Overview on Web Engineering and Web Application Modelling

WEB APPLICATION MODELLING

Mag. Dr. Wieland Schwinger, MSc
wieland.schwinger@jku.at
http://www.tk.uni-linz.ac.at/people/schwinger/
+43 (0)70 / 3343-888
+43 (0)70 / 3343-843

Department of Telecooperation

Outline of the Module 1/2

Overview on Web Engineering
- Categories of Web Applications
- Characteristics of Web Applications
- Web Engineering vs. Software Engineering

Web Application Modelling
- Motivation
- Requirements Framework
- Modelling of Web Applications
- Overview on Existing Modelling Methods
Development of Today's Web Applications

- **Current development** of Web applications, still:
  - ad-hoc implementation
  - tool-driven and technology-driven development
  - little pre-planning

  The "dog house" approach!

- **Consequently,**
  - poor quality of web applications
  - un-fulfilled user requirements
Development of Today's Web Applications

Even, seldomly works for a dog house!

Even for a dog house you should have a plan!

Development of Today's Web Applications

Certainly doesn't work for complex constructions

Plan (=Models) are vital!
Modeling Necessary

- Modeling addresses one of the major problems of today's development: little planning of Web Applications prior to implementation.
- Consequently, modeling necessary for building Web applications:
  - comprehension in its entirety
  - communication among project team
  - insurance of architectural soundness
- But, traditional modeling methods can't be applied directly: some specialized modeling methods exist.

Existing Web Application Modeling Methods

- What is the problem space to cover?
- Which Web application modeling methods exist?
- How do existing Web application modeling methods compare?
- How comprehensive are those Web application modeling methods?
- Pre-requisite: understanding of the requirements

Outline of the Module 1/2

Overview on Web Engineering
- Categories of Web Applications
- Characteristics of Web Applications
- Web Engineering vs. Software Engineering

Web Application Modelling
- Motivation
- Requirements Framework
  - Modelling of Web Applications
  - Overview on Existing Modelling Methods

Requirements of Modeling Traditional Software Applications

- **Roots of Modelling:**
  - Data Engineering - focusing on structural aspects
  - Software Engineering - focusing on behavioral aspects

- **Used Modelling Formalisms:**
  - Entity Relationship Technique (ER)
  - UML 1.x, 2.0 (object-oriented, currently the "lingua franca")
Requirements of Modeling Traditional Software Applications

But *does not regard* one of the major characteristics of Web applications, namely **hypertext**.

---

Requirements Framework for Modeling Web Applications

- **Levels**
  - Presentation
  - Hypertext
  - Content

- **Graphical representation**
- **Logical composition of web pages and the navigation structure**
- **Domain-dependent data**
Requirements Framework for Modeling Web Applications

Levels
- Presentation
- Hypertext
- Content

- Separation of levels
  - strict separation of levels
  - explicit inter-dependencies between levels

- Flexible Mapping
  - mapping as flexible as possible to allow derivation between levels

- Bottom-Up and Top-Down Design
  - bottom-up: starting with the content level (e.g., given database) and derive the hypertext and presentation level
  - top-down: content level is derived from the other levels
Requirements Framework for Modeling Web Applications

Levels
- Presentation
- Hypertext
- Content

Aspects
- Structure
- Behavior

**structure**: user interface elements and their composition

**behavior**: reactions to input events, interaction and synchronization between user interface elements

**structure**: page compositions and navigational relationships

**behavior**: run-time behavior of hypertext

**structure**: domain-dependent data

**behavior**: domain-dependent application logic
Requirements Framework for Modeling Web Applications

- Modeling formalism for structure and behavior
- Standard formalism vs. proprietary formalism
- Uniform formalism vs. divers formalism

Patterns
- facilitate reuse and abstraction of structure and behavior supported by design patterns at all levels
Requirements Framework for Modeling Web Applications

Levels
- Presentation
- Hypertext
- Content

Structure

Aspects
- Behavior

Phases
- Conceptual Modeling
- Logical Modeling
- Physical Modeling
- Implementation

Analysis
Design

towards implementation

analyzing requirements
implementation
independent design
implementation decisions
Requirements Framework for Modeling Web Applications

- Define phases
- Define process
- Characteristics:
  - incremental and iterative
  - prototyping
  - intensive testing
Outline of the Module 1/2

Overview on Web Engineering
- Categories of Web Applications
- Characteristics of Web Applications
- Web Engineering vs. Software Engineering

Web Application Modelling
- Motivation
- Requirements Framework
- Modelling of Web Applications
- Overview on Existing Modelling Methods

Requirements Modeling

- Requirements Engineering techniques to identify, analyze, describe, evaluate, and manage Web application requirements
- **Use cases** preferred modeling technique for functional requirements
  - From the actors’ (people and other systems) perspectives
  - Can be supplemented by UML activity diagrams to describe the functional requirements in more detail
- Peculiarity of Web application requirements is navigation functionality
  - [BaGP01] suggests separating the functional from the navigational use cases, creating two distinct models
  - Another approach, create one single use case model, which uses the UML «navigation» stereotype
- All Web applications have at least one human user, most often anonymous
Content Modeling

- **Content modeling** includes the creation of the problem domain model, consisting of static and dynamic aspects, as known from traditional Software Engineering.
  - Static Web applications: modeling the content in the sense of pure data modeling is normally sufficient
  - Complex Web applications: require the modeling of behavioral aspects.

- **Specifics:**
  - **Document-centric character and multimedia:**
    - consider different media formats when modeling the content, including the structures the information is based on
  - **Integration of existing data and software:**
    - often build on existing data repositories and software components not initially created for Web applications
    - Two potentially contradicting objectives, i.e., it should cover the content requirements of the Web application to the best possible extent, and it should include existing data structures and software components.

Objectives:

- Transferring the information and functional requirements determined by requirements engineering to a content model
- Hypertext character of a Web application and the requirements of its presentation will not be considered

Result:

- a content model that comprising both structural aspects of the content and behavioral aspects (depending on the type of Web application)
  - structural aspects: mainly class diagram,
  - behavioral aspects: e.g. state and interaction diagrams.

Concepts:

- Builds on the concepts and methods of data modeling or object-oriented modeling
- Strives to ensure that existing information is free from redundancies and reusable
Class diagram will later serve as the basis to model the hypertext and the presentation.

W. Schwenger - Model-Driven Development of Ubiquitous Web Applications
Hypertext Modeling

- Non-linearity of hypertext is one of the most important properties to be taken into account when modeling Web applications.
- Hypertext structure has to be designed carefully.
- Can be achieved by using suitable access structures, i.e., navigation options,
  - to avoid the risk of users getting lost
  - to avoid putting users under excessive cognitive stress
- Hypertext modeling also known as navigation modeling.
Hypertext Modeling

- **Objectives:**
  - Specify the navigability through the content of a Web application
    - i.e., the navigation paths available to the users

- **Results:**
  - **Hypertext (Structure) Model (navigation structure model):**
    - defines the structure of the hypertext, i.e., which classes of the content model can be visited by navigation.
  - **Access Model (access structure model)**
    - refines hypertext structure model by access elements
  - The navigational behavior of a Web application is often not represented explicitly, since adding little additional information for the developer - "on activation invoke preceding node"

**Concepts of the Hypertext (Structure) Modeling:**
- based on the concepts of hypertext, i.e., on nodes (also called pages or documents) and links between these nodes
- Specialized notations and concepts are often employed to model the hypertext structure
  - Starting point usually is the content model
    - specified as view on the content model and is therefore sometimes also called navigational view
    - Nodes specified as a view on the content model selecting one or more objects from the content.
    - Links derived from relationships or added explicitly
  - Some other methods model hypertext structure model independently
    - E.g. OOHDM offer a scenarios, where the hypertext structure model can be built directly from the navigational requirements identified by these scenarios
  - In any case, various hypertext structure models that define hypertext "views" on the content.
    - E.g.: Personalized hypertext views for different user groups
Hypertext Structure Modeling

Reviewing System Example

Music Store Example

<<NodeModel>>
MusicHyperbase

using WUML
Various specific types of links to further refine the semantics of the hypertext structure model:

- **HDM (Hypertext Design Model) [GaMP95]**:
  - **Structural links** connect elements of the same node, e.g., from a review summary to the review details.
  - **Perspective links** put various views of a node in relation to each other, e.g., the PostScript and the PDF versions of a paper.
  - **Application links** put different nodes in relation to each other, depending on the application, e.g., a link pointing to “best paper”.

- **WebML (Web Modeling Language) [CFB+03]** (based on the possible transport of information during navigation):
  - **Contextual links** carry context information, e.g., the unique number of a reviewer, to navigate from one reviewer to the reviews he or she created.
  - **Non-contextual links** have no associated context information, e.g., links pointing from a single review to the list of all reviews.

- **WebML** (based on the distribution of nodes on the hypertext level over pages on the presentation level):
  - **Intra-page links** are used when the source and the destination of a link belong to the same page, e.g., when a link allows the user to directly navigate to the summary of a paper, which is displayed further down on the page.
  - **Inter-page links** are used when the source and the destination are on different pages, e.g., when detailed information about the authors and their papers are on different pages.

---

**UWE (UML based Web Engineering) [KoKr02]** (based on the functional requirements):

- **Navigation links** are used to navigate between nodes, e.g., links between papers and their authors.
- **Process links** point to the start node of a process, e.g., to the beginning of the review submission form.
- **External links** point to a node not directly belonging to the application, e.g., to the formatting guidelines established by the publisher of the conference proceedings, which are not directly stored in the reviewing system.

**OO-H (Object-Oriented Hypermedia) modeling method [GaCG05]**:

- **R-links** point to a start of a navigational path to meet a functional system requirement, e.g., to add a new review.
- **T-links** point to internal nodes, e.g., from an author to his or her papers.
- **E-links** point to external nodes, e.g., to external formatting guidelines.
- **S-links** point to services, e.g., to an external search engine.
The hypertext structure model is not sufficient to describe how nodes can be reached by navigation.

**Access structures** refining the hypertext structure model

- **Recurring access structures** ([GeCo00, LyRS99, RoSL98, AkGe05]), also called “hypermedia design patterns” or “navigation patterns” supported by dedicated modeling elements:
  - **Index**: access structure which allows users to select a single object (i.e. one object of the content) out of a homogeneous list of objects
  - **Menu**: allows users to access heterogeneous nodes, or further menus (i.e. submenus)
  - **Guided tour**: allows users to sequentially walk through a number of nodes
  - **Query**: allows users to search for nodes and directly access them
  - **Home**: points to the home page of a Web application
  - **Landmark**: points to a node that can be reached from within all nodes

- Users need navigation and orientation aids.
- Navigation patterns help to increase the quality of the hypertext model tremendously.
- Some of these access structures can be added to the hypertext structure model automatically ([KoKn02])
  - For example, indexes can be added automatically whenever we want to allow access to a set (> 1) of objects of a node

---

**Access Structure Modeling**

- **Access Model**
  - **Index**: AcceptedPapers
  - **Index**: RejectedPapers
  - **Index**: Reviews
  - **Index**: Authors
  - **Index**: ReviewingStatus
  - **Index**: PaperMenu
  - **Index**: Paper
  - **Index**: PapersByTitle
  - **Index**: PapersByID
  - **Index**: AssignedPapers
  - **Navigation class**: Conference
  - **Navigation class**: Paper
  - **Navigation class**: Review
  - **Navigation class**: User
  - **Navigation class**: SearchPaperByTitle

---

**Reviewing System Example**

- **Using UWE (Class Diagram)**
Interrelationship between content modeling and hypertext modeling

- Depending on the underlying modeling method, the hypertext model is more or less strongly dependent on the content model.
- Dependencies:
  - **Type level**, e.g., which classes in the content model form which node in the hypertext model.
  - **Instance level**, i.e., which sets of objects in the content model populate that node in the hypertext model.
- Not all methods describe dependencies between the content model and the hypertext model exactly.
- Some methods specify a direct derivation of the hypertext from the content by defining nodes on the basis of views (in the sense of “database views”) [ScGr02, KoKr02].
Similar to traditional Software Engineering,
- Deals with the user interface and thus with the look and feel of a Web application

In contrast to traditional applications,
- Central element of the presentation in Web applications is the page as a visualization unit
Presentation Modeling

- **Objectives:**
  - Designing the structure and behavior of the user interface to ensure that interaction with the Web application is **simple** and **self-explanatory**
  - Communication and representation task of the Web application are taken into account

- **Results:**
  - a **uniform presentation concept** by modeling recurring elements on the pages, e.g., headers and footers. It should ideally show the composition of each page and the design of the fields, texts, images, forms, etc., included in these pages.
  - description of the **behavior-oriented aspects of the user interface**, e.g., which button to click to activate a function of the application logic.
  - Give users appropriate orientation help on the presentation level
  - due to the wide variety of navigation options and
  - the inherent risk of getting lost
  - Can be achieved, e.g., by displaying the current navigation path, or pages visited during the active session

**Concepts:**

- Not all methods **support technology-independent presentation modeling concepts**
- Some rather use technology-specific concepts, such as Stylesheet languages, e.g., XSL (Extensible Stylesheet Language)
- Important factor for Web applications is the **graphical layout design** of the user interface
  - often produced by a graphic designer based on some basic drawings, or conceptualized by the tool-supported implementation of **prototypical pages**
  - Although this task is part of presentation modeling, it is **currently not supported** by modeling techniques.
Presentation Modeling

- Model elements are described on three hierarchical levels:
  - **Presentation page**: a page presented to the user as a visualization unit. It can be composed of different presentation units.
  - **Presentation unit**: serves to group related user interface elements, representing a logical fragment of the page. It presents a node stemming from the hypertext model.
  - **Presentation element**: basic building block of the presentation model. Presentation elements represent a node's set of information and can include text, images, audio, etc.
Presentation Modeling

Using UWE (Sequence Diagrams)

Presentation Modeling

Music Store Example

Using WUML
Presentation Modeling

Example

W. Schwenger - Model-Driven Development of Ubiquitous Web Applications 62

Music Store Example

W. Schwenger - Model-Driven Development of Ubiquitous Web Applications 62
Presentation Modeling

- Similarly to mapping the content model to the hypertext model
  - specify how hypertext elements should be mapped to presentation elements.
  - normally done under the assumption that all instances of a node will be displayed on the presentation level.
- Interactions triggered by a user are not necessarily limited to the presentation level only
  - objects and application logic on the content level, and for navigation on the hypertext level.

Outline of the Module 1/2

Overview on Web Engineering
  - Categories of Web Applications
  - Characteristics of Web Applications
  - Web Engineering vs. Software Engineering

Web Application Modelling
  - Motivation
  - Requirements Framework
  - Modelling of Web Applications
  - Overview on Existing Modelling Methods
Some Important Modeling Methods and their Genesis

Modeling Methods

**Araneus** *(Atzeni et al. 1998)*
content and hypertext level are refined independently from each, refinement of models

**HDM** *Hypermedia Design Method (Garzotto et al., 1997)*
hypertext systems focused, separates between design in the small and in the large

**HDM lite** *Autoweb project (Fraternali, 2000)*
automation of the development process and with automatic generation of Web applications in mind

**HERA** *(Houben et al. 2004)*
recent behavioural extension based on RMM

**OO-H** *Object-Oriented Hypermedia Method (Gomez et al., 2003)*
more recent methods, combining the benefits of WebML, OOHDM, and UWE

---

Some Important Modeling Methods and their Genesis

Modeling Methods

**OOHDM** *Object-oriented Hypermedia Design Method (Rossi et al., 1999)*
emphasizes the concept of navigation context

**OOWS** *Object-Oriented Web Solutions (Pastor et al. 2005)*
concepts of the UML, but it mainly propose their own notation

**RMM** *Relationship Management Methodology (Isakowitz et al., 1998)*
defines M-Slices over ER, gradual process for the successive refinement of models

**UWE** *UML-based Web-Engineering (Baumeister et al., 1999)*
UML-based notation and meta-model based model consistency checking
Some Important Modeling Methods and their Genesis

**Modeling Methods**

- **WAE2** - Web Application Extension (Conallen 2004)
  a UML approach that focuses on the distribution of the application logic

- **WebML** - Web-Modeling Language (Ceri et al., 2003)
  easy-to-understand and mature modeling language for data-intensive Web applications

- **W2000** - W2000 (Baresi et al. 2001)
  UML-like refinement of HDM, hypertext-centric and user-centric perspectives

- **WSDM** - Web Site Design Method (De Troyer et al.)
  methodic approach oriented towards user requirements

- **WebSA** - WebSA Web Software Architecture (Meliá et al. 2005)
  an approach for modeling Web architectures

W. Schwenger - Model-Driven Development of Ubiquitous Web Applications
Modeling Method Examples
OOHDM

Navigation Schema

Rossi et al., 1999

Modeling Method Examples
OOHDM

Context Schema

Rossi et al., 1999
Modeling Method Examples

RMM

M-Slice Diagram

- **article**
- **abstract**
- **title**
- **written_by**
- **contributor**
- **name**
- **bib_citation**

Isakowitz et al., 1998

W. Schwenger - Model-Driven Development of Ubiquitous Web Applications 71

Modeling Method Examples

UWE

Navigational Structure Model

- **mainMenu**
- **searchByNameIndex**
- **departmentIndex**
- **projectIndex**
- **department by name**
- **EmployeeContext**
- **by name**
- **by department**
- **search by name**

Baumeister et al., 1999

W. Schwenger - Model-Driven Development of Ubiquitous Web Applications 72
Overview on Existing Modeling Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>UML Support</th>
<th>Generation</th>
<th>Tool Support</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCMLite</td>
<td>HT</td>
<td>ER + own notation</td>
<td>auto</td>
<td>process for model transformation, automatic generation</td>
</tr>
<tr>
<td>HeraDB</td>
<td>ER + RDB, own notation</td>
<td>semiauto</td>
<td>authoring &amp; generation tool</td>
<td>model-driven development</td>
</tr>
<tr>
<td>OD-H</td>
<td>UML + own notation</td>
<td>semiauto</td>
<td>modeling &amp; generation tool</td>
<td>tool for automatic generation</td>
</tr>
<tr>
<td>DOORD</td>
<td>UML + own notation</td>
<td>semiauto</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOWS</td>
<td>UML + own notation</td>
<td>semiauto</td>
<td>modeling &amp; generation tool</td>
<td>advanced (commercial) tool for automatic generation</td>
</tr>
<tr>
<td>JRM</td>
<td>ER + own notation</td>
<td>semiauto</td>
<td>authoring tool</td>
<td>hybrid modeling based on ER model, predefined process</td>
</tr>
<tr>
<td>UWE</td>
<td>UML</td>
<td>RUP</td>
<td>extended UML tool &amp; generation tools</td>
<td>semiauto</td>
</tr>
<tr>
<td>W3E020</td>
<td>HT</td>
<td>UML</td>
<td>extended UML tool</td>
<td>user-centric hybrid modeling</td>
</tr>
<tr>
<td>W3E022</td>
<td>SW</td>
<td>UML</td>
<td>extended UML tool</td>
<td>implementation design, architectural design</td>
</tr>
<tr>
<td>W3M</td>
<td>UML</td>
<td>RUP</td>
<td>standard UML tools</td>
<td>user-centric approach for analysis</td>
</tr>
<tr>
<td>W3SM</td>
<td>UML</td>
<td>RUP</td>
<td>standard UML tools</td>
<td>user-centric approach for analysis</td>
</tr>
</tbody>
</table>

Tool Support

- **WebRatio Site Development Studio**
  - Commercial model-based development tool
  - WebML as base language
  - Code generator:
    - uses XSL to transform content and hypertext models represented in XML into the required database representation and database connections as well as software components and different output formats (HTML, WML, PDF, Microsoft Word)
  - Tool called EasyStyle to generate the presentation of pages, which will transform annotated pages into XSL stylesheets automatically
  - Runtime framework
    - based on a set of Java components, which can be configured by use of XML files.
    - suited for the Jakarta Struts open-source platform and for JSP tag libraries
  - www.webratio.com
Tool Support

- **VisualWADe**
  - Commercial model-based development tool
  - Based OO-H method
  - Supports modeling and automatic generation of applications based on XML, ASP, JSP, and PHP.
  - Augments a UML model with two additional models:
    - "Navigation View" is used to model the hypertext aspects
    - "Presentation View" represents interaction elements of the user interface with regard to its structure and behavior using a number of template structures (device-independent description)
  - Generation of the Web application for different runtime environments and devices possible
  - www.visualwade.com

- **OpenUWE Tool Suite**
  - Open architecture approach based on established standards
  - Realizes UWE approach
  - Based on the open source CASE tool ArgoUML
  - Includes:
    - model consistency checker,
    - layout editor,
    - code generators
  - The common data exchange language within this architecture is based on the extensible UWE meta-model.
  - still under development
  - www.pst.ifi.lmu.de/projekte/uwe
Recent and Future Developments in Web Application Modeling

- **Behavioral modeling** in terms of workflow and transaction support
- **Adaptivity** in terms of personalization and context-awareness
- **Web Services** integration
- **Model-Driven Development (MDD)** as required by OMG's Model-Driven Architecture (MDA) approach
- **Mobile Web applications** support
- **Aspect oriented modeling** of Web applications

Take Home Message

1. **Modeling of Web applications** essential prior to their realization
2. Consider structure and behavior for content, hypertext and presentation
3. A series of methods exists mainly focusing on hypertext and content/hypertext modeling
4. Easiness of application along with tool support vital for acceptance
Thank you for your attention!

Literature


---


