Topic 06

Web Engineering: Analysis Modeling Phase

Why Do We Need Modeling (Analysis and Design) of WebApp?

- **Analysis Modeling** helps you understand the nature of customer’s requirements or the problem being addressed and the “shape” of the WebApp that will allow you to address that problem.
- **Design Modeling** is about understanding the internal structure of the WebApp to-be-developed and how this creates the shape of the WebApp that was identified by the analysis model.
Importance of Analysis Modeling

- It is **impossible** to begin the design and development (D&D) of a WebApp increment if you have no understanding of what is required.
- **Analysis Modeling** helps you to understand customer’s requirements in details -- that will allow Web Engineer to satisfy better user requirements (needs).
- **Analysis Models** provide technical details regarding
  1. content of WebApp (content model),
  2. interaction with or GUI of WebApp (interaction model),
  3. main functions and behavior of WebApp (functional model), and
  4. configuration or composition of WebApp (configuration model).

How much Analysis Modeling needed? It depends on:
- Size and complexity of the WebApp increment
- Number of stakeholders (analysis can help to identify conflicting requirements coming from different sources)
- Size of the WebE team
- Degree to which members of the WebE team have worked together before (analysis can help develop a common understanding of the project)
- Degree to which the organization’s success is directly dependent on the success of the WebApp

Analysis Modeling = FUNCTION (size, complexity, team specs, experience, knowledge, etc.)

---

WebE Analysis Models: Goals and Outputs/Diagrams

<table>
<thead>
<tr>
<th>WebE Analysis Models</th>
<th>Goal</th>
<th>Diagrams to be used to reflect analysis models</th>
<th>Tools to be used</th>
</tr>
</thead>
</table>
| Interaction Model    | Describes the manner in which users interact with the WebApp. | ■ Use Cases (UCs)  
■ User interface prototypes | ■ UMLet  
■ MS Expression Studio  
■ Visual Paradigm |
| Information model (or, Content Model) | Identifies the full spectrum of content to be provided by the WebApp. Content includes text, graphics and images, and video and audio data. | ■ Content Objects (including, Data Objects)  
■ Data Flow Diagrams (DFDs)  
■ Content Model Trees (CMTs) | ■ Microsoft Visio  
■ IBM Rational Software  
■ SmartDraw |
| Functional Model     | Defines the operations that will be applied to WebApp content and describes other processing functions that are independent of content but necessary to the end user. | ■ State Transition Diagrams (STDs)  
■ Activity Diagrams (ADs)  
■ Sequence Diagrams (SDs)  
■ SwimLane Diagrams (SLDs) | ■ UMLet or SmartDraw  
■ UMLet  
■ SmartDraw |
| Configuration Model  | Describes the environment and infrastructure in which the WebApp resides. | Components:  
■ Hardware, operating systems  
■ Software  
■ Internet, browsers  
■ Data Protocols  
■ Security considerations etc. | ■ UMLet or SmartDraw  
■ MS Visio  
■ IBM Rational Software  
■ SmartDraw |
Diagrams for the Interaction Modeling

- Use Cases (UCs)
- Graphic User Interface (GUI) prototypes

<table>
<thead>
<tr>
<th>WebE Analysis Models</th>
<th>Goal</th>
<th>Diagrams to be used to reflect analysis models</th>
<th>Tools to be used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction Model</td>
<td>Describes the manner in which users interact with the WebApp.</td>
<td>Use Cases (UCs)</td>
<td>UMLet, MS Expression Studio, Visual Paradigm</td>
</tr>
<tr>
<td>Information model (or, Content Model)</td>
<td>Identifies the full spectrum of content to be provided by the WebApp. Content includes text, graphics and images, and video and audio data.</td>
<td>Content Objects (including, Data Objects), Data Flow Diagrams (DFDs), Content Model Trees (CMTs), State Transition Diagrams (STDs), Activity Diagrams (ADs), Sequence Diagrams (SDs), SwimLane Diagrams (SLDs)</td>
<td>Microsoft Visio, IBM Rational Software, SmartDraw</td>
</tr>
<tr>
<td>Functional Model</td>
<td>Defines the operations that will be applied to WebApp content and describes other processing functions that are independent of content but necessary to the end user.</td>
<td>State Transition Diagrams (STDs), Activity Diagrams (ADs), Sequence Diagrams (SDs), SwimLane Diagrams (SLDs)</td>
<td>UMLet or SmartDraw, UMLet, SmartDraw, SmartDraw</td>
</tr>
<tr>
<td>Configuration Model</td>
<td>Describes the environment and infrastructure in which the WebApp resides.</td>
<td>Components: Hardware, operating systems, Software, Internet, browsers, Data Protocols, Security considerations etc.</td>
<td>UMLet or SmartDraw, MS Visio, IBM Rational Software, SmartDraw</td>
</tr>
</tbody>
</table>
1. Use-Cases and Use-Case Scenarios

- It is crucial for WebApp designers and developers to understand users and create users’ profiles.

- Use-Case is a collection of user scenarios that describe the thread of usage of a system by various users.

- Each single scenario is described from the point-of-view of an “actor” — a person or device that interacts with the software in some way.

- Each single scenario should clearly answer the following questions:
  - Who is the primary actor (user), the secondary actor(s)?
  - What are the actor’s goals?
  - What preconditions should exist before the story begins?
  - What main tasks or functions are performed by the actor?
  - What extensions might be considered as the story is described?
  - What variations in the actor’s interaction are possible?
  - What system information will the actor acquire, produce, or change?
  - Will the actor have to inform the system about changes in the external environment?
  - What information does the actor desire from the system?
  - Does the actor wish to be informed about unexpected changes?

These days: Active Use of Tools

Modeling Languages
(to convert manual sketches into electronic diagrams = files)

- A modeling language (ML) incorporates a set of symbols, notations, and/or terms, as well as the rules for establishing associations between them.

- A modeling language often has a formally structured representation as well as a set of graphical elements.

- Some MLs are general purpose (e.g., UML) and others are more specific (e.g., WebML)

- Some examples of MLs:
  - HDM - W2000
  - RMM
  - OOHDM
  - ARANEUS
  - STRUDEL
  - TIRAMISU
  - WebML
  - Hera
  - UML Web Application Extension
  - UML-based Web Engineering (UWE)
  - ACE
  - WebArchitect
  - OOH
2. User Interface (GUI) Prototyping:
A Template-Based Approach
(Dr. Uskov’s NSF CCLI project in 2000-2004)
GUI Template for e-Learning System (an example)

Logo + Title + URL (input)

Video + Audio (output)

VCR buttons (input)

Communications Tools (input + output)

Power Point slides (output)

Slide Notes (output)

GUI of the InterLabs -- Advanced Web-Lecturing System with Streaming Media (Video, Audio, Data) at Bradley University (D&D 2002-2009)

Video/ Audio

Text

PPT slides

Pictures

Web-based Animation

Web-based programming and simulation

Email

Bulletin B.

Chat

Whiteboard

Video-conf.

Audio-conf.
Tools for User Interface (GUI) Prototyping: Design Software-Based Approach

Microsoft Expression

http://www.microsoft.com/expression/default.aspx

http://www.designervista.com/index.html

16 User Interface Prototyping Tools

Available at http://www.dexodesign.com/2008/11/07/review-16-user-interface-prototyping-tools/
Innovative Technology: Active Interface Prototype (examples)

- A prototype shows the layout of the user interface, the content, interaction mechanisms and overall aesthetic.
- Supports validation with the client of the requirements and analysis.

Diagrams for the Information (content) Modeling

- Content Objects (including, Data Objects)
- Data Flow Diagrams (DFDs)
- Content Model Trees (CMTs)
### Analysis Models: Goals and Outputs/Diagrams

<table>
<thead>
<tr>
<th>Analysis Models</th>
<th>Goal</th>
<th>Models or Diagrams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction model</td>
<td>Describes the manner in which users interact with the WebApp.</td>
<td>Use Cases (UCs)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>User interface prototypes</td>
</tr>
<tr>
<td>Information model</td>
<td>Identifies the full spectrum of content to be provided by the WebApp.</td>
<td>Content Objects (including, Data Objects)</td>
</tr>
<tr>
<td></td>
<td>Content includes text, graphics and images, and video and audio data.</td>
<td>Data Flow Diagrams (DFDs)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Content Model Trees (CMTs)</td>
</tr>
<tr>
<td>Functional model</td>
<td>Defines the operations that will be applied to WebApp content and</td>
<td>State Transition Diagrams (STDs)</td>
</tr>
<tr>
<td></td>
<td>describes other processing functions that are independent of content</td>
<td>Activity Diagrams (ADs)</td>
</tr>
<tr>
<td></td>
<td>but necessary to the end user.</td>
<td>Swim Lane Diagrams (SLDs)</td>
</tr>
<tr>
<td>Configuration model</td>
<td>Describes the environment and infrastructure in which the WebApp</td>
<td>Components:</td>
</tr>
<tr>
<td></td>
<td>resides.</td>
<td>o Hardware, operating systems</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Software</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Internet, browsers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Data Protocols</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Security considerations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>etc.</td>
</tr>
</tbody>
</table>

#### 1. Content Objects (particularly, Data Objects), Attributes and Methods

A **content object** contains a set of attributes that act as an aspect, quality, characteristic, or descriptor of the object.

A **content object** might be:
- a textual description of a product,
- an article describing news event,
- an action photograph,
- a user’s response on a forum,
- a video file,
- an audio file,
- a collection of PPT slides, etc.

**Content Object Name:**

**Student**

**Attributes:**
- **first_name**
- **last_name**
- **year of admission**
- **major**
- **courses taken**
- **credits obtained**
- **home_address** etc.

**Methods:**
- **to add to a course**
- **to drop a course**
- **to change major** etc.
Content Objects = Components of the Content Model

- **External entities** (e.g., other systems, databases, people) that produce or consume information to be used by the WebApp
- **Things** (e.g., reports, displays, video images) that are part of the information domain for the problem
- **Occurrences or events** (e.g., a quote or an order) that occur within the context of a user’s interaction with a WebApp
- **Roles** (e.g., retail purchasers, customer support, salesperson) played by people who interact with the WebApp
- **Organizational units** (e.g., division, group, team) that are relevant to an application
- **Places** (e.g., manufacturing floor or loading dock) that establish the context of the problem and the overall function of the WebApp
- **Structures** (e.g., sensors, monitoring devices) that define a class of objects or related classes of objects
2. Data Flow Diagrams: A Context Diagram (with a SINGLE procession unit)

Important notes:
- A single processing unit
- No data storage units
- Multiple data sources (providers of data, receivers of data)

An Example of Data Flow Diagram: Context Diagram of a Learning Management System (like Blackboard or Sakai)

Data Flow Diagrams' Manuals:
- Course web site
- Online
An Example of Data Flow Diagram: Level-0 Data Flow Diagram of a Learning Management System (like Blackboard or Sakai)

A single processing unit has been divided into 2...9 main subsystems (or, systems’ functions)

Data storage units are allowed

Multiple data sources (providers of data, receivers of data)

An Example of Data Flow Diagram: Level-0 Data Flow Diagram of the “University of Missouri – St. Louis Student Registration System”

• A single processing unit has been divided into main sub systems (or, systems’ functions)

• Data storage units are allowed

• Multiple data sources (providers of data, receivers of data)
Web Information Exchange – Notation
(in WebML and UML formats)

Web Information Exchange – An Example
3. Data Trees

- In some cases, the interaction and information model may benefit from a richer analysis using a Trees-based approach.

- **Data trees** depict the relationships among content objects and the hierarchy of content maintained by a WebApp.

- **Menu trees** depict the hierarchical relationships among menu items and/or the hierarchy of objects in a WebApp.

---

**Diagrams for the Functional Modeling (or, Behavioral Modeling)**
### Analysis Models: Goals and Outputs/Diagrams

<table>
<thead>
<tr>
<th>Analysis Models</th>
<th>Goal</th>
<th>Models or Diagrams</th>
</tr>
</thead>
</table>
| Interaction model | Describes the manner in which users interact with the WebApp. | - Use Cases (UCs)  
- User interface prototypes |
| Information model | Identifies the full spectrum of content to be provided by the WebApp. Content includes text, graphics and images, and video and audio data. | - Content Objects (including, Data Objects)  
- Data Flow Diagrams (DFDs)  
- Content Model Trees (CMTs) |
| Functional (behavioral) model | Defines the operations that will be applied to WebApp content and describes other processing functions that are independent of content but necessary to the end user. | - State Transition Diagrams (STDs)  
- Activity Diagrams (ADs)  
- Sequence Diagrams (SDs)  
- SwimLane Diagrams (SLDs) |
| Configuration model | Describes the environment and infrastructure in which the WebApp resides. | Components:  
- Hardware, operating systems  
- Software  
- Internet, browsers  
- Data Protocols  
- Security considerations  etc. |

### Behavioral Modeling

The behavioral model indicates **how WebApp will respond to external events.**

- State transition diagrams (STD) – most popular means of representing a software behavioral model.
- State diagrams are most useful when a user interaction triggers a change in the state of the WebApp—and hence changes the way in which it might react to a user.
- UML *state diagrams* describe dynamic behaviour of the WebApp as an interaction occurs.
Behavioral Modeling

- The behavioral model indicates how software will respond to external events.
- To create the model, the analyst must perform the following steps:
  - Identify events that drive the interaction sequence and understand how these events relate to specific objects.
  - Create a sequence of events for each use-case.
  - Build a state diagram for the system.
  - Review the behavioral model to verify accuracy and consistency.

The States of a Web App

- State - a set of observable circumstances that characterizes the behavior of a WebApp at a given time
- State transition - the movement of WebApp from one state to another
- Event - an occurrence that causes WebApp to exhibit some predictable form of behavior
- Action - process that occurs as a consequence of making a transition
Creating Behavior Models

- **Evaluate all use-cases**
  to understand the sequence of interaction within the WebApp

- **Identify events**
  that drive the interaction sequence and how these events relate to specific objects

- **Create a sequence of events and list of corresponding states**
  or event-trace for each use-case

- **Build a state transition diagram**
  for the WebApp

- **Review the behavior model**
  to verify accuracy and consistency

---

1. State Transition Diagram: An Example

![State Transition Diagram](image-url)
An Example: STD of Automated Teller Machine


An Example: Multi-level STD

Top-level STD in UML format

An STD with details for the “Enrollment” state in UML format

2. Activity Diagrams

- Illustrates the processing flow and logical decisions (algorithms) within the flow.
- The construction details indicate how these operations are invoked, and the interface details for each operation are not considered until WebApp design commences.

Modeling Logic Inside Processing Units in DFDs
3. Processing of Use Cases: Sequence Diagrams or Swim Lane Diagrams

Sequence diagrams describe how user actions collaborate with analysis classes (the structural elements of a system).

Source: http://tynerblain.com/blog/2007/04/10/what-are-use-case-scenarios/
Configuration Modeling

Analysis Models: Goals and Outputs/Diagrams

<table>
<thead>
<tr>
<th>Analysis Models</th>
<th>Goal</th>
<th>Models or Diagrams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction model</td>
<td>Describes the manner in which users interact with the WebApp.</td>
<td>Use Cases (UCs), User interface prototypes</td>
</tr>
<tr>
<td>Information model</td>
<td>Identifies the full spectrum of content to be provided by the WebApp. Content includes text, graphics and images, and video and audio data.</td>
<td>Content Objects (including, Data Objects), Data Flow Diagrams (DFDs), Content Model Trees (CMTs)</td>
</tr>
<tr>
<td>Functional (behavioral) model</td>
<td>Defines the operations that will be applied to WebApp content and describes other processing functions that are independent of content but necessary to the end user.</td>
<td>State Transition Diagrams (STDs), Activity Diagrams (ADs), Sequence Diagrams (SDs), SwimLane Diagrams (SLDs)</td>
</tr>
<tr>
<td>Configuration model *)</td>
<td>Describes the environment and infrastructure in which the WebApp resides.</td>
<td>Components: Hardware, operating systems, Software, Internet, browsers, Data Protocols, Security considerations etc.</td>
</tr>
</tbody>
</table>
The Configuration (Implementation) Model

- On Server side:
  - operating system environments,
  - hardware,
  - Internet,
  - Web servers, data servers,
  - security considerations,
  - access to databases,
  - data protocols,
  - etc.

- On the client side:
  - Local OS
  - Browser software
  - Client hardware variations

Network Configuration Model: An Example (from U of Michigan)

Available at https://www.racf.bnl.gov/terapaths/software/network-hardware/configuration/hardware-configuration-examples/terapaths-bnl-umich-phase-3-network-router-configuration
Network Configuration Model: An Example (from CISCO company)

Lab # 1: Diagramming Tools for Analysis Modeling of Web-Based Systems and Design Modeling

4 REQUIRED tools to learn:
⭐ UMLet or Visual Paradigm,
⭐ IBM Rational Software,
⭐ Microsoft Visio,
⭐ Visual Paradigm,

OPTIONAL: ⭐ SmartDraw – (7-day trial version)
CS593 Midterm Assignment
(see detailed assignment on course web site)

Learn the assigned existing Web system.

Create 8 diagrams for WebE Analysis and Design Models using 4 different diagramming tools, including
♦ UMLet or Visual Paradigm,
♦ IBM Rational Software,
♦ Microsoft Visio,
♦ Visual Paradigm

OPTIONAL: ♦ SmartDraw – (7-day trial version)

Why do we need to know all those diagrams and models, and tools?
I recently had a round of interviews for a position as a Software Engineer. Prior to taking your class, I had no experience in Software Engineering. My expectation for the start of my career was to be a programmer for a few years to gain experience. Going into the interviews for a Software Engineering position, I thought I was probably getting in a little over my head.

Throughout all of the interviews, I was repeatedly asked questions that I was able to answer from what I have learned in your class.

I was not asked a single question about coding. Everything I was asked involved the analysis and design process and I was able to apply my knowledge of Use Cases, Class Diagrams, Data Flow Diagrams, State Transition Diagrams, Entity Relationship Diagrams, etc.

The guys that I interviewed with understood that I'm still a student and I'm no expert, but I was able to speak their language and display my ability to learn and understand what I will need to know as a Software Engineer.

I have since been offered a position as an entry level Software Engineer with the company and I believe that is due in large part to my experiences in your class.

Matt Gihring