

社会科学系学部における 情報科学（コンピュータ）教育（その3）

—電子商取引を行う情報システム構築—

藤 尾 好 則

目 次

1. はじめに
2. 電子商取引に基づく販売管理
3. 電子商取引の枠組み
4. 設計
5. おわりに

1. はじめに

「社会科学系学部における情報科学（コンピュータ）教育（その2）」[1]では、情報システムの構築に関する講義の内容を学生に具体的に理解させるため、従来の実取引を行う情報システムを対象とした実習教材について述べた。本論文では、電子商取引を行う情報システムを対象とした教材を取り上げ、情報システムの構築や運用について従来の情報システムとの違いについて考察する。

2. 電子商取引に基づく販売業務

次のような電子商取引を行うシステムを構築する。

- 1) 複数の電子商店から構成される電子商店街（バーチャル・モール）をインターネットのホームページ上に開設する。
- 2) 顧客は電子商店街の中から買いたい商品がある電子商店を選び、そのホームページにある商品を選び、ショッピングカートに入れる。
- 3) 商店街の精算所で、ショッピングカートの中の商品を注文する。

図1に買い物の手順を図示する。

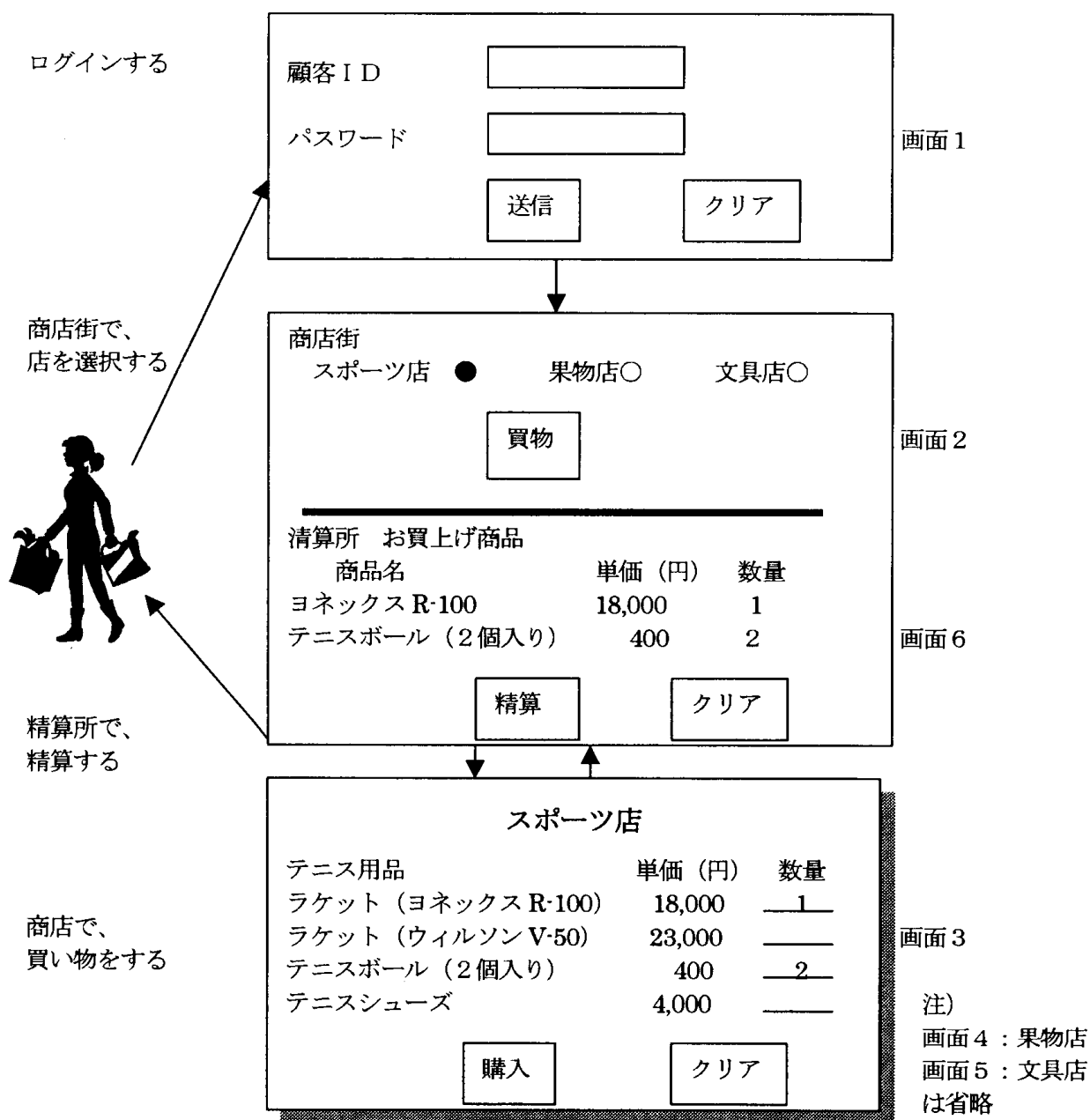


図1 買い物の手順

電子商取引は、従来の販売業務の実取引を仮想取引（ホームページの画面）で行うことが一番の違いで、その他の配送、受入、経理の業務は従来と基本的には同じである。

3. 電子商取引の枠組み

システムのアーキテクチャは3つの要素であるヒューマンインタフェース（画面）、ビジネスプロセス（業務手順）、データベースで構成される。論文[1]では業務処理をアプリケーション又はアプレットで実現したが、クライアントに負荷がかかり処理速度が遅くなるため、本論文ではWebサーバ上で業務処理を効率よく実施できるサーバレットの使用を前提に2とおりの設計方法でシステムを構築した。従って、2つのシステムはクライアントよりもサーバ中心のシステム構成であるため、クライアントの負荷が少なくモバイルパソコンでも使用できる。

3.1 ヒューマンインタフェース（プレゼンテーション層）

ヒューマンインタフェースは顧客が商品を注文するWebの入出力画面を設計することである。その画面の例（画面1、画面2、画面3、画面6）を図1に図示している。画面3のスポーツ店と同様な処理画面として、画面4の果物店、画面5の酒店があるが、この画面の図示は省略した。

3.2 ビジネスプロセス（ロジック層）

クラス／オブジェクトを抽出してUML記述で販売業務に関するクラス図（図2）を作成する[2]。このクラス図を基に画面やデータベースとの注文処理手順をシーケンス図（図3）に図示する。

3.3 データベース

注文処理に関する表のリレーション（関係）を設計する。このデータベースの設計は（その2）[1]と同じデータベースを使用する。

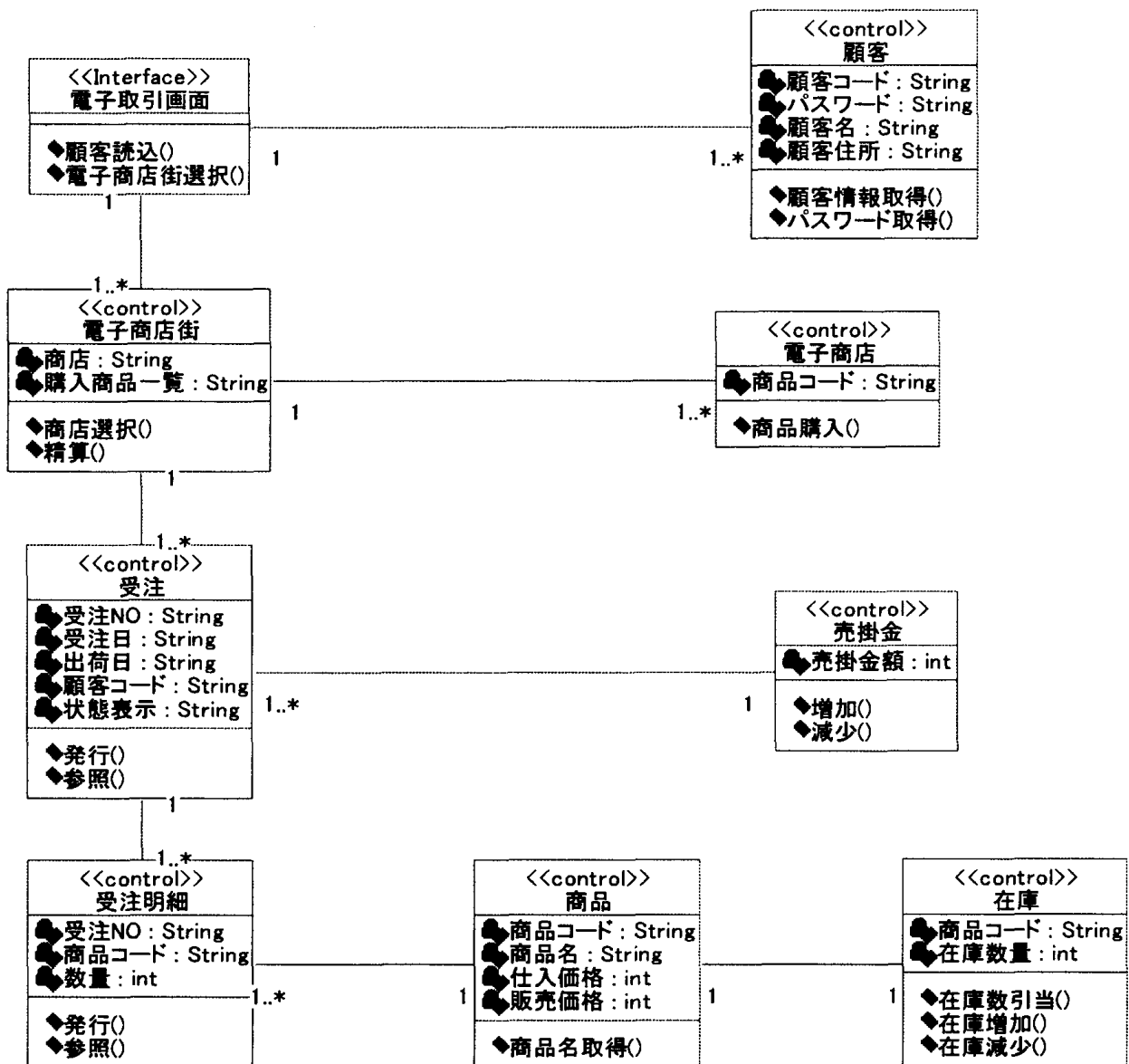


図2 クラス図

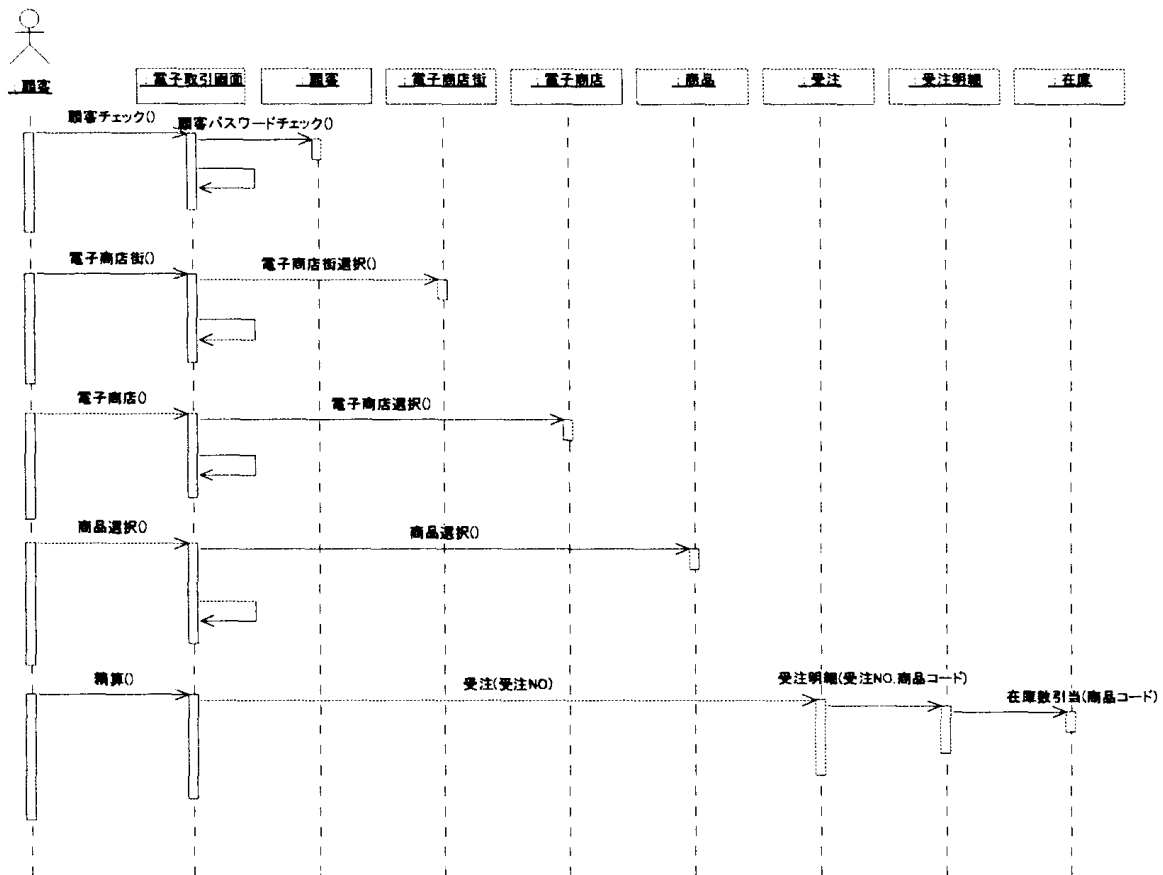


図3 シーケンス図

4. 設計

4.1 システムの物理的な構成

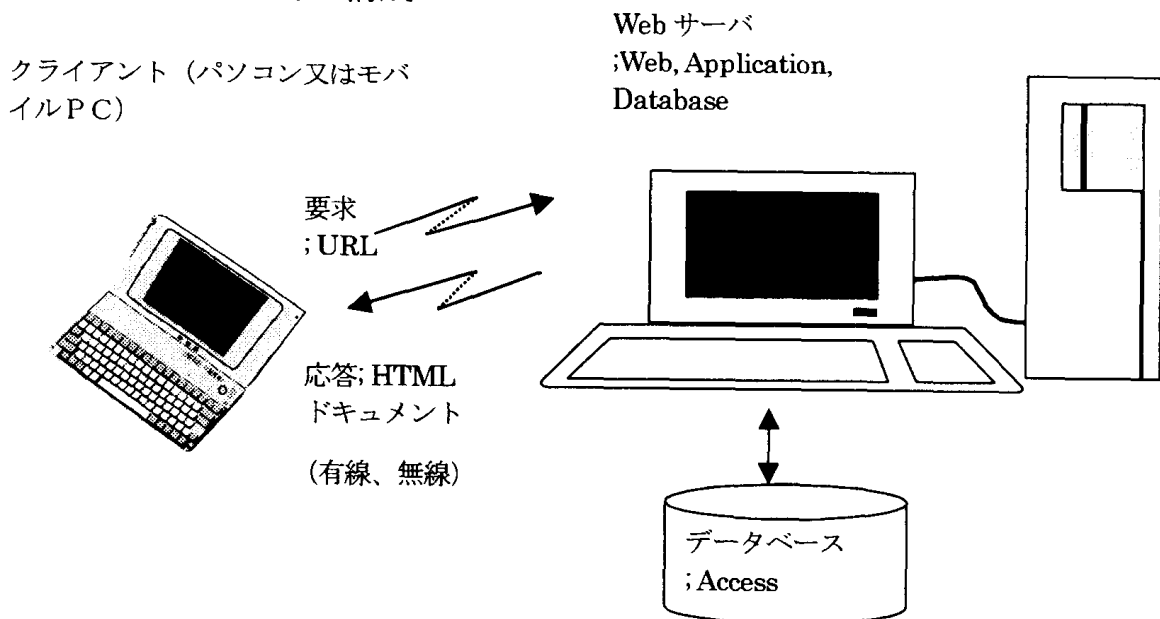


図4 システム構成

4.2 システム・アーキテクチャ

電子商取引システムを構築するには、3つの機能に分割できる。Webへデータを表示するプレゼンテーション層、ビジネスの業務を処理するロジック層及び顧客や商品のデータを蓄積するデータベースである。ここでは2つのシステム・アーキテクチャで開発してオブジェクト指向の適用、部品化や保守性を比較する材料を提供する。一つはサーブレット1個でプレゼンテーション層、ロジック層、データベースとの検索や更新等、一連の操作や処理を実施するアーキテクチャ（図5）で開発する。他方のアーキテクチャは、プレゼンテーション層をJSP複数個（5個）で処理し、ロジック層とデータベースの検索や更新をサーブレット複数個（6個）で処理する方法である（図6）。

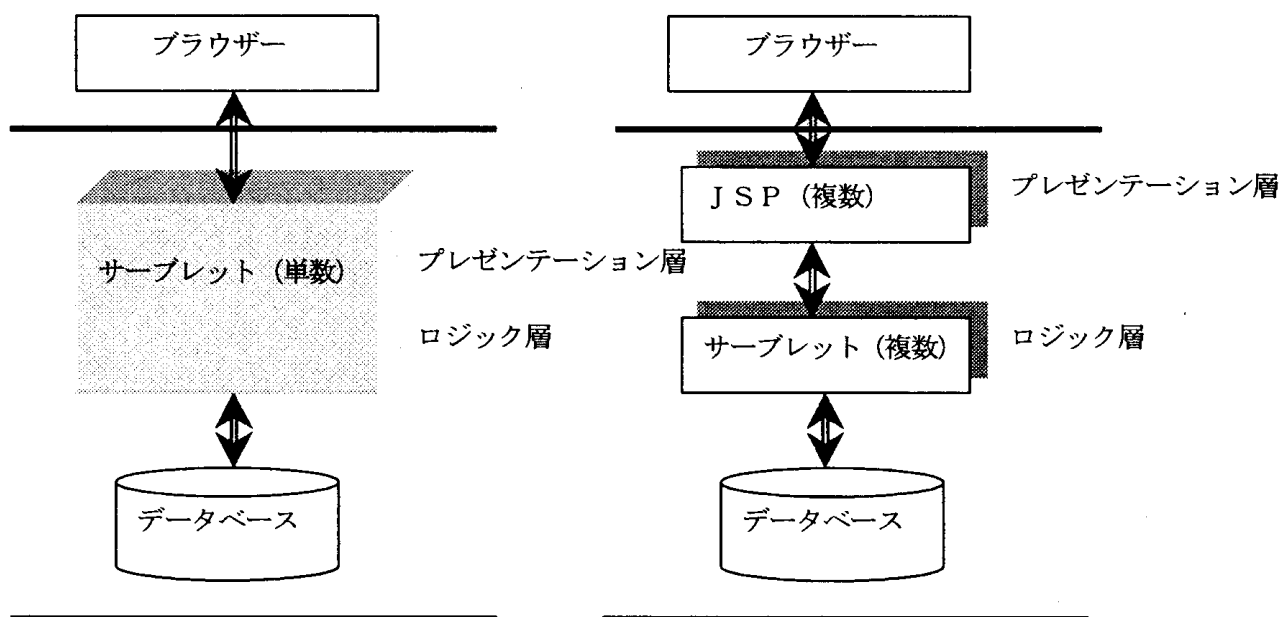


図5 サーブレット

図6 サーブレット+JSP

(1) サーブレット (単数) の処理の流れ

- ①クライアントはサーバに対して、実行させるサーブレットのURLを指定してサーブレット (Servlet) の実行を要求する。
- ②サーバはクライアントからの要求を解読し、サーブレットの実効命令であればサーブレットを起動する。
- ③サーバから起動されたサーブレットが処理を行う。

④サーブレットは、実行結果をHTMLドキュメントとしてクライアントに送る。

⑤クライアントは、サーバから送られてきた実行結果を表示する。

次にクライアントのWeb画面から、サーバのサーブレットへの通信（パラメータの受渡し）について述べる。

サーブレットの起動は、HTMLドキュメントの<FORM>タグのACTION属性で設定する。

“http://ホスト部.ドメイン部/ディレクトリ/サーブレット名”

又、フォームで入力されたデータをサーブレットへ送信して処理する方法は<FORM>タグのMETHOD属性であるPOSTメソッドを用る[3][4]。

例 <FORM ACTION = “http://www.pu-kumamoto.ac.jp/servlet/ServletApp” METHOD = “POST” >

フォームの内容を記述する[5]。

</FORM>

<INPUT>タグは<FORM>タグの中で使用し、入力エリアとボタンを表示する。

次の3つの属性がある。

TYPE = “text | password | radio | hidden | submit | reset”

NAME = “入力したデータを引数として渡すときのフィールド名”

VALUE = “データの初期値” | “ボタン名”

SIZE = テキストボックスの横幅（ピクセル単位）

例 <INPUT TYPE=“text” NAME=“user” VALUE=“Shimei”>

TYPE属性で指定した場合の機能は次ぎのとおりである。

- text テキストを入力するためのテキストボックスを表示する。
- password パスワードを入力するためのテキストボックスを表示する。

入力文字数と同じ数だけ“*”が表示される。

- radio ラジオボタンを表示する。
- hidden 不可視フィールドを指定する。
- submit フォームの値を指定したURLに送信するためのボタンを表示す

る。

- reset 入力したフィールドの内容を全て初期値に戻すためのボタンを表示する。

<SELECT>… </SELECT>タグは選択リストを作成するために使用し、<OPTION>と</OPTION>に挟まれた文字列、数値が選択項目となる。その属性NAMEは、入力したデータを引数として渡すときのリスト名を表す。

```
例 <SELECT NAME = "list">
    <OPTION>選択値 1 </OPTION>
    <OPTION>選択値 2 </OPTION>
</SELECT>
```

(2) サーブレット (単数) の概要フローチャート

WWWにおいて、クライアントとWebサーバ間の通信に使用されるHTTPプロトコルはステート・レスなプロトコルで、1回のリクエスト・レスポンス・チェーンでコネクションが切断される。次ぎの表示ページをリクエストすると新たなリクエスト・レスポンス・チェーンに切り替わるため、Webサーバ側では1つ前のリクエストのクライアントと同じクライアントであることを識別できない。

クライアントを識別するにはクライアントの情報を保持するための機能である SrvletAPI の HTTP Cookieとセッション管理の2つの方法がある[6][7]。Cookie はクライアント側で Session オブジェクトを管理する。しかし、Cookie はブラウザの機能を利用するためブラウザが対応していなければ、この機能を利用できない。一方、セッション管理はサーバ側で Session オブジェクトを管理する。このためリクエスト・レスポンス・チェーンとは独立に永続的に Session オブジェクトが維持される。本論文ではセッション管理[4]を用いてクライアントを識別することにする。

買い物の手順 (図1) と概要フローチャート (図7) を基にサーブレット処理の詳細について説明する。

顧客 (クライアント) はサーバのサーブレットにアクセスする。サーブレットへ初めてのアクセスであるため (図7 ; 最初のセッション=YES)、顧客 I

Dとパスワードを入力する画面1（図1）が表示される。顧客は顧客ID、パスワードを入力して送信ボタンをダブルクリックする。再度サーバの同じサブレットに制御が渡り（図7の①へブランチ）、サブレットを起動する。2回目のサブレット起動（最初のセッション=NO）では顧客IDとパスワード入力へブランチせず、処理判別（図7）の処理に移る。処理判別ではどの画

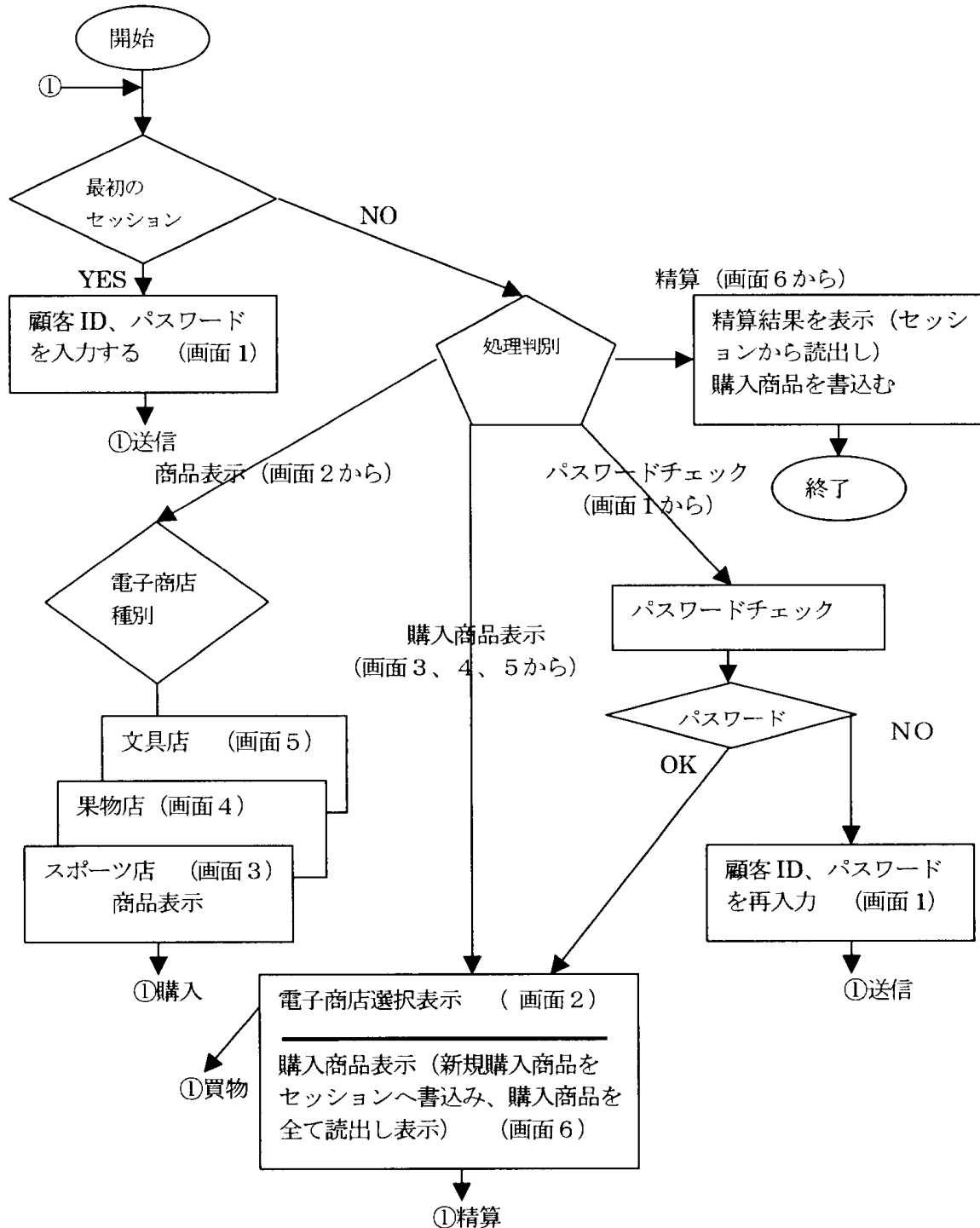


図7 サーブレット（単数）の概要フローチャート

面からブランチしてきたかを判別する。画面1からブランチしてきた場合は、パスワードをチェックしてOKなら画面2と画面6（図1）を表示する。パスワードエラーの場合の処理は省略する。

電子商店選択画面2（図1）では電子商店を選択して買物ボタンをダブルクリックする（図1ではスポーツ店を選択）。再度サーバレットのへブランチし、処理判別では画面2からのブランチであるためスポーツ店の画面3（図1）を表示する。この画面で購入商品の数量を入力して、購入ボタンをダブルクリック。再度サーバレットへブランチ、処理判別では画面3からのブランチであるため画面2及びこれまでに購入した商品と画面6を表示する。画面2で再度、別の電子商店を選択して商品を購入することもできるが、これ以上商品を購入せず精算処理を行う場合は画面6の精算ボタンをダブルクリックする。この処理により購入商品・購入金額をデータベースに書き込み買い物の処理を終了する。これで一連のセッションを完了する。

(3) サーブレット（複数）+ J S P [8][9]（複数）の処理の流れと概要フローチャート

- ①クライアントは、顧客ID、パスワードを入力する画面のあるSHTMLのURLを指定してSHTMLを表示（図8；画面1）させる。
- ②クライアントは顧客ID、パスワードを入力した後、サーバのパスワードをチェックするサーバレットを起動する。
- ③サーバレットはパスワードをチェックして、OKなら電子商店街と精算画面（図8；画面2と画面6）を表示するJ S Pへ制御を渡す、NOであれば、顧客IDとパスワードを再入力する画面を表示するJ S Pへ制御を渡す。
- ④電子商店街と精算画面では商品を購入するため電子商店の選択を実行する。
- ⑤サーバの表示商店選別サーバレットが起動されて、選択された電子商店の商品を表示するJ S Pへ制御を渡す。
- ⑥J S Pは電子商店の商品を表示する（図8；画面3）。顧客は購入商品の数量を入力し、サーバの新規購入商品を書き込むサーバレットへ制御を渡す。

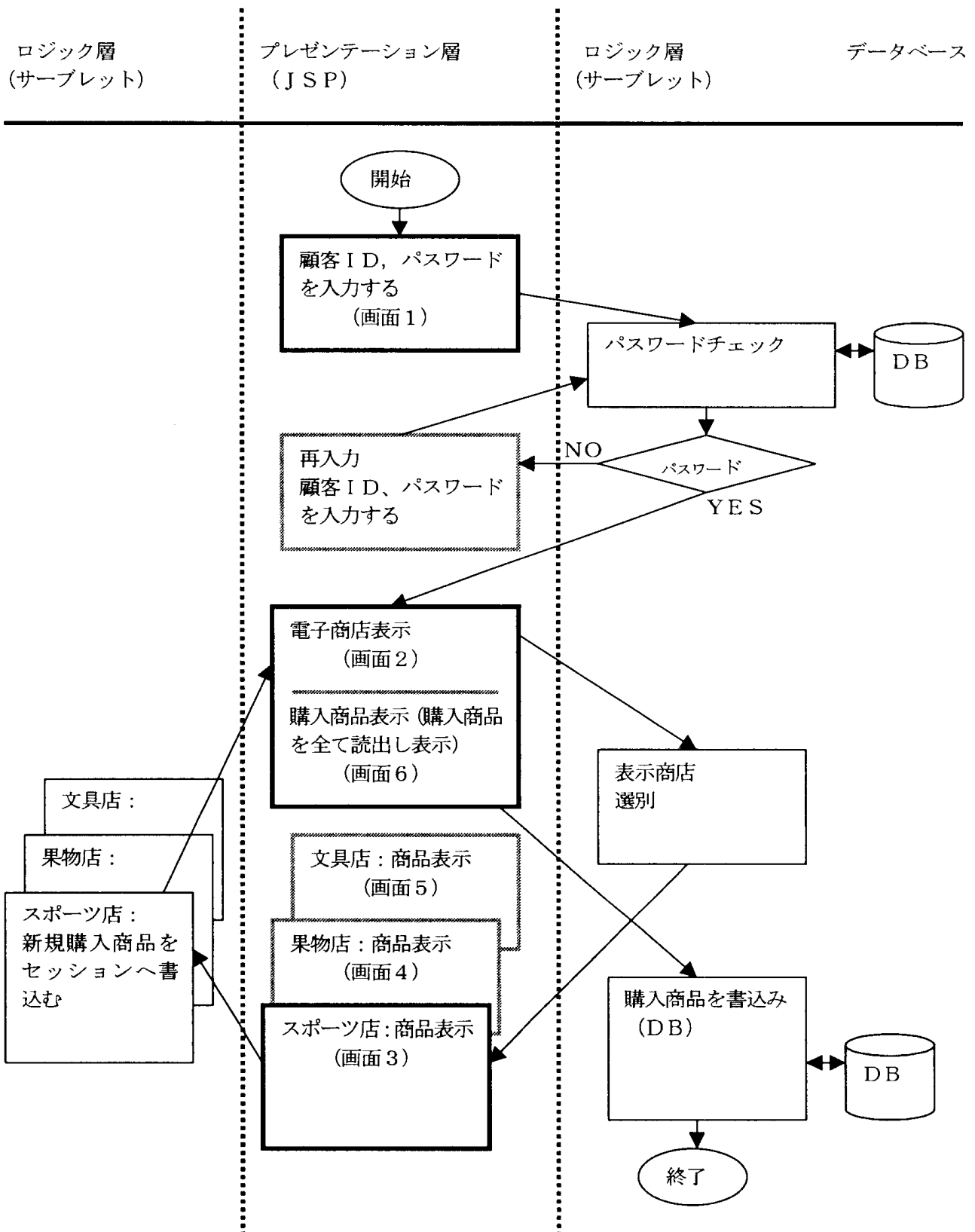


図8 サーブレット（複数）+JSP（複数）の概要フローチャート

- ⑦サーブレットは新規購入商品をセッションに書込み、再度、電子商店街及びこれまでに購入した商品全てと精算画面を表示する。
- ⑧顧客は、まだ購入する商品があれば別の電子商店を選択する。これ以上購入する商品がない場合は精算を選択すると制御はサーバの商品書込みサーブレットに渡される。
- ⑨商品書込みサーブレットは、データベースに購入した商品コード、数量、金額等を書込み一連の処理を終了する。

JSPはプレゼンテーション層、サーブレットはロジック層に機能を分割し、さらにJSP、サーブレットも機能別に分割して複数個で構成した。将来の仕様変更や設計変更が発生した場合にも、その影響は変更部分のJSP又はサーブレットに留まるため、保守性が一段と向上する。

機能を分割して設計したためにプレゼンテーション層とロジック層の間で通信（パラメータの受渡し）が頻繁に発生する。この通信が設計のポイントとなるが、まずサーブレットからJSPへの通信は次ぎのとおりである。[10]

例 String url = "/Commerce/sample.jsp";

→ urlはJSPファイルがあるWebサーバ、パスとJSPファイルを指定する。“http://ホスト部.ドメイン部/ディレクトリ/JSP名”、但しサーブレットと同じWebサーバ上にJSPファイルがある場合は、urlはJSPファイルへのパスとJSPファイルを指定する。

```
ServletContext context = getServletContext();
```

```
RequestDispatcher rd = context.getRequestDispatcher(url);
```

```
rd.forward(request, response);
```

一方、JSPからサーブレットへの通信は、(1)サーブレットの処理の流れで述べたものと同じ方法を用いる。即ち、JSPのHTMLドキュメントの<FORM> タグのACTION属性で設定する。

“http://ホスト部.ドメイン部/ディレクトリ/サーブレット名”、但しJSPファイルと同じWebサーバ上にサーブレットがある場合は、“http://ホスト部.ドメイン部”を省略して、サーブレットへのパスとサーブレット

名（.classを除いて）を指定する。

同様にフォームで入力されたデータをサーブレットへ送信して処理する方法も<FORM>タグのMETHOD属性であるPOSTメソッドを用る。

例 <FORM ACTION="/servlet/Commerce.Check" METHOD="POST">

フォームの内容を記述する[5]。

</FORM>

→CommerceプロジェクトのCheck.classサーブレットで、ディレクトリservlet下にあるCommerce.jarに含まれるCheckサーブレットである。

5. おわりに

論文[1]では、次の手順に従って対象システムを分析、設計した。

現状の対象システムに関して、卸売店の組織と情報の流れ（現状システム）、販売業務の伝票の流れ（現状システム）を分析し、次に新システムの構想に基づく、卸売店の組織と情報の流れ（新システム）、販売業務の情報、伝票の流れ（新システム）を設計した。次に新システムの枠組み、部品を用いたプログラム設計を行った。

電子商取引においても、同様な手順を踏み分析設計を実施した。しかし、本論文では前半部分の現状システムと新システムの分析・設計を省略し、後半部分の電子商取引の枠組みと設計に焦点をあて、論文[1]との違いについて述べた。その違いは、論文[1]で用いたアプレットと、本論文で使用したサーブレットによるところが大きい。

アプレットはクライアントに負荷がかかる。原因はサーバからクライアントにプログラムをロードしてくるため実行できるようになるまでに時間がかかること、GUI画面をJavaBeans[5]を用いて設計しているため伝票と同じ様式で画面を設計できるが、欠点としてロードモジュールが大きくなりクライアントのメモリ容量を多く必要とする。クライアントからサーバのデータベースへア

クセスするため処理時間がかかる。又、アプレットの表示がブラウザの機能に依存するためInternet ExplorerとNetscape NavigatorではHTMLの記述が異なる[11]。これを解決するため2種類の記述を用意する必要がある。

一方、サーブレットはサーバで動作するためクライアントの負荷が少ない。サーブレットで作成したGUI画面のHTMLをクライアントに送信し、クライアントはその画面を表示するだけである。サーブレットはサーバで実行されるためアプレットに較べて応答時間やデータベースへのアクセスが早い。しかし多数のクライアントからの要求を処理するにはサーバに負荷がかかる。このためサーバには高性能（メモリ、CPU）なコンピュータが要求される。

次に、同じシステムを異なるシステム・アーキテクチャで設計してオブジェクト指向や保守性を比較した。サーブレット+JSPのシステム・アーキテクチャは、オブジェクト指向のメリットを最大限に活用して保守性に富むシステムを構築することができる。

本論文では販売管理について述べたが、仕入・商品管理・経理業務も同様にして設計できる。全体システムが完成すれば、具体的な商品について販売・仕入・商品管理・経理業務を模擬的に実施して商店の運用を学生に体験させれば有益である。

又、販売商品であるが電子商取引にはどのような商品が適しているか、魅力的なホームページを作るにはどのようにするか等、マーケティング、デザインや広告の観点からの考察も必要になる。さらに外部のシステムを含めた決済、物流の方法等、ビジネスモデルの研究へと発展させシステム全体を総合的に考えさせる体験型教育教材となるものである。

参考文献

- [1] 藤尾好則：「社会科学系学部における情報科学（コンピュータ）教育（その2）」、アドミニストレーション、第6巻2～3合併号、pp.1-12、熊本県立大学総合管理学会(2000)
- [2] Terry Quatrani: Visual Modeling with Rational Rose and UML, Addison Wesley(1998)

- [3] John December: The World Wide Web 1997, samsnet(1997)
- [4] IPA, 京都ソフトウェアアプリケーション：ホームページ作成入門、京都ソフトウェアアプリケーション(1997)
- [5] インプライズ：Jbuilder開発者ガイド、インプライズ(1999)
- [6] Dustin R. Callaway: Inside Servlets, Addison Wesley(1999)
- [7] 原田洋子：Java Servlet、秀和システム(2000)
- [8] Damon Hougland, Aaron Tavistock: core JSP, Printece Hall(2001)
- [9] Larne Pekowsky: Java Server Pages, Addison Wesley(2000)
- [10] 杉野博史：「JAVAプログラムの呼び出し」、JavaServer Pages詳細、OpenDesign増刊, pp - 95、CQ出版(2000)
- [11] Sun Microsystems, Inc.: Java Plug-in HTMLの仕様、2000年4月10日検索www
:http://java.sun.com/products/plugin/1.2/docs/ja/tags.html