Integration Case Studies

Mike Nowak
mnowak@umich.edu

Session agenda

- Review: Five integration questions to answer
- 6 case studies
- When we didn’t use the MTS Engine
Five integration questions to answer

Input
1. How will I collect my data?
2. Where will my data be stored and how do I get to it?

Processing
3. What's my strategy for running the MTS Engine?
4. What triggers the tailoring process?

Output
5. How do I handle the tailored output?

Project Quit Phase 1 Diagram

WebObjects app
HTTP interface to Engine
Tailoring Engine
XML data files
Tailored HTML
Web survey
Tailored pages
Data collection

- Web survey, 6 month phone follow up.

Data storage

- Data stored in XML format in files on the server.

Engine integration

- Engine runs native as its own server with an HTTP interface. All data passed in GET.

Tailoring trigger

- WebObjects application requests tailored results on demand when the user hits a page.

Tailored output

- HTML results are integrated by the WebObjects application into pages.

Goal: Provide tailored web pages for a smoking intervention in 6 sessions over 6 weeks. 2000 participants.

MENU Diagram
Goal: Provide tailored web pages for a fruit and vegetable intervention in 4 sessions over 4 months. Last 2 sessions are tailored. 2500 participants.

<table>
<thead>
<tr>
<th>Data collection</th>
<th>Data storage</th>
<th>Engine integration</th>
<th>Tailoring trigger</th>
<th>Tailored output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web survey, 3 month web follow up, 6 month phone follow up.</td>
<td>Data stored in XML format in files on the server.</td>
<td>Engine runs native as its own server with an HTTP interface. Subject id passed in GET.</td>
<td>WebObjects application requests tailored results on demand when the user hits a page.</td>
<td>HTML results are integrated by the WebObjects application into pages.</td>
</tr>
</tbody>
</table>

**Project Quit Phase 2 Diagram**
**Project Quit Phase 2**

Goal: Provide tailored web pages for a smoking intervention in 6 sessions over 6 weeks. 1800 participants.

<table>
<thead>
<tr>
<th>Data collection</th>
<th>Data storage</th>
<th>Engine integration</th>
<th>Tailoring trigger</th>
<th>Tailored output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web survey, 6 month phone follow up.</td>
<td>Data stored in MySQL database on server.</td>
<td>Engine runs native as as its own server with an HTTP interface. Subject id passed in GET.</td>
<td>WebObjects application requests tailored results on demand when the user hits a page.</td>
<td>HTML results are integrated by the WebObjects application into pages.</td>
</tr>
</tbody>
</table>

**SteppingUp Diagram**

- Drupal app
- HTTP interface to Engine
- SSH tunnel
- MySQL
- Tailored pages
- Tailored HTML
- Subject
- Results
- Tailoring Engine
- HTTP request
SteppingUp

Goal: Provide progress charts, 6 tailored sessions and 100-200 tailored tips. First example of tailoring engine running as a service for another group. Richardson’s group now runs their own engine.

<table>
<thead>
<tr>
<th>Data collection</th>
<th>Data storage</th>
<th>Engine integration</th>
<th>Tailoring trigger</th>
<th>Tailored output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web survey.</td>
<td>Data stored in several tables in MySQL database on remote server; accessed over ssh tunnel.</td>
<td>Engine runs as its own server with an HTTP interface over ssh tunnel. Subject id passed in GET.</td>
<td>Drupal (PHP) application gets all tailored results in batch and stores them in the database.</td>
<td>Stored HTML results are integrated by the Drupal application into pages.</td>
</tr>
</tbody>
</table>

Eat for Life Diagram

[Diagram showing data flow from WebObjects app, Cocoa app, MySQL, and Dogi results to Tailoring Engine]
Eat for Life

Goal: Provide 4 printed tailored booklets and newsletters across 12 weeks. Multiple layouts for different ethnic identity types. 1200 participants. Uses Java code to compute ethnic identity.

<table>
<thead>
<tr>
<th>Data collection</th>
<th>Data storage</th>
<th>Engine integration</th>
<th>Tailoring trigger</th>
<th>Tailored output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone survey. Data</td>
<td>Data is loaded into local MySQL database.</td>
<td>Engine runs as its own server with an HTTP interface. Subject id passed in GET.</td>
<td>Scheduled program produces data files for operator who uses a custom print app.</td>
<td>Kid templates used to embed tailored results in web pages that are printed as booklets.</td>
</tr>
<tr>
<td>is provided in</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>encrypted files</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>delivered by FTP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MTS Workbench Diagram

[MTS Workbench Diagram]

MTS Workbench

... Message Editor plug-in

Preview plug-in

Tailoring plug-in

Jython

subj

result

Tailoring Engine

University of Michigan Tailoring Workshop
MTS Workbench

Goal: Allow writers to author tailored messages and preview results.

<table>
<thead>
<tr>
<th>Data collection</th>
<th>Data storage</th>
<th>Engine integration</th>
<th>Tailoring trigger</th>
<th>Tailored output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author creates a dictionary for the project and test cases with values.</td>
<td>Data is stored in files in JSON format.</td>
<td>Engine runs embedded in Jython environment.</td>
<td>Authors select a message document and a test case and clicks Preview button.</td>
<td>HTML displayed in Preview results, with error messages and search.</td>
</tr>
</tbody>
</table>

Project summary

<table>
<thead>
<tr>
<th>Project</th>
<th>Data collection</th>
<th>Data storage</th>
<th>Engine integration</th>
<th>Tailoring trigger</th>
<th>Tailored output</th>
</tr>
</thead>
<tbody>
<tr>
<td>PQ Phase 1</td>
<td>Web survey</td>
<td>XML</td>
<td>HTTP</td>
<td>Web app</td>
<td>Web page</td>
</tr>
<tr>
<td>MENU</td>
<td>Web surveys</td>
<td>XML</td>
<td>HTTP</td>
<td>Web app</td>
<td>Web page</td>
</tr>
<tr>
<td>PQ Phase 2</td>
<td>Web survey</td>
<td>MySQL</td>
<td>HTTP</td>
<td>Web app</td>
<td>Web page</td>
</tr>
<tr>
<td>SteppingUp</td>
<td>Web survey</td>
<td>MySQL over ssh</td>
<td>HTTP</td>
<td>Web app</td>
<td>Web page</td>
</tr>
<tr>
<td>Eat for Life</td>
<td>Phone survey</td>
<td>MySQL</td>
<td>HTTP</td>
<td>Web app</td>
<td>Print</td>
</tr>
<tr>
<td>MPOWER</td>
<td>Web survey</td>
<td>MySQL</td>
<td>Python</td>
<td>Python app</td>
<td>SMS message</td>
</tr>
<tr>
<td>MTS Workbench</td>
<td>Testcase editor</td>
<td>JSON</td>
<td>Jython</td>
<td>Java app</td>
<td>HTML Preview</td>
</tr>
</tbody>
</table>
When we didn't use the MTS Engine

- Projects which tailor on order more than content
  - Guide to Decide
  - PuffCity and PuffCity 2

- Smaller or fast track projects without much tailoring
  - Decider Guider
  - CSATS
  - Colorectalweb

- MTS Engine is now agile enough to use for smaller projects and to experiment with different kinds of tailoring.