
Vorlesung „Methoden des Software-Engineering“

Block F: „Web-Engineering“
Methoden und Techniken zur Entwicklung von
Web-Anwendungen

Nora Koch

Einheit F.2, 06.02.2007

WS 2006/2007

Gliederung Block F

- Web-Software, Vorgehensmodelle und Projektmanagement
- Methoden und Techniken zur Entwicklung von Web-Anwendungen

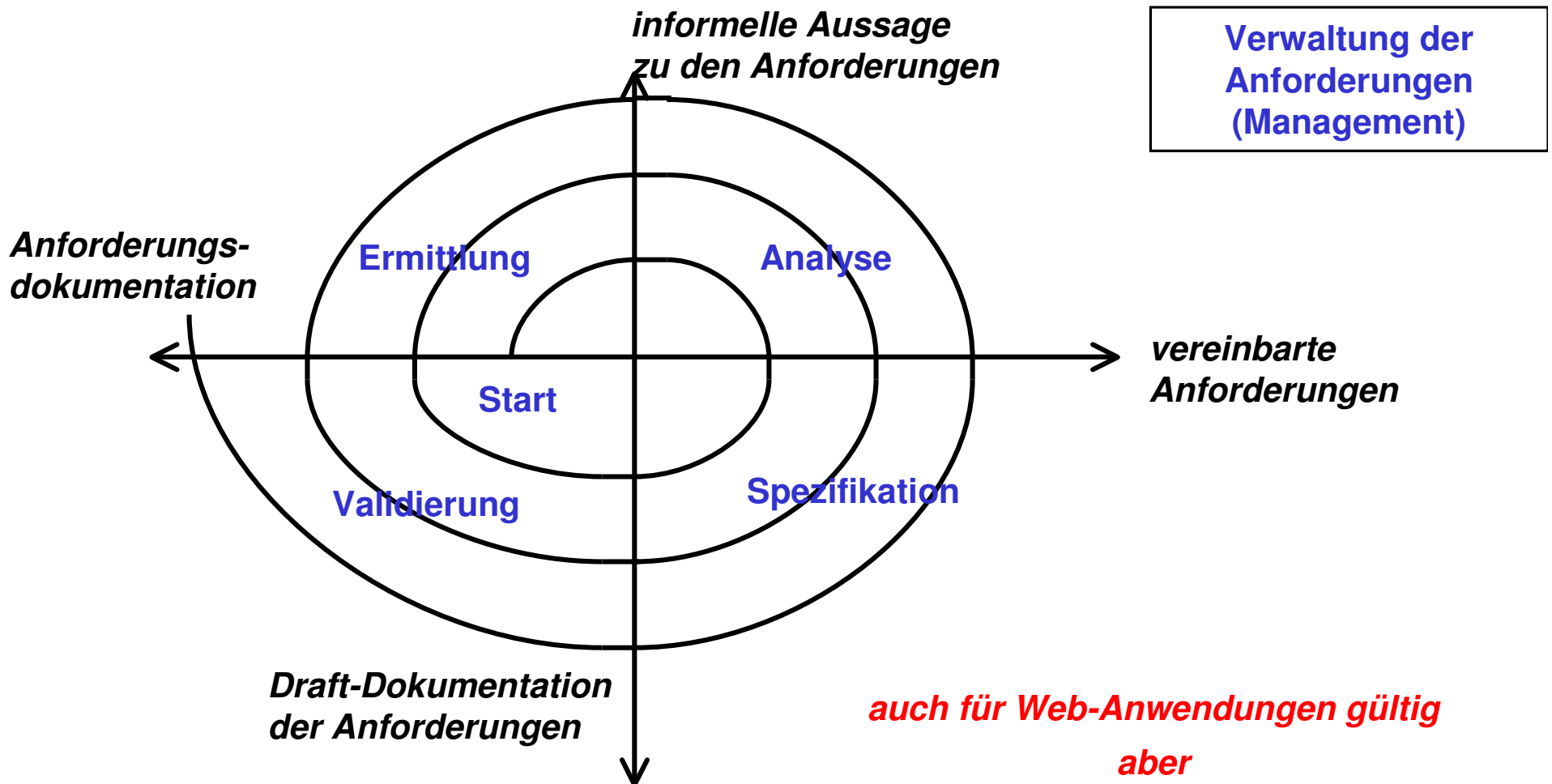
Ziele der Vorlesung

- Überblick im Rahmen des Web Engineering
 - Anforderungsanalyse (im weitem Sinne)
 - Entwurfsmethoden
 - Implementierungstechnologien

Anforderungen an Web-Systeme

- Beispiele
 - Das System muss am 1. September 2007 online verfügbar sein
(Zeitplanungsangabe für das Projekt)
 - Web-Anwendung soll in Java Script geschrieben werden
(Einschränkung zur Web-Technik)
 - Die Übertragung sämtlicher Geschäftsdaten muss gesichert erfolgen
(Sicherheitsbedingung)
 - Die Benutzerschnittstelle muss auch von Behinderten bedienbar sein
(Qualitätsziel für die Präsentation)
 - Ein beliebiger Benutzer muss in der Lage sein, ein gewünschtes Produkt spätestens nach dem Verfolgen von drei Links finden können
(Anforderung an die Navigationsstruktur)
 - Der Benutzer soll durch Auswahl eines Symbols jederzeit die im Warenkorb befindlichen Artikel anzeigen können
(Anforderung an die Funktionalität)

Anforderungsprozess



*auch für Web-Anwendungen gültig
aber
der Prozess wird schneller durchlaufen*

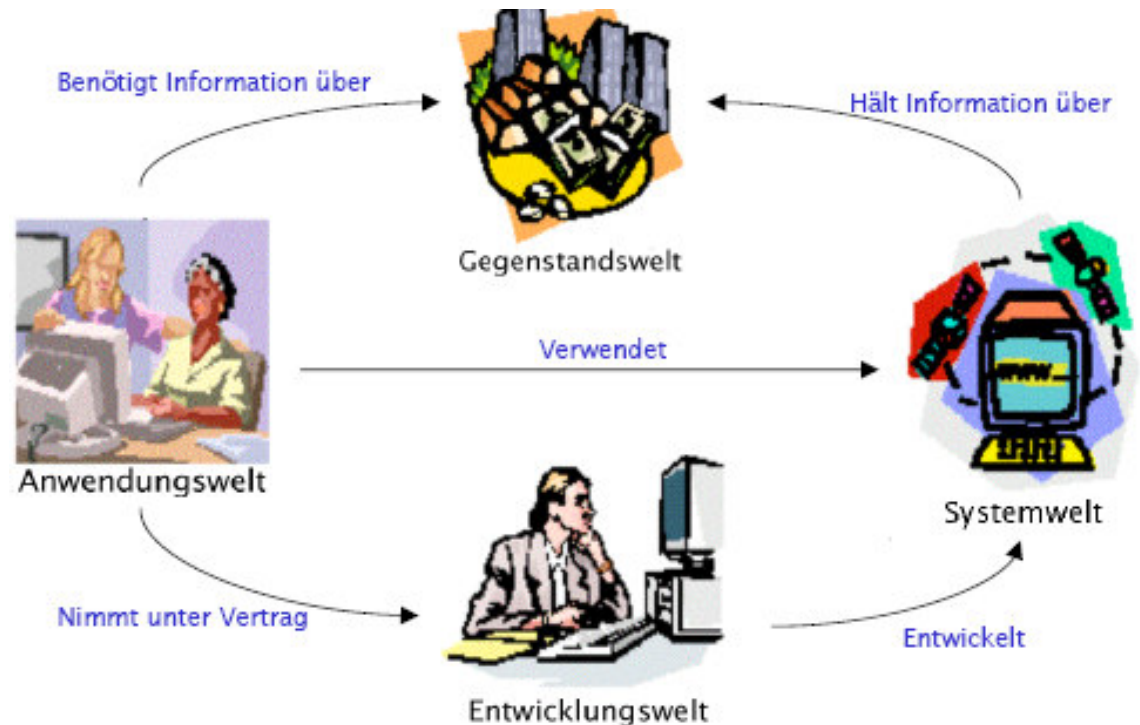
SWEBoK (Software Engineering Body of Knowledge),
IEEE Projekt

Anforderungstypen

- Funktionale Anforderungen
 - Content
 - Navigation
 - Benutzerschnittstelle
 - Anpassungsfähigkeit (Adaptivity)
 - Workflow
- Nicht-funktionale Anforderungen
 - Verfügbarkeit
 - Zuverlässigkeit
 - Usability
 - Bedienbarkeit
 - Verständlichkeit
 - Erlernbarkeit
 - Performance
 - Änderbarkeit
 - Übertragbarkeit

4 Welten

- **Gegenstandswelt**
 - Content, Navigation, Präsentation, Adaptivität, ...
 - **Anwendungswelt**
 - Umgebung: Web
 - Benutzer: anonym
 - **Systemswelt**
 - DB-Transaktionen, Client-Server Architektur, ...
 - **Entwicklungswelt**
 - Zeitrahmen, Entwicklungsteam, nicht-funktionale Anforderungen, ...
- Vergleich zu Produkt-, Nutzungs-, Entwicklungs- und Evolutionssicht



Ermittlung

- Anforderungen sind das Ergebnis eines gemeinsamen Lern- und Konsensbildungsprozesses (potentielle Nutzer, Auftraggeber, Entwickler, ...)
- im Web sind Nutzer in meisten Fällen nicht greifbar für eine Befragung
- Anwendbar sind zahlreiche Technik, wie
 - Interviews (WSDM, NDT, WebML)
 - Fragebogen/Checklisten (NDT)
 - Brainstorming
 - Sketching (UWE)
 - Storyboarding (WebML, UWE)
 - Concept Mapping
 - Dokumentenanalyse (WebML)

Analyse und Spezifikation

- Analyse der ermittelten Anforderungen
 - Konzepterarbeitung
 - Verhandlungen
- Anforderungsdokument für ermittelten Anforderungen
 - Pflichtenheft
 - Konzept
- zahlreiche Beschreibungsformen
 - informale (z.B. User Stories aus eXtreme Programming, formatierte Spezifikation wie Formulare)
 - semi-formal (z.B. visuelle Modellierung in UML)
 - formal (z.B. Spezifikationssprache Z)
- Wahl der Beschreibungsform abhängig von
 - Projektrisiko
 - Adressatenkreis (Stakeholder und Vorwissen)

Techniken zur Anforderungsspezifikation

Techniken	WSDM	OO-H	NDT	OOHDM	UWE	W2000	WebML
Natürliche Sprache (Stories)	✓		✓				✓
Glossar			✓		✓		
formatierte Spezifikation			✓				✓
Use Cases			✓	UIDs	✓	✓	✓
Formale Spezifikation							
Prototyping		✓					

Informale Beschreibung

- Umgangssprachliche Beschreibungen
- Verständnis zwischen Kunden und Entwicklern

Der Kunde kann die ausgewählten Produkte im Online-Warenkorb noch prüfen. Nach Klicken auf <Weiter> werden die Eingaben geprüft. Wird kein Fehler gefunden, wird die Bestellung übernommen und eine E-Mail zur Bestätigung an den Kunden geschickt.

Formatierte Spezifikation

- Unterschiedliche Vorlagen für
 - Benutzergruppen
 - Inhalt
 - Funktionalität und
 - Interaktion
- MOF Metamodell für jeder diese Anforderungsgruppe
- Querverlinkungen durch Identifikatoren
- Wartungsprobleme
- Werkzeugunterstützung
- Beispiel
 - Navigation Development Techniques (NDT)

IR-01	Flight information request
Actors	AC-01: Tourist AC-02: Agency
Description	System shows information about a flight
Entry parameters	The user indicates the flight number
Associated functionality	FR-01: Search by date and departure time
Showed information	SR-01.Company SR-01.Departure SR-01 Arrival
Exit	IR-02
Entry	IR-03

Werkzeug: NDT-Tool

The screenshot displays the NDT-Tool interface for a project titled "PROYECTO: PRO-00-Proyecto de difusión del patrimonio mueble". The interface includes a menu bar with options: "Proyectos", "Requisitos", "Diagrama conceptual", "Modelo Navegacional", and "Modelo Interfaz Abstracta".

On the left, a "Generalización de actores" window shows a hierarchy where "AC-03" is the actor and "AC-04" is inherited from it.

The central "Matriz de trazabilidad" window lists requirements (RA-01 to RA-13) and non-functional requirements (NA-01 to NA-03) in a scrollable list.

The "Prototipos de visualización" window is the active editor, showing a table of prototypes:

Código	Nombre descriptivo
PV-01	Datos de identificación y localización del bien
PV-02	Datos de descripción del bien
PV-03	Datos de análisis del bien
PV-04	Datos del estado de protección del bien
PV-05	Datos del estado de conservación
PV-06	Datos de las intervenciones que se realizan sobre las piezas
PV-07	Datos de los museos

Below the table, the "Datos del prototipo" section includes:

- código:** PV-01
- nombre descriptivo:** Datos de identificación y localización del bien
- descripción:** El sistema deberá permitir la visualización de los datos concretos que se muestran a continuación y la navegación expresada y que representan los
- precondición:** Tras ejecutar una de las frases anteriores se elegirá un bien mueble que será para el que se muestren los datos de identificación.
- comentarios:** (Empty text area)

Buttons for "Nuevo", "Recoger cambios", and "Eliminar" are present. At the bottom, a navigation bar includes tabs for "Datos básicos", "Actores", "Frases", "Funcionalidad asociada", and "Información visualizada".

NDT: M. J. Escalona (2004),
 Universität Sevilla, Spanien

Validierung

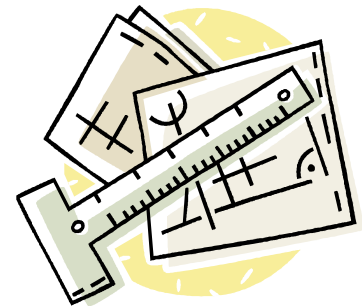
- Ziele
 - Validierung : »Haben wir das Richtige spezifiziert?«
 - Verifikation : »Haben wir richtig spezifiziert?« (Fehler bei der Spezifikation)
- Verfahren
 - Reviews oder Walk-throughs
 - Audits oder Inspektionen
 - Prototyping

Verwaltung

- Änderungen sind ein wesentliches Merkmal von Web-Projekten
 - Anforderungen sind in der Regel nicht statisch
 - kontinuierliche Änderungen
- Das Verwalten von Anforderungen (*Requirements Management*) unterstützt
 - Einbringen neuer Anforderungen
 - Änderungen bestehender Anforderungen
 - Verwaltung der Abhängigkeiten zwischen Anforderungen
 - Verwaltung der Beziehungen zu anderen Entwicklungsergebnissen (*traceability*)
- Werkzeugunterstützung

Entwurf und Modellierung

- Ingenieurswissenschaftliche Disziplinen verwenden Modelle mit großem Erfolg
- Vorteile:
 - **Dokumentation** von Entwurfsentscheidungen
 - **Kommunikation** innerhalb von Projektteams
- Ziel:
 - **Spezifikation** des zu erstellenden Systems
- Ergebnis:
 - **Modelle**: vereinfachte und lesbare (z.B. visuelle) Darstellung der relevanten Aspekte des zu erstellenden Systems



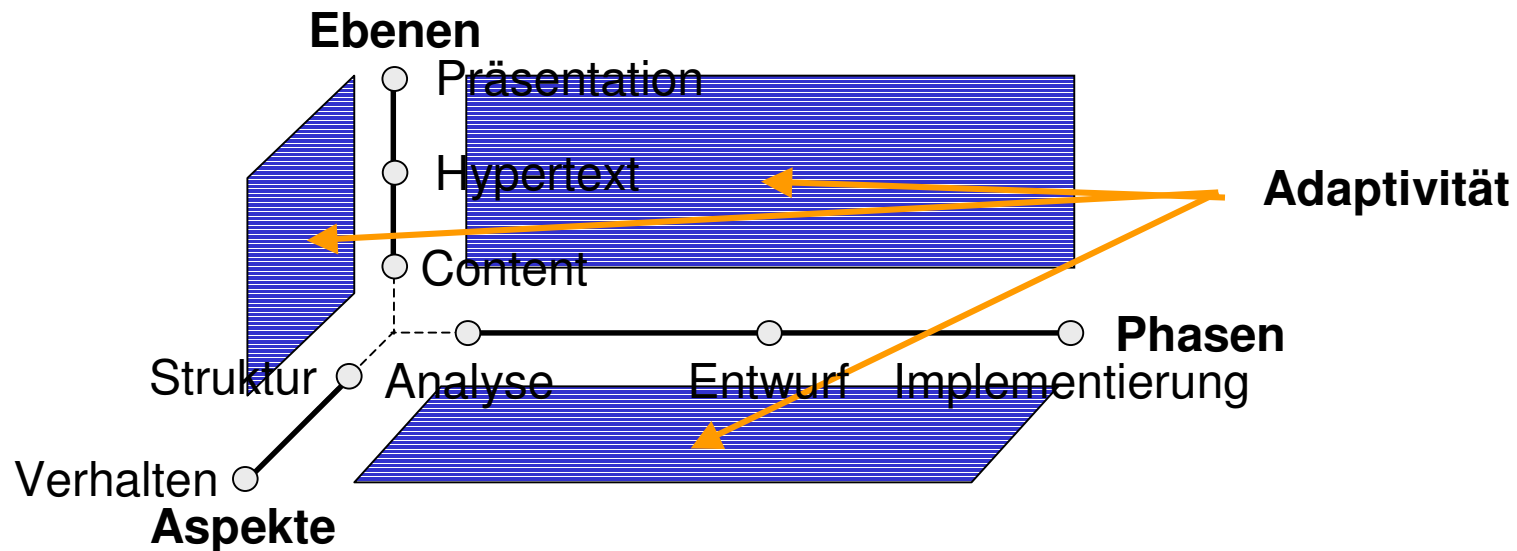
Ziele der Modellierung von Web Anwendungen

- Reduktion der Komplexität bei Analyse und Entwurf
- Dokumentation der Entwurfsentscheidungen
- Vereinfachte, lesbare und visuelle Darstellung
- Grundlage zur Kommunikation im Projekt

und speziell bei Web-Anwendungen

- Trennung der Aspekte
 - Content
 - Geschäftslogik
 - Hypertext
 - Präsentation
 - Adaptivität

Dimensionen der „Web-Modellierung“



- **Verschiedene Vorgehensweisen möglich**
 - Informationsgetriebenes Vorgehen ("content first")
 - Präsentationsgetriebenes Vorgehen ("layout first")
 - Agiles Vorgehen ("test first")
- **Empfehlung: einheitlicher Modellierungsformalismus, e.g. UML**

Anforderungsmodellierung mit UWE

- Spezifikation der Anforderungen mit unterschiedlichen Techniken
- Use Cases für die visuelle Modellierung von funktionalen Anforderungen
 - Navigation
 - andere Funktionalität
- Aktivitätsdiagramme für die detailliertere Beschreibung der Use Cases
- Anforderungen aus Sicht der Akteure

Fallstudie: Konferenzbegutachtungssystem

- **Akteure**
 - Autoren, Gutachter (Programmkomitee) und Chair der Konferenz
- **Funktionale Anforderungen**
 - Einreichung von Konferenzbeiträgen
 - Zuteilung der zu begutachtenden Arbeiten
 - Verfassung der Begutachtungsberichte
 - Durchführung des Begutachtungsprozesses
 - Erstellung der Listen der angenommenen / abgelehnten Konferenzbeiträge

UWE: UML-based Web Engineering Methode

- verschiedene Modelle

- Anforderungsmodell
- Content Modell
- Navigationsmodell
- Präsentationsmodell
- Prozessmodell
- Adaptivitätsmodell

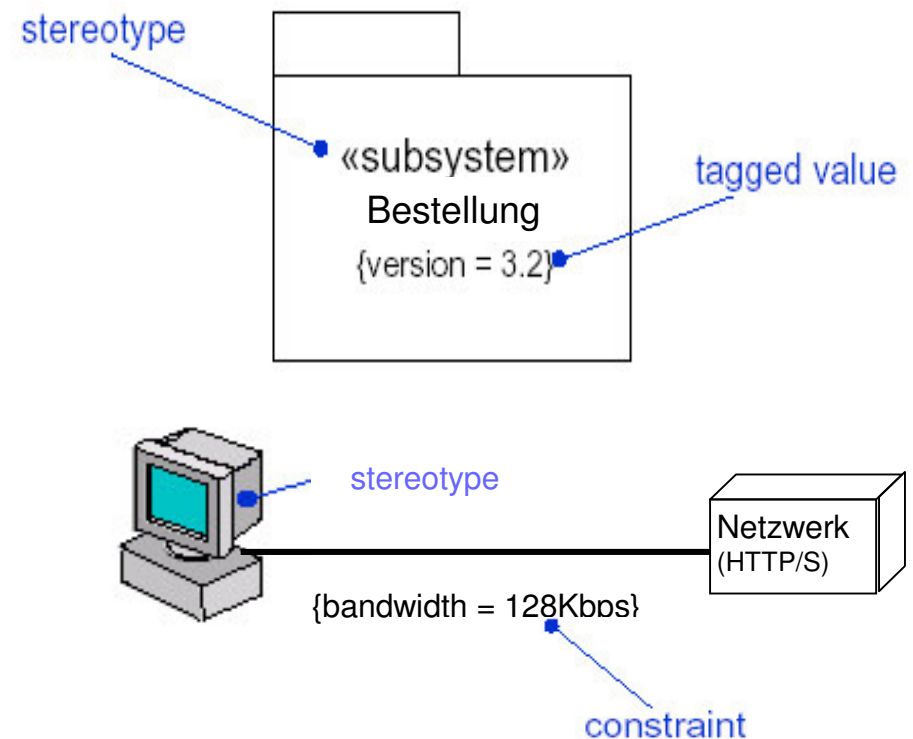
- systematisches Vorgehen

- Notation basierend auf

- Erweiterungsmechanismus der UML

- stereotypes
- tagged values
- constraints

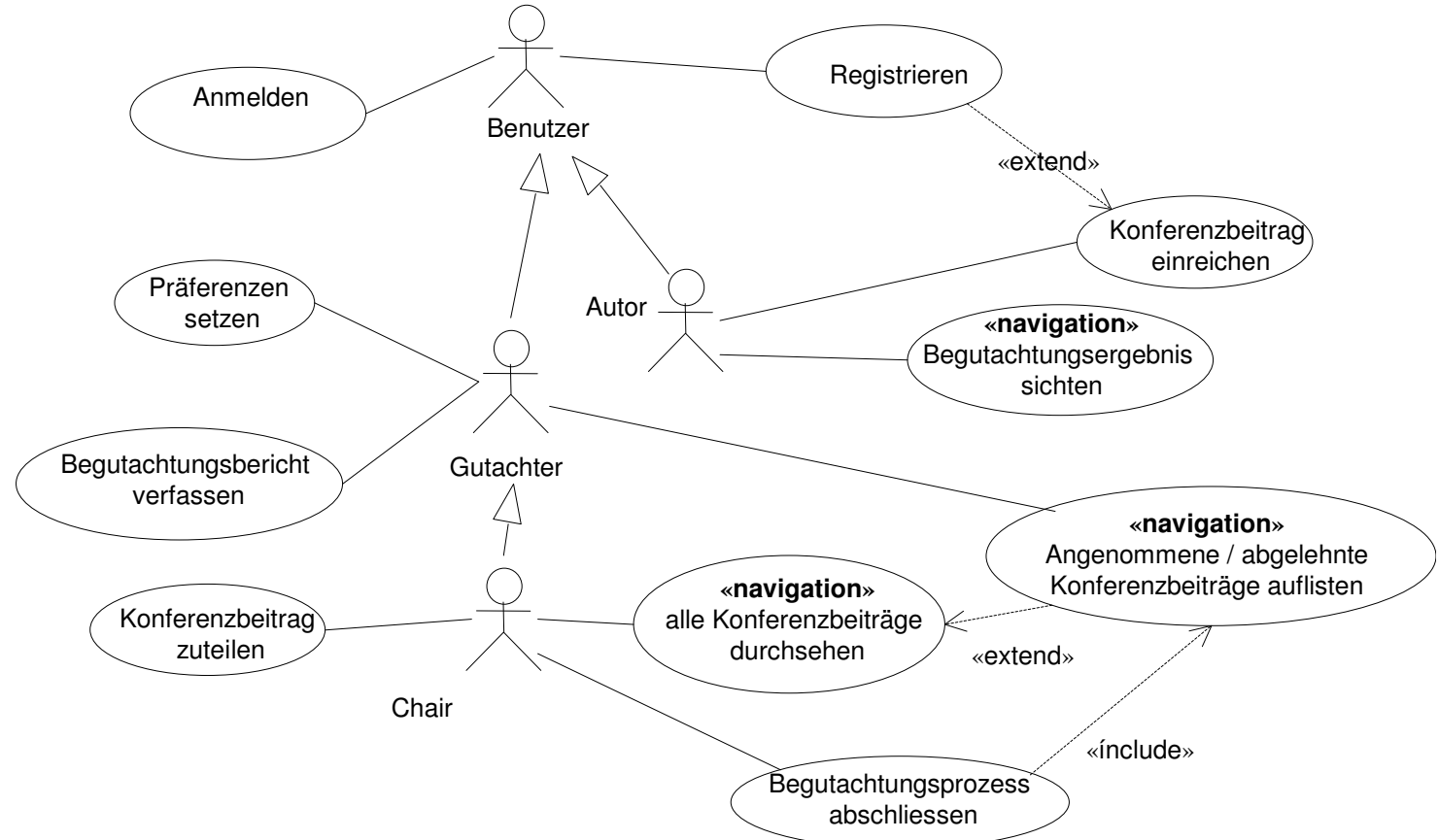
- UWE wird von einem Werkzeug (CASE-Tool) unterstützt: ArgoUWE



Anforderungsmodell (1)

Fallstudie: Konferenzbegutachtungssystem

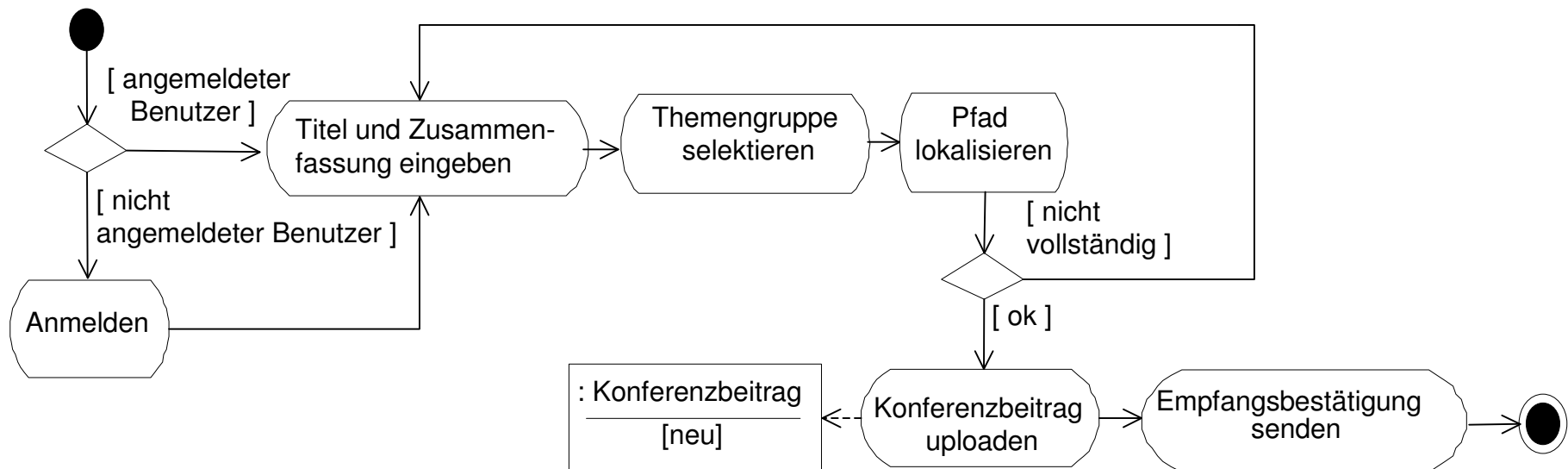
- UML Use Case Diagramm mit Erweiterung von UWE
- Kennzeichnung der Anwendungsfälle, die Navigationfunktionalität darstellen, durch einen Stereotyp «navigation»



Anforderungsmodell (2)

Fallstudie: Konferenzbegutachtungssystem

- Detaillierung der Abläufe mit Aktivitätsdiagrammen
 - Repräsentation der Anwendungslogik
 - Darstellung der Funktionalität eines Web-Services
- Beispiel eines Aktivitätsdiagramms für den „Einreichprozess eines Beitrages“

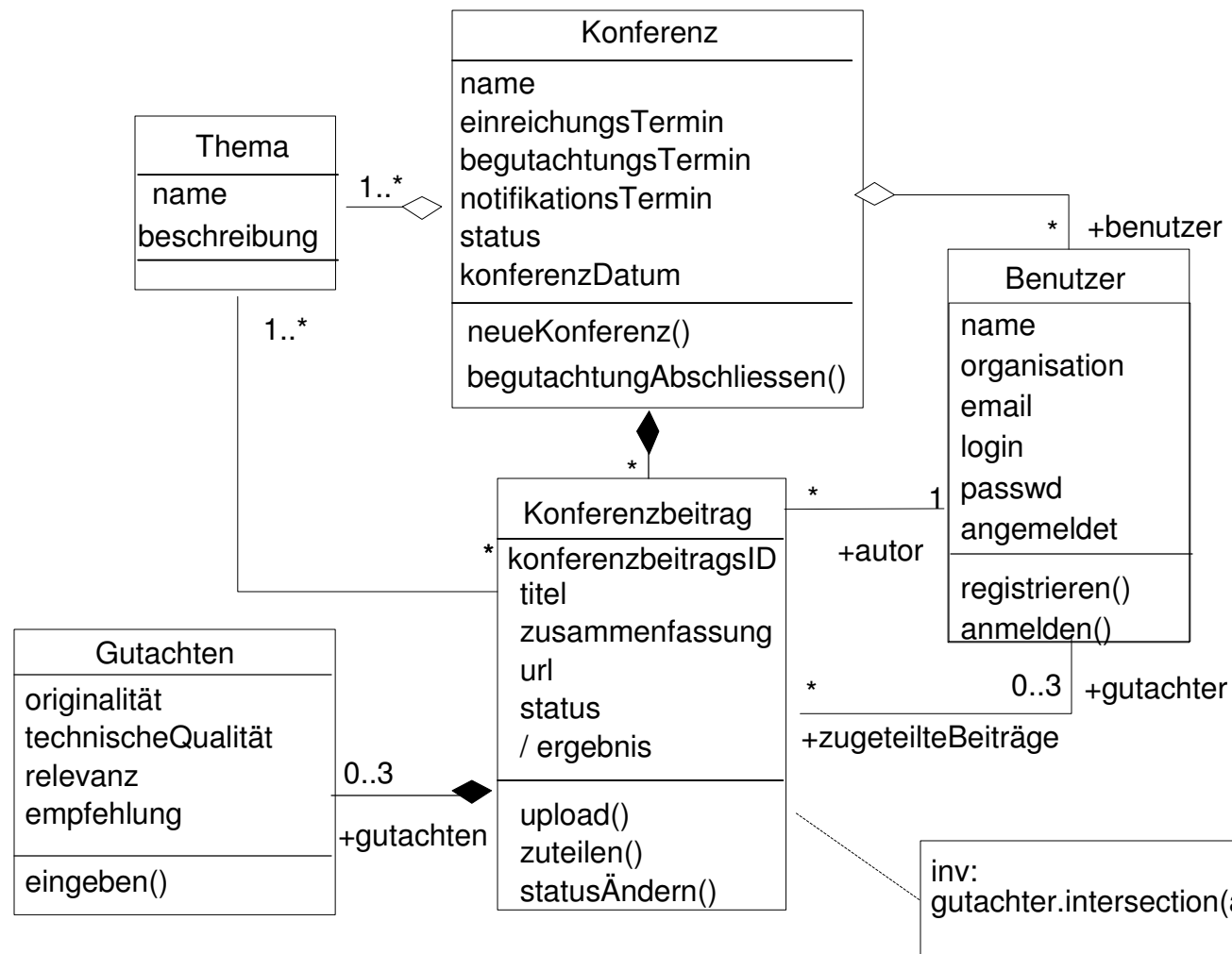


Content-Modellierung

- Ziele
 - Erstellung des Problembereichmodells
 - Trennung des Inhaltes von Hypertextstruktur und Präsentation
- Charakteristika der Content-Modelle
 - Dokumentenzentrierter Charakter
 - Multimedialität
 - Integration existierender Daten und Software
- Ergebnisse
 - Struktur: klassisches ER- oder Klassendiagramm
 - Verhalten: Zustandsdiagramme
 - Methoden unterscheiden sich wenig voneinander

Content-Modellierung: Struktur

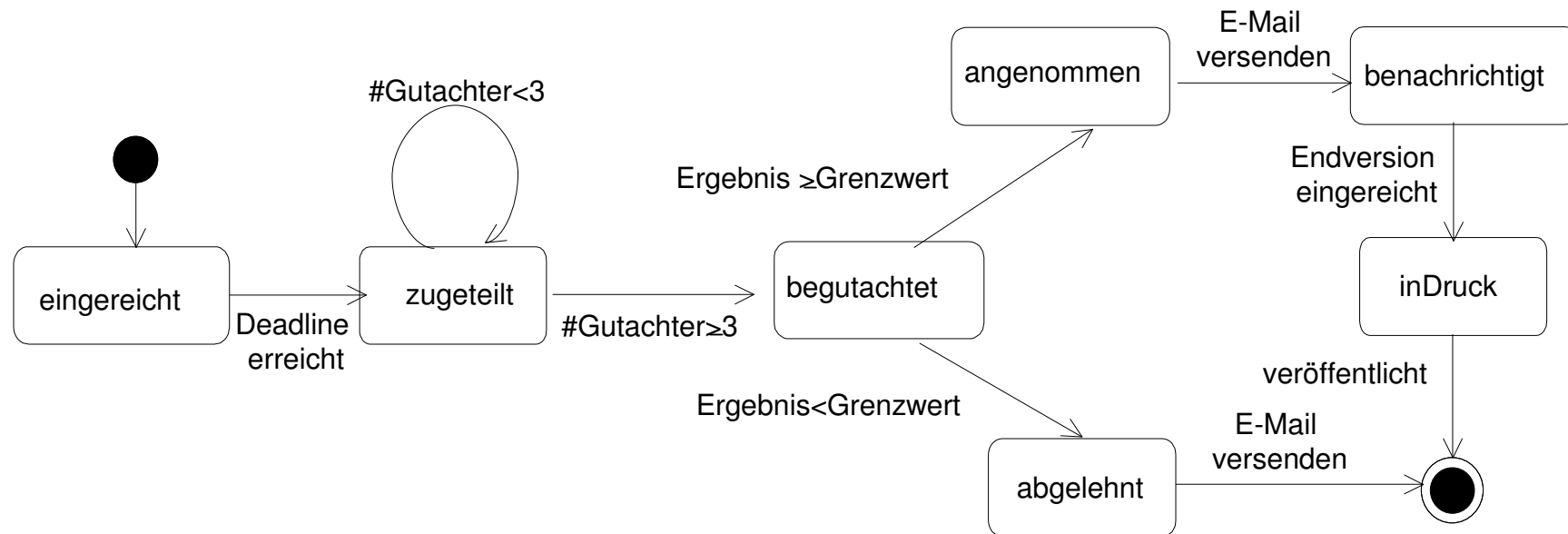
Fallstudie: Konferenzbegutachtungssystem



Content-Modellierung: Verhalten

Fallstudie: Konferenzbegutachtungssystem

- Lebenszyklus von Konferenzbeitragsobjekten
- Darstellung anhand eines Zustandsdiagramms



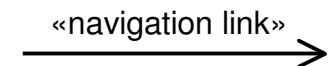
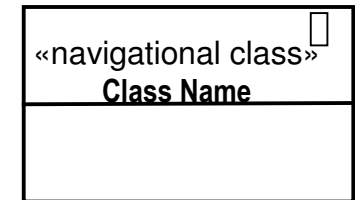
Hypertext-Modellierung

- Ziele
 - (Semi)-Automatische Generierung aus Content-Modell und Anforderungsmodell
 - Darstellung der **Knoten und Verweise** der Hypertextstruktur
 - Entwurf der **Navigationspfade**
- Charakteristika der Hypertext-Modelle
 - Unterschiedliche Notation je nach Methode
 - Unterschiedliche diagrammatische Darstellung je nach Methode
- Ergebnis
 - Navigationsmodell

Hypertext-Modellierung

Konzepte der Hypertext-Modellierung

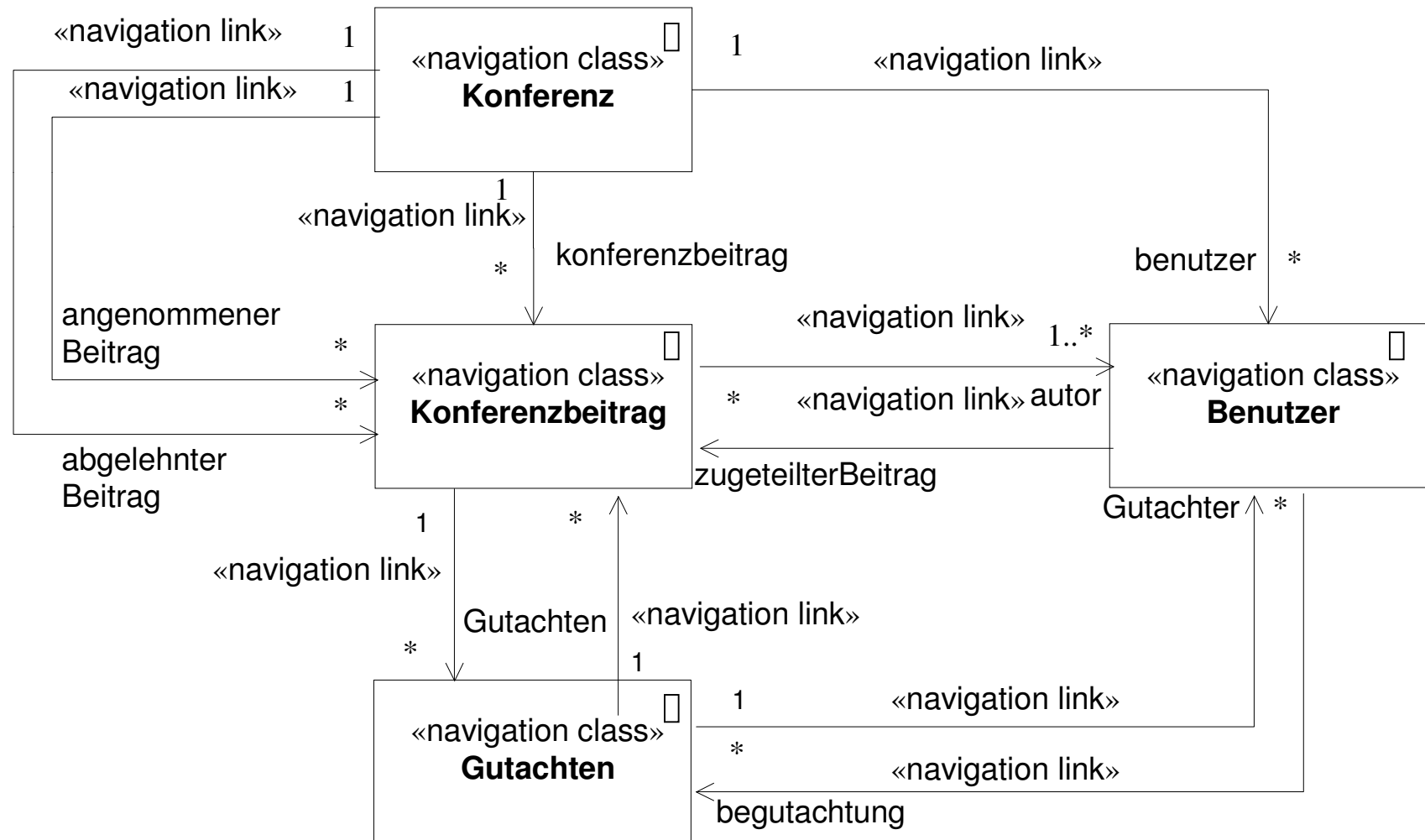
- Modellierungskonzepte für die Hypertext-Struktur
 - «navigation class» für Navigationsknoten
 - «navigation link» für Navigationsverweise
- Methodik (semi-automatisches Verfahren)
 - Navigationsklassen aus navigationsrelevante Klasse des Content-Modell (manuelle Markierung)
 - Navigationslinks für Assoziationen im Content-Modell
 - zusätzliche Navigationslinks basieren auf dem Anforderungsmodell
 - ...



Notation UWE

Hypertext-Modellierung

Fallstudie: Konferenzbegutachtungssystem – Navigationsmodell in UWE (1)

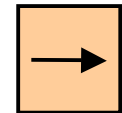
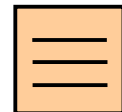


Hypertext-Modellierung

Konzepte der Hypertext-Modellierung

- Modellierungskonzepte für den Zugriff

- «menu» für den Zugriff auf Knoten unterschiedlicher Klassen
- «index» für den Zugriff auf Knoten einer Klasse
- «guidedTour» für den sequentiellen Durchlauf einer Folge von Knoten
- «query» für die Suche nach und direkten Zugriff auf Knoten



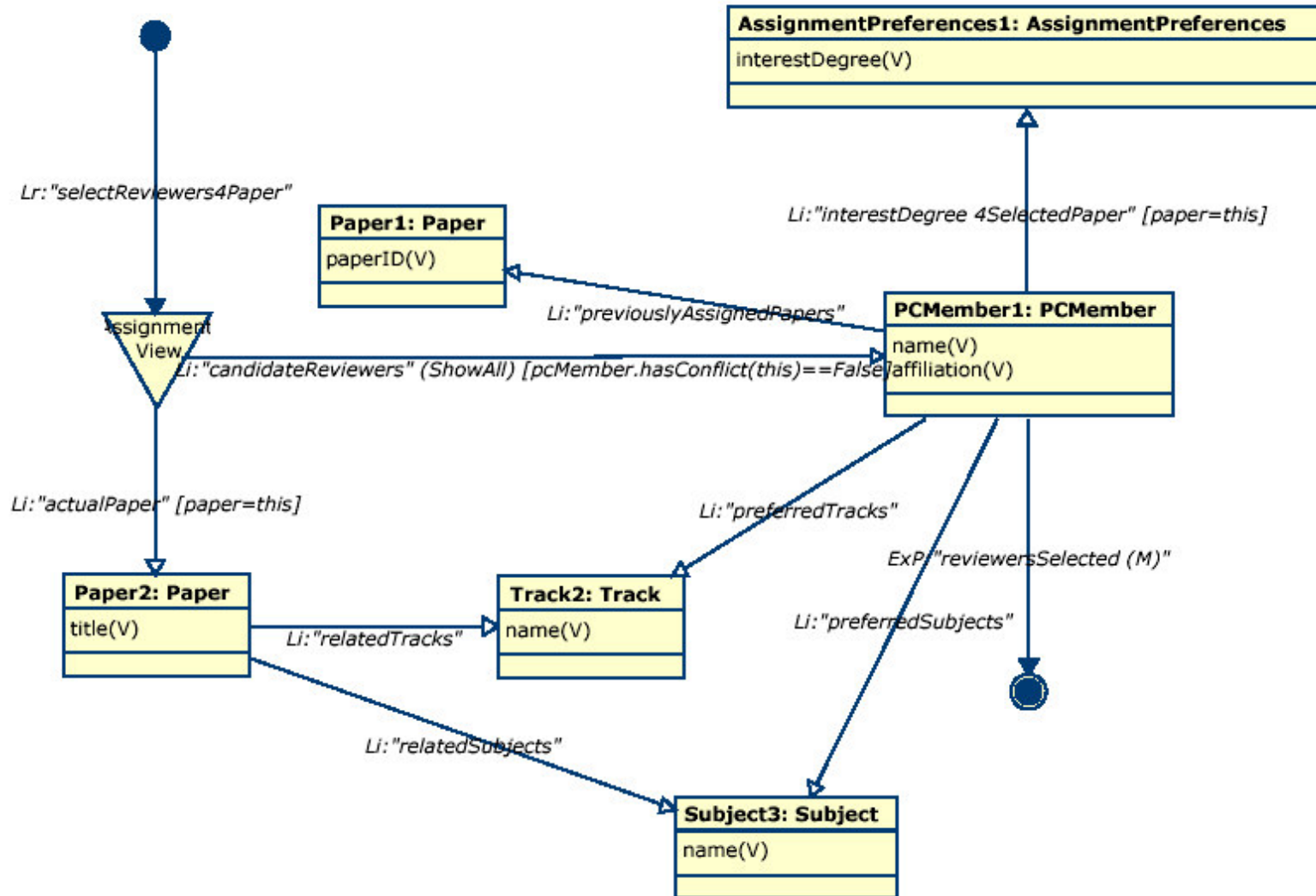
- Methodik (automatisches Verfahren)

- ein Index wird für alle Navigationslinks mit Multiplizität > 1
- ein Menü wird jeder Klasse mit mehr als einem ausgehenden Navigationslink (mit Kompositionsbeziehung hinzugefügt)
- Menüitems nach Rollennamen von ausgehenden Navigationslinks benennen

Notation UWE

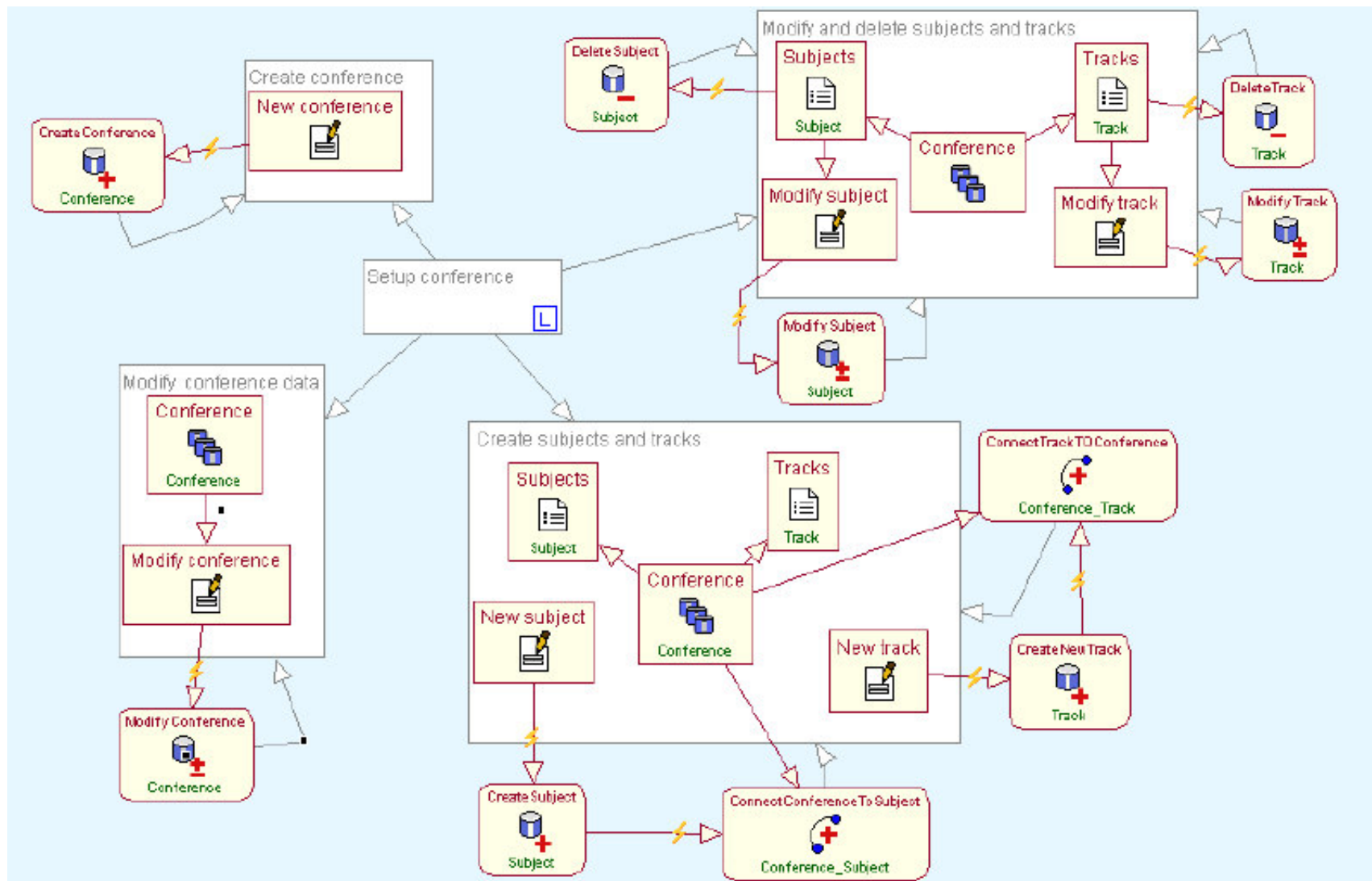
Hypertext-Modellierung

Fallstudie: Konferenzbegutachtungssystem – Navigationsmodell in OO-H (2)



Hypertext-Modellierung

Fallstudie: Konferenzbegutachtungssystem – Navigationsmodell in WebML



Hypertext-Modellierung

Konzepte für Verweise der unterschiedlicher Methoden

- **WebML**
 - kontextuelle Verweise
 - nicht-kontextuelle Verweise
 - Intra-Seiten-Verweise
 - Inter-Seiten-Verweise
- **WSDM**
 - einfache Verweise
 - konditionale Verweise
 - vielfache Verweise
- **UWE**
 - Navigationsverweise
 - Prozessverweise
 - externe Verweise
- **OO-H**
 - funktionale Verweise
 - interne Verweise
 - externe Verweise
 - externe Dienste

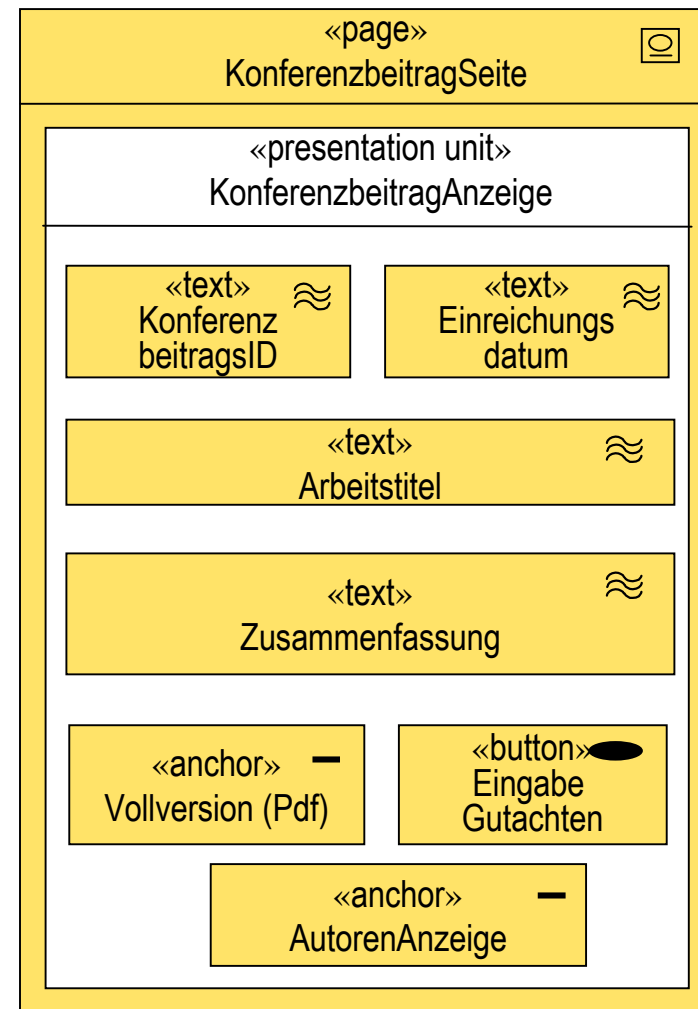
Präsentationsmodellierung

- Ziele
 - Gestaltung der Struktur und des Verhaltens der Benutzerschnittstelle
- Charakteristiken der Präsentationsmodellierung
 - hierarchische Zusammensetzung von Seiten aus Präsentationselementen
 - von wenigen Methoden unterstützt
 - als Alternative: prototypische Entwicklung der Web-Anwendung
- Ergebnisse
 - statisches Präsentationsmodell (Klassendiagramm)
 - dynamisches Interaktionsmodell (Sequenz- bzw. Kollaborationsdiagramm)

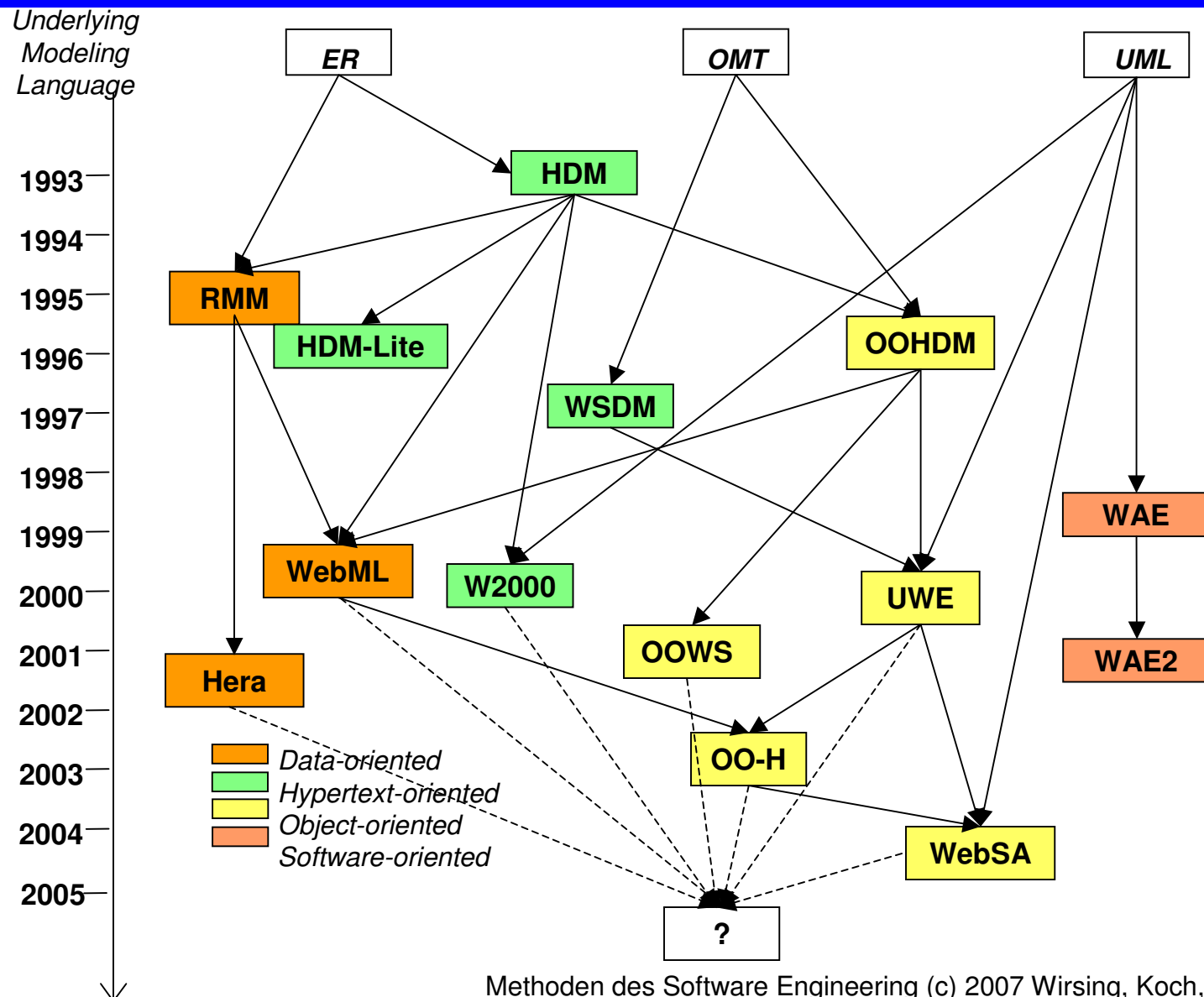
Präsentationsmodellierung

Fallstudie: Konferenzbegutachtungssystem – Präsentationsmodell in UWE

- «page» ist die minimale Präsentationseinheit, die dem Benutzer angezeigt wird
- «presentation unit» dient zur Gruppierung zusammengehöriger Präsentationselemente, die ein logisches Fragment der Seite darstellen
- Präsentationselemente sind u.a.:
 - «anchor»
 - «text»
 - «image»
 - «image»
 - «button»



Übersicht: Methoden zur Entwicklung von WA



Methoden zur Entwicklung von WA

HDM	Hypertext Design Model
HDM-Lite	
Hera	
OO-H	Object-Oriented Hypermedia Method
OOHDM	Object-Oriented Hypermedia Design Method
OOWS	Object-Oriented Web Solution
RMM	Relationship Management Methodology
UWE	UML-based Web Engineering
W2000	
WAE	Web Application Extension
WebML	Web Modeling Language
WebSA	Web Software Architecture
WSDM	Web Site Design Method

Methoden im Vergleich

Methoden	Notation	Schwerpunkte	Werkzeug
OO-H	UML + eigene Notation	Präsentationsspezifikation	Modellierung Generierung
OOHDM	UML + eigene Notation	mächtige Konzepte für kontextuelle Navigation	Modellierung
RMM	ER + eigene Notation	vordefinierter Prozess	Modellierung Generierung
UWE	UML	UML-basiertes Metamodell	(Erweiterung) Standard UML- Werkzeuge
W2000	UML	benutzerzentrierte Hypertext-Modellierung	Erweiteres UML- Werkzeug
WAE	UML	Implementierungs-orientiert Architekturentwurf	Standard UML- Werkzeuge
WebML	eigene Notation	ausgereifte Notation Datenbankintegration	Modellierung, Generierung
WSDM	eigene Notation	benutzerzentrierte Vorgehensweise bei der Analyse	

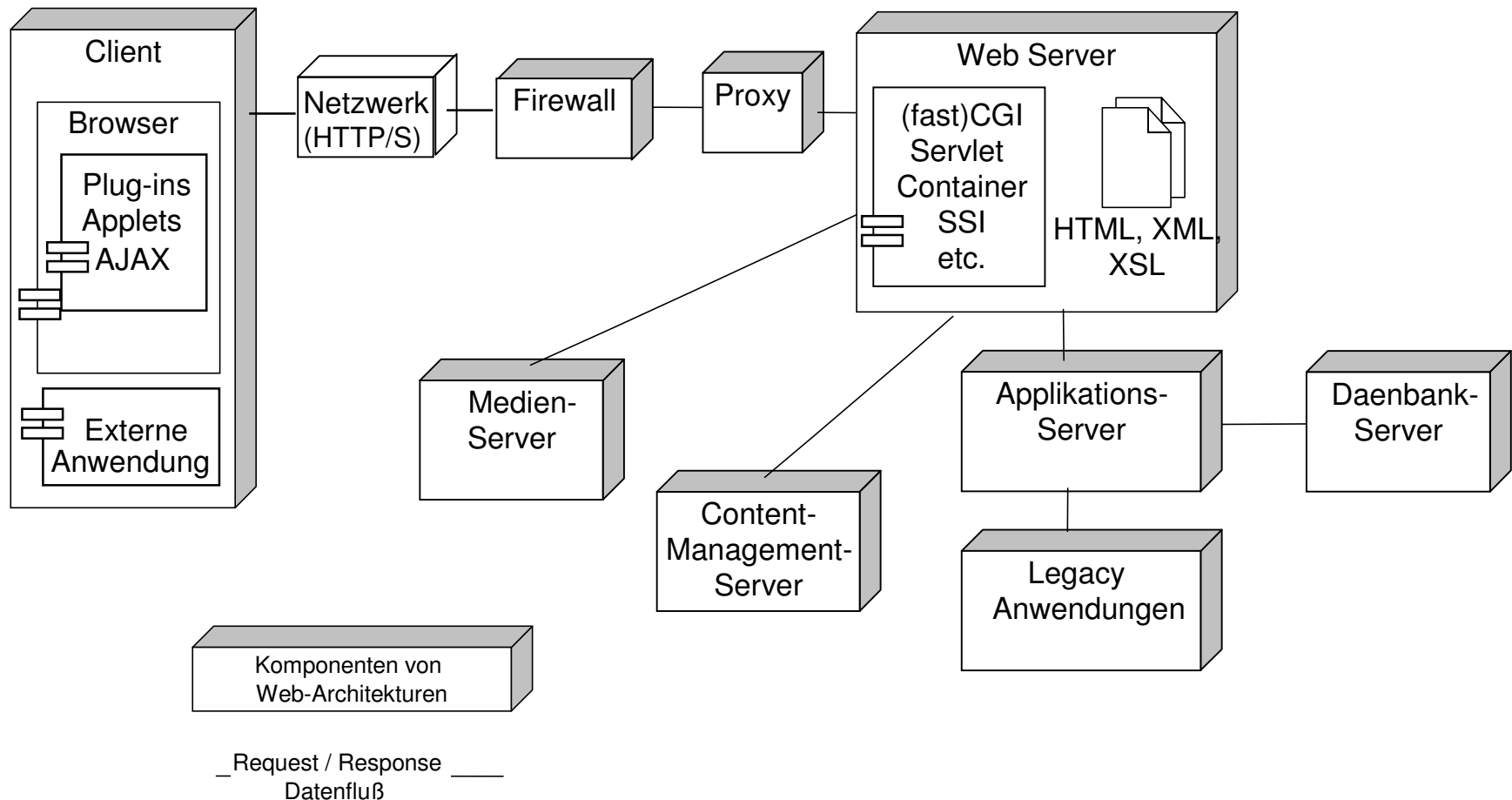
* nicht weiter entwickelt

Implementierung von Web-Anwendungen

- Umsetzung der Entwurfsergebnisse
- (die auf den Anforderungsdokumentation basieren)
- programmiert mit Web-spezifischen Techniken
- in einer spezieller Hardware- und Software-Umgebung (Client-Server; Netzwerk, Firewall, ...)

- Unreife technischer Lösungen
 - kurze Produktlebenszyklen
 - Open-Source Produkte
- Fehlende Standards

Komponenten von Web-Architekturen



Grundlagen – Markup und Styles

- **Markup (Auszeichnungs-) Sprachen**
 - Begriff aus dem Verlagswesen (Layout);
 - Bsp: TeX (procedural), LaTeX (procedural, descriptive), HTML (presentational, procedural, descriptive), Wiki notations (presentational)
 - im Web-Umfeld ursprünglich basierend auf **SGML** (Standard Generalized Markup Language), weitgehend ersetzt durch **XML**-basierende Definitionen
 - Web-Vertreter: **HTML** (Hypertext Markup Language), **XHTML** (eXtensible HTML), **DHTML** (Dynamic HTML: Ebenen, Widgets, Animationen)
- **Style sheets**
 - HTML trennt nicht konsequent zwischen Inhalt, Struktur und Formatanweisung (<I>,
, ...)
 - **CSS** (Cascading Style Sheets) für Layoutanweisungen für HTML und XML
 - allgemeiner für XML: **XSL** (eXtensible Stylesheet Language) mit den Teilen **XSLT** (XSL Transformations), **XSL FO** (XSL Formatting Objects) und **XPath** (Zugriff auf XML-Baumstruktur)
- **Dynamisierung von Inhalten und Animierung von Webseiten**
 - Clientseitige Skriptsprachen (JavaScript, ...)
 - Zugriff auf **DOM** (Document Object Model)

Technologien für Dynamische Webseiten

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
  <title>Test</title>
  <style type="text/css">
    h2 {background-color: lightblue; width: 100%}
    a {font-size: larger; background-color: goldenrod}
    a:hover {background-color: gold}
    #example1 {display: none; margin: 3%; padding: 4%; background-color: limegreen}
  </style>
  <script type="text/javascript">
    function changeDisplayState (id) {
      d=document.getElementById("showhide"); e=document.getElementById(id);
      if (e.style.display == 'none' || e.style.display == "") {
        e.style.display = 'green-box-shown'; d.innerHTML = 'Hide example';
      } else {
        e.style.display = 'none'; d.innerHTML = 'Show example'; } }
  </script>
</head>
<body>
  <h2>How to use a DOM function</h2>
  <div><a id="showhide" href="javascript:changeDisplayState('example1')">Show example</a></div>
  <div id="example1">This is the example. (Additional information, which is only displayed on
request)</div>
  <div>The general text continues...</div>
</body>
</html>
```

[Test](#)

Grundlagen – Client/Server-Kommunikation

- **Protokolle**

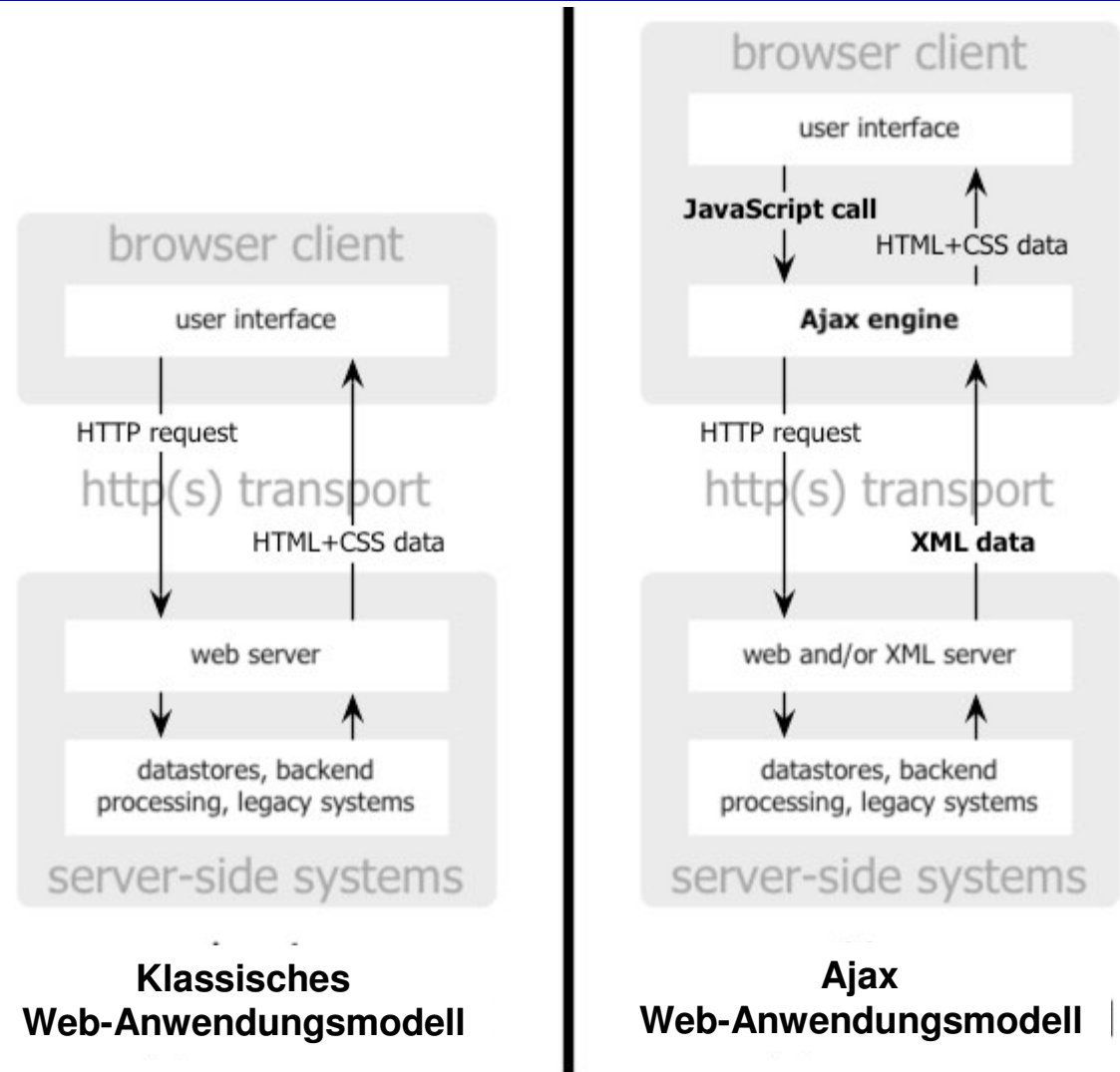
- Simple Mail Transfer Protocol (SMTP)
- Real Time Streaming Protocol (RTSP)
- Hypertext Transfer Protocol (HTTP)

- **Session Tracking**

- Cookies
 - Textdateien zur Speicherung von Serverdaten auf dem Client
 - vom Web Server erzeugt
 - im Header von HTTP-Responses zum Client übertragen
 - vom Web-Browser beim Client speichert
 - vom Web-Browser bei jedem Request zum Server übertragen
- URL Rewriting
 - alle Session-relevanten Daten werden als Parameter der URL hinzugefügt und übertragen

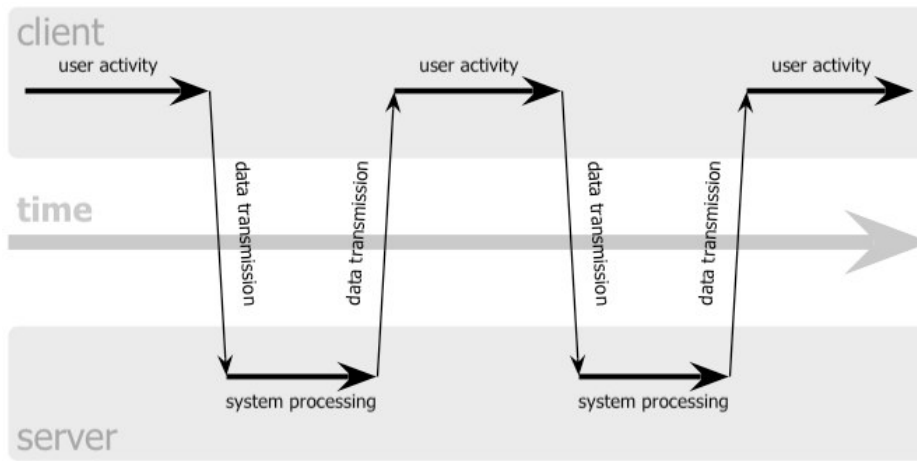
Ajax – Übersicht (1)

- Asynchron JavaScript und XML
 - Technologie für interaktive Webseiten
 - asynchron Client/Server-Kommunikation unter Benutzung von XMLHttpRequest
- Begriff “Ajax” geprägt durch Jesse James Garrett in 2005, aber frühere ähnliche Ansätze sind bekannt



Jesse James Garrett: Ajax: A New Approach to Web Applications, 2005.

Ajax – Übersicht (2)



Klassisches Modell

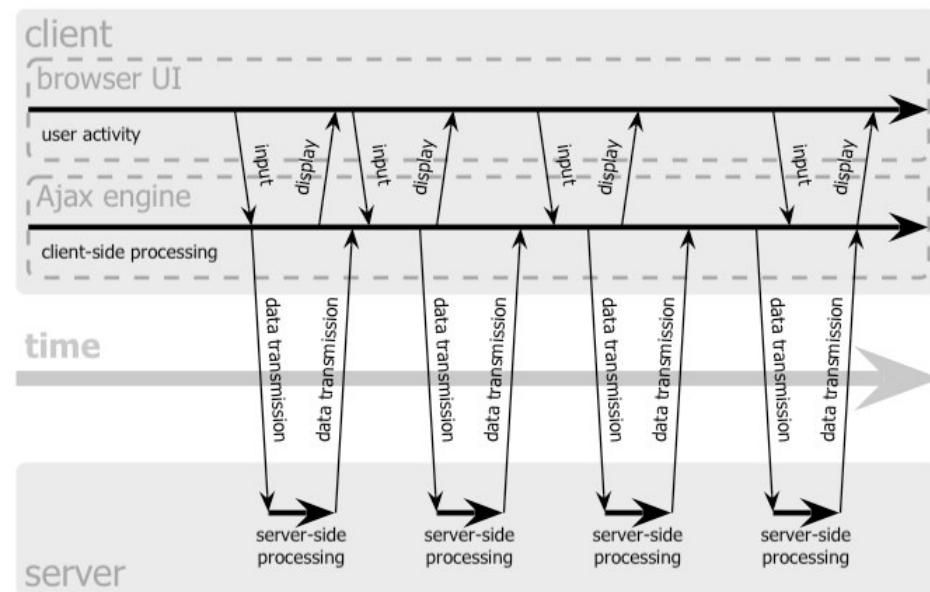
Client/Server Kommunikation wird von der Benutzerinteraktion unterbrochen

Synchron Kommunikation

Ajax-Modell

Nahtlose kontinuierliche Interaktion des Benutzers mit dem Web-Browser

Verwendung von **Callbacks** für nebenläufige **asynchron** Kommunikation



Jesse James Garret: Ajax: A New Approach to Web Applications, 2005.

Ajax – Beispiel

```
<html>
<head>
<title>The Hello World of AJAX</title>
<script language="JavaScript" type="text/javascript">
    function getXmlHttpRequestObject() {
        if (window.XMLHttpRequest) {
            return new XMLHttpRequest();
        } else if(window.ActiveXObject) {
            return new ActiveXObject("Microsoft.XMLHTTP");
        } else {
            alert("Your browser doesn't support the XmlHttpRequest
object.");
        }
    }
    var receiveReq = getXmlHttpRequestObject();
    function sayHello() {
        if (receiveReq.readyState == 4 || receiveReq.readyState == 0) {
            receiveReq.open("GET", 'SayHello.html', true);
            receiveReq.onreadystatechange = handleSayHello;
            receiveReq.send(null);
        }
    }
    function handleSayHello() {
        if (receiveReq.readyState == 4) {
            document.getElementById('span_result').innerHTML = receiveReq.responseText;
        }
    }
</script>
</head>
<body>
    <a href="javascript:sayHello();">Say Hello</a><br />
    <span id="span_result"></span>
</body>
<!--http://www.DynamicAJAX.com for more great AJAX-->
</html>
```

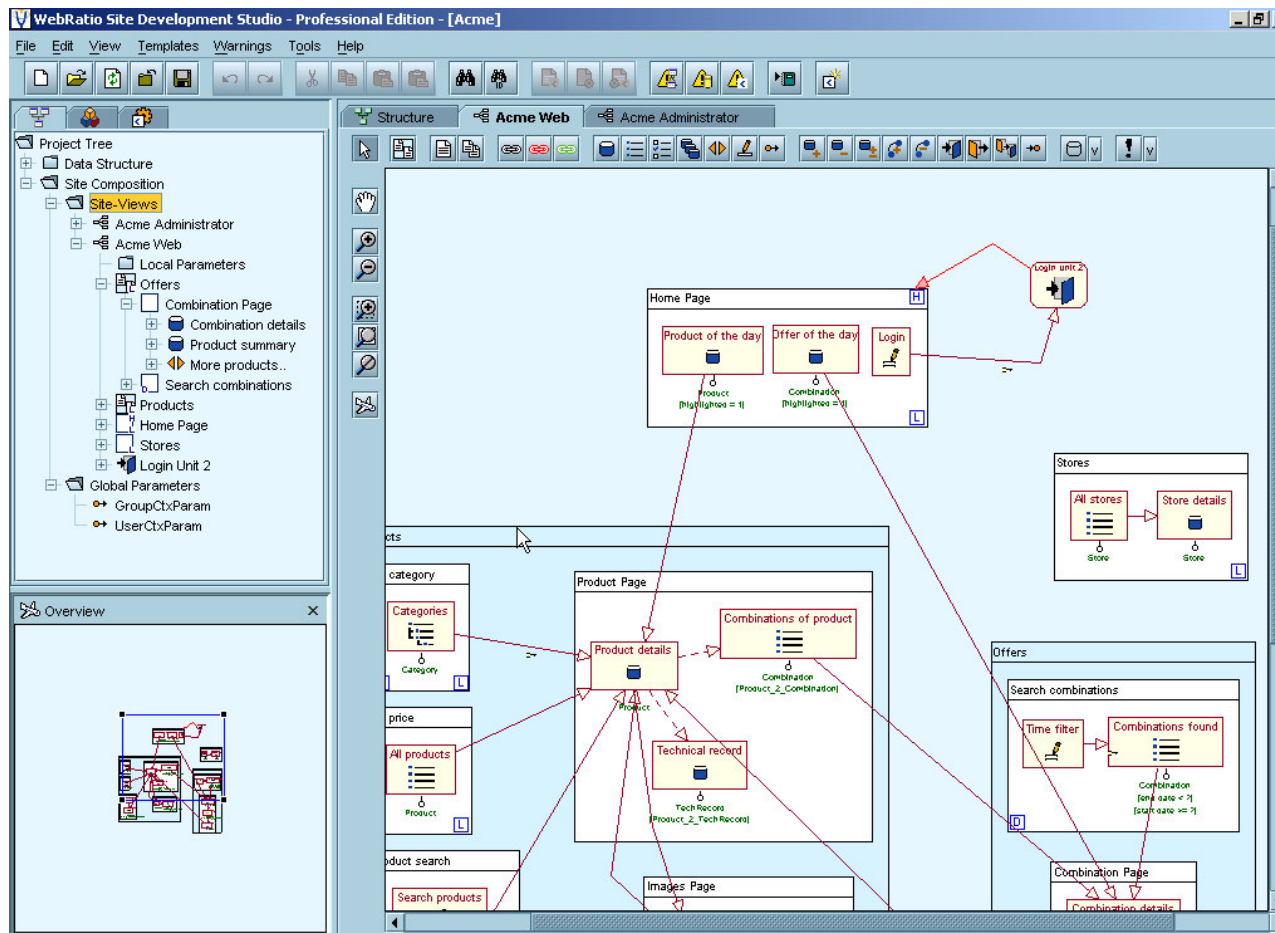
Serverseitige Technologien

- **URI Handler**
 - Common Gateway Interface (CGI)
 - Server-Side Scripting (ASP, PHP, ColdFusion)
 - Servlets
 - Java Server Pages
 - ...
- **Web-Services**
 - SOAP
 - WSDL
- **Applikationsserver**
 - Enterprise Java Beans
 - J2EE Application Server (JBoss, WebLogic Server, WebSphere, &c.)
 - ...

Werkzeugunterstützung

- Computer-aided software engineering (CASE)-Tools
 - für die Erstellung und Pflege der Modelle
 - zur Konsistenzüberprüfung
 - für automatische Codegenerierung
- Werkzeugarten
 - modellgetriebene Entwicklung
 - CASE-Tools, wie IBM Rational Modeller, Magic Draw, Enterprise Architect
 - WebRatio
 - VisualWADE
 - ArgoUWE
 - prototypische Entwicklung
 - FrontPage
 - Flash
 - DreamWeaver

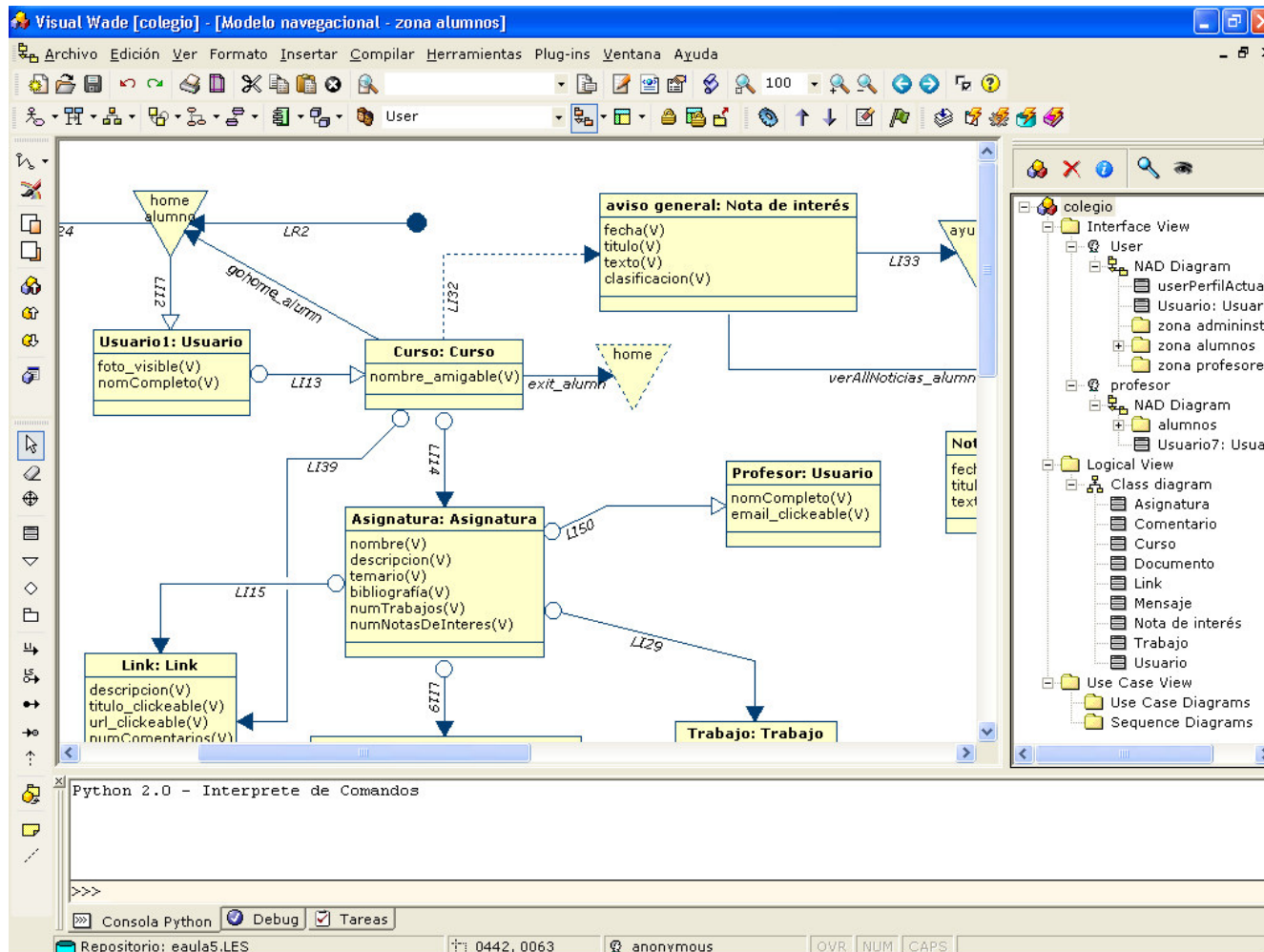
WebRatio



- unterstützt WebML
- ER & UML Notation
- XML & XSL
- Ausgabeformate: HTML, PDF, WML, Word
- geeignet für Jakarta-Struts, JSP-Tag & Microsoft .NET Technologie

www.webratio.com

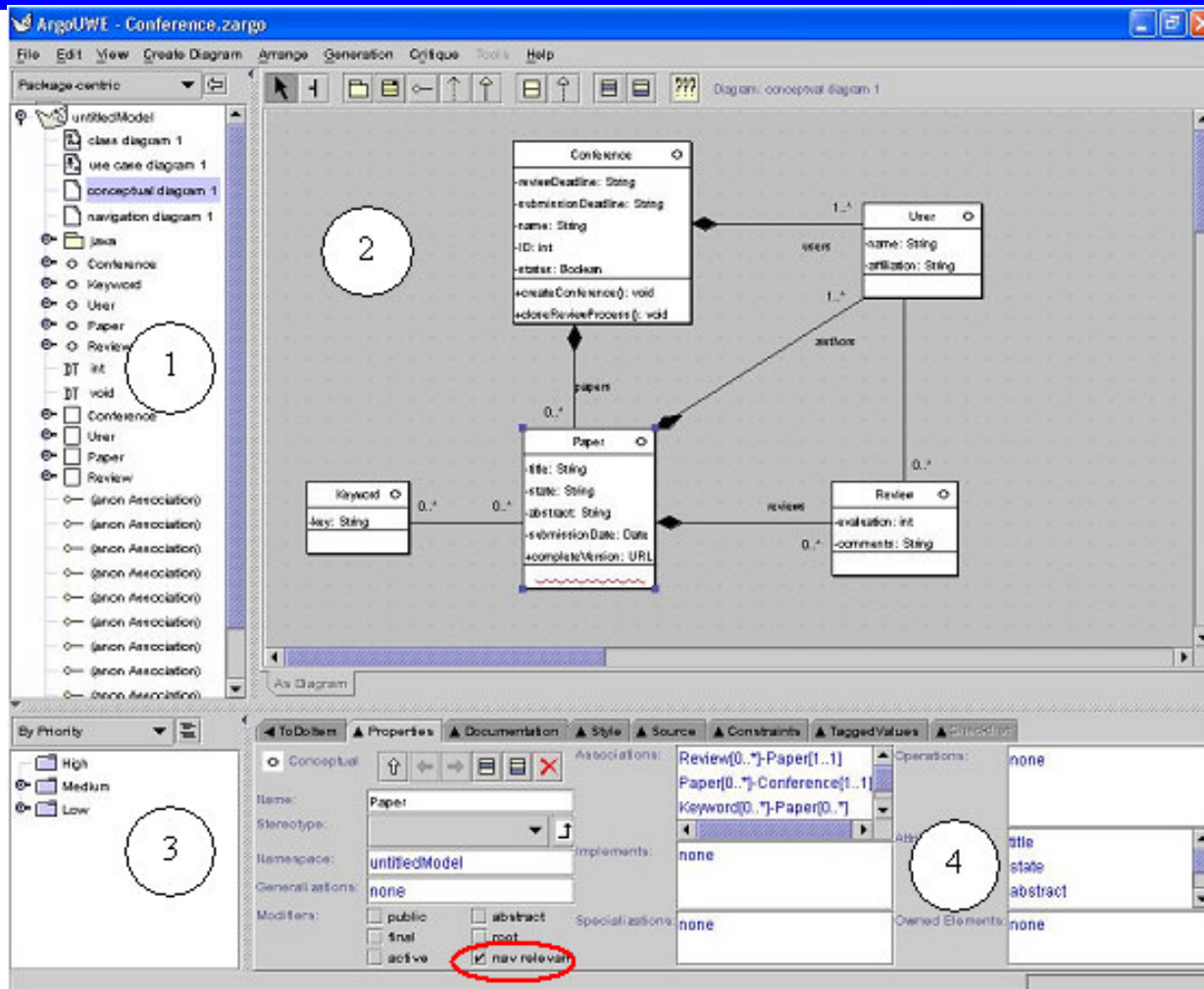
VisualWADE



- unterstützt OO-H
- Template-
strukturen für die
Präsentation
- XML, ASP, JSP
und PHP
- unterschiedliche
Laufzeitum-
gebung

www.visualWADE.com

ArgoUWE



- unterstützt UWE
- ArgoUWE = Erweiterung von ArgoUML
- open-source
- Model Checking basierend auf OCL constraints

www.pst.informatik.uni-muenchen.de/projekte/openuwe

Zusammenfassung

- Anforderungsspezifikation für Web-Software
- Unterschiedliche Modellierungsmethoden beim Entwurf - Konvergenz möglich
- UML Notation setzt sich durch
- Werkzeugunterstützung für die Etablierung einer Methode entscheidend
- Agile Methoden vs. MDA Ansätze

Vorlesung Web Engineering im SS07

- Alexander Knapp, Nora Koch
- 2-stündig, Übung 2-stündig (14-tägig)

- Inhaltsübersicht
 - Anforderungsanalyse
 - Web-Technologien
 - Modellierung von Web-Anwendungen
 - Web-Architekturen
 - Implementierungstechniken
 - Validierung, Verifikation und Testen
 - Nicht funktionale Aspekte
 - Entwicklungsprozesse
 - Projektmanagement

Literatur

- *Web Engineering – Systematische Entwicklung von Web-Anwendungen*
Gerti Kappel, Birgit Pröll, Siegfried Reich & Werner Retschitzegger
dpunkt.verlag, 2003
- *Web Engineering*
Reiner Dumke, Matthias Lothar, Cornelius Wille & Fritz Zbrog
Pearson, 2003
- *Hypermedia & the Web – An Engineering Approach*
David Lowe & Wendy Hall
John Wiley & Sohn, 1999
- *Software for Use: A Practical Guide to the Models and Methods of UsageCentered Design*
Larry L. Constantine & Lucy A.D. Lockwood
ACM Press

Literatur

- *WebML*: Stefano Ceri, Piero Fraternali, Aldo Bongio, Marco Brambilla, Sara Comai, Maristella Matera. *Designing Data-Intensive Web Applications*. Morgan-Kaufmann (2002)
- *OO-H*: Jaime Gómez, Cristina Cachero, Oscar Pastor. On Conceptual Modeling of Device-Independent Web Applications: Towards a Web-Engineering Approach. *IEEE Multimedia*, 2001.
- *WSDM*: Olga De Troyer. *Audience-driven Web Design*, In *Information modelling in the new millennium*, IDEA Group Publishing, (2001)
- *WAE*: Jim Conallen. *Building Web Applications with UML*. Addison-Wesley, 2003.
- *UWE*: <http://www.pst.ifi.lmu.de/projekte/uwe>
- *ArgoUWE*: <http://www.pst.ifi.lmu.de/projekte/argouwe>