

Title	フレームワークの進化に対応可能なWebアプリケーション作成手法の研究
Author(s)	呉, 暁雷
Citation	
Issue Date	2013-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/11327
Rights	
Description	Supervisor:鈴木 正人, 情報科学研究科, 修士

修 士 論 文

フレームワークの進化に対応可能な Web アプリケーション作成手法の研究

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

呉 暁雷

2013 年 3 月

修 士 論 文

フレームワークの進化に対応可能な Web アプリケーション作成手法の研究

指導教官 鈴木正人 准教授

審査委員主査 鈴木正人 准教授

審査委員 青木利晃 准教授

審査委員 緒方和博 准教授

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

1010207 呉 暁雷

提出年月: 2013 年 2 月

概要

本論文では、フレームワークの進化に対応する Web アプリケーションの変更手法を提案する。フレームワークの進化によって、フレームワークの構造や仕組みが変更された。このため、アプリケーションの実装を変更すべきである。ただし、アプリケーションの変更を行うには開発者はフレームワークの進化前後の両方の知識に精通していなければならない、変更作業が困難になっている。したがって、従来は仕様から要求分析を行った結果をユースケースなどで記述していたが、提案手法では、要求を構成する要素と実現を構成する要素及び要素の対応関係の3種類の構造に注目し、要求－実現－関連グラフを作成することを提案する。また、グラフの変更を実装の変更に変化することにより、フレームワークの変更をアプリケーションの変更に正しく反映させることができるフレームワークの進化に対応する Web アプリケーションの体系的な変更手法を提案する。

目次

第1章	はじめに	1
1.1	背景.....	1
1.2	目的.....	1
1.3	本論文の構成.....	2
第2章	フレームワークを使用した開発とその問題点	3
2.1	フレームワークとその役割.....	3
2.2	フレームワークの進化とアプリケーションへの影響.....	4
第3章	進化を考慮した手法	6
3.1	要求構成グラフ.....	6
3.2	実現構成グラフ.....	8
3.3	要求構成グラフと実現構成グラフの関連.....	9
3.4	フレームワークの進化への対応.....	10
第4章	要求－実現－関連グラフに基づくアプリケーションの実現法	11
4.1	要求構成グラフの作成.....	11
4.2	実現構成グラフの作成.....	12
4.3	要求－実現－関連グラフの作成.....	13
第5章	適用事例	14
5.1	要求構成グラフ.....	15
5.2	Struts1 を使用する場合の実現構成グラフ	22
5.3	Struts2 を使用する場合の実現構成グラフ	27
5.4	相違分析と変更規則.....	30
5.5	変更規則の評価.....	33
5.5.1	機能追加要求構成グラフ	34
5.5.2	機能追加実現構成グラフ	35
5.5.3	評価	39

第6章	評価	40
6.1	評価.....	40
6.2	考察.....	40
第7章	おわりに	41
7.1	まとめ.....	41
7.2	今後の課題.....	41
付録A	ホテル予約管理システムの全体のユースケース記述	44

目次

2.1	フレームワークを用いた開発のイメージ.....	3
2.2	フレームワークの進化に伴うアプリケーションの変更.....	5
3.1	要求構成グラフの例(ユーザー登録)	7
3.2	実現構成グラフの例(ユーザー登録)	8
3.3	要求－実現－構成グラフの例(ユーザー登録)	9
4.1	要求構成グラフの作成例	12
4.2	実現構成グラフの作成例	12
5.1	ホテル予約管理システムのユースケース図	15
5.2	ホテル予約管理システム要求構成グラフ(ユーザー登録).....	16
5.3	ホテル予約管理システム要求構成グラフ(ユーザー認証).....	17
5.4	ホテル予約管理システム要求構成グラフ(予約作成)	19
5.5	ホテル予約管理システム要求構成グラフ(予約照会)	20
5.6	ホテル予約管理システム要求構成グラフ(解約)	21
5.7	Struts1 を使用する場合の実現構成グラフ(ユーザー登録).....	23
5.8	Struts1 を使用する場合の実現構成グラフ(ユーザー認証).....	23
5.9	Struts1 を使用する場合の実現構成グラフ(予約作成).....	24
5.10	Struts1 を使用する場合の実現構成グラフ(予約照会).....	25
5.11	Struts1 を使用する場合の実現構成グラフ(解約).....	25
5.12	Struts2 を使用する場合の実現構成グラフ(ユーザー登録、認証)	28
5.13	Struts2 を使用する場合の実現構成グラフ(予約作成).....	28
5.14	Struts2 を使用する場合の実現構成グラフ(予約照会).....	29
5.15	Struts2 を使用する場合の実現構成グラフ(解約).....	30
5.16	実現構成グラフ全体(Struts1).....	31
5.17	実現構成グラフ全体(Struts2).....	31
5.18	実現構成グラフの変更パターン(その 1)	32
5.19	実現構成グラフの変更パターン(その 2)	32

5.20	予約照会機能追加の要求構成グラフ.....	35
5.21	Struts1 を使用する場合の機能追加実現構成グラフ(1).....	36
5.22	Struts1 を使用する場合の機能追加実現構成グラフ(2)	36
5.23	Struts2 を使用する場合の機能追加実現構成グラフ.....	38

表 目 次

3.1 ユースケース記述の内容	7
5.1 実現構成要素の一覧(Struts1)	26
5.2 追加と変更の実現構成要素の一覧(Struts1)	37
A.1 ユースケース記述(ユーザー登録する)	44
A.2 ユースケース記述(ユーザー認証する)	45
A.3 ユースケース記述(予約作成する)	46
A.4 ユースケース記述(予約照会する)	47
A.5 ユースケース記述(解約する)	47

第 1 章 はじめに

1.1 背景

近年、Web アプリケーションは、オンライン・ショッピングや電子掲示板など、Web を利用したサービスにおいて幅広く利用されている。また、アプリケーションがますます大規模化し、アプリケーションの構築は決して容易ではない。しかも、Web アプリケーション開発案件は多くの場合、大変な短納期である。ことため、最近 Web アプリケーションの開発を効率的に行うために、フレームワークを利用した開発に注目が集まっている。しかしながら、フレームワークを構成する要素技術は変化が激しくて、フレームワークは、機能拡張及び変更を継続的に行っている。フレームワークの進化によって、フレームワークの構造や仕組みが変更された。このために、旧バージョンのフレームワークで作成されたアプリケーションはそのまま新バージョンのフレームワークでは動作しないという問題が発生している。この際には、フレームワークの進化によるアプリケーションの変更が正しく反映する必要がある。しかし、変更を行うには開発者は進化前と進化後の両方のフレームワークに精通していなければならない、変更作業のコストが増大している。

本研究では、フレームワークの進化に対応する Web アプリケーションの変更手法を提案する。

1.2 目的

本研究の目的は、フレームワークの進化に対応するアプリケーションの変更を支援し、変更コスト削減するためのアプリケーションの構成手法の提案を行う。変更手法として、アプリケーションの要求構成要素と実現構成要素及び要素の対応関係の 3 種類の構造に注目し、要求の構造をグラフで表現したものを要求構成グラフ、実現の構造をグラフで表現したものを実現構成グラフ、要求要素と実現要素の関連を表現したものを要求－実現－関連グラフとして定義する。また、そのグラフの上でフレームワーク

の進化に伴うアプリケーションの変更箇所の特特定とその変更操作を自動的に抽出することを目的とする。

1.3 本論文の構成

本論文の構成を以下に示す。

- 第2章では、フレームワークを使用した開発とその問題点を提起する。
- 第3章では、2章で挙げた問題点に対し、本研究ではどのようなアプローチをとるか説明する。
- 第4章では、第3章で提案した「要求－実現－関連グラフ」を用いてアプリケーションを具体的に実現する方法を説明する。
- 第5章では、提案手法の有効性を確認するために適用事例を説明する。
- 第6章では、提案手法の評価と考察を述べる。
- 第7章では、本研究についてまとめ、今後の課題を述べる。

第2章 フレームワークを使用した開発とその問題点

2.1 フレームワークとその役割

フレームワークとは、アプリケーションを開発する際に頻繁に必要とされる汎用的な機能をまとめて提供し、アプリケーションの土台として機能するソフトウェアのことである。Webアプリケーションフレームワークとは、Webアプリケーションの骨組み、枠組みとなり、より具体的に定義づけをすればWebアプリケーションを構築する上で元となる土台の部分を提供するものである。たとえば、入力データの検証の仕組みと画面遷移制御の仕組み及びビジネスロジックを呼び出す仕組みなどを提供する。そのため、Webアプリケーションフレームワークを使用してアプリケーションを開発する時に、開発者は画面と業務固有部分の作成に集中できる。図 2.1 にフレームワークを用いた開発のイメージを示す。

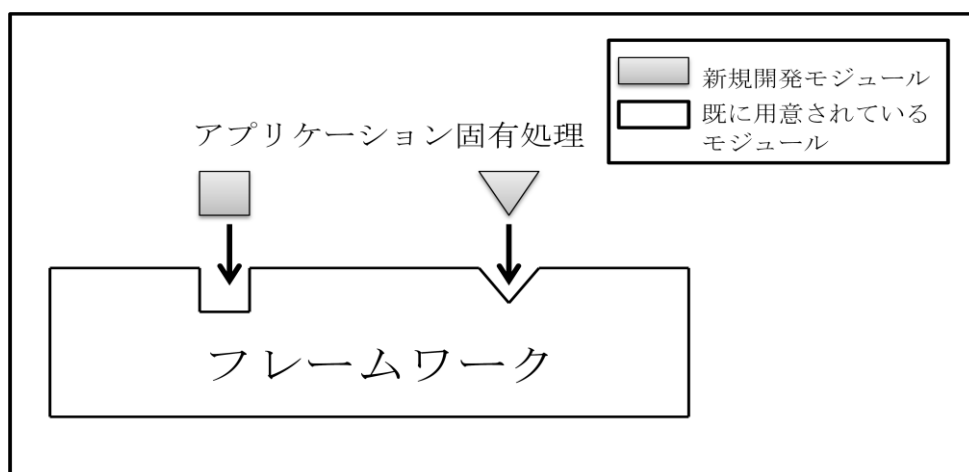


図 2.1: フレームワークを用いた開発のイメージ

Web アプリケーションフレームワークを導入することによって、アプリケーションを開発するメリットがある。

- 開発効率の向上

Web アプリケーション開発には短納期のものが多いため、工数を減らすことが必要となる。しかし、基本的に Web アプリケーションの開発においては、非常に手間がかかる。開発者はアプリケーションの全体を作成しなければならない。それに対して、フレームワークを導入すれば、Web アプリケーションで共通となる機能(画面遷移、入出力など)を提供してくれるため、開発者は毎回似たような機能を実装する必要がなくなり、開発工数を他のビジネスロジックの実装に割り当てることができるようになり、開発効率が向上する。

- 保守性の向上

Web アプリケーションフレームワークを使用して開発を行っていれば、ある程度ルール化されたコードによって開発されたものとなっているため、その Web アプリケーションフレームワークを知っている人であれば、誰でも短時間で実装を理解できる。機能拡張やバグ改修が必要となった場合でも、ビジネスロジックの設計、実装に注力することができるため保守性が向上する。

2.2 フレームワークの進化とアプリケーションへの影響

現在、Web アプリケーションの規模がますます増大し、顧客のニーズを反映するために新機能を追加されることが多い。また、Web アプリケーションフレームワークを構成する要素技術は変化が激しい。このために、Web アプリケーションフレームワークは継続的に機能拡張及び変更が行われている。それはフレームワークが進化していると呼ばれる。

フレームワークの進化によって、フレームワークの構造や仕組みが変更された。たとえば、フレームワークの決めている設定ファイルの記法や意味が変更された。また、業務固有処理や画面の実装方法のルールが変更された。その原因で、フレームワークの進化に伴うアプリケーションの実装が変更されるため、アプリケーションの構造が変更された。このために、旧バージョンのフレームワークで作成されたアプリケーションはそのまま新バージョンのフレームワークでは動作しないという問題が発生している。この際には、アプリケーションの変更に正しく反映する必要がある。ただし、変更を行うには開発者はフレームワークの進化前後の両方の知識に精通していなければならない、アプリケーションの変更箇所を特定することが非常に難しく、変更作業が困難になっ

ている。たとえば、今までの **Struts** における作成された **Web** アプリケーションに対して、フレームワークを変更すると、アプリケーションを実装する時のコントロール層とモデル層の部分を変更しなければならない。

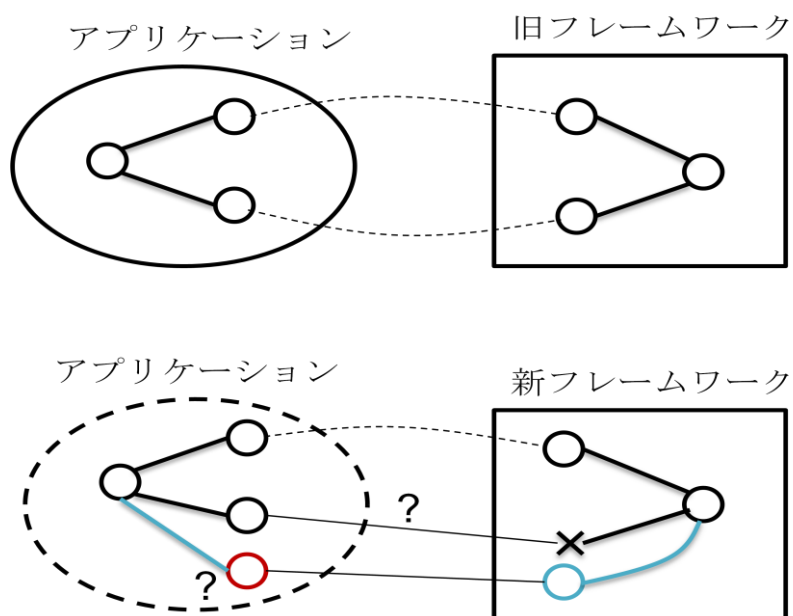


図 2.2: フレームワークの進化に伴うアプリケーションの変更

フレームワークが変更されることを継続的に行っている。このために、フレームワークが変更されるたびにアドホックな修正を繰り返すとアプリケーションの変更を継続的に行わなければならない。その結果、アプリケーションの構造がますます複雑し、保守のコストが増大する。そのために、フレームワークの進化に対応する体系的な変更手法が必要である。

第 3 章 進化を考慮した手法

本章では、前の章で述べた問題を解決するためにフレームワークの進化を考慮した開発手法を提案する。従来は仕様から要求分析を行った結果をユースケースなどで記述していたが、提案手法では、(1)要求を構成する要素、(2)実現を構成する要素、(3)要素の対応関係の 3 種類の構造に注目し、要求－実現－関連グラフを作成する。要求－実現－関連グラフ (Requirement－Realization－Relationship Graph, R3Graph) とは、要求構成要素と実装構成要素の間に存在する 1 対 1 や 1 対多などの関連を辺として持つグラフである。

3.1 節では要求構成グラフについて、3.2 節では実装構成グラフについて、3.3 節では要求構成グラフと実装構成グラフの関連について、3.4 節ではフレームワークの進化への対応について詳しく記述する。

3.1 要求構成グラフ

本節では、要求構成要素について説明する。要求構成要素は最初に要求分析を行い、主に機能要求の詳細を定義することで得られる。要求分析とは、システム開発の対象となる業務を調査し、開発するシステムの利用方法について分析することである。要求分析では、要求仕様からシステムに必要な機能を列挙し、UML のダイアグラムの 1 つである「ユースケース図」およびユースケース記述として定義する。ユースケース記述とは、ユースケース図におけるユースケースとシステムの利用者であるアクターとのやり取りを文章で記述するものである。ユースケース記述には必ず 1 つの主シーケンスと複数の例外的な状況のための代替シーケンスが必要である。各シーケンスにおける必要なデータおよびデータベース操作などを決定する必要がある。シーケンスにおける各ステップに対し、必要とするデータおよびそれを処理する機能単位を要求構成要素とする。表 3.1 はユースケース記述の内容である。

表 3.1: ユースケース記述の内容

ユースケース名	ユースケース図のどのユースケースと対応するか
アクター	このユースケースを利用するアクター
前提条件	このユースケースを開始する前に満たすべき条件
主シーケンス	このユースケースの通常の流れ
代替シーケンス	例外時の流れ
事後条件	このユースケースが終了した後で満たされるべき条件

Web アプリケーションで一般的に用いられる「ユーザー登録」というユースケースを例とする。主シーケンスと代替シーケンスは以下のように記述される。

主シーケンス:

1. ユーザーはユーザーID とパスワードをシステムに入力する。
2. システムはユーザーID とパスワードを利用者 DB に登録する。
3. システムは登録成功メッセージをユーザーに表示する。

代替シーケンス:

2.でユーザーID が登録済みの場合

=> A1. システムは ID 重複メッセージをユーザーに表示する。

A2. システムは再入力を促すため入力画面を表示する。

図 3.1 では、要求構成グラフの例(ユーザー登録)を示す。

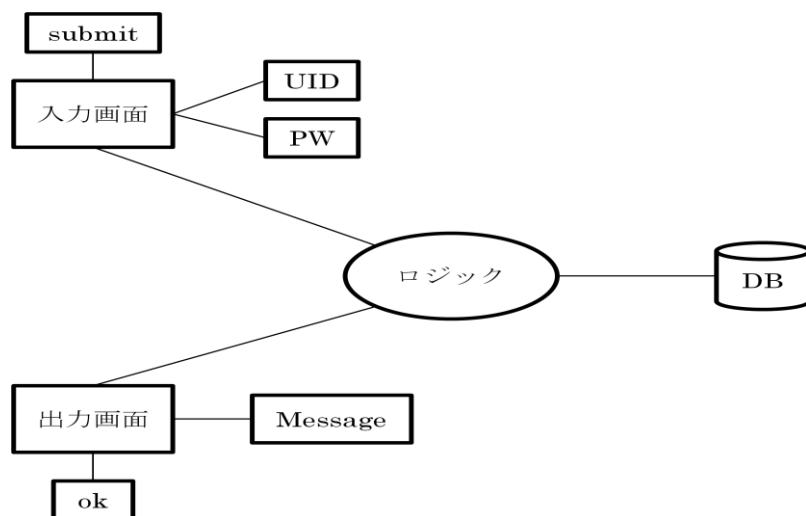


図 3.1: 要求構成グラフの例(ユーザー登録)

3.2 実現構成グラフ

実現構成要素とは、アプリケーションを構成する部品であり、個々の部品は要求を満たすために必要な機能あるいはデータ実体を表すものである。フレームワークを使用してアプリケーションを開発する時に、そのフレームワークによるプログラムの配置場所と命名規則に従うことによって、開発者は必要とする部品を容易に得ることができる。

前節のユーザー登録の機能を実現する場合を考える。フレームワークを使用しない場合には、以下のものを作成する必要がある。

- 入力画面: ユーザーID とパスワードの入力欄および送信ボタン
- 出力画面: 成功または失敗のメッセージの表示欄および終了ボタン
- 処理: ユーザーID の重複検査と利用者 DB への登録

図 3.2 では、実現構成グラフの例(ユーザー登録)を示す。

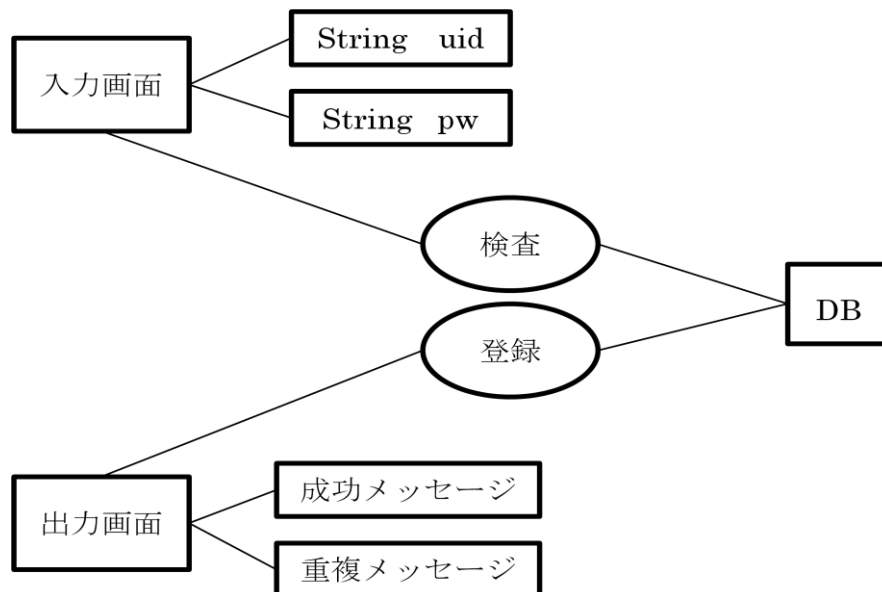


図 3.2: 実現構成グラフの例(ユーザー登録)

3.3 要求構成グラフと実現構成グラフの関連

要求構成要素に示される個々の機能要求が実現構成要素に示される部品の集合によって実現可能であることを明らかにするために、要求構成要素の各項目と実現構成要素の各項目の関連を定義する。関連は 1 対 1 と 1 対多及び多対 1 の 3 種類がある。

ユーザー登録の例の要求－実現－関連グラフを図 3.3 に示す。1 対 1 の関連は黒い線で接続する。1 対多の関連は赤色線で接続する。

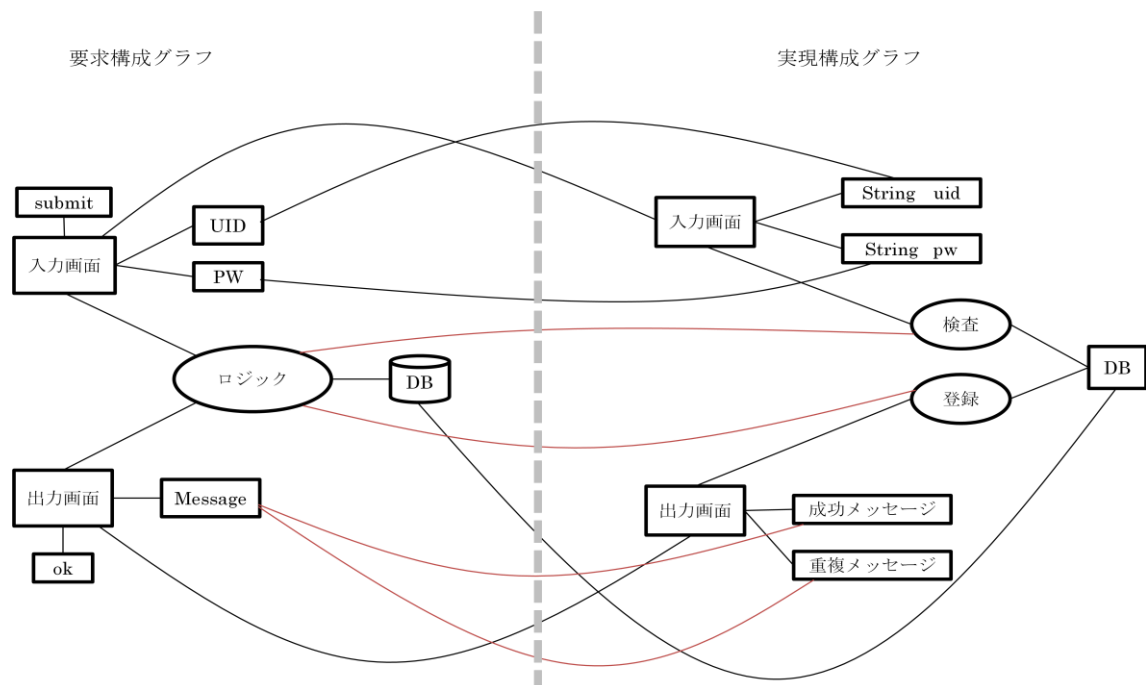


図 3.3：要求－実現－関連グラフの例(ユーザー登録)

3.4 フレームワークの進化への対応

あるフレームワークを使用してアプリケーションを開発し、フレームワークが進化すれば、アプリケーションを実装する時に必要な部品が変わった。たとえば、Web アプリケーションフレームワークである **Struts** はバージョン 1 からバージョン 2 に進化した時に、情報の保持(モデル)方法が変わりました。**Struts1** では、モデル部分の実装には、「**JavaBeans**」が用いられる。**Struts2** では、モデル部分の実装は、特別なクラスの継承やインタフェースの実装を必要としなくて、**POJO**(Plain Old Java Objects)として作成することが可能である。

そのため、要求構成要素が不変の場合、異なるバージョンのフレームワークにおける作成した要求－実現－関連グラフが異なる。その異なる要求－実現－関連グラフに対して、グラフの変更が必要となる箇所を発見しやすい。そして、変更規則を決定する。したがって、グラフの変更を実装の変更に変化することにより、フレームワークの変更をアプリケーションの変更に正しく反映させることができると考える。

第 4 章 要求－実現－関連グラフに基づくアプリケーションの実現法

本章では、第 3 章で提案した「要求－実現－関連グラフ」を用いてアプリケーションを具体的に実現する方法を説明する。4.1 節では要求構成グラフの作成について、4.2 節では実装構成グラフの作成について、4.3 節では要求－実現－関連グラフの作成およびそれに基づくアプリケーションの作成について述べる。

4.1 要求構成グラフの作成

要素構成要素を頂点、関連を辺としてグラフで表現したものが要求構成グラフである。まず、全ての要求構成要素を明確に名前を付ける必要がある。次に要素を分類する。異なる種類の要素を頂点として異なる色の四角形で表現する。特に、アクター要素が名前を付いた人型アイコンで表現する。そして、要素の関連を辺として黒い実線で表現し、グラフになる。以下は、要求構成要素の分類と表示色を定義する。

- － アクター要素： 名前を付いた人型アイコンで表現
- － データ要素： 黒い四角形で表現
- － シーケンス要素： 緑色四角形で表現
- － データベース要素： オレンジ四角形で表現

図 4.1 は、要求構成グラフの作成例を示す。

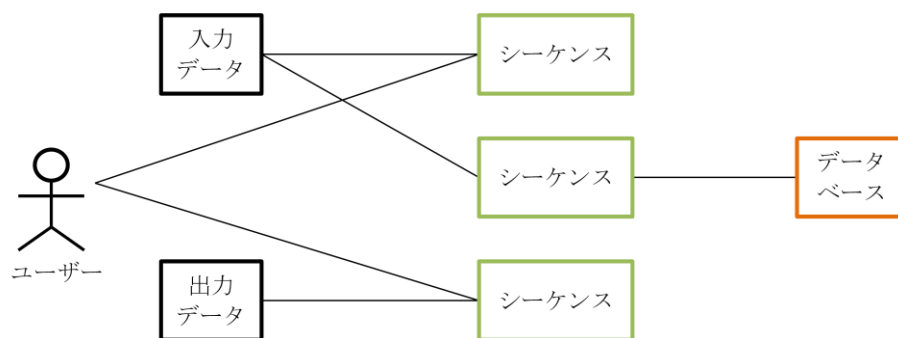


図 4.1: 要求構成グラフの作成例

4.2 実現構成グラフの作成

実現構成要素を頂点、関連を辺としてグラフで表現したものが実装構成グラフである。実現構成要素の表現法は 4.1 節で記述した手順と大体同じように実施する。要素の関連も辺として黒い実線で表現する。以下は、実現構成要素の分類と表示色を定義する。

- ビュー層要素： 黒い四角形で表現
- コントローラ層要素： 緑色四角形で表現
- モデル層要素： オレンジ四角形で表現

図 4.2 は、実現構成グラフの作成例を示す。

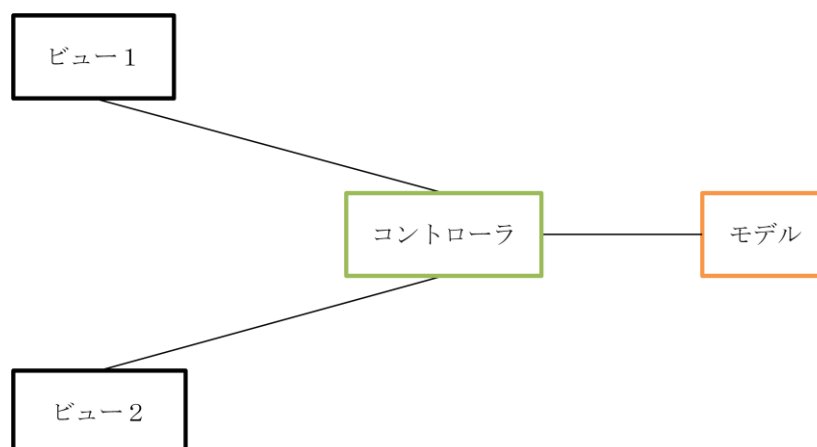


図 4.2: 実現構成グラフの作成例

4.3 要求－実現－関連グラフの作成

要求構成グラフと実現構成グラフを要素の対応関係により接続したものが要求－実現－関連グラフである。関連は3つの種類があると考え、頂点と頂点の関連、辺と辺の関連および頂点とサブグラフの関連である。異なる種類の関連を接続する時に異なる色の太実線を用いる。

要求－実現－関連グラフを作成したことにより、アプリケーションを構成する部品であり、個々の部品は要求を満たすために必要な機能あるいはデータ実体を得られる。

第 5 章 適用事例

本章では、提案手法の有効性を確認するために適用事例としてホテル予約管理システムを対象とする。使用するフレームワークは **Struts** とする。まず要求構成グラフを作成し、**Struts1** における実装構成グラフと **Struts2** における実装構成グラフを比較し、変化した部分を得ることを可能にする要求－実現－関連グラフの変更の規則を発見する。

5.1 節では要求構成グラフについて、5.2 節では **Struts1** を使用する場合の実装構成グラフについて、5.3 節では **Struts2** を使用する場合の実装構成グラフについて、5.4 節では相違分析と変更規則について、5.5 節では変更規則の評価について順番に記述する。

対象とするホテル予約管理システムの仕様：

- 顧客はシステムを利用する前にユーザー登録する。
- ユーザー登録際には、ユーザーID とパスワードを入力する。ユーザーID 重複ならば、重複エラーメッセージを表示し、再入力を促す。
- 顧客はユーザー認証際に、登録したユーザーID とパスワードを入力する。入力されたものが登録されているものと一致したら、認証に成功する。失敗したら、認証失敗エラーメッセージを表示する。
- 認証成功したら、顧客が以下の 3 つの操作が可能である。
新規予約作成、既存予約照会、既存予約解約
- 新規予約作成する際には、顧客は予約内容(開始日、終了日、室数)をシステムに入力する。
予約が可能なのは当日から 30 日後までとする。開始日と終了日がこの範囲になれば、日付不正エラーメッセージを表示し、再入力を促す。
- 指定された期間の全ての日において、指定された数以上の空室がある場合に予約記録を作り、予約 DB に登録し、予約 DB から帰ってきた予約 ID を画面に表示する。
- 期間中の 1 日でも空室が不足する場合は予約失敗とし、空室不足エラーメッセージを画面に表示する。

- 既存予約照会の際に、顧客は予約IDをシステムに入力する。入力された予約IDに対応する予約記録が予約DBに存在しない場合、ID不正エラーメッセージを表示する。
予約記録存在するが作成者のユーザーIDと利用者のユーザーID異なる場合は、権限なしエラーメッセージを表示する。
- 照会に成功した後で、顧客は表示された予約を解約することができる。システムが対応する予約記録を予約DBから削除し、成功メッセージを表示する。

要求分析の結果よりホテル予約管理システムが 5 つのユースケースにより実現されている。図 5.1 にホテル予約管理システムのユースケース図を示す。

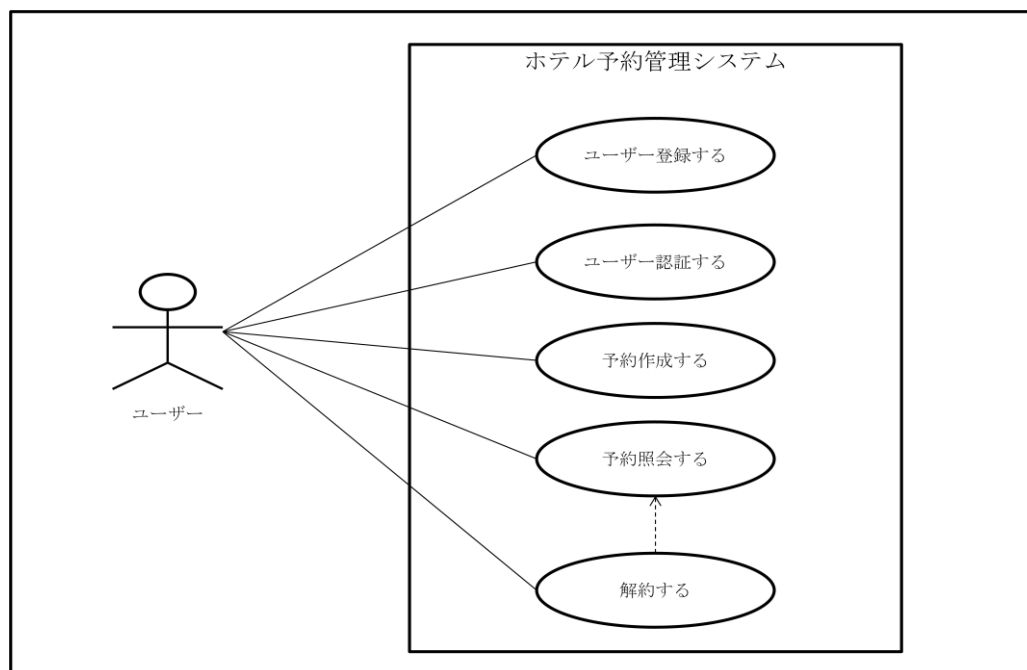


図 5.1: ホテル予約管理システムのユースケース図

5.1 要求構成グラフ

図 5.1 における各ユースケースの詳細記述から要求構成グラフを作成する。全体のユースケース記述は付録を参照。

- ユースケース名: ユーザー登録する

主シーケンス:

1. 利用者はユーザーIDとパスワードをシステムに入力する。
2. システムはユーザーIDを重複検査する
3. システムはユーザーIDとパスワードをユーザーDBに登録する。
4. システムは登録成功メッセージを利用者に表示する。

代替シーケンス:

2. でユーザーID が登録済みの場合

=> A1 システムは ID 重複エラーメッセージを利用者に表示する。

A2 システムは再入力を促すため入力画面を表示する。

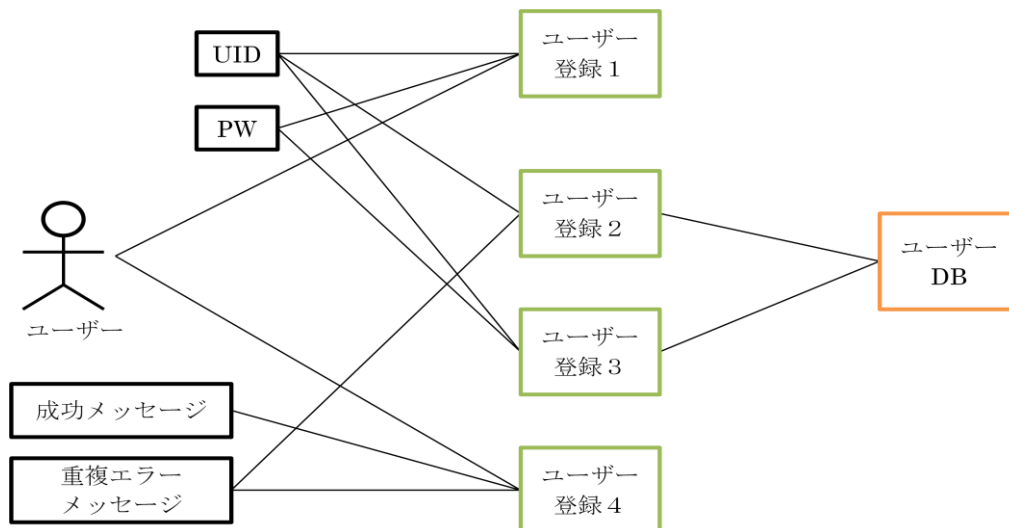


図 5.2: ホテル予約管理システム要求構成グラフ(ユーザー登録)

図 5.2 はユーザー登録の要求構成グラフを示す。以下は、各要素の説明することである。

- アクター要素: ユーザー
- データ要素: 入力データ UID(ユーザーID)と PW(パスワード)
出力データ成功メッセージと重複エラーメッセージ
- シーケンス要素: ユーザー登録1から4までのシーケンス(UID と PW 入力、UID の妥当性検査、UID と PW 登録、登録結果返す)
- データベース要素: ユーザーDB(ユーザーテーブル)

- ユースケース名: ユーザー認証する

主シーケンス:

1. 利用者はユーザーID とパスワードをシステムに入力する。
2. システムはユーザーID とパスワードをユーザーDB に送り、正当性を
検査する。
3. システムは認証成功メッセージを利用者に表示する。

代替シーケンス:

2. でユーザーID とパスワードが不当な場合、システムは認証失敗エラーメッセ
ージを利用者に表示する。

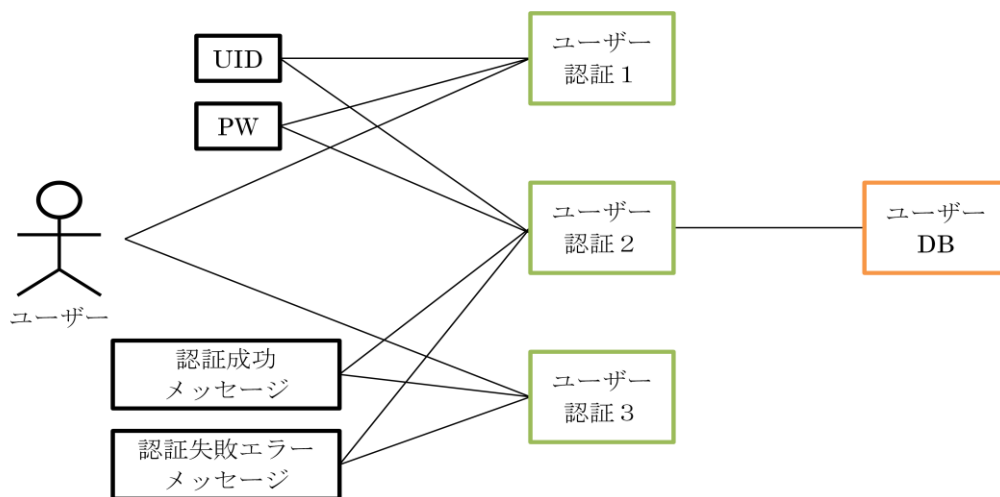


図 5.3: ホテル予約管理システム要求構成グラフ(ユーザー認証)

図 5.3 はユーザー認証の要求構成グラフを示す。以下は、新要素の説明することである。

- データ要素: 出力データ認証成功メッセージと認証失敗エラーメッセージ
- シーケンス要素: ユーザー認証1から4までのシーケンス(UID と PW 入力、UID と PW 正当性検査、認証結果返す)

- ユースケース名: 予約作成する

主シーケンス:

1. 利用者は予約内容(開始日、終了日、部屋数)をシステムに入力する。
2. システムは日付の正当性を検査する。
3. システムは日付の範囲を検査する。
4. システムは客室 DB に予約内容を送り、該当期間の空室数を得る。
5. システムは(作成者、作成日、予約内容)から予約記録を作成する。
6. システムは期間内空室リストに基づいて客室 DB を更新する。
7. システムは予約記録を予約記録 DB に登録し、予約記録 DB は予約 ID をシステムに返す。
8. システムは利用者に予約 ID を表示する。

代替シーケンス:

2. で日付が不当な場合(例: 9 月 31 日の場合)、システムは利用者に日付エラーメッセージを表示する。
3. で予約期間が不正な場合($0 \leq \text{OUT} - \text{IN} \leq 30$ の範囲以外)、システムは利用者に範囲エラーメッセージを表示する。
4. で空室数が不足な場合、システムは利用者に空室不足エラーメッセージを表示する。

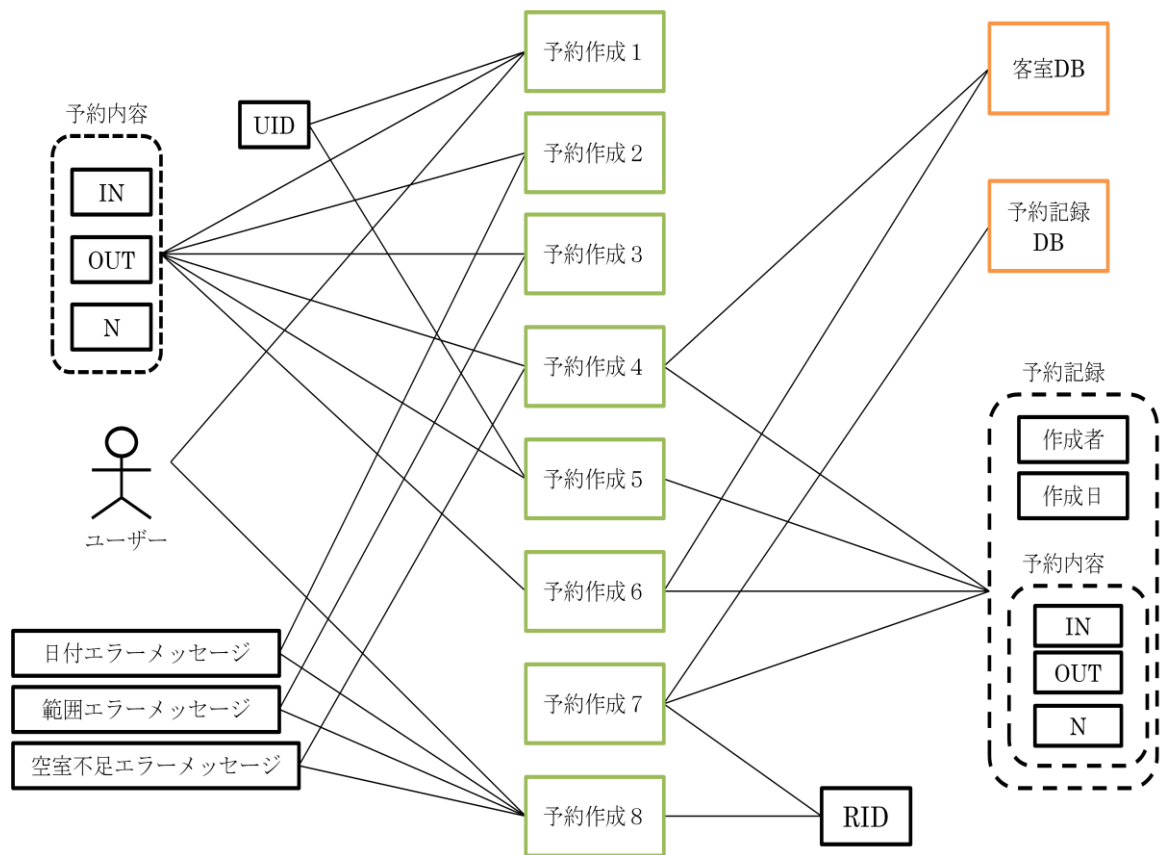


図 5.4: ホテル予約管理システム要求構成グラフ(予約作成)

図 5.4 は予約作成の要求構成グラフを示す。以下は、新要素の説明することである。

- データ要素： 入力データ予約内容 (IN(開始日)、OUT(終了日)、N(部屋数));出力データ RID(予約 ID)、日付エラーメッセージ、範囲エラーメッセージ、空室不足エラーメッセージ;予約記録(作成者、作成日、予約内容)
点線の部分是对应するデータをまとめることである。
- シーケンス要素： 予約作成 1 から 8 までのシーケンス(予約内容入力、日付正当性検査、日付の範囲検査、空室数得る、予約記録作成、客室 DB 更新、予約記録登録、作成結果返す)
- データベース要素： 客室 DB(客室情報テーブル)、予約記録 DB(予約記録テーブル)

- ユースケース名: 予約照会する

主シーケンス:

1. 利用者は予約 ID をシステムに入力する。
2. システムは予約 ID の正当性を検査する。
3. システムは予約記録 DB から予約記録を取得する。
4. システムは権限があるかを検査する。
5. システムは利用者に予約内容を表示する。

代替シーケンス:

2. で予約 ID が不正な場合、システムは利用者に ID 不正エラーメッセージを表示する。
4. で利用者に予約記録を参照する権限がない場合、システムは利用者に権限なしエラーメッセージを表示する。

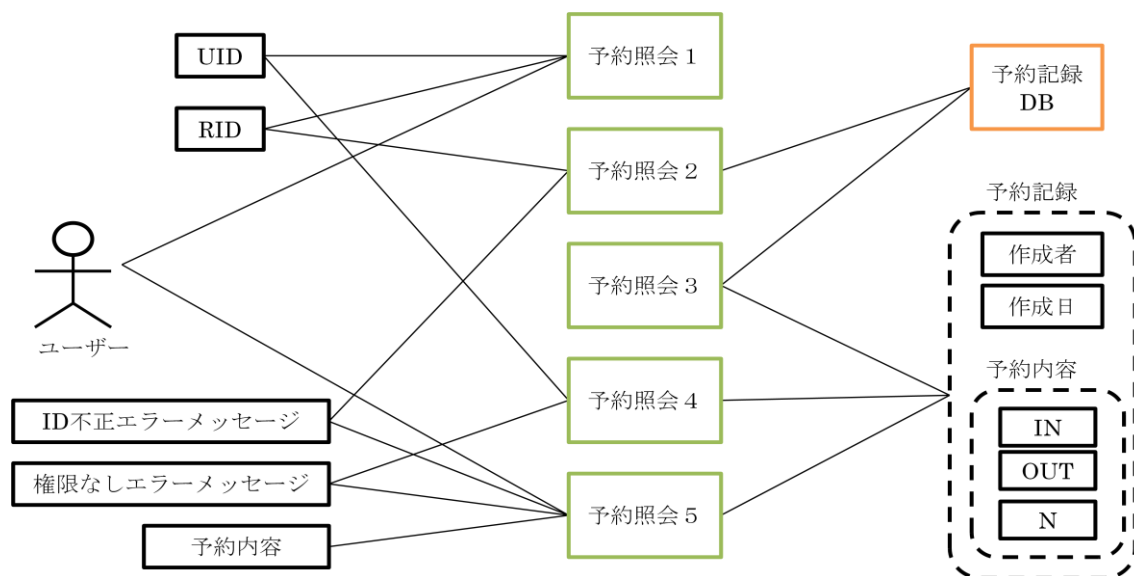


図 5.5: ホテル予約管理システム要求構成グラフ(予約照会)

図 5.5 は予約照会の要求構成グラフを示す。以下は、新要素の説明することである。

- データ要素: 出力データ ID 不正エラーメッセージと権限なしエラーメッセージ
- シーケンス要素: 予約照会 1 から 5 までのシーケンス(RID 入力、RID 正当性検査、予約記録取得、権限検査、照会結果返す)

- ユースケース名：解約する
前提条件：利用者は予約照会に成功している
主シーケンス：

1. システムは部屋割り当てに基づいて客室 DBを更新する。
2. システムは予約記録 DB から予約記録を削除する。
3. システムは成功メッセージを利用者に表示する。

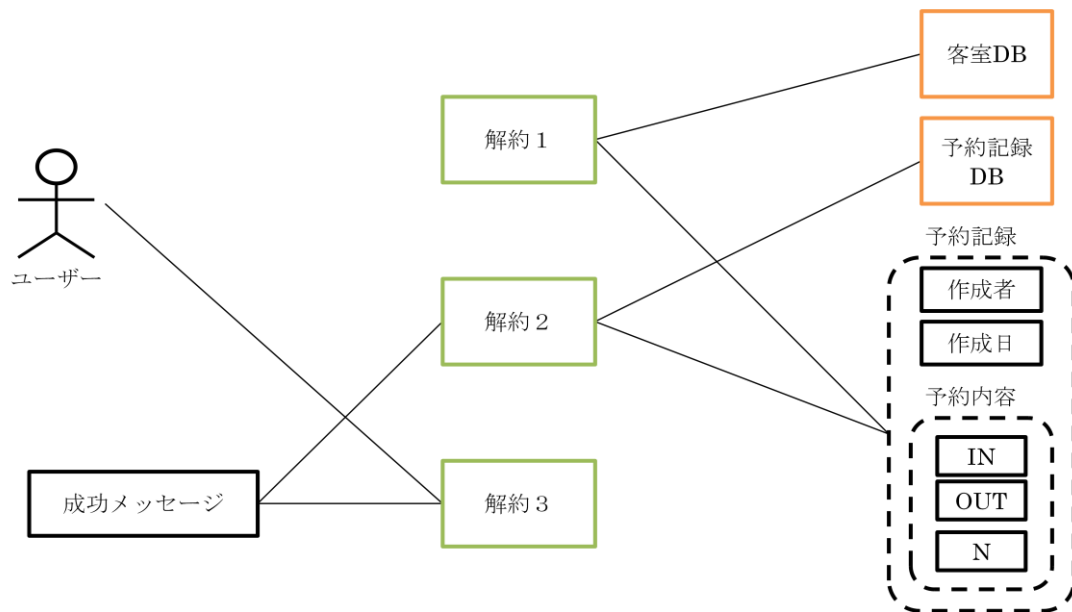


図 5.6: ホテル予約管理システム要求構成要素グラフ(解約)

図 5.6 は解約の要求構成グラフを示す。以下は、新要素の説明することである。

- データ要素：出力データ成功メッセージ
- シーケンス要素：解約 1 から 3 までのシーケンス(客室 DB 更新、予約記録削除、解約結果返す)

5.2 Struts1 を使用する場合の実現構成グラフ

本節では、Struts1[2]を使用する場合の実現構成グラフを説明する。まず、Struts1 における MVC アーキテクチャの仕組みについて説明する。

Struts1 におけるコントローラ (Controller) 部分は、ActionServlet クラス、RequestProcessor クラス、ActionForm クラスと Action クラス 4 つのクラスで構成されている。

1. ActionServlet クラスは、クライアントからの全てのリクエストを受け取るクラスである。
2. RequestProcessor クラスは、ActionServlet クラスから受け取ったリクエストに応じて処理を実行するクラスである。
3. ActionForm クラスは、ユーザーとシステムとの情報の受け渡しを行うクラスである。ユーザーから入力された情報は ActionForm クラスに格納される。それは、格納された情報を Action クラスに渡す。
4. Action クラスは、ロジック部分の処理を呼び出すクラスである。Controller 部分から Model 部分の処理を呼び出す。

モデル (Model) 部分の開発には、JavaBeans が用いられるのが一般的である。開発者が作成する部分である。

ビュー (View) 部分の開発には、JSP を用いる。Struts1 は、JSP において利用できる豊富なタグ・ライブラリを提供する。

以上の記述することより、対象とするホテル予約管理システムが Struts1 を使用する場合の実現構成グラフを作成する。各ユースケースに対して、5 つのグラフを作成した。

以下の図 5.7 から図 5.11 までは、全ての実現構成要素を揃っており、グラフで示した。ビュー層要素を黒い四角形で表現する。コントローラ層要素を緑色四角形で表現する。モデル層要素をオレンジ四角形で表現する。点線四角形の部分は、それを接続している要素内に保持する情報である。点線楕円の部分は、それを接続している Action 要素の担当するロジック処理である。

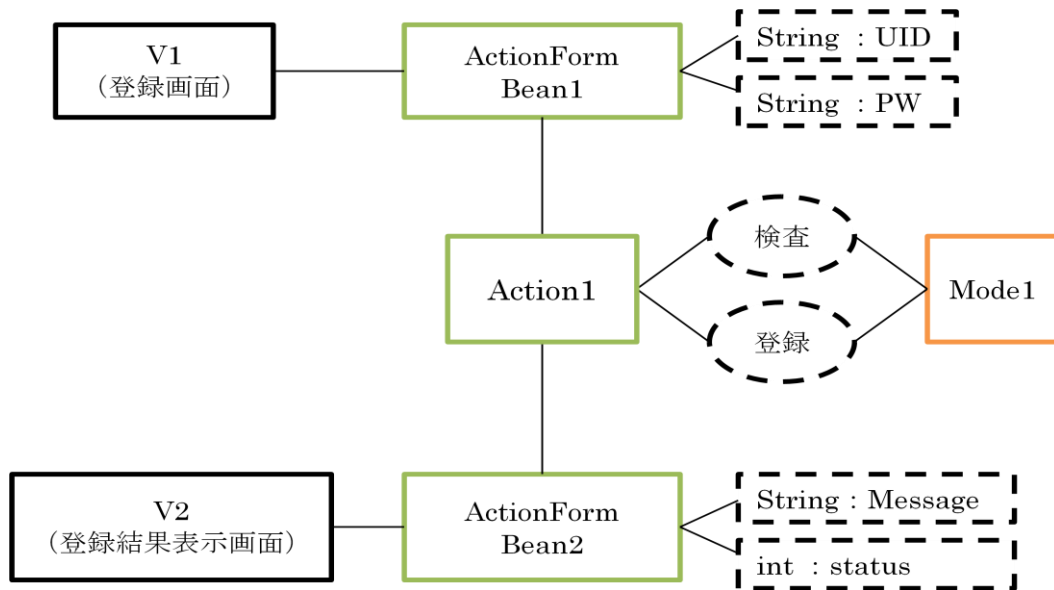


図 5.7: Struts1 を使用する場合の実現構成グラフ (ユーザー登録)

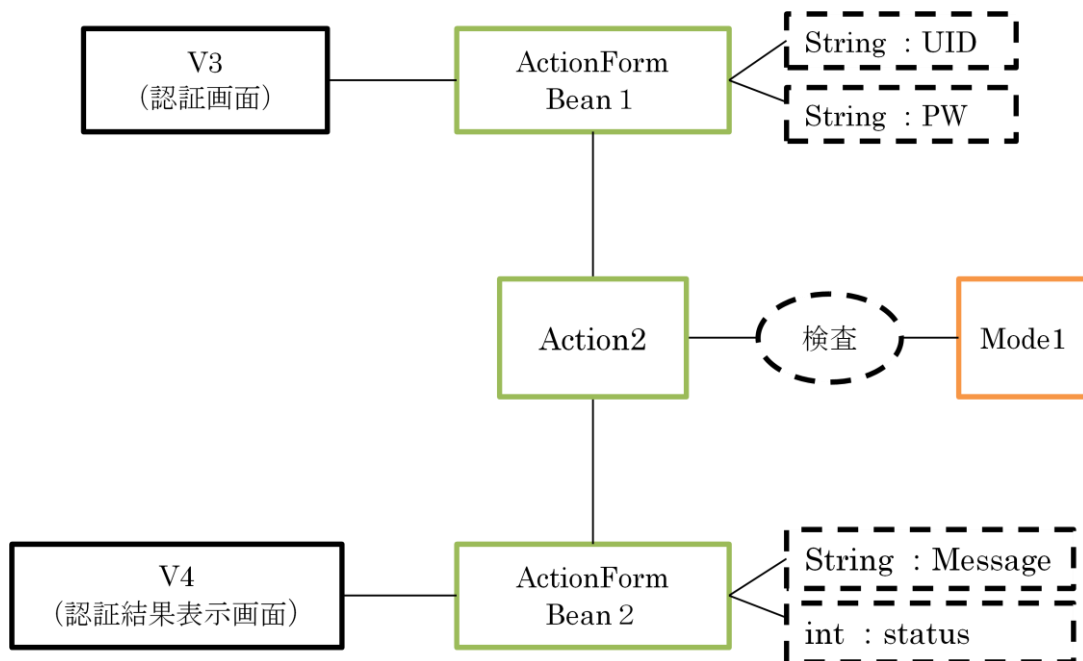


図 5.8: Struts1 を使用する場合の実現構成グラフ (ユーザー認証)

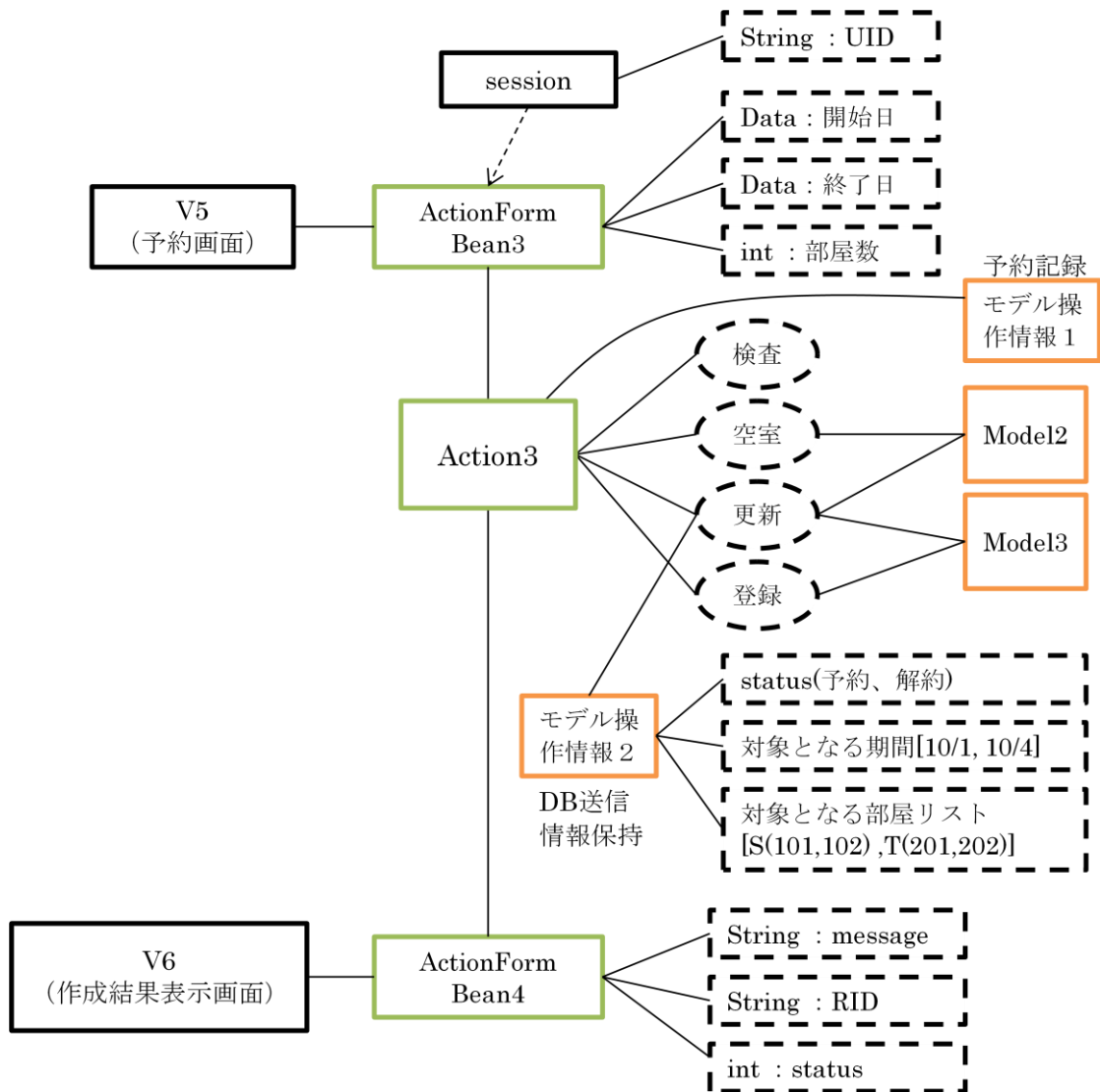


図 5.9: Struts1 を使用する場合の実現構成グラフ(予約作成)

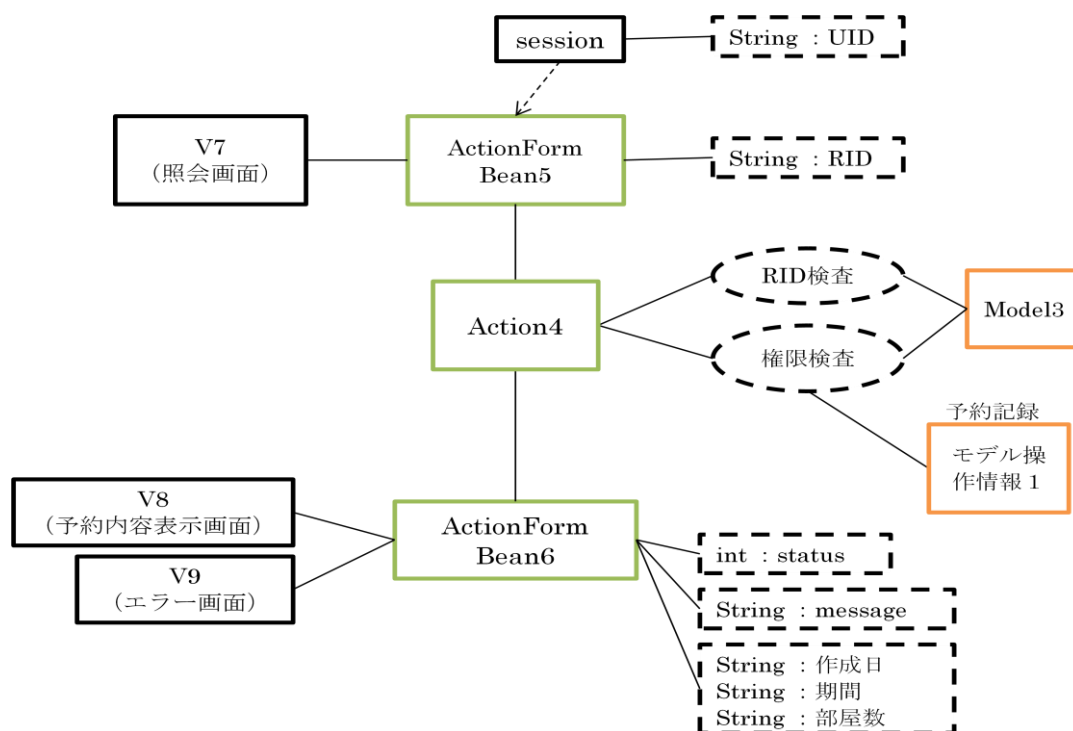


図 5.10: Struts1 を使用する場合の実現構成グラフ(予約照会)

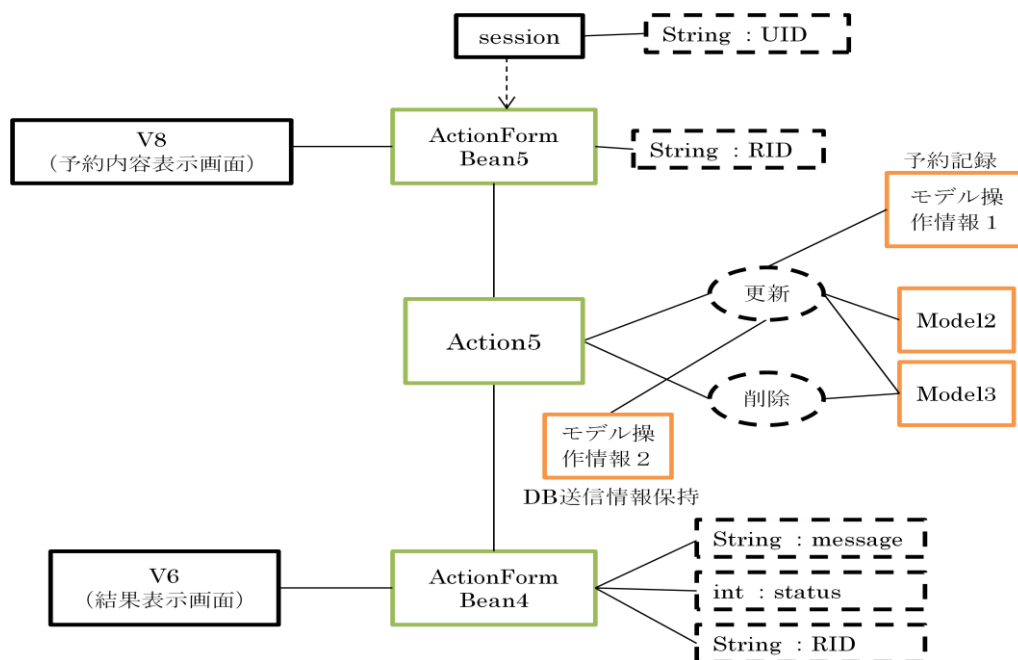


図 5.11: Struts1 を使用する場合の実現構成グラフ(解約)

表 5.1 は実現構成要素の一覧を示す。ここで全ての ActionForm Bean 要素を AFB 略語で示す。

表 5.1: 実現構成要素の一覧(Struts1)

要素種類	要素名前	内容
ビュー層要素	登録画面 (V1)	
	登録結果表示画面 (V2)	
	認証画面 (V3)	
	認証結果表示画面 (V4)	
	予約画面 (V5)	
	作成、解約結果表示画面 (V6)	
	照会画面 (V7)	
	予約内容表示画面 (V8)	
	エラー画面 (V9)	
コントローラ層要素	登録情報保持 AFB (AFB1)	(String uid, String pw)
	登録結果情報保持 AFB (AFB2)	(String mes, Enum res)
	予約情報保持 AFB (AFB3)	(Date std, Date ed, int n)
	作成、解約結果情報保持 AFB (AFB4)	(String rid, String mes, Enum res)
	照会情報保持 AFB (AFB5)	(String rid)
	照会結果情報保持 AFB (AFB6)	(String mes, Rinfo r, Enum res) Rinfo=(Date std, Date ed, int n)
	ユーザー登録 Action (A1)	UID の妥当性検査、新しい利用者の UID と PW をユーザーDB に登録する
	ユーザー認証 Action (A2)	UID と PW の正当性検査
	予約作成 Action (A3)	日付の正当性と範囲検査、空室数を得る、客室 DB の更新、予約記録を予約 DB に登録する
	予約照会 Action (A4)	RID の正当性検査、権限有無の検査
	解約 Action (A5)	客室 DB の更新、予約記録 DB から予約記録を削除する

モデル層要素	ユーザー情報モデル (M1)	(String uid, String pw)
	客室情報モデル (M2)	(Date d, Room r, Enum st) Enum=(空室、予約済み)
	予約記録情報モデル (M3)	(String rid, Date d, Person p, Rinfo c) Rinfo=(Date std, Date ed, int n)
	モデル操作情報 M3 の一部のコピー (MA1)	(String rid, Date d, Person p, Rinfo c) Rinfo=(Date std, Date ed, int n)
	モデル操作情報 M2 の更新用 (MA2)	(Date std, Date ed, List<Room> l, Enum op) Enum op=[予約、解約]

5.3 Struts2 を使用する場合の実現構成グラフ

本節では、Struts2[6]を使用する場合の実現構成グラフを説明する。まず、Struts2 における MVC アーキテクチャの仕組みについて説明する。

コントローラ部分は Struts2 のコア部分であり、FilterDispatch が全ての制御を担っている。

モデル部分は、Action クラスとなっていて、アプリケーションで用意するクラスである。Action クラスは、リクエストパラメータの保存と業務ロジックを担当する。Action を実装するクラスは、特別なクラスの継承やインタフェースの実装を必要としない。すなわち、POJO (Plain Old Java Objects) として作成することができる。

ビュー部分は、URI リクエストの結果ページを表示する。JSP の作成を助けて、様々なカスタムタグを提供している。

以上の記述することより、対象とするホテル予約管理システムが Struts2 を使用する場合の実現構成グラフを作成する。ビュー層要素を黒い四角形で表現する。モデル層要素をオレンジ四角形で表現する。

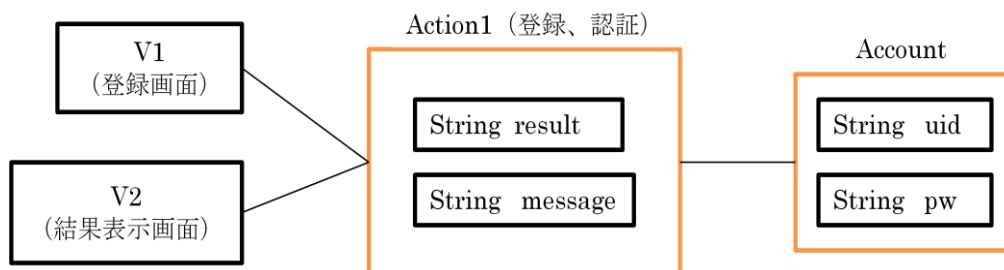


図 5.12: Struts2 を使用する場合の実現構成グラフ(ユーザー登録、認証)

図 5.12 では、ユーザー登録と認証の実現構成グラフを示す。Action1 の result 値は”success”と”input”とする。success なら、結果表示画面に行く。input なら、登録画面に戻る。message 値は空でなければ、表示する。Account はユーザーの情報(ユーザーID とパスワード)を保持するオブジェクトである。

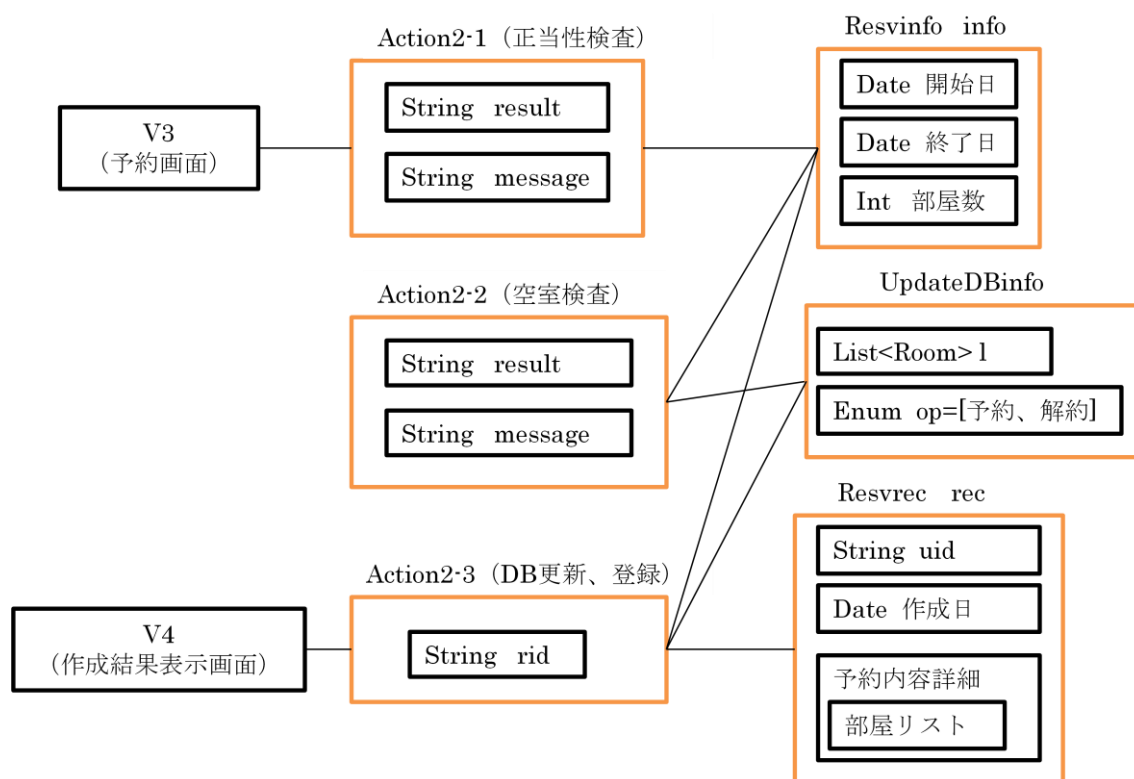


図 5.13: Struts2 を使用する場合の実現構成グラフ(予約作成)

図 5.13 では、予約作成の実現構成グラフを示す。Action2-1 の result 値は”success”と”input”とする。success なら、Action2-2 に行く。input なら、予約画面に戻る。message 値は空でなければ、表示する。Action2-2 の result 値は”success”とする。常に、Action2-3 に行く。message 値は空でなければ、表示する。Action2-3 の RID 値は空でなければ、表示する。Resvinfo は予約内容の情報(開始日、終了日、部屋数)を保持するオブジェクトである。UpdateDBinfo は客室 DBを更新するための情報を保持するオブジェクトである。Resvrec は予約記録の情報を保持するオブジェクトである。

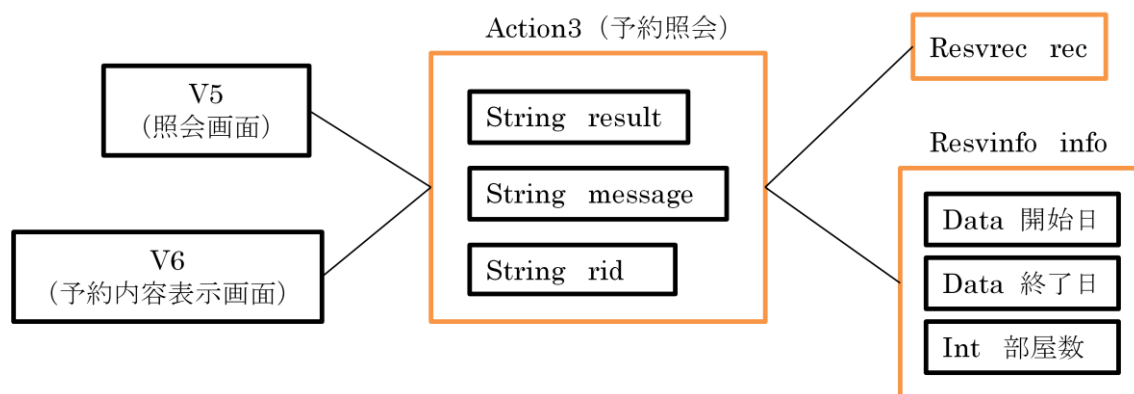


図 5.14: Struts2 を使用する場合の実現構成グラフ(予約照会)

図 5.14 では、予約照会の実現構成グラフを示す。Action3 の result 値は”success”とする。常に、予約内容表示画面に行く。message 値は空でなければ、表示する。Resvrec の値は空でなければ、表示する。

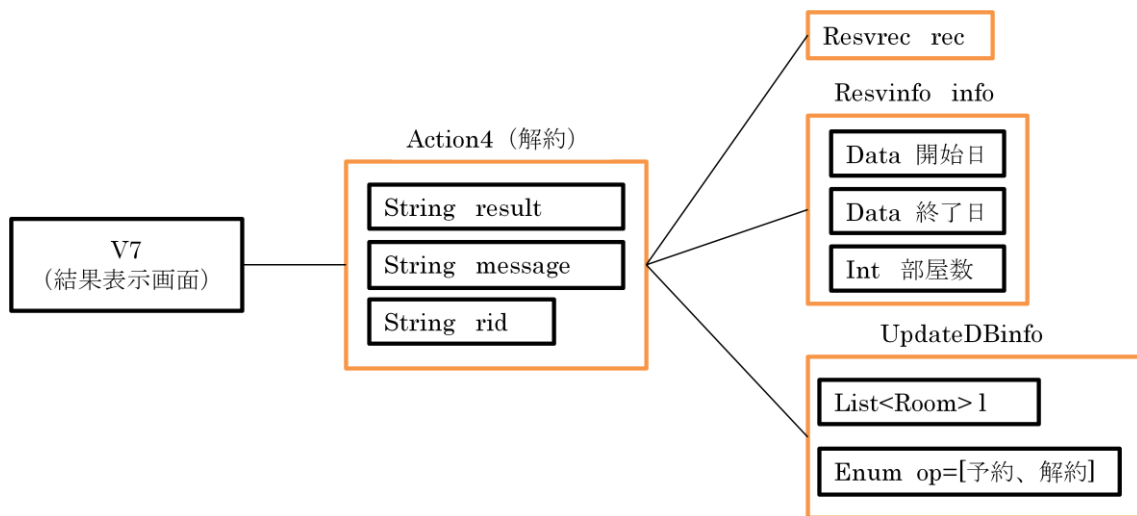


図 5.15: Struts2 を使用する場合の実現構成グラフ(解約)

図 5.15 では、解約の実現構成グラフを示す。Action4 の result 値は”success”とする。常に、結果表示画面に行く。message 値を定数で表示する。

5.4 相違分析と変更規則

本節では、5.2 節と 5.3 節で記述した Struts1 と Struts2 におけるそれぞれの実装構成グラフの相違を分析し、変更規則を記述する。まず、Struts1 と Struts2 におけるそれぞれの全体実現構成グラフを作成する。そして、2 つのグラフを比較し、Struts1 から Struts2 へのグラフの変更を必要となるパターンを抽出し、変更規則を定義する。以下の図 5.16 と図 5.17 では、Struts1 と Struts2 におけるそれぞれの実現構成グラフを示す。

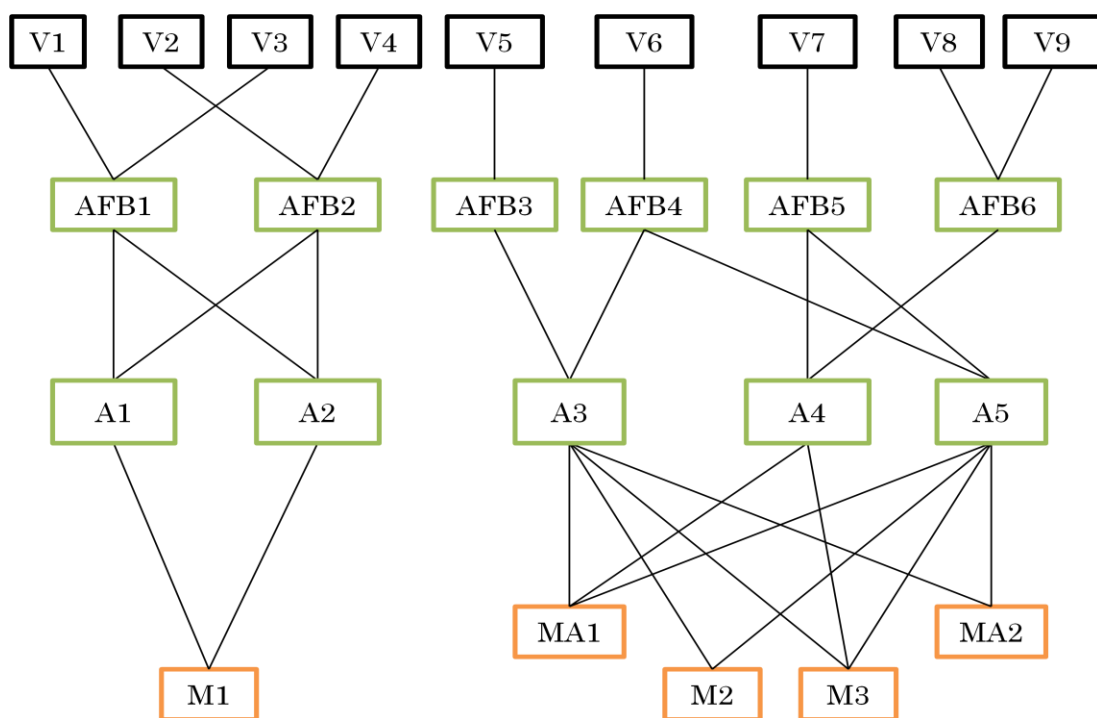


図 5.16: 実現構成グラフ全体 (Struts1)

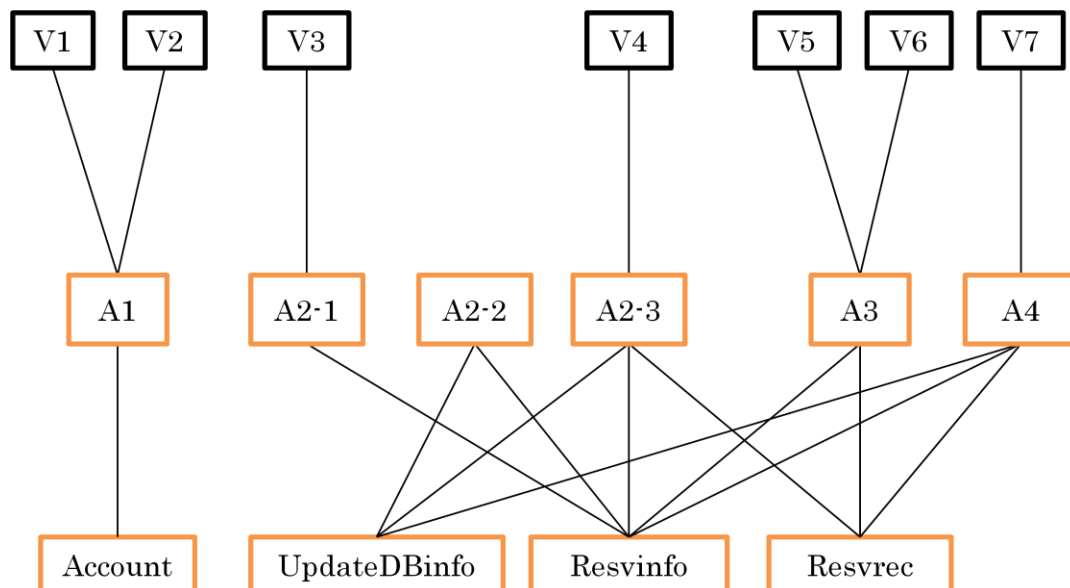


図 5.17: 実現構成グラフ全体 (Struts2)

2つのグラフを比較すると、Struts1からStruts2へ対象とするアプリケーションを適合させる際に必要となる変更パターンを2つ抽出する。図5.18と図5.19では、抽出する変更パターンを示す。

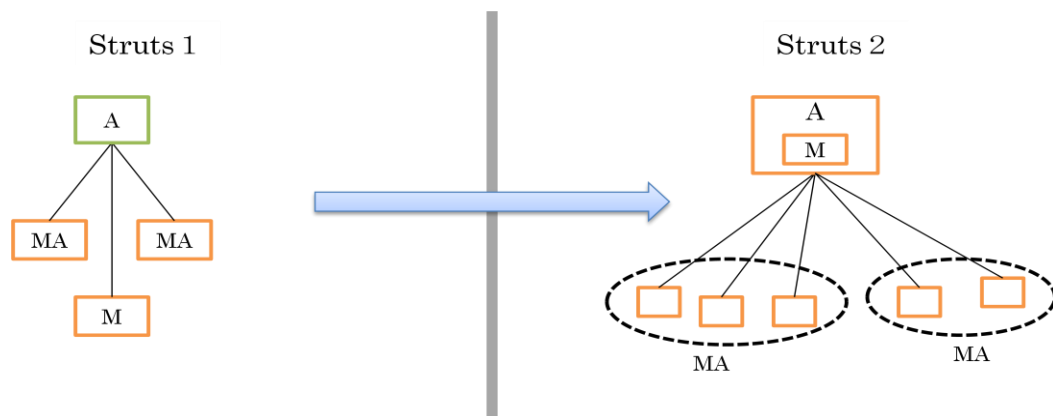


図 5.18：実現構成グラフの変更パターン(その 1)

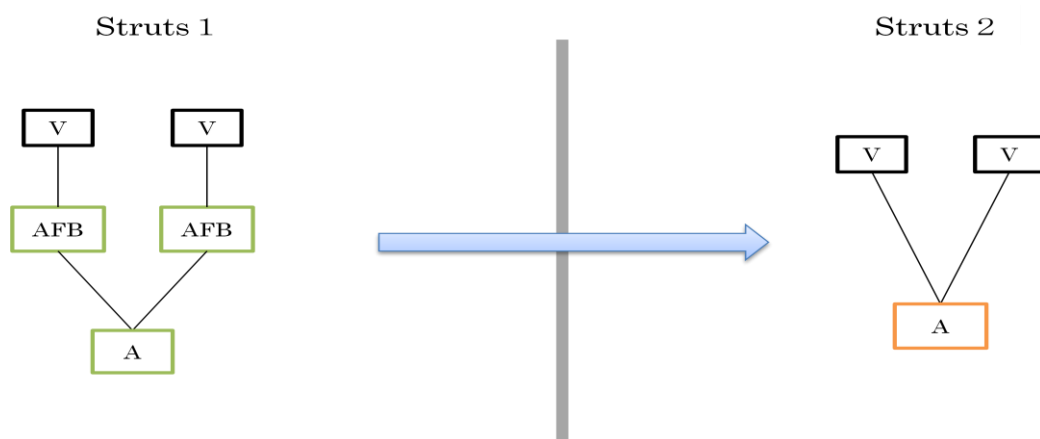


図 5.19：実現構成グラフの変更パターン(その 2)

以上の抽出する変更パターンより、2 つの変更規則を定義する。

1. モデルのアクション内部への移動

フレームワーク側でアクションの呼び出し方法が変わった。**Struts2** ではロジックとデータが **Action** クラスにまとまり、すっきりしたオブジェクト指向的な設計になっている。**Action** クラスは、**POJO** として作成することが可能である。

2. AFB のアクションへの統合

アクションからの情報抽出および表示の方法が変更された。**AFB** の情報をアクションに保持する。**ActionForm Bean** がなくなり、保守性の向上と開発容易になる。

5.5 変更規則の評価

本節では、対象とするホテル予約管理システムの機能を追加することを利用して、変更規則を評価する。

対象とするホテル予約管理システム機能追加の仕様：

- 照会で予約 ID ではなく、利用者が期間をシステムに入力する。システムはその期間を含む予約を全て予約 DB から取り出す。システムは予約 ID のリストを画面に表示する。入力期間内の予約がない場合、システムは予約なしメッセージを利用者に表示する。
- 利用者は予約 ID と期間をシステムに両方とも入力すれば、重複エラーメッセージを表示する。入力する期間が不正の場合、期間不正エラーメッセージを表示する。
- 予約 ID、期間ともに空の場合は、そのユーザーが作成した全ての記録の予約 ID リストを表示する。
- 上のリストの中から、1つを選択し、照会の対象とする。(ボタンを作成する)
- 照会結果出力画面に解約ボタンを用意し、対象の解約できるようにする。

5.5.1 機能追加要求構成グラフ

対象とするホテル予約管理システム機能追加の仕様から、要求分析した結果より、要求構成グラフを作成する。

- ユースケース名: 予約照会 (機能追加)

主シーケンス:

1. 利用者は期間をシステムに入力する。
2. システムは入力期間を検査する。
3. システムは期間内の予約があるかを検査する。
4. システムは予約 DB から予約 ID を取得する。
5. システムは利用者に予約 ID リストを結果表示画面に表示する。
6. 利用者は 1 つの RID を選択する。
7. システムは予約 DB から予約記録を取得する。
8. システムは利用者に予約内容を表示する。

代替シーケンス:

予約 ID と期間の両方が入力された場合、システムは重複エラーメッセージを表示する。

予約 ID、期間ともに空の場合、システムは利用者が作成した全ての記録の予約 ID リストを表示する。

2. で入力期間が不正の場合、システムは利用者に期間不正メッセージを表示する。

3. で期間内の予約がない場合、システムは利用者に予約なしメッセージを表示する。

図 5.20 では予約照会機能追加の要求構成グラフを示す。全ての赤い部分は追加する要求構成要素である。それは 5.1 節で予約照会の要求構成グラフに基づいて、拡張したグラフである。以下は、追加要求構成要素を説明する。

- データ要素: 入力データ予約期間(開始日:Date、終了日:Date);出力データ重複エラーメッセージと期間不正メッセージおよび予約なしメッセージ;予約 ID リスト(List<RID> rl)
- シーケンス要素: 予約照会 1-1 から予約照会 1-2 までのシーケンス(予約抽出、予約抽出期間指定、RID 選択)

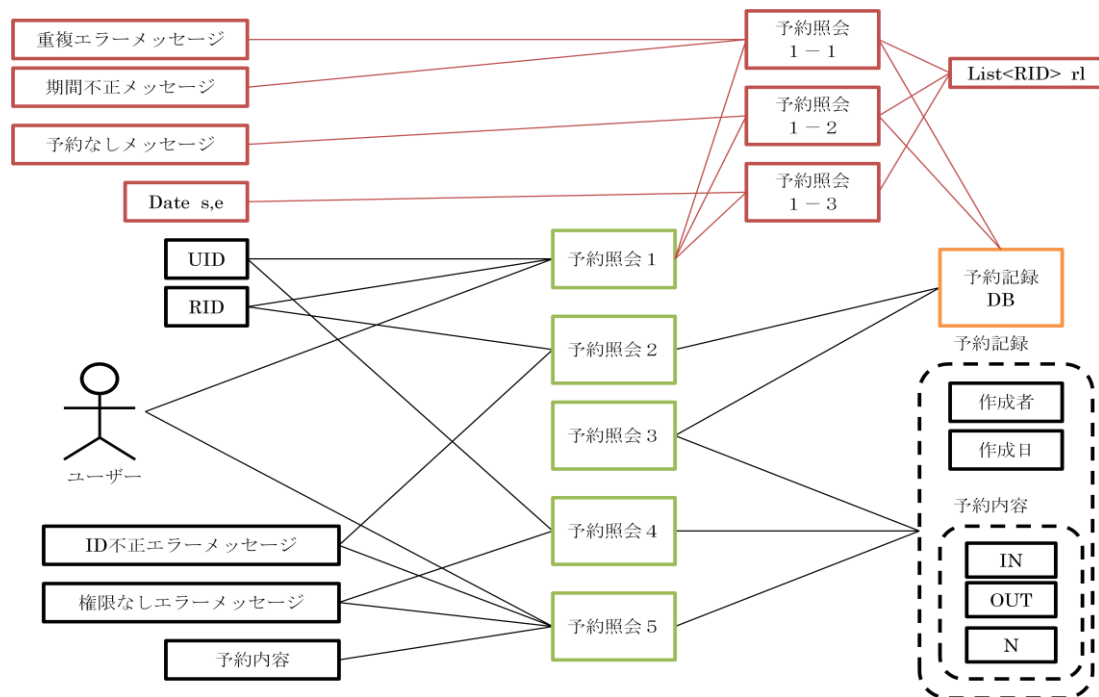


図 5.20：予約照会機能追加の要求構成グラフ

5.5.2 機能追加実現構成グラフ

対象とするホテル予約管理システムの機能を追加する時に Struts1 を使用する場合の実現構成グラフを作成する。赤色四角形で表示する要素は追加する実現構成要素である。青い四角形で表示する要素は既存の要素を変更した要素である。図 5.21 と図 5.22 では、機能追加の実現構成グラフを示す。

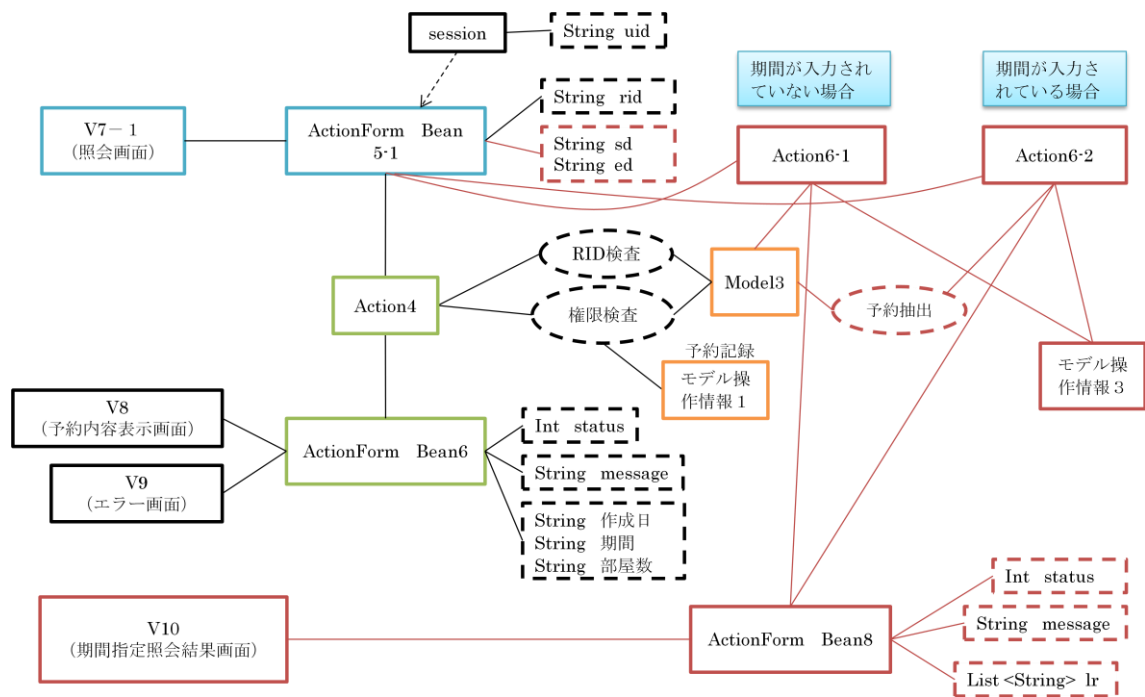


図 5.21 : Struts1 を使用する場合の機能追加実現構成グラフ (1)

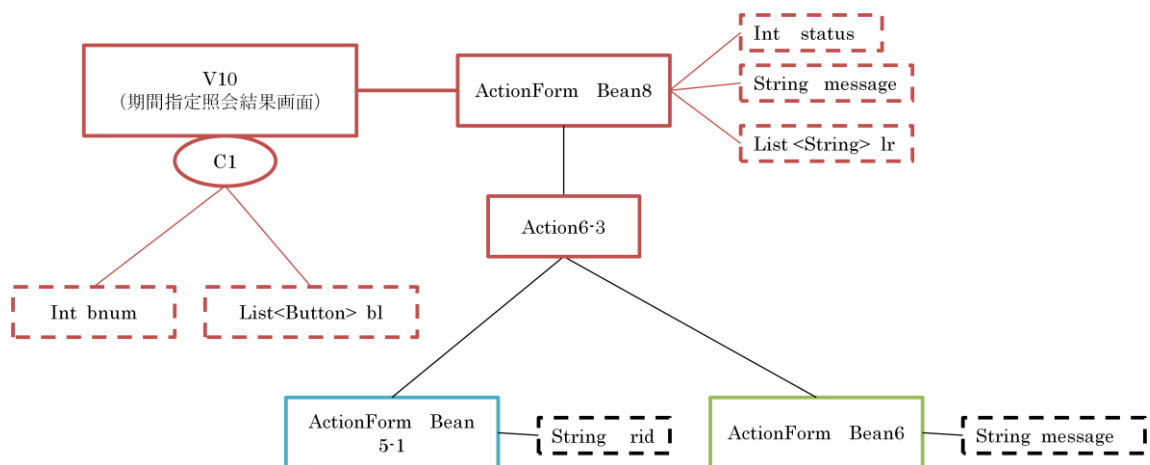


図 5.22 : Struts1 を使用する場合の機能追加実現構成グラフ (2)

表 5.2 では、機能追加に対応する追加と変更の実現構成要素の一覧である。ここで全ての ActionForm Bean 要素を AFB 略語で示す。

表 5.2: 追加と変更の実現構成要素の一覧(Struts1)

	要素名前	内容
追加要素	期間指定照会結果画面 (V10)	
	ボタンのコントロール (C1)	(int bnum, List<Button> bl)
	期間指定照会結果情報保持 AFB (AFB8)	(Enum res, String mes, List<String> rl)
	予約抽出 Action (A6-1)	全ての RID 抽出し、MA3 に書く
	予約抽出期間指定 Action (A6-2)	期間内の RID 抽出し、MA3 に書く
	予約 ID 選択 Action (A6-3)	AFB8 のリスト中から選択された RID を AFB5-1 に書く
	M3 の期間指定及び抽出結果情報 (MA3)	(Date s, Date e, List<String> rl)
変更要素	照会画面変更 (V7-1)	
	照会情報保持変更 AFB (AFB5-1)	(String rid, String sd, String ed)

以下では、対象とするホテル予約管理システムの機能を追加する時に Struts2 を使用する場合の実現構成グラフを作成する。赤色の部分は追加する要素である。青い部分は既存の要素を変更した要素である。図 5.23 では、Struts2 を使用する場合の機能追加実現構成グラフを示す。

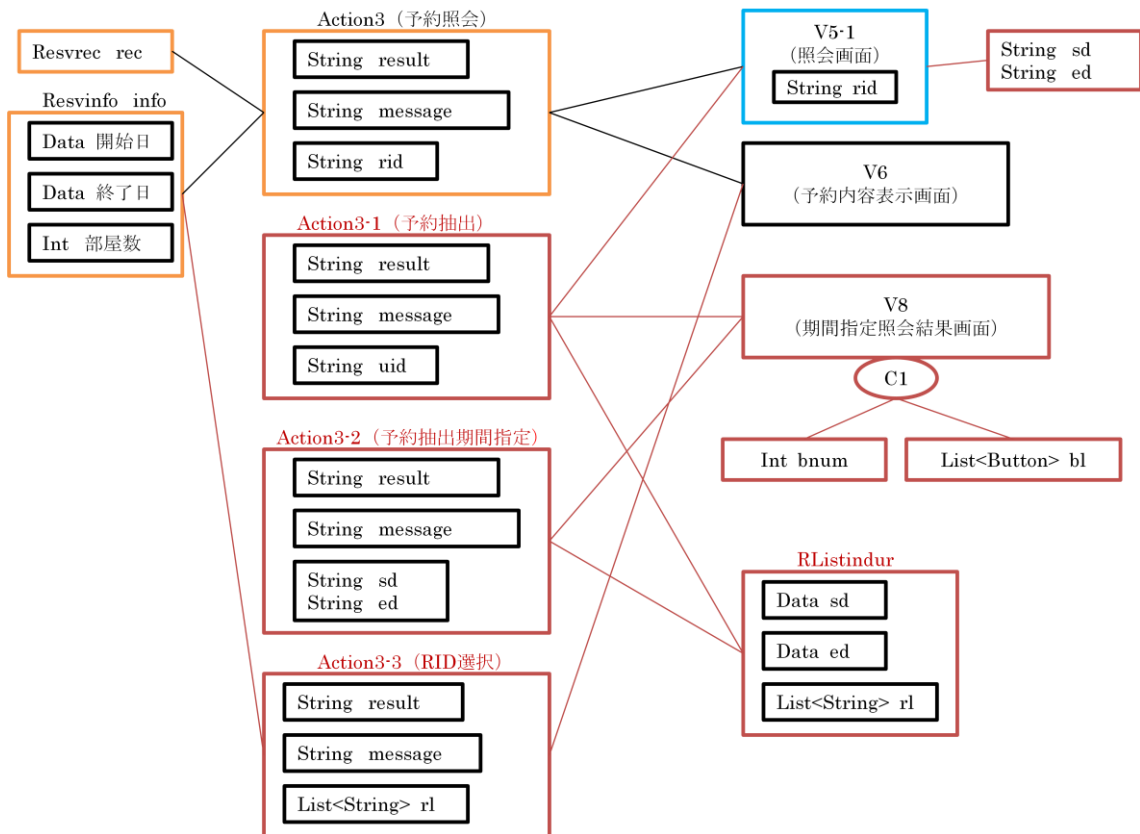


図 5.23: Struts2 を使用する場合の機能追加実現構成グラフ

図 5.23 では、Struts2 を使用する場合の実現構成要素を示す。以下は、追加する各要素を説明する。

- Action3-1(予約抽出)の起動条件は `rid==空文字列かつ(sd==空文字列 or ed==空文字列)` である。動作は予約記録 DB から作成者が現在の uid と一致する全ての予約記録の rid をリストで返し、RListindur 内の rl に格納することである。result の値は常に”success”で、message の値は常に空文字列である。
- Action3-2(予約抽出期間指定)の起動条件は `rid==空文字列かつ(sd!=空文字列 and ed!=空文字列)` である。動作は予約記録DBから作成者が現在の uid と一致し、予約記録中の期間が指定された期間と重なる全ての予約記録の rid をリストで返し RListindur 内の rl に格納することである。result の値は常に”success”である。message の値は期間が正当ならば、空文字列である。そうでなければ、対応するエラーメッセージである。

- Action3-3(RID 選択)の起動条件は Rlistindur の rl に値があることである。動作は rid の値を rl[bnum]とし、rid を予約記録 DB に送り、予約記録を取得し、予約記録中の予約内容を info に格納することである。result の値は常に”success”で、message の値は常に空文字列である。
- Rlistindur は照会期間と予約 ID リストの情報を保持するオブジェクトである。

5.5.3 評価

5.5.2 小節で作成したグラフにより、追加必要な実現構成要素を以下にまとめる。

- Struts1 で追加必要な要素：
 - 追加機能に対応するアクション
 - 入力を保持する ActionForm Bean と対応するビュー
 - モデルを更新するためのモデル操作情報(MA)
- Struts2 で追加必要な要素：
 - 追加機能に対応するビュー
 - 追加機能に対応するアクション
 - 追加機能の実現に必要なデータを保持するオブジェクト(POJO)

以上の追加必要な要素を通じて、5.4 節での変更規則に従って、Struts1 で追加必要な実現構成要素を変更したら、Struts2 で追加必要な実現構成要素になれる。すなわち、Struts1 を使用する場合の機能追加実現構成グラフから Struts2 を使用する場合の機能追加実現構成グラフに変更ができる。それによって、対象とするアプリケーションの変更ができるようになった。しかし、システムの規模により、変更規則が不足かもしれないので、また改善する必要があると考える。

第 6 章 評価

本章では、本研究の提案手法は第 5 章の適用事例に基づく評価と考察を述べる。

6.1 評価

第 5 章では、提案手法を用いて適用事例に行った。提案手法を用いて、対象とするホテル予約システムの要求構成グラフと Struts1 と Struts2 におけるそれぞれの実装構成グラフを作成することができた。そして、異なる実装構成グラフを比較して、グラフの変更箇所が得られやすかった。抽出した変更パターンによって、変更規則を定義しやすかった。その結果、Struts1 における要求－実現－構成グラフを Struts2 における要求－実現－構成グラフに変更ができる。すなわち、Struts1 を使用して開発するアプリケーションを Struts2 を使用して開発するアプリケーションに変更ができる。

対象とするシステムは、グラフを利用しない場合に、フレームワークが進化した時に、アプリケーションの変更箇所が分散しており、変更作業が困難であり、変更コストが大きい。それに対し、提案手法のグラフを利用して、グラフの変更が発見しやすくて、体系的な変更規則を定義することができる。変更作業をしやすくなり、変更コストの低減が期待できる。

6.2 考察

本研究の提案手法では、アプリケーションの要求構成要素とフレームワークを利用した開発する実現構成要素に対して、各要素とその関連を構成するグラフの変更することを行った。グラフの変更をアプリケーションの変更に正しく反映させることである。それにより、本手法では、1 つのフレームワークの進化に対応することだけではなく、一般的に任意的な Web アプリケーションフレームワークの進化に対応可能である。また、フレームワークの多種類により、実現構成要素の確定することを正確に行う必要があると考える。

第 7 章 おわりに

7.1 まとめ

本論文は、フレームワークの進化に対応可能な Web アプリケーション作成手法について、グラフを利用した作成及び変更手法の研究した結果をまとめたものである。本研究では、アプリケーションの要求を構成する要素と実装を構成する要素及び要素の対応関係の 3 種類の構造に注目し、要求－実現－構成グラフを提案した。次に、要求－実現－構成グラフに基づくアプリケーションを実現する方法を説明した。最後に、提案手法を用いて、具体的にホテル予約管理システムを対象とする例として実施した。適用事例を実施した結果によって、提案手法は有用になった。

7.2 今後の課題

今後の課題は以下の 2 つについて検討してもよい。

1. ルール発見抽出に関して
ルールを改善するために、**Struts** の他の機能(セキュリティなど)を使用するアプリケーションに対する事例を収集する。また、他のフレームワーク(**Spring** など)に対する事例の収集及び変更ルールの発見と抽出することを行う。
2. ルールのアプリケーションへの反映に関して
複数の機能を使用するアプリケーションに対して、複数の変更ルールの適用に関する一貫性を保持すると考える。

謝辞

本研究を行うにあたり、終始熱心なご指導して頂きました北陸科学技術大学院大学 情報科学研究科 鈴木正人准教授に心から深く感謝申し上げます。

本研究の審査員として、大変有益なご意見とご助言を頂きました同大学院 青木利晃准教授と緒方和博准教授に深く感謝申し上げます。

また、本研究を進めるにあたり、色々な助言を頂きました鈴木研究室の皆様に、深く感謝申し上げます。

最後に、いつも暖かく応援していただきました家族と友達に、心から感謝申し上げます。

参考文献

- [1] Struts 本家のホームページ, <http://struts.apache.org/>
- [2] 中村健二編 Struts を活用した Web アプリケーション開発 工学社 , 2005.10
- [3] 高安厚思, 西川麗著 Struts による Web アプリケーションスーパーサンプル ソフトバンククリエイティブ , 2007.4
- [4] 石井真、阿島哲夫著 Jakarta プロジェクトカンタン Struts 秀和システム , 2003.6
- [5] Practical Apache Struts2 Web 2.0 projects / Ian Roughley Berkeley, Calif. : Apress New York : Distributed to the book trade worldwide by Springer-Verlag New York , c2007
- [6] Struts2 design and programming Second Edition/ Budi Kurniawan [Vancouver] : BrainySoftware , 2008

付録 A ホテル予約管理システムの 全体のユースケース記述

この付録では、対象とするホテル予約管理システムの全体のユースケース記述を述べる。

表 A.1: ユースケース記述(ユーザー登録する)

ユースケース名	ユーザー登録する
アクター	利用者
前提条件	特になし
主シーケンス	1. 利用者はユーザーID とパスワードをシステムに入力する。 2. システムはユーザーID を重複検査する 3. システムはユーザーID とパスワードをユーザーDB に登録する。 4. システムは登録成功メッセージを利用者に表示する。
代替シーケンス	2.でユーザーID が登録済みの場合 => A1 システムは ID 重複エラーメッセージを利用者に表示する。 A2 システムは再入力を促すため入力画面を表示する。
事後条件	ユーザー情報が登録されている。

表 A.2: ユースケース記述(ユーザー認証する)

ユースケース名	ユーザー認証する
アクター	利用者
前提条件	特になし
主シーケンス	1. 利用者はユーザーID とパスワードをシステムに入力する。 2. システムはユーザーID とパスワードをユーザーDB に送り、正当性を検査する。 3. システムは認証成功メッセージを利用者に表示する。
代替シーケンス	2.でユーザーID とパスワードが不当な場合、システムは認証失敗エラーメッセージを利用者に表示する。
事後条件	認証されている。

表 A.3: ユースケース記述(予約作成する)

ユースケース名	予約作成する
アクター	利用者
前提条件	認証している
主シーケンス	1. 利用者は予約内容(開始日、終了日、部屋数)をシステムに入力する。 2. システムは日付の正当性を検査する。 3. システムは日付の範囲を検査する。 4. システムは客室 DB に予約内容を送り、該当期間の空室数を得る。 5. システムは(作成者、作成日、予約内容)から予約記録を作成する。 6. システムは期間内空室リストに基づいて客室 DB を更新する。 7. システムは予約記録を予約記録 DB に登録し、予約記録 DB は予約 ID をシステムに返す。 8. システムは利用者に予約 ID を表示する。
代替シーケンス	2.で日付が不当な場合(例:9月31日の場合)、システムは利用者に日付エラーメッセージを表示する。 3.で予約期間が不正な場合($0 \leq \text{OUT} - \text{IN} \leq 30$ の範囲以外)、システムは利用者に範囲エラーメッセージを表示する。 4.で空室数が不足な場合、システムは利用者に空室不足エラーメッセージを表示する。
事後条件	予約が作成されている。

表 A.4: ユースケース記述(予約照会する)

ユースケース名	予約照会する
アクター	利用者
前提条件	予約作成している
主シーケンス	1. 利用者は予約 ID をシステムに入力する。 2. システムは予約 ID の正当性を検査する。 3. システムは予約記録 DB から予約記録を取得する。 4. システムは権限があるかを検査する。 5. システムは利用者に予約内容を表示する。
代替シーケンス	2.で予約 ID が不正な場合、システムは利用者に ID 不正エラーメッセージを表示する。 4.で利用者に予約記録を参照する権限がない場合、システムは利用者に権限なしエラーメッセージを表示する。
事後条件	特になし

表 A.5: ユースケース記述(解約する)

ユースケース名	解約する
アクター	利用者
前提条件	予約照会成功している
主シーケンス	1. システムは部屋割り当てに基づいて客室 DB を更新する。 2. システムは予約記録 DB から予約記録を削除する。 3. システムは成功メッセージを利用者に表示する。
代替シーケンス	特になし
事後条件	解約がされている。