

Data Querying, Extraction and Integration II: Applications

Recuperación de Información

2007

Lecture 5.

Goal today: Provide examples for useful XML based applications

◆ Motivation:

- Integrating Legacy Databases, etc.
- Extractions from Websites

◆ Wrapper Generators for the Web

◆ APIs, Software

◆ Further interesting XML standards and applications

◆ What's missing?

Problems with the Data Integration: extracting information

- ◆ Different formats, different syntax, etc.
- ◆ few proprietary standards, e.g. EDIFACT, SWIFT, etc.

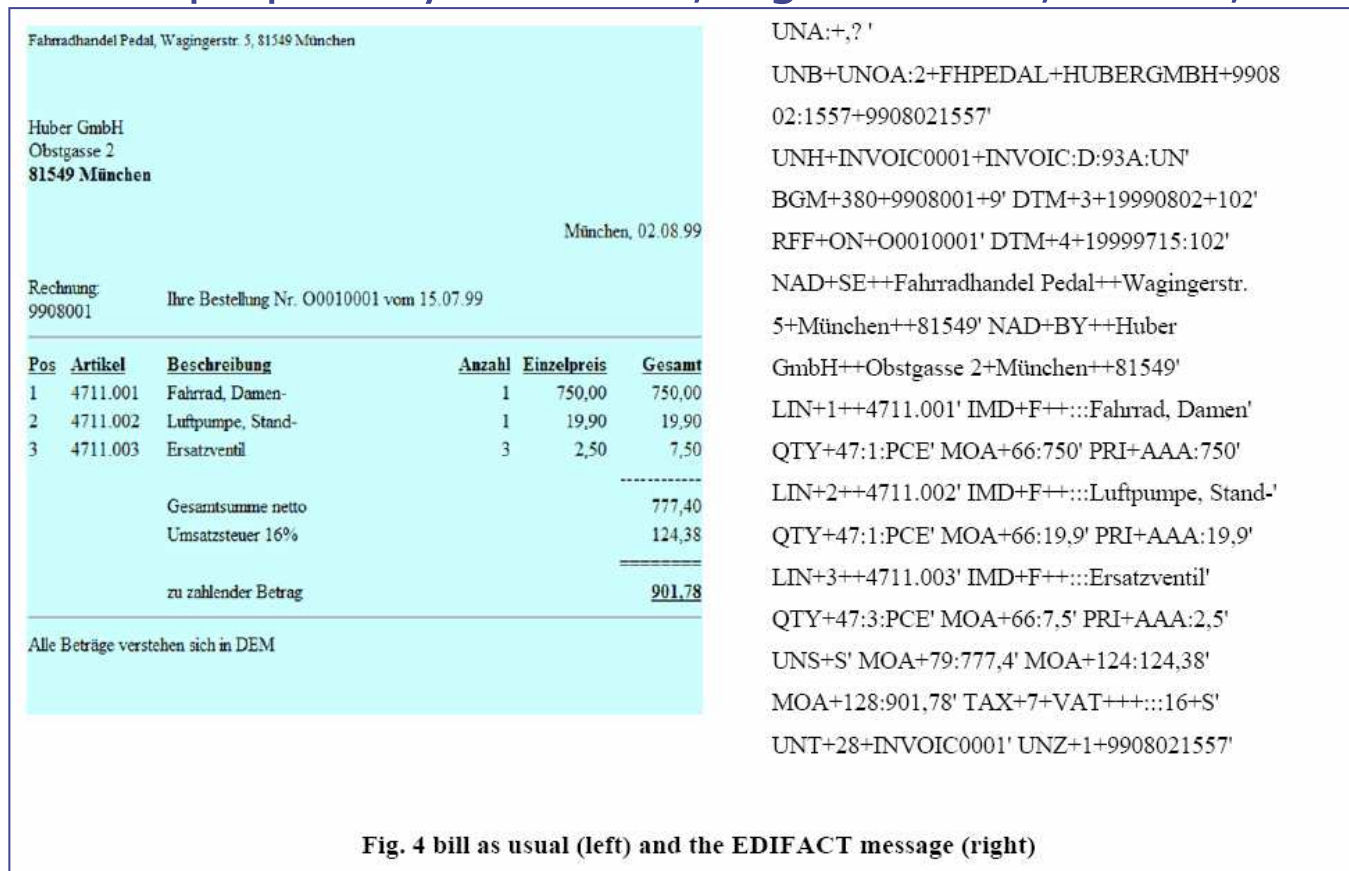


Fig. 4 bill as usual (left) and the EDIFACT message (right)

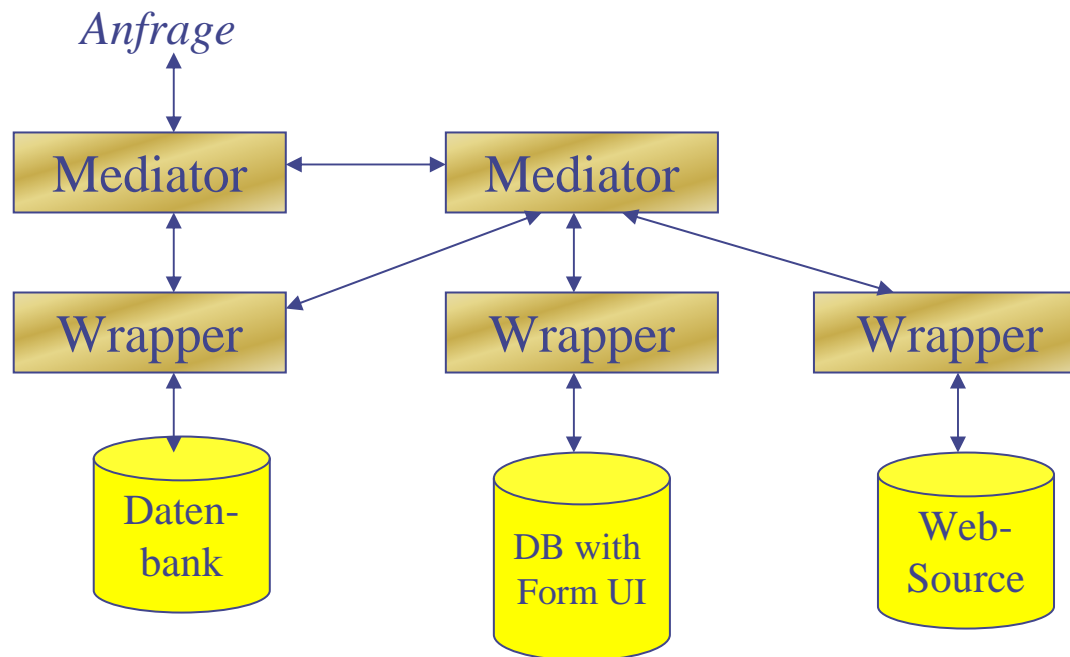
Problems with the Data Integration: combining information

Fahrradhandel Pedal, Wagingerstr. 5, 81549 München Huber GmbH Obstgasse 2 81549 München Rechnung: 9908001 Ihre Bestellung Nr. 00010001 vom 15.07.99			München, 02.08.99 Ihre Bestellung Nr. 00010001 vom 15.07.99	UNA:;?' UNB+UNOA:2+FHPEDAL+HUBERGMBH+9908 02:1557+9908021557' UNH+INVOIC0001+INVOIC:D:93A:UN' BGM+380+9908001+9' DTM+3+19990802+102' RFF+ON+00010001' DTM+4+19999715:102' NAD+SE++Fahrradhandel Pedal++Wagingerstr. 5+München++81549' NAD+BY++Huber																																
<table border="1"> <thead> <tr> <th>Pos</th> <th>Artikel</th> <th>Beschreibung</th> <th>Anzahl</th> <th>Einzelpr</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>4711.001</td> <td>Fahrrad, Damen-</td> <td>1</td> <td>750</td> </tr> <tr> <td>2</td> <td>4711.002</td> <td>Luftpumpe, Stand-</td> <td>1</td> <td>19</td> </tr> <tr> <td>3</td> <td>4711.003</td> <td>Ersatzventil</td> <td>3</td> <td>2</td> </tr> <tr> <td colspan="2"></td> <td>Gesamtsumme netto</td> <td colspan="2"></td> </tr> <tr> <td colspan="2"></td> <td>Umsatzsteuer 16%</td> <td colspan="2"></td> </tr> <tr> <td colspan="2"></td> <td>zu zahlender Betrag</td> <td colspan="2"></td> </tr> </tbody> </table>	Pos	Artikel	Beschreibung	Anzahl	Einzelpr	1	4711.001	Fahrrad, Damen-	1	750	2	4711.002	Luftpumpe, Stand-	1	19	3	4711.003	Ersatzventil	3	2			Gesamtsumme netto					Umsatzsteuer 16%					zu zahlender Betrag			{1:F01MIDLGB22AXXX0548034693}{2:I103BKTRUS33XBRDN3}{3:{108:MT103}}{4: :20:8861198-0706 :23B:CRED :32A:000612USD5443,99 :33B:USD5443,99 :50K:GIAN ANGELO IMPORTS NAPLES :52A:BCITITMM500 :53A:BCITUS33 :54A:IRVTUS3N :57A:BNPAFRPPGRE :59:/20041010050500001M02606 KILLY S.A. GRENOBLE :70:/RFB/INVOICE 559661 :71A:SHA -}
Pos	Artikel	Beschreibung	Anzahl	Einzelpr																																
1	4711.001	Fahrrad, Damen-	1	750																																
2	4711.002	Luftpumpe, Stand-	1	19																																
3	4711.003	Ersatzventil	3	2																																
		Gesamtsumme netto																																		
		Umsatzsteuer 16%																																		
		zu zahlender Betrag																																		

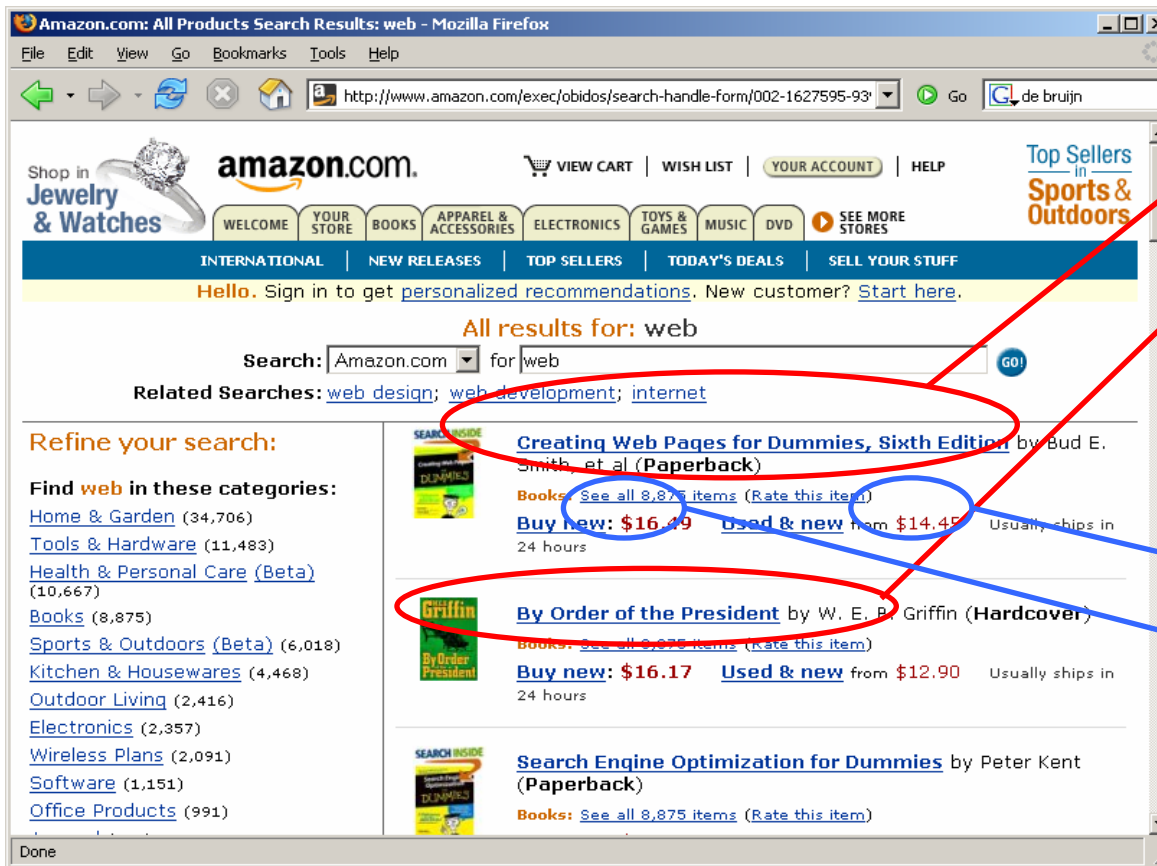
Fig. 4 bill as usual (

Possible Solutions:

- ◆ Write mediators from scratch, or:
- ◆ Use Wrappers and Mediators!
- ◆ Common format (XML)
- ◆ Easy transformation (XSLT)



Analogous problems with the Web: extracting information



Which book is about the Web?

What is the price of the book?

Problems with the Web: combining information

I want the cheapest copy of “A Semantic Web Primer”.

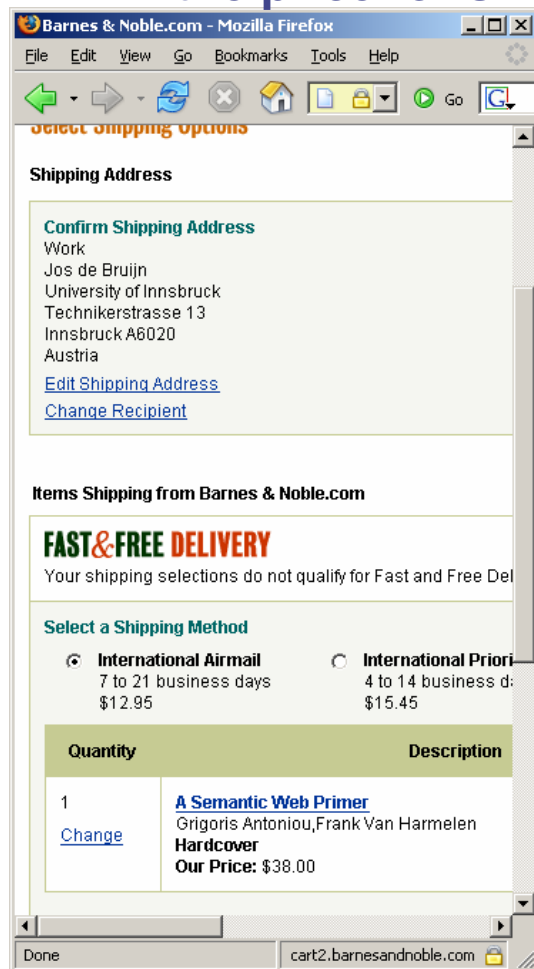
The image displays two screenshots of web browser windows, illustrating the process of finding the cheapest copy of a book by comparing prices across different retailers.

The top screenshot shows the Amazon.com search results for "A Semantic Web Primer". The search results include the book title, authors (Grigoris Antoniou, Frank van Harmelen), and the price: **Buy new: \$34.82** and **Used & new from \$32.75**. The page also features a "Refine your search" section and a "So You'd Like to..." recommendation.

The bottom screenshot shows the Barnes & Noble.com search results for "A Semantic Web Primer". The search results include the book title, authors (Grigoris Antoniou, Frank Van Harmelen), and the price: **List Price: \$40.00**, **B&N Price: \$38.00 (Save 5%)**, and **Member Price: \$36.10**. The page also features a "Pre-order the sixth Harry Potter book now!" section and a "SIGN UP" button.

The Problems with the Web: combining information

I want the cheapest copy of “A Semantic Web Primer”; taking into account the price for shipping the book.



The screenshot shows the Barnes & Noble shipping options page. It includes a shipping address section, a 'FAST & FREE DELIVERY' notice, and a table of shipping methods. The selected item is 'A Semantic Web Primer' priced at \$38.00.

Quantity	Description
1	Change A Semantic Web Primer Grigoris Antoniou, Frank Van Harmelen Hardcover Our Price: \$38.00



The screenshot shows the Amazon shipping rate page for 'Standard International Shipping'. It includes a table of shipping rates for various items and a section for 'Expedited International Shipping'.

	Per Shipment	Per Item
CDs, DVDs, music cassettes, VHS videotapes, vinyl	\$4.49	\$2.49
Books*	\$4.49	\$4.49
Any combination of the above items	Highest applicable per-shipment charge	As above

Expedited International Shipping

	Per Shipment	Per Item
CDs, DVDs, music cassettes, VHS videotapes, vinyl	\$7.49	\$2.99
Books*	\$7.49	\$5.49
Any combination of the above items	Highest applicable per-shipment charge	As above

On average 10 clicks to find out what the shipping rate is!

The solution particularly for Web integration:

◆ 2 alternatives:

- **Top-down: Create wrappers for current web sites and extract data automatically (wrappers)**
 - ◆ *Today we mainly focus on this part!*
- Bottom-up: Instead of publishing natural language, publish machine-processable data directly (semantic Web idea!)

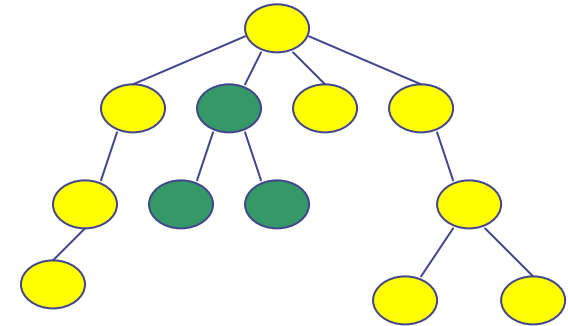
Wrappers for Websites:

- ◆ Create XML from Websites
- ◆ Advantage for Web-based integration:
HTML from websites is already very close to XML (or even XHTML already!)
- ◆ with XSLT, XPath, etc. we already have almost everything done (for tree-based wrapping)!

Wrappers for Websites – Ways of extraction:

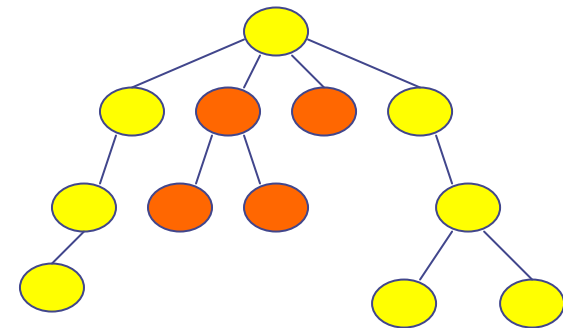
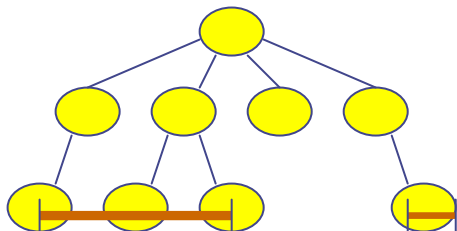
◆ Tree extraction

- Distinction "subtree" und "tree region"
- A subtree is represented by its root (a node in the original tree)
- A tree region consists of a list of sibling nodes.



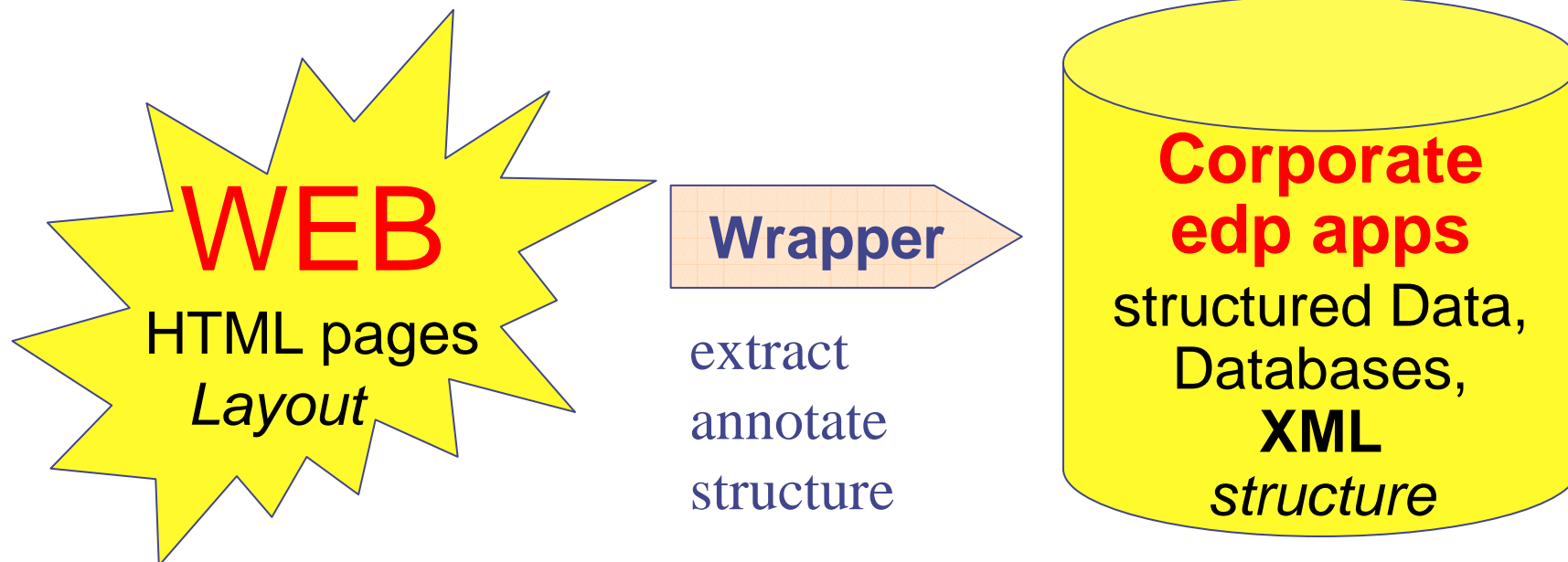
◆ Stringextraktion

- Operates on substrings of the (HTML) Document, e.g. by regular expressions.



Motivation for Web Extraction: Bridge the Gap

Goal: Make Web content accessible for electronic data exchange.



An Example: deri.at members page

The screenshot shows the DERI Innsbruck members page in Mozilla Firefox. The browser title is "DERI Innsbruck . Digital Enterprise Research Institute - Mozilla Firefox". The address bar shows "http://www.deri.at/aboutderi/members/". The page content includes a sidebar with navigation links and a main area with member profiles. A blue arrow points from the profile of Alice Carpentier to a yellow box containing XML code.

DERI Innsbruck . Digital Enterprise Research Institute - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://www.deri.at/aboutderi/members/

onlineLehrveranstalt... W3C Member Site Institut AIFB - Grun... akaedmischesjahr DBLP Bibliography ResearchIndex [NE... >>

current projects
completed projects
working groups
research clusters
deri mailinglists
deri research seminar
ontology

Teaching


summer 2005
master course
archive

Business Opportunities


business outreach

Events


events
future events
earlier events
invited talks
future talks
earlier talks

 Institute of Computer Science
University of Innsbruck
Phone: +43 512 507 6482
Fax: +43 512 507 9872
E-Mail: johannes.breitfuss@deri.org


Johannes Breitfuss

 Institute of Computer Science
University of Innsbruck
Phone: +43 512 507 6475
Fax: +43 512 507 9872
E-Mail: jos.debruijn@deri.org

Jos de Bruijn

 **Alice Carpentier**
www.deri.org/members/alicec
Institute of Computer Science
University of Innsbruck
Phone: +43 512 507 6484
Fax: +43 512 507 9872
E-Mail: alice.carpentier@deri.org


Alice Carpentier

 **Dr. Ying Ding**
www.deri.org/
Institute of Co
University of In
Phone: +43 51
Mobile: +43 66
Fax: +43 512
E-Mail: ying.di

Ying Ding

 **Cristina Feier**
www.deri.org/members/cristinaf
Institute of Computer Science
University of Innsbruck
Phone: +43 512 507 6453
Fax: +43 512 507 9872
E-Mail: cristina.feier@deri.org

Cristina Feier

 **Univ.-Prof. Dr.**
www.deri.org/
Institute of Co
University of In
Phone: +43 51
Fax: +43 512
E-Mail: dieter.

Dieter Fensel

```
<members>  
...  
<person>  
  <FName>Alice</FName>  
  <LName>Carpentier</LName>  
  <email>alice.carpentier@uibk.ac.at</email>  
</person>  
...  
<person>  
  <title>Univ.-Prof. Dr. </title>  
  <FName>Dieter</FName>  
  <LName>Fensel</LName>  
  <url>http://www.fensel.com</url>  
</person>  
...  
</members>
```

Other examples:

- Combine results from different search engines
- Stock quotes
- News filters (from several pages)
- Price-comparison (e.g. I want to buy a laptop)
- Keep informed about concurrent competitors
- etc.

Common Task:

Similar looking pages, where you want to semi-automatically wrap information out of the HTML structure.

Additional, useful features would be:

- Automatic requerying schedule
- Automatization of forms and login/access control!
- notification of the wrapper does not work any longer due to site changes, etc.

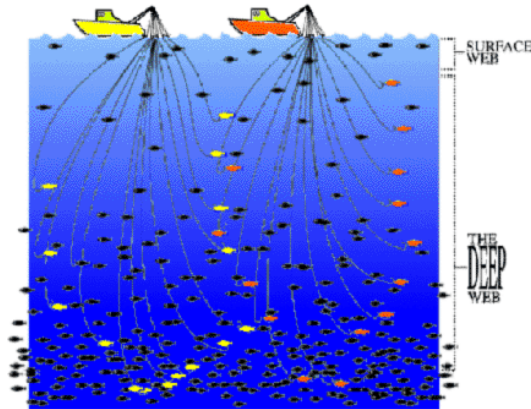
Extraction from Websites

- ◆ more problems before actual extraction step:
Most services on the web interact with Web interfaces (forms, applets, htaccess, logins, etc.)...

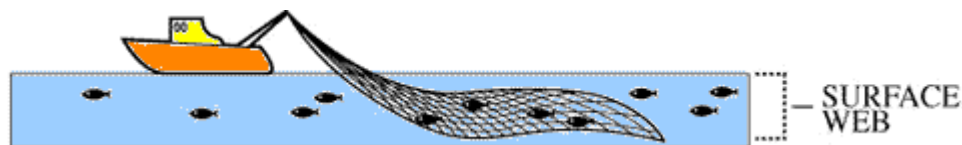
The screenshot shows the Alitalia website interface in a Mozilla Firefox browser. The browser window title is "Alitalia - Home - Mozilla Firefox" and the address bar contains "http://www.alitalia.at/". The website features a navigation menu on the left with options like "Buchung und kaufen", "Flugplan", and "Angebote". The main content area is titled "Ticket buchen und kaufen" and includes a flight booking form with fields for "Von:" (milano) and "Nach:" (san francisco), departure and return dates (11 April), and flight class (Economy). A "Suchen" button is visible. Below the booking form, there is a section for "Aktuelle Informationen und Angebote" featuring a promotion for "Frühling in Rom" with prices starting from 116,- €.

Extraction from Websites

- ◆ ... the real data is hidden in this "deep web"



- ◆ much more data than on the "publicly indexable" surface web



- ◆ "The invisible portion of the Web will continue to grow exponentially before the tools to uncover the hidden Web are ready for general use"

<http://www.press.umich.edu/jep/07-01/bergman.html>

- ◆ Need for **scripting** of interaction with forms, etc. for dynamic wrappers!

Web integration and extraction – Basics 1/3

HTTP Protocol

- ◆ Web Servers use HTTP Protocol:
- ◆ Request: HTTP GET, HTTP POST:

Request method GET or POST

Requested source

```
GET /main.html HTTP/1.0
accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/vnd.ms-powerpoint, application/vnd.ms-excel,
application/msword, */*
proxy-connection: Keep-Alive
host: www.philips.de
accept-language: de
connection: keep-alive
user-agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
```

host

Header can additionally contain info on e.g. whether cookies are accepted, etc.

<request body>

Normally empty for HTTP GET, parameter data for HTTP POST

Example of how parameters can be encoded in a get request:

```
GET /search/de?o=1&p=chocolate&co=i&h=c&g=0&n=20 HTTP/1.0
```

...

```
host: de.search.yahoo.com
```

Web integration and extraction – Basics 1/3

HTTP Protocol

◆ HTTP Reply:

Some alternatives: 404 (not found), 302 (page moved)

```
HTTP/1.0 200 OK
proxy-connection: keep-alive
accept-ranges: bytes
cache-control: no-cache
content-type: text/html
connection: keep-alive
content-length: 2627
server: Apache/1.3.12 (Unix) (Red Hat/Linux) PHP/4.0.6
mod_perl/1.21
```

Content type!

<HTML>...

Usually HTML

We need to know these basics in order to write some automatic wrappers...

Website Wrapper Tools examples

- ◆ A complex example: LiXto Visual Wrapper (elog): not only for Web sources, also PDF, etc.
- ◆ Back to basic! Tidy+XSLT transformations

We'll only briefly demonstrate the visual wrapper here

Quell
Publikation

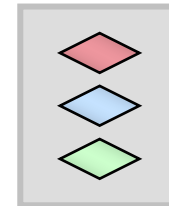
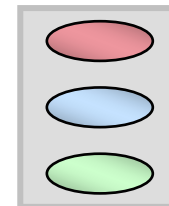
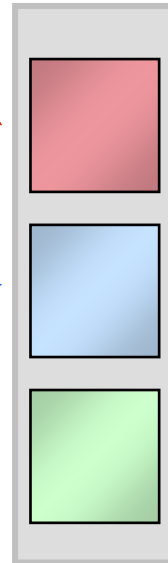
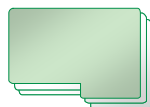
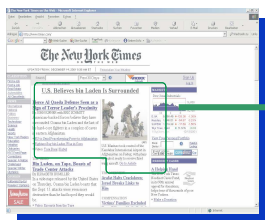
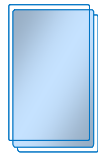
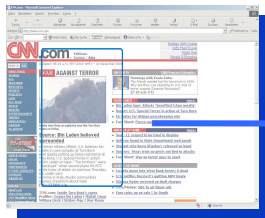
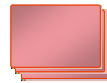
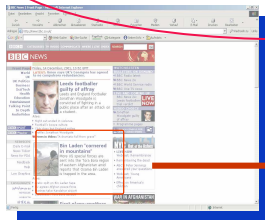
Extraktion

Integration Transformation

Publikation

Lixto Visual Wrapper

Lixto Transformation Server



Cell phone

Browser

email

HTML

XML

WML, XHTML, Text

Our example:

DERI Innsbruck . Digital Enterprise Research Institute - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://www.deri.at/aboutderi/members/

onlineLehrveranstalt... W3C Member Site Institut AIFB - Grun... akaedmischesjahr DBLP Bibliography ResearchIndex [NE...]

DERI INNSBRUCK

deri.at home deri international jobs@deri

About DERI

- mission
- partners
- sponsors
- management
- members
 - current members
 - archive
- links
- location

Publications

- books
- presentations
- technical papers

Research

- cooperations
- projects
 - current projects
 - completed projects
- working groups
- research clusters
- deri mailinglists
- deri research seminar
- ontology

Teaching

- summer 2005

deri.members

Members Total: 36

Sinuhé Arroyo
www.deri.org/members/sinuhe
Institute of Computer Science
University of Innsbruck
Phone: +43 512 507 6480
Fax: +43 512 507 9872
E-Mail: sinuhe.arroyo@deri.org

Sinuhé Arroyo

Daniel Bachlechner
www.deri.org/members/danielb
Institute of Computer Science
University of Innsbruck
Phone: +43 512 507 96822, 96823
Fax: +43 512 507 9872
E-Mail: daniel.bachlechner@deri.org

Daniel Bachlechner

Johannes Breitfuss
www.deri.org/members/johannesb
Institute of Computer Science
University of Innsbruck
Phone: +43 512 507 6482
Fax: +43 512 507 9872
E-Mail: johannes.breitfuss@deri.org

Johannes Breitfuss

Jos de Bruijn
www.deri.org/members/josd
Institute of Computer Science
University of Innsbruck
Phone: +43 512 507 6475
Fax: +43 512 507 9872
E-Mail: jos.debruijn@deri.org

Jos de Bruijn

Patterns:

Name: bold, always on a similar position in the table

Title: always occurs exactly before name as part of the name

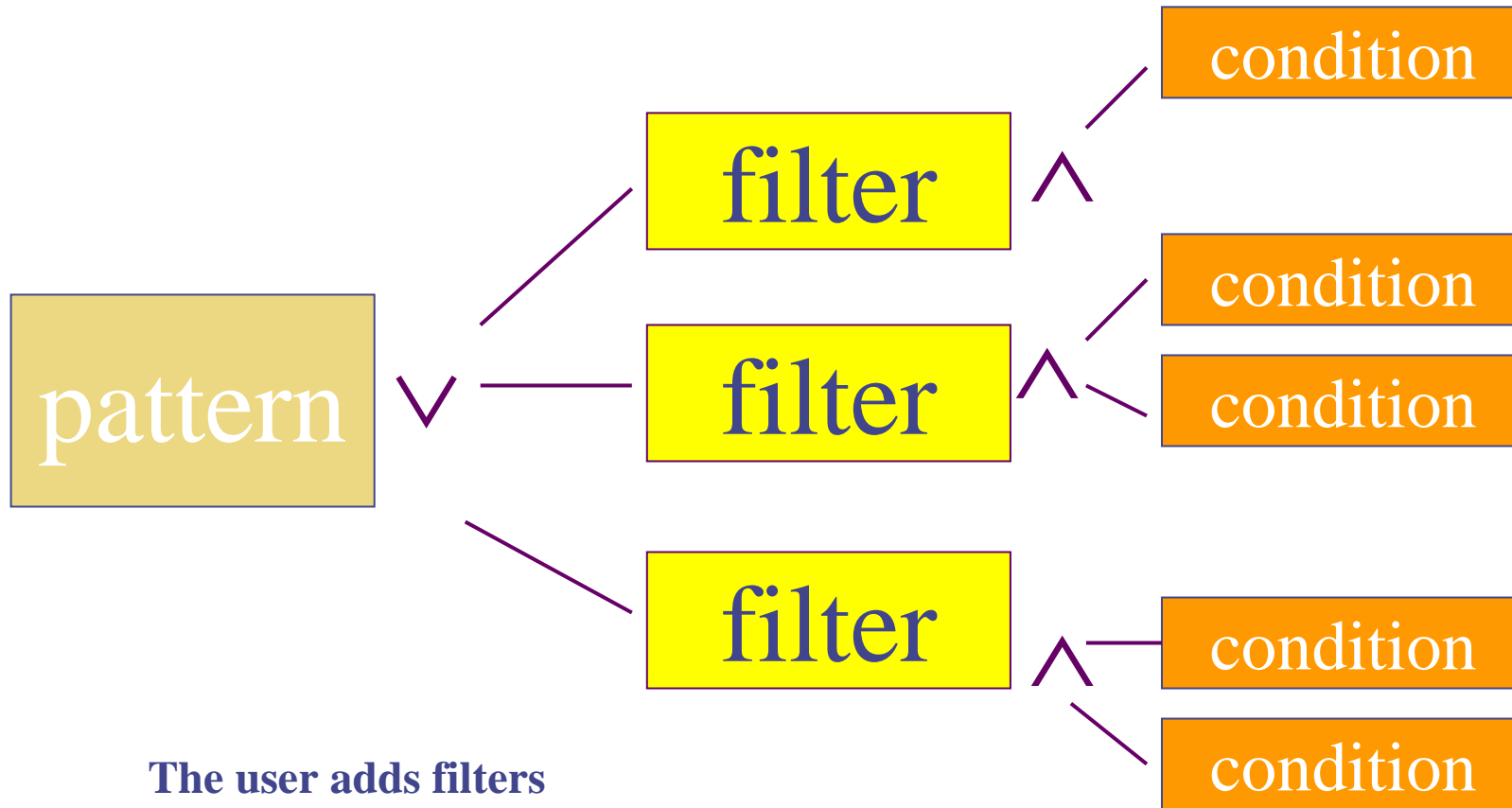
Email: is a link, preceded by the String "E-Mail:"

Lixto: Wrapper Generation

Different approach to XSLT:

- ◆ **Example-based** approach
- ◆ **Visual** interaction
- ◆ Declarative Language Elog
- ◆ Hierarchical Extraction using Patterns & Filters.
- ◆ Extract XML with XML Tool
- ◆ Tree-based and string-based, regular expressions
- ◆ "Fuzzy" filter functions, etc.

Patterns in Lixto:



The user adds filters and conditions, characterizing the wanted information patterns by stepwise refinement/extension

- ◆ A Wrapper in Lixto consists of a set of (hierarchical) patterns
- ◆ Each filter extracts a set of instances
- ◆ Conditions restrict the numbers of instances
- ◆ The instances of a pattern is the union of the instances of its filters

Different Patterns:

◆ Tree Patterns:

- ◆ Use HTML properties and the tree structure for identifying relevant elements or element lists

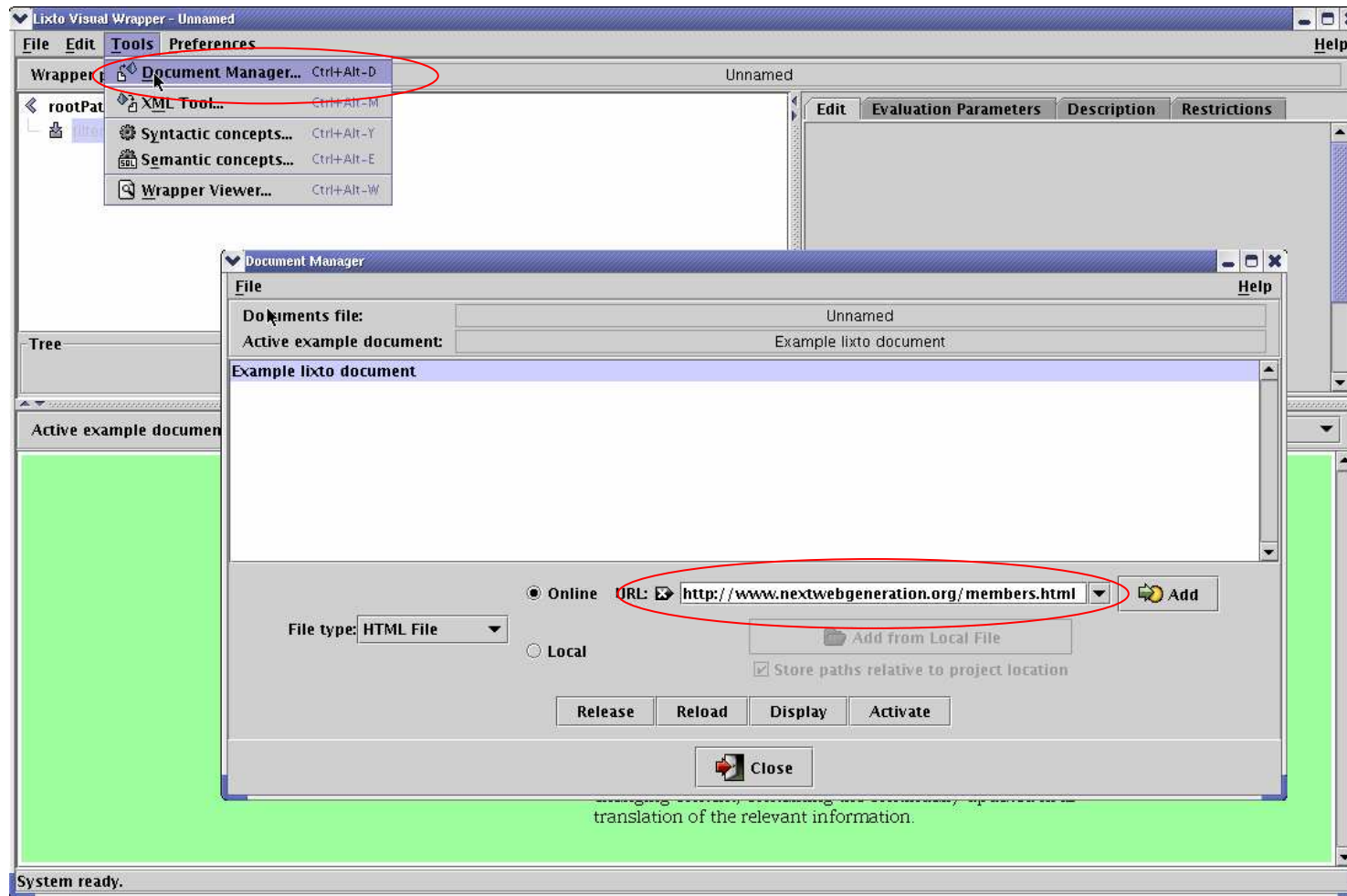
◆ String Patterns:

- ◆ operate on “flat” strings (e.g. divide first name and last name by regexps),
- ◆ Mostly for HTML leaf elements
- ◆ can also be used for “invisible” content (attributes)

◆ Document Patterns: Link to other documents:

- ◆ Allow to navigate to further documents (e.g. via a “Next page” link in the page you are wrapping)

LiXto: Add the URL of a page to wrap: The Document manager



Lixto: Patterns & Filters

Current program (wrapper)

Define and modify patterns and filters here

Browser Window for selecting and testing filters...

Willkommen im Lixto Visual Wrapper!

Lixto Visual Wrapper ist ein interaktives System zur Wrappergenerierung, basierend auf Algorithmen zur Erstellung von Wrapperprogrammen und der deklarativen logikbasierten Sprache Elog.

Lixto Visual Wrapper ist plattformunabhängig, verfügt über eine interaktive visuelle Benutzerschnittstelle, erlaubt ausdrucksfähige, flexible hierarchische Datenextraktion unter Verwendung eines HTML Baumes sowie Stringextraktionstechniken.

Lixto Visual Wrapper übersetzt relevante Teile von Webseiten nach XML. Es kann ein sogenannter *XML Companion* für eine Webseite mit

System bereit.

Alternative to Wrapper Tools:

- ◆ Doing it from scratch, using APIs and JAVA... Ingredients:

- ◆ W3C Tidy:

<http://tidy.sourceforge.net/>

<http://jtidy.sourceforge.net/>

- ◆ XSLT processor e.g. Apache XALAN:

Java: <http://xml.apache.org/xalan-j/index.html>

C++: <http://xml.apache.org/xalan-c/index.html>

The solution particularly for Web integration:

◆ 2 alternatives:

- Top-down: Create wrappers for current web sites and extract data automatically (wrappers)
 - ◆ *Today we mainly focus on this part!*
- **Bottom-up: Instead of publishing natural language, publish machine-processable data directly (semantic Web idea!)**

Towards a semantic Web:

- ◆ What's missing?

Semantic Matching! An Example:

- ◆ An agent for the Internet Shopping domain:

Program a KB-agent to search relevant pages for a certain query?

Q: What means relevant?

Generic Online Store

Select from our fine line of products:

- Computers
- Cameras
- Books
- Videos
- Music

```
<h1>Generic Online Store</h1>
<i>Select</i> from our fine line of products:
<ul>
<li> <a href="http://gen-store.com/compu">Computers</a>
<li> <a href="http://gen-store.com/camer">Cameras</a>
<li> <a href="http://gen-store.com/books">Books</a>
<li> <a href="http://gen-store.com/video">Videos</a>
<li> <a href="http://gen-store.com/music">Music</a>
</ul>
```

Example: What knowledge does an agent for the Internet Shopping domain need?

"I'll find relevant offers at pages linked from OnlineStores which are linked via pages **relevant to my query** and which contain an offer"

"An offer is a page which contains an object relevant to my query and an **option to buy** or a **price**."

Amazon 2 OnlineStores $\mathcal{A}E$ Homepage(amazon,"http://www.amazon.com/")

Ebay 2 OnlineStores $\mathcal{A}E$ Homepage(Ebay,"http://www.ebay.com/")

GenStore 2 OnlineStores $\mathcal{A}E$ Homepage(GenStore,"http://www.gen-store.com/")

Relevant(page,url,query) , \exists store, home store 2 OnlineStores $\mathcal{A}E$ Homepage(store,home)

$\mathcal{A}E \exists$ url₂ RelevantChain(home,url₂,query) $\mathcal{A}E$ Link(url₂,url) $\mathcal{A}E$ page=GetPage(url)

RelevantChain(start,end,query) , start=end

ζ (\exists u,text LinkText(start,u,text $\mathcal{A}E$ RelevantCategoryName(query,text) $\mathcal{A}E$ RelevantChain(u,end,query))

RelevantCategoryName(query,text) ,

\exists c₁,c₂ Name(query,c₁), Name(text,c₂) $\mathcal{A}E$ (c₁ μ c₂ ζ c₂ μ c₁)

The last logical expression says (informally) something like:

"An link is relevant to my query if it has a category which is a superclass or a subclass of my queried category in the link text."

Example: An agent for the Internet Shopping domain

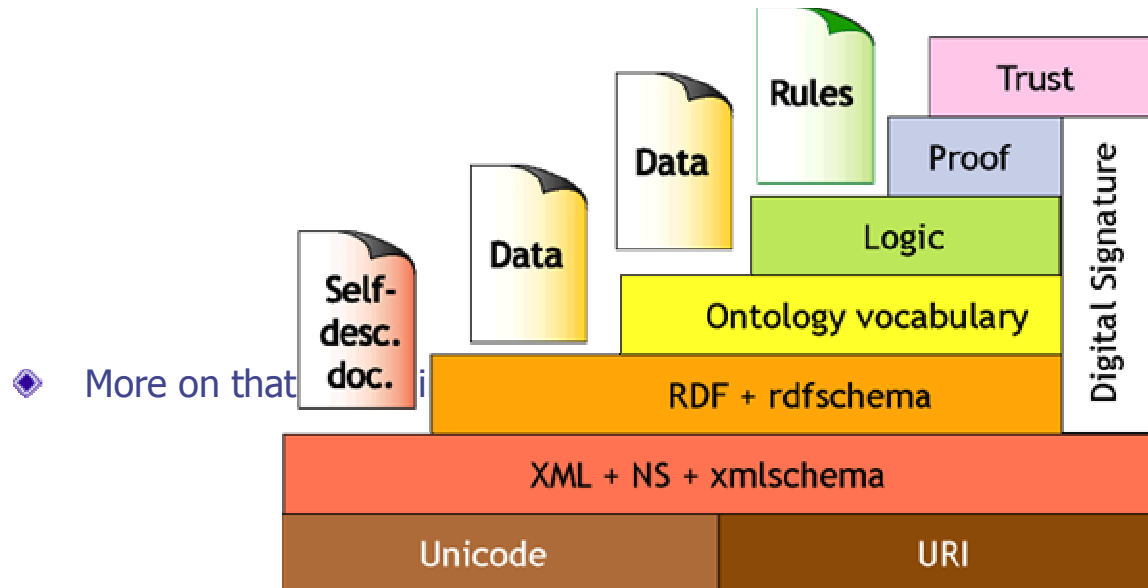
◆ Taxonomy for product categories:

<i>Books</i> \subset <i>Products</i>	<i>Name</i> ("books", <i>Books</i>)
<i>MusicRecordings</i> \subset <i>Products</i>	<i>Name</i> ("music", <i>MusicRecordings</i>)
<i>MusicCDs</i> \subset <i>MusicRecordings</i>	<i>Name</i> ("CDs", <i>MusicCDs</i>)
<i>MusicTapes</i> \subset <i>MusicRecordings</i>	<i>Name</i> ("tapes", <i>MusicTapes</i>)
<i>Electronics</i> \subset <i>Products</i>	<i>Name</i> ("electronics", <i>Electronics</i>)
<i>DigitalCameras</i> \subset <i>Electronics</i>	<i>Name</i> ("digital cameras", <i>DigitalCameras</i>)
<i>StereoEquipment</i> \subset <i>Electronics</i>	<i>Name</i> ("stereos", <i>StereoEquipment</i>)
<i>Computers</i> \subset <i>Electronics</i>	<i>Name</i> ("computers", <i>Computers</i>)
<i>LaptopComputers</i> \subset <i>Computers</i>	<i>Name</i> ("laptops", <i>LaptopComputers</i>)
<i>DesktopComputers</i> \subset <i>Computers</i>	<i>Name</i> ("desktops", <i>DesktopComputers</i>)
...	...
(a)	(b)

- ◆ For successful search on the Internet, taxonomies of categories are important! Make the structure of knowledge available online! Ontologies!

Semantic Web

- ◆ Instead of publishing natural language, publish machine-processable meta-data directly (semantic Web idea!)
- ◆ Provide standards on top of XML to describe the meaning of published knowledge
- ◆ This meta-data shall ideally also enable standardization of also the wrapping step.
- ◆ Provide the means to publish data on relations and taxonomies of data on the Web

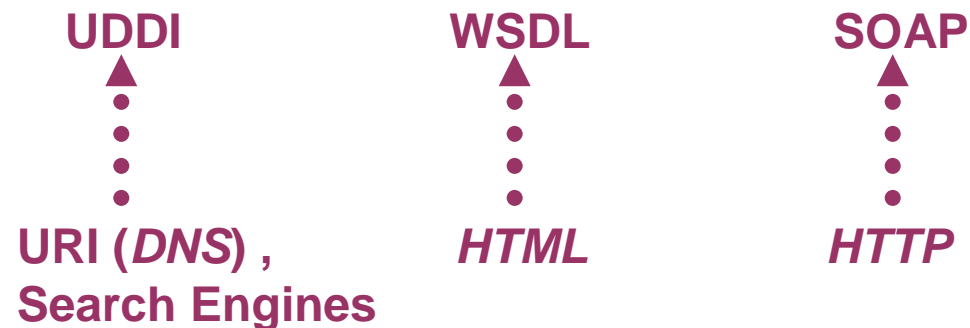


Web Services

- ◆ By annotating Web static Web pages with meta-data, we still haven't solved the issue about interaction with services on the Web.
- ◆ Web services standardize this interaction, i.e. standardize service integration like XML standardized data representation!

Web Services: SOAP, WSDL, UDDI

- ◆ SOAP: common protocol for message exchange
- ◆ WSDL: a language for defining interfaces, partly based on XML Schema
- ◆ UDDI(Universal Description, Discovery and Integration): a repository and API standard for advertising and finding Web services.



- ◆ Later coming lectures...

References

- ◆ **XPath:** <http://www.w3.org/TR/xpath>
- ◆ **XSLT:** <http://www.w3.org/Style/XSL/>
- ◆ **LiXto:** <http://www.lixto.com/> (only commercial, no downloads :-()
- ◆ **More Wrapper Tools** (a bit outdated)
<http://www.wifo.uni-mannheim.de/~kuhlins/wrappertools/>