The lixto System

Applications in Business Intelligence & the Semantic Web

Georg Gottlob

Oxford University

ESWC Innsbruck 2007

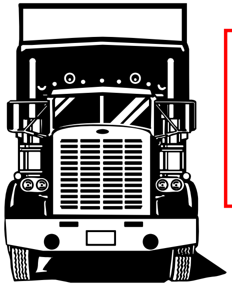
Technische Universität Wien
Vienna University of Technology
TU WIEN
DBAI

Talk Outline

- Motivation: Need of Web data extraction
- The LiXto Approach (+ short demo)
- Theoretical Results
- Lixto Suite + Applications
- Creating RDF Bases
- Research Issues: Web feature extraction etc.

Electronic Commerce

E-procurement,
quality assurance:



Quality norm 06865 for tires

OEM PORTAL

IVECO

Pirelli

Michelin

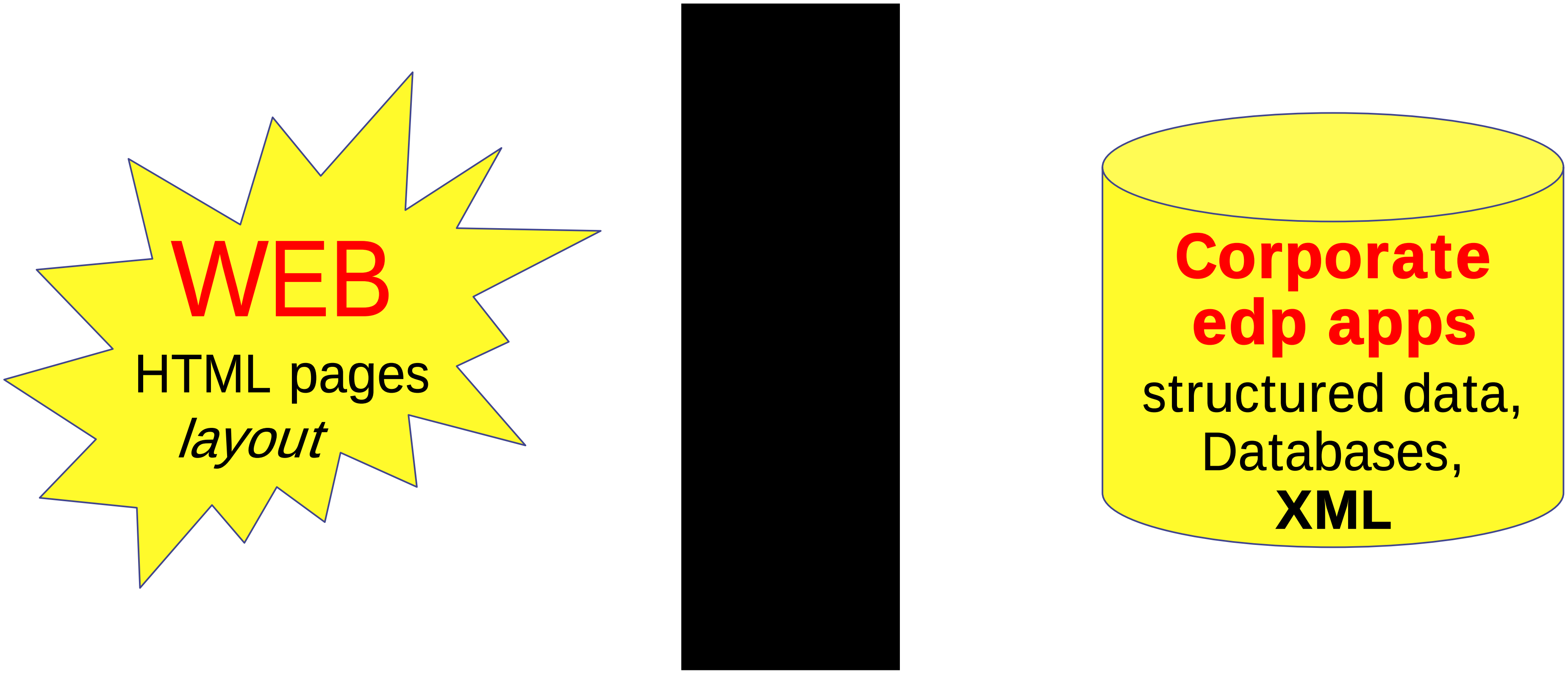
Avon

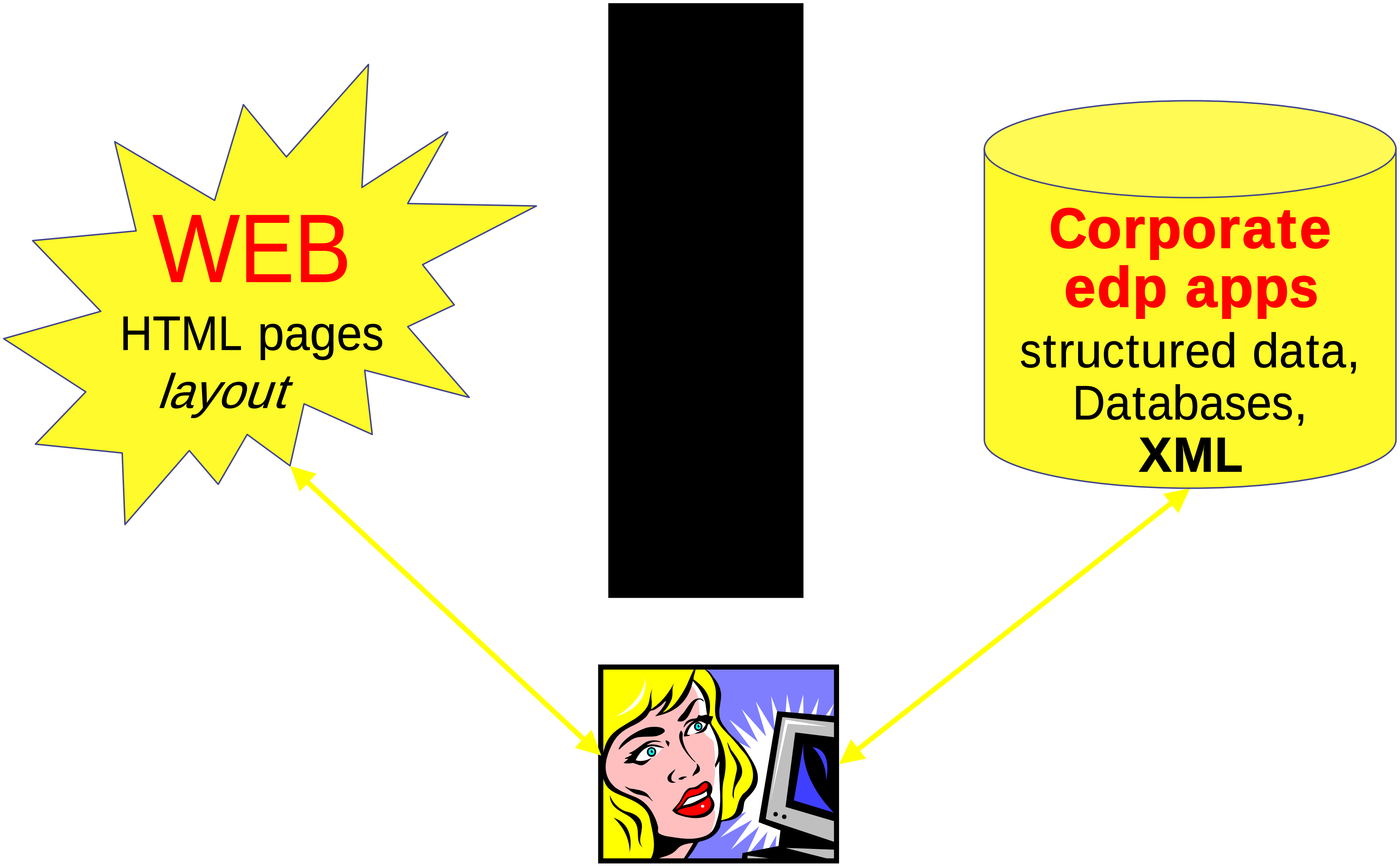
Dunlop

Firestone

The Wall

Problem: Make web contents accessible to electronic data processing




WEB
HTML pages
layout
Corporate
edp apps
structured data,
Databases,
XML

Solutions:

Semantic Web ? Yes, but...

Intelligent agents? Yes, but...

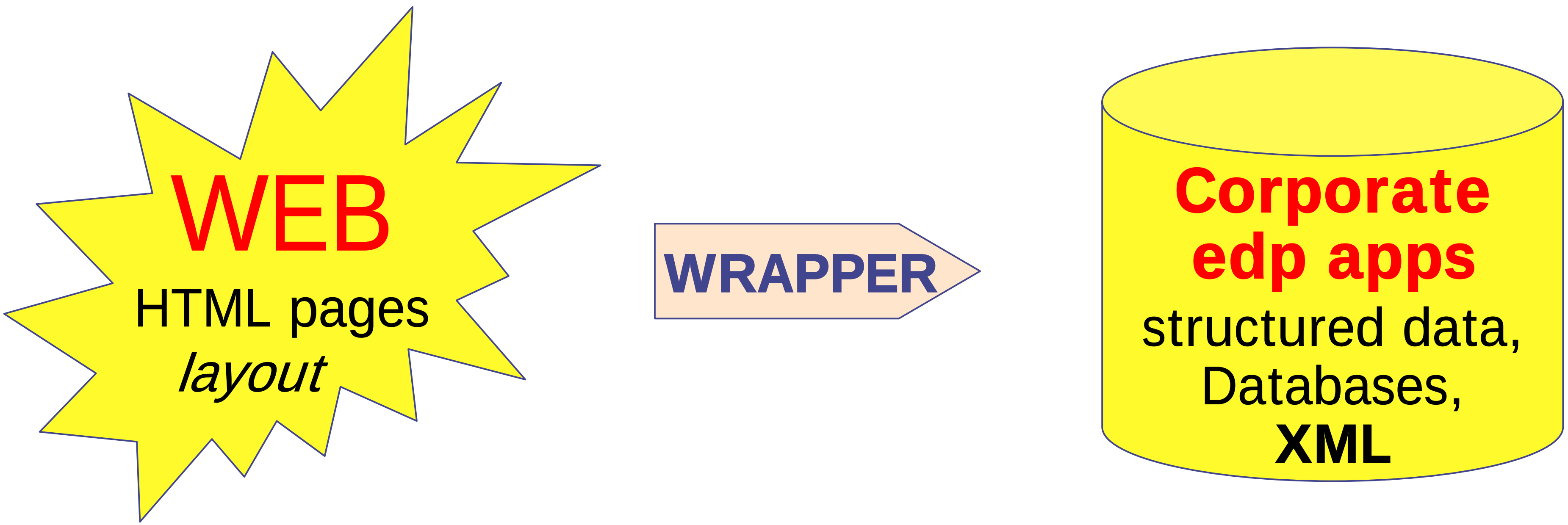
Web wrapping

Web information extraction

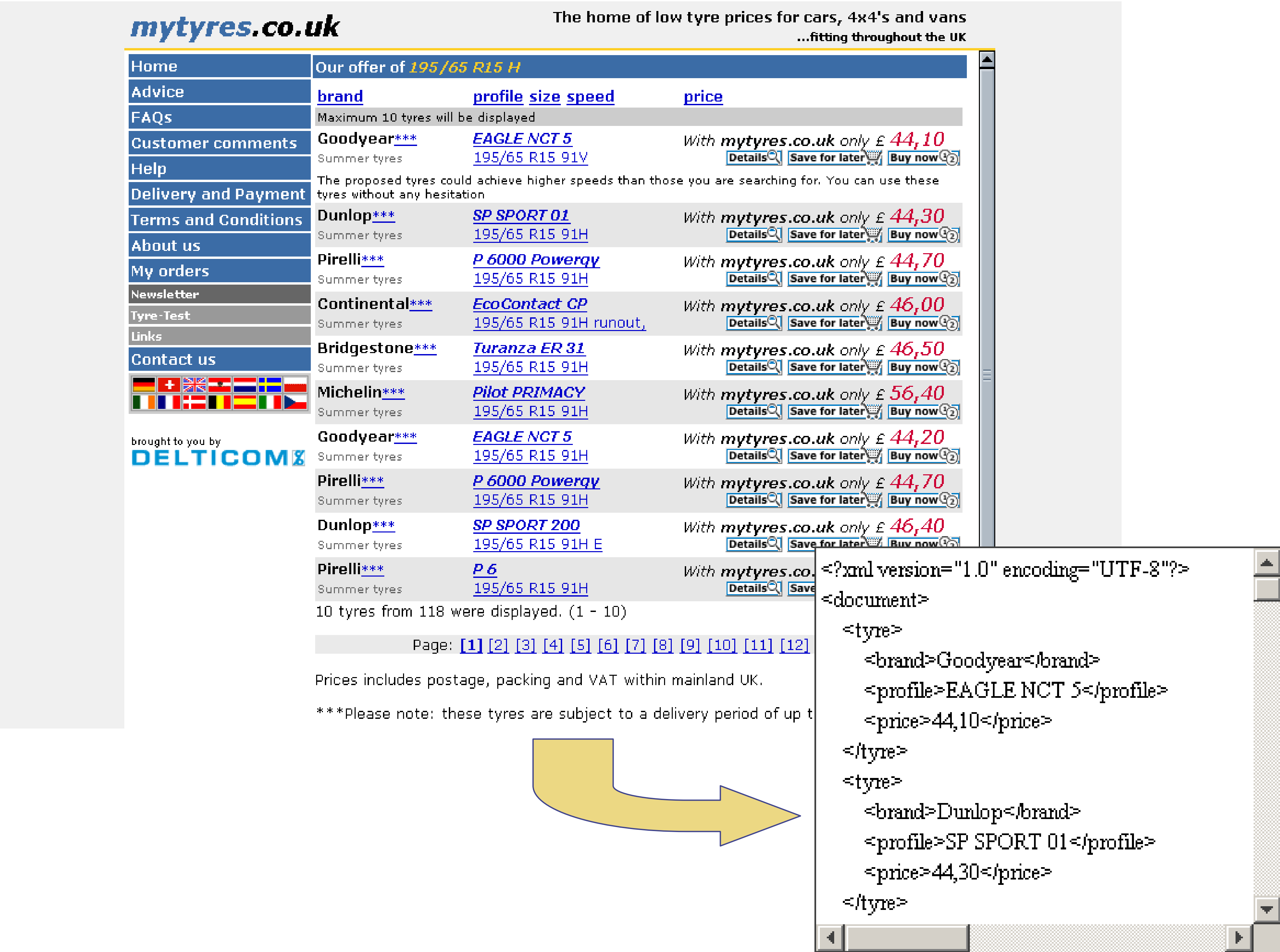
Goal: Make web contents accessible to electronic data processing

Web information extraction

Goal: Make web contents accessible to electronic data processing

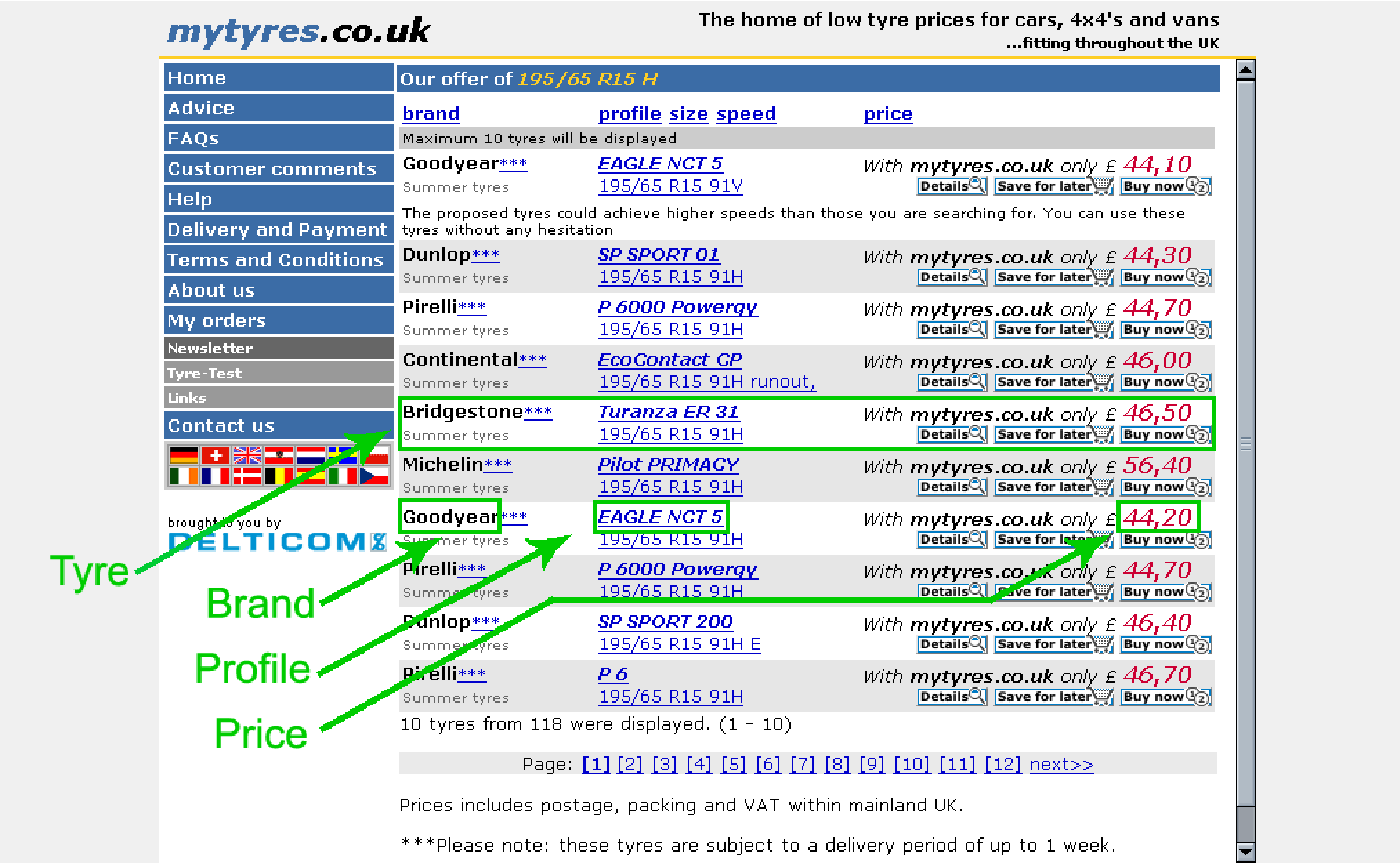


Wrappers: **HTML**→select → extract → annotate →**XML**

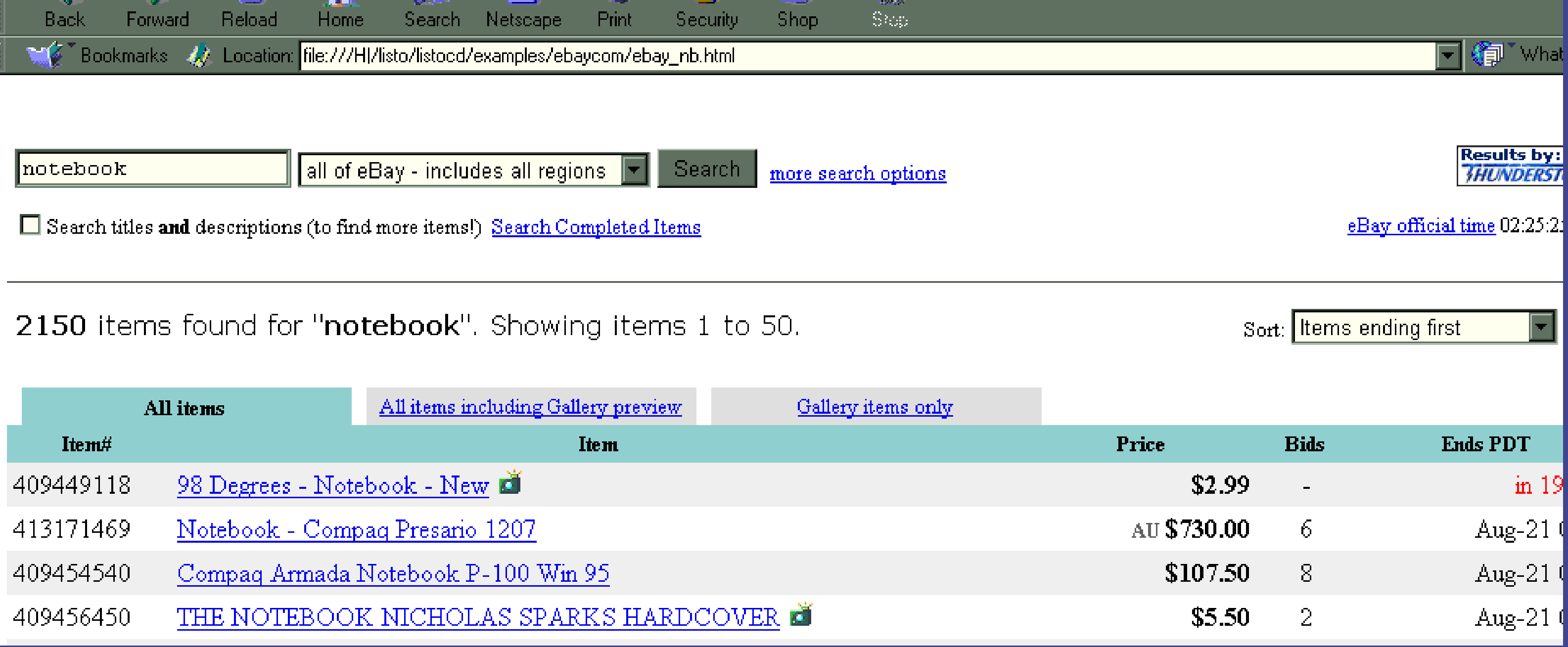


mytyres.co.uk
The home of low tyre prices for cars, 4x4's and vans
...fitting throughout the UK
Home
Advice
FAQs
Customer comments
Help
Delivery and Payment
Terms and Conditions
About us
My orders
Newsletter
Tyre-Test
Links
Contact us
brought to you by
DELTICOM
Our offer of 195/65 R15 H
brand profile size speed price
Maximum 10 tyres will be displayed
Goodyear*** Summer tyres EAGLE NCT 5 195/65 R15 91V With mytyres.co.uk only £ 44,10 Details Save for later Buy now
The proposed tyres could achieve higher speeds than those you are searching for. You can use these tyres without any hesitation
Dunlop*** Summer tyres SP SPORT 01 195/65 R15 91H With mytyres.co.uk only £ 44,30 Details Save for later Buy now
Pirelli*** Summer tyres P 6000 Powergy 195/65 R15 91H With mytyres.co.uk only £ 44,70 Details Save for later Buy now
Continental*** Summer tyres EcoContact CP 195/65 R15 91H runout, With mytyres.co.uk only £ 46,00 Details Save for later Buy now
Bridgestone*** Summer tyres Turanza ER 31 195/65 R15 91H With mytyres.co.uk only £ 46,50 Details Save for later Buy now
Michelin*** Summer tyres Pilot PRIMACY 195/65 R15 91H With mytyres.co.uk only £ 56,40 Details Save for later Buy now
Goodyear*** Summer tyres EAGLE NCT 5 195/65 R15 91H With mytyres.co.uk only £ 44,20 Details Save for later Buy now
Pirelli*** Summer tyres P 6000 Powergy 195/65 R15 91H With mytyres.co.uk only £ 44,70 Details Save for later Buy now
Dunlop*** Summer tyres SP SPORT 200 195/65 R15 91H E With mytyres.co.uk only £ 46,40 Details
Pirelli*** Summer tyres P 6 195/65 R15 91H With mytyres.co. Details
10 tyres from 118 were displayed. (1 - 10)
Page: [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]
Prices includes postage, packing and VAT within mainland UK.
***Please note: these tyres are subject to a delivery period of up t
<?xml version="1.0" encoding="UTF-8"?>
<document>
<tyre>
<brand>Goodyear</brand>
<profile>EAGLE NCT 5</profile>
<price>44,10</price>
</tyre>
<tyre>
<brand>Dunlop</brand>
<profile>SP SPORT 01</profile>
<price>44,30</price>
</tyre>

Patterns:



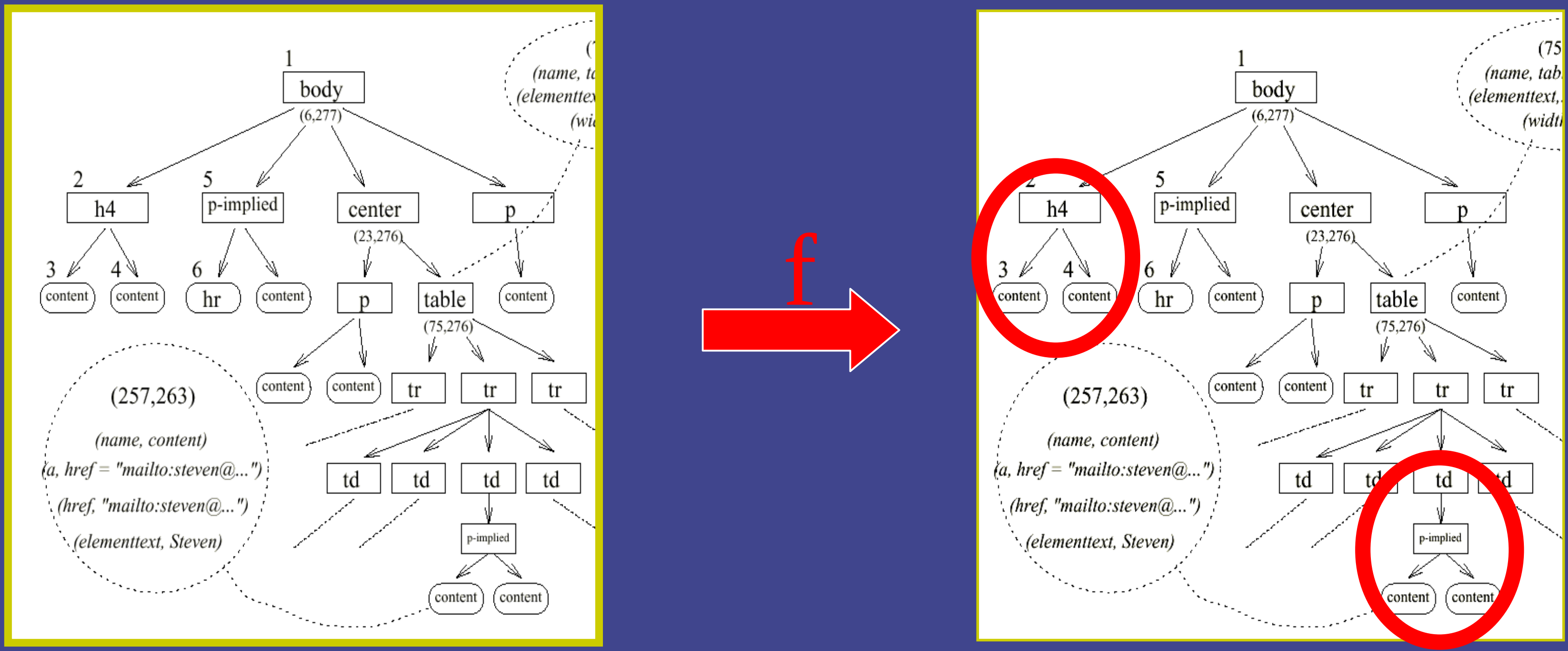
mytyres.co.uk
The home of low tyre prices for cars, 4x4's and vans
...fitting throughout the UK
Home
Advice
FAQs
Customer comments
Help
Delivery and Payment
Terms and Conditions
About us
My orders
Newsletter
Tyre-Test
Links
Contact us
brought to you by
DELTICOM
Our offer of 195/65 R15 H
brand
profile size speed
price
Maximum 10 tyres will be displayed
Goodyear*** Summer tyres EAGLE NCT 5 195/65 R15 91V With mytyres.co.uk only £ 44,10
The proposed tyres could achieve higher speeds than those you are searching for. You can use these tyres without any hesitation
Dunlop*** Summer tyres SP SPORT 01 195/65 R15 91H With mytyres.co.uk only £ 44,30
Pirelli*** Summer tyres P 6000 Powergy 195/65 R15 91H With mytyres.co.uk only £ 44,70
Continental*** Summer tyres EcoContact CP 195/65 R15 91H runout, With mytyres.co.uk only £ 46,00
Bridgestone*** Summer tyres Turanza ER 31 195/65 R15 91H With mytyres.co.uk only £ 46,50
Michelin*** Summer tyres Pilot PRIMACY 195/65 R15 91H With mytyres.co.uk only £ 56,40
Goodyear*** Summer tyres EAGLE NCT 5 195/65 R15 91H With mytyres.co.uk only £ 44,20
Pirelli*** Summer tyres P 6000 Powergy 195/65 R15 91H With mytyres.co.uk only £ 44,70
Dunlop*** Summer tyres SP SPORT 200 195/65 R15 91H E With mytyres.co.uk only £ 46,40
Pirelli*** Summer tyres P 6 195/65 R15 91H With mytyres.co.uk only £ 46,70
Details
Save for later
Buy now
10 tyres from 118 were displayed. (1 - 10)
Page: [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] next>>
Prices includes postage, packing and VAT within mainland UK.
***Please note: these tyres are subject to a delivery period of up to 1 week.
Tyre
Brand
Profile
Price




```xml
<?xml version="1.0" encoding="UTF-8"?>
<document>
     <record>
          <number>409449118</number>
          <item>98 Degrees - Notebook - New</item>
          <picture/>
          <price>2.99</price>
          <currency>$</currency>
          <bids>-</bids>
     </record>
     <record>
          <number>413171469</number>
          <item>Notebook - Compaq Presario 1207</item>
          <price>730.00</price>
          <currency>AU $</currency>
[...]
```

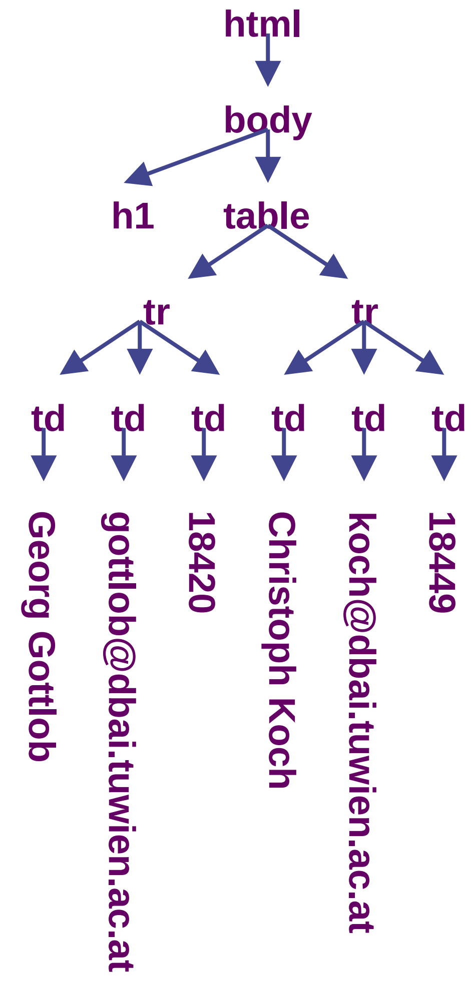
Function f: HTML Parse tree ➲ Subtrees

Leaves of subtrees are among leaves of orig. tree



HTML page = tree

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html> <body>
<h1>People @ DBAI</h1>
<table border="1" cellpadding="3" cellspacing="1">
  <tr> <td>Georg Gottlob</td>
       <td>gottlob@dbai.tuwien.ac.at</td>
       <td>18420</td>
  </tr>
  <tr> <td>Christoph Koch</td>
       <td>koch@dbai.tuwien.ac.at</td>
       <td>18449</td>
  </tr>
</table>
</body> </html>
```

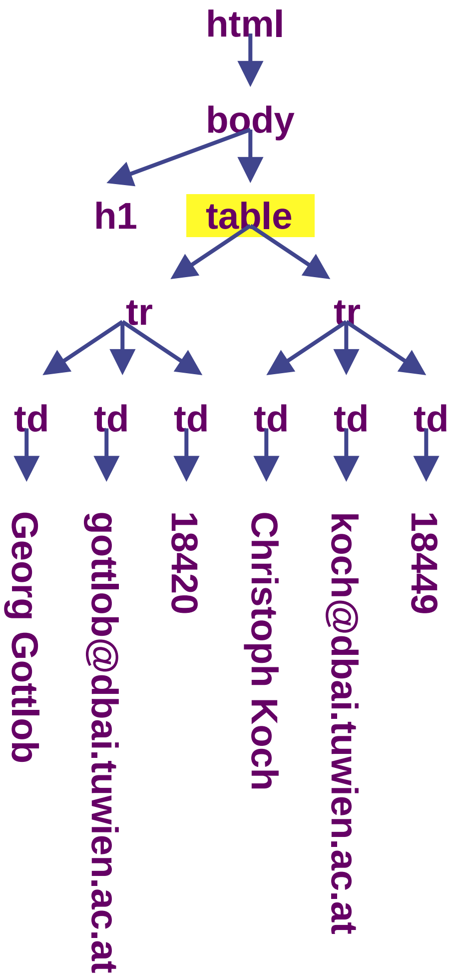


People @ DBAI

Georg Gottlob	gottlob@...	18420
Christoph Koch	koch@...	18449

Predicate *employeetable*

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html> <body>
<h1>People @ DBAI</h1>
<table border="1" cellpadding="3" cellspacing="1">
    <tr> <td>Georg Gottlob</td>
            <td>gottlob@dbai.tuwien.ac.at</td>
            <td>18420</td>
    </tr>
    <tr> <td>Christoph Koch</td>
            <td>koch@dbai.tuwien.ac.at</td>
            <td>18449</td>
    </tr>
</table>
</body> </html>
```

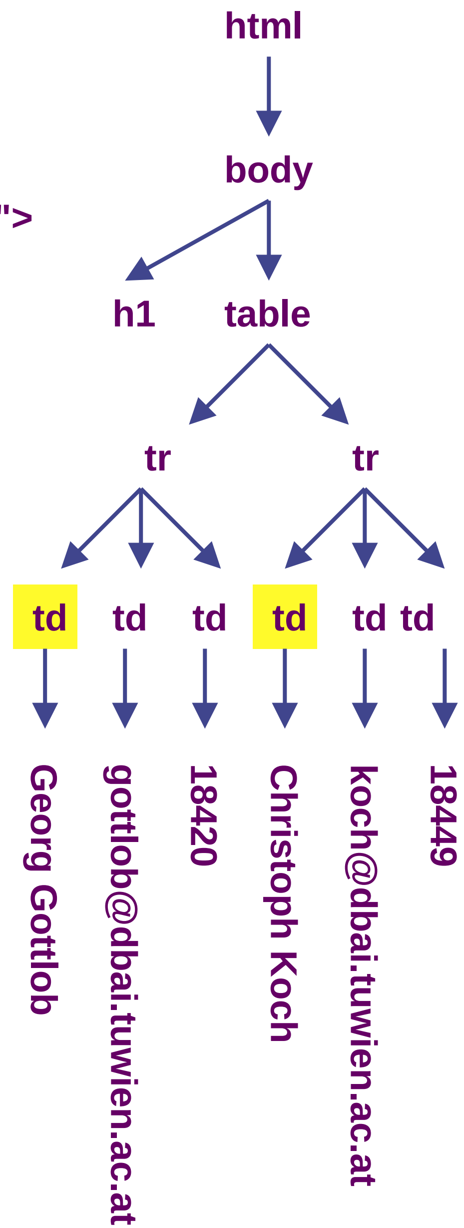


People @ DBAI

Georg Gottlob	gottlob@...	18420
Christoph Koch	koch@...	18449

Predicate *employee*

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html> <body>
<h1>People @ DBAI</h1>
<table border="1" cellpadding="3" cellspacing="1">
   <tr> <td>Georg Gottlob</td>
        <td>gottlob@dbai.tuwien.ac.at</td>
        <td>18420</td>
   </tr>
   <tr> <td>Christoph Koch</td>
        <td>koch@dbai.tuwien.ac.at</td>
        <td>18449</td>
   </tr>
</table>
</body> </html>
```

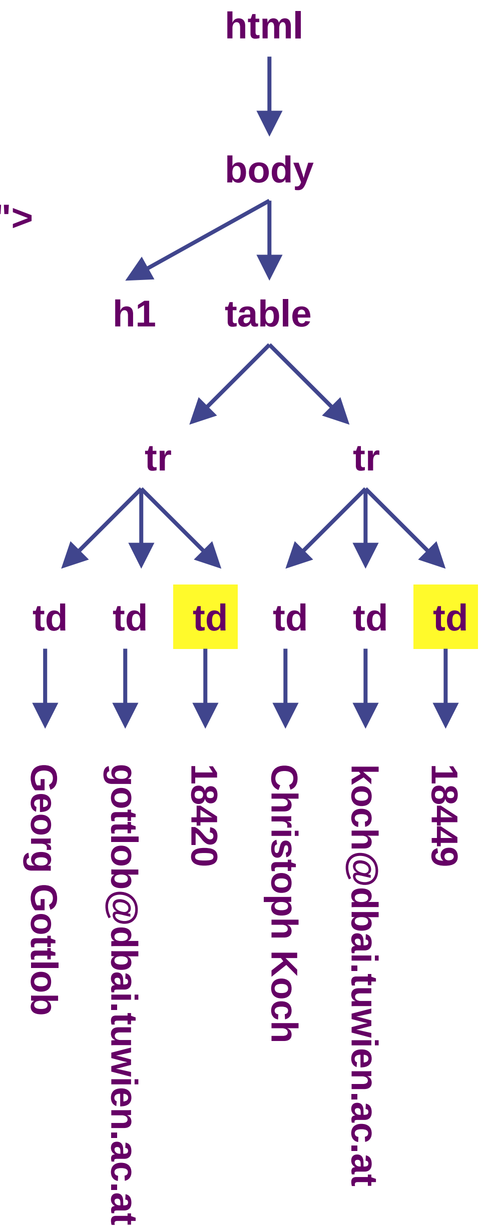


People @ DBAI

Georg Gottlob	gottlob@...	18420
Christoph Koch	koch@...	18449

Predicate *phone*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html> <body>
<h1>People @ DBAI</h1>
<table border="1" cellpadding="3" cellspacing="1">
   <tr> <td>Georg Gottlob</td>
        <td>gottlob@dbai.tuwien.ac.at</td>
        <td>18420</td>
   </tr>
   <tr> <td>Christoph Koch</td>
        <td>koch@dbai.tuwien.ac.at</td>
        <td>18449</td>
   </tr>
</table>
</body> </html>
```



People @ DBAI

Georg Gottlob	gottlob@...	18420
Christoph Koch	koch@...	18449

Desiderata of Tree Wrapping Languages

- Clear semantics.
- Good trade-off between complexity and expressiveness.
- Easy to use as wrapper programming language.
- Suitable for being incorporated into visual tools.

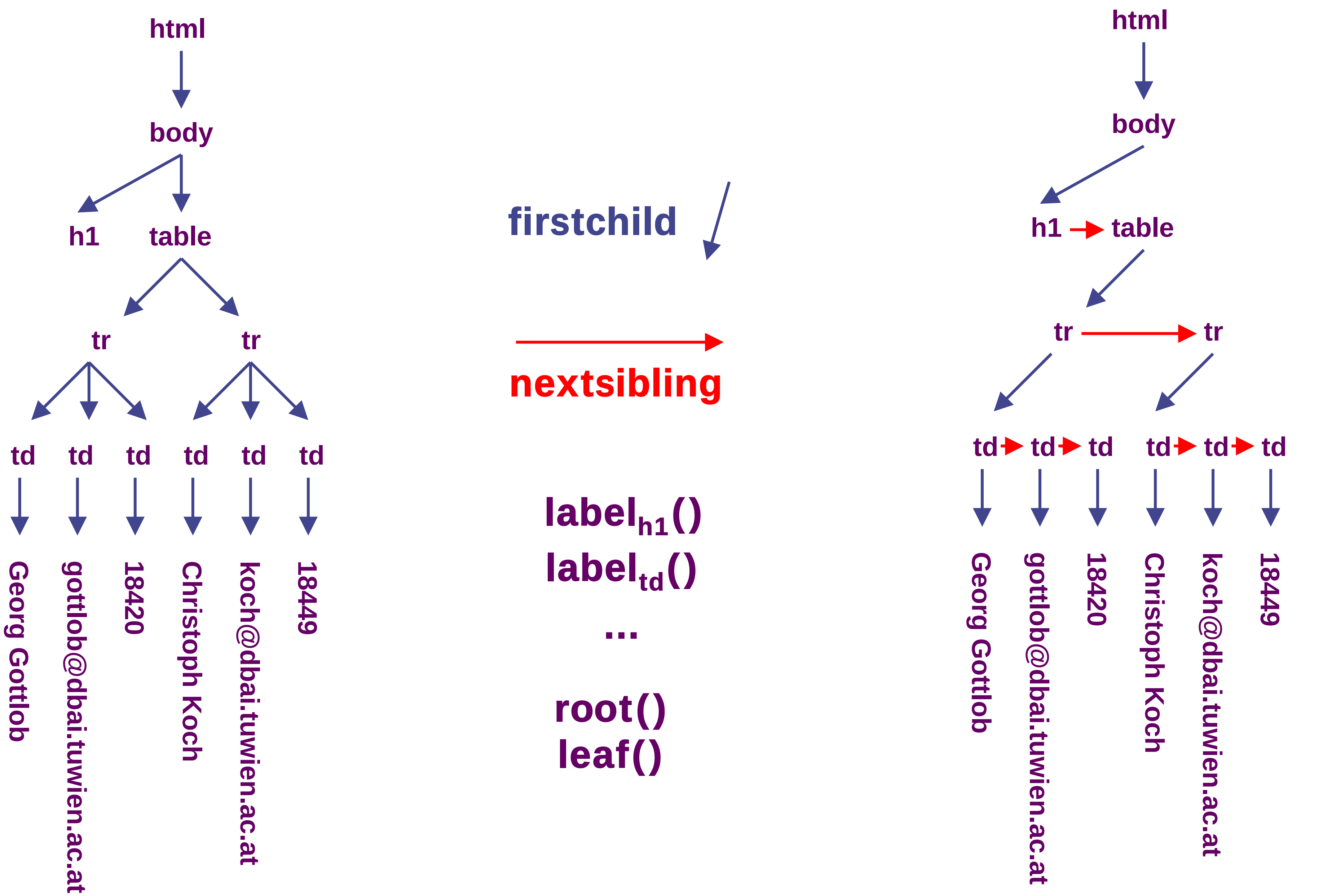
Expressiveness Yardstick: MSO

- MSO captures exactly the essence of data extraction:
  - ✓ Define sets of nodes of a document
  - ✓ Re-label tree nodes by monadic predicates
  - ✓ Declare some as irrelevant

- Expressiveness, complexity, semantics well understood:
  - ✓ MSO over trees: perfect logical semantics
  - ✓ MSO over trees: high expressive power (tree automata)
  - ✓ MSO queries over trees: low data complexity

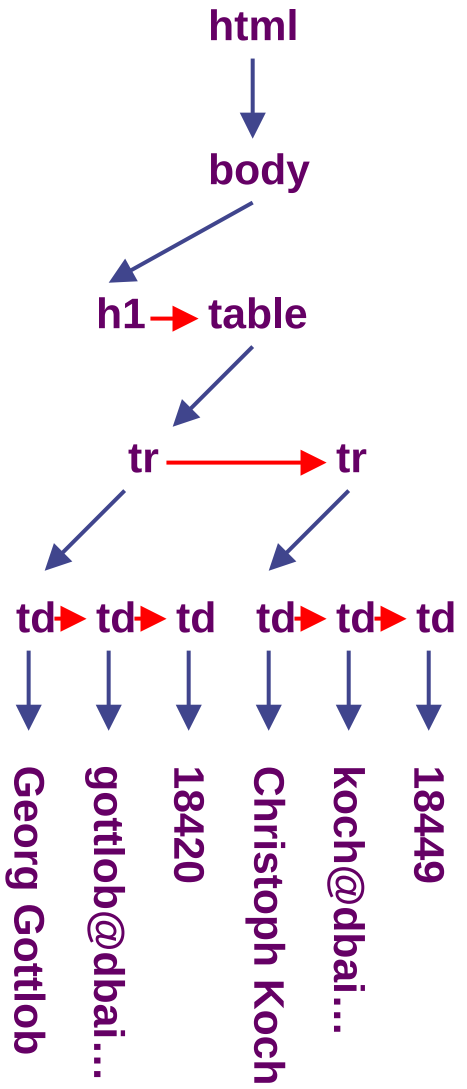
- Drawbacks: - hard to use, no visual specification,
  - high query complexity (bad scalability).

Ordered Trees as finite structures



MSO over Trees

Extract from a binary tree all roots of sub-trees with an odd number of leaves



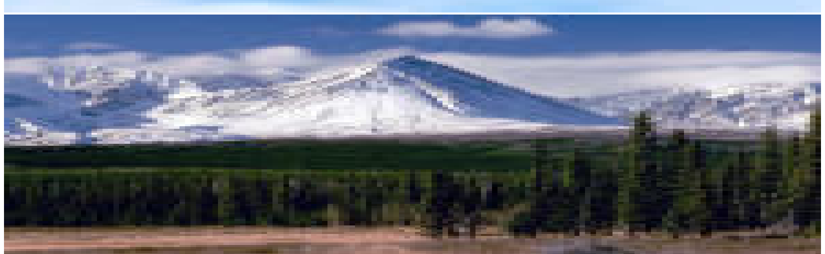
$$\exists S\ \forall x\ [\ S(u)\ \ \&\ \ (\ \mathrm{leaf}(x) \rightarrow S(x))\ \ \&$$
$$\forall\ x,y,z\ (((\mathrm{firstchild}(x,y)\ \&\ \mathrm{nextsibling}(y,z)) \rightarrow$$
$$(S(x) \leftrightarrow \neg(S(y) \leftrightarrow S(z))))]$$

Logic

Database theory

DB programming

Application design



Logic

Database theory

DB programming

Application design

MSO

Logic

MSO

Database theory

Monadic Datalog

DB programming

Application design

Logic

Database theory

DB programming

Application design

MSO

||

Monadic Datalog

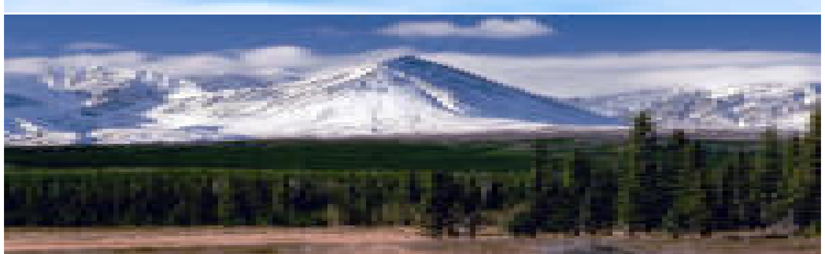
Logic	MSO
	||
Database theory	Monadic Datalog
DB programming	Elog
Application design	

Logic

Database theory

DB programming

Application design



MSO

$=$

Monadic Datalog

$\supseteq$

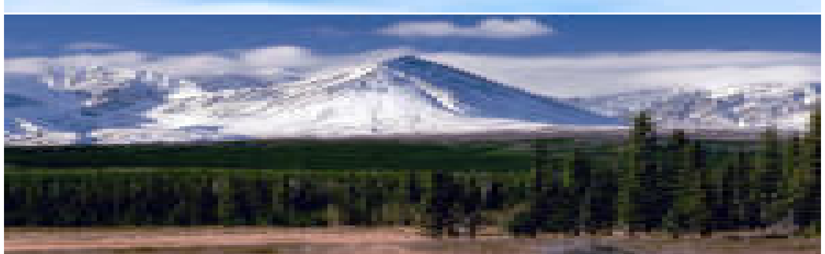
Elog

Logic

Database theory

DB programming

Application design



MSO

$=$

Monadic Datalog

$\subseteq$

Elog

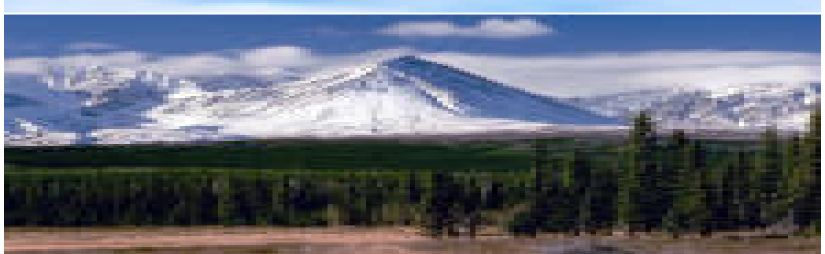
Lixto Visual Wrapper

Logic

Database theory

DB programming

Application design



MSO

$=$

Monadic Datalog

$\supseteq$

Elog

$\supseteq$

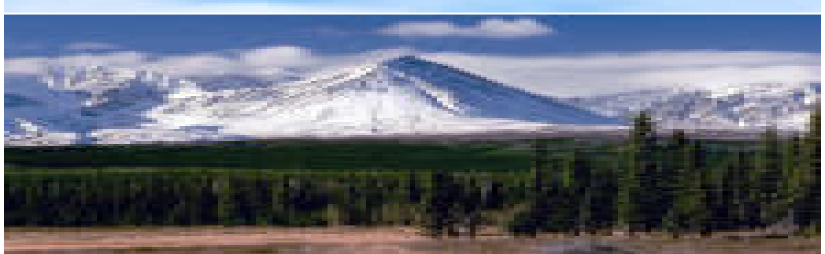
Lixto Visual Wrapper

Logic

Database theory

DB programming

Application design



MSO

$=$

Monadic Datalog

$\subseteq$

Elog

$\subseteq$

Lixto Visual Wrapper

$\subseteq$

lixto Suite

Monadic Datalog as a Wrapping Language

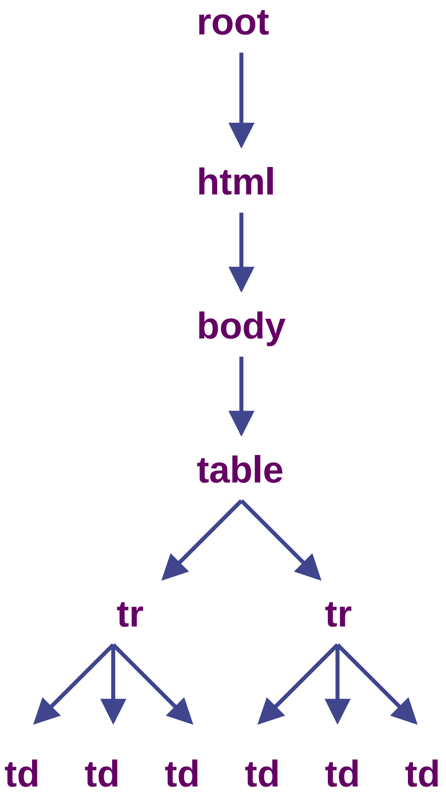
entry(X) :- root(R), firstchild(R,U), label[html](U),
firstchild(U,V), label[body](V),
firstchild(V,W),label[table](W),
firstchild(W,X), label[tr](X).

entry(X):- entry(Y), nextsibling(Y,X).

name(X) :- entry(E), firstchild(E, X), label[td](X).

email(X) :- name(N), nextsibling(N, X), label[td](X).

phone(X) :- email(M), nextsibling(M, X), label[td](X).



Monadic Datalog as a Wrapping Language

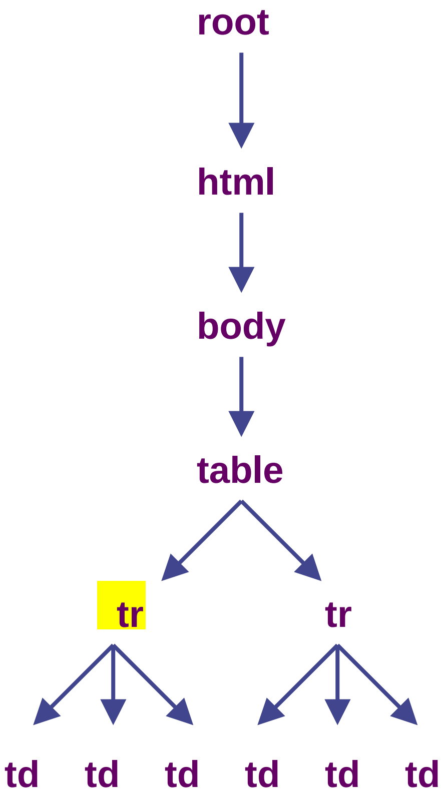
entry(X) :- root(R), firstchild(R,U), label[html](U),
firstchild(U,V), label[body](V),
firstchild(V,W),label[table](W),
firstchild(W,X), label[tr](X).

entry(X):- entry(Y), nextsibling(Y,X).

name(X) :- entry(E), firstchild(E, X), label[td](X).

email(X) :- name(N), nextsibling(N, X), label[td](X).

phone(X) :- email(M), nextsibling(M, X), label[td](X).



Monadic Datalog as a Wrapping Language

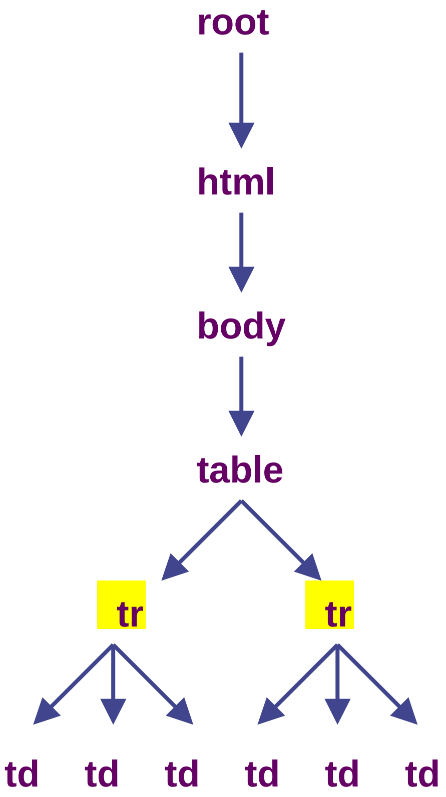
entry(X) :- root(R), firstchild(R,U), label[html](U),
firstchild(U,V), label[body](V),
firstchild(V,W),label[table](W),
firstchild(W,X), label[tr](X).

entry(X):- entry(Y), nextsibling(Y,X).

name(X) :- entry(E), firstchild(E, X), label[td](X).

email(X) :- name(N), nextsibling(N, X), label[td](X).

phone(X) :- email(M), nextsibling(M, X), label[td](X).



Monadic Datalog as a Wrapping Language

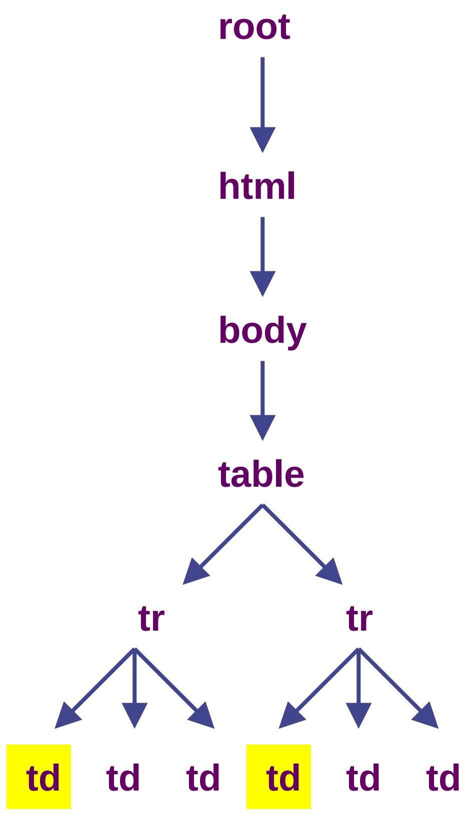
entry(X) :- root(R), firstchild(R,U), label[html](U),
firstchild(U,V), label[body](V),
firstchild(V,W),label[table](W),
firstchild(W,X), label[tr](X).

entry(X):- entry(Y), nextsibling(Y,X).

name(X) :- entry(E), firstchild(E, X), label[td](X).

email(X) :- name(N), nextsibling(N, X), label[td](X).

phone(X) :- email(M), nextsibling(M, X), label[td](X).



Monadic Datalog as a Wrapping Language

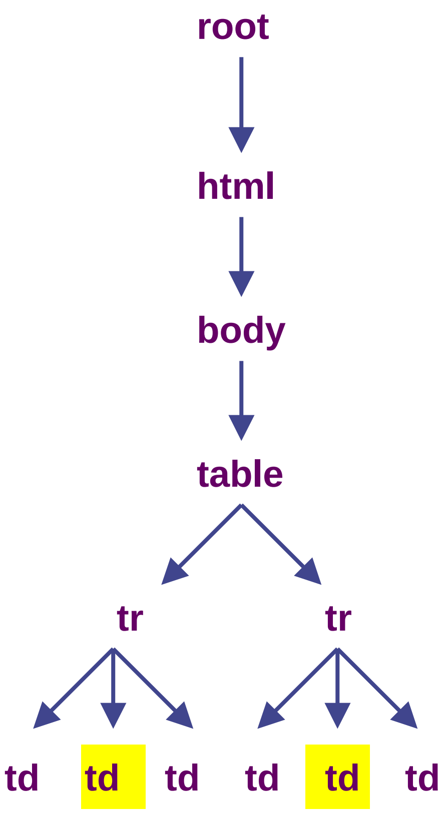
entry(X) :- root(R), firstchild(R,U), label[html](U),
firstchild(U,V), label[body](V),
firstchild(V,W),label[table](W),
firstchild(W,X), label[tr](X).

entry(X):- entry(Y), nextsibling(Y,X).

name(X) :- entry(E), firstchild(E, X), label[td](X).

email(X) :- name(N), nextsibling(N, X), label[td](X).

phone(X) :- email(M), nextsibling(M, X), label[td](X).



Monadic Datalog as a Wrapping Language

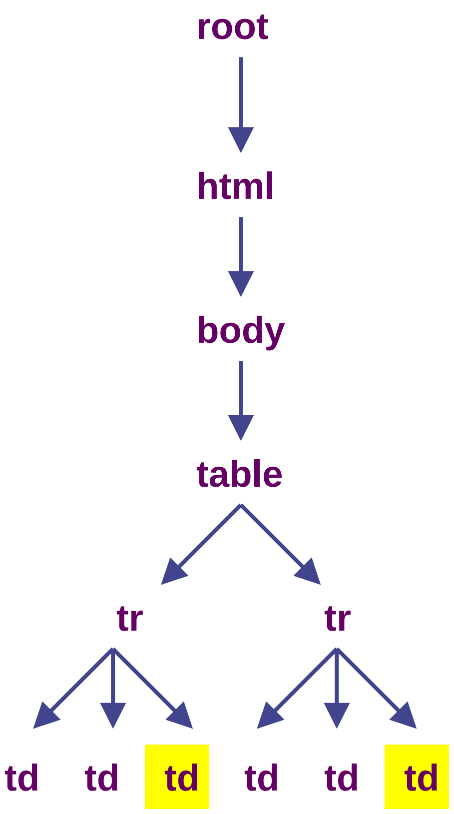
entry(X) :- root(R), firstchild(R,U), label[html](U),
firstchild(U,V), label[body](V),
firstchild(V,W),label[table](W),
firstchild(W,X), label[tr](X).

entry(X):- entry(Y), nextsibling(Y,X).

name(X) :- entry(E), firstchild(E, X), label[td](X).

email(X) :- name(N), nextsibling(N, X), label[td](X).

phone(X) :- email(M), nextsibling(M, X), label[td](X).



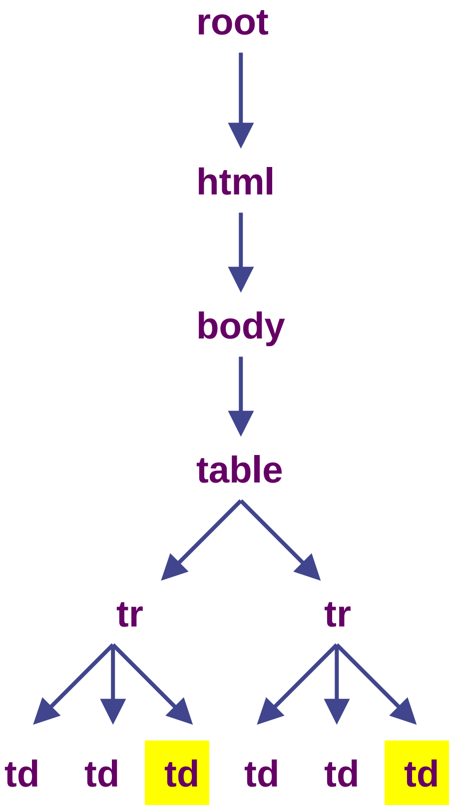
entry(X) :- root(R), firstchild(R,U), label[html](U),
firstchild(U,V), label[body](V),
firstchild(V,W),label[table](W),
firstchild(W,X), label[tr](X).

entry(X):- entry(Y), nextsibling(Y,X).

name(X) :- entry(E), firstchild(E, X), label[td](X).

email(X) :- name(N), nextsibling(N, X), label[td](X).

phone(X) :- email(M), nextsibling(M, X), label[td](X).




```
<?xml version="1.0"?>
<peopledb>
<entry> <name>Georg Gottlob</name>
        <email>gottlob@dbai.tuwien.ac.at</email>
        <phone>18420</phone>
</entry>
<entry> <name>Christoph Koch</name>
        <email>koch@dbai.tuwien.ac.at</email>
        <phone>18449</phone>
</entry>
</peopledb>
```

The ELOG Programming Language

Monadic Datalog enriched with several practical primitives for handling objects on Web pages:

- Expressing a tree-path at once
- Extracting data items from plain text
- Following URLs and extracting from several pages
- Keeping track of the pattern hierarchy
- Access to ontological databases

ELOG Program for eBay pages

$$\begin{aligned}
\mathtt{tableseq}(S, X) \leftarrow\ & \mathtt{document}(\text{“www.ebay.com/”}, S), \mathtt{subsq}(S, (.\mathtt{body}, []), (.\mathtt{table}, []), (.\mathtt{table}, []), X), \\
& \mathtt{before}(S, X, (.\mathtt{table}, [(\mathtt{elementtext}, \mathtt{item},]), 1, 1, _, _), \mathtt{after}(S, X, (.\mathtt{hr}, []), 1, 1, _, _)
\end{aligned}$$

$$\mathtt{record}(S, X) \leftarrow \mathtt{tableseq}(\_, S), \mathtt{subelem}(S, .\mathtt{table}, X)$$

$$\mathtt{itemnum}(S, X) \leftarrow \mathtt{record}(\_, S), \mathtt{subelem}(S, \star.\mathtt{td}, X), \mathtt{notbefore}(S, X, (.\mathtt{td}, []), \mathtt{maxint})$$

$$\mathtt{itemdes}(S, X) \leftarrow \mathtt{record}(\_, S), \mathtt{subelem}(S, (\star.\mathtt{td}.\star.\mathtt{content}, [(\mathtt{a}, , 0)], X)$$

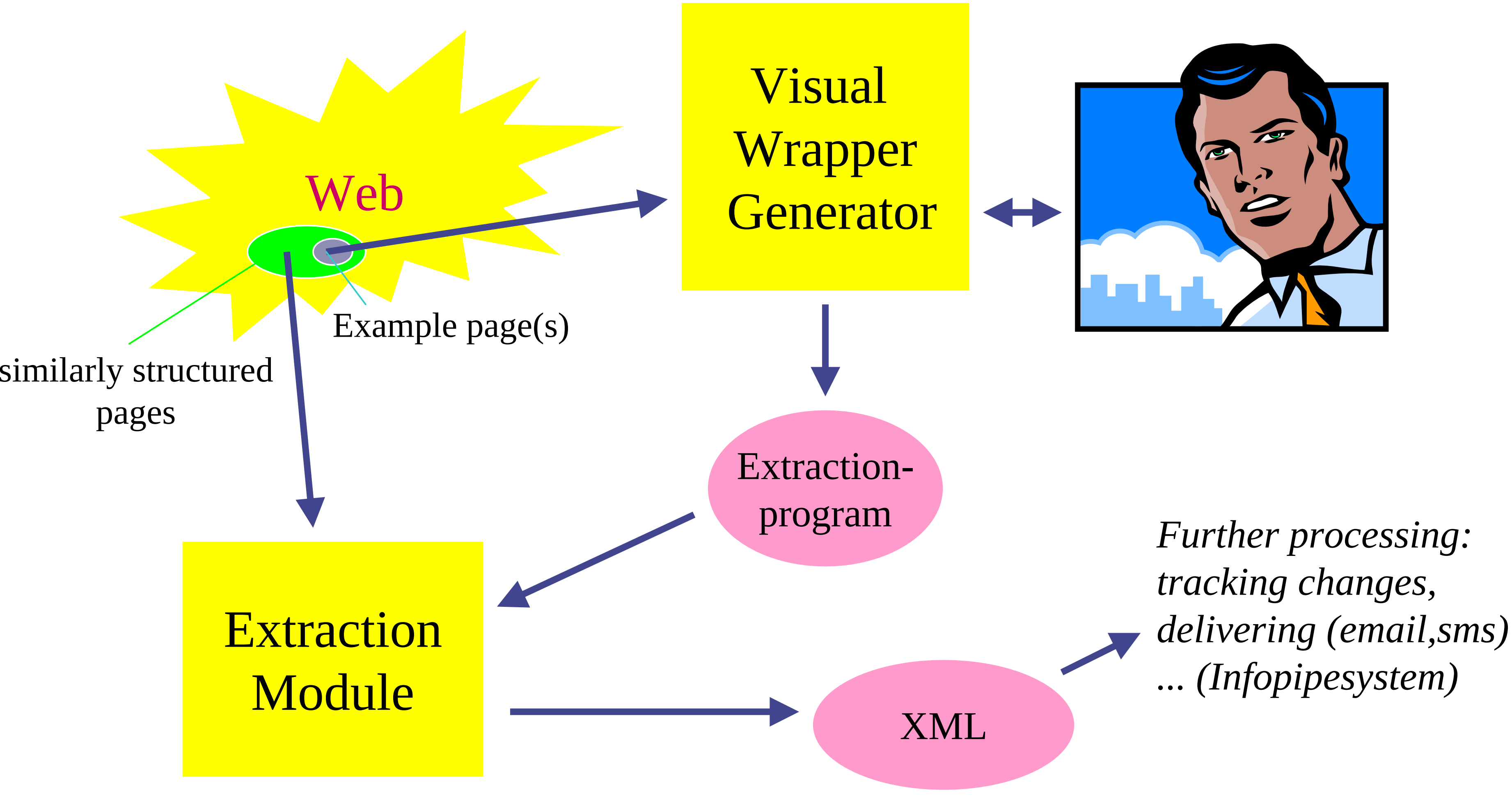
$$\mathtt{price}(S, X) \leftarrow \mathtt{record}(\_, S), \mathtt{subelem}(S, (\star.\mathtt{td}, [(\mathtt{elementtext}, Y, 1)]), X), \mathtt{valuta}(Y)$$

$$\mathtt{bids}(S, X) \leftarrow \mathtt{record}(\_, S), \mathtt{subelem}(S, \star.\mathtt{td}, X), \mathtt{before}(S, X, (.\mathtt{td}, []), 1, 30, Y, \_), \mathtt{price}(S, Y)$$

$$\mathtt{currency}(S, X) \leftarrow \mathtt{price}(\_, S), \mathtt{subtext}(S, Y, X), \mathtt{valuta}(Y)$$

$$\mathtt{pricewc}(S, X) \leftarrow \mathtt{price}(\_, S), \mathtt{subtext}(S, [0-9]^{+}, X)$$

The LiXto Visual Wrapper



SHORT DEMO

Paris Restaurants

Some theoretical results…

Monadic Datalog:
Expressive Power and Complexity

Theorems [G. & Koch LICS 2002, JACM 2004]:

Monadic Datalog = MSO (over trees)

A data extraction task is definable in MSO if and only if it is feasible via a monadic datalog program.

Monadic Datalog has complexity:

$O(|data| * |program|)$

Good scaling behavior!

Expressive power of *LiXto*

Elog$^-$: Monadic kernel of Elog

Theorems [G., Koch PODS'02, SOFSEM'06]

ELOG$^-$ expresses monadic datalog

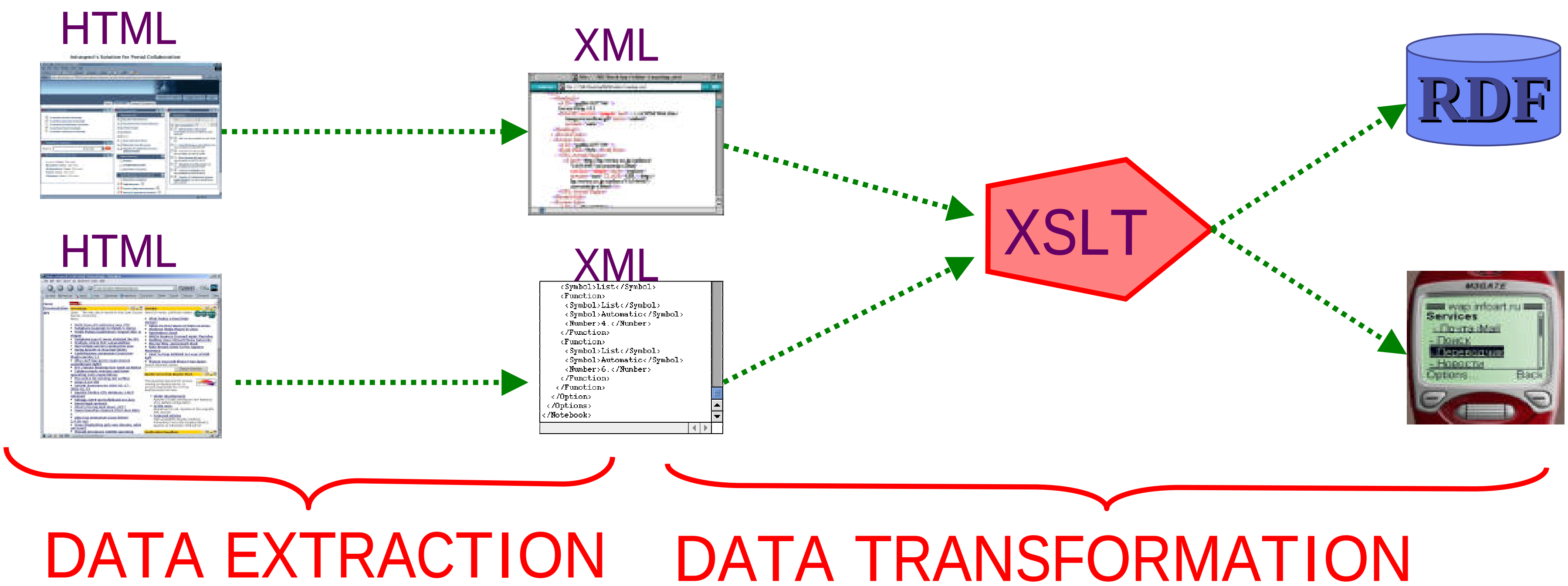
All of ELOG$^-$ is graphically programmable via ***LiXto***

Corollary:

LiXto expresses all MSO wrapping tasks.

The LiXto Suite...

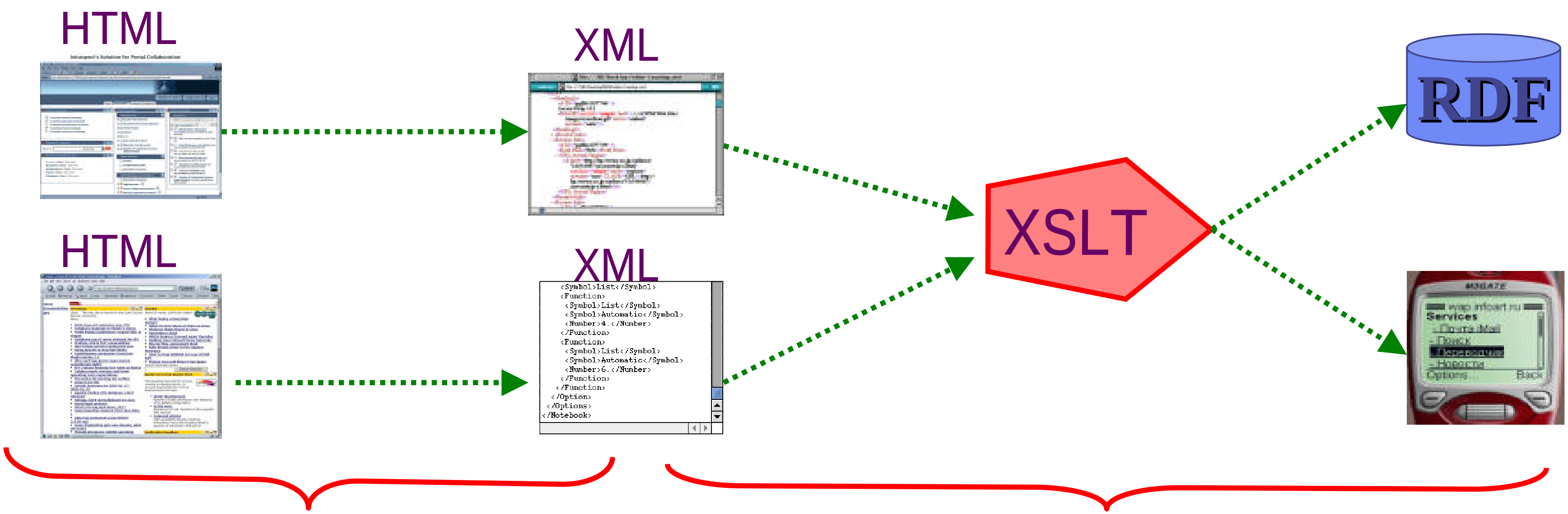
Extraction vs. Querying and Transforming



selection
annotation
hierarchical (re)grouping

combination (joins,etc.)
querying (XSLT, XQUERY)
personalization

Extraction vs. Querying and Transforming



DATA EXTRACTION

DATA TRANSFORMATION

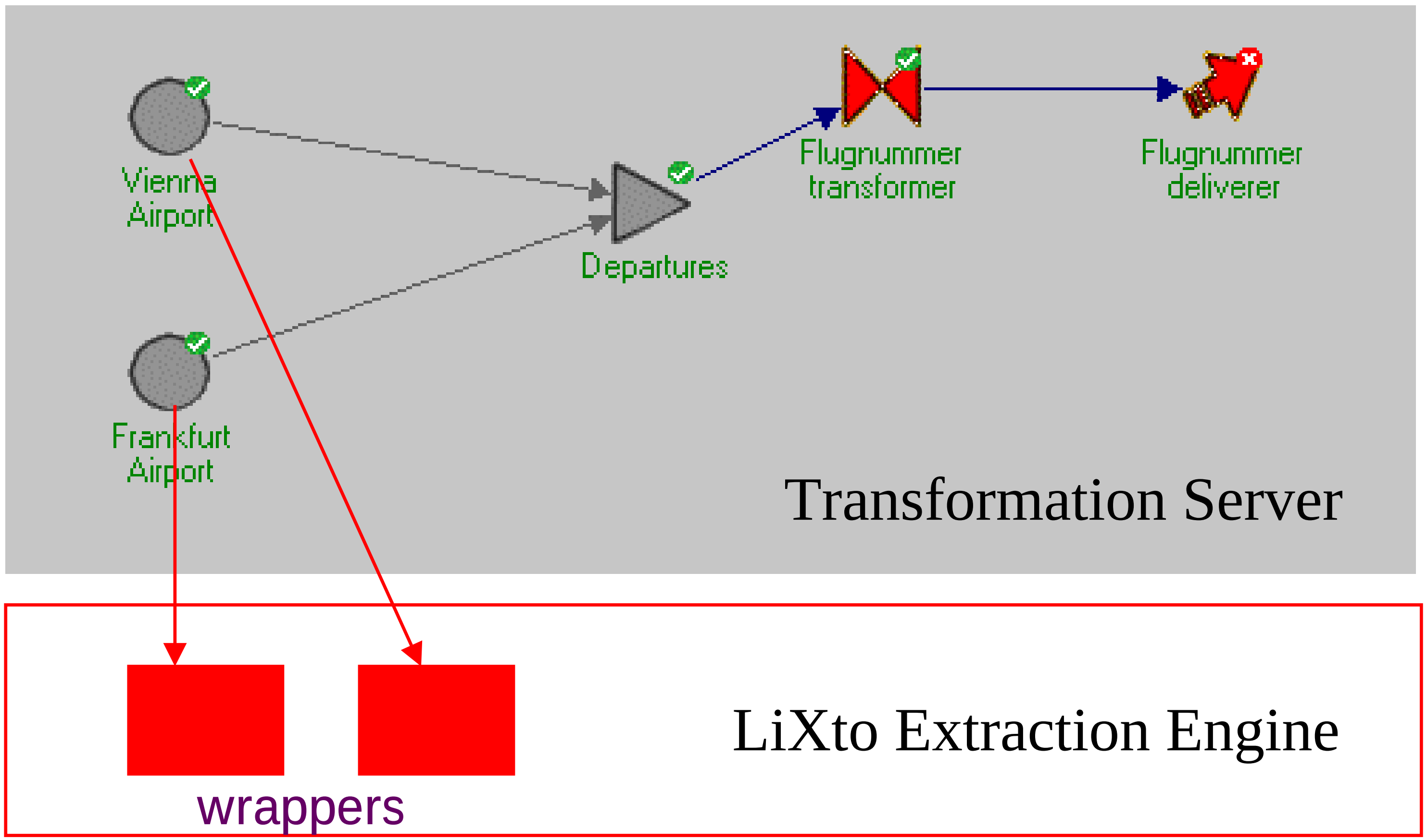
selection
annotation
hierarchical (re)grouping

combination (joins,etc.)
querying (XSLT, XQUERY)
personalization

lixto Visual Wrapper

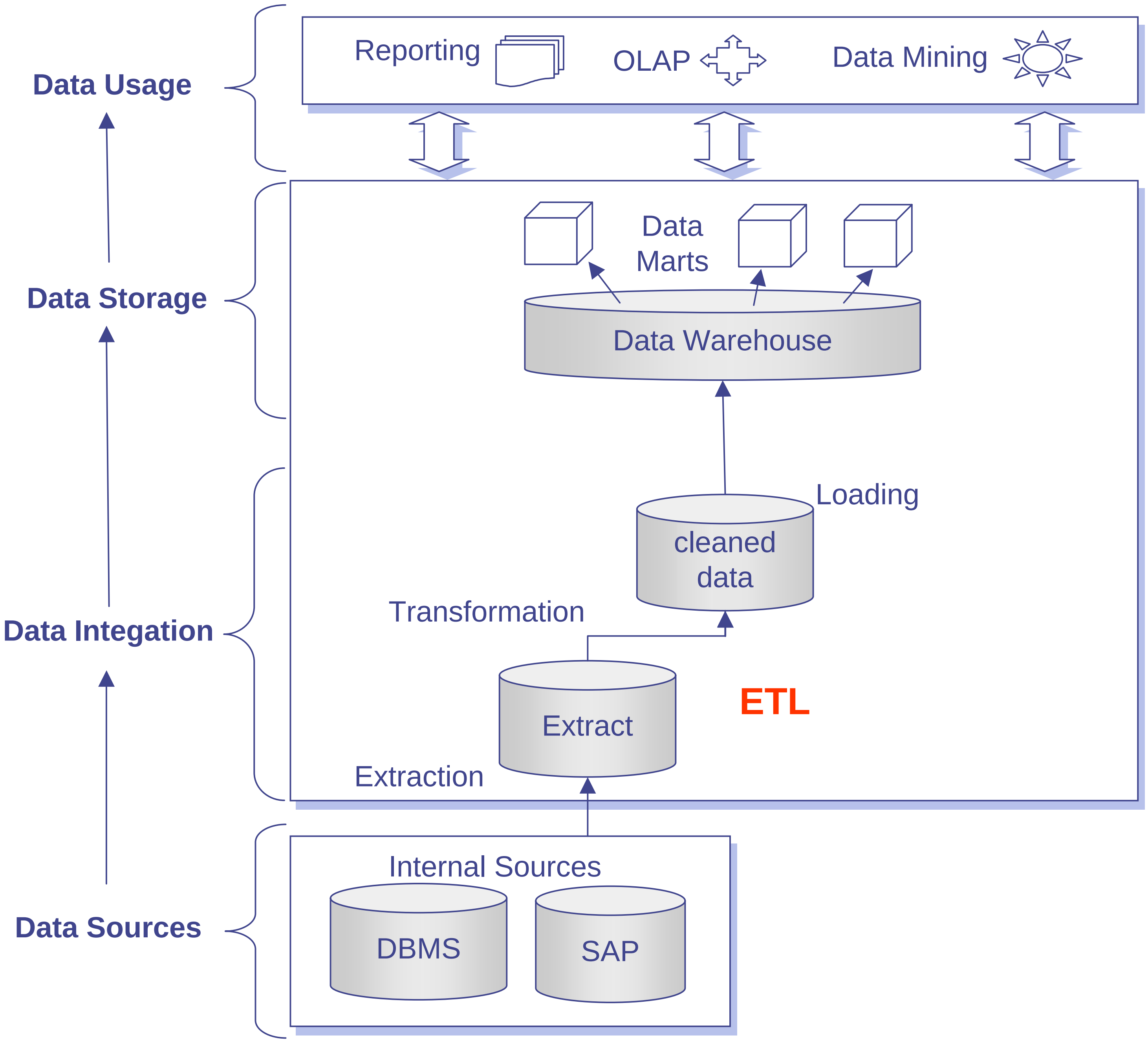
lixto Transformation Server

Product Architecture

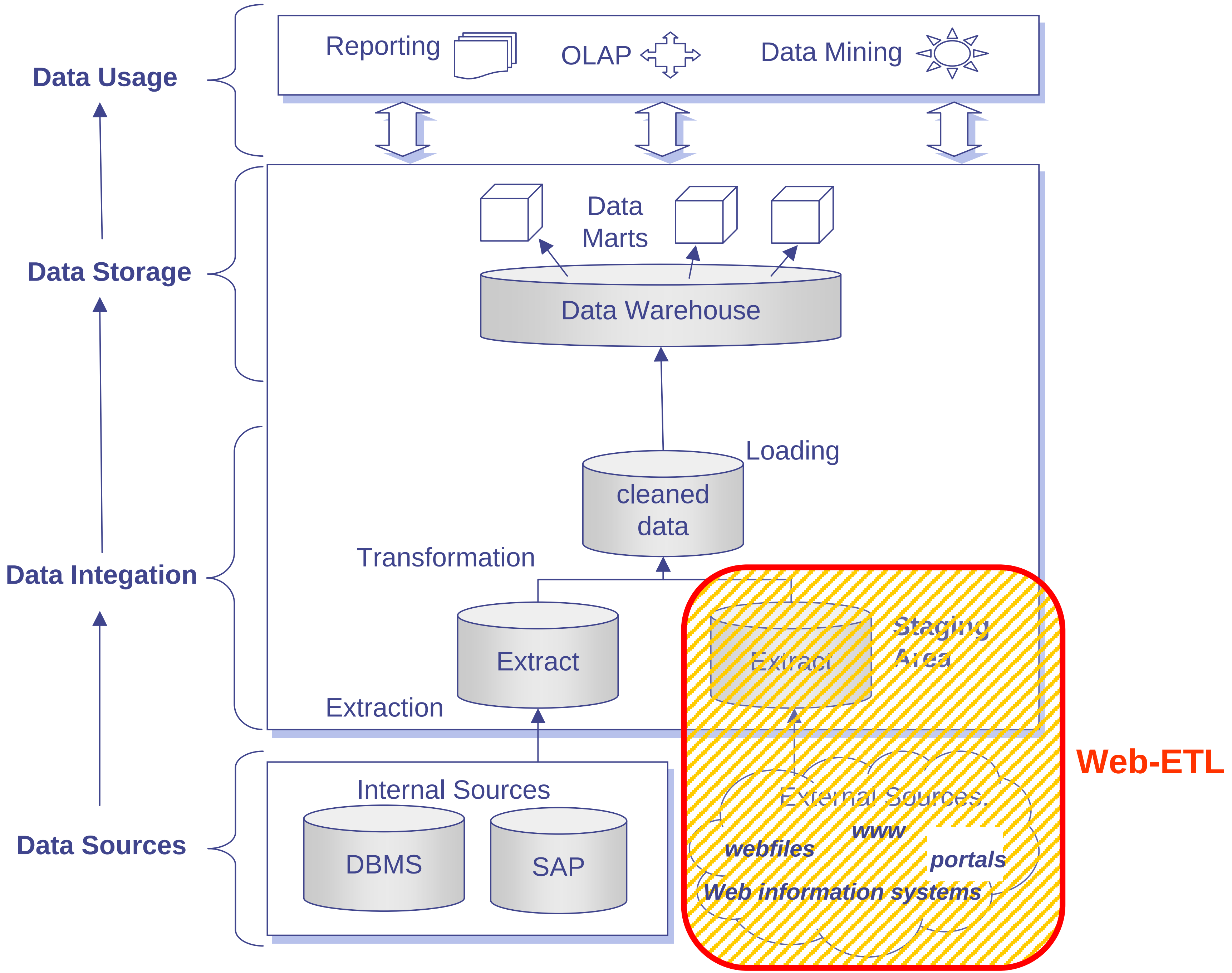


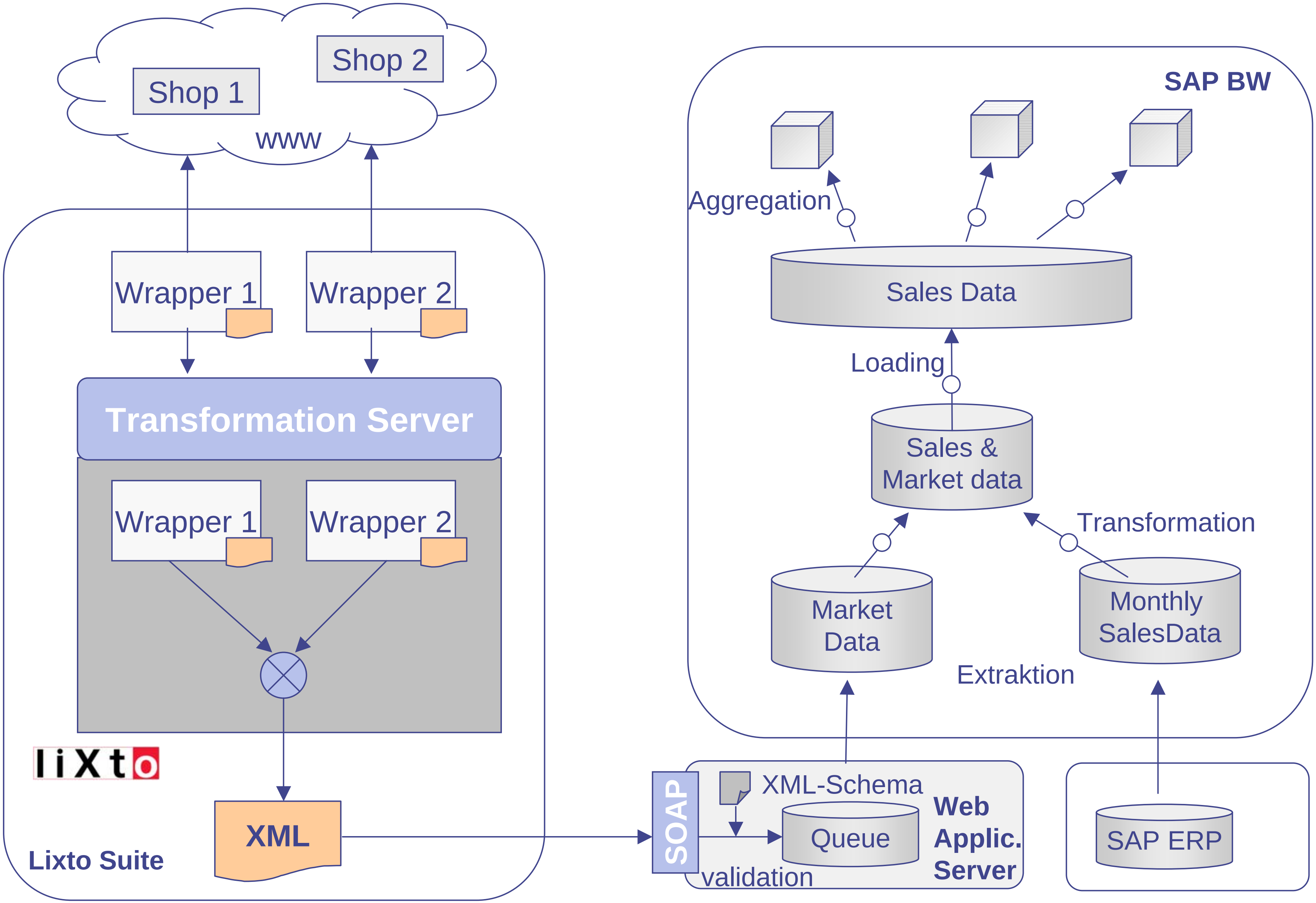
WEB ETL

FOR DATA WAREHOUSING

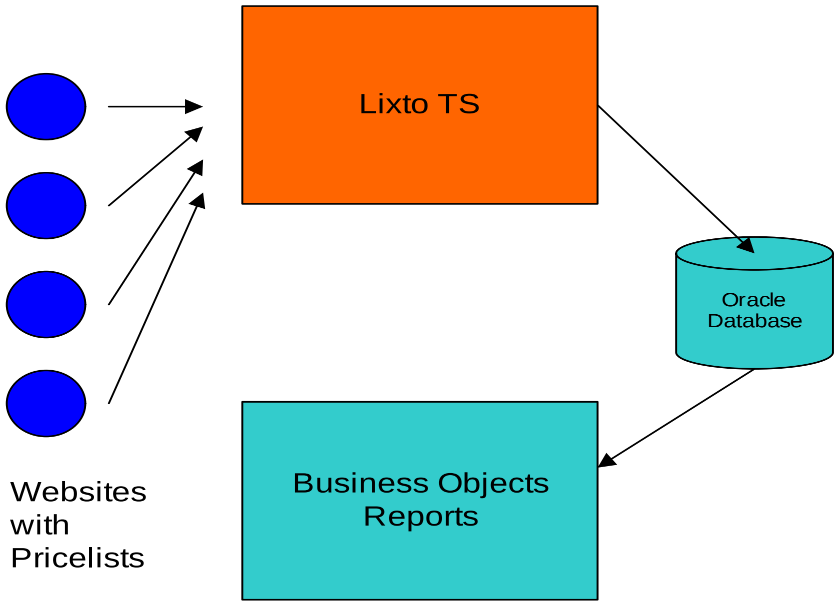


Data Usage
Reporting
OLAP
Data Mining
Data Storage
Data
Marts
Data Warehouse
Data Integration
Loading
cleaned
data
Transformation
Extract
ETL
Extraction
Data Sources
Internal Sources
DBMS
SAP


Data Usage
Reporting
OLAP
Data Mining
Data Storage
Data Marts
Data Warehouse
Data Integration
Loading
cleaned data
Transformation
Extract
Staging Area
Extract
Extraction
Data Sources
Internal Sources
DBMS
SAP
External Sources:
www
webfiles
portals
Web information systems
Web-ETL


Shop 1
Shop 2
www
Wrapper 1
Wrapper 2
Transformation Server
Wrapper 1
Wrapper 2
lixto
XML
Lixto Suite
SAP BW
Aggregation
Sales Data
Loading
Sales &
Market data
Transformation
Market
Data
Monthly
SalesData
Extraktion
SOAP
XML-Schema
Queue
validation
Web
Applic.
Server
SAP ERP

Architecture Overview



Involved Components:

- Lixto TS
- BI Tool
- Oracle 9

B2B Process Integration

Goal:

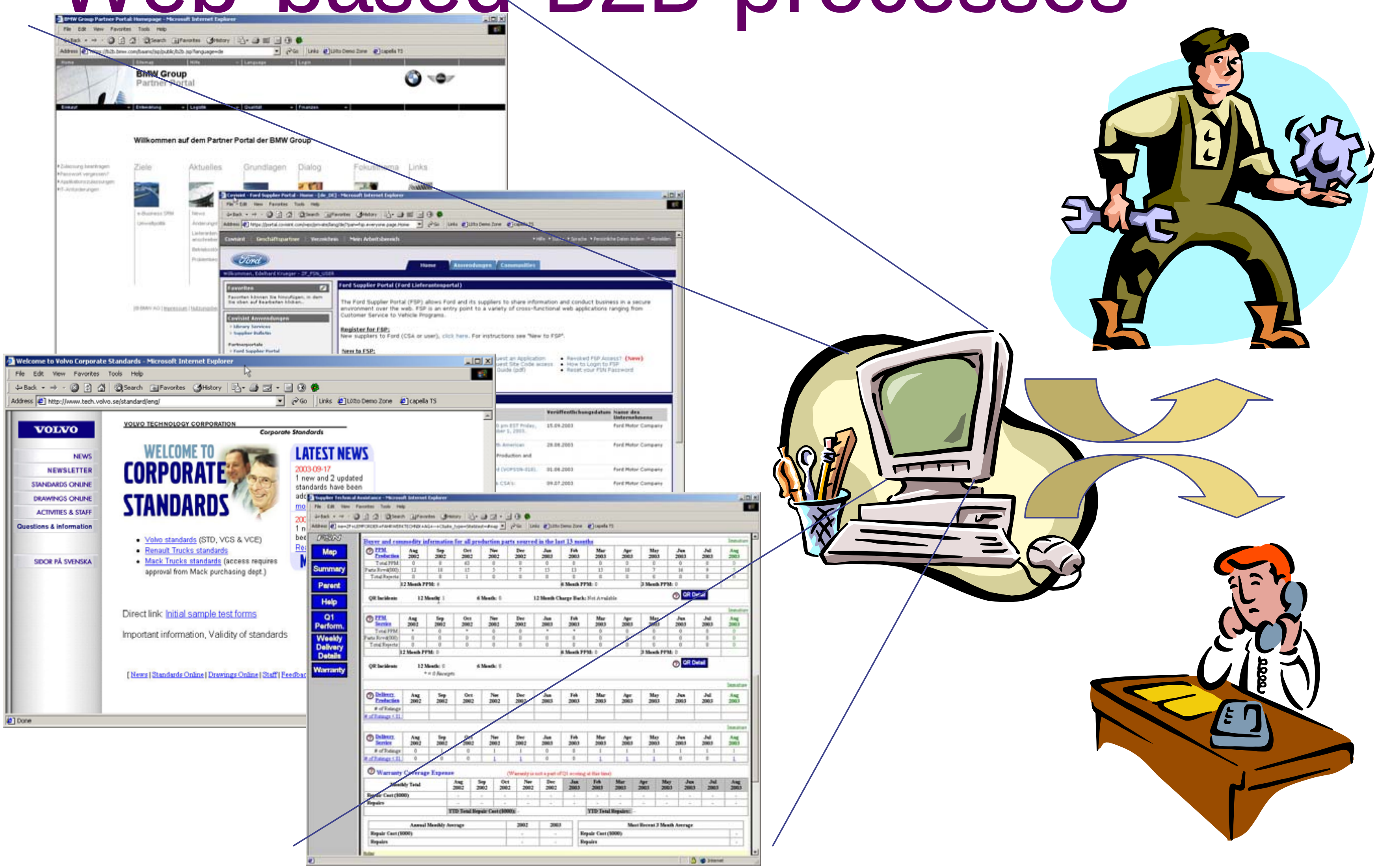
Automate portal-based quality management tasks

Numbers:

- 9 major automotive OEM portals
- ~50 supply plants

Project duration: ongoing

Web-based B2B processes



Process

OEM

Fiat

Ford

GM

BMW

Automatic Extraction

lixto

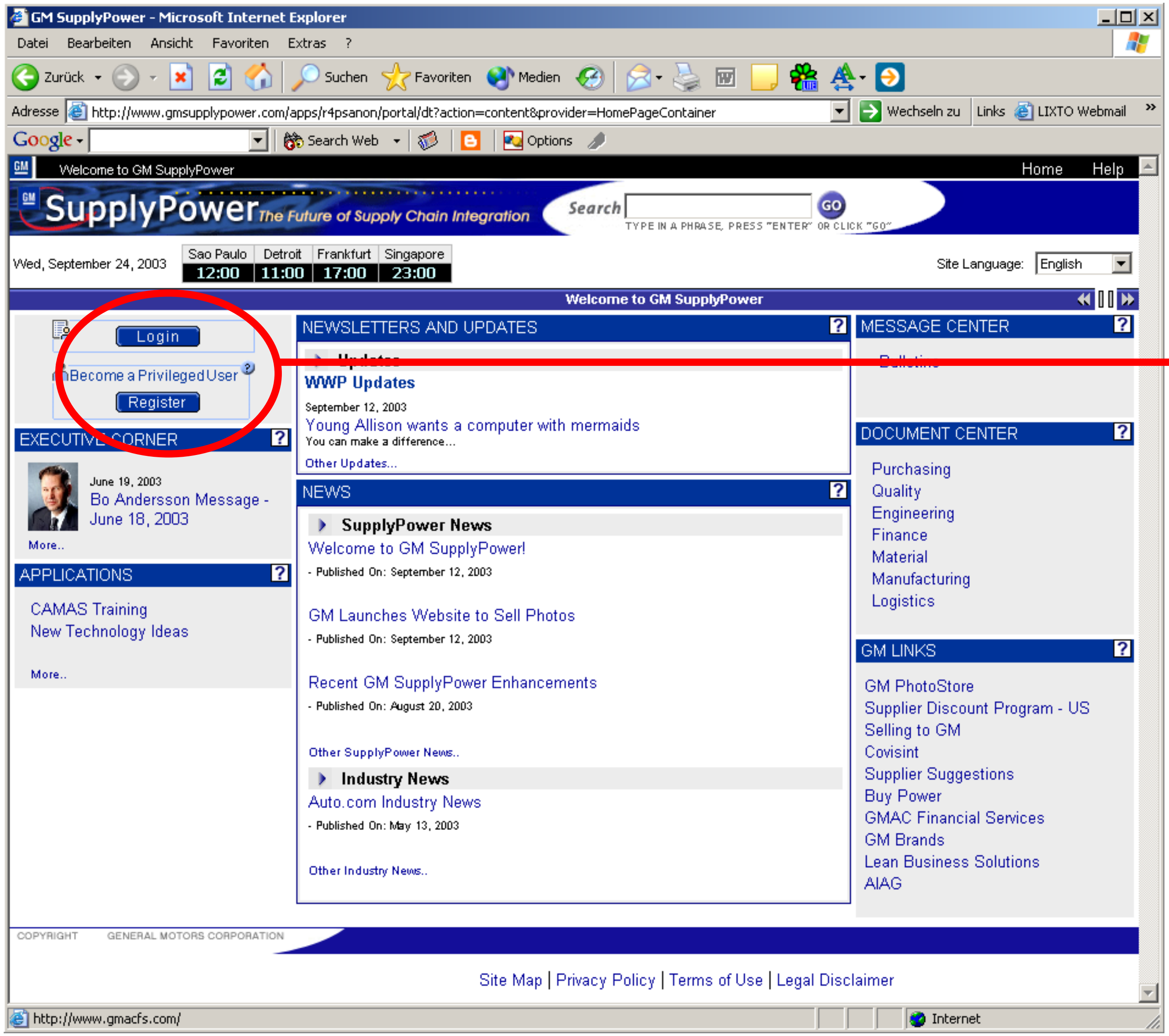
Email

Intranet

SAP Enterprise Portal

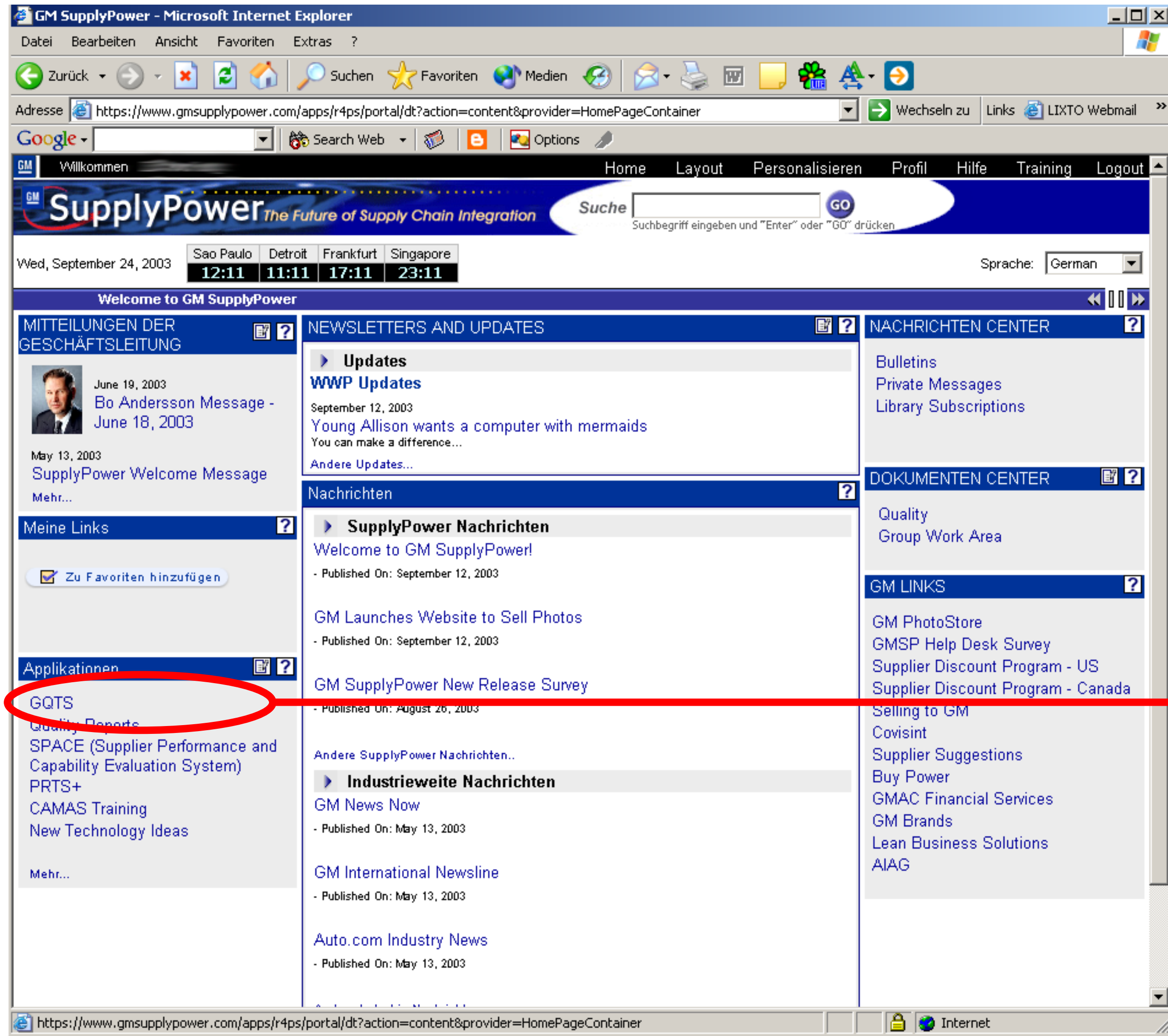
SAP R/3 QM

Automate portal applications



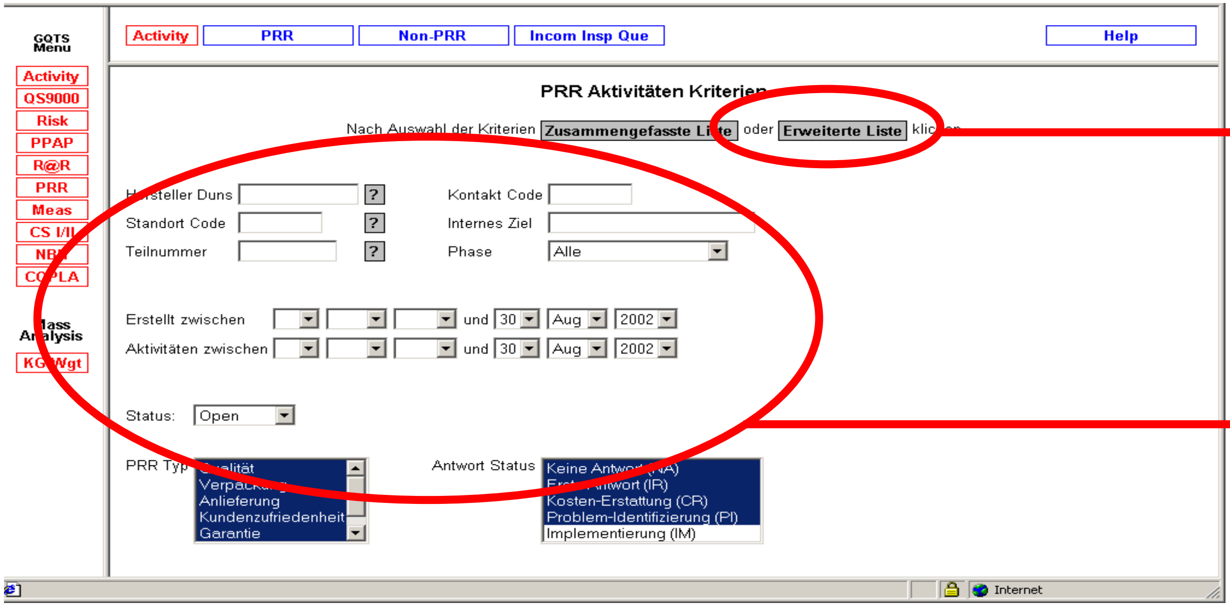
1. Login

Automate portal applications



2. Navigation

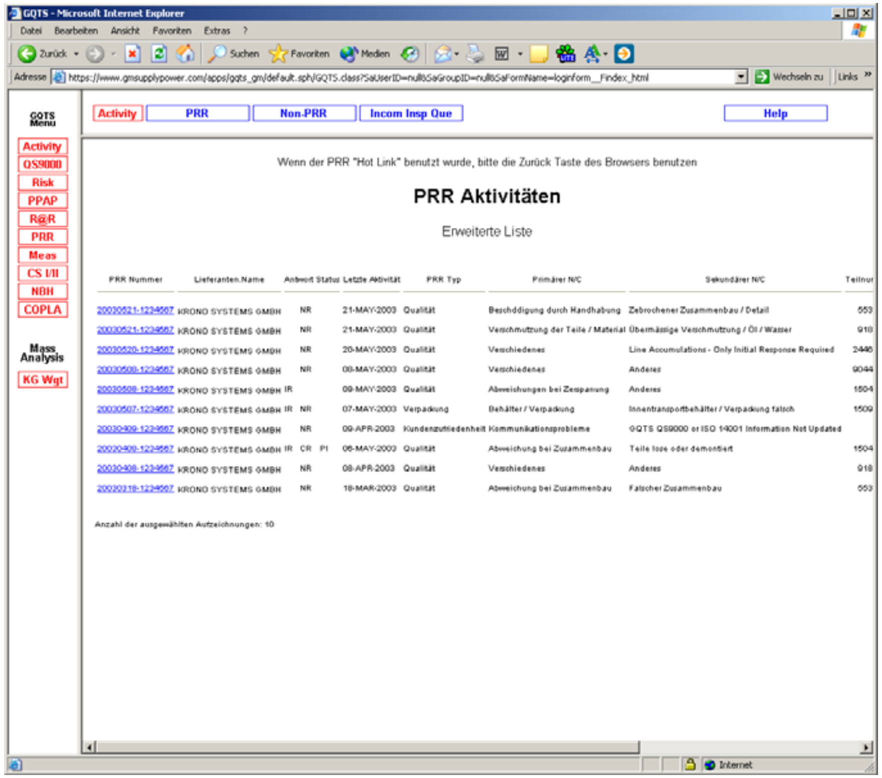
Automate portal applications



4. Click

3. Fill Forms

Automate portal applications

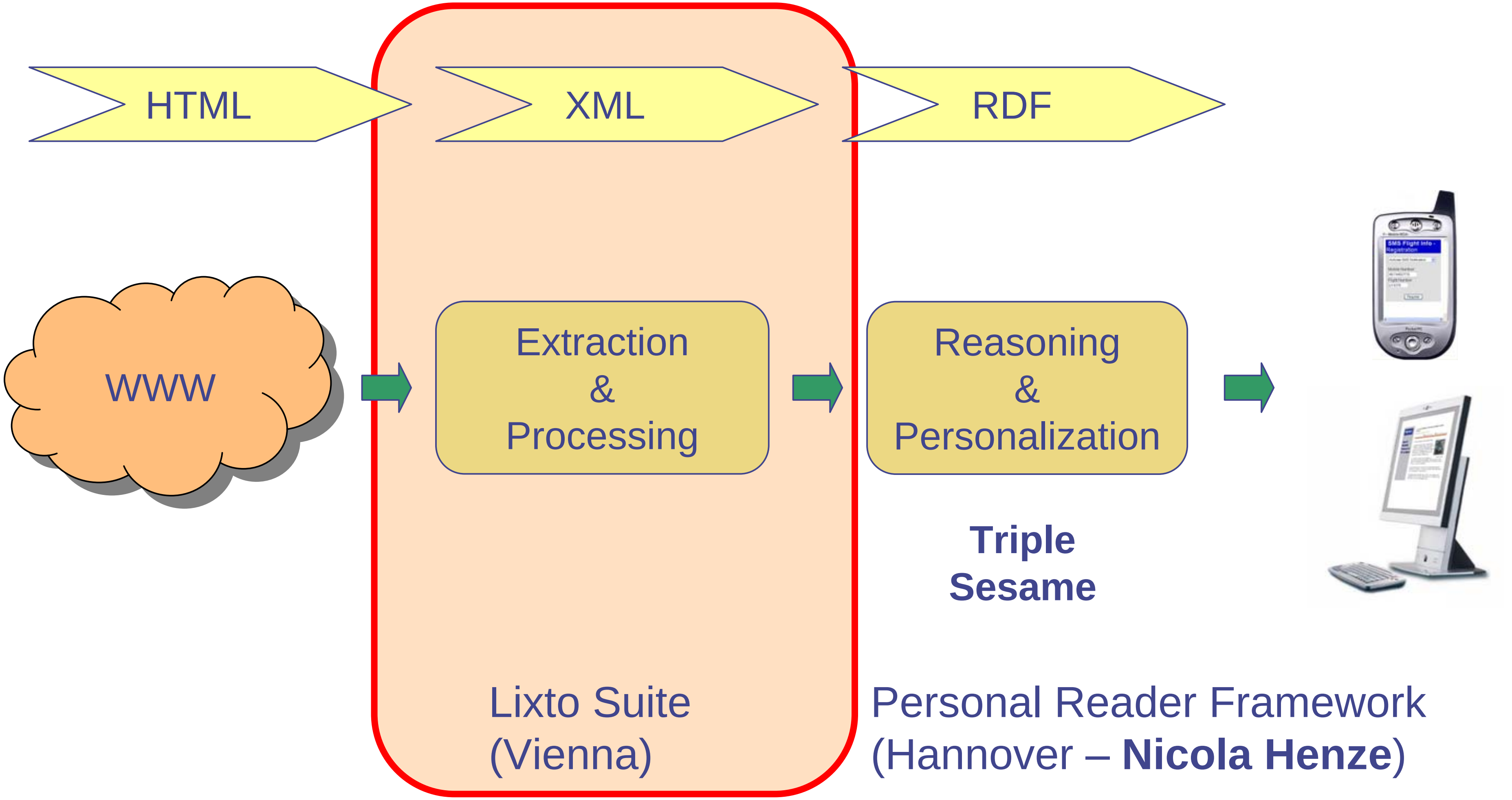


<?xml version="1.0" encoding="iso-8859-1"?>
<PRRReport>
<PRRNumber>20030521-1234567</PRRNumber>
<SupplierName>KRONO SYSTEMS GMBH</SupplierName>
<Status>NR</Status>
<LastActivity>21-MAY-2003</LastActivity>
…

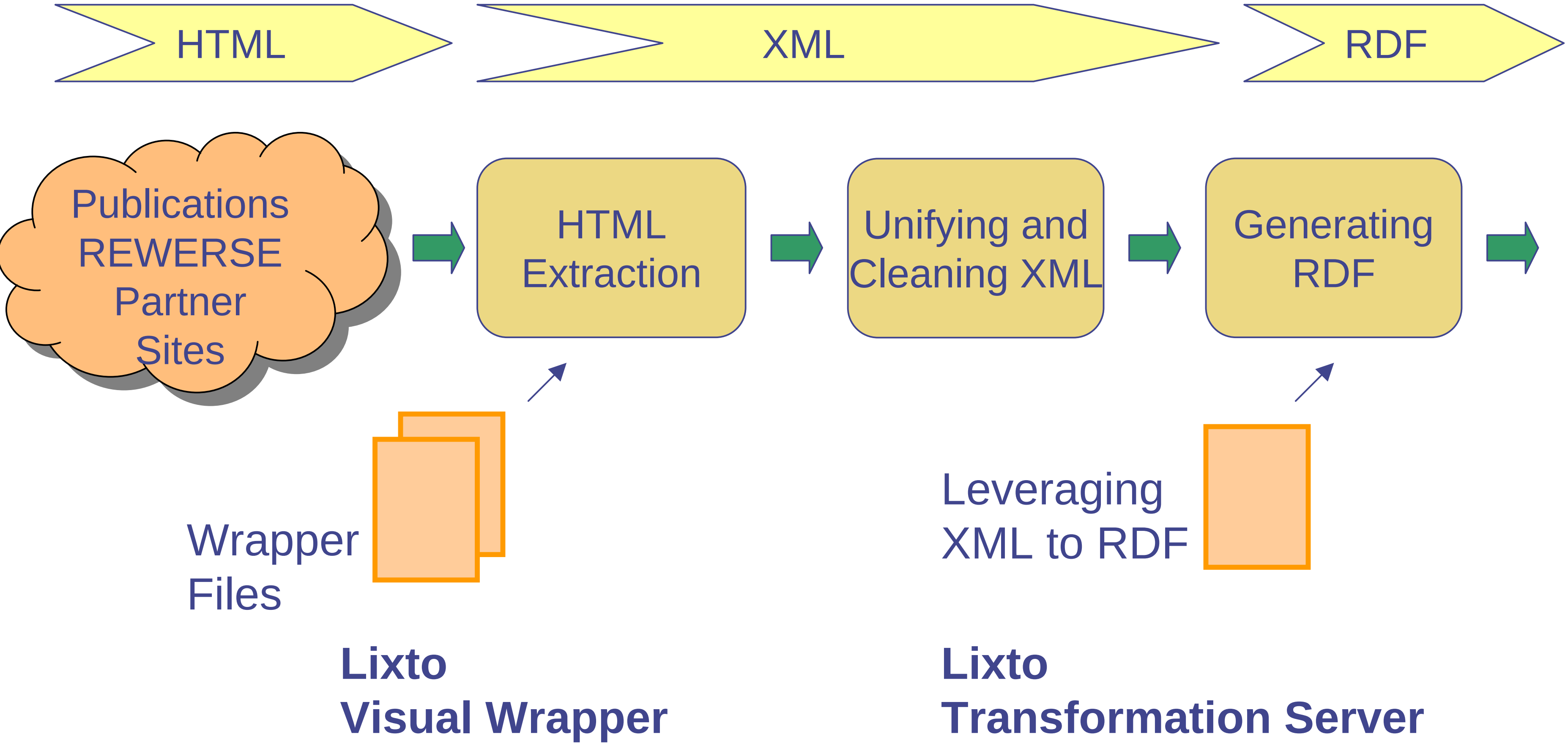
Creating RDF repositories...

Personal Publication Reader

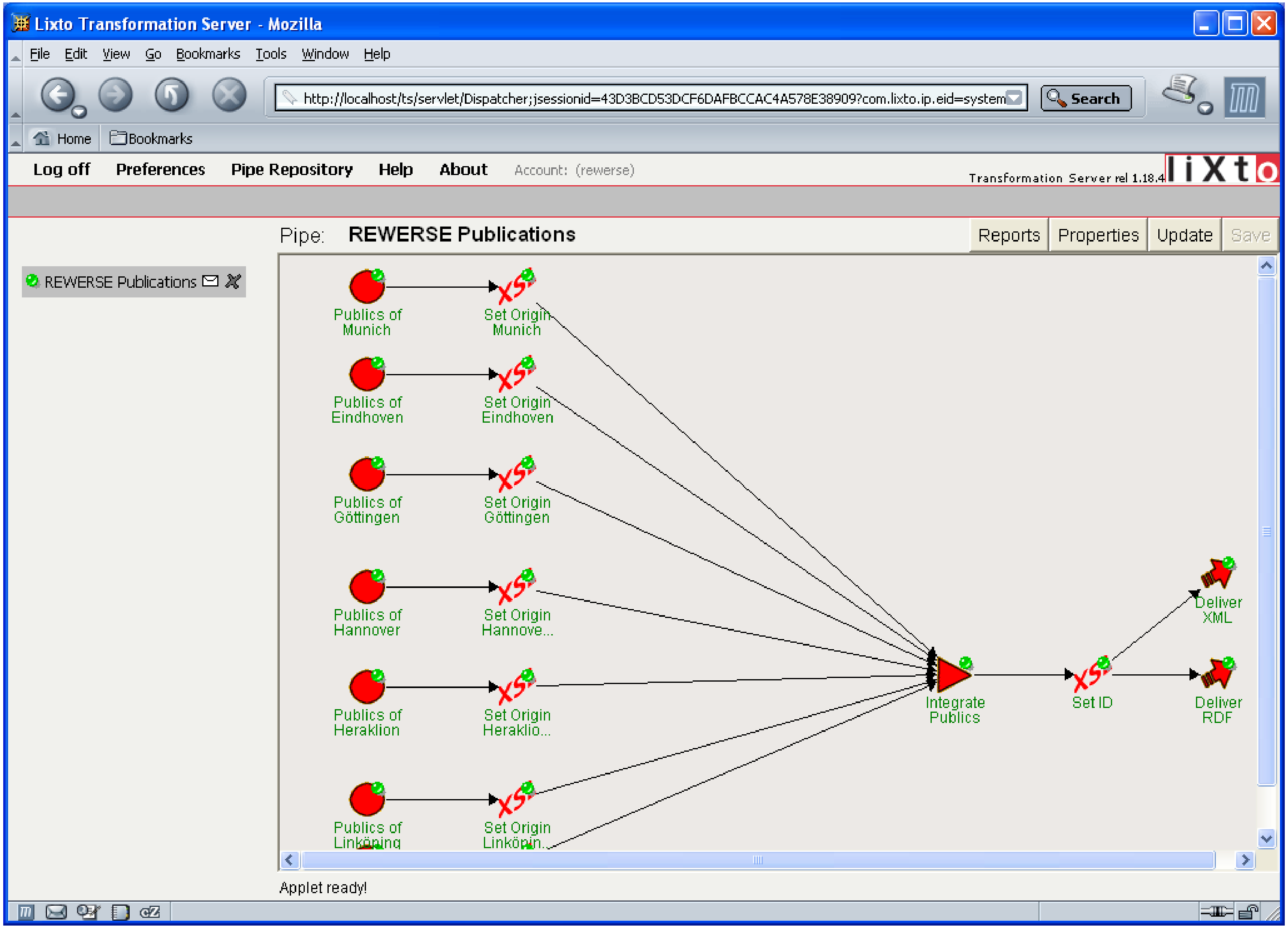
- Generating a publication ontology



PPR Example: Extraction & Processing



Lixto Transformation Server PPR Screenshot

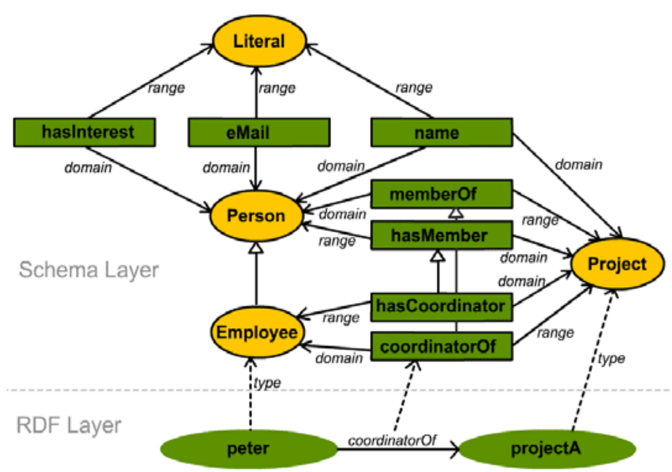


Rewerse Publication Ontology

```
<rdf:Description
  rdf:about="http://www.pms.informatik.uni-muenchen.de/
  publikationen/PMS-FB/PMS-FB-2004-12.pdf">
 <dc:publisher>University of Munich</dc:publisher>
 <dc:title>Visual Exploration and Retrieval of XML Document
  Collections with the Generic System X2</dc:title>
 <dc:creator>
   <rdf:Seq>
       <rdf:li rdf:resource="#Holger Meuss"/>
       <rdf:li rdf:resource="#Klaus U. Schulz"/>
       <rdf:li rdf:resource="#Felix Weigel"/>
       <rdf:li rdf:resource="#Simone Leonardi"/>
       <rdf:li rdf:resource="#Francois Bry"/>
   </rdf:Seq>
 </dc:author>
 <dc:date:year>2004</dc:date>
 <dc:identifier>http://www.pms.informatik.uni-muenchen.de/
  publikationen/PMS-FB/PMS-FB-2004-12.pdf</dc:identifier>
</rdf:Description>
```

REWERSE Ontology:
extends SWRC with project-specific aspects
OWL, generated with Protege, ~5000 lines
59 Classes, 78 Datatype and object properties, 157 Instances, 269 inst. properties

Publications Ontology:
- weekly crawled & created (Wednesday noon)
- 6 partners from REWERSE currently
- ~15.000 lines, ~2000 publications,
- dynamic, changing, growing!



Research challenges in Data Extraction

- Interactive pattern learning:
  positive and negative examples, heuristics, theory.
- Auto-adaptation of Wrappers when layout changes.
- Extracting from PDF and similar sources.
- Optimization of Meta-Search.
- Better interaction with terminological reasoning.
- Generic wrappers: Feature extraction.

Generic Data Extraction...

Context of the problem

- automatic product retrieving
- product feature detection & extraction
- from web tabular data.

Context of the problem

- automatic product retrieving
- product feature detection & extraction
- from web tabular data.

Using Ontologies for Extracting Product Features from Web Pages*

Wolfgang Holzinger, Bernhard Krüpl, and Marcus Herzog

Database and Artificial Intelligence Group, Vienna University of Technology,
Favoritenstraße 9-11, A-1040 Wien, Austria
{holzing, kruepl, herzog}@dbai.tuwien.ac.at

Abstract. In this paper, we show how to use ontologies to bootstrap a knowledge acquisition process that extracts product information from tabular data on Web pages. Furthermore, we use logical rules to reason about product specific properties and to derive higher-order knowledge about product features. We will also explain the knowledge acquisition process, covering both ontological and procedural aspects. Finally, we will give an qualitative and quantitative evaluation of our results.

ISWC'06

Typical Candidate

Typical Candidate

General	
Weight	5.8 oz
Width	3.3 in
Depth	0.9 in
Height	2.2 in
Body material	Stainless steel
Main Features	
Sensor resolution	3.2 megapixels
Optical sensor type	CCD
Effective sensor resolution	3,200,000 pixels
Gross sensor resolution	3,300,000 pixels
Optical sensor size	1/2.7 in
Light sensitivity	ISO 50, ISO 100, ISO [illegible] 400

Essential Problem

map table contents to product model

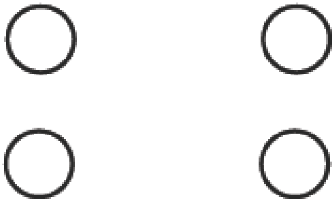
- textual content in table cells
- table layout
- background knowledge

Containment forms Context

Battery	
Weight	200 g
Size	11x15x8 mm

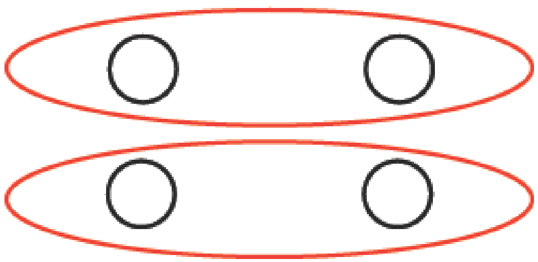
Containment forms Context

Battery	
Weight	200 g
Size	11x15x8 mm



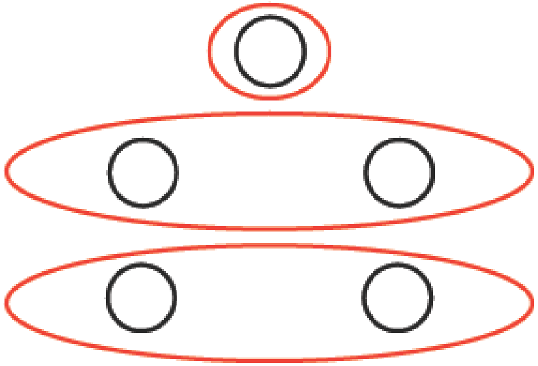
Containment forms Context

Battery	
Weight	200 g
Size	11x15x8 mm



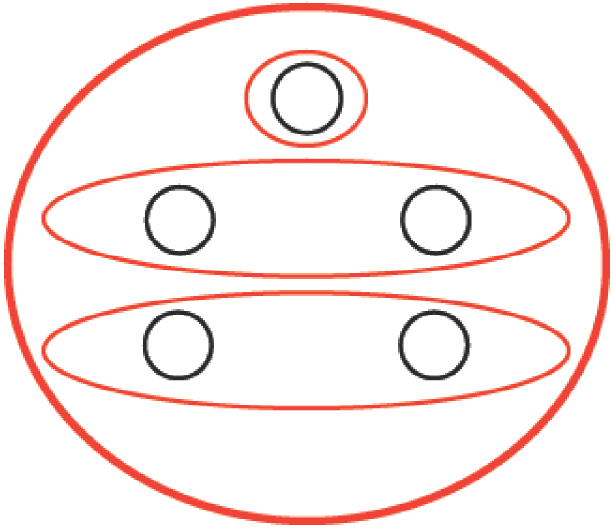
Containment forms Context

Battery	
Weight	200 g
Size	11x15x8 mm



Containment forms Context

Battery	
Weight	200 g
Size	11x15x8 mm



Containment forms Context

- cells are basic building blocks
- cells contain isolated words / phrases
- tabular arrangements group cells - multiple levels possible
- Proposal: these groupings determines which parts form a semantical unit

Dual Interpretation

a HTML <table> structure with 52 <tr> elements, each containing 1 or 2 <td>

a digital camera named “Powershot 116a”, manufactured by “Canon”, 5 Megapixel sensor, etc...

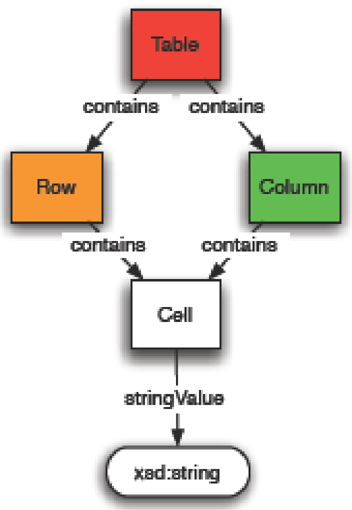
Dual Interpretation

a HTML <table> structure with 52 <tr> elements, each containing 1 or 2 <td>

a digital camera named "Powershot 116a", manufactured by "Canon", 5 Megapixel sensor, etc...

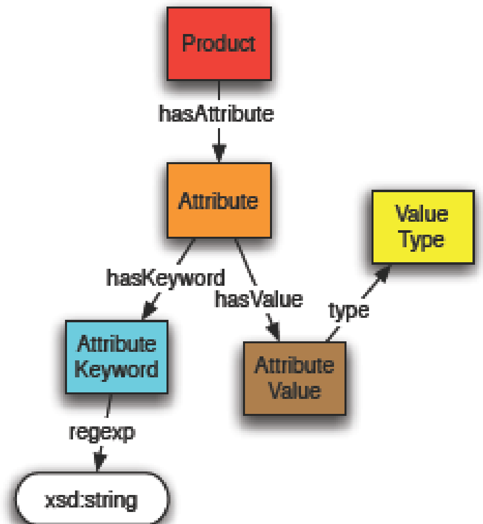
- two distinct ontologies:
  - table
  - product

Dual Interpretation



a HTML <table> structure with 52 <tr> elements, each containing 1 or 2 <td>

a digital camera named "Powershot 116a", manufactured by "Canon", 5 Megapixel sensor, etc...



- two distinct ontologies:
  - table
  - product

Table Ontology

A table consist of rows, columns and cells.

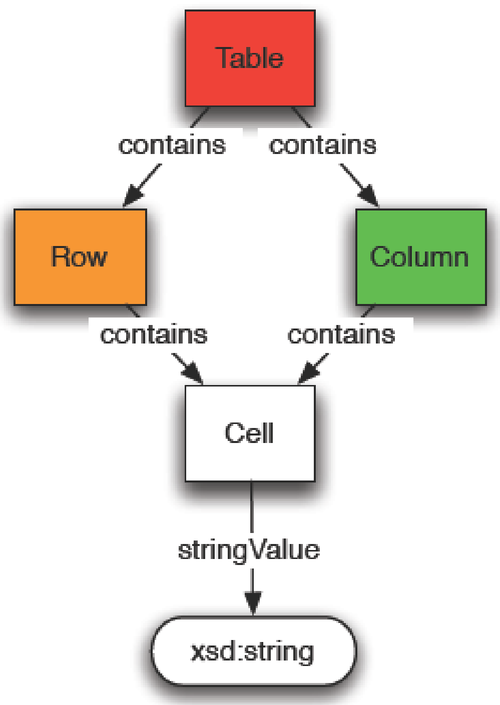
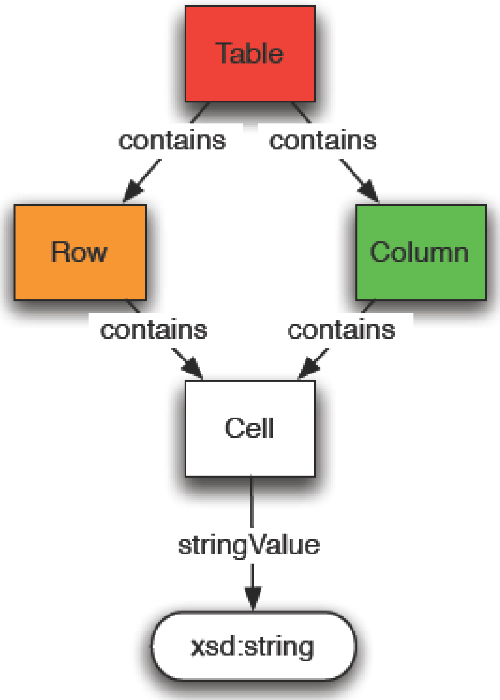


Table Ontology

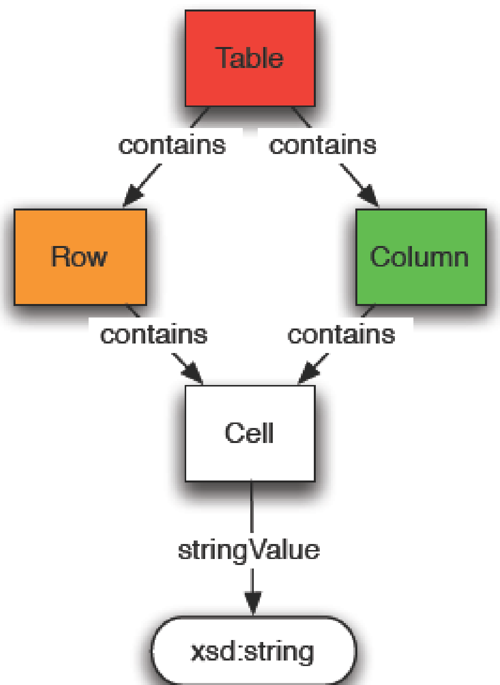
A table consist of rows, columns and cells.



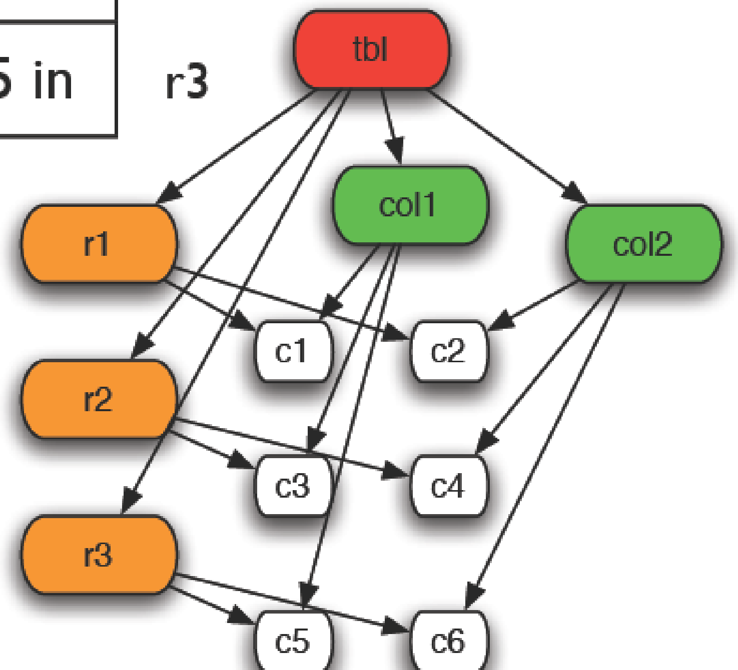
	col1	col2	
	width	2 in	r1
	height	1 in	r2
	depth	1.5 in	r3

Table Ontology

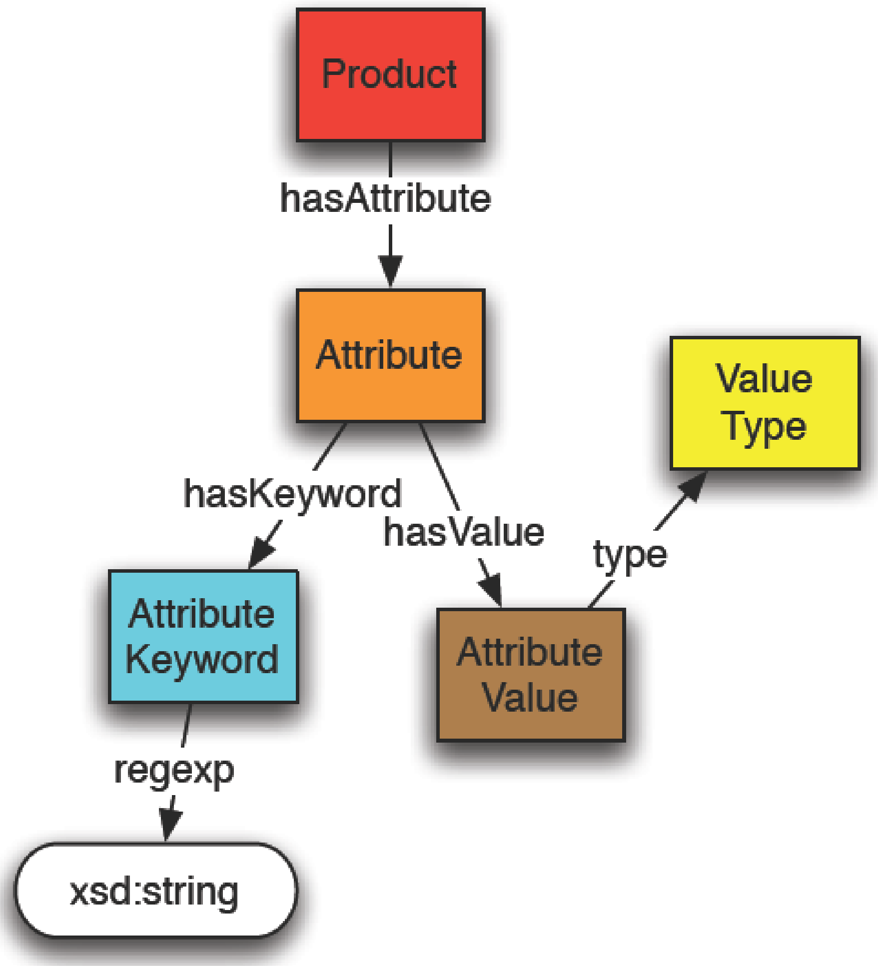
A table consist of rows, columns and cells.



	col1	col2
r1	width	2 in
r2	height	1 in
r3	depth	1.5 in



Product Ontology: Basic Structure



- simple model
- product is a collection of attributes
- attributes have
  - a keyword
  - a value

Value-Types & Spotters

- ProductType
  - 1024 x 768
  - 1cm x 1cm x 1cm
- FractionType
  - 1/500
  - 1:1.8
- UnitType
  - 7.5 lbs.
  - 4.4"

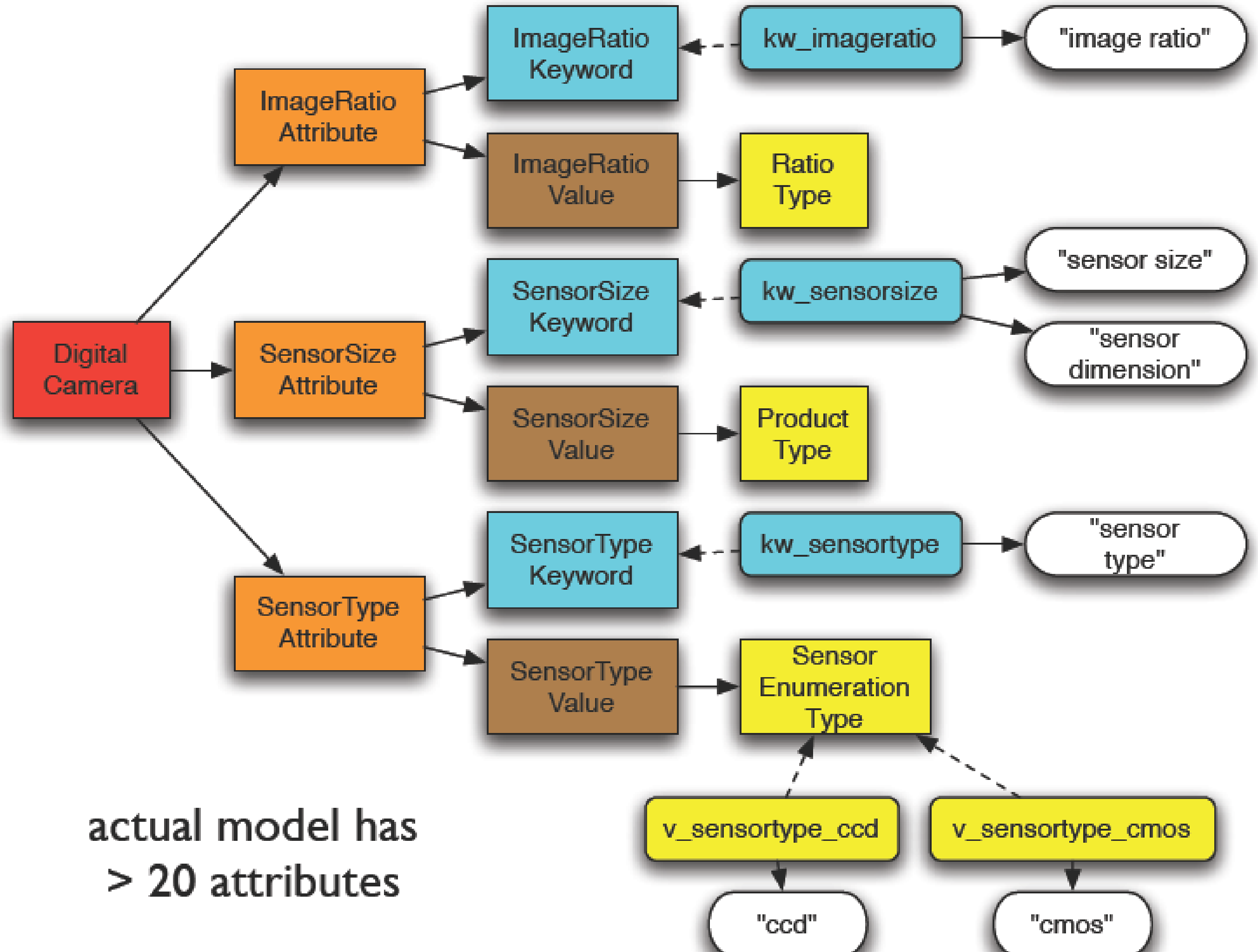
Spotters are small procedures that can recognize a value type in a given text.

Value-Types are recognized via *regular expressions* and have distinct OWL concepts each.

Value-Types can be nested: `1cm x 1cm x 1cm`

equivalent units: `1 kg = 2 lbs`

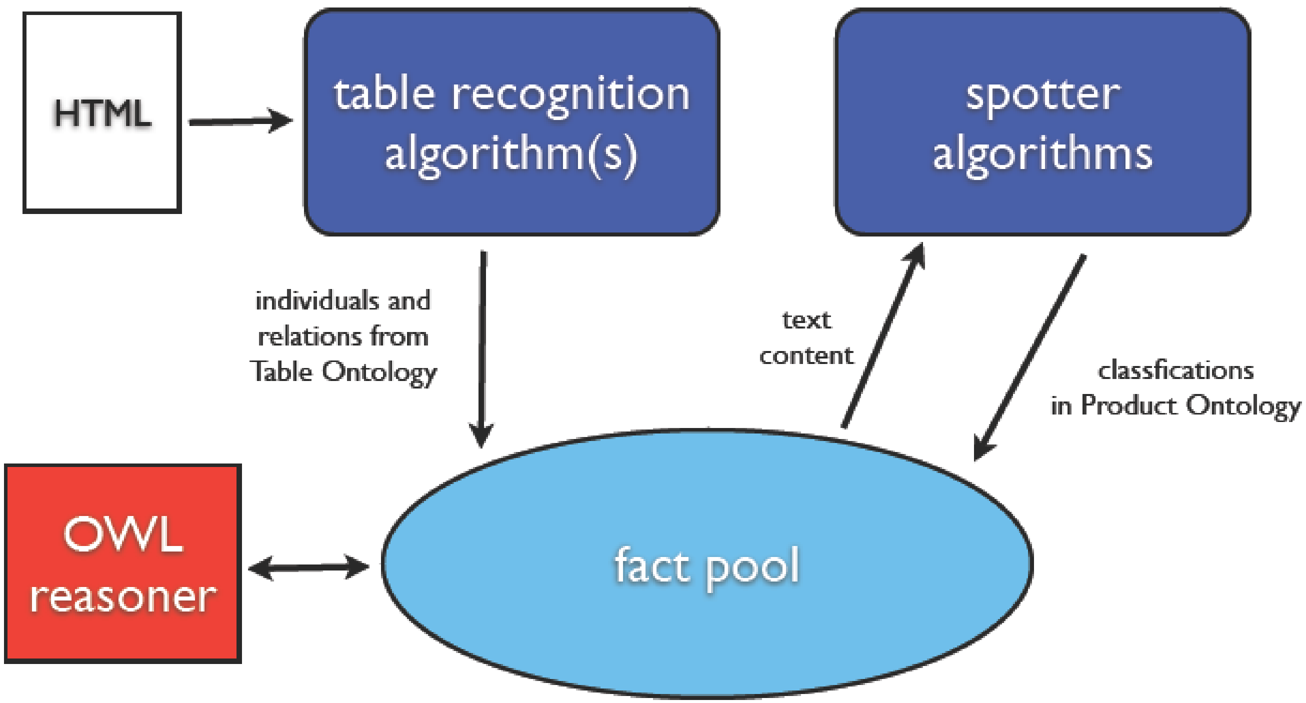
Digicam Ontology



- specifies each keyword and value in detail
- different spellings, abbreviations
- external types

actual model has > 20 attributes

Basic Information Flow



Example (I)

facts asserted
by table recognition

these are all instances of
concepts of the **table**
ontology

Example (1)

facts asserted
by table recognition

these are all instances of concepts of the **table** ontology

- tbl is a **table**
- r1, r2, r3 is a **row**.
  - tbl **contains** r1,r2,r3
- col1, col2 is a **column**.
  - tbl **contains** col1,col2
- c1,c2,...,c6 is a **cell**.
  - c1 hasText *"Weight"*
  - c2 hasText *"8.5 oz"*
  - c3 hasText *"Width"*
  - r1 **contains** c1,c2
  - r2 **contains** c3,c4
  - r3 **contains** c5,c6
  - col1 **contains** c1,c3,c5
  - col2 **contains** c2,c4,c6

Example (2)

facts asserted by content spotters

1) c1 hasText "Weight"

Example (2)

facts asserted by content spotters

1) c1 hasText "Weight"

SPOT: "Weight" contains the keyword **kw_weight**

Example (2)

facts asserted by content spotters

1) c1 hasText "Weight"

2) c1 contains **kw_weight**

Example (2)

facts asserted by content spotters

1) c1 hasText “Weigh

c1 a Cell

kw_weight a WeightKeyword

2) c1 contains **kw_weight**

Example (2)

facts asserted by content spotters

1) c1 hasText “Weigh

c1 a Cell

kw_weight a WeightKeyword

2) c1 contains **kw_weight**

Product ontology says:

$$\text{WeightKeyword} \equiv$$
$$\text{Cell} \sqcap \text{contains.WeightKeyword}$$

Example (2)

facts asserted by content spotters

1) c1 hasText "Weight"

2) c1 contains **kw_weight**

3) c1 a **WeightKeyword**

Example (2)

higher level classification

Example (2)

higher level classification

- c1 a **WeightKeyword**
- c2 a **WeightValue**
- r1 contains c1,c2

TECHNISCHE UNIVERSITÄT WIEN
VIENNA UNIVERSITY OF TECHNOLOGY

DBAI

Example (2)

higher level classification

- c1 a **WeightKeyword**
- c2 a **WeightValue**
- r1 contains c1,c2

DEF: WeightAttribute $\equiv$ contains.WeightKeyword $\sqcap$ contains.WeightValue

Example (2)

higher level classification

- c1 a **WeightKeyword**
- c2 a **WeightValue**
- r1 contains c1,c2

DEF: WeightAttribute $\equiv$ contains.WeightKeyword $\sqcap$ contains.WeightValue

$\Rightarrow$ r1 a **WeightAttribute**

Example (3)

top level classification

Example (3)

top level classification

- r1 a **WeightAttribute**
- r2 a **SizeAttribut**
- tbl contains r1,r2

Example (3)

top level classification

- r1 a **WeightAttribute**
- r2 a **SizeAttribut**
- tbl contains r1,r2

DEF: SimpleProduct ≡ contains.WeightAttribute ∩ contains.SizeAttribute

Example (3)

top level classification

- r1 a **WeightAttribute**
- r2 a **SizeAttribut**
- tbl contains r1,r2

DEF: SimpleProduct $\equiv$ contains.WeightAttribute $\cap$ contains.SizeAttribute

$\Rightarrow$ tbl a **SimpleProduct**

Empirical Results - I

- groundtruth table
  - manually annotated tables
  - tool for marking up concepts on a webpage
- compare manual recall + accuracy

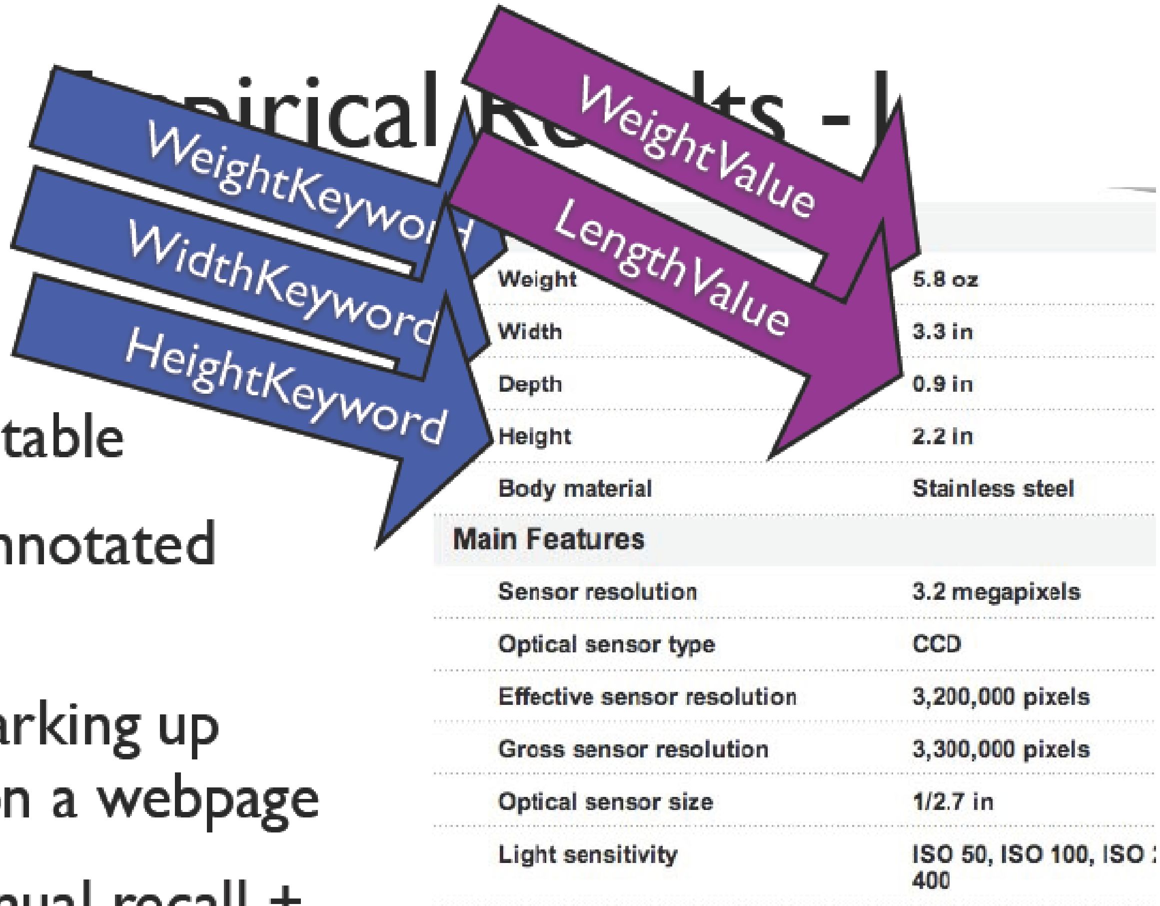
We annotated 30 tables selected randomly out of about 2000

Empirical Results - I

- groundtruth table
  - manually annotated tables
  - tool for marking up concepts on a webpage
- compare manual recall + accuracy

General	
Weight	5.8 oz
Width	3.3 in
Depth	0.9 in
Height	2.2 in
Body material	Stainless steel
Main Features	
Sensor resolution	3.2 megapixels
Optical sensor type	CCD
Effective sensor resolution	3,200,000 pixels
Gross sensor resolution	3,300,000 pixels
Optical sensor size	1/2.7 in
Light sensitivity	ISO 50, ISO 100, ISO 400

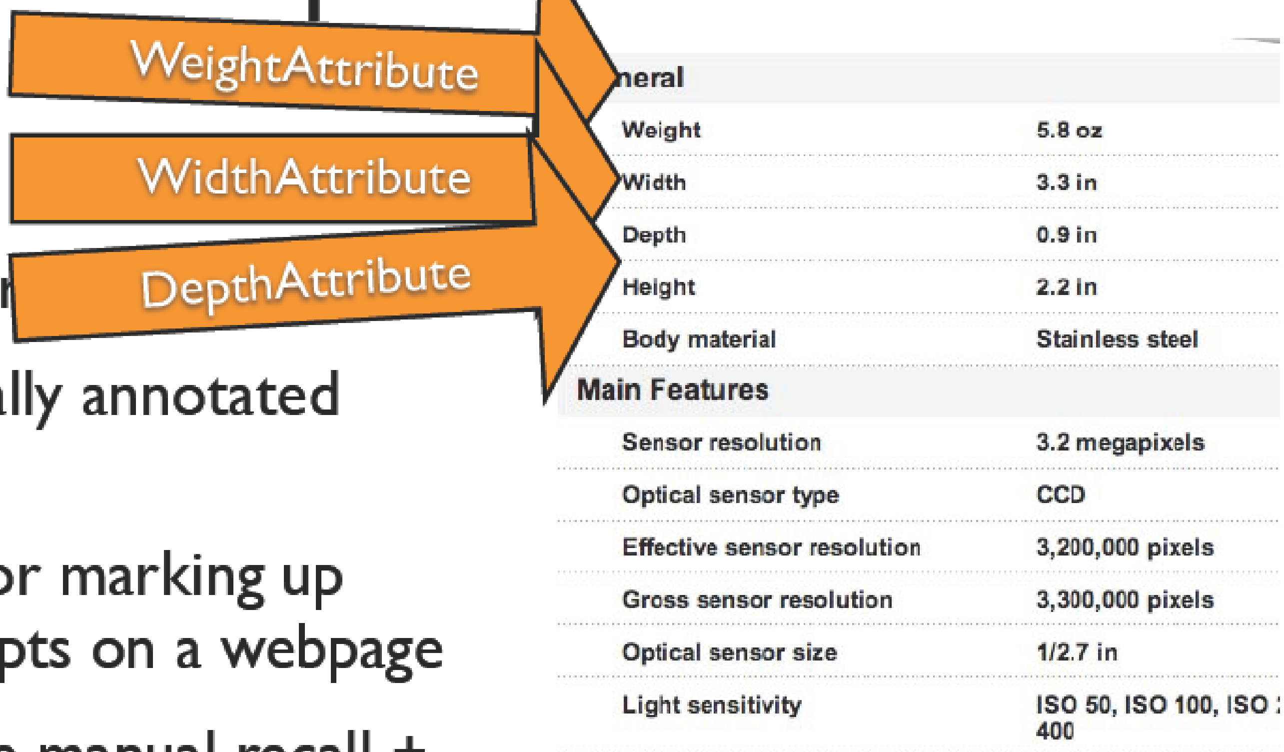
We annotated 30 tables selected randomly out of about 2000



- groundtruth table
  - manually annotated tables
  - tool for marking up concepts on a webpage
- compare manual recall + accuracy

We annotated 30 tables selected randomly out of about 2000

Empirical Results - I



neral	
Weight	5.8 oz
Width	3.3 in
Depth	0.9 in
Height	2.2 in
Body material	Stainless steel
Main Features	
Sensor resolution	3.2 megapixels
Optical sensor type	CCD
Effective sensor resolution	3,200,000 pixels
Gross sensor resolution	3,300,000 pixels
Optical sensor size	1/2.7 in
Light sensitivity	ISO 50, ISO 100, ISO : 400

- groundtr
  - manually annotated tables
  - tool for marking up concepts on a webpage
- compare manual recall + accuracy

We annotated 30 tables selected randomly out of about 2000

Empirical Results - 2

Average number of (per document) **ground truth**			Average number (per document) **identified**		
keywords	values	attributes	keywords	values	attributes
5.13	6.78	4.82	3.22	4.02	3.38

Recall			**Precision**		
keywords	values	attributes	keywords	values	attributes
62.8%	59.3%	70.1%	65.8%	41.6%	92.4%

F-measure		
keywords	values	attributes
64.27%	48.90%	79.72%

Empirical Results - 2

Average number of (per document) **ground truth**			Average number (per document) **identified**		
keywords	values	attributes	keywords	values	attributes
5.13	6.78	4.82	3.22	4.02	3.38

Recall			**Precision**		
keywords	values	attributes	keywords	values	attributes
62.8%	59.3%	70.1%	65.8%	41.6%	92.4%

F-measure		
keywords	values	attributes
64.27%	48.90%	79.72%

Empirical Results - 2

Average number of (per document) **ground truth**			Average number (per document) **identified**		
keywords	values	attributes	keywords	values	attributes
5.13	6.78	4.82	3.22	4.02	3.38

Recall			**Precision**		
keywords	values	attributes	keywords	values	attributes
62.8%	59.3%	70.1%	65.8%	41.6%	92.4%

F-measure		
keywords	values	attributes
64.27%	48.90%	79.72%

Summary

- exploit dual interpretation of data present by making use of two ontologies
- top level strategy for bridging the gap: *containment forms context*
- mix procedures freely with declarative statements

summary

- **PRO**
- declarative & procedural approaches combined
- spotters, TR, etc nicely decoupled
- „easy“ modifications by editing model

- **CON**
- slow performance, reasoner optimization?
- some concepts do not model well in OWL DL
- rules should replace some algorithms
- value ranges should be fuzzy

Thank You.