

# Webアプリケーション 学習用軽量コンテナ Sabaphy

OSSAJ主催第4回ミニセミナー

2005年10月13日

東京国際大学商学部情報システム学科

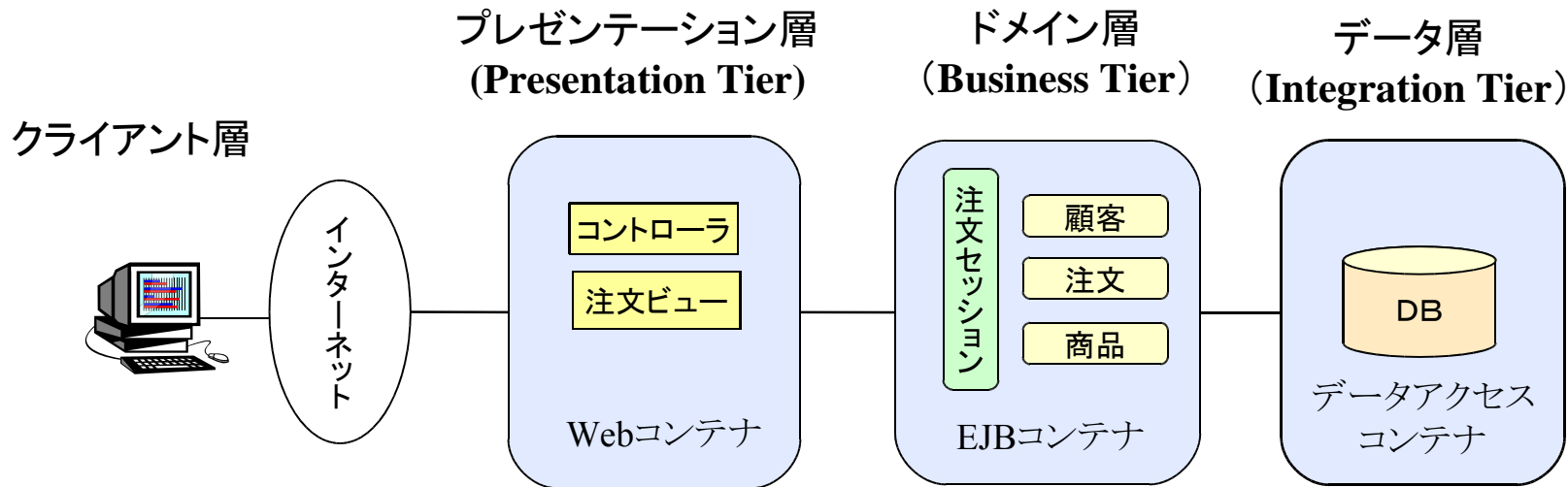
佐藤 英人

# Sabaphyとは

- **Webアプリケーション学習用オープンソースのコンテナとフレームワーク**
- **J2EEに準じた4層アーキテクチャに対応**
- **主な特徴**
  - 言語は **PHP5** (Javaライクの構文、キーワード)
  - 特別なインストールや設定は不要
  - 簡単なことは簡単に実行できる
  - 段階的に学習できる

Sabaphy: Sharable Architecture for Business Applications, PHP library edition

# J2EE アーキテクチャ



**1999.6 J2EE 発表**

**1999.12 EJB1.1 公開**

**2001.10 EJB2.1 承認**

EJBは難しい  
使いにくい

**2001~2003 Spring, Seaserなど軽量コンテナ登場**  
**Hibernate (O/Rマッピングコンテナ)**



**2004 EJB3 draft 検討開始**  
**PHP5 登場**

# Sabaphyの特徴

## 1. 習得が容易な言語を使用

- 静的型付けのJavaは難しい ⇒ スクリプト言語 PHP5

## 2. オープンソースのみを使用

- 教室でも自宅でも区別なく利用できる

## 3. 特別なインストール・設定は不要

- ライブラリをドキュメントルートにコピーするだけ
- 設定やテーブル定義が不要なシリアルDBも内蔵

## 4. 標準的な概念で構成

- DIコンテナ、EJB3相当、Hibernate型O/Rマッピング

## 5. 段階的な学習が可能

- 入門レベル → MVC → フレームワーク

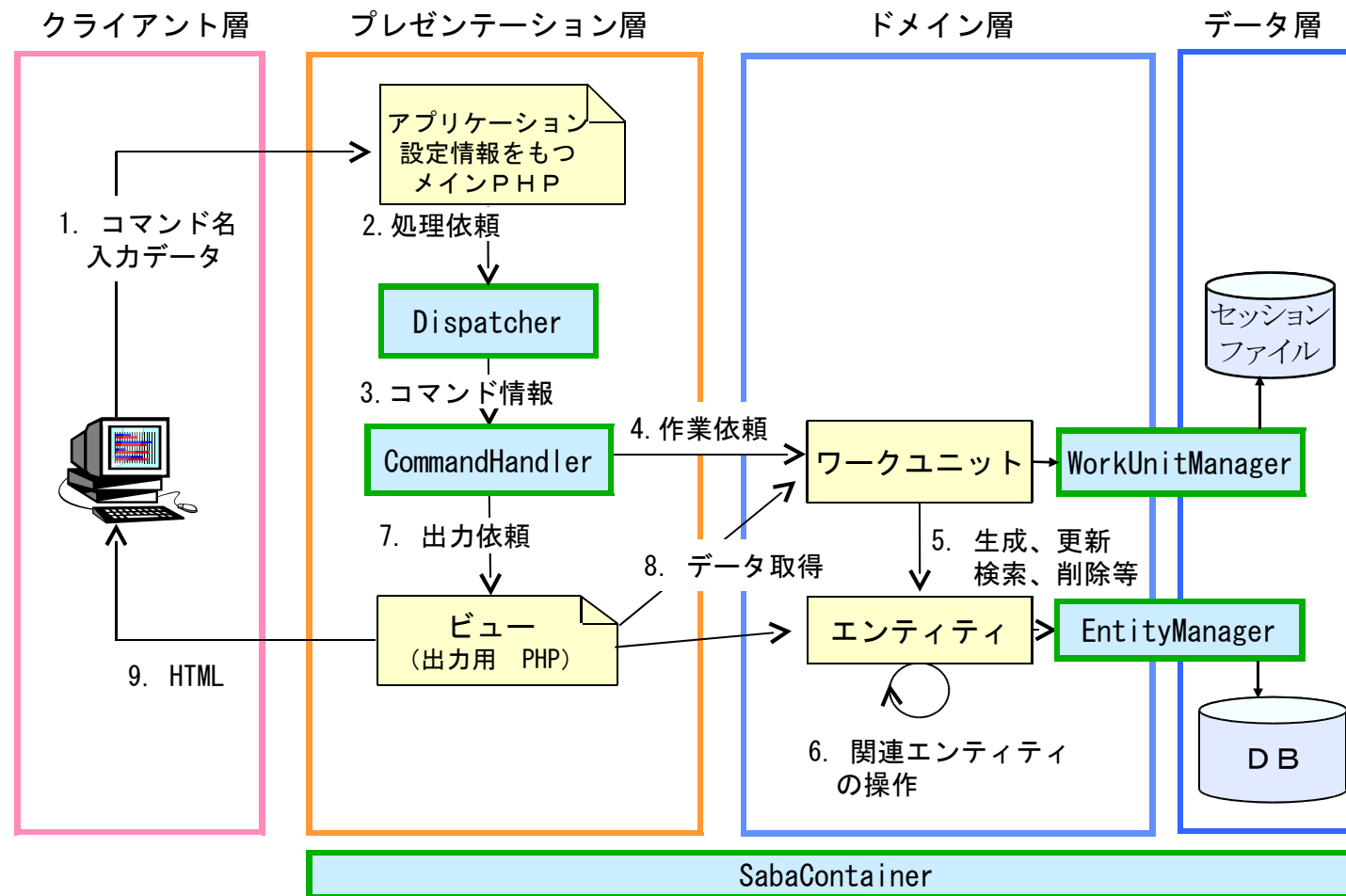
## 6. 可能な限り単純に

- 必要になるかどうか分からない機能は追加しない

## 7. ある程度実用的に使える

- アクセスが集中するサイトは別。通常用途では使用可能

# Sabaphyアプリケーションの構成 (フレームワーク使用時)



(凡例) **lib class** : ライブラリクラス **app class** : 利用者定義クラス **app php** : PHPプログラム (主としてデータ)

# 利用者作成プログラム

- **メインPHP**

- クライアントから呼び出されるPHPファイル。
- フレームワーク利用時は、アプリケーションの設定情報を与えて、Dispatcherに処理を依頼するだけ。

- **ワークユニット**

- セッションが維持されている間データを保持し、1つのトランザクションを実行するオブジェクト。
- EJBのStateful SessionBeanに相当する。
- 何も継承しない普通のクラス (JavaでいえばPOJO)

- **エンティティ**

- 属性データをDBに格納する永続オブジェクト。
- EJBのEntityBeanに相当する。
- 何も継承しない普通のクラス (JavaでいえばPOJO)

- **ビュー**

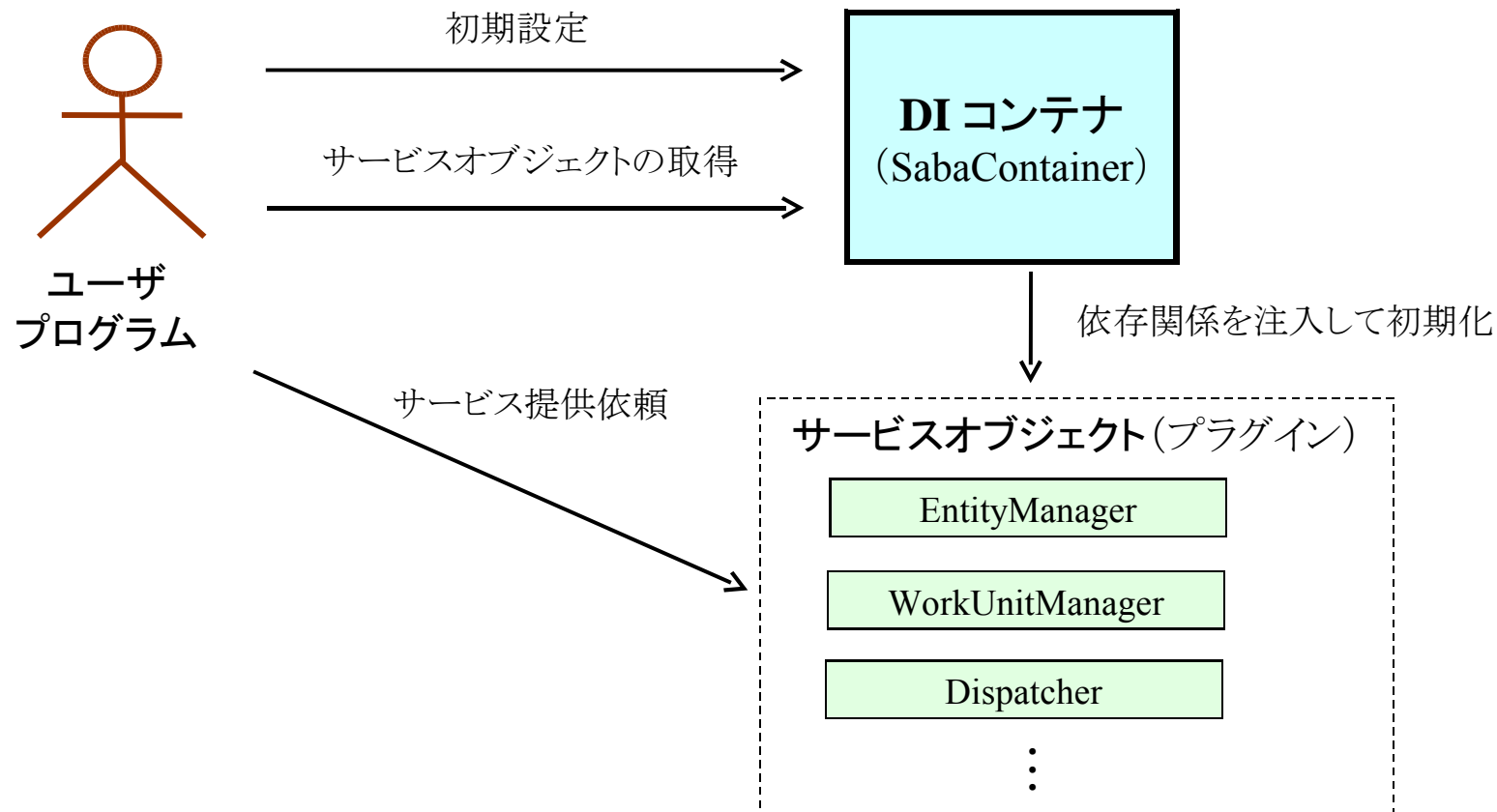
- 出力HTMLを生成するもの。
- Sabaphyでは、通常のPHPファイルを使用する。
- JavaでいえばJSPに相当する。

# 主なライブラリクラス

- **SabaContainer**
  - SabaphyのDIコンテナ
  - サービスオブジェクト間の関連付けとサービスオブジェクトの提供 (サービスロケータ機能)を行う。
- **Dispatcher**
  - Sabaphyフレームワークのファサード
  - 要求されたコマンドに応じた処理の振り分けを行う。
- **CommandHandler**
  - Sabaphyフレームワークのサブコンポーネント
  - ユーザ認証、ワークユニットの生成/再現/削除の依頼、ワークユニットがもつアクションの起動、ビュー生成の依頼を行う。
- **WorkUnitManager**
  - ワークユニットの新規保存/再現/削除を行う。
- **EntityManager**
  - エンティティの新規保存/更新/検索/削除を行う。
  - Hibernate Sessionを模したもので、EJB3のEntityManagerに相当する。  
。

# Sabaphyの仕組み

DIコンテナ/サービスマネージャ一般の動作



Sabaphyでは、主要なサービスオブジェクトは **グローバル関数** で取得できる。  
PHP5では、Javaと異なり、安全にグローバル関数、Static変数を使用できるから。



# ミニサンプル（顧客登録）

(1) 入力画面 (inputCustomer.html.php)

顧客入力フォーム - Microsoft Internet Explorer

アドレス(D) http://localhost/sabaphy02/miniSample/CustomerMgr/main.php 移動

## 顧客データの入力

名前:

住所

ページが表示されました イン트라ネット

(2) 確認画面 (confirmCustomer.html.php)

入力結果 - Microsoft Internet Explorer

アドレス(D) http://localhost/sabaphy02/miniSample/CustomerMgr/main.php 移動

名前=山田太郎  
住所=東京都目黒区碑文谷1-2-3

ページが表示されました イン트라ネット

入力終了

戻る

(3) 一覧表示画面 (showCustomers.html.php)

顧客一覧表示 - Microsoft Internet Explorer

アドレス(D) http://localhost/sabaphy02/miniSample/CustomerMgr/main.php 移動

## 顧客一覧表示

---

No.C0001 名前=山田太郎  
住所=東京都目黒区碑文谷1-2-3

No.C0002 名前=木村花子  
住所=埼玉県さいたま市栄2-3-4

ページが表示されました イン트라ネット

送信

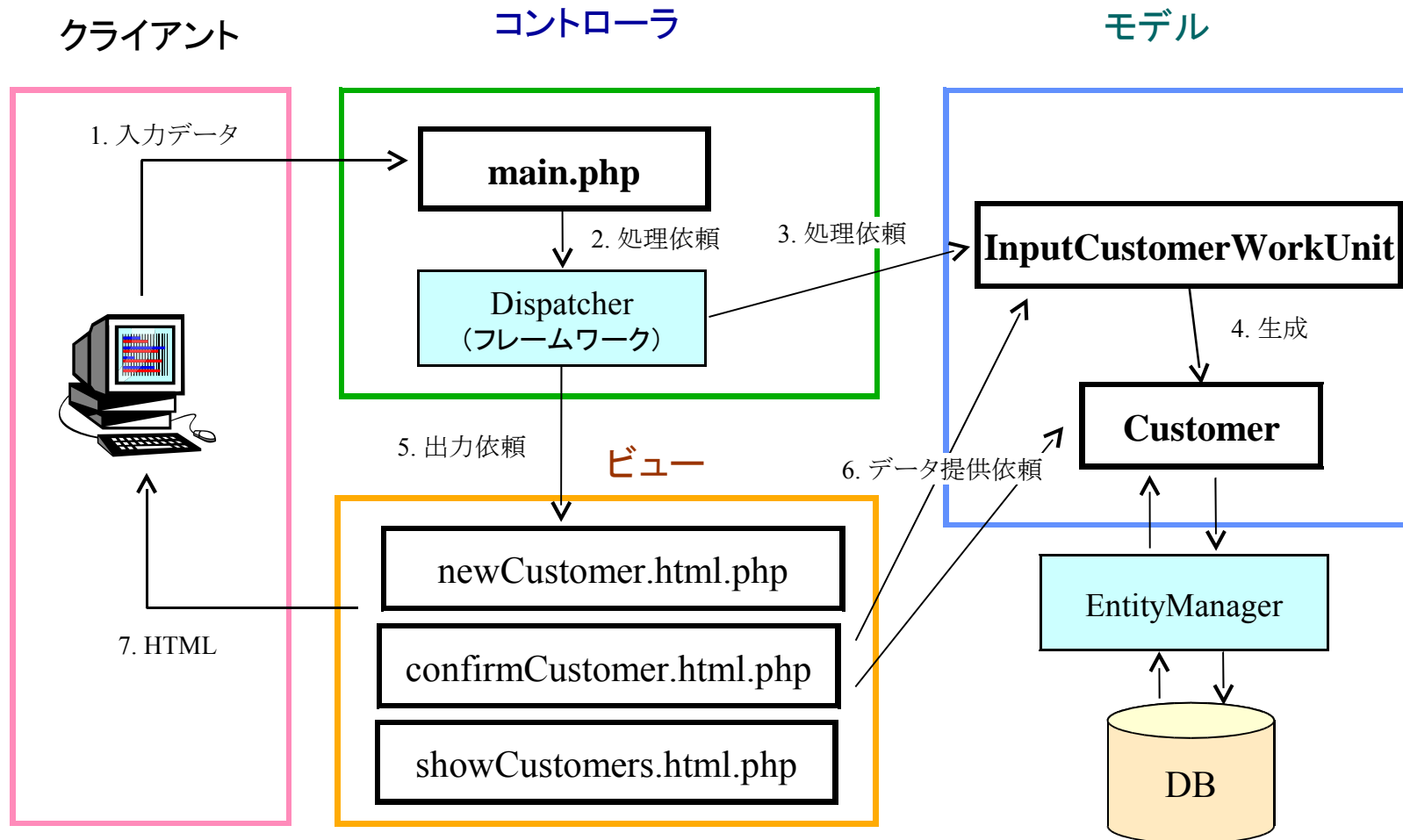
/ receiveInput()

この内容で保存

/ fixInput()

戻る

# ミニサンプルの構成



# エンティティ

## エンティティ

永続オブジェクト

EJBのEntityBeanに相当

## O/Rマッピング情報

メソッドの戻り値で与える

keyName()

tableName()

initialKeyValue()

## 生成/更新/削除

EntityManagerのサービス

EJB3と同様

## EntityManagerの取得

グローバル関数を利用

getEntityManager()

```
<?php
class Customer {
    public $id;
    public $name;
    public $address;

    function keyName() { return "id"; } //主キー属性
    function tableName() { return "Customer"; } //テーブル名
    function initialKeyValue() { return "C0000"; } //自動付番初期値

    function init($name, $address) {
        $this->name = $name;
        $this->address = $address;
        getEntityManager()->save($this); //DBに保存を依頼
    }

    function changeAddress($newAddress) {
        $this->address = $newAddress;
        getEntityManager()->update($this); //DBに更新を依頼
    }

    function delete() {
        getEntityManager()->delete($this); //DBに削除を依頼
    }
}
?>
```

# ワークユニット

## ワークユニット

セッションファサード

EJBのStateful SessionBeanに相当

- セッションデータをキャッシュ
- エンティティに処理を振り分ける

ワークユニットの生成/再現/削除はアプリケーション設定をもとにフレームワークが自動的に行う。

```
<?php
class InputCustomerWorkUnit {
    public $name;
    public $address;
    // アクションメソッド
    function receiveInput() {
        // 入力データをセッションデータとしてキャッシュ
        $this->name = util::getRequest('name');
        $this->address = util::getRequest('address');
    }
    function fixInput() {
        // キャッシュデータを使ってオブジェクトを生成
        $customer = new Customer();
        $customer->init($this->name, $this->address);
        // 出力用に最後に生成されたオブジェクトをキャッシュ
        $this->lastCustomer = $customer;
    }
    function getAllCustomers() {
        return getEntityManager()->findAll('Customer');
    }
}
?>
```

# ビュー

## ビュー

出力用HTMLの生成  
Sabaphyでは通常のPHPファイル

プログラムは、ワークユニットやエンティティの参照によるデータ取得のみとし、ビジネスロジックは含ませない。

```
<?
$workunit = getWorkUnitManager()->
                restore('InputCustomerWorkUnit');
?>

<html>
<head><title>入力結果</title></head>
<body>
名前=<?= util::toWebText($workunit->name) ?><br>
住所=<?= util::toWebText($workunit->address) ?><br>
<form method="POST">
    <input type="submit" name="_COMMAND"
        value="この内容で保存">
    <input type="submit" name="_COMMAND" value="戻る">
</form>
</body>
</html>
```

# メインPHP (フレームワーク未使用)

## メインPHP

クライアントから呼び出されるコントローラプログラム

(1)使用するプログラムファイルの指定

(2)初期設定

使用するDBの情報を与えてDIコンテナを初期設定する。

(3)処理の振り分け

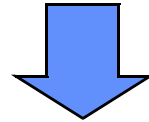
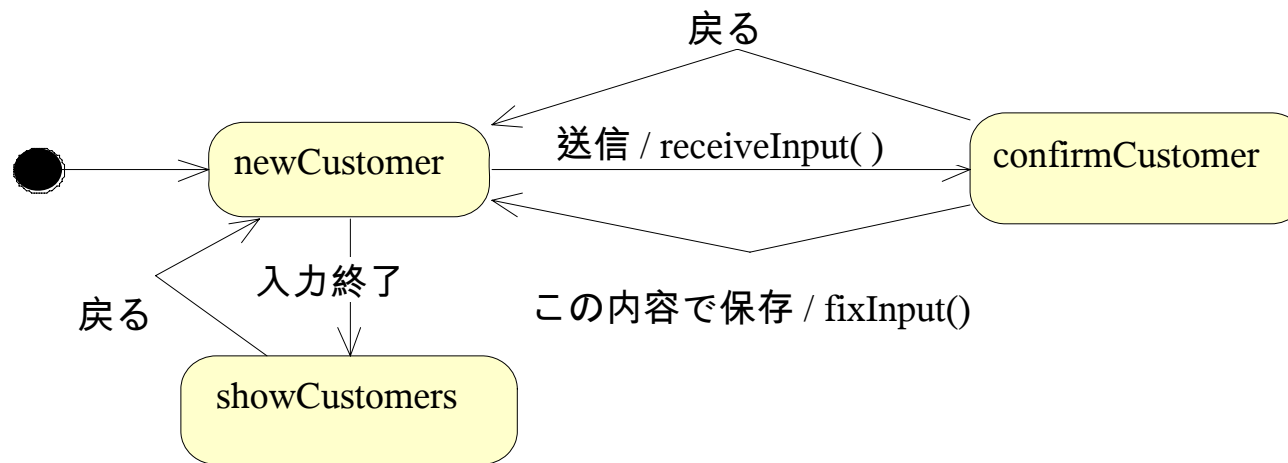
要求されたコマンド(押されたボタンの名称)に応じた処理を記述する。

必要に応じて、ワークユニットを生成/再現し、処理を委譲する。

正常終了すれば、DBにコミットして、変更を確定する。

```
<?php
// (1) 使用するプログラムファイルの指定
ini_set('include_path', './../..');
require_once 'sabaphy/SabaContainer.php';
require_once 'InputCustomerWorkUnit.class.php';
require_once 'Customer.class.php';
// (2) 初期設定
session_start();
$cnt = new SabaContainer();
$cnt->setup(array('dbConnection'=>'SdbConnection',
                'file'=>'db/database'));
// (3) 処理の振り分け
$cmd = null;
if (isset($_REQUEST['_COMMAND'])) $cmd =
    $_REQUEST['_COMMAND'];
if (($cmd == null) || ($cmd == '戻る')) {
    require 'newCustomer.html.php';
} elseif ($cmd == '送信') {
    $workunit = new InputCustomerWorkUnit();
    $workunit->receiveInput();
    getWorkUnitManager()->save($workunit);
    require 'confirmCustomer.html.php';
} elseif ($cmd == 'この内容で保存') {
    $workunit = getWorkUnitManager()-
>restore('InputCustomerWorkUnit');
    $workunit->fixInput();
    require 'newCustomer.html.php';
    getEntityManager()->commit();
    getWorkUnitManager()->delete($workunit);
} elseif ($cmd == '入力終了') {
    require 'showCustomers.html.php';
}
?>
```

# 画面遷移図とコマンド情報



コマンド名	ワークユニット				ビュー
	クラス名	前処理	アクション	後処理	出力生成PHP
null	-	-	-	-	newCustomer.html.php
送信	InputCustomerWorkUnit	new	receiveInput()	-	confirmCustomer.html.php
この内容で保存	InputCustomerWorkUnit	restore	fixInput()	close	newCustomer.html.php
戻る	-	-	-	-	newCustomer.html.php
入力終了	-	-	-	-	newCustomer.html.php

# メインPHP (フレームワークを使用)

## Sabaphyのフレームワーク

Dispatcherにコマンド情報を与えて、実行を指示する。

コマンド情報は、PHPの連想配列を使用して与える。

```
array(key1=>value1,  
      key2=>value2, ....)
```

```
<?php  
// (1) 使用するプログラムファイルの指定  
ini_set('include_path', './:../');  
require_once 'sabaphy/SabaContainer.php';  
require_once 'InputCustomerWorkUnit.class.php';  
require_once 'Customer.class.php';  
  
// (2) 初期設定  
session_start();  
$cnt = new SabaContainer();  
$cnt->setup(array('dbConnection'=>'SdbConnection', 'file'=>'db/database'));  
  
// (3) ディスパッチャの生成  
$d = $cnt->getDispatcher();  
  
// (4) コマンド情報の設定  
$d->addCommand('null', array('view'=>'NewCustomer.html.php'));  
$d->addCommand('送信', array(  
    'workunitClass'=>'InputCustomerWorkUnit', 'workunit'=>'new',  
    'action'=>'receiveInput', 'view'=>'confirmCustomer.html.php'));  
$d->addCommand('この内容で保存', array(  
    'workunitClass'=>'InputCustomerWorkUnit', 'workunit'=>'restore/close',  
    'action'=>'fixInput', 'forward'=>'null'));  
$d->addCommand('戻る', array('forward'=>'null'));  
$d->addCommand('入力終了', array('view'=>'showCustomers.html.php'));  
  
// (5) コマンドの実行  
$d->execute();  
?>
```



# インターネット書店サンプル

## SabaShop

(1) 検索指示画面

Saba Books ホームページ

### 書籍の検索

検索文字列

指定した文字をタイトルに含む本が検索されます。  
英字も全角で入れて下さい。

検索を行わず

「Perl」と入力して「検索」ボタンを押す

(2) 検索結果画面

### 検索結果

書籍番号	ISBN	タイトル	出版社	著者	価格	選択
B0000011	4797316349	実践Perl DBI	ソフトバンクパブリッシング	太田俊哉	2600	<input type="checkbox"/>
B0000012	4873110963	プログラミングPerl volume 1	オライリー・ジャパン	ラリー・ウォール	5300	<input checked="" type="checkbox"/>
B0000013	4894715309	MySQL & Perl Webアプリケーション開発	ピアソンエデュケーション	ポール・デュボワ	4200	<input type="checkbox"/>
B0000014	4873110971	プログラミングPerl volume 2	オライリー・ジャパン	ラリー・ウォール	4700	<input checked="" type="checkbox"/>
B0000015	4894713004	オブジェクト指向Perlマスターコース	ピアソンエデュケーション	ダミアン・コンウェイ	6200	<input type="checkbox"/>

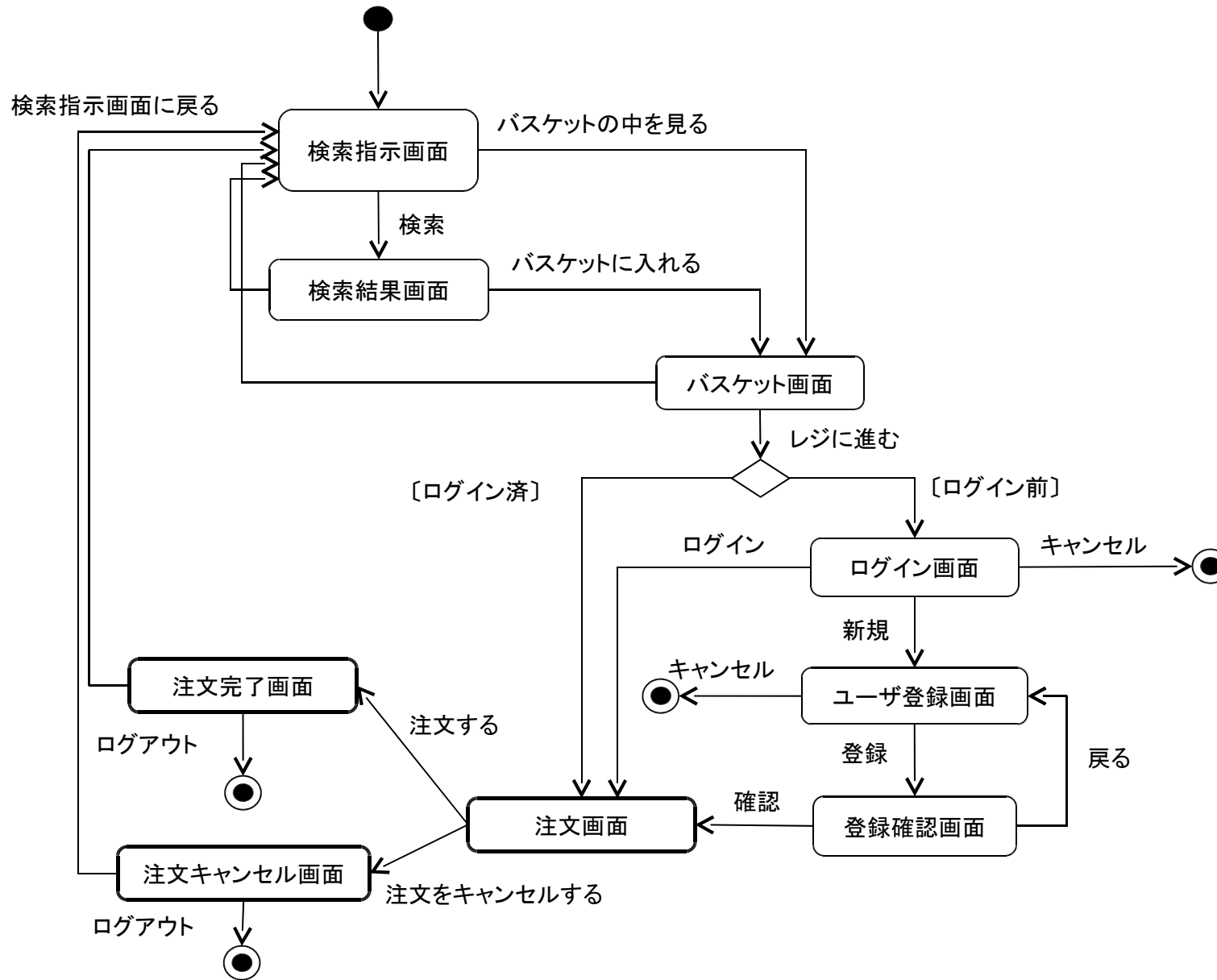
(3) Basket画面

### Basketの内容

書籍番号	ISBN	タイトル	出版社	著者	価格	数量	選択
B0000012	4873110963	プログラミングPerl volume 1	オライリー・ジャパン	ラリー・ウォール	5300	1	<input checked="" type="checkbox"/>
B0000014	4873110971	プログラミングPerl volume 2	オライリー・ジャパン	ラリー・ウォール	4700	1	<input checked="" type="checkbox"/>

2件チェックして、「Basketに入れる」ボタンを押す

# SabaShopの画面遷移図



# SabaShopのコマンド情報

	受取コマンド	ユーザ クラス	アクション	ビュー	送出コマンド
検索・注文	1. null		OrderWorkUnit::new	search.html	(検索) (バスケットの中を見る)
	2. 検索		OrderWorkUnit::restore doSearch()	foundBooks.html	(バスケットに入れる) (検索指示画面に戻る)
	3. バスケットの中を見る		-	-	バスケットに入れる
	4. バスケットに入れる		OrderWorkUnit::restore addSelection()	basket.html	(レジに進む) (basket-検索指示画面に戻る)
	5. 検索指示画面に戻る		OrderWorkUnit:new   restore	search.html	(検索) (バスケットの中を見る)
	6. basket-検索指示画面に戻る		OrderWorkUnit::restore changeSelection()	-	検索指示画面に戻る
	7. レジに進む		OrderWorkUnit::restore checkLogin()	-	showLoginPage showOrderPage
	8. showOrderPage	Customer	OrderWorkUnit::restore prepareOrder()	order.html	(注文する) (注文をキャンセルする)
	9. 注文する	Customer	OrderWorkUnit::restore/close fixOrder()	orderCompleted.html	(ログアウト) (検索指示画面に戻る)
	10. 注文をキャンセルする	Customer	OrderWorkUnit::restore/close cancelOrder()	cancelCompleted.html	(ログアウト) (検索指示画面に戻る)
ログイン・ログアウト	11. showLoginPage		-	login.html	(ログイン) (キャンセル) (新規)
	12. ログイン		LoginWorkUnit::temp doLogin()	-	showOrderPage
	13. キャンセル		-	-	null
	14. 新規		-	inputCustomer.html	(登録) (キャンセル)
	15. 登録		NewCustomerWorkUnit::new doInput()	confirmCustomer.html	(確認) [戻る]
	16. 確認		NewCustomerWorkUnit::restore/close doSave()	-	showOrderPage
	17. ログアウト		LoginWorkUnit::temp doLogout()	-	null

# 簡略化している機能

- 分散オブジェクトは未サポート
- データベーステーブルの制限
  - 主キーは単純キーのみ(複合キーは未サポート)
  - 列名はエンティティのプロパティ名と同じ
  - 一度に使用できるDB接続は1本のみ
  - 排他制御だけの単純化したトランザクション
  - コネクションプール、二相コミットは未サポート
- 簡易セッション管理機能
  - PHP標準のセッション管理機能を利用
- 入力データ
  - PHP標準のリクエストデータをそのまま使用
- その他
  - 高度なHTMLテンプレート機能
    - Smarty (PHPベースのHTMLテンプレートエンジン)を組み込むことで可能と思われる。
  - AOP(アスペクト指向)機能
    - コマンドチェーンの拡張で、類似のことは可能。

既存DBの再利用や  
アクセスが集中するサ  
イトには不向き

# Sabaphyを用いた段階的学習

## 1. Webアプリケーション入門

- 標準PHPのみ使用

## 2. オブジェクト永続化とセッション管理

- EntityManager, WorkUnitManagerを使用

## 3. MVCパターン

- エンティティ、ワークユニットの役割の明確化
- プログラム構成の変更

## 4. フレームワーク

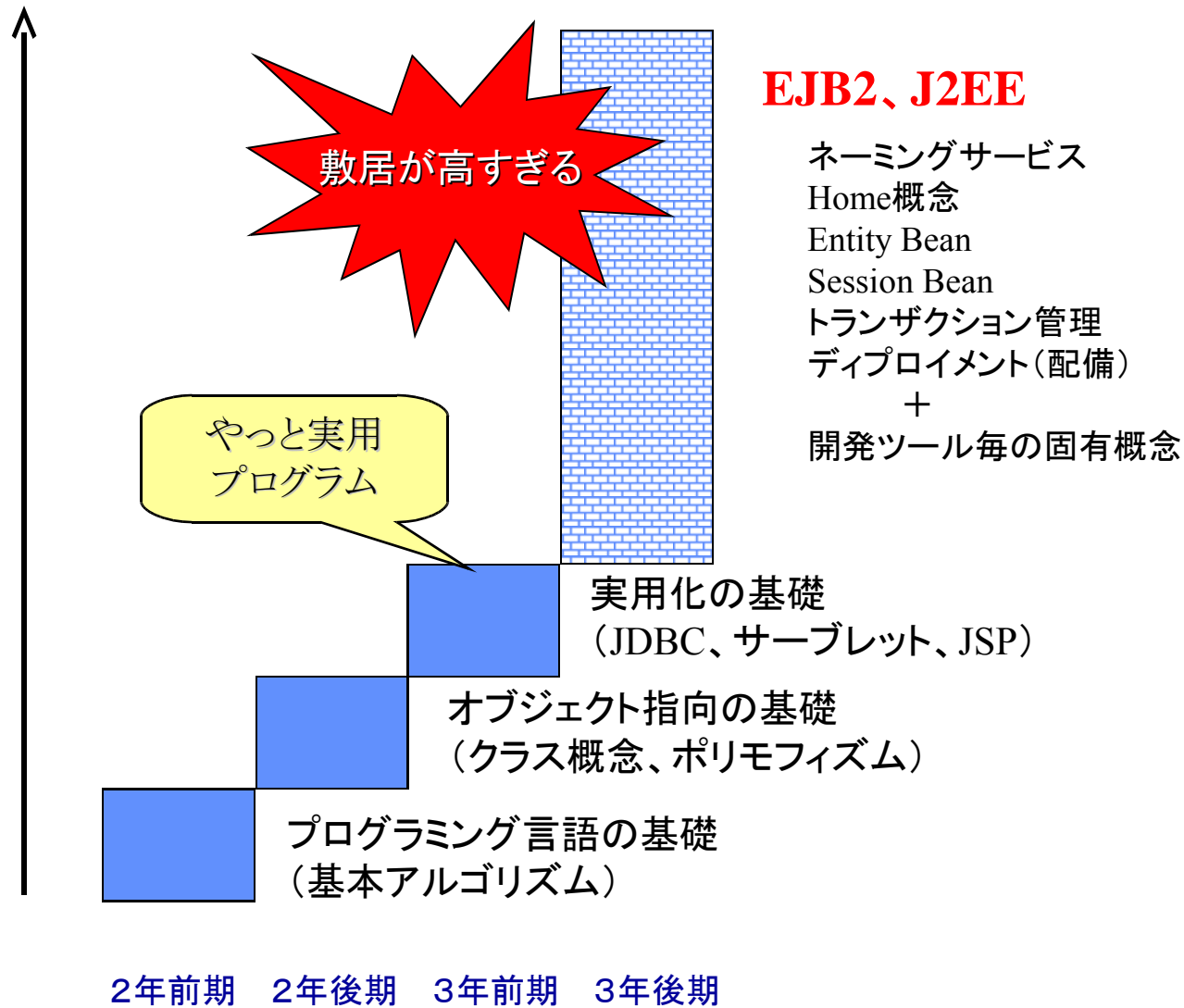
- Dispatcherを使用
- 自作コントローラの置き換え

## 5. 個別機能の高度化

- リレーショナル・データベースの利用(SQLを使った検索)
- ユーザ認証(LoginUserManagerを使用)
- エラー報告ページのカスタマイズ(コンポーネントの入れ替え)

# Sabaphy以前のWebアプリ教育

難しさ



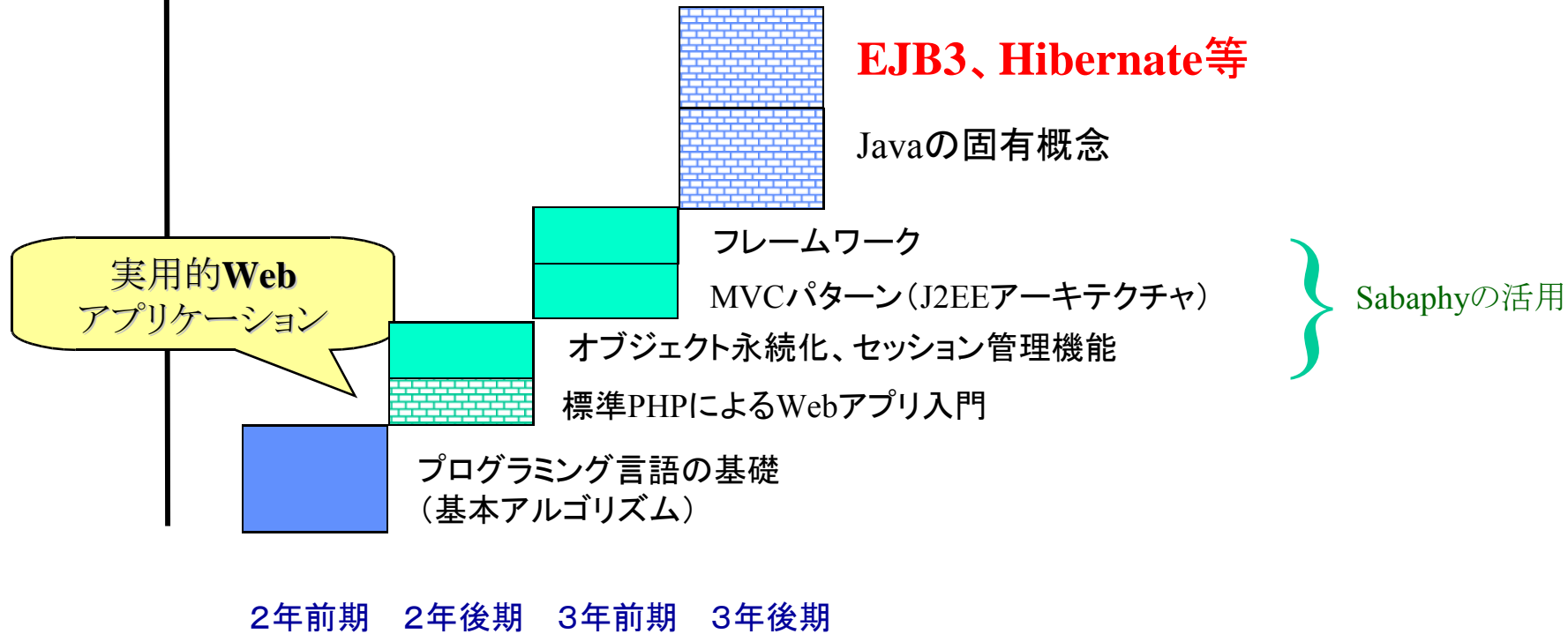
# Sabaphyを用いたWebアプリ教育

難しさ

PHP5 ⇒ 入門レベルでも実用的なものが作れる  
 Sabaphy ⇒ 入門レベルから段階的に高度な概念に進むことができる



裾野を広げることができた



# まとめ

## Webアプリケーションは今日のビジネス情報システムの中心的存在

### 2. それを教育するための体系や教材が不足

- Sabaphyとそれを用いた教材は、この間隙を埋めるためのもの
- Sabaphyでの簡略化が適切か否か、議論が必要

### 3. 「より簡単に（Ease of Development）、よりはやく」というのは時代の潮流

- スクリプト言語(PHP5) + 簡易コンテナというSabaphyは、この流れに沿った1つの解の提案
- その妥当性を実務家の方々に評価していただけると、ありがたい。
- また、よりよいやり方があれば、積極的に取り入れていきたい。



# Sabaphyに関する補足



# オブジェクト永続化サービス

- **EntityManagerの機能**
  - オブジェクトの永続化(新規保存、更新、削除)
  - 異なるタイプのDBの透過的利用
  - 条件検索を除き、SQLの知識は不要
- **利用できるDB**
  - シリアルDBとRDB(MySQL, SQLite)
  - いずれも簡易トランザクション機能が利用可能
- **付帯機能**
  - キー値自動付番
  - 参照オブジェクトの格納
  - 複数クラスデータが混在するテーブルの利用
    - サブクラスのインスタンスの混在が可能
  - 読み込みテーブルと書き込みテーブルの区別
    - ビューテーブルの活用が可能

# セッション管理サービス

- **WorkUnitManagerの機能**
  - セッション継続中のワークユニットの再現
  - 役目を終えたワークユニットの削除
  - ログアウト時のワークユニットの一括削除
- **LoguinUserManagerの機能**
  - ユーザ認証
    - フレームワーク利用時のユーザ認証の自動化
  - 多重継承可能なユーザ権限階層
    - 多重継承が使えるインタフェースを利用したユーザ管理
  - ログアウト時のユーザ情報の一括削除

# オブジェクト検索機能

- **主キー値による検索** (C0003の顧客を検索する)

```
$customer = getEntityManager()->find('Customer', 'C0003');
```

- **条件検索** (20歳以上の男を検索する)

```
$customers = getEntityManager()->  
findByQuery('Person',  
    '(?object->age >= ?1) && (?object->gender == "?2")',  
    // シリアルDB向け検索条件  
    'select * from Person where age >= ?1 and gender = "?2"',  
    // リレーショナルDB向け検索条件(SQL文)  
    array(20, 'M')  
    // パラメータ(?1, ?2)の値  
);
```

- **Fetchを使った条件検索** (20歳以上の男を検索する)

```
getEntityManager()-> prepareFetchByQuery('Person', ...上と同様……);  
while ($person = getEntityManager()->getNextByFetch()) {  
    <$personの処理>  
}
```

# 情報提供サイト

**Sabaphy on PHP**

SabaphyはDコンテナを使うV0.2になりました。  
通常の利用では、初期設定を除き、V0.1と同じです。  
旧版のサービスロケータ型はV0.1をご覧ください。

V0.2.0:	2005.7.11
V0.2.6:	2005.8.29

- Sabaphyとは**
  - Webアプリケーション学習用軽量コンテナ PHP5版
  - [そのねらい...](#)
  - [技術的な特徴...](#)
  - [サンプルでみるSabaphyの概要](#) (プログラムのダウンロード)
- ライブラリ**
  - Sabaphyライブラリ (0.2.6 2005/8/29 [ライブラリのダウンロード](#))
  - [Sabaphyユーザマニュアル](#) (0.2.6 2005/8/29 [ダウンロード](#))
- 学習用教材**
  - 準備**
    - [PHP利用の準備](#)
  - PHPの基礎**
    - WebApp1: [簡単なPHP Webアプリケーション](#) (プログラムのダウンロード)
  - Sabaphyの利用**
    - WebApp2: [永続化とセッション管理](#) (プログラムのダウンロード)
    - WebApp3: [MVCパターン](#) (プログラムのダウンロード)
    - WebApp4: [フレームワーク](#) (プログラムのダウンロード)
- サンプル**
  - SabaShop**
    - [インターネットショッピングサンプル](#) (プログラムのダウンロード)
    - [デモの起動](#)
- Sabaphy Q&A** ([Q&Aページへ](#))
- PHPのデバッガ** ([PHP Simple Debuggerの紹介](#))
- 連絡先**

東京国際大学商学部情報システム学科  
佐藤 英人 ([sato@tiu.ac.jp](mailto:sato@tiu.ac.jp))

<http://satolab.tiu.ac.jp/SabaphyHome/>

1. Sabaphyの概要
2. ユーザマニュアル
3. 学習用教材
4. サンプル (SabaShop)
5. Q&A

等の情報を提供 (ソースコード付)

# マニュアル

The screenshot shows a web browser window titled "Sabaphy 0.2.6 ユーザマニュアル - Microsoft Internet Explorer". The address bar shows the URL "http://satolab.tiu.ac.jp/SabaphyHome/sabaphy.doc/index.html". The page content includes a navigation menu on the left, a main heading "Sabaphy 0.2.6", a subtitle "Webアプリケーション学習用コンテナ", and a "Table of Contents" section with a list of links. The browser's status bar at the bottom shows "インターネット".

**sabaphy**

sabaphy | sabaphy\_ds | sabaphy\_exceptions | userclass

**sabaphy**

Description  
Class trees  
Index of elements  
Tutorials/Manuals  
Package-level  
Sabaphy 0.2.6  
データベースの利用  
WindowsからUnixへ

Classes  
BaseContainer  
SabaContainer

Functions  
getContainer  
getEntityManager  
getWorkUnitManager

dom  
Classes  
AutoNumBase  
EntityManager  
WorkUnitManager

fw  
Classes  
CommandHandler  
CommandInfo  
Dispatcher

util  
Classes  
ExceptionHandler  
exceptionX  
LoginUserManager  
util  
Functions  
sabaphyDefaultErrorHa

phpDocumentor v  
1.3.0RC3

Previous Next  
データベースの利用

## Sabaphy 0.2.6

### Webアプリケーション学習用コンテナ

佐藤 英人 (東京国際大学)  
Copyright 2005, Hideto Sato

## Table of Contents

- はじめに
- Sabaphyアプリケーションの構成
- Sabaphyの仕組み
- Sabaphyのパッケージ構成
- このマニュアルの使い方
- 用語
- バージョン情報

## はじめに

SabaphyはWebアプリケーション学習用の軽量コンテナで、このドキュメントはその利用者向けマニュアルである。利用者に直接関係ない内部的な機能の解説は省略している。

## Sabaphyアプリケーションの構成

<http://satolab.tiu.ac.jp/SabaphyHome/sabaphy.doc/index.html>