openHPI Web Technologies 2015

Week 6

Prof. Dr. Christoph Meinel
Hasso Plattner Institute
University of Potsdam, Germany

openHPI

The Web Today and Tomorrow: Social, Semantic, and Service Web

Prof. Dr. Christoph Meinel
Hasso Plattner Institute
University of Potsdam, Germany

The Web Today and Tomorrow (1/3)

Evolution of the World Wide Web in the last 25 years shaped today's web through numerous transformations

- **Web 1.0: Read-only or Static Web** (before 2000)
  - Only few (expert) authors were able to publish content
  - Web resources were mostly static
- **Web 2.0: Read-Write or Participation Web** (until now)
  - Users became able to publish own content through social web applications ➔ the **Social Web**
  - (Dynamic) web sites arise and become more and more complex
  - New technologies allow developers to build feature-rich web-based applications
- …

The Web Today and Tomorrow (2/3)

Evolution of the World Wide Web in the last 25 years shaped today's web through numerous transformations

- …
- **Web 3.0: Semantic and Execution Web** (still emerging)
  - Information becomes machine readable and understandable
    Example:
    - Today's search engines use keyword-based mechanisms
    - Search algorithms use string comparison and don't understand the meaning of search terms and documents
  - The **Semantic Web** no longer consists of documents but of **things** (i.e. people, places, events, …) and their **relation**
  - Web applications become inter-connectable through web services
    ➔ the **Service Web**
- …

The Web Today and Tomorrow (3/3)

Evolution of the World Wide Web in the last 25 years
shaped today's web through numerous transformations

- …
- **Web 4.0: Mobile Web** (today)
  - Parallel, alternative version of the existing web
  - Adds greater importance to location context
  - Largely uses the → Service Web (i.e. for server-side part of apps)
- **And beyond?**

  What could **Web 5.0** look like?

  ➔ Discuss your ideas in the forum!

Social, Semantic, and Service Web

Will focus on **three aspects** of today's and tomorrow's web

- **Social Web**
  - What are social web applications?
  - How did they emerge?
- **Semantic Web**
  - How do today's search engines work and what are their limitations?
  - What are semantics and the principles of Semantic Web?
- **Service Web**
  - How do Web Services (the enabling technology for the service web) work?

openHPI

Social Web and Social Media

Prof. Dr. Christoph Meinel
Hasso Plattner Institute
University of Potsdam, Germany

Emergence of Social Web

Users' perception and usage of the Internet changes around 2005 …

- **Participation Principle** of the Web 2.0
  - Users are editors and not only readers of static web information
  - Users share their data resources in the Web, e.g. bookmarks, photos, professional information, videos, …
  - Community feeling
- Web became increasingly used as a communication platform and the boundary blurs between Web applications and applications running on the local computing device (PC)
  - so-called "Rich" Internet applications are developed
- **Example**:
  - *Gmail* instead of using a local mail client

Wikis

So-called **wikis** are an early form of social web applications

- Wiki is Hawaiian for "quick"
- Realize the idea of collaborative text editing and commenting others' contributions in the WWW

- First wiki installations were already released in 1995 (WikiWikiWeb)
- In 2001, the online encyclopedia **Wikipedia** was launched and advanced the popularity of the concept
  - Wikipedia was listed amongst the top 10 most popular websites
- Wikis became popular as enterprise knowledge bases, too
  - Employees could share their knowledge and persist it

Blogging

- **Web-blogs** or, shortly, **blogs** are another implementation of the social web. They emerged at the end of the 1990s
- Blogs are personal diaries or journals publicly published in the web. One or more authors – the Web-Logger(s) or, shortly, **Blogger(s)** contribute postings (**posts**) about their live or a dedicated topic
- Blog readers can actively participate by adding feedback as comments, ratings, …
- Users can subscribes blogs by **feeds** (e.g. RSS, Atom)
- Technically, Blogs are realized by
  - blog hosting services, e.g. **Blogger**, **tumblr.**, …
  - hosting it on an own Server, e.g. **WordPress**

tumblr.

WordPress

Microblogging

- **Microblogging** was introduced for casual personal or corporate status updates. Here, posts have a size of 200 character or less
- Posts can be created publicly or privately by means of a
  - browser, app, email, or instant messaging
- For microblogs, the concept of **asynchronous followers** was introduced: Followers receive status updates of the microblogger, they are following, but not vice versa
- Currently the most popular microblogging service is **Twitter**. Twitter has started 2006 and is meanwhile an important media channel for public persons, political activists, …
- To identify trending topics in the social web, **#Hashtags** are used

Social Networks

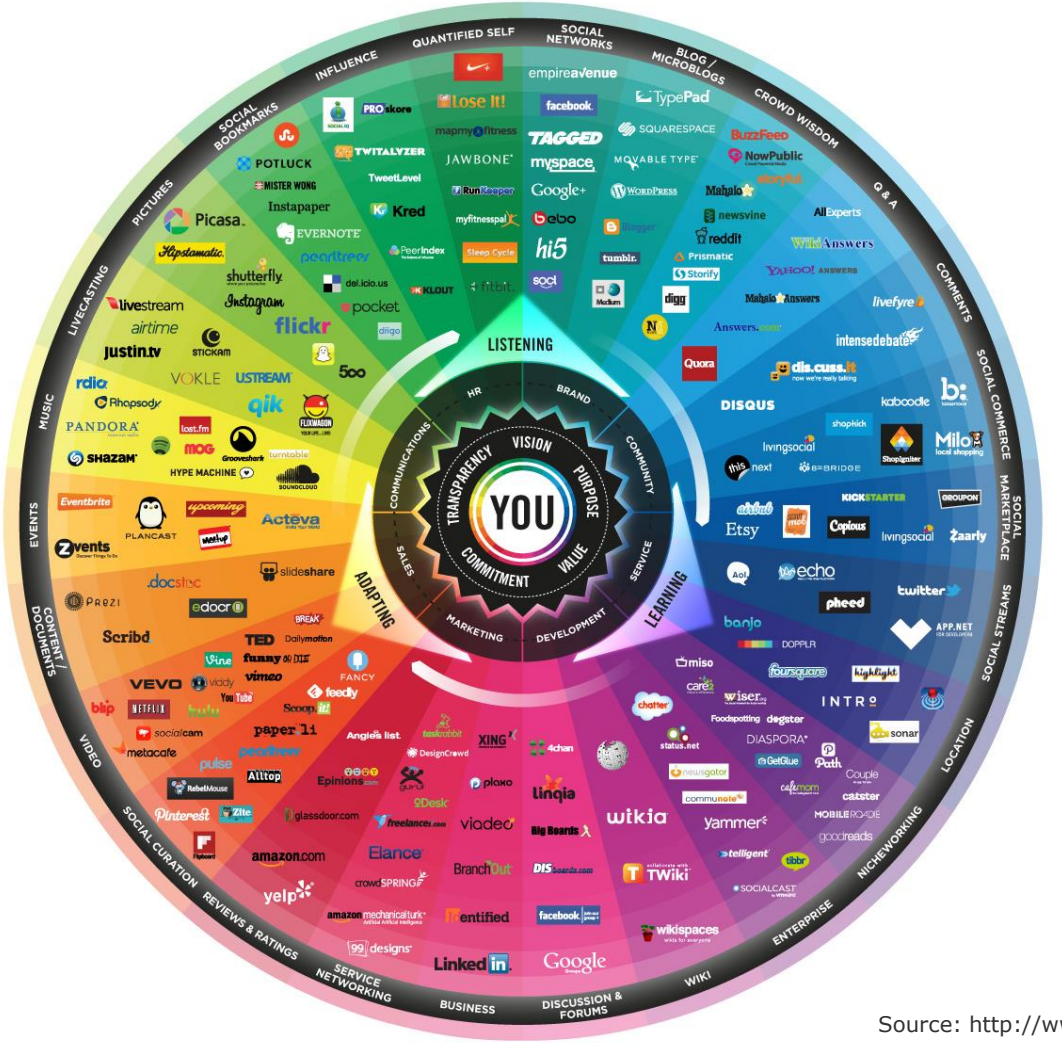
Social networking services or, shortly, **social networks** have been established together with microblogging

- Social networks combine blogging / microblogging and add a list of additional common features:
  - Personal profile
  - Friend list (in contrast to the follower concept social networks establish bi-directional user relationships)
  - Activity stream (combines status updates of a user's friends and allows commenting)
  - Private messaging with other users or user groups
  - Express personal interests, e.g. "likes"
  - Social applications (e.g. games) → Social networks become platforms for own applications

Social Media

- Due to dropping prices for storage and high speed bandwidth, various social media portals have been introduced since 2005
- For users social media offer to share their media resources like images, audios, or videos with friends or the public
  - Authors can get feedback for their media
  - Authors can store media for inclusion in their personal website without considering storage issues
- Today's popular social media services are:
  - Images – **Instagram**, **flickr**, …
  - Audio – **Soundcloud**, …
  - Video – **YouTube**, **Vimeo**, …

Social Web – Multitude of Variations



Source: http://www.conversationprism.com

Critical Discussions about Social Media

Privacy

- Social network providers often claim extensive usage rights on user contributed content
- Personal information can be used for personalized advertisement and sold to advertisers for targeting

Trustworthiness

- Everyone can create correct or incorrect content, e.g. "Edit Wars" on Wikipedia
- Fake accounts under wrong names can be created

Reliability

- Services can be blocked by countries to limit access to information

…

openHPI

Introduction to Search Engines

Prof. Dr. Christoph Meinel
Hasso Plattner Institute
University of Potsdam, Germany

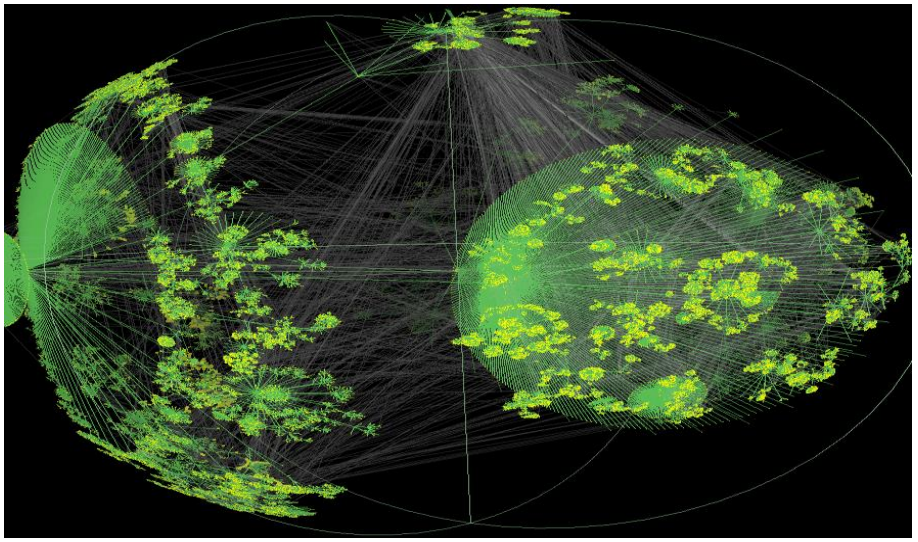
Finding the Needle in the Haystack (1/2)

WWW provides access to an enormously huge amount of information

- Until July 2008, Google alone found a trillion (10^{12} = 1.000.000.000.000) linked URLs …
  - Since then, Google does no longer publish this number

It is nearly impossible to estimate the current number of documents

- Document collection in the WWW doubles approximately every 6 months
- It is not very likely that this trend will come to an end soon
- Search engines help to find the relevant information



Finding the Needle in the Haystack (2/2)

There are many different **search engines** on the market that offer different ways to search the web:

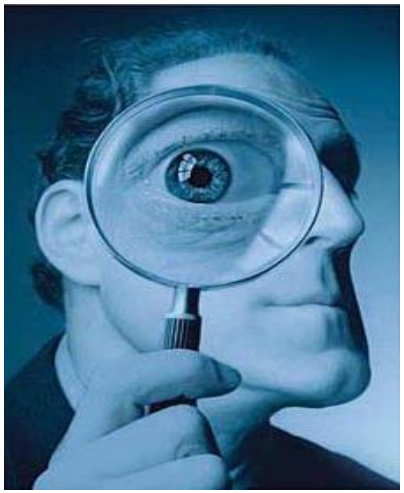
Search Services on the Web

Search services should fulfill the following requirements:

- Support users in information retrieval in the WWW
- Build most extensive collection of data possible, ideally including all documents in the WWW
- Assign specific key terms to the individual WWW documents
- Speedily find every information provider – even new ones that may not have been listed so far

Most important criteria for users:

- Completeness (Recall)
  - Find **all relevant** documents
- Accuracy (Precision)
  - Find **only relevant** documents



Various Types of Search Engine (1/2)

- **Web catalogues** (catalogue-based search engines)
  - Web documents are organized in categories by human editors
  - Users can browse the categories
- **Index-based search engines**
  - Automated agents (crawlers) visit websites by following hyperlinks
  - Website content is analyzed, important terms are stored as keywords
- **Meta-search engines**
  - Search different search engines
  - Combine their top results
- **Paid placement search engines**
  - Information provider pays for being listed

Various Types of Search Engine (2/2)

- …
- **Topic-oriented search engines**
  - Shopping
  - News
  - People
  - Computer / Internet
  - Entertainment
  - Business
  - Science
  - …
- **Regional search engines**
- **Blog and RSS search engines**

openHPI

Index-based Search Engines and Crawlers

Prof. Dr. Christoph Meinel
Hasso Plattner Institute
University of Potsdam, Germany

Index-Based Search Engines

Today the most popular search engines are index-based ones, e.g.

Index is a data structure for mapping descriptors (e.g. keywords, relevant terms) to documents

Basic functions of index-based search engines:

- Data collection
- Document analysis and document review
- Generation and administration of index data structures
- Answering queries involving relevance values

Index-Based Search Engines: Data Collection – What?

Data diversity of the WWW:

- Static HTML documents
- Dynamically generated HTML documents
- Multimedia data (images, audio, video files)
- Postscript, PDF, Word, PowerPoint documents, etc.
- Source code of software
- and many more …

Index-Based Search Engines
Data Collection – When?

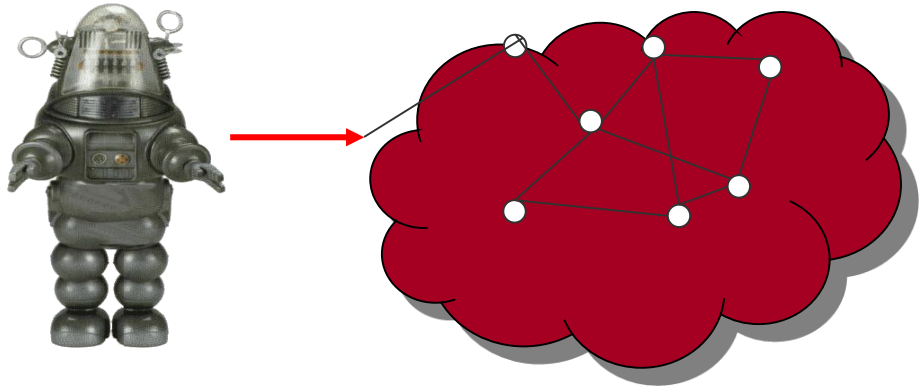
The WWW lives …

- Data and documents in the knowledge space WWW
  - … usually only have a “short” lifetime
  - … are subject to constant change
  - … are connected to other documents via links and depend on them

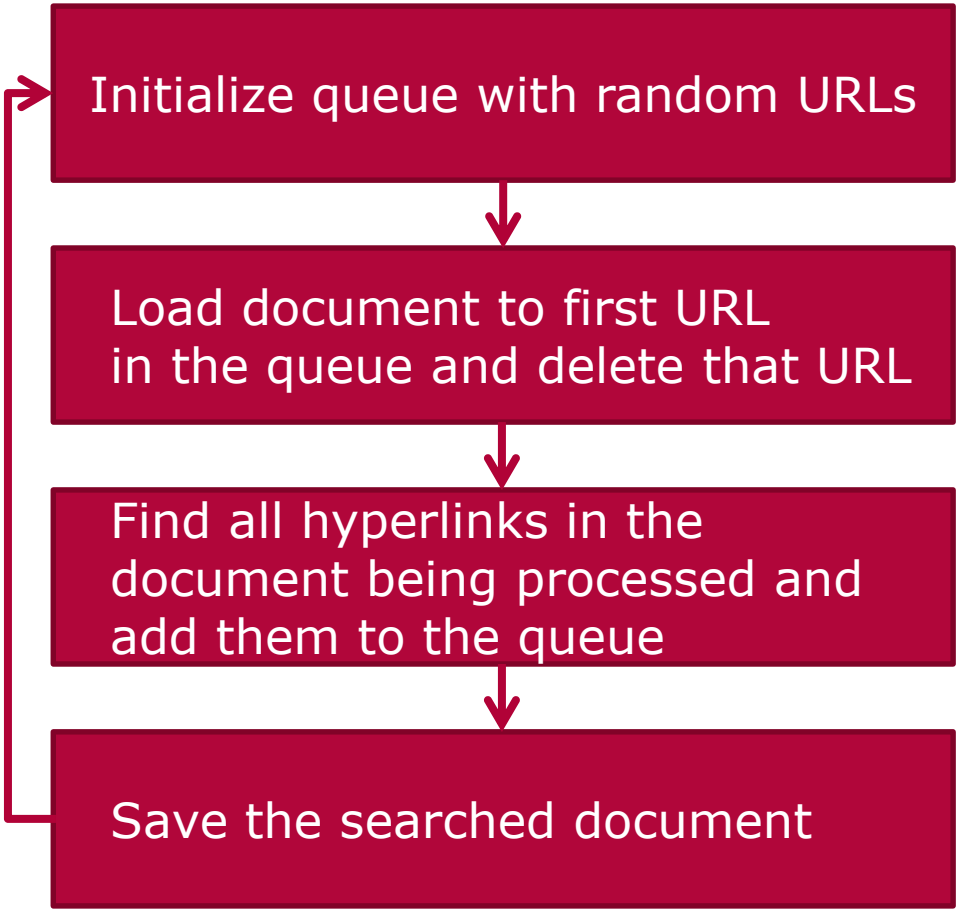
➔ Captured database must be **maintained at periodic intervals**

Index-Based Search Engines: Data Collection – How?

- Data from the WWW are collected by means of special, autonomously working software tools:
  - **web robots** or **robots** or **web crawler** or **crawler**
- Crawler can automatically find web pages and documents in the WWW by following the URLs behind links (e.g. in other documents)
- Documents found by crawlers are stored in a database
- Documents already stored in the database must be periodically checked for consistency / changes / deletion

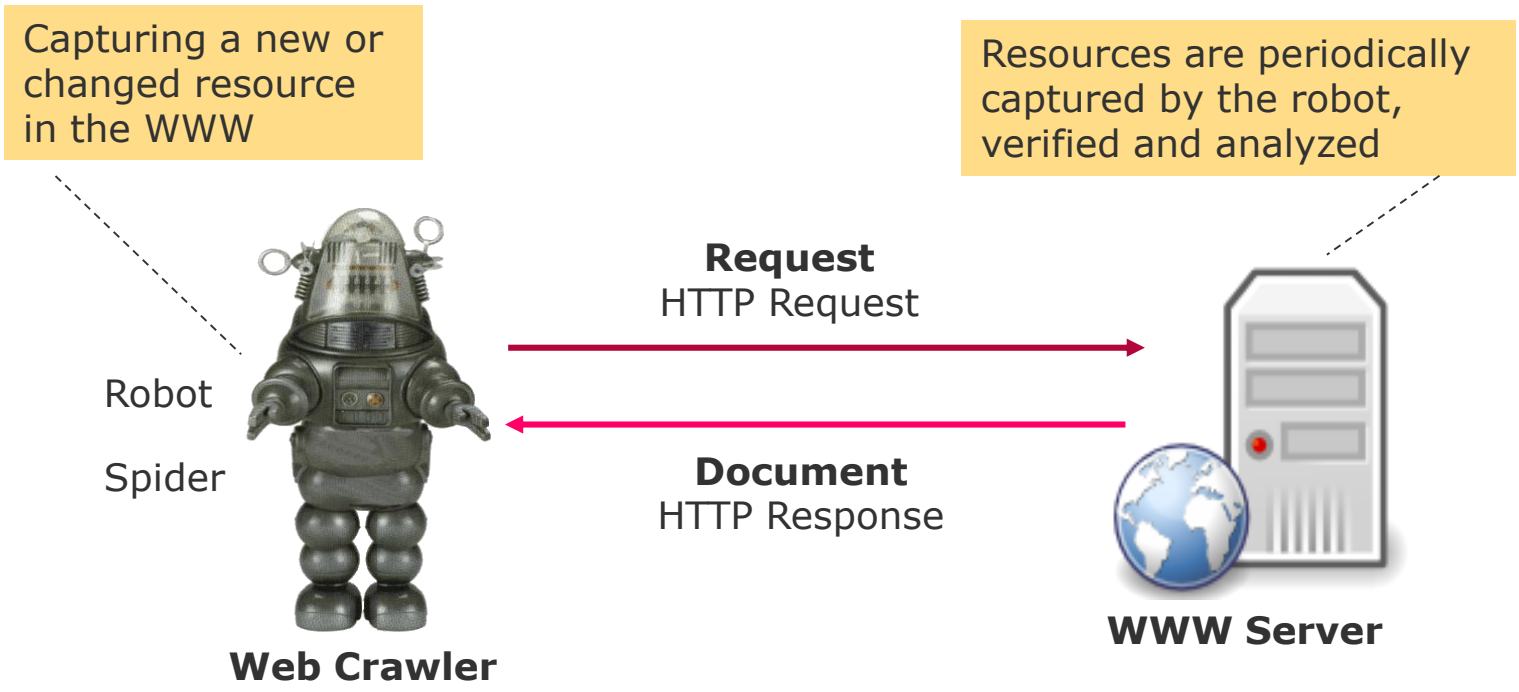


Index-Based Search Engines
Operating Principle of a Web Crawler (1/2)



Index-Based Search Engines
Operating Principle of a Web Crawler (2/2)

Web crawlers work distributed according to the client / server principle:



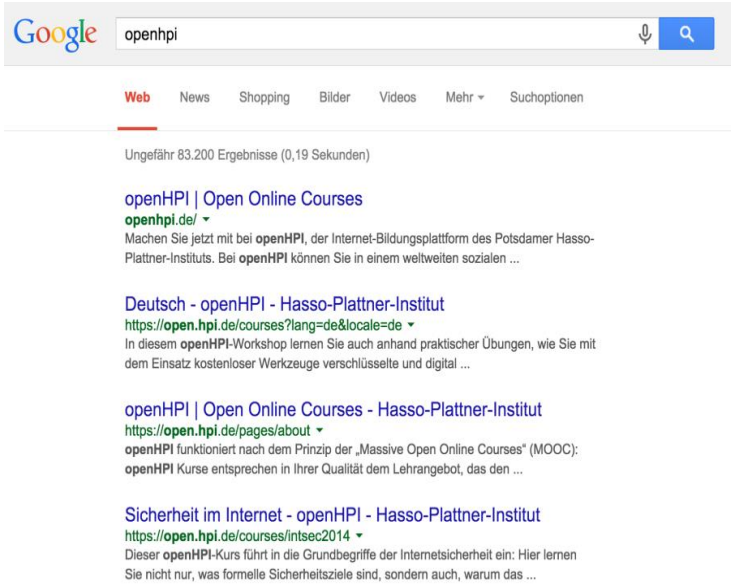
Index-Based Search Machines: Document Analysis and Evaluation

Analysis and evaluation of a document's content is done completely automatically by software tools ➔ **Information retrieval systems**

- Information retrieval systems identify **key content** in the examined documents
- Analyzed documents are stored in a database according to the identified key contents (e.g. keywords)
- Individual documents are weighted by their → **relevance**
- Methods for creating a searchable database are referred to as **indexing**

Index-Based Search Machines: Answering Queries

- Search is done by entering one or more search terms (**query**)
- Automatic similarity comparison with each document in the database leads to a selection of resulting documents corresponding to the entered query
- Selection of the result documents is performed by **query processor** (that is actually the "search engine")
- Found documents are displayed in an order according to their relevance



Index-Based Search Machines: Pro & Cons

Advantages:

- Automatic data collection allows the efficient creation of a huge corpus of a largely up-to-date and complete dataset

Disadvantages:

- Accuracy depends on the algorithms used for
  - analyzing and evaluating documents and
  - determining relevance with respect to a query
- Automatic relevance assessment can lead to qualitative inferior results

openHPI

Data Processing and Ranking of Search Results

Prof. Dr. Christoph Meinel
Hasso Plattner Institute
University of Potsdam, Germany

Data Processing and Analysis (1/3)

After having collected documents by web crawlers, they must be transformed into an **efficiently searchable dataset**

➔ **Content indexing of text documents**:

- First, various document types (HTML, Postscript, PDF, Doc, …) need to be converted into uniform document type
- Then content need to be extracted by **semantic analysis**
  - Basic idea: Find all relevant character strings in the text file:
    - key words
    - headers
    - bullet points, …
- Assign **"descriptors"** (key words) describing the document
- Collect ranking information due to given evaluation criteria

Data Processing and Analysis (2/3)

Web Crawler

- Data Normalization
- Word Identification
- Language Identification
- Word Stemming
- Descriptors
- Black List

Information retrieval

Data Processing and Analysis (3/3)

Obtaining Key Words: Descriptors

Goal:

- Document is to be represented as completely as possible content-wise by the use of only a few key words
- Use **keyword relevance filtering** for partial context analysis, e.g.
  - HTML <Hx> tags, text highlighting, etc…
  - omission of filler words, connecting words, pronouns, etc…
- Frequency analysis of keywords → **relevance**

George Kingsley Zipf (1902-1950)

Zipf's Law: It is always easier for the author of a text to repeat certain words in describing a topic than to constantly search for new terms.

Document Vectors

Vector Space Model

- Use of proven mathematical methods from information retrieval

Idea:

- Document is considered as a **vector** in a **n-dimensional vector space**, n ← number of descriptors
- Each **basic vector** represents another descriptor
- **Document vector** is obtained as a linear combination of all its basic vectors, whereby each basic vector is multiplied by the number of occurrences of a word

→ Document analysis can then be performed by means of mathematical operations and computations from linear algebra

Example for Document Vectors

The pineapple is a citrus fruit and not a vegetable.

The potato is neither fruit nor vegetable. Onions are vegetables.

Inverted Index:

Word	Basic Vectors	#d1	#d2
Pineapple	e1=(1,0,0,0,0,0)	1	
Vegetable	e2=(0,1,0,0,0,0)	1	2
Potato	e3=(0,0,1,0,0,0)		1
Fruit	e4=(0,0,0,1,0,0)	1	1
Citrus	e5=(0,0,0,0,1,0)	1	
Onion	e6=(0,0,0,0,0,1)		1

…

Document Vectors:

$d1 = e1 + e2 + e4 + e5 = (1,1,0,1,1,0)$

$d2 = 2 * e2 + e3 + e4 + e6 = (0,2,1,1,0,1)$

Finding Documents

Similarity Determination

- Query Q is also represented as a document vector
  - Search words are the descriptors here
- Calculation of the similarity of the document vector of Q with all document vectors D
- Similarity between Q and D is calculated as the cosine of the angle θ between the two document vectors Q and D

$$\boldsymbol{sim}(Q, D) = \cos(\theta)$$

Ranking the Search Results

In order to achieve high quality search results, the documents obtained from the document index must be weighted according to their **relevance**

➔ **What is important?**

Term Frequency Algorithm (**TFA**)

- **Zipf's Law**:
  - The more often a key word appears in a text, the more important it must be
- Simplest weight: Absolute word frequency
- Alternative possibility: Relative word frequency

There are many more complex relevance criteria used by search engines, i.e. the famous ➔ Google PageRank

Example: Google's Relevance Model

Google PageRank

- In order to achieve high quality search results, the documents obtained from the inverted index must be weighted according to their relevance

Relevance weighting: Google distinguishes between "*important*" and "*unimportant*" documents:

(1) A document is of higher "*importance*" the more other documents refer to it via links

(2) A document that refers to an "*important*" document is also considered "*important*"

(3) The more links a document contains referring to other documents, the less "*important*" each single link is

Google's PageRank algorithm is one of the central corporate assets and, therefore, **not disclosed**

openHPI

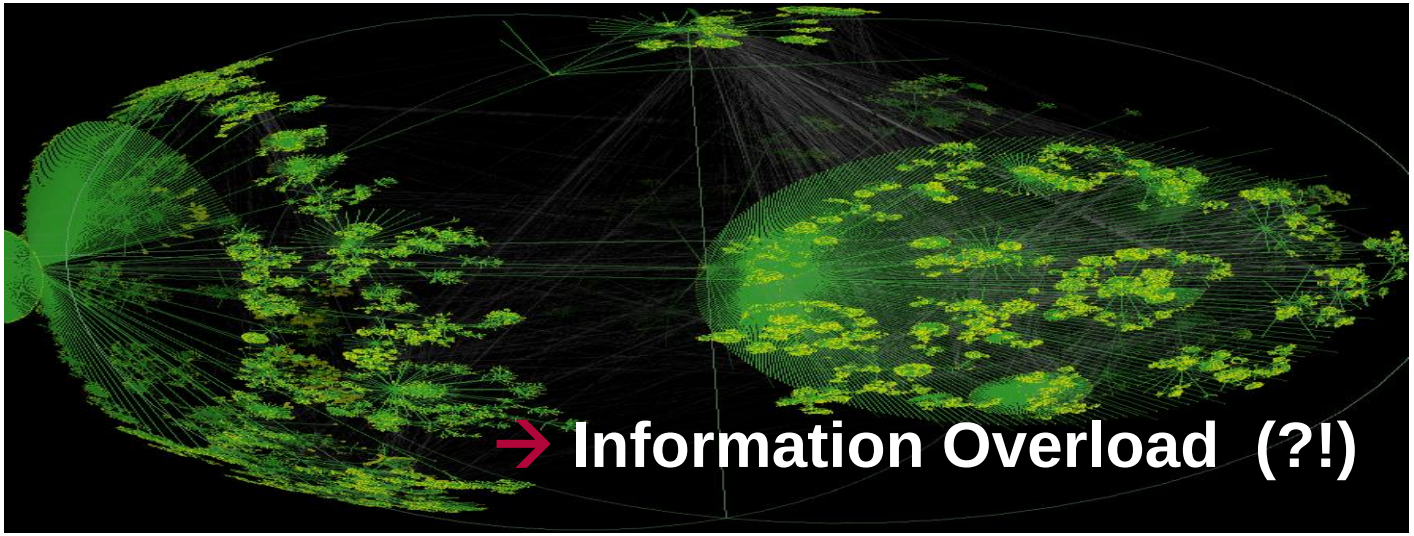
Semantic Web – An Introduction

Prof. Dr. Christoph Meinel
Hasso Plattner Institute
University of Potsdam, Germany

Limitations of the World Wide Web (1/7)

Remember: The WWW allows access to an enormously large source of information

- There are billions of documents in the WWW – however, only a small fraction has been indexed by search engines
- Additional billions of documents can be found in the deep web
- The document base of the WWW doubles every 6 months



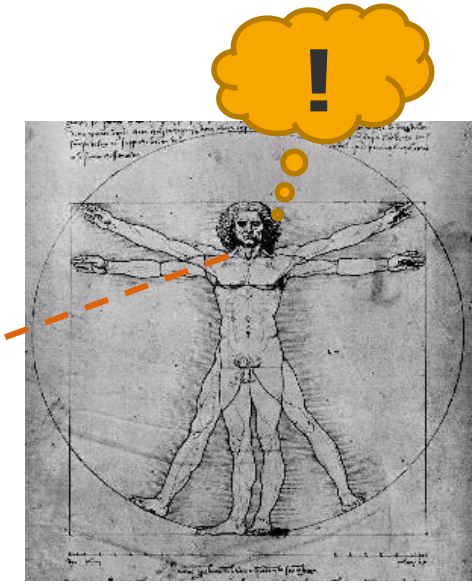
Limitations of the World Wide Web (2/7)

Relevant information?

- What is **important**, what isn't?
- Where is **information**, where is advertisement?
- What is the meaning of the information?
- How **credible** is the information?
- What belongs **together**?
- Which parts are actually **redundant**?

Limitations of the World Wide Web (3/7)

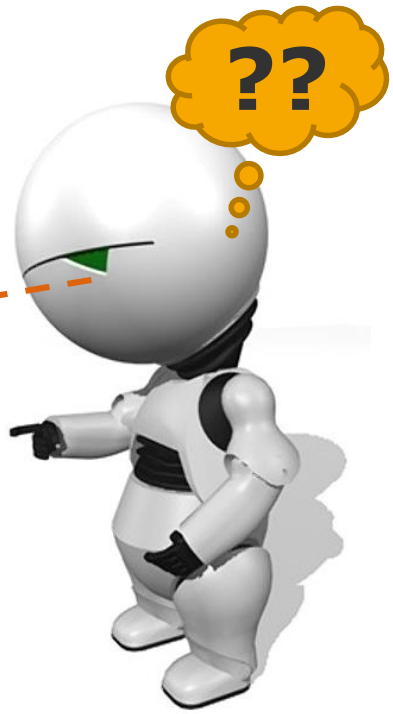
Is that huge amount of information actually useful?



Human users have **contextual knowledge / world knowledge**, can thus interpret (most) information correctly

Limitations of the World Wide Web (4/7)

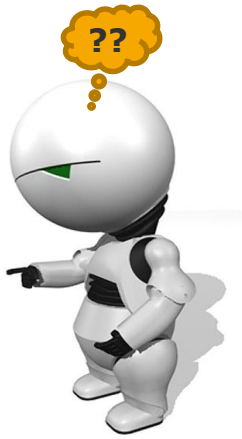
A **robot** (program) cannot distinguish important and insignificant information, since it **can't understand meaning**



Limitations of the World Wide Web (5/7)

Robot needs information about meaning of a document and it's different parts

➔ **Metadata**



Advertisement

However: HTML only offers very limited options for featuring metadata

Limitations of the World Wide Web (6/7)

Robot needs information about

```
<meta charset="UTF-8" />
<meta http-equiv="cache-control" content="no-cache" />
<meta http-equiv="content-language" content="en-US" />
<meta http-equiv="content-type" content="text/html; />
<meta name="author" content="Jane Doe" />
<meta name="description" content="...summary of web page..."
/>
<meta name="robots" content="all" />
<meta name="keywords" content="meta tags, meaning" />
```

All metadata must be created by the author of a website

H y limited options for featuring metadata

Limitations of the World Wide Web (7/7)

- WWW originally was **made for humans**
- Enabling technology for the WWW is the markup language HTML
- HTML and CSS describe
  - how information is **structured** and **presented**
  - how information can be **interlinked**
  - but **not the meaning** of these information

Semantics ???

Semantics

Semantics is a subfield of **linguistics**, dealing with

- **sense** and
- **meaning**

of **language** and **linguistic symbols**

Semantics tries to answer the question how

- sense and meaning of complex terms can be deducted from those of simple terms

Semantics builds on **syntax**

- **Communication** requires a common understanding of **syntax** and **semantics** of the interchanged language symbols and character strings

➔ Mutual **understanding** is only possible, if sender and receiver have agreed on **identical semantics**

Problem Area 1: Search for Information

Keyword-based search

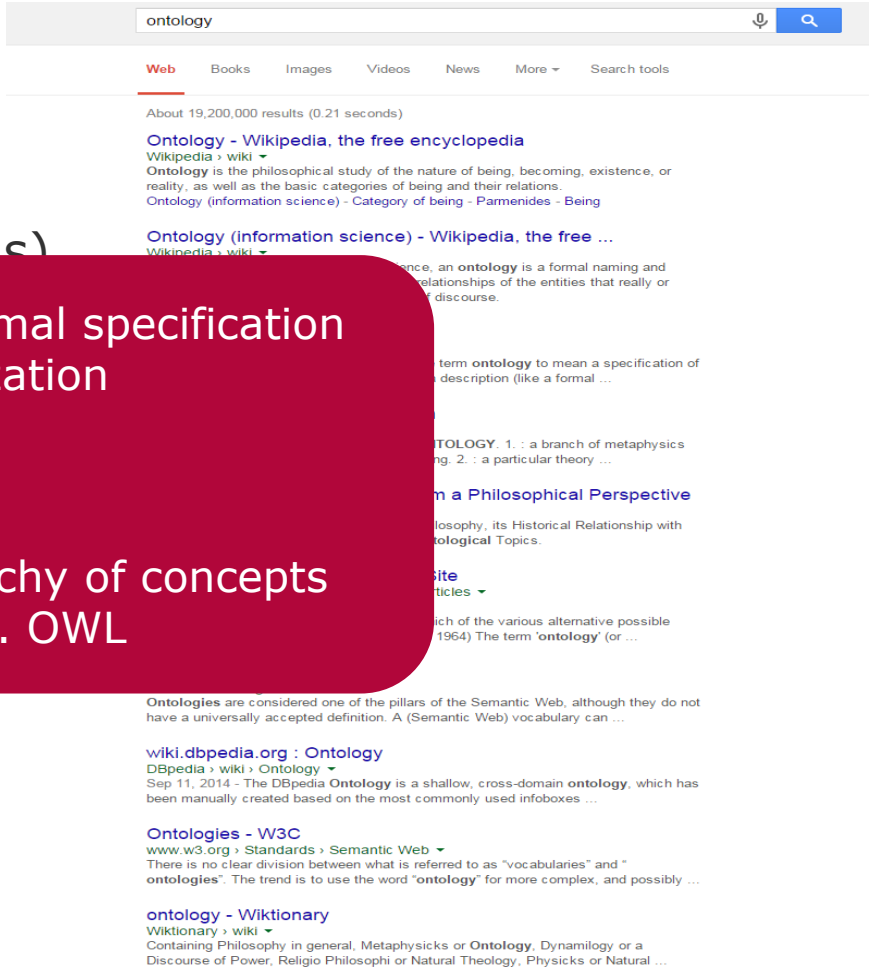
- retrieves many **irrelevant** results
  - different meaning (homonyms)
  - diffe
- can't fir
  - sync
  - lack

Ontology: An explicit formal specification of a common conceptualization

An ontology consists of:
- a **taxonomy**, i.e. hierarchy of concepts
- a common language, i.e. OWL

Requirement:

➔ Introduction of a formal terminology / **ontologies**



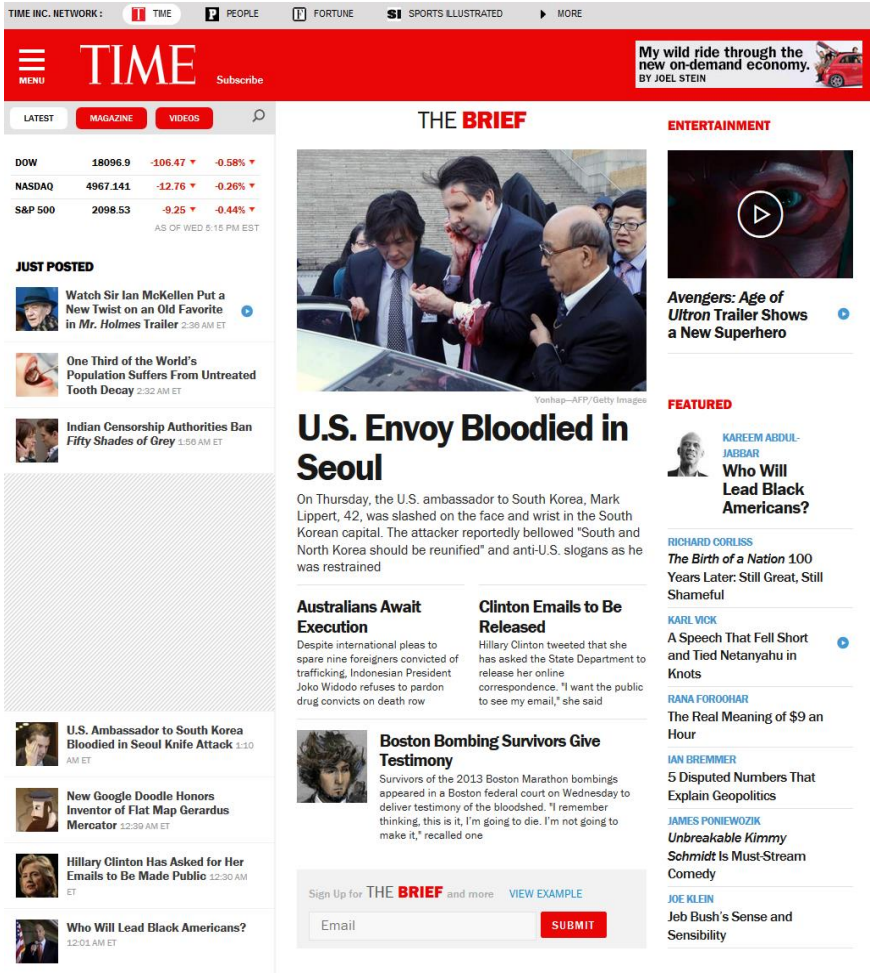
Problem Area 2: Information Extraction

- Only human „agents“ can extract information correctly
- Software-agents (robots) lack
  - **contextual knowledge**
  - **world knowledge**

  which are required to extract information from the text- or image-based representation

Requirement:

➔ **Aggregation and integration of information** from different sources

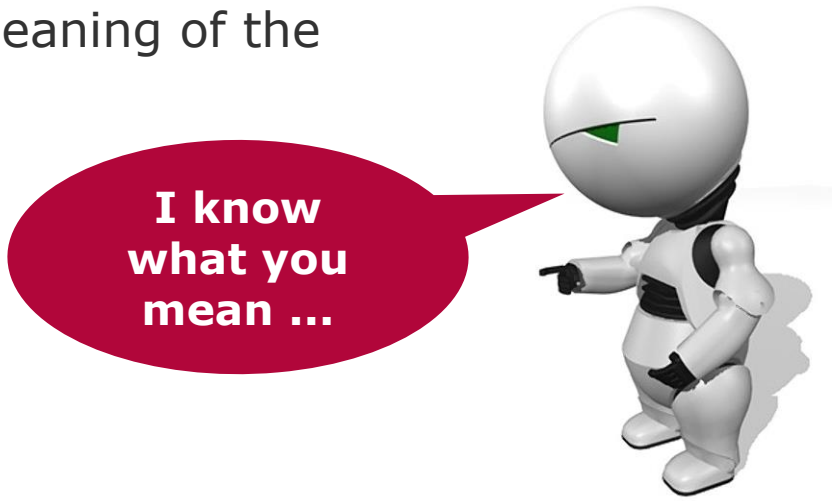


Vision for the Semantic Web

- **Tim Berners-Lee**, Sept 1998:

  „The Web was designed as an information space, with the goal that it should be useful not only for human-human communication, but also that machines would be able to participate and help… “

- **HTML** as language for information presentation in the WWW lacks a possibility to express the meaning of the displayed information



Potential of the Semantic Web (1/3)

Example 1: Search Engines – Today

Search engine query: *„GOLF from 2015"*

User must refine queries, if search results are ambiguous

- Inquiry on *„GOLF"*:
  - *car* or *sport* or *others*? ➔ Refinement: *car*
- Inquiry on *„2015"*:
  - *Year of construction* or *technical parameter*?
    ➔ Refinement: *Year of construction*
- Inquiry on *„Golf from 2015"*:
  - *Sales offer*, *manual* or *other*?
    ➔ Refinement: *Manual*
- …

Search machine has to ask the user to get the meaning of its query

Potential of the Semantic Web (2/3)

Example 1: Search Engines – Tomorrow

Search engine query: *„GOLF from 2015"*

Search Engine „knows" the user

- He/she checked results of professional golf tournaments regularly in the last months
- He/she has bought a set of golf clubs last year (via online shopping)
- He/she has subscribed to the news group „en.rec.golf.balls"
- He/she chose the desktop background image „PGA Tour 2014"
- He/she owns the internet domain www.thegolffanatic.com
- ...

Due to this knowledge, search machine can conclude the meaning of the query and answer it correctly …

Potential of the Semantic Web (3/3)

Example 2: Planning a Holiday Trip

Input for Agent: Vacation, seaside, $\geq$ 10 days, family, $\leq$ 2000 €

- Agent checks calendars of all family member and identifies free slots
- Agent checks holiday offerings for the identified dates according to the requirements
  - geographic parameters
  - family-friendly?
  - golf court nearby?
  - ...
- Agent checks free capacity and compiles a selection of matching offers
- Agent organizes transportation according to the chosen location …

Objectives of the Semantic Web

- Organization of knowledge in separate domains according to its meaning
- Automated tools for maintenance or removal of inconsistencies, and extraction of new knowledge
- Replacing keyword-based search by content-based search with specific question- and answer-cycles
- User-friendly search
- Extraction and presentation of knowledge
- Answering questions with dispersed documents
- Exact determination, who may access which part of a specific information

Technical Realization of the Semantic Web (1/2)

The realization of Semantic Web requires a number of **technologies** for different purposes:

- Any information source in Semantic Web must have a unique identifier

  ➔ **URI**

- Information in Semantic Web must be expressed formally in order to be processible by automatic agents. This requires a unified syntax

  ➔ **XML, XMLSchema**

- Simple semantic relations between information entities must be expressible

  ➔ **RDF (Resource Description Framework), RDFSchema**

- …

Technical Realization of the Semantic Web (1/2)

The realization of Semantic Web requires a number of **technologies** for different purposes:

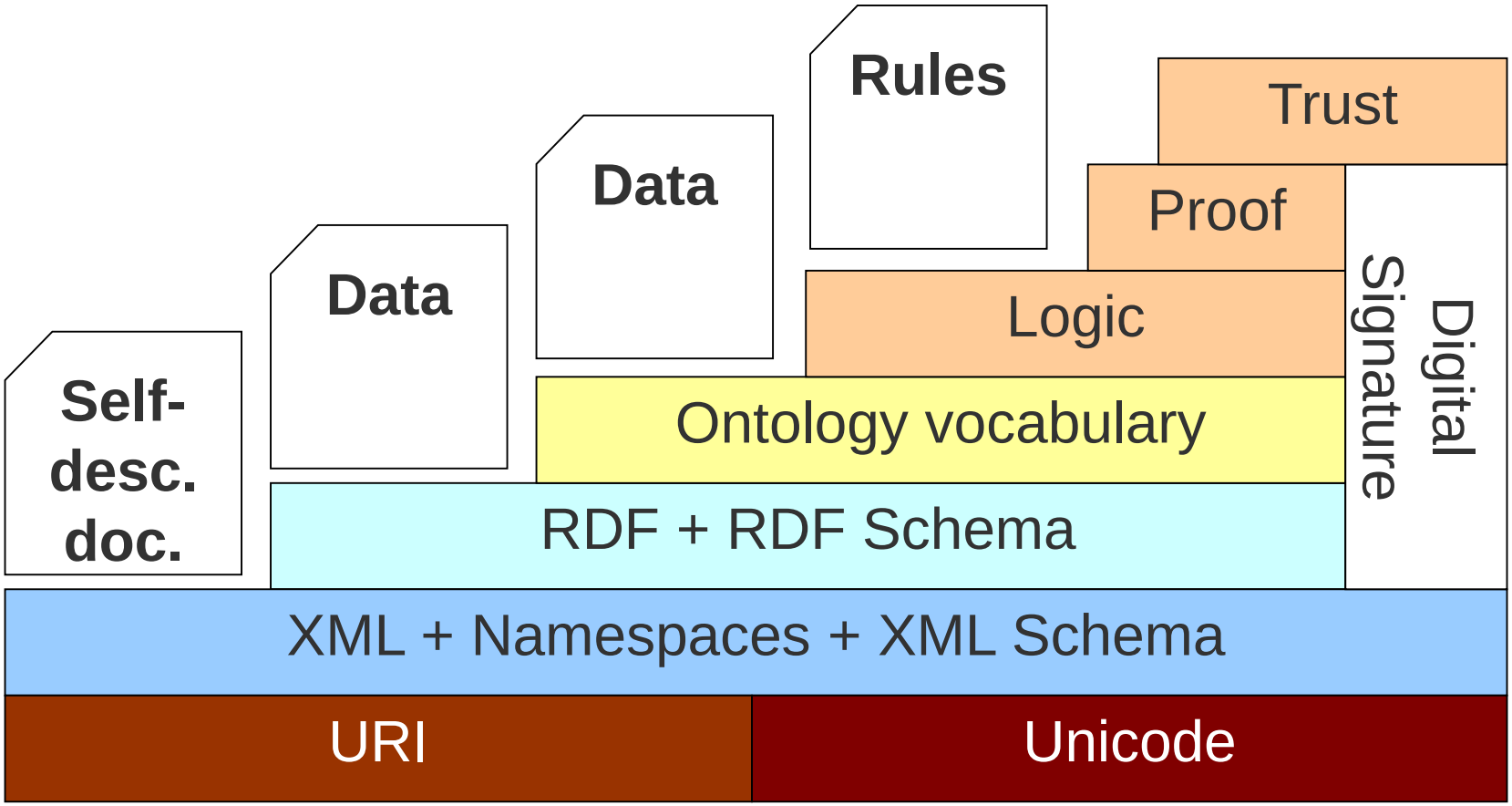
- …
- Different semantics must be explainable in order to combine seemingly inconsistent information

  ➔ **Ontologies**
- New information shall be concluded from existing information

  ➔ **Inference mechanisms**
- Technology for data protection (in terms of confidentiality and integrity) must be applied

  ➔ **XMLEncryption** and **XMLSignature**

Semantic Web Technology Stack



Further information:

➔ openHPI courses on **Semantic Web Technologies** by Dr. Harald Sack

openHPI
SOAP-based Web Services

Prof. Dr. Christoph Meinel
Hasso Plattner Institute
University of Potsdam, Germany

Web Services – Introduction (1/2)

Web services are a general approach for the development of distributed applications based on the interconnection of distributed object systems across the Web

- Web services subsume various technologies for platform independent access to services from different providers in distributed systems and are independent of operating systems, programming languages, and binary transmission protocols
- Web services make the Internet a universal platform for **communication between distributed applications**

Definition: [A **web service** is] a software system designed to support interoperable machine-to-machine interaction over a network. *(W3C)*

Reminder: Lightweight web services have been explained earlier; this lesson deals with a more resourceful kind of web services

Web Services – Introduction (2/2)

- Web services are based on **remote procedures** or **remote method calls**
- In contrast to middleware solutions used until now, e.g. CORBRA, no new binary protocol (such as IIOP) is deployed
- Instead, protocols are used that rely directly on the application layer protocol HTTP – such as the XML-based → **SOAP**

Middleware Solutions

Web Services are conceptually based on **middleware solutions** for the execution of code on remote systems

- **RPC** – **Remote Procedure Call** (RFC 1057)
  - allows a program at one machine to cause a subroutine to be executed at another machine
- **RMI** – **Remote Method Invocation**
  - is a Java framework that offers an object-oriented equivalent to RPC
- **CORBA** – **Common Object Request Broker Architecture**
  - is a standard that allows the interaction of different software on different machines across a network, independent of the programming language they were created

Web Services Inventory (1st Generation)

Web Service protocols of the first generation still form the basis for many of the Web service standards used today:

SOAP

- XML-based network protocol for exchange of data between systems and invoke RPCs in an uniform message format (W3C standard)
- Formerly an acronym for Simple Object Access Protocol; today this meaning no longer applies to SOAP

WSDL – Web Service Description Language

- XML-based descriptive language for interfaces of Web services

[**UDDI** – Universal Discovery, Description and Integration] – has meanwhile not proven successful …

- Directory service in which service providers register their web services and users can search for them (has not proven successful)

SOAP (1/2)

- **SOAP envelopes** are an XML format and consist of – like HTTP – a **header** and a **body**

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="…">
  <soap:Header> … </soap:Header>
  <soap:Body> … </soap:Body>
</soap:Envelope>
```

- The SOAP envelope is sent as payload in the body of a HTTP request or response
- All requests are usually carried out using HTTP POST
- Because of its format as a separate envelope, SOAP does not necessarily rely on HTTP as a transport protocol …

SOAP (2/2)

Calls a function `getLecturer` with parameters `courseCode: 'webtech2015'` and `courseName: 'Web Technologies'`

- **SOAP Request:**

```
...
<m:GetLecturer xmlns:m="http://namespaces.open.hpi.de">
      <m:courseCode>webtech2015</m:courseCode>
      <m:courseName>Web Technologies</m:courseName>
</m:GetLecturer>
...
```

Returns the requested lecturer

- **SOAP Response:**

```
...
<m:GetLecturerResponse xmlns:m="http://...">
      <m:lecturer>Christoph Meinel</m:lecturer>
</m:GetLecturerResponse>
...
```

WSDL (1/4)

- Web service can be seen as an application with multiple methods, each providing a specific functionality
- Access to the functionality is carried out by invoking the appropriate method of the application
- For the remote method invocation, the application must implement an interface for the web service
- Specification of such interfaces is platform-independent done with **WSDL** – **Web Service Description Language**
- WSDL allows to describe
  - which functions can be provided over which ports
  - which messages and protocols may be used in order to use the individual functions

WSDL (2/4)

WSDL description specifies

- Which web service it is and what functions are rendered by the web service – "**What**"
  - `portType` (WSDL1.1)
  - `interface` (WSDL1.2)
- How can the offered service be accessed? (generally via SOAP/HTTP) – "**How**"
  - `binding`
- Over what URL can a web service function be invoked? – "**Where**"
  - `service`

WSDL (3/4)

WSDL Example (1/2)

```
<wsdl:types>
  <xsd:schema targetNamespace="http://...">
    <xsd:element name="GetLecturer">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="courseCode" type="string" />
          <xsd:element name="courseName" type="string" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    ...
```

Definition of data types used in the described Web Service

Datatype `GetLecturer` contains sequence of `courseCode` and `courseName`

WSDL Example (2/2)

```
...
    <xsd:element name="GetLecturerResponse">...</xsd:element>
    <xsd:element name="GetLecturerFault">...</xsd:element>
  </xsd:schema>
</wsdl:types>
<wsdl:message name="GetLecturerRequest">
  <wsdl:part name="body" element="xsd:GetLecturer" />
</wsdl:message>
<wsdl:message name="GetLecturerResponse">
  ...
</wsdl:message>
```

Message definition for `GetLecturer` request references data type for parameters

WS* Protocols (1/3)

- With **1st generation** Web Service protocols, it is possible to
  - describe web services (WSDL)
  - find web services (UDDI)
  - invoke web services (SOAP and WSDL)
- … but there were no standards for
  - protection of web services,
  - composition of multiple web services (orchestrator),
  - reliable execution of web services

  and many other requirements

➔ Therefore, the **WS* protocol suite** – web service standards of the **2nd generation** – was developed

WS* Protocols (2/3)

WS*Standards (excerpt)

- **WS-Security, WS-Policy** – allows, e.g., end-to-end encryption or specifying access restrictions for web services
- **WS-Coordination** – framework for implementation of distributed applications, allows the involvement of several web services in distributed transactions
- **WS-Federation** – Summary of web services
- **WS-Notification** – Notification system for communication between web services (publish / subscribe pattern)
- Many standards are little known, e.g. WS-Atomic Transaction, WS-Resource Lifetime, WS-Base Faults, …
  ➔ **WS*** stands for **very many** web service standards …

Web Services Standards Overview