Where we are?

<table>
<thead>
<tr>
<th>#</th>
<th>Date</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5th March</td>
<td>Web Engineering Introduction and Overview</td>
</tr>
<tr>
<td>2</td>
<td>12th March</td>
<td>Requirements Engineering for Web Applications</td>
</tr>
<tr>
<td>3</td>
<td>19th March</td>
<td>Web Application Modeling</td>
</tr>
<tr>
<td>4</td>
<td>9th April</td>
<td>Web Application Architectures</td>
</tr>
<tr>
<td>5</td>
<td>16th April</td>
<td>Developing Applications with WebML</td>
</tr>
<tr>
<td>6</td>
<td>23rd April</td>
<td>Testing and Usability of Web Applications</td>
</tr>
<tr>
<td>7</td>
<td>30th April</td>
<td>Maintenance and Performance of Web Applications</td>
</tr>
<tr>
<td>8</td>
<td>7th May</td>
<td>Web Technologies I</td>
</tr>
<tr>
<td>9</td>
<td>14th May</td>
<td>Web Technologies II</td>
</tr>
<tr>
<td>10</td>
<td>21st May</td>
<td>Web Application Development Process</td>
</tr>
<tr>
<td>11</td>
<td>28th May</td>
<td>Project Management for Web Applications</td>
</tr>
<tr>
<td>12</td>
<td>4th June</td>
<td>Web Application Security</td>
</tr>
<tr>
<td>13</td>
<td>11th June</td>
<td>Mobile Application Development I</td>
</tr>
<tr>
<td>14</td>
<td>18th June</td>
<td>Mobile Application Development II</td>
</tr>
<tr>
<td>14</td>
<td>25th June</td>
<td>Final Exam</td>
</tr>
</tbody>
</table>
Recap

Web applications have 3 major building blocks:

1. Content
   - Document-centric, multimediaility
   - Generating, updating, integrating, quality assurance

2. Hypertext (Navigation)
   - Non-linear nature of hypertextual documents
   - Hypertext models: links, nodes (URL), anchors
   - Disorientation

3. Presentation
   - User Interface
   - Self-explanatory
   - Aesthetics
INTRODUCTION

Why do we need modeling?

Why models?
Why models?

- Define an abstract view of a real-world entity
  - Finding & discovering objects/concepts in a domain
  - Assigning responsibilities to objects
- Tool of thought
  - Reduce complexity
  - Document design decisions
- Means of communication in your development team

Software Application Modeling

- Levels – the “how” & “what” of an application
- Aspects
  - Structure: objects, attributes, and relationships
  - Behavior: function & processes
- Phases – Development cycle
Web Application Modeling

Levels
- Presentation
- Hypertext
- Content
- Structure
- Behavior

Aspects
- Levels – Information, node/link structure, UI & page layout separate.
- Aspects – Same as Software Applications
- Phases – Approach depends upon type of application
- Customization – Context information, personalization

Modeling requirements

- Use cases for functional requirements (graphical representation)
- Navigational use cases
  - See example (conference system like easychair)!
- In Web Engineering, we are concerned with
  - Content modeling
  - Hypertext modeling
  - Presentation modeling
Web Modeling

- Modeling static & dynamic aspects of content, hypertext, and presentation

- We focus on object-oriented analysis & design
  - Analysis: Finding & discovering classes of objects/concepts in a domain
  - Design: Defining software objects & how they interact to fulfill requirements.

Objects

- Software entities – like real-world entities - that consist of states and behaviors

  - States:
    - Variables store the states of an object's properties
    - Hidden from the outside world (data encapsulation)

  - Behaviors:
    - Methods define the object's behaviors
    - Used by objects to communicate with other objects

  - Classes
    - blueprints for creating objects of a particular type
Discovering Objects in a Domain

• The way we represent a domain’s software model should resemble the physical model as closely as possible

• To find key classes of objects:
  – Reuse existing models, if they exist
  – Make a category list
    • People, places, things
    • Transactions
    • Events
  – Identify noun phrases

• When naming classes, use terms that are commonly used in the domain
  – i.e., terms users would understand

Assigning Responsibilities

• Responsibilities are an object’s obligations, or behaviors related to its role in the system

• What does an object do?
  – Doing something (to) itself
  – Pass actions (messages) to other objects
  – Controlling & coordinating the activities in other objects

• What does an object know?
  – Private, encapsulated data
  – Its related objects
  – Items it can derive or calculate
A simple walkthrough case study

REFERENCE SCENARIO

The Conference Review System

• This case study was presented at IWWOST 2001 to compare different Web application modeling methods

• The purpose of the system is to support the process of submission, evaluation and selection of papers for a conference.
Actors I

- PC Chair
  - creating the conference
  - determining the conference topics (or tracks) and subjects
  - establishing the Program Committee
  - defining the final list of accepted and rejected papers
  - defining the conference deadlines: submission, review, and notification.

- PC Member
  - evaluating a set of papers assigned to him
  - indicating another person as a reviewer of a paper
  - advising the PC Chair for the final list of accepted papers

Actors II

- Reviewer
  - responsible for reviewing a paper

- Author
  - submitting a paper for acceptance at the conference
  - PC Members and Reviewers may also be Authors, they must have different Ids for each role
Functions I: Paper Submission

• Any registered author may submit a paper
  – The author must register: the title, the abstract, the conference track, and a set of subjects chosen from a list previously determined by the PC Chair, if there is one
  – The system, after checking the authors’ registrations, assigns a paper ID to the new paper, and allows the user to submit it by uploading a file
  – At any moment, an author is allowed to check the data about his submitted papers. Until the submission deadline, the author is also allowed to substitute the uploaded file by a new one, or to change any of the informed data about the paper

Functions II: Assignment of Papers to PC Members

• The PC Chair may indicate potential conflicts of interest between PC Members and submitted papers

• Once the submission deadline has been reached
  – PC Members may indicate their interest and also conflicts of interest with some papers
  – In case of conflict of interest, the PC Member will not see any information about the paper
  – The PC Chair assigns papers to PC Members for reviewing, an email message with the list of papers, and a URL to a page where he can access these papers is sent
Functions III: Entering a Review

- A PC Member, or a Reviewer, may enter a review for a paper assigned to him
- The review is entered by accessing a form containing all the evaluation items
- A PC Member may see other reviews (entered by others) for any of the papers he is reviewing, but only after he has entered his own review
- The PC Chair has full access to all papers and all reviews

Function IV: Choosing Accepted and Rejected Papers

- Once the review deadline has been reached, the review process is closed
- The PC Chair, taking into account the recommendations of the PC Members and reviewers, chooses the papers that will be accepted and rejected
- Once the process is marked as finalized by the PC Chair, the system issues a notification message to paper authors, which includes the appropriate parts of the reviews submitted by the PC Members and reviewers
How to model the data underlying a Web application

CONTENT MODELING

Introduction

- Purpose: To model the information requirements of a Web application
  - Diagramming the structural (i.e., information objects) & behavioral aspects of the information.
  - NOT concerned with navigation or presentation.

- Primary Models
  - Class diagrams – enough for static applications.
  - State machine diagrams – captures dynamic aspects
Content Structure Model

Content Behavior Model
How to model the hypertext of a Web application

**HYPERTEXT MODELING**

**Introduction**

- Purpose: To model the navigation paths available to users
- UWE Artifacts* (UML-based Web Engineering)
  - Hypertext structure model (navigation model)
  - Access model
- Focuses on the structure of the hypertext & access elements
- Use "<<navigation class>>" annotation to distinguish from content classes
- Based on content models

* [http://uwe.pst.ifi.lmu.de/index.html](http://uwe.pst.ifi.lmu.de/index.html)
Different Models... Different Links...

- **HDM (Hypertext Design Model)**
  - Structural links connect elements of the same node (e.g. review summary to review details)
  - Perspective links put various views of a node in relation (e.g. PS and PDF of a paper)
  - Application links put different nodes in relation depending on application (e.g. link pointing to best paper)

- **WebML (Web Modeling Language)**
  - Contextual links carry context information (e.g. ID of reviewer)
  - Non-contextual links have no associated context information (e.g. link from single review to list of all reviews)
  - Intra-page links: source and destination are on the same page further down/up
  - Inter-page links: different pages

- **UWE**
  - Navigation links (navigate between nodes, e.g. Papers and authors)
  - Process links (start node of a process, e.g. Beginning of review submission)
  - External links (external page)

- ...

Navigation Structure Model
Navigation Access Model

- Hypertext structure models describe navigation, but not orientation.
- Access models describe both through Navigation patterns, used to consistently describe conventional elements.
  - <<index>> (list of objects of the same type)
  - <<menu>> (list of heterogeneous objects)
  - <<guided-tour>> (sequential links)
  - <<query>>
How to model the look & feel of a Web application

PRESENTATION MODELING

Presentation modeling

• Purpose: To model the look & feel of the Web application at the page level.

• The design should aim for simplicity and self-explanation.

• Describes presentation structure:
  – Composition & design of each page
  – Identify recurring elements (headers/footers)

• Describes presentation behavior:
  – Elements => Events
Levels of Presentation Models

- Presentation Page – “root” element; equivalent to a page container.

- Presentation Unit
  - A fragment of the page logically defined by grouping related elements.
  - Represents a hypertext model node

- Presentation Element
  - A unit’s (node’s) informational components
  - Text, images, buttons, fields

Presentation Structure Model

```
<page>
  <presentation unit>
    <text>
      PaperID
    </text>
    <text>
      SubmissionDate
    </text>
    <text>
      Title
    </text>
    <text>
      Abstract
    </text>
    <anchor>
      FullVersion(Pdf)
    </anchor>
    <anchor>
      SubmitReview
    </anchor>
    <anchor collection>
      Authors
    </anchor collection>
  </presentation unit>
</page>

<page>
  <presentation unit>
    <text>
      Name
    </text>
    <text>
      Affiliation
    </text>
    <text>
      E-mail
    </text>
    <anchor>
      SubmitChanges
    </anchor>
  </presentation unit>
</page>
```
Presentation Behavior Model

From sketch models to code models

MODEL DRIVEN DEVELOPMENT
Why Models at All?

- When it comes down to it, the real point of software development is cutting code.
- Diagrams are, after all, just pretty pictures.
- No user is going to thank you for pretty pictures; what a user wants is software that executes.

*M. Fowler, "UML Distilled", 1st edition, Addison Wesley, 1997*

Unified Modeling Language (UML)

- “The Unified Modeling Language is a visual language for specifying and documenting the artifacts of systems.”
- Language of choice (and ISO standard) for diagramming notation in OO development
  - Structural – Class diagrams (domain models)
  - Behavioral – Use Cases, Sequence diagrams
- Currently at version 2.5, although many analysts and designers still use 1.x
The Role of Model in the Development

- **Models as sketch**
  - For communicating ideas and alternatives
  - Essence: Selectivity
  - “Sketchers” don’t have to care much about

- **Models as blueprint**
  - All design decisions (maybe of a particular area) are laid out
  - Essence: Completeness – programming should be pretty straightforward
  - Issue of reverse engineering

- **Models as program**
  - Applications are automatically generated
  - Essence: models become the source code

Model - Code Interplay

- Code only
- Code Visualization
- Roundtrip Engineering
- Model-centric / Model-driven
- Model only

"Models as Code"
Model-Driven ...

• Systematic development on basis of models

• Models become the first hand artifacts in the software development cycle

• Key concepts
  – abstraction from implementation detail
  – systematic transformations

• Related Terminology
  – Model Driven [Software] Engineering (MDE),
  – Model Driven [Software] Development (MDD/MDSD),
  – Model Driven Architecture (MDA)
  – Model Driven Web Engineering (MDWE)

What is Model Driven Architecture?

• MDA is defined and developed by the Object Management Group (OMG) since March 2001

• MDA is:
  – "Model-Driven ..."-framework for software development, defined by the OMG
  – open, vendor-neutral approach to interoperability using OMG's modeling specifications:
    • Unified Modelling Language (UML), Meta-Object Facility (MOF) and Common Warehouse Model (CWM)

• Main ideas:
  – Addresses the complete system development life cycle
  – Separate specification from implementation
  – Specify a system that is independent of a platform
  – Code generation
Model-Driven Development (MDD)
The Vision

- Should go far beyond the notion of CASE (Computer Aided Software Engineering) tools of the 80’s

- Reduced gap between problem and realization domain
  - models as primary artefact throughout the lifecycle instead of code
  - models as program instead of models as sketch/blueprint

- Systematic transformations of abstract models to concrete implementations

- Standards for uniform storage, exchange, and transformation of models

Developing in the MDA

- PIM
  - Platform Independent Model (PIM) represents business functionality and behavior without technology details

- PSM
  - Applies a standard mapping to create or generate a Platform Specific Model (PSM) from the PIM

- Code Model
  - Create or generate the code for PSM
Modeling Methods (not all are MDA)

UML for Web Engineering

- UML Web Engineering (UWE) notation
  - UML-compliant
  - Comprehensive modeling tool
  - Download it here: http://www.pst.ifl.mtu.de/projekte/uwe/
  - Requires MagicDraw UML (the free edition is enough)
OVERVIEW OF OTHER MODELING METHODS
That’s almost all for day…

WRAP-UP

Things to keep in mind
(or summary)

• Modeling is fundamental
  – Helps development
  – Supports communication

• Model Driven Design and Development
  – Automatic code generation of Web applications

• One model for each layer
  – Content
  – Navigation
  – Presentation

• Different methods have different expressive power
Bibliography

• Mandatory reading

• Suggested
  – UML-based Web Engineering
    • [http://uwe.pst.ifl.lmu.de/index.html](http://uwe.pst.ifl.lmu.de/index.html)

Next Lecture

<table>
<thead>
<tr>
<th>#</th>
<th>Date</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5th March</td>
<td>Web Engineering Introduction and Overview</td>
</tr>
<tr>
<td>2</td>
<td>12th March</td>
<td>Requirements Engineering for Web Applications</td>
</tr>
<tr>
<td>3</td>
<td>19th March</td>
<td>Web Application Modeling</td>
</tr>
<tr>
<td>4</td>
<td>9th April</td>
<td>Web Application Architectures</td>
</tr>
<tr>
<td>5</td>
<td>16th April</td>
<td>Developing Applications with WebML</td>
</tr>
<tr>
<td>6</td>
<td>23rd April</td>
<td>Testing and Usability of Web Applications</td>
</tr>
<tr>
<td>7</td>
<td>30th April</td>
<td>Maintenance and Performance of Web Applications</td>
</tr>
<tr>
<td>8</td>
<td>7th May</td>
<td>Web Technologies I</td>
</tr>
<tr>
<td>9</td>
<td>14th May</td>
<td>Web Technologies II</td>
</tr>
<tr>
<td>10</td>
<td>21st May</td>
<td>Web Application Development Process</td>
</tr>
<tr>
<td>11</td>
<td>28th May</td>
<td>Project Management for Web Applications</td>
</tr>
<tr>
<td>12</td>
<td>4th June</td>
<td>Web Application Security</td>
</tr>
<tr>
<td>13</td>
<td>11th June</td>
<td>Mobile Application Development I</td>
</tr>
<tr>
<td>14</td>
<td>18th June</td>
<td>Mobile Application Development II</td>
</tr>
<tr>
<td>15</td>
<td>25th June</td>
<td>Final Exam</td>
</tr>
</tbody>
</table>