

Donbiki勉強会

lick Framework

EY-Office 吉田

Clickの一般情報

- URL <http://click.sourceforge.net/>
- メイン開発者 **Malcolm Edgar**
- 歴史
 - Version 0.1 – 2003年
 - Version 0.3 – 2005年3月 初公開
 - Version 1.0 – 2006年10月
- [Avoka Technologies](#) で商用サポート

Clickのコンセプト

- シンプルで学習が容易
- コンポーネント指向、ページ指向
- イベント・モデル
- JSPやMVC(POJO)にこだわらない
- Velocity(JSP)によるページ生成
- Fromの自動生成、Client/Serverバリデーション
- 高性能

ページ指向

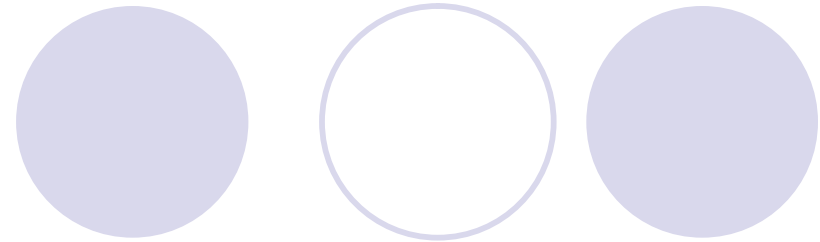


- ページ毎に Page を継承したページクラスを作成する。
- テンプレート名/URIはページクラス名から決まる。XxxYyy.java → xxx-yyy.html (カスタイズ可)
- ページクラスのpublicなオブジェクト(コンポーネント)はテンプレートから参照可能。

コンポーネント指向/イベント・モデル

- タグに対応したコンポーネントをページクラスに(Javaコード)で定義する。
- コンポーネント(例 ボタン)にはイベントリスナーを定義でき、ボタンがクリックされた場合定義したリスナーが呼び出される。
- また、ページクラスに定義された `onInit()`, `onGet()` などのイベントメソッドもある。

サンプルコード (1)



画面

Member

Name*

Email

Add

Name	Email
Yuumi Yoshida	yy@ey-office.com
Tarao Yamada	taro@yamada.com
Jiro Yamada	jiro@yamada.com

[先頭/前] 1 [次/最後]

テンプレート

```
<html>
  <head>
    <title>Click Sample1</title>
    $imports
    #if ($headInclude) #parse($headInclude) #end
  </head>
  <body>
    $form
    $table
  </body>
</html>
```

サンプルコード(2)

```
public class Sample1 extends Page {
    public Form form = new Form();
    public Table table = new Table();
    private TextField name = new TextField("name");
    private EmailField email = new EmailField("email");

    public Sample1() {
        form.setColumns(1);
        FieldSet fieldSet = new FieldSet("fieldSet", "Member");
        form.add(fieldSet);
        name.setRequired(true);
        fieldSet.add(name);
        fieldSet.add(email);
        form.add(new Submit("Add", this, "onAddClick"));

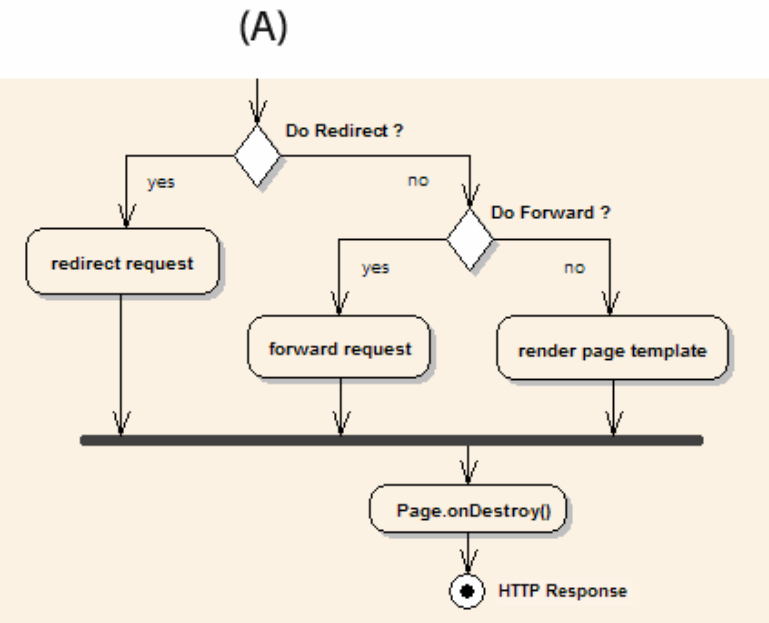
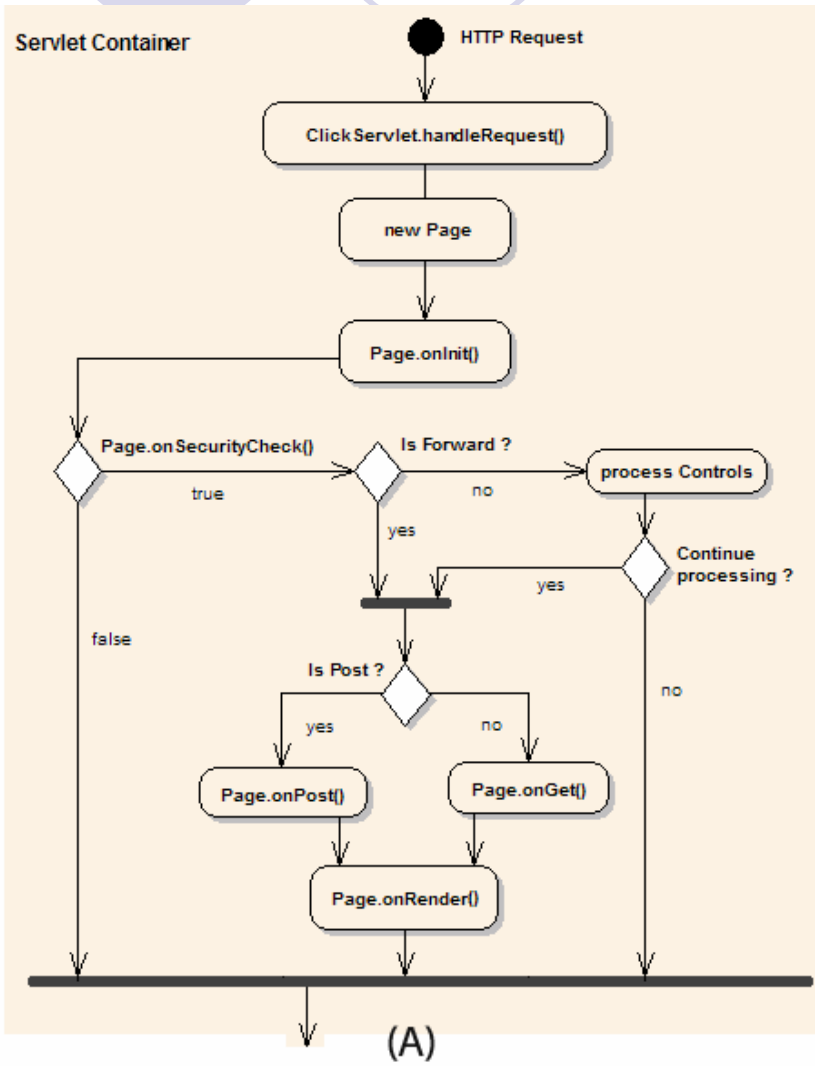
        table.setClass("its");
        table.setPageSize(10);
        Column column = new Column("name");
        table.addColumn(column);
        column = new Column("email");
        column.setAutolink(true);
        table.addColumn(column);
    }
}
```

2007/1/31

```
public boolean onAddClick() {
    if (form.isValid()) {
        addMember(name.getValue(),
                  email.getValue());
    }
    return true;
}

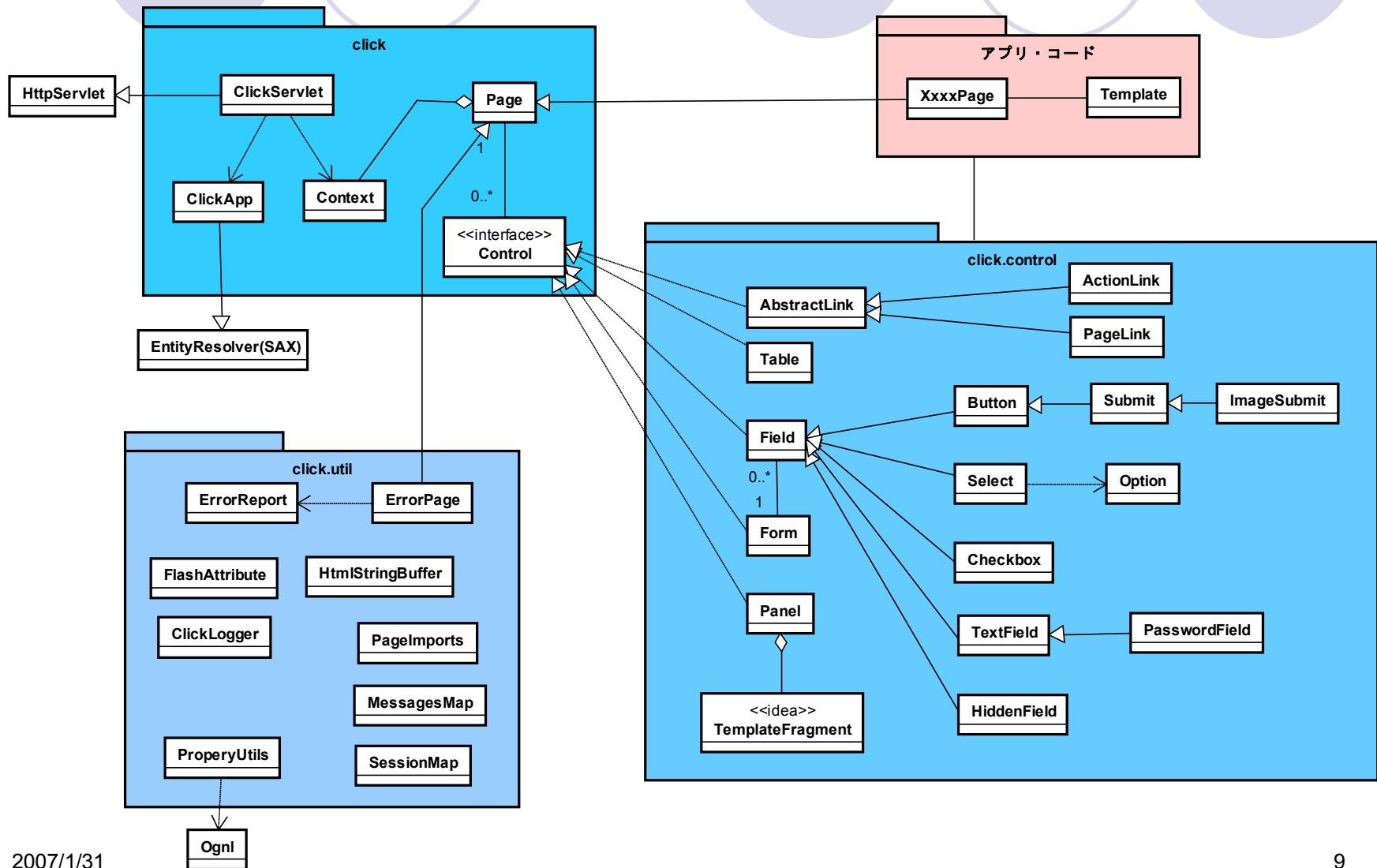
public void onRender() {
    table.setRowList(getMembers());
}
}
```

ClickServletの動作



<http://click.sourceforge.net/docs/pages.html> より

クラス・パッケージ構成



拡張コンポーネント

- Cayenne I/F: Form, テーブル表示、検索
- コントロール: 特定目的テキストなど...
Email, CreditCard, Telephone
- フィルター: 圧縮フィルター
- Hibernate I/F: Formなど
- パネル: GUIコンポーネント
- Spring: Springを組み込んだClickServlet

コンポーネント定義をコードに書くのはどうなんでしょうか？

- メリット

- HTMLをパースしなくてよい。
- HTML(タグ)とのひも付けをしなくて良い。
- フレームワークが小さく出来る。

- デメリット

- デザイナーとの共同作業が難しい。
- アプリケーションのコードが大きくなる。

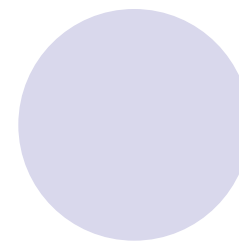
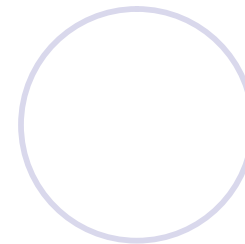
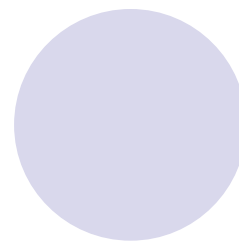
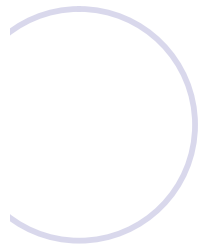
感想



- たしかにシンプル。
- 少ないコード量でそこそこのページが出来、お得感がある。(Railsもそうかもしれない)
- デザインが要求されないシステムでは使える。
- Examples はJavascriptを多用しておりClickの本質とは無関係に良さそうな印象を与える。
- テストコードが少ないのでは？



EY-Office



ご清聴、ありがとうございます。

有限会社 イーワイオフィス <http://www.ey-office.com>
吉田裕美 yy@ey-office.com