

Einführung

Albert Weichselbraun

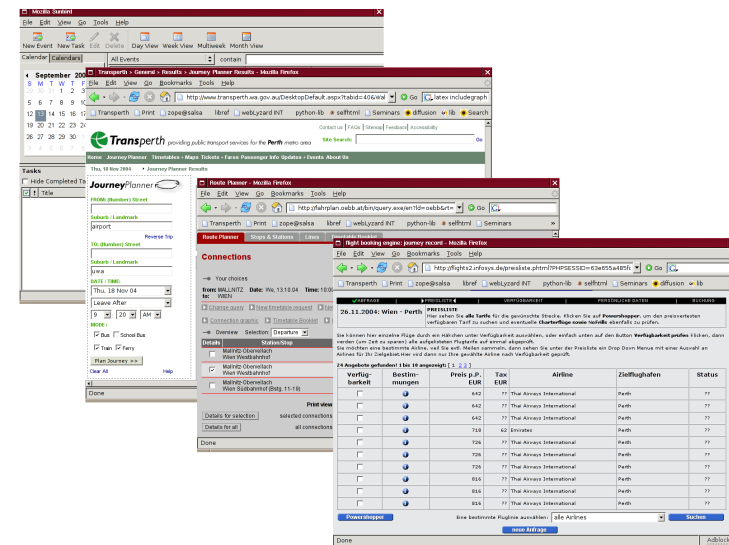
Agenda

- Usecase / Teaser
- Grundlegende Konzepte
- Anwendungen und Probleme
- Semantic Web Technologien
 - Unicode
 - Uniform Resource Locator (URL)
 - eXtensible Markup Language (XML)

Usecase: Urlaub buchen

Einige der Dinge, die zu tun sind...

- passenden Termin finden
- Hotel mit entsprechendem Unterhaltungs-/Sportprogramm
- Buchen eines (günstigen) Fluges → Kalender
- Fahrt zum/vom Flughafen organisieren



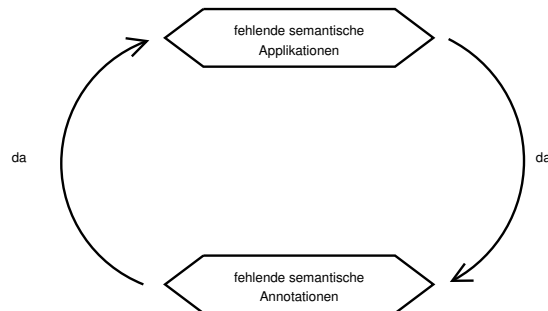
Usecase - Bestandsaufnahme

- die notwendigen Informationen sind bereits mehr oder weniger strukturiert in elektronischer Form abrufbar
- Problem: intelligente Kombination der Informationsquellen
- Hintergrund: Ressourcen sind für Maschinen nicht "verständlich"

Standard Technologien

- Information Retrieval → unstrukturierte Informationsquellen → Suchabfragen durchführen
 - Suche nach Schlüsselwörtern (Kontext fehlt!)
 - Ambivalenzen, Pseudonyme, ...
→ Disambiguierung notwendig
 - kognitive Fähigkeiten von Computern beschränkt
- Semantische Technologien
 - versehen Ressourcen mit semantischen Informationen
 - Ansätze: Information Extraction vs. Annotation durch den Autor der Information

Semantic Web - Problemstellung



Semantic Web - Ansatz

- strukturiert *bestehende* Webressourcen
- für Maschinen *und* Menschen verstehbar (Erfassen der Bedeutung!)
- Erweiterung des derzeitigen Webs (2. Schicht)
- dezentral
 - Recommendersysteme
 - Web of Trust
 - Communities können eigene (auch widersprüchliche!) Ontologien definieren

Anwendungen

- Abfragen von Daten-Repositories:
 - Online-Portale (geizhals.at, ...)
 - Preisvergleich, Reiseinformationen, ...
- Suchmaschinen
 - Hintergrundwissen in Abfragen:

“Informationen zu Tieren, welche Sonar verwenden, aber nicht Fledermäuse oder Säugetiere sind”
- Datenaustausch zwischen Applikationen (OpenDocument Format)

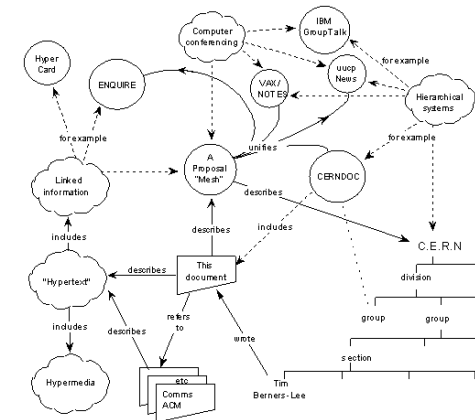
Anwendungen

- Lokalisierung und Nutzung von Services
 - Ubiquitous Computing (Mobile ← Fernsteuerung, ...)
 - *Web Services*
- Delegation von komplexen Tasks an Agenten
 - Buchung von Reisen
 - Recommendersysteme (Kauf von Komplexen Gütern)

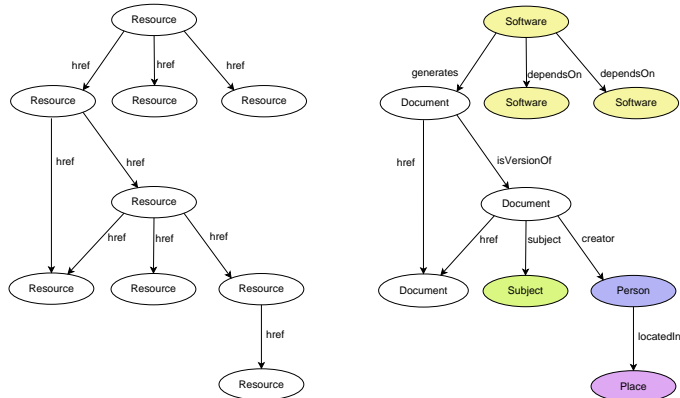


Cartoon Ubiquitous Computing (Berners-Lee et al., 2001)

Semantic Web - The Vision (Berners-Lee, 1989)



Semantic Web vs. Current Web



World Wide Web vs. Semantic Web (Koivunen und Miller, 2001).

Syntax vs. Semantic

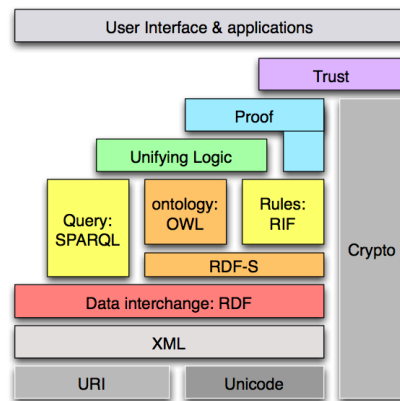
Syntax vs Semantic

- Syntax: Struktur von Daten (XML, RDF)
- Semantic: Bedeutung der Daten (OWL)

Voraussetzungen für Interoperabilität:

- *Gemeinsame* Syntax (Grammatik)
- *Gemeinsames* Verständnis (ontology)

Semantic Web - Enabling Technologies 2006 (Berners-Lee)



Semantic Web Stack (<http://www.w3.org/2006/Talks/0811-sb-W3Cemergingtech/>)

Technology - Unicode

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1	XON	!	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

Technology - Unicode

- Idee: Bitfolgen encodieren Zeichen
- Vorgänger:
 - ASCII (American Standard Code for Information Interchange); 7-bit Werte für Zeichen der englischen Sprache;
 - ISO-8859-1: 8-bit Werte; enthalten Zeichen für westeuropäische Schriften
 - weitere Encodings, um Zeichen in anderen Sprachen darzustellen (ISO-8859-x, KOI8, JIS, ...)
 - Umsetzung: teilweise nicht standardkonforme und abhängig vom Betriebssystem

Technology - Unicode

- Unicode 5.0: definiert 98.884 grafische Zeichen und mathematische Symbole
- "provides a unique number for every character"
- Betriebssystem-unabhängig; erweiterbar
- Technische Umsetzung:
 - Platz für 1.114.112 code-points; organisiert in 17 Ebenen zu je 2^{16} (65.536) Zeichen
 - die ersten 256 Positionen entsprechen ISO 8859-1
 - Serialisierungsformate u.a.: UTF-8 (variabel), UTF-16 (variable) und UTF-32 (fix)

Technology - Uniform Resource Locators

- Identifizieren Ressourcen
- Allgemein: `schema:location`
- Für viele Anwendungen folgender hierarchischer Aufbau:
`schema://[user[:pwd]@]host[:port]/[path][?q][#frag]`
- Beispiele:
 - `mailto:lisa.maurer@tuwien.ac.at`
 - `http://www.heise.de/verify.py?browser=firefox`
 - `ldap://at.net/?dn=cn=albert,ou=admin,o=atnet,c=at`

Extensible Markup Language (XML)

- Aktuelle Spezifikation: Extensible Markup Language (XML) 1.0 (Fourth Edition); 16. August 2006
XML 1.1 (16. August 2006)
- definiert die *Struktur* von Daten
- "well-formed": XML-Dokument erfüllt alle XML-Syntaxregeln
- "validating": Struktur entsprechend Document Type Definition (DTD) oder XML-Schema

Beispiel: Struktur von Daten

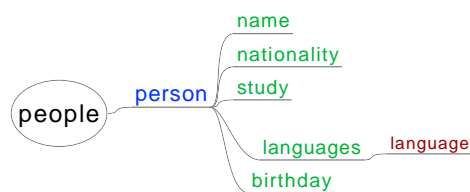
- Problem: flat data files schwierig erweiterbar
- XML: Auszeichnung der Daten, hierarchische Baumstruktur mit *einem* Wurzelement
- XML-Approach: Start-/End-Tag, Daten bilden ein XML Element
- XML-Tag Syntax Elemente:
 - Processing instructions: `<? data ?>`
 - Tags: `<name>`, `</name>`, `<name />`
 - Attribute: `<name attrname="attrwert" />`
 - Kommentare: `<!-- Kommentar -->`
 - Entitäten: `&name;` `&#nnnn;` `&#xhhhh;`

Beispiel XML-Dokument

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- Wurzelement; weitere Person moeglich? -->
3 <Person type="Student">
4   <name>Borbala Nagy</name>
5   <dob>08-12-80</dob>
6   <nationality>hu</nationality>
7   <languages>
8     <language>de</language>
9     <language>en</language>
10    <language>hu</language>
11  </languages>
12  <study xml:lang="en">mathematics</study>
13  <study xml:lang="de">Mathematik</study>
14 </Person>
```

Designpattern und Konventionen

- Klassennamen: beginnen mit Großbuchstaben
- Instanzen und Eigenschaften: Kleinbuchstaben
- Pattern: Identifizierung von Klassen
→ Name der Instanz als id-Attribut übergeben



Identifizierung von Klassen

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <People id="EMS-Sch&uuml;ler">
3   <Person id="Borbala Nagy">
4     <name>Borbala Nagy</name>
5     <dob format="dd-mm-yy">08-12-80</dob>
6     ...
7     <study>mathematics</study>
8   </Person>
9   <Person id="Martin Pospisil">
10    <name>Martin Pospisil</name>
11    <dob format="dd-mm-yy">16-11-82</dob>
12    ...
13  </Person>
14 </People>
```

Namensräume - Motivation

```
SLR Single Lens Reflex (camera)
SLR Satellite Laser Ranging
SLR Sending Loudness Rating (telecommunications)
SLR Service Level Report
SLR Side Looking Radar
SLR Single Linear Recording
SLR Slide Raft (aircraft door)
...
SLR Sri Lanka Rupee (national currency)
SLR Statutory Liquidity Ratio
SLR Stock Level Report
SLR Stock Level Requirement
SLR Straight Leg Raise
SLR System Level Requirement (s)
```

Namensräume

- Problemstellung: xml-Tags/Attribute mit gleichem Namen
- Lösung: durch die Verwendung von Namensräumen werden die XML-Elemente durch die *Kombination* von *Namespace URL* und *Lokalen Namen* eindeutig identifiziert.
- Umsetzung in XML: Definition von Namensräumen mittels des `xmlns`-Attributes
 1. Definition des Default-Namespace: `xmlns`-Attribut
 2. Definition eines Präfix: `xmlns:prefix`-Attribut
- Namespace Deklaration gilt für den entsprechenden Zweig
- Syntax für Qualifizierte Elemente: `<prefix:tag>`

Namensräume - Syntax

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <People id="EMS-Sch&uuml;ler"
3   xmlns="http://ai.wu.ac.at/~aweichse/2005#"
4   xmlns:dc="http://purl.org/dc/elements/1.1#" >
5   <dc:publisher>
6     http://ai.wu.ac.at/~aweichse
7   </dc:publisher>
8   <Person id="Borbala Nagy">
9     <dob format="dd-mm-yy">08-12-80</dob>
10    ...
11    <study>mathematics</study>
12  </Person>
13  ...
14 </People>
```