



Web 2.0的機能の開発環境

Ajax開発支援環境・テストツールのハイライト

XML コンソーシアム Web 2.0 部会
日立ソフトウェアエンジニアリング株式会社 宮崎 昭世
Sky株式会社 玉川 竜司



本プレゼンテーションの目的

- ノウハウの共有
 - Web2.0アプリの開発における手法、ツールの紹介
 - 開発プロセスを通しての、一貫的なストーリーでは**ありません**
 - 「私の抱えている課題は、他の誰かの課題でもある」
 - 疑問・質問・情報提供をよろしくお願いいたします
- コンソーシアム・部会の様子のご紹介
 - 「協調かつ競合」の'cooperative'な関係へ、ようこそ！



Web2.0アプリの開発プロセス



- 設計とプロトタイピング
 - 仕様書の限界
 - プロトタイピング手法のご紹介
- 実装
 - 注目されているのは
LL Framework (Ruby on Rails / TurboGears / CakePHP)、Atlas
 - Web2.0部会の初回部会で紹介済み
 - 資料はXMLコンソーシアムからご覧ください(要入会)
- テスト
 - オープンソースのSelenium / JMeterをご紹介
- 「永遠の 」



設計プロセス



- 特に Ajax を採用する場合の画面設計
 - Web 2.0 の UI としての画面設計はどうすればいいか?
 - 画面仕様”書”で大丈夫
紙ベースの設計書だと細かい動作までわからない...

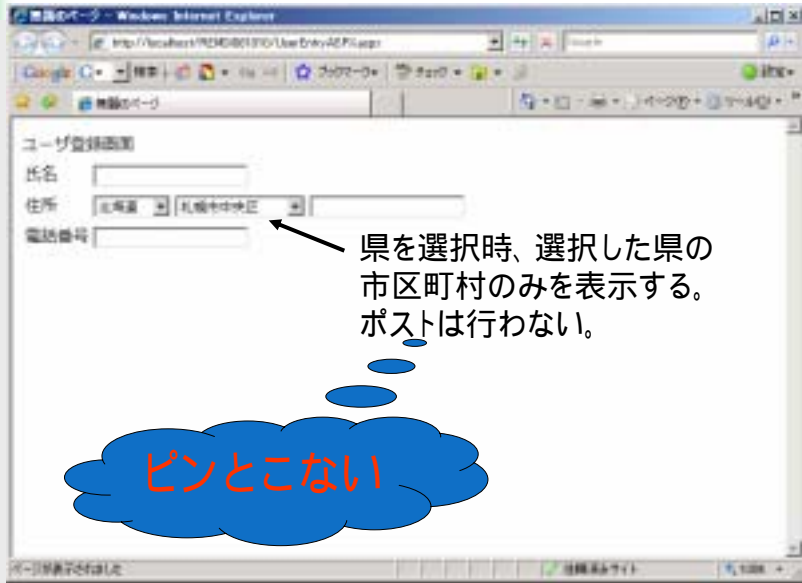
プロトタイピングによる
設計

- 操作性を確認するプロトタイプシステムを
容易に実装できる環境が必要
 1. Ruby on Rails に代表される Web アプリケーションを簡単に
実装する技術
 2. ASP.NET AJAX, jMaki のようなコントロールを貼り付ける事により
UI の動作までも実装する技術
 3. prototype.js に代表される JavaScript によるコードをより簡単に
実装する技術

画面の細かい動作まで実装する場合には、2のコントロール貼り付けベースの方が容易に実装可能だが、細かい部分の動作まで正確に実装する場合には、3のスクリプト拡張ベースも必要



画面設計書？



プロトタイプシステム



- 実際にさわってもらうとわかりやすい



Ajax 開発環境



XML Consortium

JavaScript 拡張

コントロール貼り付け

ASP.NET AJAX

Prototype

jQuery

MochiKit

script.aculo.us

Rialto javascript API

Rico

Dojo

Yahoo! UI Library

LiveControls (ASP.NET)

jMaki (JSF)

マスカット

コントロール貼り付けベースの方が容易だが、細かい動作を書く時にはスクリプトベースも必要

JavaScript 拡張ベースのライブラリについては下記URLが参考となる
http://www.atmarkit.co.jp/fwcr/special/ajax_kaitai03/01.html



デモ



XML Consortium

- JavaScript 拡張 ライブラリ
- たとえば、非同期でのWebサービス呼び出しは？

JavaScriptのみ	ASP.NET AJAX
Web サービス呼び出し部分	
<pre>xmlhttp.onreadystatechange = responseEventSelect1OnChange; var txtData = document.getElementById("Text1"); var strDat = txtData.value xmlhttp.open("GET", "/Service.aspx/GetData?strKey=" + strDat); xmlhttp.send(null);</pre>	<pre>WebService.HelloWorld(\$get("Text1").value, OnComplete);</pre>
Web サービス 取得データ処理部分	
<pre>function handleHttpEvent() { if (xmlhttp.readyState == 4) { if (xmlhttp.status == 200) { var objDOM = xmlhttp.responseXML; var strData; if (objDOM.documentElement) { strData = objDOM.getElementsByTagName("HelloWorldResult").item(0).text; if (strData == null) strData = '1'; } else { strData = '2'; } document.getElementById("Div1").innerText = strData; } else { window.alert("通信エラー"); } } }</pre>	<pre>function OnComplete(result) { \$('Div1').innerText = result; }</pre>



デモ



- コントロール貼り付け
同様の動作を実現するには？



Web 2.0 開発環境の動向



- 常に状況が変化する
 - 様々な環境が存在し常に変わっていく
 - 最新情報を共有しましょう。
- 使い分けのポイント
 - 要求されるSLAのレベルによって選択肢は変化する
 - 要求される UI の要件、開発期間によって選択肢は変化する
プロトタイプ開発時はコントロールベース
よりこった操作性が要求される場合はスクリプトベース
 - 複数のライブラリの共用もありうる
ASP.NET AJAXは、他のライブラリとの同時使用を考慮し、スクリプトベースで使えるメソッドの名前を考慮
ex) \$('name').\$get('name')
 - Web 2.0 部会では、より深い部分についても情報共有

情報の探し方も
Web 2.0 的



Webアプリのテストツール



- テストは自動化するべきもの
 - 「操作して、その結果が正しいか」
 - 永遠の ；
「で、修正していないところはほんとに前のまま？」
- ブラウザそのものの動作で確認する
Selenium (<http://www.openqa.org>)
- プロトコルレベルで確認する
JMeter (<http://jakarta.apache.org/jmeter/>)



Selenium



- 基本的なアイデア
 - Webアプリをブラウザのフレーム内で動作させる
 - 他のフレームからJavaScriptでDOMを操作し、アプリケーションを駆動する
 - 結果がどうなったかもDOMで取得、予想と一致しているか確認
 - **実装スタイル(Ajaxを使っているかどうか)は問題になりにくい**
- 初回のシナリオ作成は手間だが、あとは自動化できる
回帰テストに特に向いている
- デモンストレーション



Selenium



XML Consortium

- 製品構成
 - Selenium Core : テストの実行エンジン
 - Selenium IDE : テストシナリオ作成用IDE
 - Selenium Remote Control : 複雑なテストシナリオを実施するためのツール
 - Selenium on Rails : Railsアプリをより簡単にテスト
 - 使ったことがないので今回は省略...
- ライセンス : Apache License 2.0

© XML Consortium



Selenium Core



実施するテストシナリオのリスト、HTMLのテーブルで記述

テストシナリオ、HTMLのテーブルで記述、テストを実施すると、成否が色分けされていく

コントロールパネル、テストの開始・終了・結果表示

The screenshot shows the Selenium TestRunner application. On the left, there is a 'Passing Test Suite' section with a list of test cases. The main area displays a table of test cases with columns for 'Test Type', 'URL', 'username', 'password', and 'Expected outcome'. The table contains several rows of test data. On the right, there is a 'Selenium TestRunner' control panel with buttons for 'Run', 'Stop', and 'Pause', and a status display showing '3 tests', '0 passed', '3 failed', and '0 incomplete'.

Test Type	URL	username	password	Expected outcome
verifyElementPresent	id=login_selenium_test_case_page1.html	username	password	title
verifyElementPresent	id=login_selenium_test_case_page1.html	username	password	Default username not found
verifyElementPresent	id=login_selenium_test_case_page1.html	password	password	title
verifyElementPresent	id=login_selenium_test_case_page1.html	password	password	Default password not found
verifyTitle	id=login_selenium_test_case_page1.html	username	password	title
verifyTextPresent	id=login_selenium_test_case_page1.html	username	password	title

404 Not Found

/selenium/console/demo/passing.html/test_case_page1.html was not found on the server

© XML Consortium



Selenium IDE



XML Consortium

- テストシナリオ作成用IDE
- テストシナリオの「録画」
- 録画したシナリオの編集
- テストシナリオのデバッグ
- CoreだけでなくRemote Control用のスクリプトも生成可能
- デモンストレーション

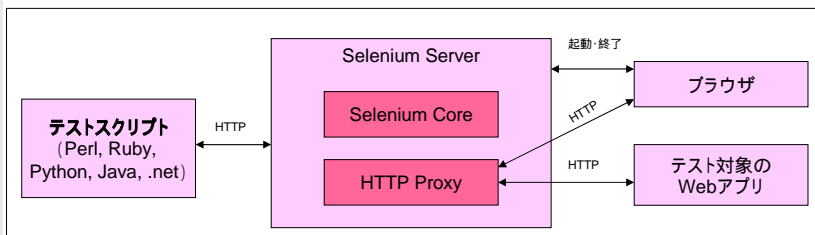


Selenium Remote Control



XML Consortium

- Selenium Serverがテストの実行を制御
- コマンドラインからテスト実行
- 各種スクリプト言語からテスト実行
 - 環境設定 テスト実施 結果確認を自動化
- デモンストレーション





Tips



- DOMやフレームに依存していることからくる限界
 - アプリケーション側のJavaScriptによってはテスト不可
 - きれいなHTMLの生成を心がける
 - FireFox ExtensionのHTML Validatorは便利
 - <http://users.skynet.be/mgueury/mozilla/>
 - DOMの要素にはIDをつけよう
- 機能不足と思ったら
 - Contributed User-Extensionsは要チェック
 - <http://wiki.openqa.org/display/SEL/Contributed+User-Extensions>
- それでも機能不足だと思ったら
 - Selenium CoreはJavaScriptで書かれている
 - 必要なら自力で改造可能
 - もちろんExtensionも自作可能
- 複雑なテストシナリオは自動生成を考えよう
 - テストシナリオはHTML
 - LLやExcelで、テストパターンからシナリオを自動生成



JMeter



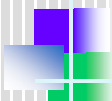
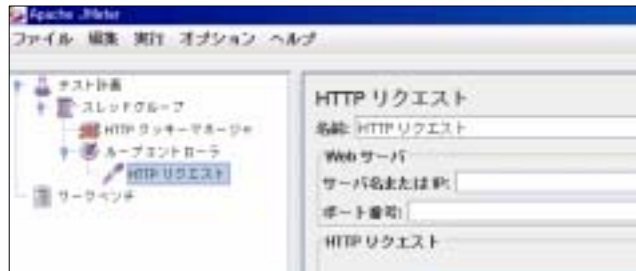
- 基本的なアイデア
 - クライアントのふりをして、サーバと通信する
 - プロトコルのレベルでのテスト
 - データを受信した結果の、クライアント(ブラウザ)の動作は関知しない
- 特徴
 - Pure Javaのアプリケーション
 - マルチスレッドで、大量の平行クライアント・アクセスをシミュレートできる(メモリとCPUがあれば)
 - ストレス・テストに特に向いている
- ライセンス: Apache License 2.0
- デモンストレーション



JMeterの機能



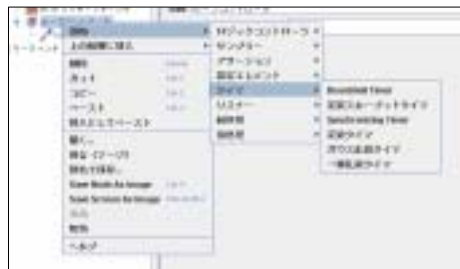
- HTTPクライアントのシミュレート
 - ひとつのJMeterのインスタンスで、複数のクライアントインスタンスを独立スレッドで動作可能
 - クッキーをサポート(スレッドごと)



JMeterの機能



- 柔軟な実行制御
 - ループのスピード
(一定時間のウェイト / 単位時間当たりのリクエスト数一定 / ランダム)
 - スレッド群の起動制御 (一挙に起動 / すこしずつ起動)
 - 関数やスクリプトの利用





JMeterの機能



- テストの結果をさまざまな方法で処理できる
 - HTTPレスポンスでアサーション
 - 成功・失敗・レスポンスタイムを表やグラフで表示
 - レスポンスデータをすべてを保存することもできるが、メモリを大量に消費する



試験レポート

名前: 3/21レポート

全てのデータをファイルに保存

ファイル名: ログエラーのみ

名前	# Samples	Average	Median	95% Line	Min	Max	Err%	Throughput	UI Error
HTTP リクエスト	8143	1825	226	2641	31	818175.62%	31.7/sec	1433.50	
合計	8143	1825	226	2641	31	818175.62%	31.7/sec	1433.50	



JMeterの機能



- 多彩なプロトコルサポート
 - HTTP / HTTPS / ftp / LDAP / JDBC / SOAP / JMS / AJP / POP3 / IMAP
- リモート・テスト
 - 一台のGUIから、複数台のJmeterエンジンを制御
 - 莫大な負荷をかけるストレス・テストに有効



Tips



- まずはProxyを活用しましょう
 - 記録して、不要分を除いて、ループに入れる
- ドキュメントに一通り目を通しましょう
 - JMeterは非常に多機能
 - 1時間ドキュメントを読むことで、後の3日が節約できるかも
- 関数とスクリプト
 - 乱数や連続した数値を生成する関数がある
 - 組み込み関数でも足りない場合は、JavaScriptやBeanShellを使うこともできる(難易度高 - 資料が少ないので)
- テストシナリオはXML



注意



- メモリは大量に必要(1GBくらいは欲しい)
- GCには要注意
 - レスポンスタイムを厳密に計ることはできない。特にワーストケースに注意
 - GCの間にレスポンスが来ると、非常に悪いレスポンスタイムとして記録されてしまう
 - JMeterの起動時に、VMにメモリを多めに与える
 - 必要ならGCのチューニングを



それではまた、
XMLコンソーシアム Web2.0部会で
お会いしましょう！