必要があるため、フレームワークモデルの画面作成ステップにおいて、画面部品をウィジェットと機能付加部品に分割し、様々な機能を持つ画面部品を容易に拡張できる枠組みを作る。これにより、画面部品を組み合わせで増やすことができるようになったほか、計測においてロジックの実装量が大幅に削減される効果を確認した。適応型フレームワークにおいては、試作したフレームワークにおいて、人材交流イベントを支援するサービスを事例として効果を確認する。人材のマッチングを行うためのロジック切り替えを参加者人数に応じた最適な応答時間で行うため、呼出判別表を動的に更新する仕組みを導入し、その効果を確認した。また、呼出判別表のクラス階層の仕組みにより、応答時間があまりに長いと予測される場合

に簡易的なマッチングロジックへ縮退する機構を実現した。

第5章では、フレームワーク移行について述べる。いったんあるフレームワークを採用してアプリケーションを開発すると、他のフレームワークに移行するのは困難である。この点は、他のフレームワークから本モデルをベースとしたフレームワークに移行する際も同様である。この問題を解決するため、モデル駆動型開発を用いたアプリケーションのフレームワーク間の移行技術を開発・実証する。モデル駆動開発は、Web アプリケーションの仕様をプラットフォームに独立な形で形式的に記述することで、様々な実装フレームワークに展開することができるようになるという考え方である。ここでは、まず Web アプリケーションの仕様記述方法である WebWare 意味モデルを開発する。次に Struts フレームワークに準拠した検証用アプリケーションを用意し、研究用に作成したリバースエンジニアリングツールで検証用アプリケーションに対応する WebWare 意味モデルを導出する。さらに、WebWare 意味モデルから本章のサーバー型フレームワークに準拠したアプリケーションを生成することで、フレームワーク間の移行ができることを検証する。

第6章では、本研究のまとめを行い、今後の課題について述べる。

論文提出者氏名	松塚 貴英
論文題目	環境及び要件の変化に対応する Web アプリケーションフレームワークの研究
指導教員	堂坂 浩二
論文審査委員	主査 教授 <u>堂坂 浩二</u> ㊞ 副査 教授 <u>西垣 正勝</u> ㊞ (静岡大学 創造科学技術大学院) 准教授 <u>草苺 良至</u> ㊞

論文審査結果要旨

Web アプリケーションは、90年代の World Wide Web から現在の IoT 時代に至るまで、様々な用途に使用されてきたインターネット利用の基幹インフラである。Web アプリケーションが利用されるビジネス分野では、開発者はシステムの要件や動作環境の変化に対応し続ける必要がある。こうした変化への対応策として静的対応と動的対応がある。静的対応とはソフトウェアのソースコードを変更することでシステムの要件の変化に対応し、動的対応とは実行中にソフトウェアの設定や構造を変えることでシステム動作環境の変化に対応するものである。従来、静的対応はフレームワーク技術により扱われてきた。フレームワークとは、アプリケーションにおいて共通に使われるコードを整備したソフトウェア部品群である。Web アプリケーションはサーバーとクライアントから成るが、従来技術では、ソフトウェアの構造が異なるサーバーとクライアントに対して共通のフレームワークでは対応できず、実装や学習のコストが大きいという課題があった。また、従来のフレームワーク技術は動的対応を考慮しておらず、開発効率が悪いという課題があった。本論文は、これらの課題を解決し、静的対応と動的対応の双方を実現する Web アプリケーションを開発するためのフレームワークモデルを提案し、その有効性を明らかにすることを目的とする。本論文は全 6 章で構成されている。

第 1 章の序論に続き、第 2 章では従来のフレームワーク技術の課題を論じている。第 3 章では、課題を解決するため、コマンド、入出力オブジェクト、呼出判別表という抽象化した概念でフレームワークをモデル化したフレームワークモデルを提案した。提案モデルからサーバー・クライアント双方のフレームワークを実装することが可能となり、さらに、環境をデータで表し、データの変化に応じて実行する処理を切り替えることで動的対応を扱うことが可能となった。第 4 章では、サーバー型、クライアント型、適応型のフレームワークについて、提案モデルによるフレームワークの実装例を示し、開発効率、保守性ならびに環境変化に対する応答時間の観点からモデルの有効性を実証した。第 5 章では、他のフレームワークから本モデルのフレームワークに移行するための方法を示した。第 6 章は結論である。

以上、本論文は、数多くの Web アプリケーション開発経験をモデルに昇華させ、新奇な Web アプリケーションフレームワークモデルを提案したという学術的貢献だけでなく、企業で実際に使用される技術を開発したという工学的価値も高い。また、本研究課題と関連する研究業績として、査読付き学術誌論文 3 編、査読付き国際会議発表 5 件、国内学会発表 6 件を公表している。

よって、本論文は博士（工学）の学位論文として合格と認める。