



XML-Technologien SS14
Alte Gebäude am Nutzerstandort
Meilenstein-Präsentation

Projektgruppe 5
Institut für Informatik
Freie Universität Berlin

25. Juni 2014

Einführung

- Das Team

- Zielsetzung und Kriterien

- Ausgewählte Techniken

Server

- Manuelle Vorarbeit

- Design

Client

- Design

- Demonstration

Ausblick

- ▶ Martin Görick
- ▶ Tay Ho
- ▶ Ahmet-Serdar Karakaya
- ▶ Moritz Maxeiner
- ▶ Andre Plötze
- ▶ Frank Zechert

- ▶ Standortbasierte Anzeige von historischen Gebäudebildern¹

¹Berlinischen Galerie Landesmuseum für Moderne Kunst

- ▶ Standortbasierte Anzeige von historischen Gebäudebildern¹
- ▶ Kriterien:

¹Berlinischen Galerie Landesmuseum für Moderne Kunst

- ▶ Standortbasierte Anzeige von historischen Gebäudebildern¹
- ▶ Kriterien:
 - ▶ XML-Datenbank mit XQuery

¹Berlinischen Galerie Landesmuseum für Moderne Kunst

- ▶ Standortbasierte Anzeige von historischen Gebäudebildern¹
- ▶ Kriterien:
 - ▶ XML-Datenbank mit XQuery
 - ▶ XSLT

¹Berlinischen Galerie Landesmuseum für Moderne Kunst

- ▶ Standortbasierte Anzeige von historischen Gebäudebildern¹
- ▶ Kriterien:
 - ▶ XML-Datenbank mit XQuery
 - ▶ XSLT
 - ▶ Selbstdefinierte XML-Schemas

¹Berlinischen Galerie Landesmuseum für Moderne Kunst

- ▶ Standortbasierte Anzeige von historischen Gebäudebildern¹
- ▶ Kriterien:
 - ▶ XML-Datenbank mit XQuery
 - ▶ XSLT
 - ▶ Selbstdefinierte XML-Schemas
 - ▶ DBpedia-Daten (SPARQL)

¹Berlinischen Galerie Landesmuseum für Moderne Kunst

- ▶ Standortbasierte Anzeige von historischen Gebäudebildern¹
- ▶ Kriterien:
 - ▶ XML-Datenbank mit XQuery
 - ▶ XSLT
 - ▶ Selbstdefinierte XML-Schemas
 - ▶ DBpedia-Daten (SPARQL)
 - ▶ Webservice (REST/SOAP)

¹Berlinischen Galerie Landesmuseum für Moderne Kunst

- ▶ Java² als Programmiersprache für Server und Client

²<http://java.com/>

³<https://source.android.com/>

⁴<https://developer.android.com/google/play-services/index.html>

⁵<https://code.google.com/p/icesoap/>

⁶<http://www.eclipse.org/>

⁷<http://basex.org/>

- ▶ Java² als Programmiersprache für Server und Client
- ▶ Android³ als mobile Client-Plattform

²<http://java.com/>

³<https://source.android.com/>

⁴<https://developer.android.com/google/play-services/index.html>

⁵<https://code.google.com/p/icesoap/>

⁶<http://www.eclipse.org/>

⁷<http://basex.org/>

- ▶ Java² als Programmiersprache für Server und Client
- ▶ Android³ als mobile Client-Plattform
- ▶ Google Play Services⁴ für Lokalisierung

²<http://java.com/>

³<https://source.android.com/>

⁴<https://developer.android.com/google/play-services/index.html>

⁵<https://code.google.com/p/icesoap/>

⁶<http://www.eclipse.org/>

⁷<http://basex.org/>

- ▶ Java² als Programmiersprache für Server und Client
- ▶ Android³ als mobile Client-Plattform
- ▶ Google Play Services⁴ für Lokalisierung
- ▶ icesoap⁵ für SOAP-Anbindung unter Android

²<http://java.com/>

³<https://source.android.com/>

⁴<https://developer.android.com/google/play-services/index.html>

⁵<https://code.google.com/p/icesoap/>

⁶<http://www.eclipse.org/>

⁷<http://basex.org/>

- ▶ Java² als Programmiersprache für Server und Client
- ▶ Android³ als mobile Client-Plattform
- ▶ Google Play Services⁴ für Lokalisierung
- ▶ icesoap⁵ für SOAP-Anbindung unter Android
- ▶ Eclipse⁶ als Entwicklungsumgebung

²<http://java.com/>

³<https://source.android.com/>

⁴<https://developer.android.com/google/play-services/index.html>

⁵<https://code.google.com/p/icesoap/>

⁶<http://www.eclipse.org/>

⁷<http://basex.org/>

- ▶ Java² als Programmiersprache für Server und Client
- ▶ Android³ als mobile Client-Plattform
- ▶ Google Play Services⁴ für Lokalisierung
- ▶ icesoap⁵ für SOAP-Anbindung unter Android
- ▶ Eclipse⁶ als Entwicklungsumgebung
- ▶ BaseX⁷ als XML-Datenbank (XQuery,Java-API,Standalone)

²<http://java.com/>

³<https://source.android.com/>

⁴<https://developer.android.com/google/play-services/index.html>

⁵<https://code.google.com/p/icesoap/>

⁶<http://www.eclipse.org/>

⁷<http://basex.org/>

- ▶ Vorliegende CSV-Datensätze

⁸<https://www.flickr.com/>

- Vorliegende CSV-Datensätze
 - mit Geodaten aufbereitet

⁸<https://www.flickr.com/>

- ▶ Vorliegende CSV-Datensätze
 - ▶ mit Geodaten aufbereitet
 - ▶ zu XML konvertiert (mittels CsvToXml)

⁸<https://www.flickr.com/>

- ▶ Vorliegende CSV-Datensätze
 - ▶ mit Geodaten aufbereitet
 - ▶ zu XML konvertiert (mittels CsvToXml)
 - ▶ in BaseX importiert

⁸<https://www.flickr.com/>

- ▶ Vorliegende CSV-Datensätze
 - ▶ mit Geodaten aufbereitet
 - ▶ zu XML konvertiert (mittels CsvToXml)
 - ▶ in BaseX importiert
- ▶ Rohe Gebäudebilder von Flickr⁸ zu unserem Server exportiert

⁸<https://www.flickr.com/>

- ▶ Vorliegende CSV-Datensätze
 - ▶ mit Geodaten aufbereitet
 - ▶ zu XML konvertiert (mittels CsvToXml)
 - ▶ in BaseX importiert
- ▶ Rohe Gebäudebilder von Flickr⁸ zu unserem Server exportiert
- ▶ Teilbilder der Rohbilder in XML aufgestellt

⁸<https://www.flickr.com/>

- ▶ Vorliegende CSV-Datensätze
 - ▶ mit Geodaten aufbereitet
 - ▶ zu XML konvertiert (mittels CsvToXml)
 - ▶ in BaseX importiert
- ▶ Rohe Gebäudebilder von Flickr⁸ zu unserem Server exportiert
- ▶ Teilbilder der Rohbilder in XML aufgestellt
- ▶ XML-Überprüfer zur Validierung von XML-Daten mit den XML-Schemata geschrieben

⁸<https://www.flickr.com/>

Abbildung: XML-Schema für aufbereitete Rohbild-Daten

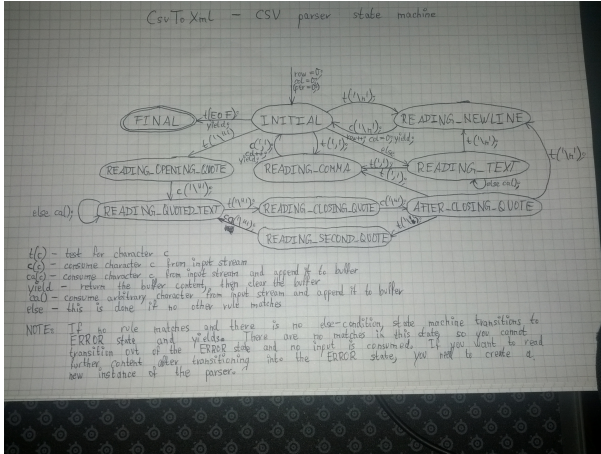
```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Foto">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Dateiname" type="xsd:string"/>
        <xsd:element name="flickerid" type="xsd:string"/>
        <xsd:element name="Date" type="xsd:string"/>
        <xsd:element name="Längengrad" type="xsd:string"/>
        <xsd:element name="Breitengrad" type="xsd:string"/>
        <xsd:element name="Radius" type="xsd:string" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```


Abbildung: XML-Schema für Teilbilder

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="Picture">
    <xsd:complexType mixed="true">
      <xsd:choice maxOccurs="unbounded" minOccurs="0">
        <xsd:element name="FlickerID" type="xsd:string" />
        <xsd:element name="Filename" type="xsd:string" />
        <xsd:element name="Section">
          <xsd:complexType mixed="true">
            <xsd:choice maxOccurs="unbounded" minOccurs="0">
              <xsd:element name="xValue" type="xsd:integer" />
              <xsd:element name="yValue" type="xsd:integer" />
              <xsd:element name="Height" type="xsd:integer" />
              <xsd:element name="Width" type="xsd:integer" />
            </xsd:choice>
            <xsd:attribute name="id"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Abbildung: CsvToXml - CSV Parser



- ▶ Standalone, keine Anwendungsserver (wie z.B. tomcat)

⁹Werkzeug wsgen, Annotationen

¹⁰Metadaten-Webservice noch Stub

- ▶ Standalone, keine Anwendungsserver (wie z.B. tomcat)
- ▶ Drei Collections in BaseX: geodata, imagedata, metadata

⁹Werkzeug wsgen, Annotationen

¹⁰Metadaten-Webservice noch Stub

- ▶ Standalone, keine Anwendungsserver (wie z.B. tomcat)
- ▶ Drei Collections in BaseX: geodata, imagedata, metadata
- ▶ Ein SOAP-Webservice pro XML-Collection⁹¹⁰

⁹Werkzeug wsgen, Annotationen

¹⁰Metadaten-Webservice noch Stub

- ▶ Gliederung in

- Gliederung in
 - Inhaltsbeschaffung

- Gliederung in
 - Inhaltsbeschaffung
 - Ortungsdienst

- ▶ Gliederung in
 - ▶ Inhaltsbeschaffung
 - ▶ Ortungsdienst
 - ▶ Informationsdarstellung

- ▶ Gliederung in
 - ▶ Inhaltsbeschaffung
 - ▶ Ortungsdienst
 - ▶ Informationsdarstellung
 - ▶ Testoberfläche

- ▶ Live-Demonstration der bisherigen Funktionalität

- Server
 - SOAP-Dienst für die Metadaten implementieren
 - Anbindung an die DBpedia für zusätzliche Metadaten
- Client
 - Metadaten vom Server abrufen und anzeigen
 - Automatischer Hintergrund-Dienst