Seasar 2.5

ブルーオーシャン戦略とは

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

ブルーオーシャン戦略とは

- ブルーオーシャン
  - 競争の無い未開拓市場
    - Wii
- レッドオーシャン
  - 血みどろの戦いが繰り広げられる既存の市場
    - これまでのゲーム機

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

レッドオーシャン

- 特徴
  - 同じ市場でライバルに差別化しようとする。
  - パイの奪い合い。
  - すぐに追いつかれるので差別化できない。

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

レッドオーシャン

- Java EE(EJB3, JPA, JSF)
  - 同一の仕様で実装を競い合う。
    - 差別化は難しい。
  - 機能の肥大化
    - あらゆる状況に対応しようとして仕様が膨らむ。
    - 仕様が膨大なので学習が大変。
    - 複雑な仕様を実装する必要があるのでパフォーマンスが出ない。
  - ゲーム機と同じ間違い

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

ブルーオーシャン

- ブルーオーシャンを開拓するには
  - Wiiに学ぶ
    - 複雑になりすぎたゲームではなく、
    - もっと簡単で覚えやすいゲームを作れないか。
  - Seasar2
    - 複雑になりすぎたJava EEではなく、
    - もっと簡単で覚えやすいフレームワークを作れないか。

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

Without Java EE

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

ブルーオーシャン戦略

- Seasar2のブルーオーシャン戦略
  - JavaEEの機能を削る。
    - 削る勇気を持つ。
  - Seasarのコミュニティで育てた機能をSeasar2本体に取り込む。

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

ブルーオーシャン戦略

- S2Persistence
  - S2Dao + DBFluteの仕様をベース
- S2Presentation
  - Teeda + S2JSF without JSF

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

Seasar2.5の構成

- S2Framework
  - org.seasar.framework
- S2Extension
  - org.seasar.extension
- S2Persistence
  - org.seasar.persistence
- S2Presentation
  - org.seasar.presentation

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

ブルーオーシャン戦略の三つの柱

- 大胆にソースコードを削る
- 過去の習慣と常識にとらわれない
- 迷いをなくす

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

ブルーオーシャン戦略

- publicフィールド対応
  - setter, getterメソッド不要。
  - フレームワークと自分自身しか触らないフィールドはpublicでかまわない。
  - DIのために必要なフィールドはpublicでかまわない。
  - テーブルのカラムと一対一で結びついているフィールドはpublicでかまわない。
    - 表示のために加工が必要ならConverterで対応する。

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

ブルーオーシャン戦略

- インターフェースは基本不要
  - 1インターフェース1実装なら不要。
  - モックは継承して作ればよい。

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

ブルーオーシャン戦略

- 必要最小限の登場人物
  - Page
    - HTMLと一対一
  - Service
    - ユースケースと一対一
  - Entity
    - テーブルと一対一

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

ブルーオーシャン戦略

- ユースケースの粒度
  - 基本はサブアプリケーションごと
  - 複数のサブアプリケーションから共通的に使われる機能は共通のユースケースとして抽出する。

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

方針が明確で誰がやっても迷わない

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

employeeList.html

```
<table>
<tbody p:items="#{employeeItems}">
<span id="hiredate"
  p:value="#{employee.hiredate}"/>
<p:dateTimeConverter
  format="yyyy/MM/dd" for="hiredate"/>
</tbody>
</table>
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

EmployeeListPage.java

```
public EmployeeService employeeService;
public List<Employee> employeeItems;
public Employee employee;

public void prerender() {
  employeeItems = employeeService.findAll();
}
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

EmployeeService.java

```
public PersistenceManager pm;

public List<Employee> findAll() {
  return pm.findAll(Employee.class);
}
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

Employee.java

```
public Long id;
@Required @Length(30)
public String empName;
public Timestamp hiredate;
public Integer version;
public Department department;
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

Seasar2.5の開発の進め方

- MLでオープンに仕様を話し合う。
  - Seasar Specification Request(SSR-xxxxx)
  - 誰でも仕様作成に参加できる。
- 実装 & ドキュメント & βリリース
  - 仕様が決まったものから
  - 優先順位に応じて
  - 実装 & ドキュメント & βリリース

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

Seasar2.5の開発の進め方

- RCリリース
  - すべてのSSRがリリースされたらRCのリリース
- 正式リリース
  - タイミングを見て

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

Seasar2.5の開発環境

- Maven2が基本
  - プロジェクトの生成。
  - テーブルからエンティティの生成。
  - エンティティからテーブルを更新。
  - SQLファイルからDTOを作成。
  - Scaffold。
- Maven2の敷居の高さはSeasar2が吸収
  - Eclipseからキックできるようにする。
  - Eclipseのプラグインとの組み合わせも含めて提供する。

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

これまでのプロダクトはどうなるのか

- Teeda
  - 1.1でSeasar2.5に対応。
  - S2Presentationのかわりに使うこともできる。
- Kuina-Dao
  - 1.1でSeasar2.5に対応。
  - S2Persistenceのかわりに使うこともできる。
- Dolteng
  - Teeda Plugin, S2Flex2 Pluginに分解。

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

新規Eclipse Plugin

- Super Agile Plugin
  - Scaffold
  - サブアプリケーションの追加。
    - HTMLの格納ディレクトリ。
    - Pageの格納パッケージ。
    - Serviceの雛形。

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

新規Eclipse Plugin

- S2Prensentation Plugin
  - HTMLの雛形生成。
  - HTMLからPageを自動生成。
  - HTMLとPageの相互移動。
  - HTMLの自動補完とバリデーション
    - ValueBinding
      #{}
    - Validator
      <p:lengthValidator maximum="5"/>
    - Converter
      <p:dateTimeConverter pattern="yyyy/MM/dd"/>

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

新規Eclipse Plugin

- S2Persistence Plugin
  - テーブルからエンティティの自動生成。
  - エンティティからテーブルの更新。
  - SQLファイルからDTOを作成。

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

S2Persistence(案)

- Dao不要
  - 高水準なPersistenceManager API
- ネストしたManyToOne, OneToOneサポート
- OneToManyサポート
- RDBMSを生かすPaging処理
- パフォーマンスの向上
  - 完全なHOT deploy対応。
  - データベースのメタデータを使わずデフォルトのルールとアノテーションを使う。
  - PreparedStatementをキャッシュする。

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- キーで検索

```
Employee emp = pm.find(Employee.class, 1);
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- =検索

```
Employee emp = new Employee();
emp.job = JobType.MANAGER;
Department dept = new Department();
dept.departmentName = “RESEARCH”;
emp.department = dept;
List<Employee> employees =
  pm.findAll(Employee.class, emp);
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- 自動生成されるSQL

```
select … from employee e join department d
on e.department_id = d.id
where e.job = ‘MANAGER’
and d.departmentName = ‘RESEARCH’
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- >=,<=検索

```
EmployeeCriteria criteria =
    new EmployeeCriteria();
criteria.salary_GE = 1000;
criteria.salary_LE = 3000;
List<Employee> employees =
    pm.findAll(Employee.class, criteria);
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- 自動生成されるSQL

```
select … from employee
where salary >= 1000
and salary <= 3000
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- IS NOT NULLやLIKE検索

```
EmployeeCriteria criteria =
    new EmployeeCriteria();
criteria.commission_IS_NOT_NULL = true;
DepartmentCriteria dcriteria =
    new DepartmentCriteria();
dcriteria.departmentName_STARTS = “E”;
criteria.departmentCriteria = dcriteria;
List<Employee> employees =
    pm.findAll(Employee.class, criteria);
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- 自動生成されるSQL

```
select … from employee e join department d
on e.department_id = d.id
where e.commission is not null
and d.departmentName like ‘E%’
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- Paging検索

```
EmployeeCriteria criteria =
    new EmployeeCriteria();
criteria.offset = 100;
criteria.limit = 10;
List<Employee> employees =
    pm.findAll(Employee.class, criteria);
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- 自動生成されるSQL

  select … from employee

  offset 100 limit 10

  #PostgreSQLの場合

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- FETCH JOIN(ManyToOne)検索

```
EmployeeCriteria criteria =
    new EmployeeCriteria();
criteria.departmentFetchJoinType =
    FetchJoinType.Outer;
List<Employee> employees =
    pm.findAll(Employee.class, criteria);
for (Employee e : employees) {
    System.out.println(e.department);
}
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- 自動生成されるSQL

```
select e.*, d.* from employee e left outer join
  department d on e.department_id = d.id
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- FETCH JOIN(OneToMany)検索

```
DeparmentCriteria criteria =
  new DepartmentCriteria();
criteria.employeesFetchJoinType =
  FetchJoinType.INNER;
Department department =
  pm.find(Deparment.class, criteria);
for (Employee e : d.employees) {
  System.out.println(e);
}
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- 自動生成されるSQL

```
select * from department;
select * from employee
  where department_id in (…);
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- NamedQuery

  examples/sql/aaa.sql

  select …where …hoge = /*hoge*/1

```
List<EmployeeDto> dtoList =
    pm.getNamedQuery("examples/sql/aaa.sql")
    .setParameter("hoge", 2)
    .getResultList(EmployeeDto.class);
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- DynamicQuery

```
StrinbBuilder sb = new StringBuilder();
…
List<EmployeeDto> list =
   pm.createQuery(sb.toString())
   .setParameter(“hoge”, 2)
   .getResultList(EmployeeDto.class);
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- Query As Map

```
StrinbBuilder sb = new StringBuilder();
…
List<Map> list =
   pm.createQuery(sb.toString())
   .setParameter(“hoge”, 2)
   .getResultList(Map.class);
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- INSERT

```
Employee e = new Employee();
e.employeeName = “SCOTT”;
pm.insert(e);
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- 配列INSERT

```
List<Employee> employees = new
   ArrayList<Employee>();
…
pm.insert(employees);
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- Bulk INSERT

  examples/sql/bbb.sql

  insert into … select … where hoge = /*hoge*/1

  pm.getNamedQuery(“exmples/sql/bbb.sql”)
  .setParameter(“hoge”, 2)
  .executeUpdate();

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- UPDATE

```
Employee e = pm.find(Employee.class, 1);
e.employeeName = “SCOTT”;
pm.update(e);
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- 配列UPDATE

```
List<Employee> employees =
    pm.findAll(Employee.class);
...
pm.update(employees);
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- Bulk UPDATE

  examples/sql/bbb.sql

  update set … where hoge = /*hoge*/1

  ```
  pm.getNamedQuery(“exmples/sql/bbb.sql”)
    .setParameter(“hoge”, 2)
    .executeUpdate();
  ```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- DELETE

```
Employee e = pm.find(Employee.class, 1);
pm.delete(e);
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- 配列DELETE

```
List<Employee> employees =
    pm.findAll(Employee.class);
…
pm.delete(employees);
```


Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.

PersistenceManager API(案)

- Bulk DELETE

  examples/sql/bbb.sql

  delete from … where hoge = /*hoge*/1

  pm.getNamedQuery(“exmples/sql/bbb.sql”)
  .setParameter(“hoge”, 2)
  .executeUpdate();

Copyright© 2004-2007 The Seasar Foundation and the others. All rights reserved.