Netzbasierte Informationssysteme

Rich Web Clients
Metadaten im Web

Prof. Dr.-Ing. Robert Tolksdorf
Freie Universität Berlin
Institut für Informatik
Netzbasierte Informationssysteme
mailto: tolk@inf.fu-berlin.de
http://www.robert-tolksdorf.de

W3C Rich Web Clients Activity

Freie Universität Berlin

- Das Web ist Plattform für Anwendungen jenseits reiner Informationssysteme
  - Online-Shops
  - Anwendungen wie Texterstellung
  - Lernanwendungen
  - Campus Management
  - ...

Rich Web Clients

- Verlagerung von Anwendungsteilen zusammen mit GUI zum Browser/Clienten
  - Applets
  - Javascript
  - …
- Teile der Anwendung „laufen“ im Browser
- Sie kombinieren verschiedene Medien zu einer Oberfläche

W3C Rich Web Clients Activity

Freie Universität Berlin

- Drei Arbeitsgruppen
  - Compound Document Formats Working Group
    - „Compound Document“: Zusammengesetztes Dokumenten aus mehreren Teilen in unterschiedlichen Sprachen
    - „develop specifications which combine ***selected*** existing document formats from the W3C and elsewhere, and which specify the runtime behaviour of such combined documents“
  - Web APIs Working Group
    - „develop specifications that enable improved client-side application development on the Web“
  - Web Application Formats Working Group
    - develop specifications that enable improved client-side application development on the Web. This includes the development of languages for applications, especially user interfaces
    - XUL, XAML

Zusammengesetzte Dokumente

Freie Universität Berlin

- Compound Document by Reference Framework 1.0
  - W3C Working Draft 22 November 2006
  - http://www.w3.org/TR/CDR/
- „Compound Document“: Dokument das aus mehreren Teildokumenten in unterschiedlichen Auszeichungssprachen zusammengesetzt ist
  - XHTML + SVG + MathML
  - XHTML + SMIL
  - XHTML + XForms
  - XHTML + VoiceML
- Problem: Jeweilige Spezifikationen decken nicht alle Aspekte solcher Kompositionen ab
- Compound Document by Reference Framework, CDRF ist generischer Rahmen für die Verarbeitung solcher Dokumente

XML Namensräume

Freie Universität Berlin

- Eine XML-Sprache wird durch eine URN bezeichnet
- Darin sind alle Sprachelemente eindeutig und definiert
- Mit XML-Namensräumen können in einem Dokument Elemente aus unterschiedlichen XML-Sprachen kombiniert werden wobei die Elementnamen eindeutig interpretierbar bleiben
- Möglichkeit 1:
  <element xmlns="URN"> legt fest, daß <element> ... </element> und alle Tags darin aus der durch URN bezeichneten XML-Sprache stammen
  - <math xmlns="http://www.w3.org/1998/Math/MathML">
    ...
    <math>

XML Namensräume

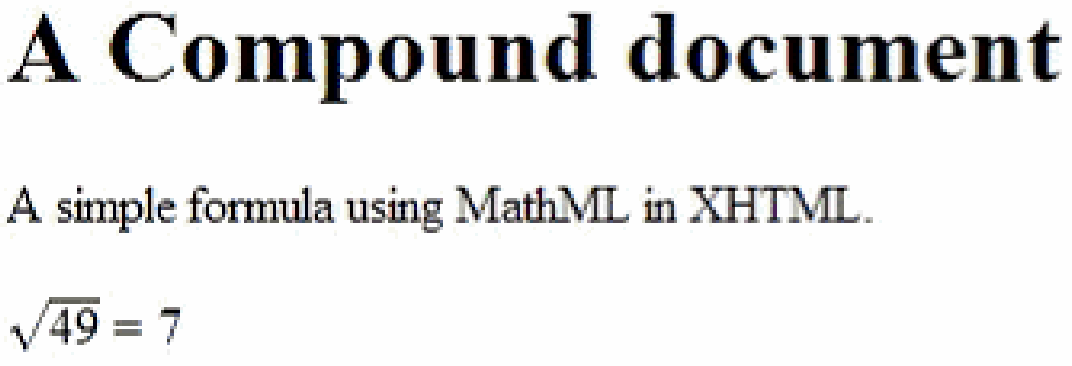
- Möglichkeit 2:
  <element xmlns:name="URN"> definiert ein Präfix name, das allen Tags aus der durch URN bezeichneten XML-Sprache vorangestellt werden
  - <xhtml:html
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:mathml="http://www.w3.org/1998/Math/MathML">

     <mathml:math>...</math>

Compound Document by Inclusion (CDI)

Freie Universität Berlin

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xhtml:html xmlns:xhtml="http://www.w3.org/1999/xhtml">
  <xhtml:body>
    <xhtml:h1>A Compound Document</xhtml:h1>
    <xhtml:p>A simple formula using MathML in XHTML.</xhtml:p>
    <mathml:math xmlns:mathml="http://www.w3.org/1998/Math/MathML">
      <mathml:mrow>
        <mathml:msqrt>
          <mathml:mn>49</mathml:mn>
        </mathml:msqrt>
        <mathml:mo>=</mathml:mo>
        <mathml:mn>7</mathml:mn>
      </mathml:mrow>
    </mathml:math>
  </xhtml:body>
</xhtml:html>
```



Compound Document by Reference (CDR)

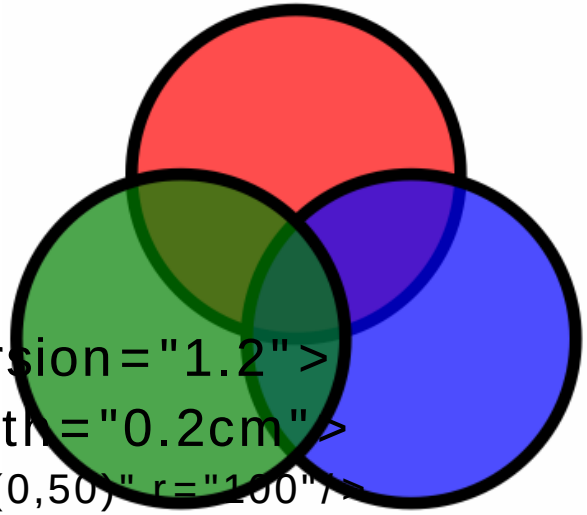
Freie Universität Berlin

- main.xhtml:

```
<?xml version="1.0"?>
<html xmlns="http://www.w3.org/1999/xhtml">
 <head><title>circles</title></head><body>
  <object height="350" width="600" type="image/svg+xml"
  data="circles.svg"/>
 </body></html>
```

- circle.svg:

```
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg"
   viewBox="0 0 100 100" baseProfile="tiny" version="1.2">
 <g fill-opacity="0.7" stroke="black" stroke-width="0.2cm">
  <circle fill="red" cx="6cm" cy="2cm" transform="translate(0,50)" r="100"/>
  <circle fill="blue" cx="6cm" cy="2cm" transform="translate(70,150)" r="100"/>
  <circle fill="green" cx="6cm" cy="2cm" transform="translate(-70,150)" r="100"/>
 </g>
</svg>
```



Wie werden Objekte referenziert?

- Existierende Mittel nutzen
- XHTML:

```
<object type="..." data="...">
  <param name="param1" value="true" />
  <param name="param2" value="123" />
</object>
```

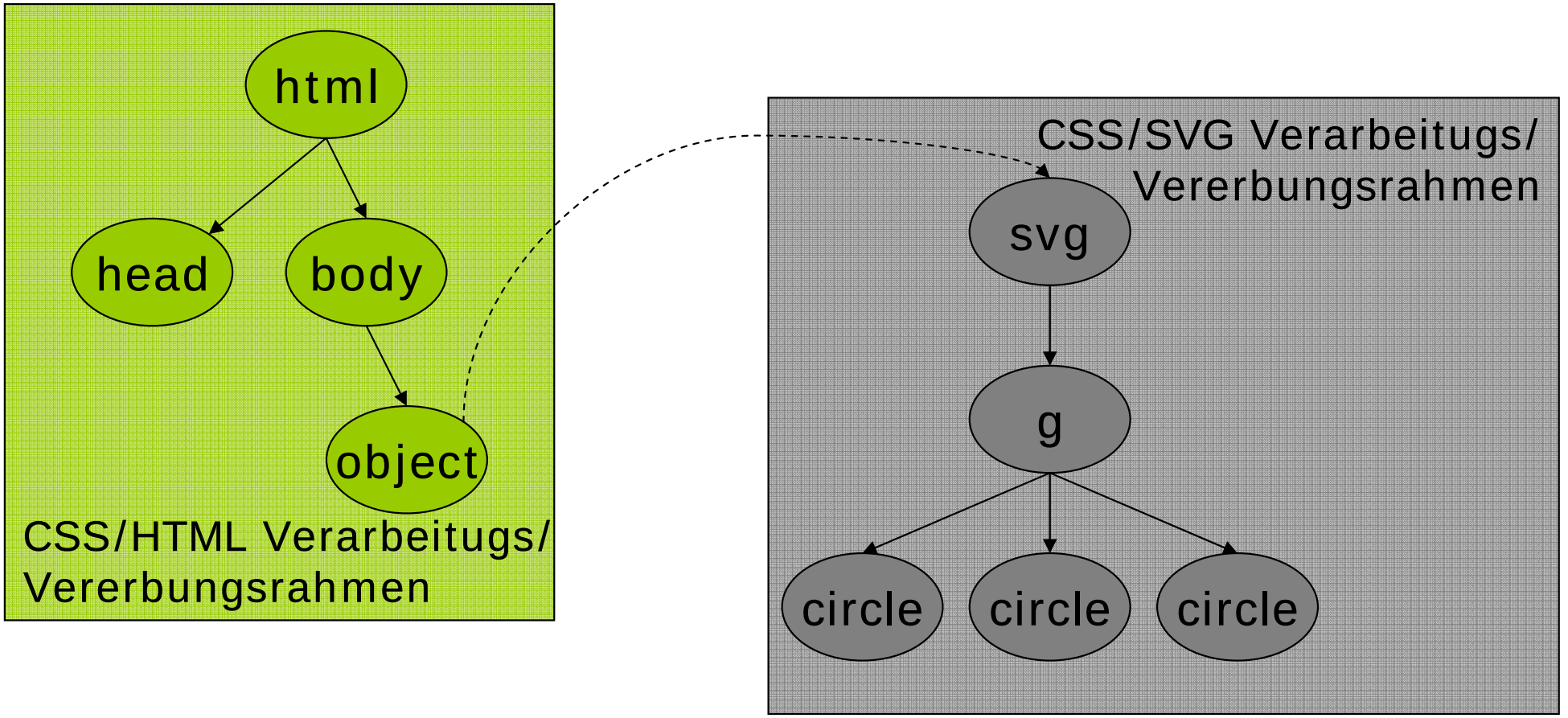
- SVG:

```
<svg xmlns="http://www.w3.org/2000/svg">
   <foreignObject>
      <metadata>
        param=value;  param=value
      </metadata>
   </foreignObject>
</svg>
```

- SMIL:

```
<ref src="http://www.example.com/herbert.face">
  <param name="mood" value="surly" valuetype="data"/>
  <param name="accessories" value="baseball-cap,nose-ring"
   valuetype="data"/>
</ref>
```

Wie sieht das Dokumentenmodell aus?



- Angleichung xhtml mobile, SVG mobile etc.?

DOM Zugang vom Kind aus?

- Window Object 1.0
  - W3C Working Draft 07 April 2006
  - http://www.w3.org/TR/Window
  - Definiert window Objekt das den Rahmen eines eingebetteten Dokuments DOM-seitig definiert

```
interface Window {
    // name attribute of referencing frame/iframe/object, or name
   passed to
    // window.open
    attribute dom::DOMString name;
    // global object of containing document
    readonly attribute Window parent;
    // global object of outermost containing document
    readonly attribute Window top;
    // referencing <html:frame>, <html:iframe>, <html:object>,
   <svg:foreignObject>,
    // <svg:animation> or other embedding point, or null if none
    readonly attribute dom::Element frameElement;
};
```

DOM Zugang vom Kind aus?

- Über frameElement Zugang zum DOM des “Elterndokumente”

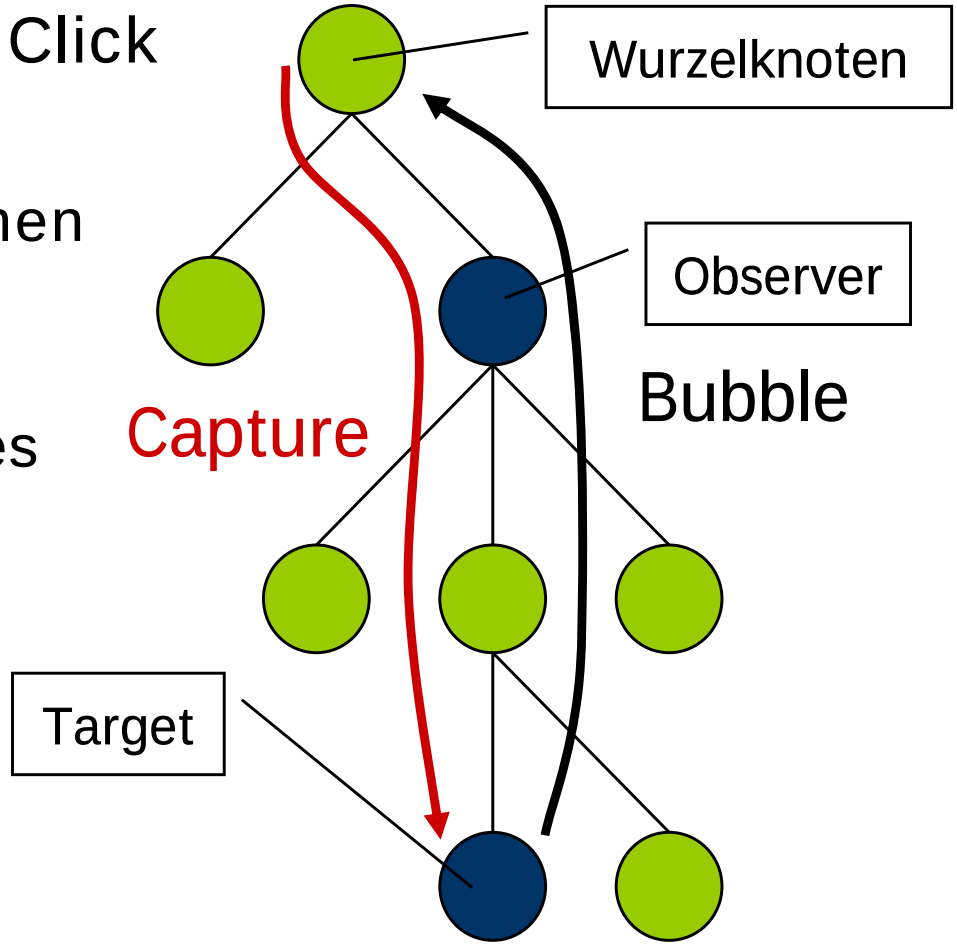
```
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg“
   height="20" version="1.1" width="20">
   <title>child-svg</title>
     <rect fill="blue" height="20"
          onload="alert('Child has seen: ' +
   document.defaultView.frameElement.ownerDocument.title)“
          width="20" x="10" y="10" />
</svg>
```

DOM Zugang zum Kind?

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>parent-xhtml</title>
  </head>
  <body>
    <object data="child.svg"
      onload="alert('Parent has seen: ' +
        this.contentDocument.getElementsByTagNameNS(
          'http://www.w3.org/2000/svg',
          'title') [0].textContent)"  />
  </body>
</html>
```

Was ist mit Events?

- Event: Asynchron auftretendes Ereignis, das auf ein Element (target) abzielt, z.B. Click
- Verarbeitung:
  - Capture-Phase: Herunterreichen des Ereignisses
  - Bubble-Phase: Herraufreichen des Ereignisses
- Ereignisse können jeweils
  - verarbeitet werden
  - gestoppt werden
- Verarbeitung ist *Action*
- *Handler* spezifiziert Action
- *Listener* bindet Ereignis an einem Ort mit Handler



Was ist mit Events?

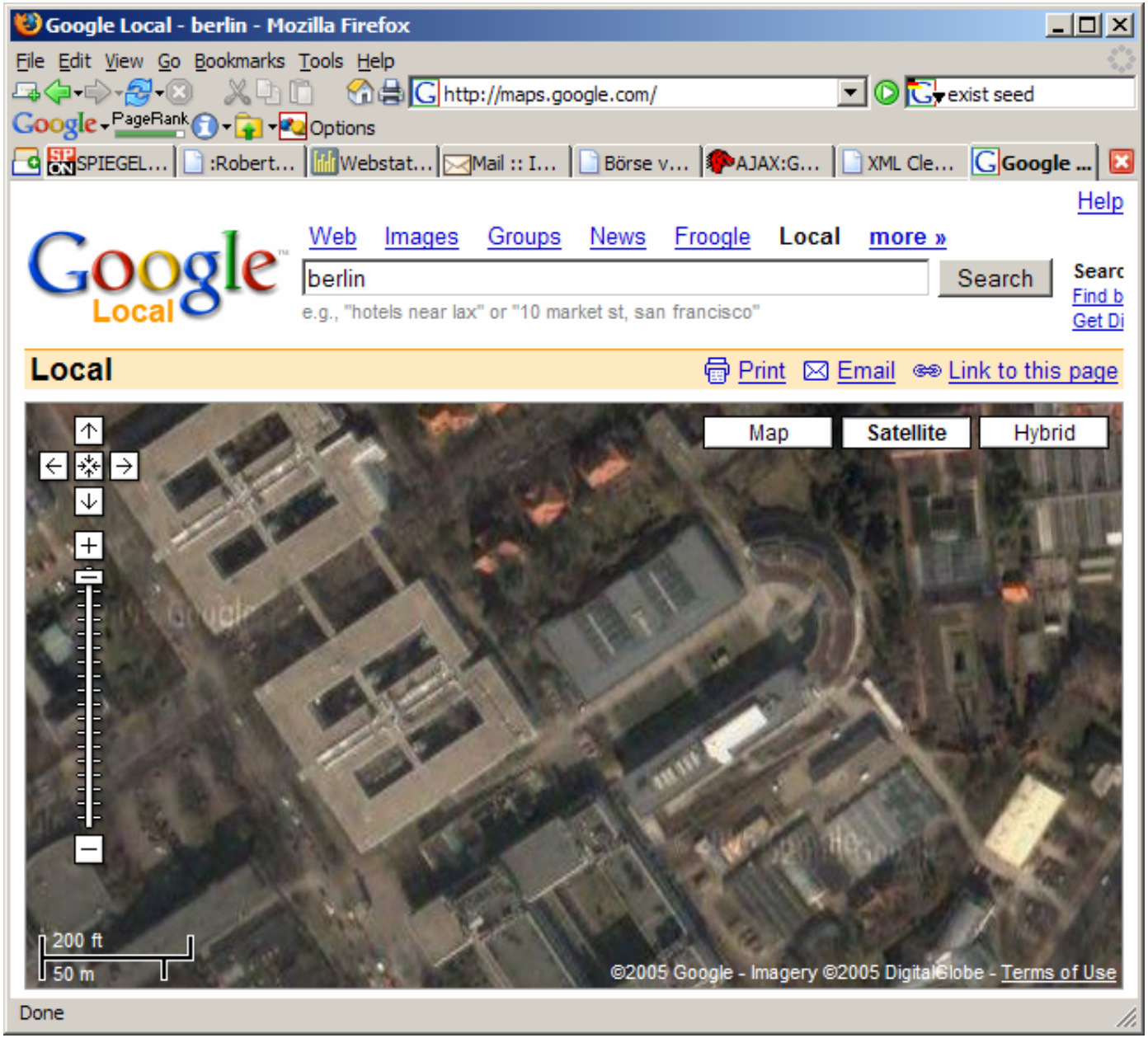
- Vom Kind- zum Elterndokument über frameElement weiterreichbar

```
var x = document.createEvent("CustomEvent");
x.initCustomEventNS("http://example.org/test", "test", true,
    false, null);
window.frameElement.dispatchEvent(x);
```

- Abbildung von sprachspezifischen Events auf DOM3 Events
- Gleiches Verhalten muss gesichert sein

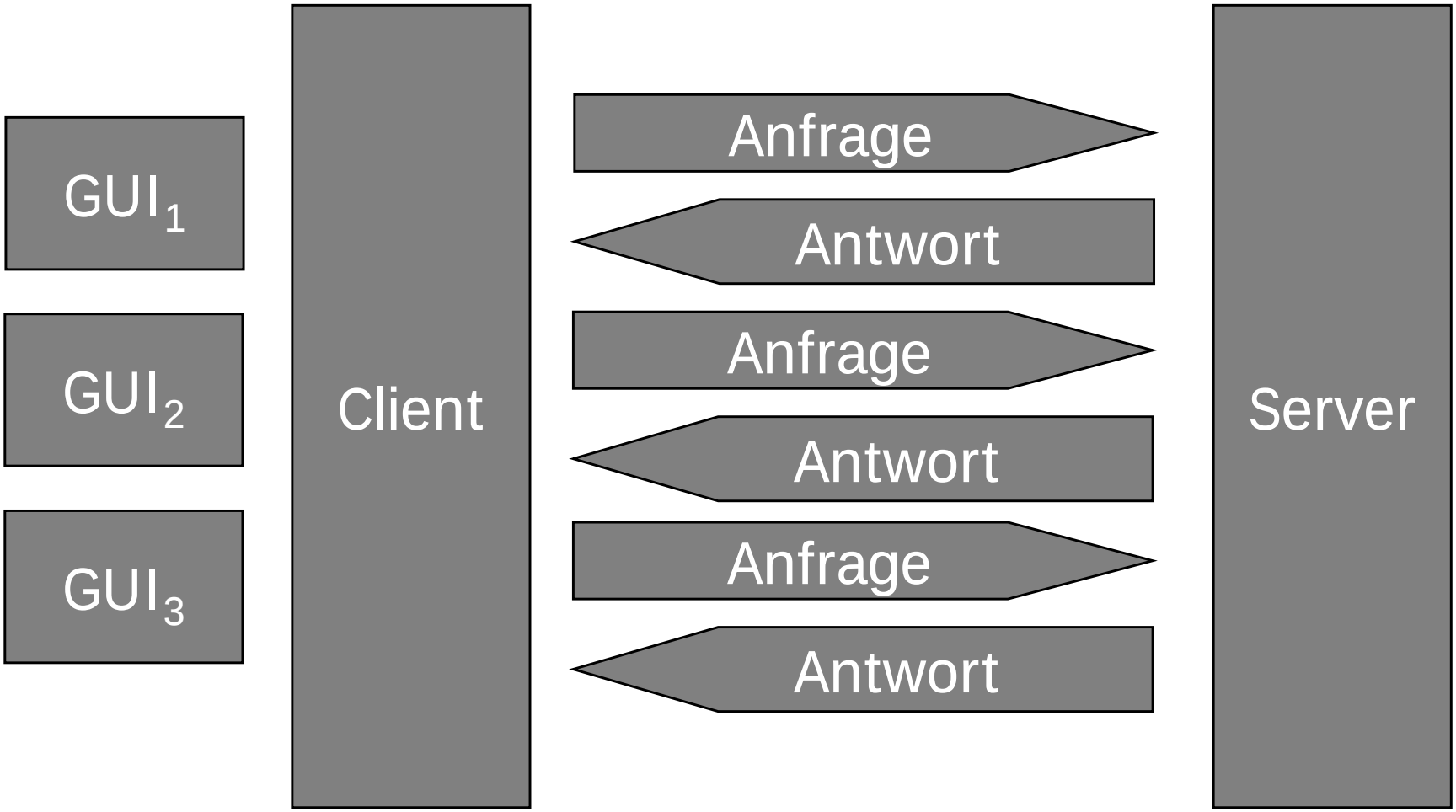
Google Maps

- Anwendung: Durchschauen von Karten und Bildern
- Arbeitet so als sei es eine Desktop-Anwendung mit nativen Mitteln der Oberfläche (vgl: Google Earth)



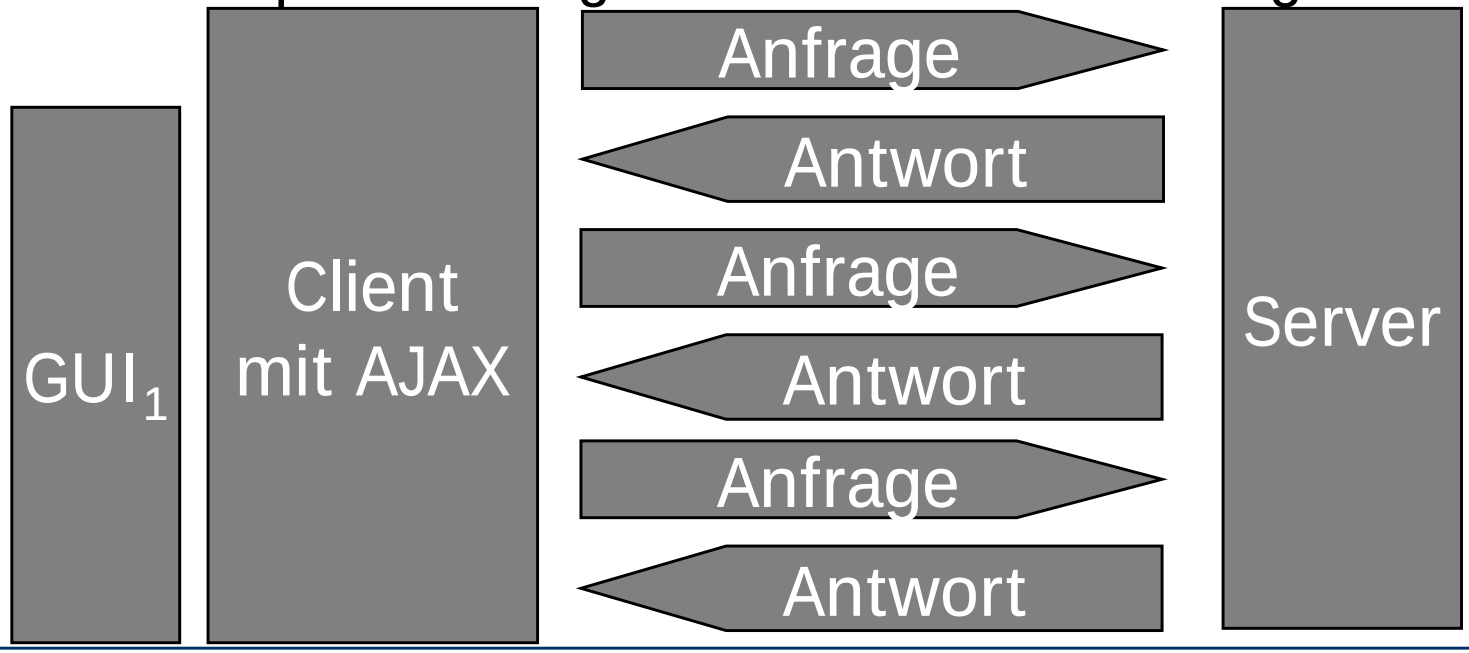
Freie Universität Berlin

- Problematik: Durch die Anfrage/Antwort Interaktion per HTTP wird bei jeder Nutzeraktion ein komplett neues GUI im Browser erzeugt



AJAX

- Asynchronous JavaScript and XML (AJAX) realisiert dies durch Kombination von
  - Präsentationssprachen XHTML und CSS
  - Interaktion und Modifikation im Browser mit DOM
  - Datenaustausch mit XML
  - Datentransfer durch asynchrone HTTP-Anfragen
  - Javascript als Integration dieser Technologien



XMLHttpRequest Object

Freie Universität Berlin

- The XMLHttpRequest Object
  - W3C Working Draft 27 September 2006
  - http://www.w3.org/TR/XMLHttpRequest/
  - “Special thanks to the Microsoft employees who first implemented the XMLHttpRequest interface, which was first widely deployed by the Windows Internet Explorer browser.”

- Definiert Schnittstelle und Verhalten eines Objektes das in Browsern für Scriptsprachen zugänglich sein soll
- Dieses Objekt kann HTTP Anfragen bearbeiten
- Die Anfragen können asynchron zur weiteren Scriptverarbeitung ausgeführt werden

„IDL“ des XMLHttpRequest Objekts

Freie Universität Berlin

```
interface XMLHttpRequest {
                  attribute EventListener   onreadystatechange;
  readonly        attribute unsigned short  readyState;
  void            open(in DOMString method, in DOMString url);
  void            open(in DOMString method, in DOMString url, in boolean async);
  void            open(in DOMString method, in DOMString url, in boolean async,
                       in DOMString user);
  void            open(in DOMString method, in DOMString url, in boolean async,
                       in DOMString user, in DOMString password);
  void            setRequestHeader(in DOMString header, in DOMString value);
  void            send();
  void            send(in DOMString data);
  void            send(in Document data);
  void            abort();
  DOMString       getAllResponseHeaders();
  DOMString       getResponseHeader(in DOMString header);
  readonly attribute DOMString      responseText;
  readonly attribute Document       responseXML;
  readonly attribute unsigned short  status;
  readonly attribute DOMString      statusText;
};
```

Anfragezustände

Freie Universität Berlin

- Anfrage durchläuft 5 Zustände
  - readonly attribute unsigned short readyState;
  - 0, Uninitialisiert
    XMLHttpRequest Objekt erzeugt
  - 1, Open
    Objekt initialisiert (open() Methode aufgerufen)
  - 2, Sent
    HTTP Anforderung gesendet (send() Methode aufgerufen)
  - 3, Receiving
    Antwort wird empfangen
  - 4, Loaded
    Antwort ist empfangen
- Bei jedem Zustandswechsel wird eine Methode aufgerufe, der “Handler”
  - attribute EventListener onreadystatechange;

Eine URL überprüfen

- Mit XMLHttpRequest Objekt HEAD Methode absetzen
- Nach Beendigung HTTP Antwortcode holen:

```
function fetchStatus(address) {
 var client = new XMLHttpRequest();
 client.onreadystatechange = function() {
  // in case of network errors this might not give reliable results
  if(this.readyState == 4)
   returnStatus(this.status);
 }
 client.open("HEAD", address);
 client.send();
}
```

- Hallo sagen:

```
function ping(message) {
 var client = new XMLHttpRequest();
 client.open("POST", "/ping");
 client.send(message);
}
```

Objekt initialisieren

- In Zustand 0: Initialisierung des Objekts
- open(method, url, async, user, password)
  - method: HTTP-Methode (inkl. WebDAV):
    GET, POST, HEAD, PUT, DELETE, PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK, VERSION-CONTROL, REPORT, CHECKOUT, CHECKIN, UNCHECKOUT, MKWORKSPACE, UPDATE, LABEL, MERGE, BASELINE-CONTROL, MKACTIVITY, ORDERPATCH, ACL
  - url: URL
  - async: Soll die Anfrage asynchron ablaufen (Default: ja)
  - user, password: Eventuell Nutzer/Password
- Zustand danach 1, open

Anfrage vorbereiten

- In Zustand 1: Setzen von Anfrage-Headern
- setRequestHeader(header, value)
  - HTTP Header setzen
  - Verschiedene Einschränkungen
    - Nichts tun bei Accept-Charset, Accept-Encoding, Content-Length, Expect, Date, Host, Keep-Alive, Referer, TE, Trailer, Transfer-Encoding, Upgrade
    - Vorhandenen Wert überschreiben bei uthorization, Content-Base, Content-Location, Content-MD5, Content-Range, Content-Type, Content-Version, Delta-Base, Depth, Destination, ETag, Expect, From, If-Modified-Since, If-Range, If-Unmodified-Since, Max-Forwards, MIME-Version, Overwrite, Proxy-Authorization, SOAPAction, Timeout
    - Sonst vorhandenen Wert kombinieren
      - client.setRequestHeader('X-Test', 'one');
client.setRequestHeader('X-Test', 'two');
      - X-Test: one, two

Anfrage stellen

- In Zustand 1: Absenden der Anfrage
- void send();
  void send(in DOMString data);
  void send(in Document data);
  - Absenden der HTTP Anfrage mit entsprechenden Header
  - Eventuell mit Inhalt als Anfragebestandteil
    - Content-Type setzen
    - application/xml sonst
  - Folgen von Umleitungen (Antworten 301, 302, 303, 307 und Location:/URI: Header)
- Zustand
  - Nach Absetzen der Anfrage: 2, Sent
  - Vor Empfang des Antwortinhalts: 3, Receiving
  - Nach Empfang des Antwortinhalts: 4, Loaded

Antwort-Header abfragen

- In Zustand 3 oder 4: Abfrage der Antwort-Header
- getAllResponseHeaders()

```
var client = new XMLHttpRequest();
client.open("GET", "test.txt", true);
client.send();
client.onreadystatechange = function() {
 if(this.readyState == 3) {
  print(this.getAllResponseHeaders());
 }
}
```

- Ausgabe:
  - Date: Sun, 24 Oct 2004 04:58:38 GMT
  - Server: Apache/1.3.31 (Unix)
  - Keep-Alive: timeout=15, max=99
  - Connection: Keep-Alive
  - Transfer-Encoding: chunked
  - Content-Type: text/plain; charset=utf-8
- getResponseHeader(header)

Antwort abfragen

- In Zustand 3 oder 4: Abfrage des Antwort-Inhalts (-fragments)
  - readonly attribute DOMString responseText;
- In Zustand 4: Abfrage des Antwort-Inhalts
  - readonly attribute Document responseXML;

- In Zustand 3 oder 4: Abfrage des HTTP Antwortcodes
  - readonly attribute unsigned short status;
  - readonly attribute DOMString statusText;

Antwort-Code berücksichtigen

- „main()“

```
var client = new XMLHttpRequest();
client.onreadystatechange = handler;
client.open("GET", "test.xml");
client.send();
```

- Management

```
function handler() {
 if(this.readyState == 4 && this.status == 200) {// so far so good
  if(this.responseXML != null &&
     this.responseXML.getElementById('test').firstChild.data)
   // success!
   test(this.responseXML.getElementById('test').firstChild.data);
  else
   test(null);
 } else if (this.readyState == 4 && this.status != 200) {
  test(null); // fetched the wrong page or network error...
 }
}
```

- Verarbeitung

```
function test(data) {// taking care of data
}
```

Metadaten im Web

Ermittlung der Semantik von Dokumenten

Freie Universität Berlin

- Ermittlung der Bedeutung von Dokumenten:
  - Manuelles Indexing: Manuelle Termvergabe
  - Automatisches Indexing: Automatische Termvergabe auf statistischer Basis
  - Filtering: Indirekt durch Einschätzung der Bedeutung für Nutzer
  - Textverstehen: Computerlinguistische Verfahren
- Explizite Bekanntgabe der Bedeutung von Dokumenten
  - Inhaltsinformationen: Textueller Inhalt
  - Objektive Metainformationen: Datum, Größe…
  - Inhaltliche Metainformationen: Term
- Durch vorgefundene Metainformationen erübrigt sich die Ermittlung von Metainformationen
- Dezentrale Bereitstellung

Syntakische und semantische Verweise in HTML

Freie Universität Berlin

- Syntaktisch:
  Berlin

- Beziehung durch Link gegeben, aber:
  - Welche inhaltliche Beziehung besteht zwischen Quell- und Zielanker?
  - Was ist die Bedeutung des Verweis?

- Semantische Information:
  Ich wohne in <a href="www.berlin.de"
  role="wohnort">Berlin.
- Schema zum gemeinsamen Verständnis ist nötig

Dublin Core

Metadaten in HTML: Dublin Core

Freie Universität Berlin

- "Dublin Core" (http://purl.org/metadata/dublin_core) ist der Versuch, ein verbreitetes Schema für Metadaten zu etablieren:

- "The Dublin Core Metadata Initiative is an open forum engaged in the development of interoperable online metadata standards that support a broad range of purposes and business models. DCMI's activities include consensus-driven working groups, global workshops, conferences, standards liaison, and educational efforts to promote widespread acceptance of metadata standards and practices."

- Dublin Core Metadata Element Set, 1.1: Reference Description
  - Quelle: http://dublincore.org/documents/1999/07/02/dces/
  - Status: DCMI Recommendation 1999-07-02
- Dublin Core Metadata for Resource Discovery
  - Status: IETF RFC 2413, September 1998
- Encoding Dublin Core Metadata in HTML
  - Status: IETF RFC 2731, December 1999

meta und link

- Semantische Informationen in HTML/XHTML:
  - meta Tag: Beschreibung eines Aspekts eines Dokuments
    <meta name="DC.Creator" content= "Tolksdorf, Robert">
  - link Tag: Verweis auf Beschreibung des verwendeten Schemas der Aspekte
    <link rel="schema.DC"
    href="http://purl.org/DC/elements/1.0/">
- Dublin Core legt mögliche Namen und Werte von Metainformationen für Verwendung in meta fest

Dublin Core Elemente

- Title: Titel des Dokuments
  <meta name="DC.Title" lang="es"
  content="La Mesa Verde y la Silla Roja" />

- Creator: Erzeuger des Dokuments
  <meta name="DC.Creator"
  content="Gogh, Vincent van" />
- Contributor: Jemand, der beigetragen hat
  <meta name="DC.Contributor"
  content="Curie, Marie">

- Publisher: Der die Resource verfügbar macht
  <meta name="DC.Publisher" content="O'Reilly">

Dublin Core Elemente

- Subject: Thema des Dokuments
  <meta name="DC.Subject"
  scheme="MESH" content=
  "Myocardial Infarction; Pericardial Effusion" />
- Description: Inhaltsbeschreibung
  <meta name="DC.Description" content=
  "A tutorial and reference manual for Java.">
- Type: Art des Dokuments
  <meta name="DC.Type" content="web home page">
  <meta name="DC.Type" scheme="DCT1"
  content="dataset">

- Coverage: Gültigkeitsbereich des Dokuments
  <meta name="DC.Coverage" content=
  "Columbus, Ohio, USA; Lat: 39 57 N Long: 082 59 W">

Dublin Core Elemente

- Date: Ein für das Dokument wichtiges Datum
  <meta name="DC.Date.Created"
  content="1998-05-14">
  <meta name="DC.Date.Available"
  content="1998-05-21">
  <meta name="DC.Date.Valid"
  content="1998-05-28">
- Format: Repräsentation des Dokuments
  <meta name="DC.Format" scheme="IMT"
  content="image/jpeg">
- Rights: Aussage über Rechte
  <meta name="DC.Rights" content=
  "Copyright Acme 1999 - All rights reserved.">

Dublin Core Elemente

- Identifier: Bezeichner des Dokuments

```
<meta name="DC.Identifier"
      content="http://foo.bar.org/zaf">
<meta name="DC.Identifier"
      content="urn:ietf:rfc:1766">
<meta name="DC.Identifier"
      scheme="ISBN"
      content = "1-56592-149-6">
```

- Source: Quelle der Information

```
<meta name="DC.Source" content=
      "Shakespeare's Romeo and Juliet">
```

Dublin Core Elemente

- Language: Sprache des Dokuments

```
<meta name="DC.Language"
      scheme="rfc1766" content="en">
```

- Relation: Beziehung zu anderen Dokumenten

```
<meta name="DC.Relation.IsPartOf" content=
  "http://foo.bar.org/abc/proceedings/1998/">
<meta name="DC.Relation.IsVersionOf"
   content="http://foo.bar.org/draft9.4.4.2">
<meta name="DC.Relation.References"
      content="urn:isbn:1-56592-149-6">
<meta name="DC.Relation.IsBasedOn" content=
      "Shakespeare's Romeo and Juliet">
```

Simple HTML Ontology Extensions
SHOE

Luke, S., L. Spector, D. Rager, and J. Hendler. Ontology-based Web Agents. In *Proceedings of the First International Conference on Autonomous Agents (Agents97)*. W. L. Johnson, ed. Association for Computing Machinery, New York, 1997, 59-66.

http://www.cs.umd.edu/projects/plus/SHOE/

Informationen in HTML Seiten textuell

Freie Universität Berlin

```
<HTML>
<HEAD>
<TITLE> My Page </TITLE>
</HEAD>
<BODY>
<P> Hi, this is my web page.
    I am a graduate student and a research assistant.
<P> Also, I'm 52 years old.
<P> My name is George Stephanopolous.
<P> Here is a pointer to my <A
 HREF="http://www.cs.umd.edu/smith"> graduate advisor.</A>
<P> And <A HREF="http://www.cs.umd.edu/papers/paper.ps">
 is a paper I recently wrote.
<h3> Brun Hilda </h3>
Brun Hilda is a visiting lecturer here from Germany who doesn't have her
own web page.  However, because I am such a nice person, I have agreed
to let part of my web page space belong to her.  She is 23.
</BODY>
</HTML>
```

Informationen formalisieren

Freie Universität Berlin

- Simple HTML Ontology Extensions definiert zusätzliche HTML Elemente zur Markierung inhaltlicher Zusammenhänge
  - <META HTTP-EQUIV="SHOE" CONTENT="VERSION=1.0">
- „Instances“ sind die beschriebenen Datenobjekte
- Identifiziert durch einen Schlüssel, eine URI
  - <INSTANCE KEY="http://www.cs.umd.edu/users/george/">
- Zur Beschreibung werden Konzepte und Beziehungen genutzt, die in einer „Ontologie“ defininiert sind
  - <USE-ONTOLOGY ID="cs-dept-ontology" URL="http://www.cs.umd.edu/projects/plus/SHOE/onts/cs.html" VERSION="1.0" PREFIX="cs">
  - „cs.“ als Präfix für verwendete Konzeptnamen

Klassifikation und Beziehungen

- Klassifikation der Instanz
  - „I am a graduate student and a research assistant."
  - <CATEGORY NAME="cs.GraduateStudent">
    <CATEGORY NAME="cs.ResearchAssistant">
- Beziehungen beschreiben
  - "<P> Also, I'm 52 years old.<P> My name is George Stephanopolous."
  - <RELATION NAME="cs.name">
    <ARG POS=1
    VALUE="http://www.cs.umd.edu/users/george/">
    <ARG POS=2 VALUE="George Stephanopolous">
    </RELATION>
    <RELATION NAME="cs.age">
    <ARG POS=1
    VALUE="http://www.cs.umd.edu/users/george/">
    <ARG POS=2 VALUE="52">
    </RELATION>

Beziehungen

- “<P> Here is a pointer to my <A HREF="http://www.cs.umd.edu/smith"> graduate advisor.</A>”
- <RELATION NAME="cs.advisor">
  <ARG POS=TO
  VALUE="http://www.cs.umd.edu/users/smith">
  </RELATION>

- „<P> And <A HREF="http://www.cs.umd.edu/papers/paper.ps"> is a paper I recently wrote.„
- <RELATION NAME="publicationAuthor">
  <ARG POS=FROM
  VALUE="http://www.cs.umd.edu/papers/paper.ps">
  </RELATION>

- Geschachtelte Instanz
  - „Brun Hilda is a visiting lecturer here from Germany who doesn't have her own web page. However, because I am such a nice person, I have agreed to let part of my web page space belong to her. She is 23.“
  - <INSTANCE
    KEY="http://www.cs.umd.edu/users/george/#BRUNHILDA">
    <CATEGORY NAME="cs.Lecturer">
    <RELATION NAME= "cs.name">
    <ARG POS=TO VALUE="Brun Hilda">
    </RELATION>
    <RELATION NAME="cs.age">
    <ARG POS=TO VALUE="23">
    </RELATION>
    </INSTANCE>

HTML Informationen semantisch beschrieben

Freie Universität Berlin

```
<HTML> <HEAD>
 <META HTTP-EQUIV="SHOE" CONTENT="VERSION=1.0">
 <TITLE> My Page </TITLE> ...
to let part of my web page space belong to her. She is 23.
<INSTANCE KEY="http://www.cs.umd.edu/users/george/">
 <USE-ONTOLOGY  ID="cs-dept-ontology"
        URL="http://www.cs.umd.edu/projects/plus/SHOE/onts/cs.html"
        VERSION="1.0"   PREFIX="cs">
 <CATEGORY NAME="cs.GraduateStudent">
 <CATEGORY NAME="cs.ResearchAssistant">
 <RELATION NAME="cs.name">
      <ARG POS=TO VALUE="George Stephanopolous">
 </RELATION>
 <RELATION NAME="cs.age"><ARG POS=TO VALUE="52"></RELATION>
 <RELATION NAME="cs.advisor">
      <ARG POS=TO VALUE="http://www.cs.umd.edu/users/smith">
 </RELATION>
 <INSTANCE KEY="http://www.cs.umd.edu/users/george/#BRUNHILDA">
      <CATEGORY NAME="cs.Lecturer">
      <RELATION NAME= "cs.name">
           <ARG POS=TO VALUE="Brun Hilda">
      </RELATION>
      <RELATION NAME="cs.age"><ARG POS=TO VALUE="23"></RELATION>
 </INSTANCE>
</INSTANCE>
```

Stimmt das alles?

- Die Aussagen müssen nicht wahr sein
- Jeder kann andere und falsche Aussagen treffen
  - <RELATION NAME="cs.age"><ARG POS=TO VALUE="53"></RELATION>
  - <CATEGORY NAME=" cs.GraduateStudent" FOR="http://www.cs.umd.edu/users/smith">
- Im Web gilt die „open world“ Annahme
  - Wissen ist unvollständig
  - Nicht ableitbare Sachverhalte sind unbekannt
  - Closed world Annahme: Nicht Nicht ableitbare Sachverhalte sind falsch
- Statt Fakten werden Behauptungen ins Netz gestellt
- Immerhin!

- Falsche Behauptungen werden im realen Leben sozial sanktioniert
  - Reputation sinkt
  - Verweigerung von Mehrwert

Ontologien - Deklarationen

- Ontologien in SHOE
  - Klassen in IS-A Hierarchie
  - Direkte Beziehungen zwischen Klassen
  - Regeln als Horn-Klauseln

- Deklaration der Ontologie selber
  - <ONTOLOGY ID="cs-dept-ontology" VERSION="1.0">
  - Name ist “cs-dept-ontology”
- Sie verwendet Konzepte aus einer anderen Ontologie
  - <USE-ONTOLOGY ID="base-ontology" VERSION="1.0"
    PREFIX="base"
    URL="http://www.cs.umd.edu/projects/plus/SHOE/base.html">
  - Konzepte aus dieser Ontologie werden mit vorangestelltem “base.” verwendet

Ontologien - Klassen

- Definitionen von Klassen

```
<DEF-CATEGORY NAME="Organization" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="Person" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="Publication" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="ResearchGroup" ISA="Organization">
<DEF-CATEGORY NAME="Department" ISA="Organization">
<DEF-CATEGORY NAME="Worker" ISA="Person">
<DEF-CATEGORY NAME="Faculty" ISA="Worker">
<DEF-CATEGORY NAME="Assistant" ISA="Worker">
<DEF-CATEGORY NAME="AdministrativeStaff" ISA="Worker">
<DEF-CATEGORY NAME="Student" ISA="Person">
<DEF-CATEGORY NAME="PostDoc" ISA="Faculty">
<DEF-CATEGORY NAME="Lecturer" ISA="Faculty">
<DEF-CATEGORY NAME="Professor" ISA="Faculty">
<DEF-CATEGORY NAME="ResearchAssistant" ISA="Assistant">
<DEF-CATEGORY NAME="TeachingAssistant" ISA="Assistant">
<DEF-CATEGORY NAME="GraduateStudent" ISA="Student">
<DEF-CATEGORY NAME="UndergraduateStudent" ISA="Student">
<DEF-CATEGORY NAME="Secretary" ISA="AdministrativeStaff">
<DEF-CATEGORY NAME="Chair" ISA="AdministrativeStaff Professor">
```

Ontologien – getypte Beziehungen

Freie Universität Berlin

- Studenten werden von Professoren betreut

```
<DEF-RELATION NAME="advisor">
    <DEF-ARG POS="1" TYPE="Student">
    <DEF-ARG POS="2" TYPE="Professor">
</DEF-RELATION>
```

- Personen sind Mitglieder von Organisatonen

```
<DEF-RELATION NAME="member">
    <DEF-ARG POS="1" TYPE="Organization">
    <DEF-ARG POS="2" TYPE="Person">
</DEF-RELATION>
```

- Personen erstellen Publikationen

```
<DEF-RELATION NAME="publicationAuthor">
    <DEF-ARG POS="1" TYPE="Publication">
    <DEF-ARG POS="2" TYPE="Person">
</DEF-RELATION>
```

Ontologien – getypte Beziehungen

- Publikationen haben ein Erscheinungsdatum

```
<DEF-RELATION NAME="publicationDate">
    <DEF-ARG POS="1" TYPE="Publication">
    <DEF-ARG POS="2" TYPE=".DATE">
</DEF-RELATION>
```

- Personen haben ein ganzzahliges Alter

```
<DEF-RELATION NAME="age">
    <DEF-ARG POS="1" TYPE="Person">
    <DEF-ARG POS="2" TYPE=".NUMBER">
</DEF-RELATION>
```

- Dinge können benannt sein

```
<DEF-RELATION NAME="name">
    <DEF-ARG POS="1" TYPE="base.SHOEEntity">
    <DEF-ARG POS="2" TYPE=".STRING">
</DEF-RELATION>
```

- Professoren können auf Lebenszeit ernannt sein

```
<DEF-RELATION NAME="tenured">
    <DEF-ARG POS="1" TYPE="Professor">
    <DEF-ARG POS="2" TYPE=".TRUTH">
</DEF-RELATION>
```

Inferenzen/Schlussfolgerungen

- Aus Aussagen können Schlüsse gezogen werden
- „Wenn jemand in einem Tochterunternehmen eines Konzerns beschäftigt ist, dann ist er/sie im Konzern beschäftigt“
- Als Horn-Klausel
  - member(?org2,?person) :- member(?org1,?person) ^ subOrganizationOf(?org1,?org2)

- *Suchmaschinen können nun beantworten, wer in einem bestimmten Konzern arbeitet auch wenn es kein zentrales Personenverzeichnis gibt*
  - Höheres Recall
  - Höheres Precision

Inferenzen/Schlussfolgerungen

Freie Universität Berlin

- In SHOE

```
<DEF-INFERENCE DESCRIPTION="member(?org2,?person)
                            if member(?org1,?person)
                            and subOrganizationOf(?org1,?org2)">
    <INF-IF>
        <RELATION NAME="member">
            <ARG POS=1 VALUE="org1" USAGE=VAR>
            <ARG POS=2 VALUE="per" USAGE=VAR>
        </RELATION>
        <RELATION NAME="subOrganizationOf">
            <ARG POS=1 VALUE="org1" USAGE=VAR>
            <ARG POS=2 VALUE="org2" USAGE=VAR>
        </RELATION>
    </INF-IF>
    <INF-THEN>
        <RELATION NAME="member">
            <ARG POS=1 VALUE="org2" USAGE=VAR>
            <ARG POS=2 VALUE="per" USAGE=VAR>
        </RELATION>
    </INF-THEN>
</DEF-INFERENCE>
```

Inferenzen/Schlussfolgerungen

- „Studierende der Informatik an der Univ. Maryland werden immer von Prof. John Doe betreut“

```
<DEF-INFERENCE
   DESCRIPTION="advisor(?per,http://www.cs.umd.edu/users/johndoe.html)
                if member(http://www.cs.umd.edu,?per)
                and UnderGraduateStudent(?per)">
      <INF-IF>
            <RELATION NAME="member">
                  <ARG POS=1 VALUE="http://www.cs.umd.edu/">
                  <ARG POS=2 VALUE="per" USAGE=VAR>
            </RELATION>
            <CATEGORY NAME="UndergraduateStudent" FOR="per"
   USAGE=VAR>
      </INF-IF>
      <INF-THEN>
            <RELATION NAME="advisor">
                  <ARG POS=1 VALUE="per" USAGE=VAR>
                  <ARG POS=2
                        VALUE="http://www.cs.umd.edu/users/johndoe.html">
            </RELATION>
      </INF-THEN>
</DEF-INFERENCE>
```

Inferenzen/Schlussfolgerungen

- Für Datentypen (.STRING, .NUMBER, .DATE, .TRUTH) auch Vergleiche definiert

```
<DEF-INFERENCE>
        <INF-IF>
                <RELATION NAME="age">
                        <ARG POS=1 VALUE="per" USAGE=VAR>
                        <ARG POS=2 VALUE="a" USAGE=VAR>
                </RELATION>
                <COMPARISON OP="greaterThanOrEqual">
                        <ARG POS=1 VALUE="a" USAGE=VAR>
                        <ARG POS=2 VALUE="18">
                </COMPARISON>
        </INF-IF>
        <INF-THEN>
                <RELATION NAME="canEnter">
                        <ARG POS=1 VALUE="per" USAGE=VAR>
                        <ARG POS=2 VALUE="YES">
                </RELATION>
        </INF-THEN>
</DEF-INFERENCE>
```

SHOE Projekt

- http://www.cs.umd.edu/projects/plus/SHOE/:
  „Note: This SHOE web site is no longer being actively maintained. Work at the University of Maryland on web ontologies continues in the Semantic Web and Agents Project, which uses the Web Ontology Languages OWL and DAML+OIL. These languages are results of standardization efforts that are in part based on SHOE (Jim Hendler is coChair of the Web Ontology Working Group sponsored by the World Wide Web Consortium.
  Jeff Heflin and Sean Luke are now faculty members at Lehigh University and George Mason University, respectively. Like Dr. Hendler, Dr. Heflin is exploring the use of DAML+OIL and OWL, but is also continuing to build tools for SHOE as well. Information about his research and the latest SHOE developments can be found here.

- -> Semantic Web Initiative des W3C