Die Technologie der Mumie

Tilman Rassy
Ruedi Seiler

Technischen Universität Berlin
Fakultät II – Mathematik und Naturwissenschaften
Institut für Mathematik

Vortrag an der ETH Zürich
27. Juli 2006

1 Design-Prizipien

TOC

1 Design-Prizipien

TOC

- Trennung von Layout und Inhalt

1 Design-Prizipien

TOC

- Trennung von Layout und Inhalt
- XML-Technologie

1 Design-Prizipien

TOC

- Trennung von Layout und Inhalt
- XML-Technologie
- Dynamische Seitenerzeugung

1 Design-Prizipien

TOC

- Trennung von Layout und Inhalt
- XML-Technologie
- Dynamische Seitenerzeugung
- Theme-Konzept

1 Design-Prizipien

TOC

- Trennung von Layout und Inhalt
- XML-Technologie
- Dynamische Seitenerzeugung
- Theme-Konzept
- Robustes Referenzierungssystem

1 Design-Prizipien

TOC

- Trennung von Layout und Inhalt
- XML-Technologie
- Dynamische Seitenerzeugung
- Theme-Konzept
- Robustes Referenzierungssystem
- Versionskontrolle

1 Design-Prizipien

TOC

- Trennung von Layout und Inhalt
- XML-Technologie
- Dynamische Seitenerzeugung
- Theme-Konzept
- Robustes Referenzierungssystem
- Versionskontrolle
- Java-Servlet-Technologie

2 Trennung von Layout und Inhalt

TOC

2 Trennung von Layout und Inhalt

TOC

- Inhalt enthält keine Layout-Informationen

2 Trennung von Layout und Inhalt

TOC

- Inhalt enthält keine Layout-Informationen
- ist unabhängig vom Kontext der Darstellung

2 Trennung von Layout und Inhalt

TOC

- Inhalt enthält keine Layout-Informationen
- ist unabhängig vom Kontext der Darstellung
- Layout-Informationen in CSS- und XSL-Stylesheets

3 XML-Technologie

TOC

3 XML-Technologie

TOC

- Nicht-binäre Inhalte als XML in der Datenbank

3 XML-Technologie

TOC

- Nicht-binäre Inhalte als XML in der Datenbank
- Gilt auch für CSS, JavaScript usw.

3 XML-Technologie

TOC

- Nicht-binäre Inhalte als XML in der Datenbank
- Gilt auch für CSS, JavaScript usw.
- Transformationen (mit XSL, Java)

3 XML-Technologie

TOC

- Nicht-binäre Inhalte als XML in der Datenbank
- Gilt auch für CSS, JavaScript usw.
- Transformationen (mit XSL, Java)
- Konfiguration XML-basiert

3 XML-Technologie

TOC

- Nicht-binäre Inhalte als XML in der Datenbank
- Gilt auch für CSS, JavaScript usw.
- Transformationen (mit XSL, Java)
- Konfiguration XML-basiert
- Build XML-basiert

4 Dynamische Seitenerzeugung

TOC

4 Dynamische Seitenerzeugung

TOC

- Materialien im Text-Format (XHTML, CSS, XSL, usw.) werden dynamisch erzeugt

4 Dynamische Seitenerzeugung

TOC

- Materialien im Text-Format (XHTML, CSS, XSL, usw.) werden dynamisch erzeugt
- DB $\longrightarrow$ XML $\longrightarrow$ ... $\longrightarrow$ XHTML

4 Dynamische Seitenerzeugung

TOC

- Materialien im Text-Format (XHTML, CSS, XSL, usw.) werden dynamisch erzeugt
- DB $\longrightarrow$ XML $\longrightarrow \ldots \longrightarrow$ XHTML
  DB $\longrightarrow$ XML $\longrightarrow \ldots \longrightarrow$ CSS

4 Dynamische Seitenerzeugung

TOC

- Materialien im Text-Format (XHTML, CSS, XSL, usw.) werden dynamisch erzeugt
- DB $\longrightarrow$ XML $\longrightarrow \ldots \longrightarrow$ XHTML
  DB $\longrightarrow$ XML $\longrightarrow \ldots \longrightarrow$ CSS
  DB $\longrightarrow$ XML $\longrightarrow \ldots \longrightarrow$ XSL

4 Dynamische Seitenerzeugung

TOC

- Materialien im Text-Format (XHTML, CSS, XSL, usw.) werden dynamisch erzeugt
- DB $\longrightarrow$ XML $\longrightarrow$ ... $\longrightarrow$ XHTML

  DB $\longrightarrow$ XML $\longrightarrow$ ... $\longrightarrow$ CSS

  DB $\longrightarrow$ XML $\longrightarrow$ ... $\longrightarrow$ XSL

  usw.

4 Dynamische Seitenerzeugung

TOC

- Materialien im Text-Format (XHTML, CSS, XSL, usw.) werden dynamisch erzeugt
- DB $\longrightarrow$ XML $\longrightarrow \ldots \longrightarrow$ XHTML

  DB $\longrightarrow$ XML $\longrightarrow \ldots \longrightarrow$ CSS

  DB $\longrightarrow$ XML $\longrightarrow \ldots \longrightarrow$ XSL

  usw.
- Ermöglicht maximale Benutzer-Adaptivität

5 Theme-Konzept

TOC

5 Theme-Konzept

TOC

- Mehrere sog. **Themes** möglich

5 Theme-Konzept

TOC

- Mehrere sog. **Themes** möglich
- Theme steuert Layout

5 Theme-Konzept

TOC

- Mehrere sog. **Themes** möglich
- Theme steuert Layout
- Realisierung mit Hilfe von „generischen“ und „realen“ Dokumenten

5 Theme-Konzept

TOC

- Mehrere sog. **Themes** möglich
- Theme steuert Layout
- Realisierung mit Hilfe von „generischen“ und „realen“ Dokumenten
- Generische Dokumente sind Platzhalter für reale, mit unterschiedlicher Implementierung für jedes Theme

5 Theme-Konzept

TOC

- Mehrere sog. **Themes** möglich
- Theme steuert Layout
- Realisierung mit Hilfe von „generischen“ und „realen“ Dokumenten
- Generische Dokumente sind Platzhalter für reale, mit unterschiedlicher Implementierung für jedes Theme
- Generisches Dokument plus Theme $\longrightarrow$ reales Dokument

6 Robustes Referenzierungssystem

TOC

6 Robustes Referenzierungssystem

TOC

- Referenzierungen durch sog. **Binnen-Ids**

6 Robustes Referenzierungssystem

TOC

- Referenzierungen durch sog. **Binnen-Ids**
- Binnen-Id plus Id des referenzierenden Dokuments
  $\longrightarrow$ Id des referenzierten Dokuments (DB-Tabelle)

6 Robustes Referenzierungssystem

TOC

- Referenzierungen durch sog. **Binnen-Ids**
- Binnen-Id plus Id des referenzierenden Dokuments
  $\longrightarrow$ Id des referenzierten Dokuments (DB-Tabelle)
- Keine hartkodierten DB-ID's im Inhalt

6 Robustes Referenzierungssystem

TOC

- Referenzierungen durch sog. **Binnen-Ids**
- Binnen-Id plus Id des referenzierenden Dokuments
  $\longrightarrow$ Id des referenzierten Dokuments (DB-Tabelle)
- Keine hartkodierten DB-ID's im Inhalt
- Einfacher Update der Referenzen

6 Robustes Referenzierungssystem

TOC

- Referenzierungen durch sog. **Binnen-Ids**
- Binnen-Id plus Id des referenzierenden Dokuments
  $\longrightarrow$ Id des referenzierten Dokuments (DB-Tabelle)
- Keine hartkodierten DB-ID's im Inhalt
- Einfacher Update der Referenzen
- Bei noch nicht eingecheckten Dokumenten: Dateinamen statt DB-Id's – Wichtig für Autoren-Tools

7 Versionskontrolle

TOC

7 Versionskontrolle

TOC

- MUMIE besitzt ein **Versionskontrollsystem**

7 Versionskontrolle

TOC

- MUMIE besitzt ein **Versionskontrollsystem**
- Alte Versionen eines Dokuments bleiben bestehen und können rekonstruiert werden

8 Java-Servlet-Technologie

TOC

8 Java-Servlet-Technologie

TOC

- Konzeptionelle Vorteile gegenüber anderen Technologien (CGI, PHP)

8 Java-Servlet-Technologie

TOC

- Konzeptionelle Vorteile gegenüber anderen Technologien (CGI, PHP)
- Gute XML-Unterstützung bei Java

9 Komponenten

TOC

9 Komponenten

TOC

- Web-Server

9 Komponenten

TOC

- Web-Server
  - **Apache**

9 Komponenten

TOC

- Web-Server
  - **Apache**
- Servlet Container

9 Komponenten

TOC

- Web-Server
  - **Apache**
- Servlet Container
  - **Tomcat**

9 Komponenten

TOC

- Web-Server
  - **Apache**
- Servlet Container
  - **Tomcat**
- Servlet

9 Komponenten

TOC

- Web-Server
  - **Apache**
- Servlet Container
  - **Tomcat**
- Servlet
  - **Cocoon**

9 Komponenten

TOC

- Web-Server
  - **Apache**
- Servlet Container
  - **Tomcat**
- Servlet
  - **Cocoon**
  - erweitert um MUMIE-spezifische Komponenten

9 Komponenten

TOC

- Web-Server
  - **Apache**
- Servlet Container
  - **Tomcat**
- Servlet
  - **Cocoon**
  - erweitert um MUMIE-spezifische Komponenten
- Datenbank

9 Komponenten

TOC

- Web-Server
  - **Apache**
- Servlet Container
  - **Tomcat**
- Servlet
  - **Cocoon**
  - erweitert um MUMIE-spezifische Komponenten
- Datenbank
  - **PostgreSQL**

9 Komponenten

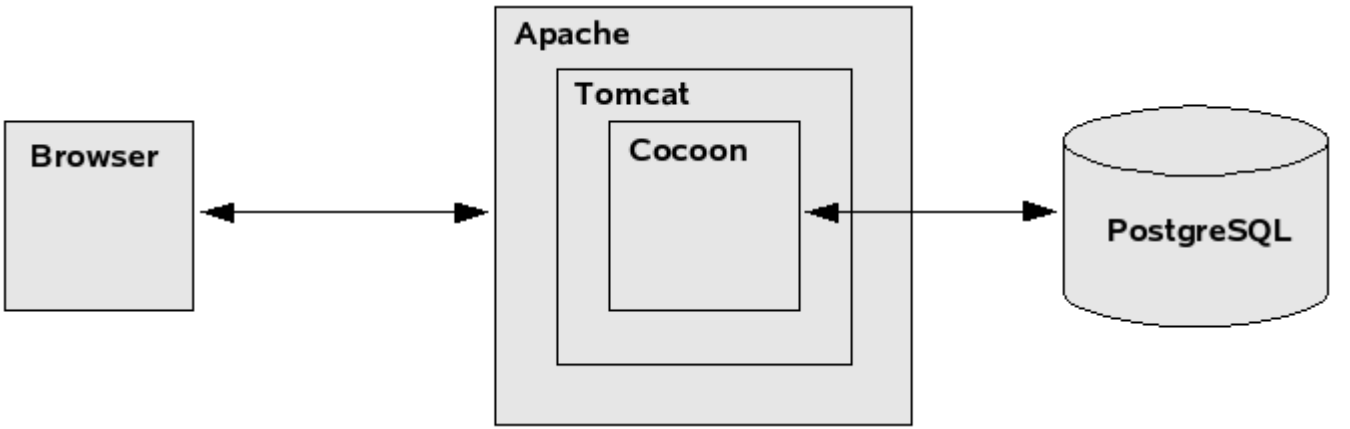
TOC

- Web-Server
  - **Apache**
- Servlet Container
  - **Tomcat**
- Servlet
  - **Cocoon**
  - erweitert um MUMIE-spezifische Komponenten
- Datenbank
  - **PostgreSQL**

Standard-Software, Open Source

10 Komponenten (Fortsetzung)

TOC



11 Cocoon

TOC

11 Cocoon

TOC

- XML-basiertes „Web Development Framework“

11 Cocoon

TOC

- XML-basiertes „Web Development Framework“
- Transformiert Dokumente durch eine sog. **Pipeline**

11 Cocoon

TOC

- XML-basiertes „Web Development Framework“
- Transformiert Dokumente durch eine sog. **Pipeline**
- Wichtigste Pipeline-Komponenten:

11 Cocoon

TOC

- XML-basiertes „Web Development Framework“
- Transformiert Dokumente durch eine sog. **Pipeline**
- Wichtigste Pipeline-Komponenten:
  - **Generator:** erzeugt Input-XML

11 Cocoon

TOC

- XML-basiertes „Web Development Framework“
- Transformiert Dokumente durch eine sog. **Pipeline**
- Wichtigste Pipeline-Komponenten:
  - **Generator:** erzeugt Input-XML
  - **Transformer:** transformiert XML nach XML

11 Cocoon

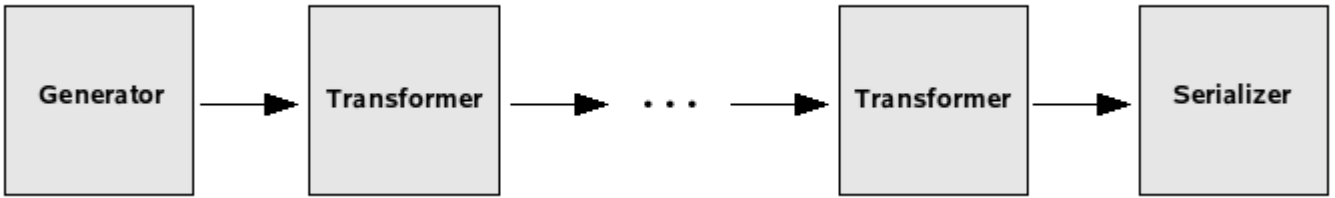
TOC

- XML-basiertes „Web Development Framework“
- Transformiert Dokumente durch eine sog. **Pipeline**
- Wichtigste Pipeline-Komponenten:
  - **Generator:** erzeugt Input-XML
  - **Transformer:** transformiert XML nach XML
  - **Serializer:** wandelt XML in Output-Format um

11 Cocoon

TOC

- XML-basiertes „Web Development Framework“
- Transformiert Dokumente durch eine sog. **Pipeline**
- Wichtigste Pipeline-Komponenten:
  - **Generator:** erzeugt Input-XML
  - **Transformer:** transformiert XML nach XML
  - **Serializer:** wandelt XML in Output-Format um



12 Typischer Ablauf bei einem Request

TOC

12 Typischer Ablauf bei einem Request

TOC

Beispiel: Browser sendet Request für generisches Dokument

12 Typischer Ablauf bei einem Request

TOC

Beispiel: Browser sendet Request für generisches Dokument

1. Auflösung generisches $\longrightarrow$ reales Dokument

12 Typischer Ablauf bei einem Request

TOC

Beispiel: Browser sendet Request für generisches Dokument

1. Auflösung generisches $\longrightarrow$ reales Dokument
2. XML-Darstellung des generischen Dokuments

12 Typischer Ablauf bei einem Request

TOC

Beispiel: Browser sendet Request für generisches Dokument

1. Auflösung generisches $\longrightarrow$ reales Dokument
2. XML-Darstellung des generischen Dokuments
   - umfasst Inhalt und Metainfos

12 Typischer Ablauf bei einem Request

TOC

Beispiel: Browser sendet Request für generisches Dokument

1. Auflösung generisches $\longrightarrow$ reales Dokument
2. XML-Darstellung des generischen Dokuments
   - umfasst Inhalt und Metainfos
3. Transformation(en)

12 Typischer Ablauf bei einem Request

TOC

Beispiel: Browser sendet Request für generisches Dokument

1. Auflösung generisches $\longrightarrow$ reales Dokument
2. XML-Darstellung des generischen Dokuments
   - umfasst Inhalt und Metainfos
3. Transformation(en)
   - Z.B. Hinzufügen benutzerspezifischer Daten

12 Typischer Ablauf bei einem Request

TOC

Beispiel: Browser sendet Request für generisches Dokument

1. Auflösung generisches $\longrightarrow$ reales Dokument
2. XML-Darstellung des generischen Dokuments
   - umfasst Inhalt und Metainfos
3. Transformation(en)
   - Z.B. Hinzufügen benutzerspezifischer Daten
   - ...

12 Typischer Ablauf bei einem Request

TOC

Beispiel: Browser sendet Request für generisches Dokument

1. Auflösung generisches $\longrightarrow$ reales Dokument
2. XML-Darstellung des generischen Dokuments
   - umfasst Inhalt und Metainfos
3. Transformation(en)
   - Z.B. Hinzufügen benutzerspezifischer Daten
   - ...
   - XSL ($\longrightarrow$ XHTML)

12 Typischer Ablauf bei einem Request

TOC

Beispiel: Browser sendet Request für generisches Dokument

1. Auflösung generisches $\longrightarrow$ reales Dokument
2. XML-Darstellung des generischen Dokuments
   - umfasst Inhalt und Metainfos
3. Transformation(en)
   - Z.B. Hinzufügen benutzerspezifischer Daten
   - ...
   - XSL ($\longrightarrow$ XHTML)
4. Auslieferung an Browser

13 Typischer Ablauf (Fortsetzung)

TOC

13 Typischer Ablauf (Fortsetzung)

TOC

- Nicht nur für XHTML-Seiten, auch für XSL-Stylesheets, CSS-Stylesheets usw.

Inhalt