CSE 135

Data-driven description of the web application

In previous episodes...

In next episodes...

• Server-side applications
  – Lectures showed simple examples of servlet and JSP technology

More technologies coming up

• Database programming: DB design, SQL
• Java beans, tag libraries, Javascript, Ajax

Complexity of examples and projects going way higher

• From dummy examples to real applications
• By 4th week we’ll have seen that jsp’s become unmanageably long and messy when applied to non-trivial problems

Success in the field is not just about technologies & coding!

• Today, data flows specification of the business process
  – Specification method tuned to human-centric web applications
• The Model-View-Controller design pattern in web applications

Today’s material will assist in Phase 1
The Larger Issue:
How do you specify an application and gradually turn it into implementation

• Very soon you will not be a solitary hacker!
  – Customers w/ contracts
  – Product managers who need to know what you build without looking into your code
  – Software architects who turn the product requirements into software architecture
  – Engineering managers who need easy access to your code
  – Colleagues that work close with you

• You will deal with communication frictions in specifying an application and turning specification into code systematically

Web Application Delivery Process and the Frictions

Business Process Owner (Customer, Manager, Pointy-haired boss, ...)

Analysis/Specification Phase
COMMUNICATION
business process and specification of Web application

Chief Architect/Technical Project Leader

Development Phase
COMMUNICATION
technical specification and development

PROBLEM:
Informal, imprecise specification by business process owner. We need to make it a precise “contract”

• Code developed may be inconsistent with spec
• Difficulties in involving multiple people to code base

Problem is even worse in application evolution phase when application logic is hidden in thousands of lines of code
The Cartoonist’s Take on the Pains of Communication

Languages for Application Specification

- Lightweight, process-oriented approaches
  - BPML specifications describe sequence of steps of business process
  - The specification can leave too much detail out

- Close-to-the-code approaches
  - UML
  - Too heavy for people interested in application behavior but not really its implementation details
Collaborative Dataflows

- Business process specification community realizes that most important aspect of an application is the flow of data across the users of the application and the code components of the application
  - IBM's Siena and Artifact-driven approaches
- CSE13S: Dataflows
- Data-centric specifications tuned to human-centric web applications
- A dataflow summarizes the application using a graph involving
  - user groups,
  - pages with actions (think of the actions as the nodes of the graph)
  - edges that describe the effect of an action of a user on what (other) users can do
- Forces you to think about the function of participants in each page of the application

An example application

A fuzzy, initial high level description of the application (the details of an application that implements this specification are still ambiguous and missing):

The WebDB2010 conference has a paper reviewing application. The chairs of the conference declare reviewers. Any registered user can submit at most one paper. A chair of the conference assigns the paper to reviewers. The reviewer reads the paper and submits a comment, a grade and possibly invites the authors to provide feedback on their paper. The reviewer does not see the feedback. The chair sees the reviews and feedback for each paper and decides whether the paper is accepted or rejected at the conference.
For the sake of explaining data flows we will now first see a complete WebDB2010 reviewing application and then we’ll see its data flow.

Remember, in reality (and in our class project) the order is the opposite: First you will build the data flow, then we’ll review it and then you’ll actually code it.
### WebDB 2010

#### Provide Feedback

<table>
<thead>
<tr>
<th>Title</th>
<th>PDF</th>
<th>Screenshot</th>
<th>Comment</th>
<th>Score</th>
<th>Invite Author Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppLogic Data Services</td>
<td><a href="#">applogic.pdf</a></td>
<td><img src="#" alt="Screenshot" /></td>
<td>Novel idea, but is it practical?</td>
<td>?</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Review Assigned Papers**

<table>
<thead>
<tr>
<th>Title</th>
<th>PDF</th>
<th>Screenshot</th>
<th>Comment</th>
<th>Score</th>
<th>Invite Author Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppLogic Data Services</td>
<td><a href="#">applogic.pdf</a></td>
<td><img src="#" alt="Screenshot" /></td>
<td>Novel idea, but is it practical?</td>
<td>?</td>
<td>Yes</td>
</tr>
<tr>
<td>Kepler Scientific Workflow</td>
<td><a href="#">kepler.pdf</a></td>
<td><img src="#" alt="Screenshot" /></td>
<td>Yes</td>
<td>No</td>
<td>cancel</td>
</tr>
</tbody>
</table>

**WebDB 2010**

#### Provide Feedback

<table>
<thead>
<tr>
<th>Title</th>
<th>PDF</th>
<th>Screenshot</th>
<th>Comment</th>
<th>Score</th>
<th>Invite Author Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppLogic Data Services</td>
<td><a href="#">applogic.pdf</a></td>
<td><img src="#" alt="Screenshot" /></td>
<td>Novel idea, but is it practical?</td>
<td>?</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Components of a dataflow

- State each group of users of the application
  - ovals at left, create a swimlane for each group
- List the pages of the application
  - Try to place a page P accessible by group G in the swimlane of group G
- In each page list the actions that users can accomplish on this page
- Note: It is important to figure out early on what are the data objects of your application
  - Eg, papers, reviews, etc
Action format

- *UserGroup* see *DataObject*
- *UserGroup* see (join of) ReportDataObject1, ReportDataObject2, ...
- *UserGroup* act *FormDataObject*
  ReportDataObject1, ...

Edges: How a user’s actions enable ability of other users to perform actions

When a *Chair* assigns (an assignment) to a reported pair of *Paper* and *Reviewer*, then this *Reviewer* can read this *Paper*. 
What about aspects that data flow is not enough to describe

- **View layer**: Provide a characteristic snapshot of each page
  - Just build with an html editor
  - The snapshots should exhibit all actions
- **Business logic**: Some conditions not captured by data flow reasonably succinctly
  - Add “bubbles” and commentary describing them
Relaxations

• Relaxation 1: Whenever there is both an enablement and a disablement action with the same source and target nodes, just keep one edge and put in a fraction format "enabling action/disabling action".

• Relaxation 2: Remove the "edit" edges.
Project

• Create data flow

• Make demo snapshots for the pages
  – An interesting snapshot has at least some data and illustrates actions that can be performed

• Add narrative/story around pages

• Submit a zip by Monday midnight