UML-based Web Engineering

Nora Koch

Web Engineering Group
Ludwig-Maximilians-Universität München (LMU)
Cirquent GmbH
Germany
Seville, 27.04.2010
Web Engineering

- Software Engineering for the Web domain
- Concept introduced by San Murugesan in 1998

Definition

- Web engineering is the application of a systematic and quantifiable approach to cost-effective requirements analysis, design, implementation, testing, operation, and maintenance of high-quality Web software.
- Web engineering is also the scientific discipline concerned with the study of these approaches.

*Gerti Kappel, Birgit Pröll, Siegfried Reich, Werner Retschitzegger
   Web Engineering (2006)*
UML-based Web Engineering (UWE)

- Main characteristic is the use of **UML** for all models
- Use of other OMG standards, such as MDA, MOF, OCL, XMI, …
- Focuses on **systematisation** and **automatic generation**
- UWE comprises
  - a **modelling language** for the graphical representation of models of Web applications
  - a **metamodel** for UWE modelling elements
  - a **development process**
  - **tools** supporting (semi-)automatic generation
- UWE web site
  - examples
  - **tutorial**
  - **tools** for downloading

[uwe.pst.ifi.lmu.de](http://uwe.pst.ifi.lmu.de)
Web Engineering Group of LMU

- **Current staff**
  - Martin Wirsing (head of department)
  - Alexander Knapp (currently at University of Augsburg)
  - Nora Koch (leader)
  - Gefei Zhang and Christian Kroiß (PhD students)
  - Marianne Busch (student)

- **Projects**
  - UWE (no funding)
  - MAEWA I and II
  - EU projects of FP6 and FP7

- **Cooperations**
  - Universities of Alicante, Extremadura, Málaga, Sevilla (ES), Milano (IT), Viena & Linz (AT), LaPlata (RA)
  - Industry: S.Co LifeScience, Siemens, ...

- **Other activities**
  - ICWE conferences since 2003
  - MDWE workshops since 2005
Outline of the Talk

- Engineering Web Software
- UWE Approach
  - Modelling Language
  - Metamodel
  - Development Process
  - Tool Support
- Outlook
Web Software

- **Web Information System (WIS)**
  - software system based on client/server technology provides information through a user interface (Web browser). Pages belong to a particular domain name or subdomain on the World Wide Web

- **Web Application (WA)**
  - software system that provides Web specific resources such as content and services through a Web browser

- **Rich Internet Application (RIA)**
  - web application, which use data that can be processed both by the server and the client. Furthermore, the data exchange takes place in an asynchronous way so that the client stays responsive while continuously recalculating or updating parts of the user interface.
  
  On the client, RIAs provide a similar look-and-feel as desktop applications and the word "rich" means particularly the difference to the earlier generation of web applications.

- **Web service (WS)**
  - software system designed to support interoperable machine to machine interaction over a network. Web services are frequently just application programming interfaces (API) that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services

1W3C, www.w3.org
Categories of Web Software

- document based
- interactive
- transactional
- workflow-based
- collaborative
- ubiquitous
- semantic web
- rich interaction

Based on Gerti Kappel, Birgit Pröll, Siegfried Reich, Werner Retschitzegger: Web Engineering, (2006)
Web Specifics in Engineering

- **Hypermedia paradigm**
  - nodes & links
  - text & multimedia

- **Omnipresent due to the nature of the Web**
  - global and permanent availability
  - comfortable and unified access
  - distributed information / services

- **Dynamic development**
  - incremental number of Web pages
  - continuous improvement of existing Web applications (content, links, layout)
  - offer of new services
  - adaptation required by new Web technologies

- **Management aspects**
  - multidisciplinary development team
  - inhomogeneous and immature technical infrastructure
  - short product lifecycles → short development cycles

Specific engineering methods for the Web domain
# Web Engineering Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDM / HDM-lite</td>
<td>Hypertext Design Model</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; method for WE</td>
</tr>
<tr>
<td>Hera</td>
<td>Navigational Development Technique</td>
<td>semantic web requirements</td>
</tr>
<tr>
<td>NDT</td>
<td>Object-Oriented Hypermedia Method</td>
<td>improved navigation</td>
</tr>
<tr>
<td>OO-H</td>
<td>Object-Oriented Hypermedia Design Method</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; OO-method</td>
</tr>
<tr>
<td>OOHDM</td>
<td>Object-Oriented Hypermedia Design Method</td>
<td>compilation of models</td>
</tr>
<tr>
<td>OOOWS</td>
<td>Object-Oriented Web Solution</td>
<td></td>
</tr>
<tr>
<td>RMM</td>
<td>Relationship Management Methodology</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; ER-based method</td>
</tr>
<tr>
<td>RUX</td>
<td>UML-based Web Engineering</td>
<td>generation of RIAs</td>
</tr>
<tr>
<td>UWE</td>
<td>UML-based Web Engineering</td>
<td>UML-based/standards</td>
</tr>
<tr>
<td>WAE / WAE2</td>
<td>Web Application Extension</td>
<td>implementation-oriented</td>
</tr>
<tr>
<td>WebML</td>
<td>Web Modeling Language</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; generation tool</td>
</tr>
<tr>
<td>WebSA</td>
<td>Web Software Architecture</td>
<td>architecture models</td>
</tr>
<tr>
<td>WSDM</td>
<td>Web Site Design Method</td>
<td>user-centered design</td>
</tr>
</tbody>
</table>
UWE focuses on Modelling

- UML-based Web Engineering (UWE) approach
  - based on UML models
  - separation of concerns (content, navigation, presentation, process, …)
  - UML extension (profile) for web specific notation
  - model-driven development
    - model2model transformations
    - model2code transformations
Why UML?

- UML is a graphical language for specifying, constructing and documenting software artifacts
- UML is a de facto industry standard and an OMG standard
- UML includes:
  - notation
  - diagram types
  - Object Constraints Language (OCL)
  - metamodel
  - well-formedness rules
- Many CASE tools supporting UML modelling
- UML does not provide a development process

- How expressive is UML for the development of Web applications?
  - UML does not include specific Web model elements
  - UML defines extension mechanisms → UML profiles
UML Extensions

- **Light weight extension**
  - called a UML profile
  - based on extension mechanisms provided by UML
  - defines stereotypes for new metaclasses
    - domain specific: EJB «bean», «session», «entity», …
    - defined in the UML: «metaclass», «trace», «file», …
  - tagged values for metaattributes
  - OCL constraints for invariants, pre- and postconditions
  - CASE tool support by UML tools

- **Heavy weight extension**
  - different notation
  - other diagram types not defined in the UML
  - need of proprietary CASE tool
  - long learning process
Dimensions of Web Modelling

**aspects**

- Modelling process
  - information-driven ("content first")
  - presentation-driven ("layout first")
  - functionality-driven ("business processes first")

**concerns**
- presentation
- hypertext
- content

**structure**
- requirements analysis
- design
- implementation

**behaviour**

**adaptivity**

**phases**
- implementation
- design
- analysis
- requirements

---

Requirements Analysis and Design Models

- **Analysis models of a Web application / RIAs**
  - functional requirements are specified by
    - use cases
    - workflows
  - data (content) requirements are specified by
    - domain models

- **Design models of a Web application / RIAs**
  - information aspects
    - content model
  - hypertext structure and navigation functionality
    - navigation model
  - layout schema
    - presentation model
  - functionality
    - process model
    - adaptivity model
Example: Simple Music Portal

- Inspired by www.mp3.com
  - contains information about albums, songs and artists
  - this information is available for free
  - offers albums for downloading
  - registered users can download them
  - for downloading they need to have enough credit on their prepaid account
  - accounts are rechargeable
Modelling Requirements with UWE

- Graphical visualization by UML use case diagram
  - to model required functionality
  - distinguishes between navigation and process use cases
- Web specific model elements (not yet implemented in MagicDraw)
  - «navigation» use cases for browsing tasks
    - browse use cases
    - view use cases
    - search use cases
  - «webProcess» use cases for other tasks
  - «personalized» use cases that require adaptation
Content Modelling

- Representation of domain information
  - persistent data

- Modelling technique
  - UML class diagram
  - plain UML
  - no additional semantics required
User Model

- Represented as UML class diagram
  - “normal” UML classes
  - Visit object for each session
    - «visitClass»
- Representation of session and user specific information
  - allows for customization
Navigation Modelling

- **Goals**
  - to represent nodes and links of the hypertext structure
  - to design navigation paths

- **Navigation model**
  - represented by a UML class diagram
  - uses specific modelling elements for Web concepts

- **Basic elements to model the core hypertext structure**
  - «navigationClass» specifies a hypertext node visited by a user/system through browsing (related to a content class)
  - «navigationLink» specifies a hyperlink used to access the target navigation object from the source navigation object
Access Primitives in the Navigation Structure

- Systematic enhancement of the navigation structure model by
  - «index» for all navigation links which have multiplicity > 1 at the directed association end
  - «menu» for all navigation classes with more than 1 outgoing association
- Design decision to include
  - «guidedTour» instead of index
  - «query» for selection of instances of a navigation class
- Shortcuts for more complex constructs (if represented in UML without extension)
Navigation Model Elements: Menu

- **Menus** are used to structure the outgoing links from a node
  - usually associated to a navigation class by composition
  - consists of a set of links to heterogeneous elements, such as indexes, guided tours, queries, instances of navigation classes or other menus
- UML stereotype: «menu»

- Semantics of menu
Use of Navigation Elements

- Tagged values
  - `{isHome}` to indicate that the node is the initial node of the application (node without incoming links)
  - `{isLandmark}` to indicate that a node is reachable from everywhere (all other nodes include a link to the landmark node)
Navigation Model
(excerpt)
Modelling Processes in UWE

- **Process model**
  - represent the dynamic aspects of a Web application
  - specifies functionality, such as transactions and complex workflows of activities

- **Process modelling consists of**
  - definition of *process classes* (for *non-navigation use cases*)
  - integration of these process classes in the *navigation model*
  - description of the behaviour through a *process flow*
Process Elements

- **Process class** represents the process through which the user will be guided in the Web application
  - for complex process that require more than a single class, an additional process model is built
  - UML stereotype: «processClass»

- **Process link** is used to model the association between a «navigation class» and a «process class»
  - indicates entry points and exit points of processes within the navigation structure
  - UML stereotype: «processLink»
Seville – 2010 – Nora Koch
Modelling the Process Flow

- The behavior of a Web process is defined by the process flow model
  - represented by UML activity diagram
  - result of the refinement of the activity diagram drawn for requirements specification
  - «processClass» stereotype
  - optional use of nested activity diagrams

- Process flow consists of
  - flow of execution represented by activity nodes connected by activity edges
  - control nodes that provide flow-of-control constructs, such as decisions and synchronization
  - object nodes that represent data flowing along object flow edges or pins associated to the actions
  - in UML2 the semantic of activities is based on control and data token flows, similar to Petri nets
  - using stereotypes «userAction» and «systemAction»
Process Model: Login
Modelling Presentation

- Representation of layout for the underlying navigation and process models
  - is an abstract presentation
  - concrete presentation requires specification of additional physical properties of the layout
    - colour, position, …
    - # of columns in table, type of menu, …

- Presentation classes represent Web pages or part of pages
  - composition of user interface elements
  - hierarchical composition of presentation elements

- UML class diagram for the structure of the presentation
  - using UML container notation

- UML state charts
  - used for modelling behaviour of presentation classes (classical UML)

- Alternatives
  - sketches
  - implementation of a prototype
Presentation Model Elements

- **Structural presentation elements**
  - «presentationGroup»: container of user interface elements representing a logic unit of presentation associated to a navigation class or process class
  - «presentationPage»: presentation class at highest level
  - «presentationAlternatives»: container for presentation classes which are not shown simultaneously

- **User interface elements**
  - «anchor»
  - «button»
  - «text»
  - «image»
  - «textInput»
  - «selection»
  - …
Modelling Rich User Interfaces

- “Rich Internet Applications (RIAs) is about giving users back what they lost when desktop applications were migrated to the Web”
  - rich look & feel, e.g. drag&drop facilities
  - better responsiveness avoiding e.g. page reloading
  - improved performance and accessibility

- We follow a pattern-based approach for modelling RIA features
  - auto completion
  - live report
  - live validation
  - live warning
  - periodic refresh
  - ...

"Rich Internet Applications (RIAs) is about giving users back what they lost when desktop applications were migrated to the Web”

- rich look & feel, e.g. drag&drop facilities
- better responsiveness avoiding e.g. page reloading
- improved performance and accessibility

We follow a pattern-based approach for modelling RIA features
- auto completion
- live report
- live validation
- live warning
- periodic refresh
- ...
Auto completion

- **Problem**
  - How can the user get immediate suggestion for values in an input field?

- **Motivation**
  - provide assistance filling in forms
    1. people make mistakes when they type
    2. typing is a tedious work
    3. typing speed remains a bottleneck

- **Solution**
  - suggest words or phrases that are likely to complete what the user is typing
  - as soon as the user moves to another input element or user inputs a character
  - the user can always overwrite the suggestion

- **Patterns**
  - modelled by state machines
Embedding RIA Patterns in existing Web methods

- RIA patterns are independent of any Web engineering approach

- Our approach consists of
  - definition of extension points in the methodology for including references to RIA functionalities
  - use of already defined state machines
  - definition of transformation rules that automatically integrate the behavior defines in the state machines into the models or code of the Web application (model-driven approach only)

- RIA developer will use
  - catalog of models of RIA widgets
  - specific model elements for indicating in the Web application models which pattern should be applied in which case
  - event language for the representation of triggering user and system actions

- No need for developers to model the behavior of the RIA widgets every time!
Pattern-based approach for RIAs development

- Our approach consist of
  - modeling the RIA widgets with state machines
  - building a RIA patterns catalog
  - selecting needed constructs of the event language
  - defining modeling language extension to embed RIA specific features

- Patterns are design solutions that could be reused in the development of applications
  - manually, i.e. copy/paste or import function in CASE tool
  - in automated development process, such as by model transformations in model-driven engineering

- Advantages
  - minimalist and concise extension of the modeling language
    - use of tags at model level
  - reducing implementation efforts
  - making applications less error-prone
Modelling Concrete Presentation

- “Abstract” presentation element in PIM describes functionality
  - e.g.: «selection»

- **Question:** how will it be realized in the generated application?
  - HTML: list, combo box, radio buttons, etc.
  - DHTML, flash applet

- “Concrete” presentation requires additional information
  - «concreteElementType» added
  - dependency to related abstract presentation element
Presentation Mapping Model

Main Idea

- include model of **component library**
- Allows integration of comp. libraries like Apache MyFaces Tomahawk, etc.
- Allows integration of custom components
Adaptive Web Applications

- Adaptation/Customization for
  - user properties: knowledge, tasks, preferences, interests
  - context properties: location (place and time) & platform (HW, SW, network)

- Update of a user model / context model
  - observation of the user behaviour or environment by the system

- Techniques for adaptation
  - **content** adaptation
    - inserting and removing text/multimedia features
    - content variants
  - **navigation** adaptation
    - link ordering
    - link annotation
    - link hiding
    - link generation
  - **presentation** adaptation
    - modality adaptation (audio or text)
    - language selection
    - layout variants (resizing of fonts, images, changing colours)
Modelling Adaptivity

- UWE uses a technique called Aspect-Oriented Modelling (AOM)
  - Identification of
    - «pointcut» (including conditions)
    - «advice»
  - Weaving the result into the web application based on
    - current state of the user model
    - information provided by link traversal
  - Example: links only visible for registered users to
    - BuyAlbum
UWE Metamodel

- UWE Metamodel is defined as a conservative extension of UML 2.0
  - model elements of the UML metamodel are not modified
  - all new elements are related by inheritance to at least one model element of the UML
  - use of OCL to specify additional semantics of the new elements
  - so-called light-weighed extension of UML
UWE Metamodel Characteristics

- UWE metamodel
  - reflects separation of concerns in the structure of Core
  - shows cross-cutting aspect of adaptation

- UWE metamodel is profileable
  - mapping to a UML profile is possible

- UWE metamodel is MOF compatible
  - uses XML metadata interchange format (XMI)
UWE Metamodel: Navigation Package

Class

NavigationNode
+isLandmark : Boolean
+isHome : Boolean
-guard : String [0..1]
-dataExpression : String [0..1]

Association

Link
-isAutomatic : Boolean
-guard : String [0..1]
-selectionExpression : String [0..1]

ExternalNode
-locationExpression : String [1]

Menu
-menus * 0..1

NavigationClass

NavigationLink

AccessPrimitive
-accessedAttributes *

NavigationProperty
-selectionExpression : String [0..1]

Property

Query
-expression : String [0..1]

GuidedTour
-sortExpression : String [0..1]

Index

IndexRowProperty

Development Process Tool Support
UWE Metamodel: Presentation Package
Mapping Profile to Metamodel

- Definition of stereotypes for each metamodel element
UWE Profile: Extension Mechanisms

Process

- UML stereotypes for Web specific concepts (specification of the process)
- Extends relationships
- UML metaclasses

```
package UWE Profile

<metaclass>
Class

<stereotype>
navigationNode

<stereotype>
processClass

- processProperty : processProperty [*]
- processActivity : Activity [0..1]

<metaclass>
Association

<stereotype>
link

<metaclass>
Property

<stereotype>
navigationProperty

<stereotype>
processLink

- processClass : processClass [1..*]

<stereotype>
processProperty

- rangeExpression : String [0..1]
- processClass : processClass [1]
- indicatedProgress : Boolean = false

<metaclass>
Action

<stereotype>
userAction

- processClass : String [1]

<stereotype>
CallAction

<stereotype>
userAction

<stereotype>
systemAction
```

Seville – 2010 – Nora
Dialects of UWE Modelling Language

- **UWE or UWE/Open**
  - serves modellers interested in having maximum freedom
  - used as graphical language for communication between people
  - focus lies on readable diagrams
  - focus on overview of structure and behaviour
  - no restriction on how UML elements and UWE elements may be used and combined in models

- **MDUWE or UWE/Strict**
  - serves modellers interested in code generation
  - code generation requires models
    - which semantics is specified more precisely
    - contains more details
    - follow strict rules
  - allows for the application of transformation rules

model-driven development approach
Model-Driven Development

- MDD approaches based on
  - models, metamodels and model transformations
- MDD approaches require languages for
  - specification of models
    - UML, BPMN, …
  - description of metamodels
    - UML, MOF, OCL, …
  - definition of model transformations
    - Java
    - Graph transformations
    - ATL, QVT, …
- Model-Driven Architecture (MDA)
  - computational independent model (CIM)
  - platform independent model (PIM)
  - platform specific model (PSM)
Model Transformations

- Goal is automatic translation between source and target models
- Translation performed by a transformation engine that executes transformation rules
- Set of rules
  - seen as a model
  - based on a transformation metamodel
- Metamodels are based on a metametamodel
- MDA model transformations
  - CIM2PIM
  - PIM2PIM
  - PIM2PSM

**Model transformation pattern (J. Bézivin, 2004)**
Several methods propose building models
  - Hera, MIDAS, NDT, OOHDM, OO-H, OOWS, UWE, WebML, …
  - separation of concerns
    - content
    - navigation
    - presentation
    - business processes
    - adaptation, …
  - similar Web specific modeling elements
  - different notations (UML-based and own notations)

Some methods define metamodels for modelling languages
Some approaches address model transformations
Goal of almost all
  - Platform specific models in a late development stage
Model-Driven Process of UWE

- MDD types of models in UWE
  - requirements model (CIM)
  - functional models (PIM)
    - content model
    - navigation model
    - ...
  - integration models (PIM)
  - models for JSF applications (PSM)

- Support
  - for modelling: CASE tool (plugins for ArgoUML, MagicDraw)
  - for generation: Eclipse and set of ATL-based transformations

Note: ArgoUML does not support UML2
ArgoUWE not updated
Support for Construction of Models

- We can use any UML CASE tools to design Web applications with UWE
  - importing the UWE profile
  - defining the stereotypes and tagged values
- Case tool MagicUWE
  - extension of MagicDraw
  - provides UWE Profile
  - supports (semi-)automatic execution of transformations
Transformation Content to Navigation

- Content2Navigation
  - generates navigation classes from content classes
  - adds a navigation links based on associations of the content model

- Implementation
  - MagicUWE plugin in Java
  - Generation tool: ATL
Refinement of Navigation Model

- Improvement based on patterns
  - index for associations with multiplicity greater than one at the directed association end
  - menu for navigation classes with multiple outgoing associations

- Implementation
  - Java in MagicUWE
  - ATL in generation tool
## Model Transformation Languages

Languages used in different versions of UWE

<table>
<thead>
<tr>
<th>Characteristics Transformation</th>
<th>Type</th>
<th>Complexity</th>
<th>Marks</th>
<th>Execution</th>
<th>Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req2Content</td>
<td>CIM to PIM</td>
<td>simple</td>
<td>WebRE profile</td>
<td>automatic</td>
<td>QVT</td>
</tr>
<tr>
<td>Req2Architecture</td>
<td>CIM to PIM</td>
<td>simple</td>
<td>-</td>
<td>manual</td>
<td>-</td>
</tr>
<tr>
<td>Content2Navigation</td>
<td>PIM to PIM</td>
<td>simple</td>
<td>navigation relevance</td>
<td>semi-automatic</td>
<td>Java, ATL</td>
</tr>
<tr>
<td>NavigationRefinement</td>
<td>PIM to PIM</td>
<td>simple</td>
<td>UWE profile &amp; patterns</td>
<td>automatic</td>
<td>Java</td>
</tr>
<tr>
<td>Req2Navigation</td>
<td>CIM to PIM</td>
<td>merge</td>
<td>WebRE profile</td>
<td>automatic</td>
<td>QVT</td>
</tr>
<tr>
<td>Navigation2Presentation</td>
<td>PIM to PIM</td>
<td>simple</td>
<td>UWE profile</td>
<td>automatic</td>
<td>Java, ATL</td>
</tr>
<tr>
<td>Functional2BigPicture</td>
<td>PIM to PIM</td>
<td>merge</td>
<td>patterns</td>
<td>automatic</td>
<td>graph transformations</td>
</tr>
<tr>
<td>Functional&amp;Architecture2Integration</td>
<td>PIM to PIM</td>
<td>merge</td>
<td>UWE &amp; WebSA profile</td>
<td>automatic</td>
<td>QVT-P</td>
</tr>
<tr>
<td>Integration2J2EE</td>
<td>PIM to PSM</td>
<td>merge</td>
<td>patterns</td>
<td>automatic</td>
<td>QVT-P, ATL</td>
</tr>
</tbody>
</table>
Tool Support for UWE

- **Goals**
  - support communication between stakeholders (customer, developer, designer,…)
  - support model-driven development process

- **Requirements**
  - support of **UWE notation** for design of Web applications and RIAs
  - separation of concerns (navigation, process, presentation,…)
  - (semi)-automatic code generation
  - support **model validation**, i.e. checking models consistency
MagicUWE Plugin supports

- **UWE profile**
  - UWE stereotyped elements in toolbar
  - stereotype icons

- **UWE diagrams and package structure**
  - creation of new content, navigation, presentation, process models

- **Transformation between UWE diagrams**
  - content to navigation
  - navigation to presentation
  - navigation to process structure and process flow

- **Set tagged values using the context menu**
  - isHome, isLandmark

- **Insert of access primitives**
  - e.g. add stereotypes «index», «query», ... to classes

- **Hints for the design of UWE models**
  - use of UWE element in an incorrect diagram (e.g. presentation class in a navigation diagram)
Current and future versions of MagicUWE

- **Current version**
  - implemented as a MagicDraw plugin
    - uses the open API from MagicDraw

- **Qualities (any CASE tool should have)**
  - easy installation (installer is provided)
  - easily configurable
    - hint messages can be changes or even disabled
    - standard identifications for UWE packages can be changed
  - adaptable
    - display of class or property in toolbar
  - extensible
    - modular implementation
    - well documented code
    - build script to create and launch new versions

- **Future versions**
  - support of new UWE features (e.g. RIAs, patterns)
  - support of preparation of models for automatic generation (transparent for the designer)
Goal: fully automatic generation of Web applications/RIAs from UWE models
UWE4JSF Characteristics

- Implemented as a set of plug-ins for the Eclipse platform
  - Uses technologies from Eclipse Modeling Project: EMF, ATL, JET
- Produces JSF 1.2 applications
- Supports flexible integration of JSF component libraries
- Easy integration of legacy code
- High flexibility in model detail level

- Uses abstract and concrete presentation model to generate the user interface
- Uses Object Graph Navigation Language (OGNL) to define queries, data selection, process behavior, etc
- Allows integration of comp. libraries like Apache MyFaces Tomahawk, etc. and integration of custom components
UWE4JSF Development Process

- **UML Metamodel**
- **UWE Profile**
- **UML Source Model**
  - PIM
  - Presentation Element Mappings
- **UML to UWE (ATL)**
- **UWE Metamodel**
- **UWE Source Model**
- **PSM to Code (JET)**
- **JSF-PSM**
- **JSF to UWE (ATL)**
- **Validation**
  - ERROR
- **Web Application**
  - UWE4JSF Framework
  - JSF
  - * .java, * .jsp, * .xml
  - non-generated Java classes, libraries, stylesheets, etc.
- **Default Presentation Configuration**
  - generated artefacts
  - non-generated artefacts
  - transformation
# Evolution of UWE

<table>
<thead>
<tr>
<th>Current Status</th>
<th>Modelling Language</th>
<th>Metamodel</th>
<th>Development Process</th>
<th>Tool support</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UML profile for content, navigation, process, presentation <em>(UWE Profil 1.8)</em></td>
<td>Conservative extension of UML metamodel (light-weight) and profileable</td>
<td>Using different type of model transformations (Java, ATL)</td>
<td>UML CASE tools, plugin for MagicDraw <em>(MagicUWE 1.3.1)</em> Eclipse plugins <em>(UWE4JSF)</em></td>
</tr>
<tr>
<td>Ongoing Work</td>
<td>Extension for RIAs, services, revised requirements modelling</td>
<td>Extending with model elements for RIAs and services</td>
<td>Use of patterns</td>
<td>Extending MagicUWE plugin for RIAs</td>
</tr>
<tr>
<td>Future Work</td>
<td>Extension for adaptive RIAs</td>
<td>Revisiting adaptation model elements</td>
<td>Consistency and evolution of Models</td>
<td>Plugins for other CASE tools (e.g. open source), editor for development environment</td>
</tr>
</tbody>
</table>
UWE is a software engineering approach for the Web domain aiming to support Web application development. The key aspect that distinguishes UWE is the UML-based design, which means that all UWE metamodels and related concepts are specified as UML diagrams.

Overview
The main focus of the UWE approach is to provide a tool support for the systematic design, and tool support for the (semi-)automatic generation of Web applications.

UWE's notation is defined as a "lightweight" extension of the UML Profile for the Web domain.
Thank you for your attention!

Questions?

Nora Koch
www.pst.ifi.lmu.de/people/staff/koch
nora.koch (at) pst.ifi.lmu.de

UWE
http://uwe.pst.ifi.lmu.de
uwe (at) pst.ifi.lmu.de