

問題領域を特化した Web アプリケーションフレームワーク構築方法の 実験と評価

Domain-Specific Framework for Web Application and the Evaluation

周 鋒†
Feng Zhou

中所 武司†
Takeshi Chusho

1 はじめに

インターネットの普及とデジタルコンテンツの拡大により、掲示板、ショッピングサイトなど、Web アプリケーションの利用が一般化している。

しかし、各アプリケーションをゼロから開発するのは容易ではない。そこで、再利用技術を用いて Web アプリケーションを効率的に開発するために、これらの Web アプリケーションに共通するアーキテクチャやクラス間の連携方法などの仕組みに着目して基盤となるフレームワークを開発する技法が重要になっている。

我々のこれまでの研究では、Struts[1]、Hibernate[2]といった汎用的なフレームワークよりも問題領域を特化した予約業務フレームワークを開発し、適用評価を行った結果、会議室予約では 61%、商品予約では 71%の再利用率を確認できた[3]。

一般には、問題領域を狭い範囲に限定すると再利用率が高くなり、広い範囲を想定すると再利用率が低くなるというトレードオフの関係がある。

今回、問題領域と再利用率のトレードオフの関係を定量的に明確化するため、広い問題領域と考えられる予約業務とそれよりも狭いルーム予約業務の2つのフレームワークを開発し、それぞれ商品販売と教室予約のアプリケーション開発への適用実験を行った。

2 システムアーキテクチャ

従来の3層アーキテクチャ[4]は、ユーザインタフェースを提供するプレゼンテーション層、アプリケーションの処理を行うアプリケーション層、データを管理するデータ層で構成されている。実装レベルでは、それぞれ Web ブラウザと Web サーバ、アプリケーションサーバ、データベースサーバに対応する。

但し、JSP/Servlet を中心とした開発では、HTML と Java コードの混在のため、互いの変更がもう一方に与える影響が大きいことや、画面フローが把握しづらいことなどにより、システム全体を理解しづらく、開発と保守の容易性を低下させてしまうという問題があり、Struts に代表される Web アプリケーションフレームワークが開発された。

Struts は MVC モデルに準拠しており、アプリケーションの状態を保持するモデルと、表示・出力を司るビュー、入力を受け取ってその内容に応じてビューとモデルを制御するコントローラの役割が明確に分離されている。

今回は、図 1 に示すように、3層アーキテクチャに Struts を適用するとともに、Struts が対応していないビジネスロジックとデータベースアクセスも加えることにより、より高い保守性と拡張性を実現した。

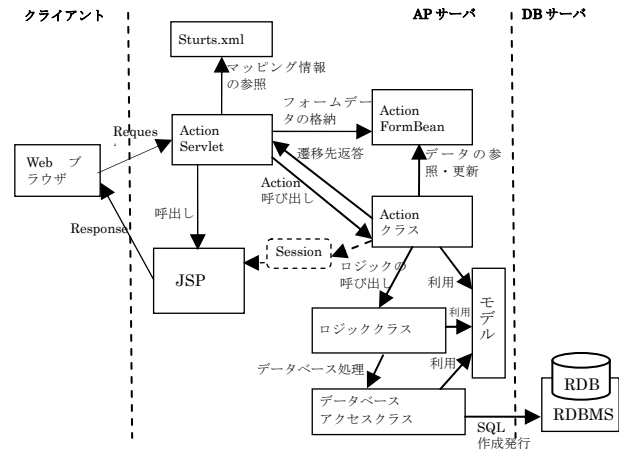


図 1. システムアーキテクチャ

3 システム概要

本研究では、対象とする問題領域を予約業務とし、従来のような書類や電話による施設予約やチケット予約などの手続きを Web 上で実現できるようにする。

フレームワークとしては、広い問題領域の予約業務と狭い問題領域のルーム予約業務の2種類を開発した。広い範囲の予約業務フレームワークにおける予約業務の定義は「存在する資源を一定期間占有することをあらかじめ宣言すること」としたので、施設予約、チケット予約、商品予約といったあらゆる予約業務が含まれる。その機能仕様を次の表 1 に示す。

表 1. 予約業務の機能仕様

	一般ユーザー	システム管理者
ユーザー管理	ログイン	ログイン ユーザー登録・変更・削除
資源管理	—	資源登録・変更・削除
予約機能	新規予約・予約照会・変更・削除	新規予約・予約照会・変更・削除

狭い範囲のルーム予約業務フレームワークでは、予約業務の子問題領域として、会議室予約、教室予約などのルームの予約を対象とする。機能としては、表 1 の予約業務機能に、時間チェックなどルーム予約に特有な機能を追加した。

なお、今回開発したフレームワークでは、表 1 の機能のうち、予約業務の核となる一般ユーザーの予約機能とログイン機能の2つの機能を対象とした。

†明治大学大学院理工学研究科基礎理工学専攻
ソフトウェア工学研究室

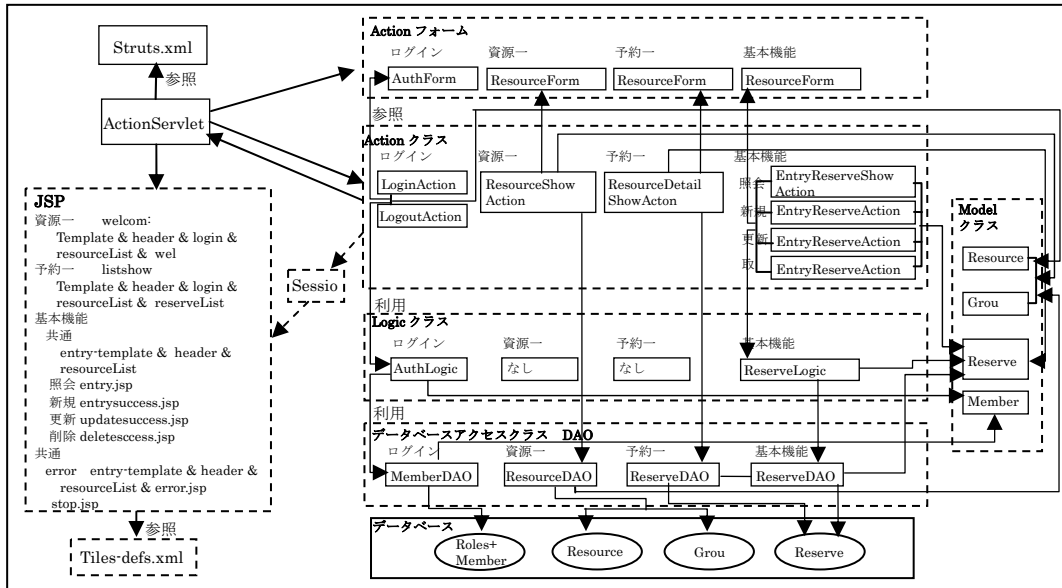


図 3. 出した予約業務フレームワーク

4 構築実験

4.1 フレームワークの構築方法

今回のフレームワークの構築と評価は、「機能分」と設計「問題開発」「フレームワーク抽出」「適用評価」という流れとなっている。特に、3の機能仕様に基づいて一般的な問題システムを開発し、この問題システムをベースに予約業務の共通部分を抽出して、フレームワークを構築した。

4.2 例題開発

3の機能仕様に基づいて、予約業務フレームワークとルーム予約業務フレームワークの両方を出すための会議室予約システムを開発した。

この問題システムでは、図2のように、ログイン画面、会議室一覧画面、明細画面からなる期画面が表示され、会議室を予約すると予約状態の照会ができ、さらにログインしたユーザの予約の照会・変更・取ることができるようになっている。

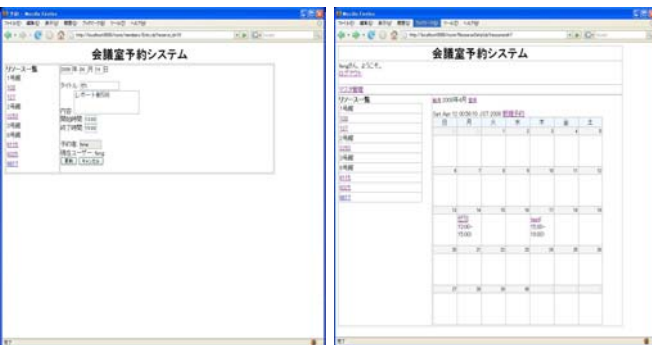


図 2. 会議室予約システムの画面

4.3 予約業務フレームワークの抽出

広い問題領域の予約業務フレームワークの抽出は、それぞれ新規予約、変更などの機能ごとに行った。各機能に対応するクラス、設定ファイル、データベースのうち、予約業務としての共通部分をフレームワークとして抽出した。

した。結果を図3に示す。

下図の要な出分の抽出について、データベース予約対象とする資源を保存するResourceテーブル、資源分類のGrouテーブル、ログインのためのmember, rolesテーブル、予約情報を管理するReserveテーブルが共通となる。各テーブル内共通のフィールドもそれぞれ抽出しているが、この抽出は以下の通りである。

Actionクラス ActionServletから呼び出され、必要な期化を行い、適切なロジッククラスを呼び出し、結果を受けてリンク先に遷移する。資源表示のResourceShowAction、予約・照会・変更・取のためのEntryReserveActionとEntryReserveActionを共通として抽出した。

Actionフォーム ユーザから入力されたデータをアクションへ渡す役割を行う。ResourceFormなどのフォームクラスとStruts.xml設定中で定義したインプットフォームを抽出した。

Modelクラス Resource, Grou, Reserve, Memberを抽出した。

Logicクラス ビジネスロジックを処理するクラスである。ReserveLogicというクラスで予約の各処理を行う。

データベースアクセスクラス 資源のアクセスのためのResourceDAOと予約のアクセスのためのReserveDAOを抽出した。

ビュー 表示するJSPファイル、レイアウトを管理するテンプレートファイルおよびその設定ファイルを抽出した。

Struts.xml アクションマッピングの設定として共通の部分を抽出した。

結果として、この問題のステップ1444のうち、抽出した予約業務フレームワークは895ステップとなった。

4.4 ルーム予約業務フレームワークの抽出

次に、問題領域を狭くしたルーム予約業務フレームワークの抽出も行った。ルーム予約業務は予約業務の子問題領域であるため、抽出したフレームワークのクラスとファイル類については、この予約業務フレームワークと

じである。下、予約業務フレームワークとのなるこののみをる。

ロジッククラス 予約したい時間 に 存の予約があるかどうかのチェックのための 処理クラス

クラス ロジッククラスでの時間チェックのための発行処理

ビュー 予約一 表示の 間画面はルーム予約業務ではスタマイ なしで共通になった 予約業務にて一 共通として 出したが、テンプレートとその設定は 全に共通になった 予約時間が 存の予約と っている の 表示 ージも追加した。

のマッピング設定も予め められるため、アクションマッピングの設定については、スタマイ なしで 全共通になった。

結果として、 出したルーム予約業務フレームワークのステップ は となり、予約業務より ステップ 加した。

5 適用実験と評価

本研究で 出した広い領域の予約業務フレームワークと狭い領域のルーム予約業務フレームワークをシステム開発に適用する実験を行い、再利用率の で評価を行った。

5.1 予約業務フレームワークの適用試験

予約業務の「存在する資源を一定期間占有することをあらかじめ宣言すること」の定義の範囲内で、フレームワークの 出用に用いた会議室予約システムとかなり性 のなる書 販売システムを適用実験の対象として した。書 販売システムは、 的な予約業務ではないが、 入したい商品を 的に占有することを宣言するという で予約業務の一種と考えられる。

開発した書 販売システムにおける予約業務フレームワークの割合 は、表2に示す。

表2.予約業務適用実験におけるフレームワークの割合

	Action クラス	Action Form	モデ ル	ロジッ ククラ ス	DAO クラ ス	ビュ ー	アクシ ョンマ ッピン グ	
全体	268	84	168	84	343	402	70	1419
フレーム ワーク	234	47	84	49	245	175	61	895
業務 有	34	37	84	35	98	227	9	524
フレームワ ークの割	87%	56%	50 %	58%	71 %	43 %	87%	63%

システム全体の1419行のソースコードの中の895行がフレームワークで提供されているので、63%の手間をくることができたことになる。よって、予約業務フレームワークの、かなり性 のなる予約業務も対応できるという が確認できた。但し、全クラスとファイルのスタマイ が 要となっている。

5.2 ルーム予約業務フレームワークの適用試験

ルーム予約業務は、問題領域を狭く取り、ルームの予約を対象とするため、適用実験では、学 の教室予約システムを した。

フレームワーク 出に用いた会議室システムの予約は 時 分から 時 分までという通 の時間 定になっているが、教室予約システムでは、あらかじめ められた時限 で予約を行う。その か、予約に 要な情報についても なる 分がある。

開発した教室予約システムにおけるルーム予約業務フレームワークの割合 は、表3に示す。

表3.ルーム予約業務適用実験におけるフレームワークの割合

	Action クラス	Action Form	モデ ル	ロジッ ククラ ス	DAO クラ ス	ビュ ー	アクシ ョンマ ッピン グ	
全体	279	98	126	78	292	465	63	1401
フレーム ワーク	234	47	84	64	266	372	63	1130
業務 有	45	51	42	14	26	93	0	271
フレームワ ークの割	83%	48%	66 %	82%	91 %	80 %	100%	81%

この適用実験で81%の再利用率を成できたことから、かなり性 のい業務に適用する、再利用率が大 に上できるといえる。問題領域を狭くすると処理する業務も明確になるため、フレームワークを通じて設 の分もある 再利用できるようになることを確認できた。

5.3 総合評価

適用実験により、下の結 を た。

- (1) トレードオフ関係の定量的な確認
問題領域と再利用率の間にトレードオフの関係があるという定性的な認 があるが、今回その関係を定量的に確認できた。すな ち、広い領域の実験では63%の再利用率であったのに対し、狭い領域の は、81%の再利用率が 成された。よって、システム開発する、出来る け一 適用できる狭いフレームワークをす きであるといえる。
- (2) 拡張性・保守性
開発したフレームワークは、MVC モデルをベースに、ビジネスロジック層、データベースアクセス層も加えたため、全体を理解しやすくなり、拡張・保守しやすくなるという利 もある。

参考

- [1] The Apache Software Foundation
<http://struts.apache.org/>
- [2] Red Hat, Inc.
<http://www.hibernate.org/>
- [3] 中 司, , 予約業務を 題とした Web アプリケーション用フレームワークの再利用性の評価, 電子情報通 学会 D-I 分 , Vol.J88-D-I, No.5, pp.930-939 (May. 2005)
- [4] 中 司, Javaによる Web アプリケーション入 , サイ ンス (Feb. 2005)