Semistructured Data and Mediation

Prof. Letizia Tanca

Dipartimento di Elettronica e Informazione

Politecnico di Milano

SEMISTRUCTURED DATA

FOR THESE DATA THERE IS SOME FORM OF STRUCTURE, BUT IT IS NOT AS

- PRESCRIPTIVE
- REGULAR
- COMPLETE

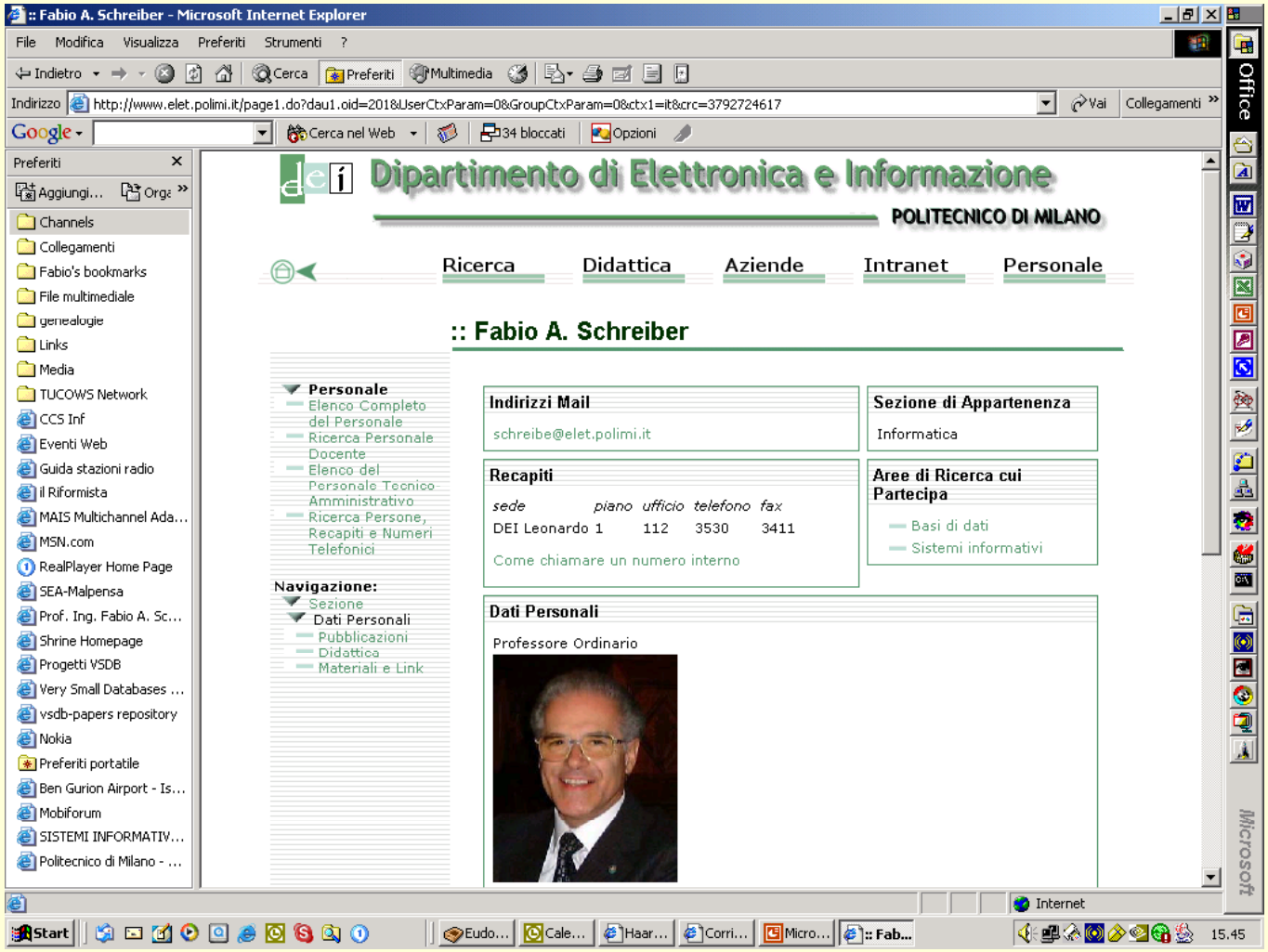
AS IN TRADITIONAL DBMSs

EXAMPLES

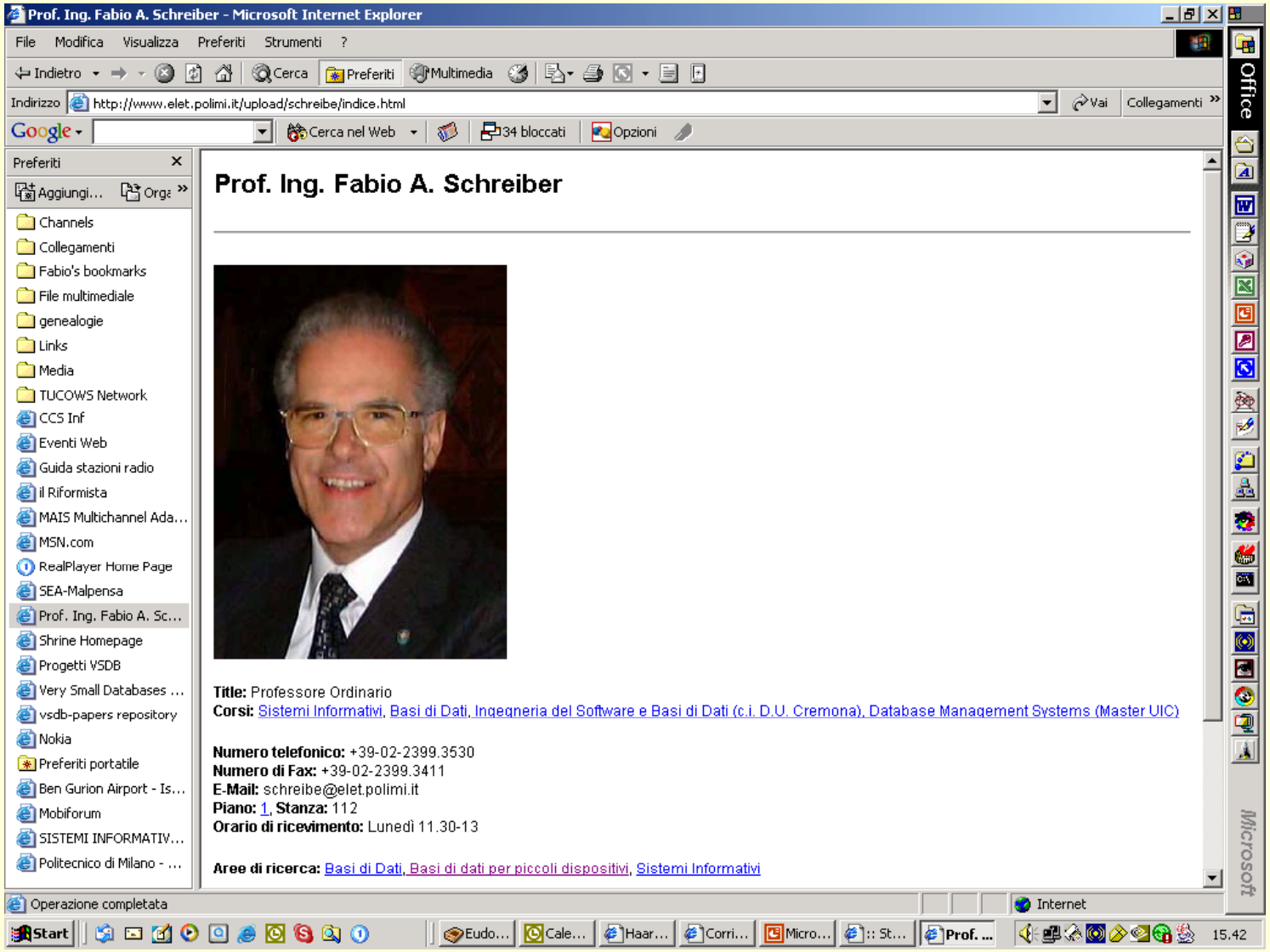
- WEB DATA
- XML DATA
- BUT ALSO DATA DERIVED FROM THE INTEGRATION OF HETEROGENEOUS DATASOURCES

AN EXAMPLE OF SEMISTRUCTURED DATA

a page produced form a database

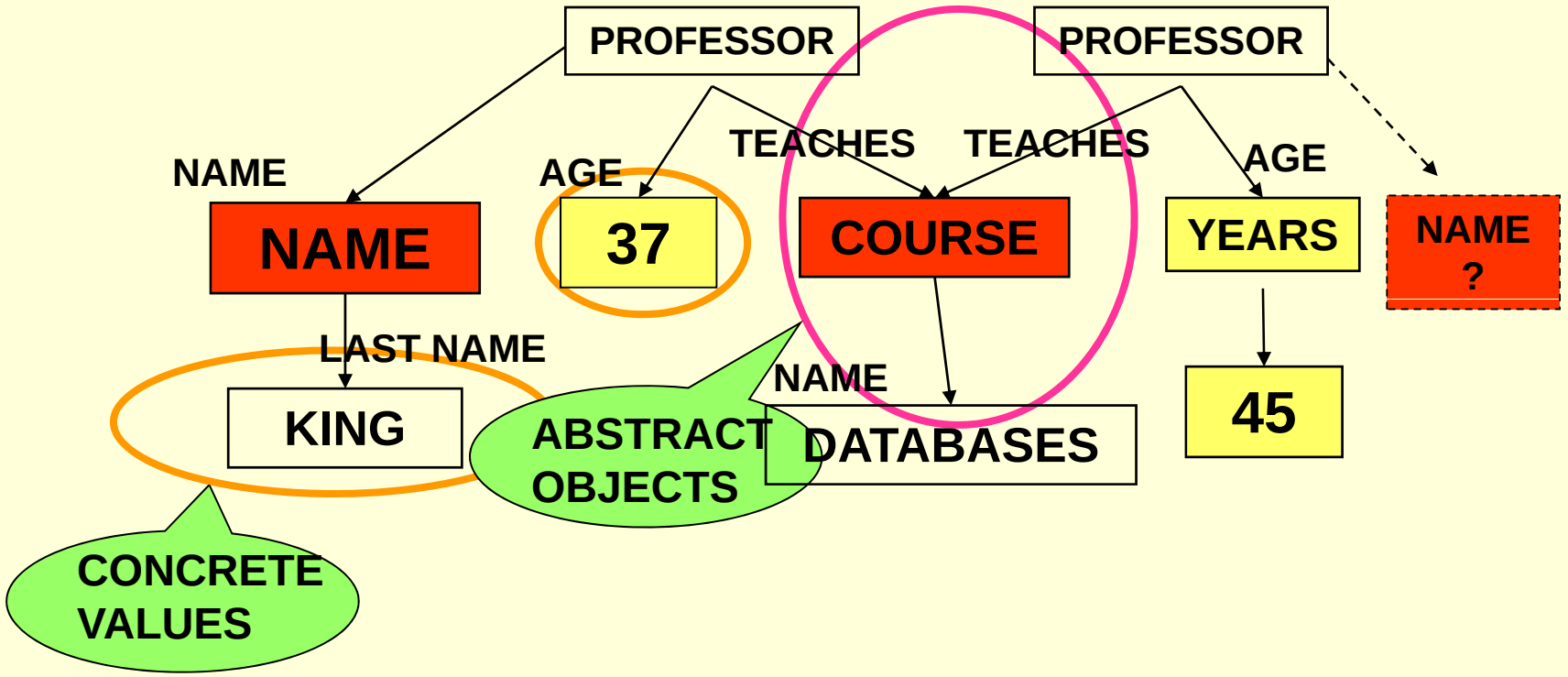


AN EXAMPLE OF SEMISTRUCTURED DATA



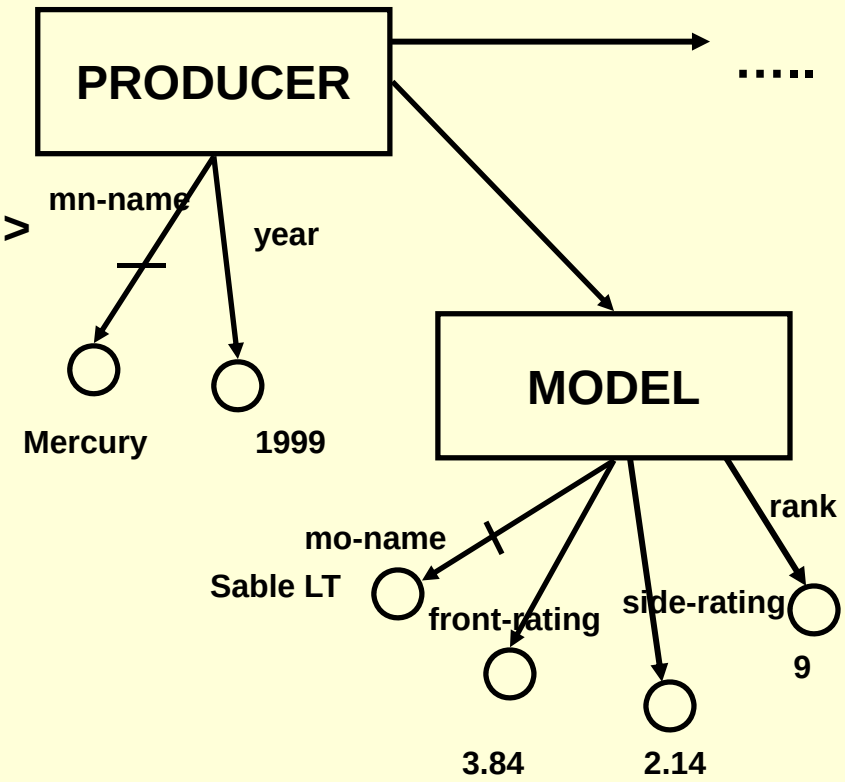
AN EXAMPLE OF SEMISTRUCTURED DATA

GRAPH-BASED REPRESENTATION: THE IRREGULAR DATA STRUCTURE APPEARS MORE CLEARLY



A SIMPLE XML DOCUMENT WITH ITS GRAPH BASED REPRESENTATION

```
<producer>
 <mn-name>Mercury</mn-name>
 <year>1999</year>
 <model>
 <mo-name>Sable LT</mo-name>
 <front-rating>3.84</front-rating>
 <side-rating>2.14</side-rating>
 <rank>9</rank>
 </model>
 ……..
</producer>
```



INFORMATION SEARCH IN SEMISTRUCTURED DATABASES

- WE WOULD LIKE TO:
  - INTEGRATE
  - QUERY
  - COMPARE

  DATA WITH DIFFERENT STRUCTURES ALSO WITH SEMISTRUCTURED DATA, JUST AS IF THEY WERE ALL STRUCTURED

DYNAMIC INTEGRATION OF SEMISTRUCTURED DATABASES

AN OVERALL DATA REPRESENTATION SHOULD BE PROGRESSIVELY BUILT, AS WE DISCOVER AND EXPLORE NEW INFORMATION SOURCES

SEMISTRUCTURED DATA MODELS

- BASED ON
  - TEXT
  - TREES
  - GRAPHS
    - LABELED NODES
    - LABELED ARCS
    - BOTH

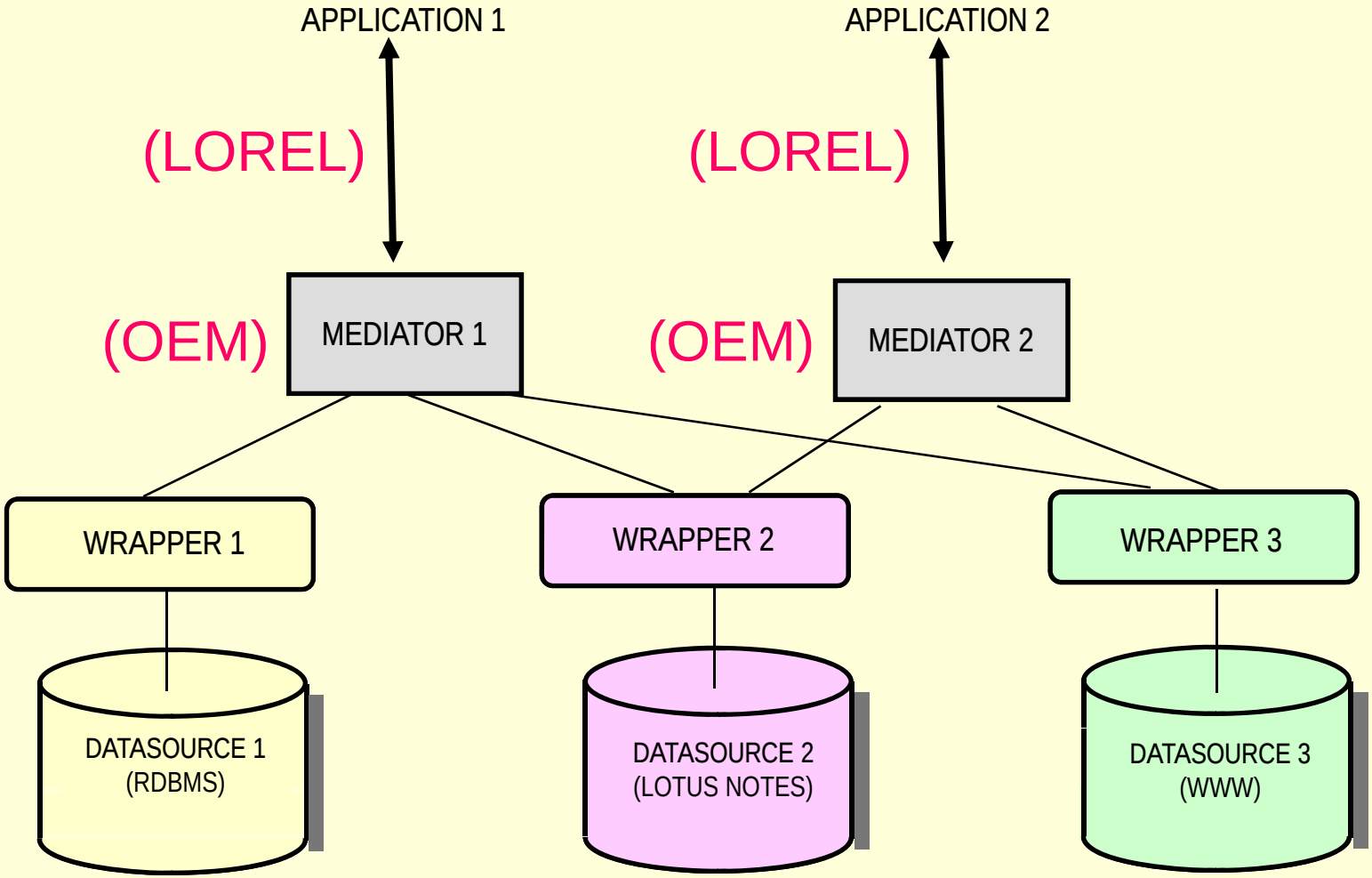
- THEY ARE ALL DIFFERENT AND DO NOT LEND THEMSELVES TO EASY INTEGRATION

Recall: MEDIATORS

The term mediation includes:

- the processing needed to make the interfaces work
- the knowledge structures that drive the transformations needed to transform data to information
- any intermediate storage that is needed (Wiederhold)

TSIMMIS


APPLICATION 1
APPLICATION 2
(LOREL)
(LOREL)
(OEM)
MEDIATOR 1
(OEM)
MEDIATOR 2
WRAPPER 1
WRAPPER 2
WRAPPER 3
DATASOURCE 1
(RDBMS)
DATASOURCE 2
(LOTUS NOTES)
DATASOURCE 3
(WWW)

Mediator-based approach

IN TSIMMIS:

- UNIQUE, GRAPH-BASED DATA MODEL
- DATA MODEL MANAGED BY THE MEDIATOR
- WRAPPERS FOR THE MODEL-TO-MODEL TRANSLATIONS

OEM (Object Exchange Model) (TSIMMIS)

- Graph-based
- Does not represent the schema
- Directly represents data : self-descriptive

```
<temp-in-farenheit,int,80>
```

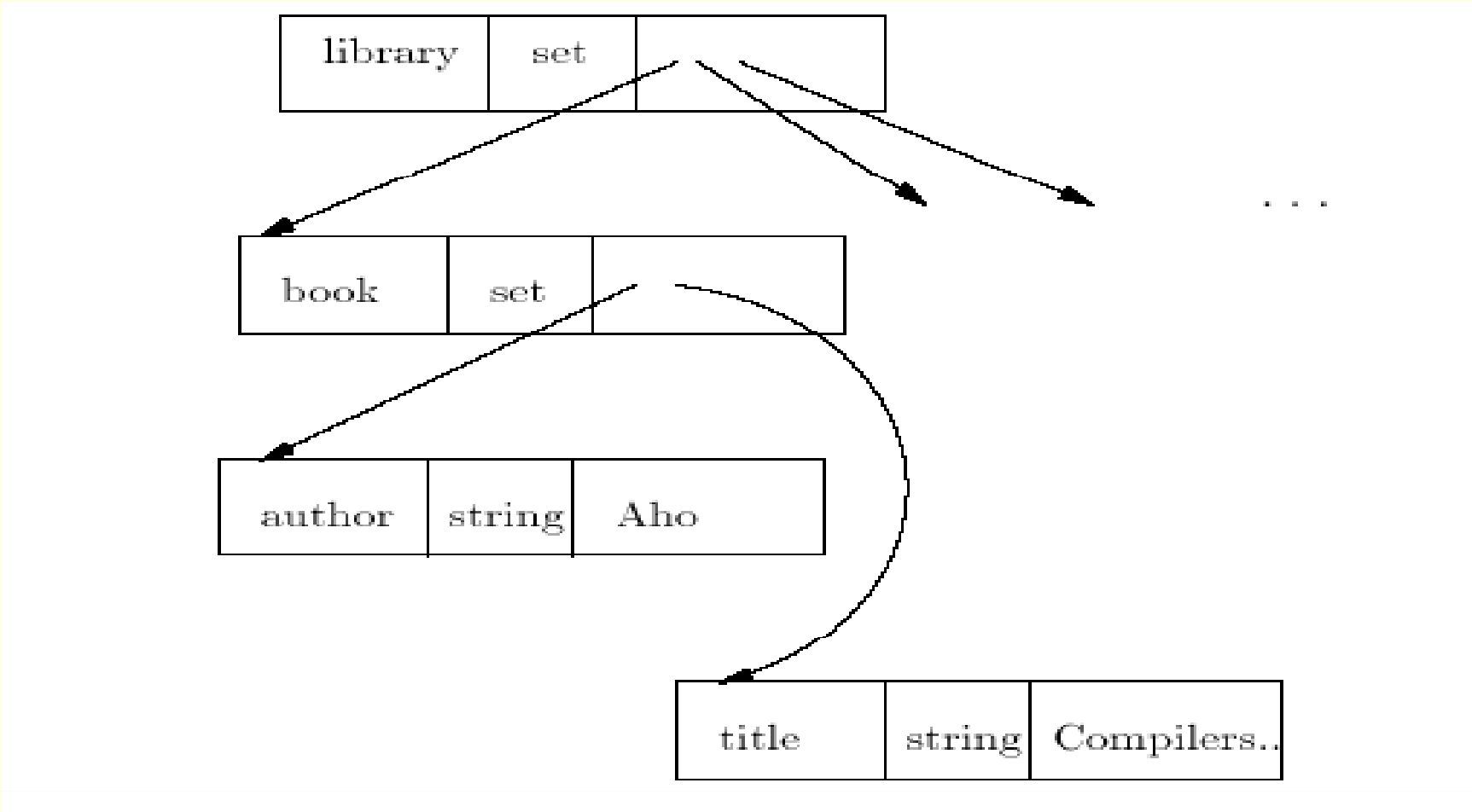
Object structure

<Object-id,label,type,value>

Nested structure

$\langle$set-of-temps, set, $\{cmp_1, cmp_2\}\rangle$
cmp_1: $\langle$temp-in-Fahrenheit, int, 80$\rangle$
cmp_2: $\langle$temp-in-Celsius, int, 20$\rangle$

OEM (Object Exchange Model) (TSIMMIS)



Typical complications when integrating

- Each mediator is specialized into a certain domain (e.g. weather forecast), thus
- Each mediator must know domain metadata , which convey the data semantics
- On-line duplicate recognition and removal (no designer to solve conflicts at design time here)

Query formulation

“Find books authored by Aho”

```
select library.book.title
where library.book.author = “Aho”
from library
```
(if more than one root is available)

OK, but if this query must be produced at run-time, how does the user (or the system, if a transformation has to be applied) know that:

- A node *library* exists, which contains nodes *book*, which in turn contain fields *author* and *title*
- TSIMMIS uses the *Dataguide*: a-posteriori schema, progressively built while exploring the data sources

TSIMMIS's language is *LOREL*

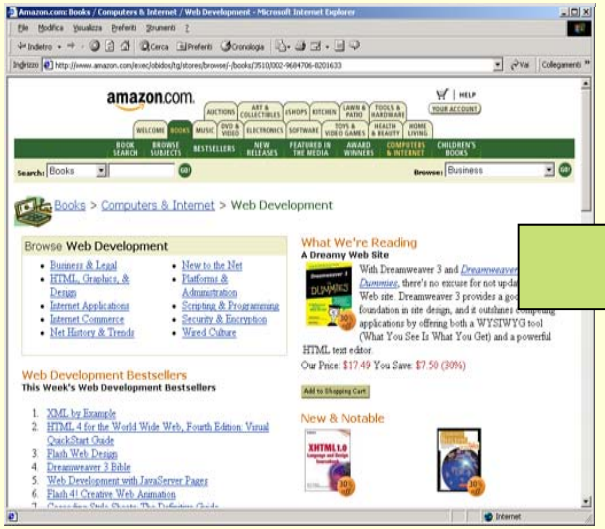
- *L*ightweight *O*bject *RE*pository *L*anguage
- Object-based
- Similar to OQL, with some modifications appropriate for semistructured data

```
select library.book.title
where library.book.author = "Aho"
from library
```

WRAPPERS (translators)

- Convert queries into queries/commands which are understandable for the specific data source
  - they can extend the query possibilities of a data source
- Convert query results from the source's format to a format which is understandable for the application

WRAPPERS



HTML page

Wrapper

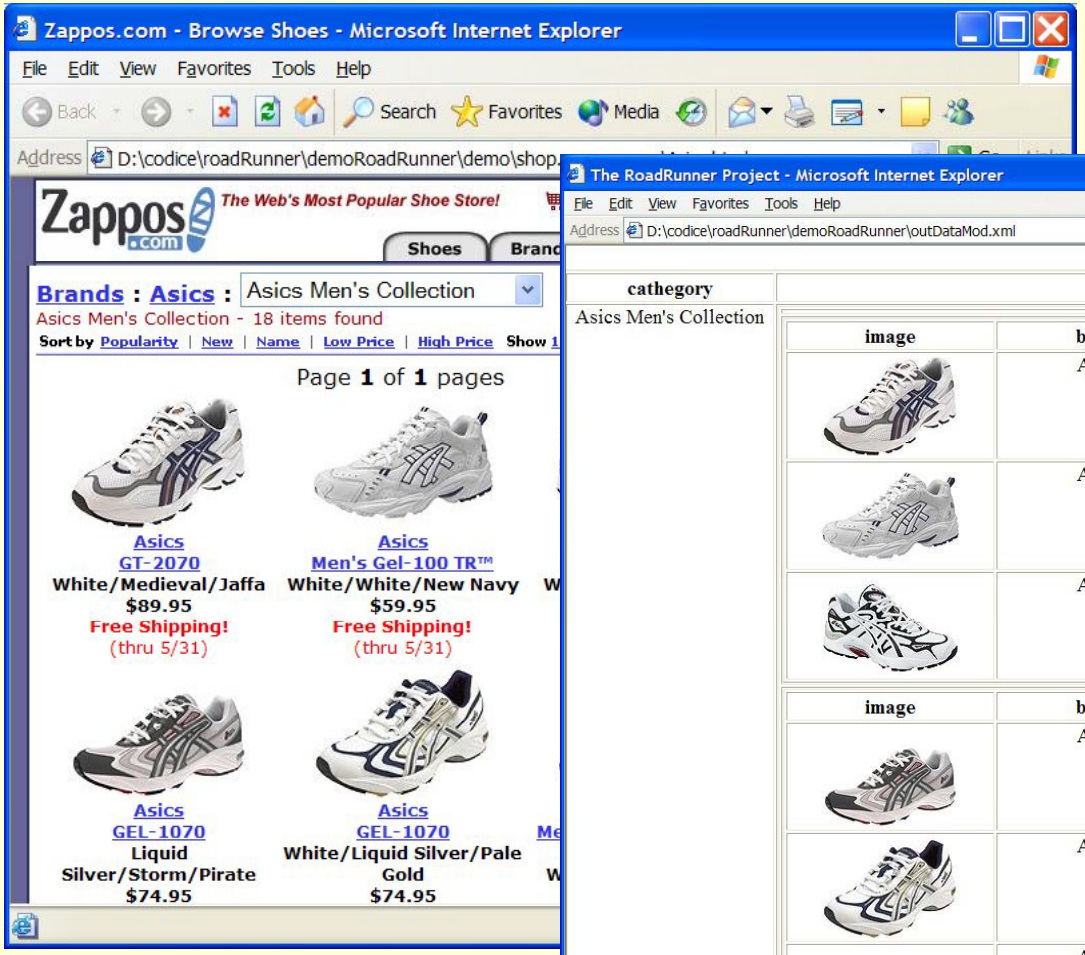
BookTitle	Author	Editor
The HTML Sourcebook	J. Graham	...
Computer Networks	A. Tannenbaum	...
Database Systems	R. Elmasri, S. Navathe	...
Data on the Web	S. Abiteboul, P.Buneman, D.Suciu	...

database table(s)
(or XML docs)

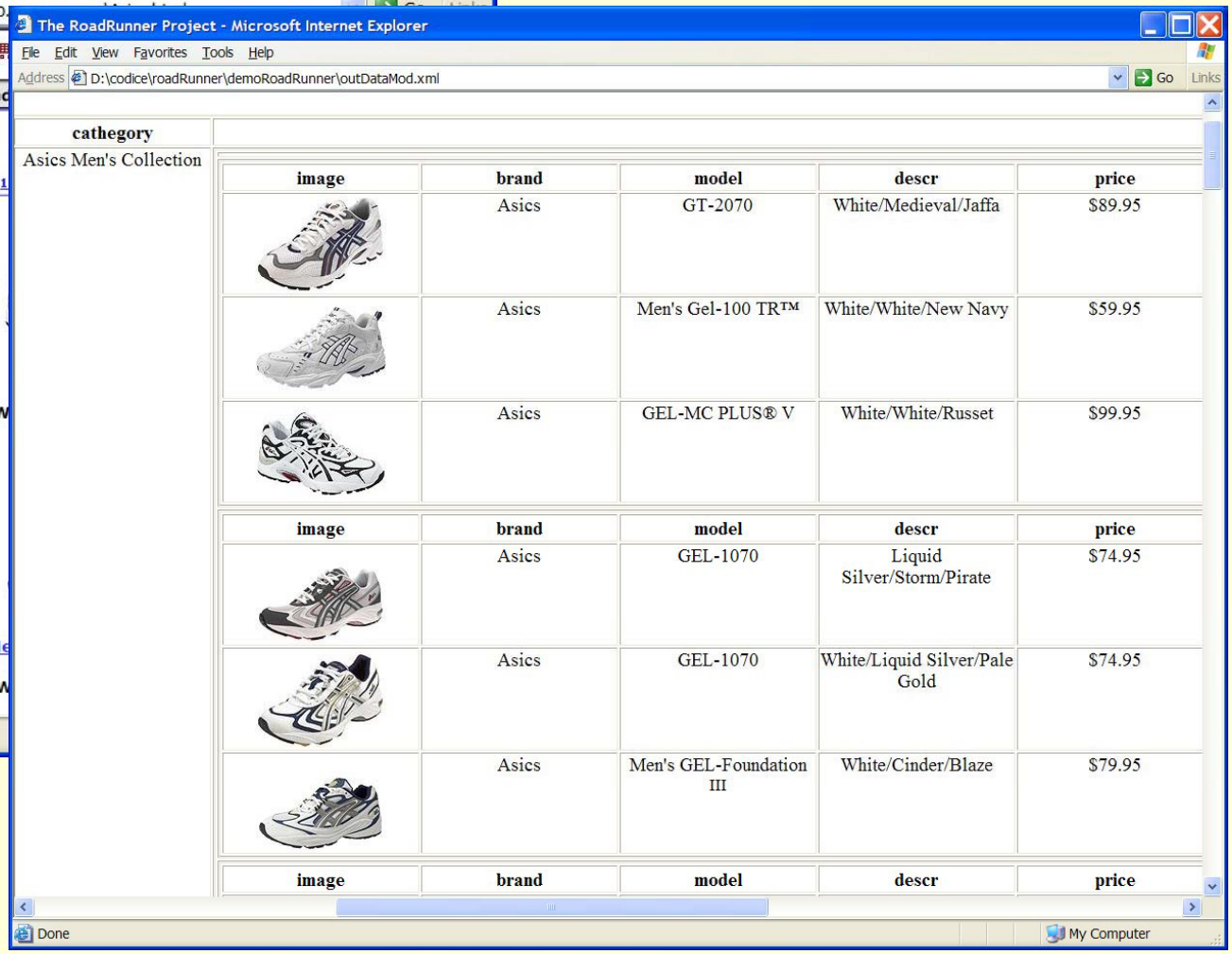
Example: estraction of information from HTML docs

- Information extraction
  - Source Format: plain text with HTML tags (no semantics)
  - Target Format: relational table (possibly nested, NF^2) or XML (we add *structure*, i.e. *semantics*)
- Wrapper
  - Software module which performs an *extraction step*
  - Intuition: use extraction rules which exploit the *marking tags*

A complex extraction process



20-30KB IN HTML



>10 attributes
with nesting

Problems

- Web sites change very frequently
- A layout change may affect the extraction rules
- Human-based maintenance of an ad-hoc wrapper is very expensive
- Better: ***automatic wrapper generation***

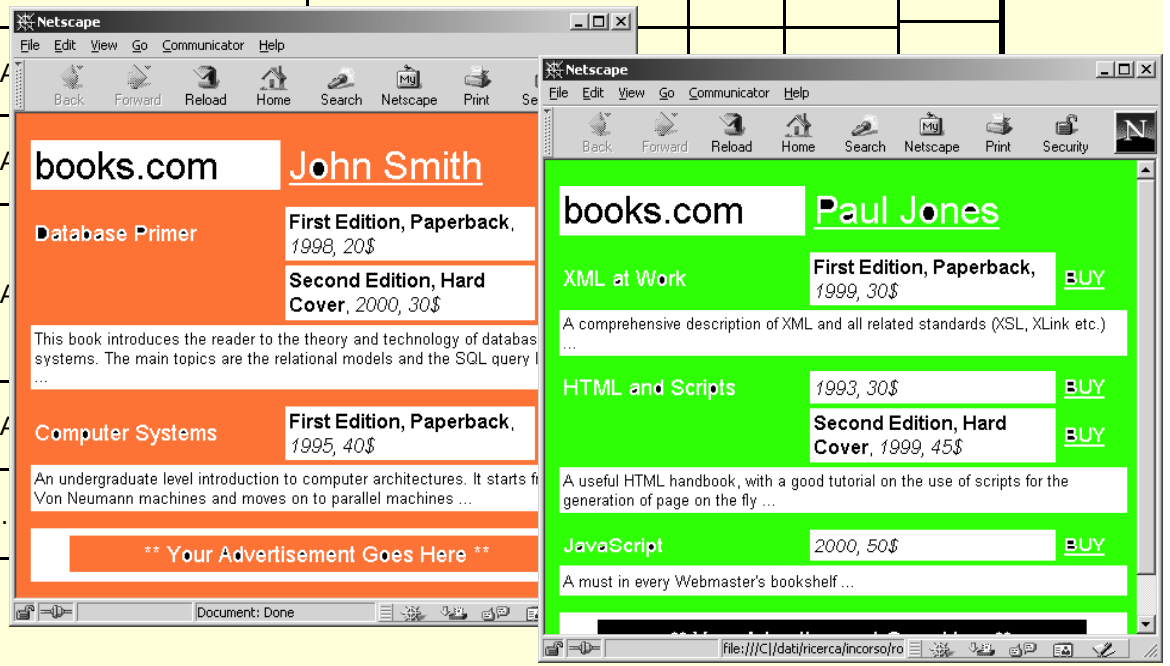
Automatic wrapper generation...

- We can only use them when pages are ***regular*** to some extent
- OK when:
  - Many pages sharing the same structure
  - e.g. pages are dynamically generated from a DB

→ ***data intensive web sites***

Online library

Name	Books				
	Title	Description	Editions		
			Details	Year	Price
John Smith	Database Primer	This book …	First Edition, Paperback	1998	20$
			Second Edition, Hard Cover	2000	30$
	Computer Systems				
Paul Jones	XML at Work				
	HTML and Scripts				
	JavaScripts				
…	…				



The ROAD RUNNER project

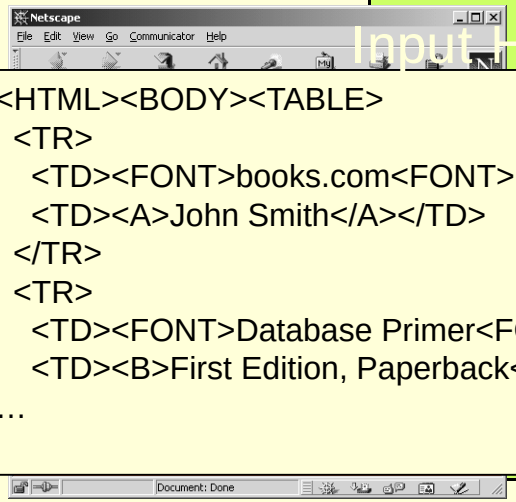
- Page Class
  - *It is the collection of all pages generated by the same script from a common dataset*
- Schema Derivation
  - *Given a set of HTML sample pages, belonging to the same class, find the underlying dataset structure (database schema)*

→ Solution: **Wrapper Generator**

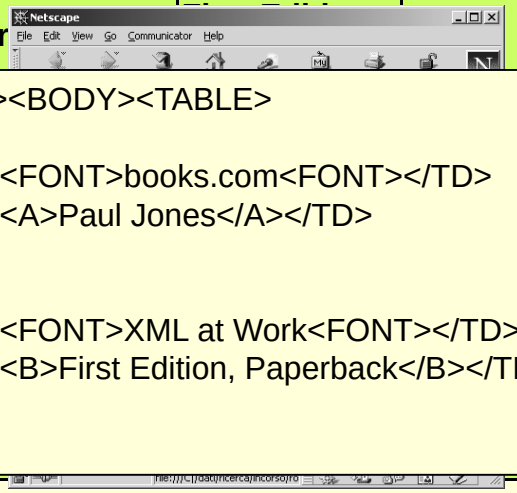
- Underlying dataset structure
- Extraction rules

A	B				
	C	D	E		
			F	G	H
John Smith	Database Primer	This book …	First Edition, …	1998	20$
			Second Edition, …	2000	30$
	Computer Systems	An under…			40$
					30$
					30$
					45$
					50$
					…

Input HTML Sources




```
<HTML><BODY><TABLE>
 <TR>
  <TD><FONT>books.com<FONT></TD>
  <TD><A>John Smith</A></TD>
 </TR>
 <TR>
  <TD><FONT>Database Primer<FONT></TD>
  <TD><B>First Edition, Paperback</B></TD>,
…
```




```
<HTML><BODY><TABLE>
 <TR>
  <TD><FONT>books.com<FONT></TD>
  <TD><A>Paul Jones</A></TD>
 </TR>
 <TR>
  <TD><FONT>XML at Work<FONT></TD>
  <TD><B>First Edition, Paperback</B></TD>,
…
```

```
<HTML><BODY><TABLE>
 <TR>
  <TD><FONT>books.com<FONT></TD>
  <TD><A> #PCDATA</A></TD></TR>
 (<TR>
  ( <TD><FONT> #PCDATA <FONT></TD>
      ( <TD><B> #PCDATA </B> )? </TD>
  ( <TR><TD><FONT> #PCDATA <FONT>
      <B> #PCDATA </B> </TD></TR> )+
 )+…
```

Target Schema

```
SET (
 TUPLE (A : #PCDATA;
        B : SET (
            TUPLE ( C : #PCDATA;
                    D : #PCDATA;
                    E : SET (
                        TUPLE ( F: #PCDATA;
                                G: #PCDATA;
                                H: #PCDATA)))
```

21° oct

Model Management approach

(Atzeni, Bernstein, others…)

- Given two different data models (e.g. OO and relational, or XML and OO, etc.) (when datasources are at least semistructured)
- Define general mappings from one model into another, which allow to
  - Map SQL schema to XML schema
  - Map data source to data warehouse
  - Map OO classes to data source tables, …
- To this end, one possibility is to use a **metamodel**

METAMODEL

- A METAMODEL IS AN ABSTRACT MODEL FOR THE SPECIFICATION OF CONCRETE MODELS

- TWO TYPES OF METAMODELS:
  1. GENERAL ENTITIES WHOSE SPECIALIZATIONS BECOME OBJECTS IN THE TARGET MODEL, E.G. GSMM

  2. ENTITIES DESCRIBING THE OBJECTS OF THE TARGET MODEL, E.G.GEOGRAPHIC DATA FILES (GDF)

Using a metamodel for integrated data representation

- TRANSLATION OF DIFFERENT MODELS INTO A UNIQUE FORMALISM
- EASY A-PRIORI COMPARISON BETWEEN THE DIFFERENT MODEL'S FEATURES
- AUTOMATIC TRANSLATION DICTATED BY THE REPRESENTATION RULES OF THE CONCRETE MODEL INTO THE METAMODEL

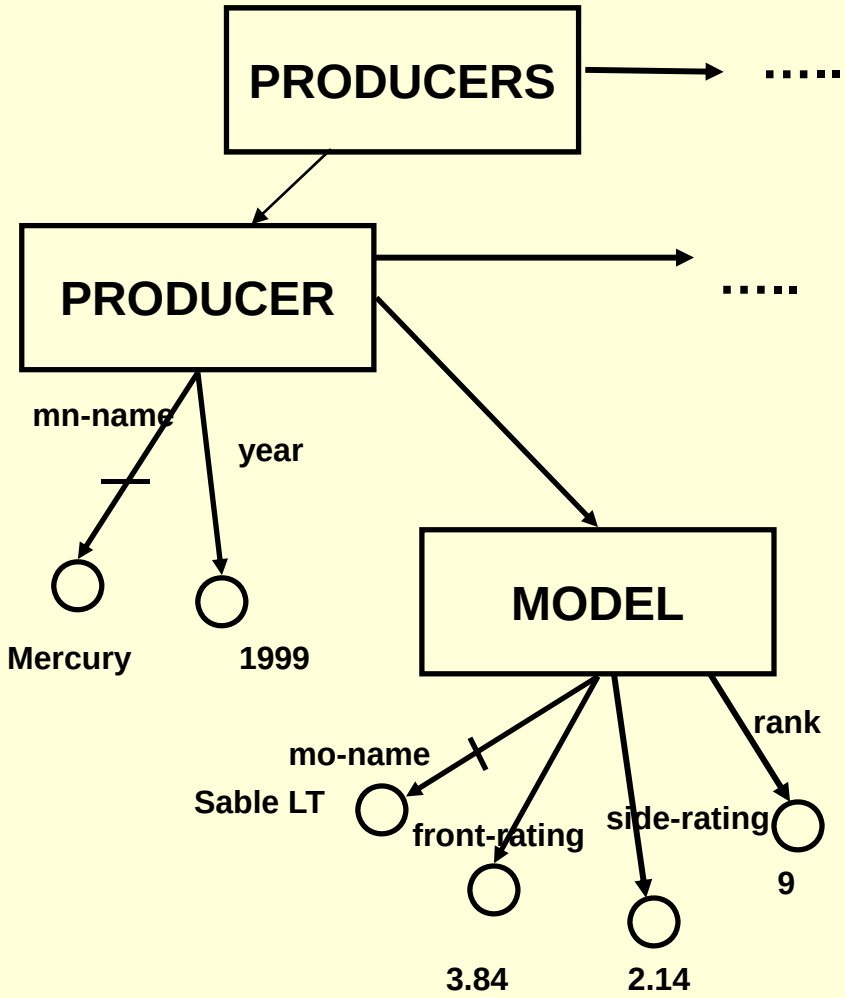
GSMM

(General Semistructured Meta-Model)

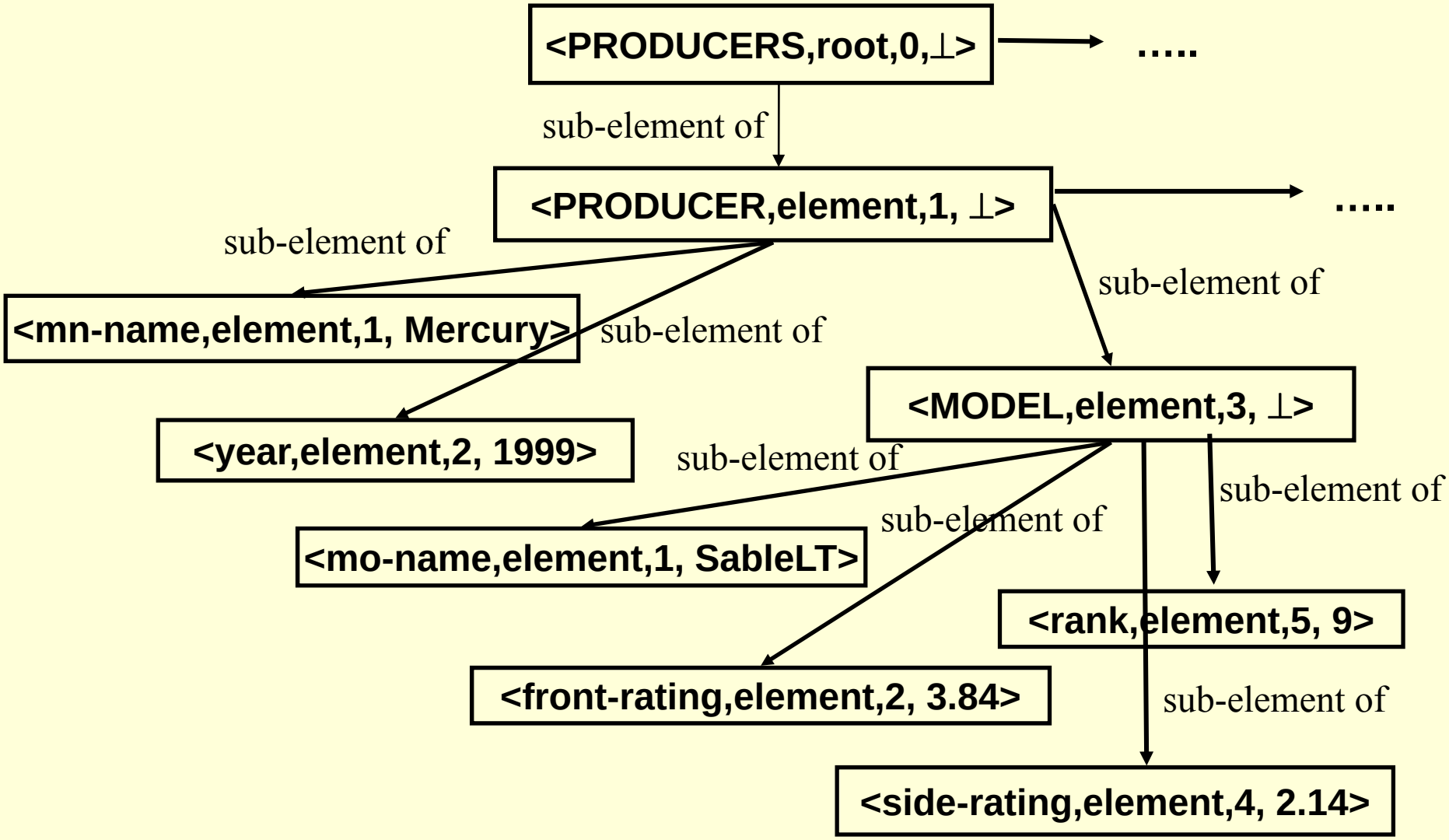
- GRAPH-BASED
- DOES NOT REPRESENT THE SCHEMA: SELF-DESCRIPTIVE AS TSIMMIS
- GSSM REPLACES THE CONCEPT OF SCHEMA WITH THAT OF CONSTRAINT
- INTRODUCTION OF FLEXIBILITY IN DATA REPRESENTATION

An XML document…

```
<producers>
 <producer>
  <mn-name>Mercury</mn-name>
  <year>1999</year>
  <model>
   <mo-name>Sable LT</mo-name>
   <front-rating>3.84</front-
   rating>
   <side-rating>2.14</side-rating>
   <rank>9</rank>
   </model>
  ……
 </producer>
  …
</producers>
```



Its representation in GSMM



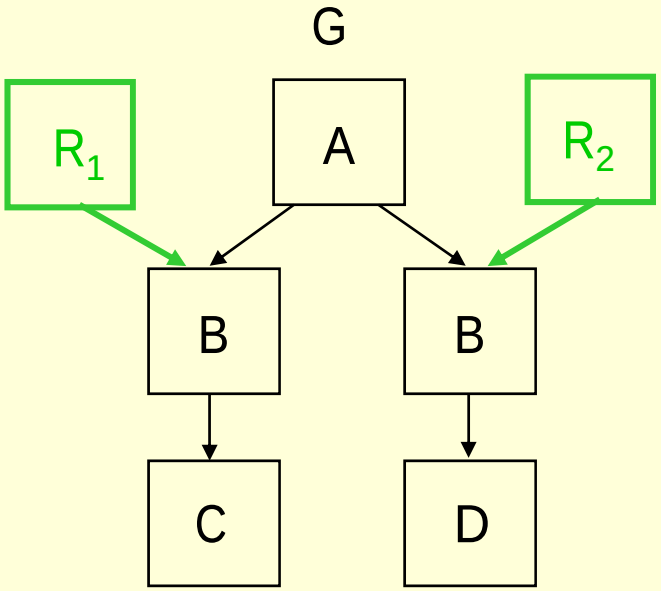
GSL

General Semistructured Language

RULE BASED, WHERE

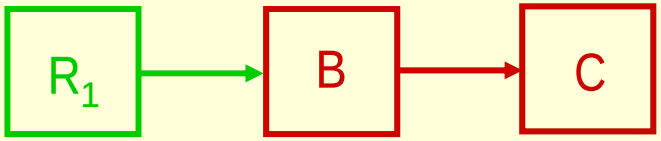
- A RULE IS A COLORED GRAPH
  - RED SOLID FOR POSITIVE PREMISES
  - RED DASHED FOR NEGATIVE PREMISES
  - GREEN SOLID FOR POSITIVE CONSEQUENCES
- RULES REPRESENT:
  - QUERIES
  - CONSTRAINTS (EMPTY CONSEQUENCE)

GSL QUERIES



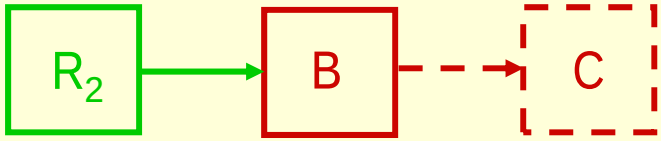
Q_1: **Find all the B nodes which have at least a *C* child and link them to a node *R_1***

Q_1

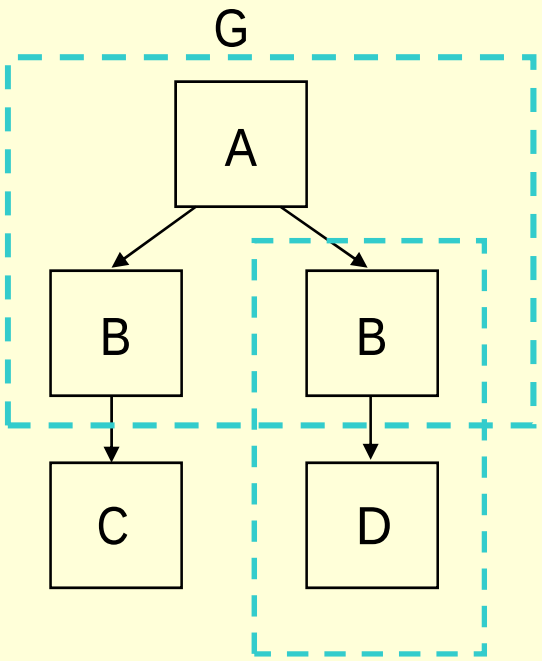


Q_2: **Find all the *B* nodes that do not have children *C* and link them to a node *R_2***

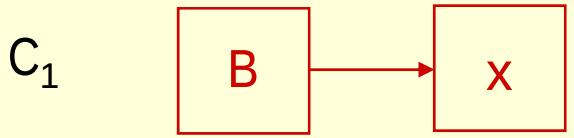
Q_2



CONSTRAINTS IN GSL



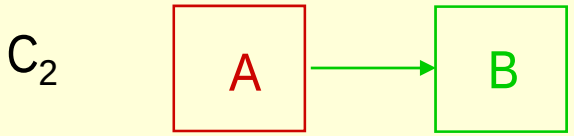
C_1: Whenever a *B* node has a child, this is a *C* node



This is false in the instance

{ x = C }

C_2: Each *A* node has at least a *B child*



This is true in the instance

APPLICATION OF THE GSMM METAMODEL

- THE METAMODEL ALLOWS THE CONSTRUCTION OF A GENERIC GRAPH WHICH REPRESENTS AN INSTANCE OF A CONCRETE MODEL
- PLUS CONSTRAINTS (also represented by means of graphs):
  - HIGH LEVEL, OR META- CONSTRAINTS – these dictate the sintactic rules of the object data model
  - LOW LEVEL CONSTRAINTS – as usual, these dictate the application domain semantics

- IT IS A METAMODEL OF THE FIRST KIND

XML expressed in GSMM

- EACH NODE LABEL HAS CARDINALITY 4:

  $n_i = <Ntag_i, Ntype_i, Norder_i, Ncontent_i>$

  $Ntype_i$ says whether the node is the root, an element, a text, …

  $Norder_i$ represents the node's position as a child of its parent node

  $Ncontent_i$ may assume a value of type PCDATA or value $\perp$

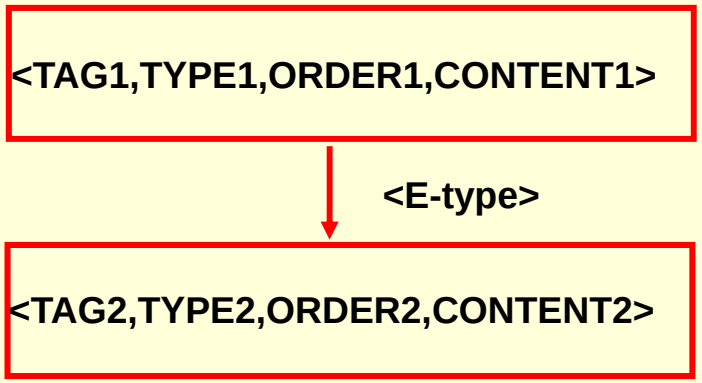
- EACH EDGE LABEL HAS CARDINALITY 1:

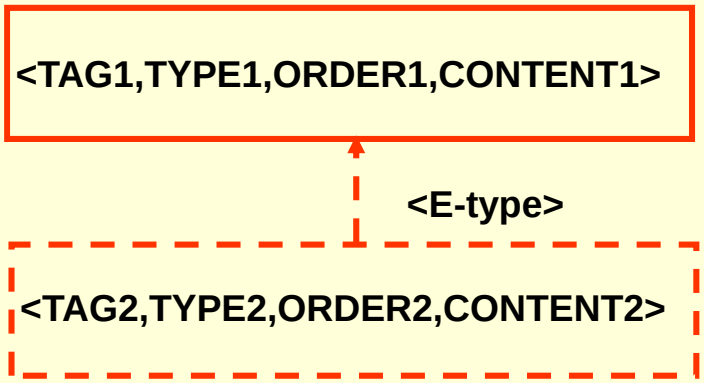
  $ej = < (nh, nk), ELj >$

  $EL_j = < Etype_j >$

  $Etype_j \in \{\text{attribute of, sub-element of}\}$

CONSTRAINT REPRESENTATION IN XML





{ E-type=SubElement-of $\rightarrow$ TYPE1=element
$\wedge$ (TYPE2=element $\vee$ TYPE2=text),
E-type=Attribute-of $\rightarrow$ TYPE1=element
$\wedge$ TYPE2=attribute }

{ TYPE1=root $\wedge$ CONTENT1=$\perp$ }

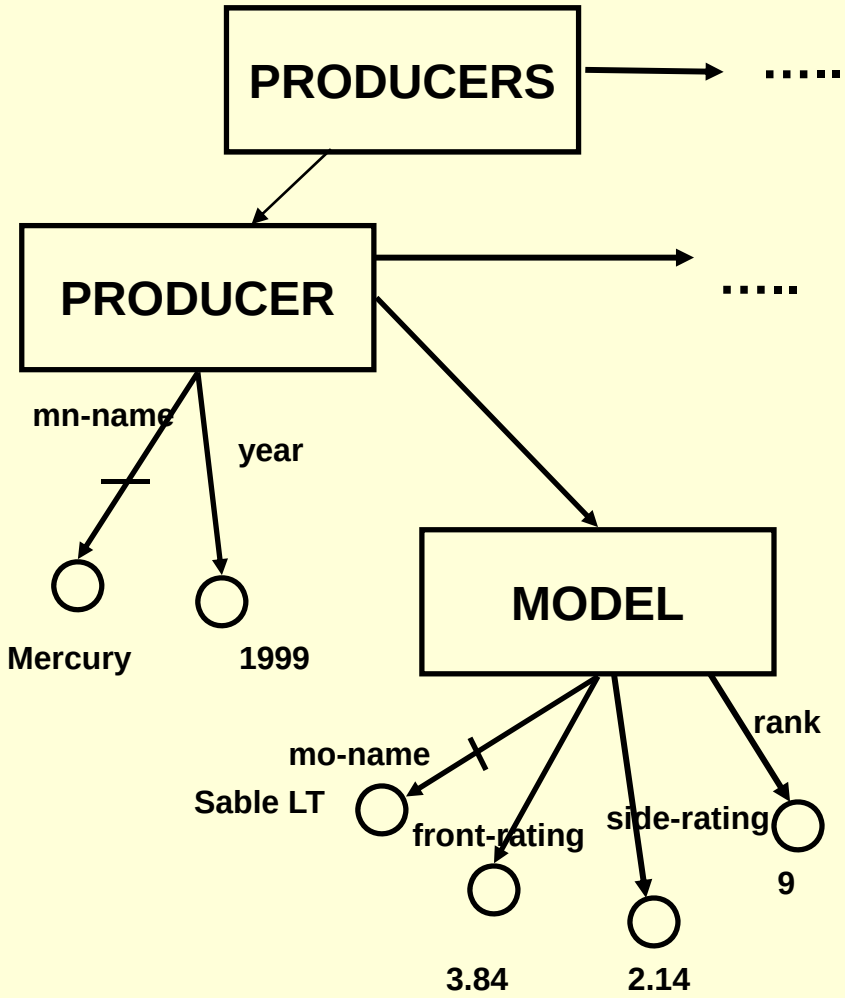
HIGH LEVEL, OR META-CONSTRAINT:
THE TYPE OF AN ARC DEPENDS ON THE TYPE OF THE DESTINATION NODES

HIGH LEVEL, OR META-CONSTRAINT:
THE GRAPH ROOT HAS TYPE *root* AND UNDEFINED CONTENT

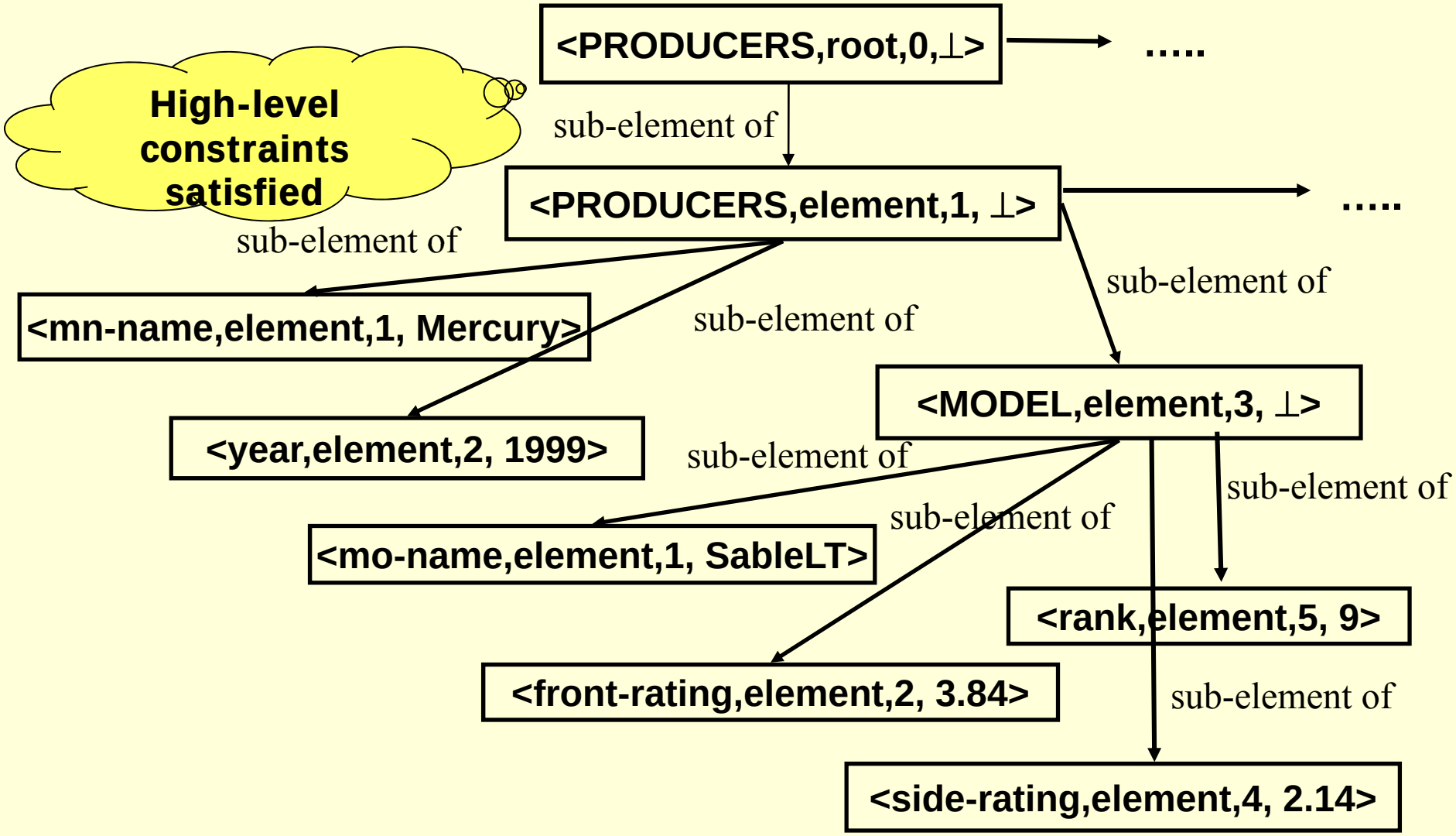
HIGH LEVEL CONSTRAINTS CONTRIBUTE TO THE DEFINITION OF THE STRUCTURE OF ANY XML DOCUMENT

An XML document…

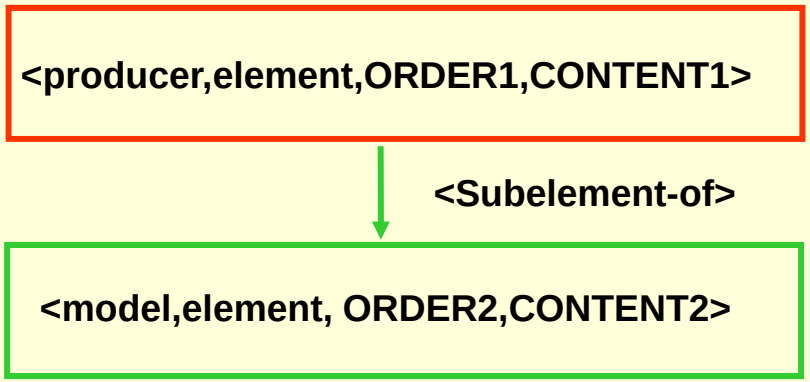
```
<producers>
 <producer>
  <mn-name>Mercury</mn-name>
  <year>1999</year>
  <model>
   <mo-name>Sable LT</mo-name>
   <front-rating>3.84</front-
   rating>
   <side-rating>2.14</side-rating>
   <rank>9</rank>
   </model>
  ……
 </producer>
  …
</producers>
```

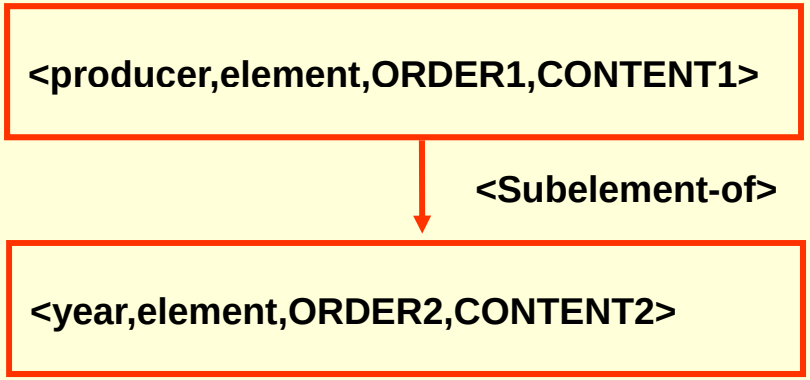


Its representation in GSMM



LOW-LEVEL, OR APPLICATION-LEVEL, CONSTRAINTS





{ CONTENT2>1990 }

Low-level constraint:
For each PRODUCER element there is at least one MODEL element

Low-level constraint:
YEAR must be greater than 1990

LOW-LEVEL CONSTRAINTS EXPRESS THE DOCUMENT'S SEMANTICS

Geographic Data Files (GDF)

- GDF is a standard, used for describing roadmaps
  - CITY STREET REGISTER
  - STREET SIGN ARCHIVE
  - TRAFFIC CONTROL SYSTEMS
  - CAR NAVIGATOR SYSTEMS (GPS)
  - …..
- IT IS A METAMODEL OF THE SECOND KIND

GDF STRUCTURE

- DATASET OF 82 ASCII CHARACTERS RECORDS
  - ENTITIES
  - ATTRIBUTES
  - RELATIONS
- 11 INTEREST THEMES
  - ROADS AND FERRIES
  - BRIDGES AND TUNNELS
  - RAILROADS
  - RIVERS
  - PUBLIC TRANSPORTATION
  - ADMINISTRATION AREAS
  - ……..
- 3 DESCRIPTION LEVELS FOR EACH THEME

GDF

- LEVEL 0 – TOPOLOGY
  - A PLANAR GRAPH:
    - POINT
    - ARC
    - NODE
    - POLYGON
  - EACH ELEMENT IS UNIQUELY IDENTIFIED BY AN *ID*
  - TOPOLOGICAL RULES GOVERN INTERNAL STRUCTURE AND RELATIONSHIPS AMONG ELEMENTS

GDF LEVELS

- LEVEL 1 – ELEMENTARY ENTITIES

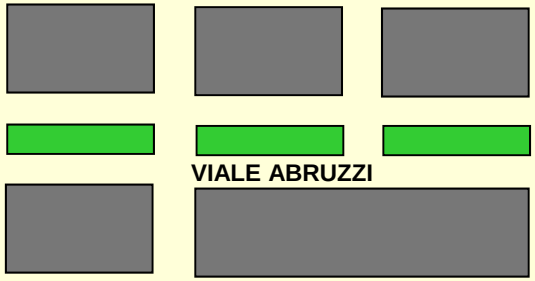
  THIS IS THE BASIS FOR THE **CITY STREET REGISTER**

  – ROAD ELEMENT
  – JUNCTION
  – TRAFFIC AREA
- LEVEL 2 – COMPLEX ENTITIES

  THIS IS THE BASIS FOR THE **GEOGRAPHIC INFORMATION SYSTEMS**

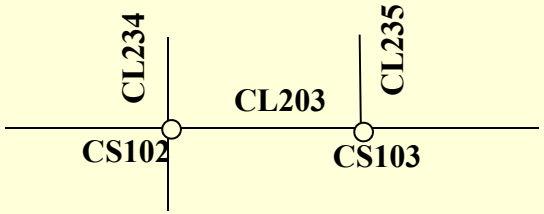
  – STREET
  – INTERSECTION

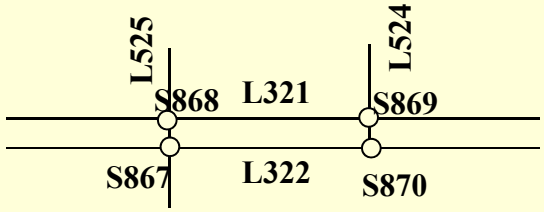
GDF LEVELS



CITY MAP

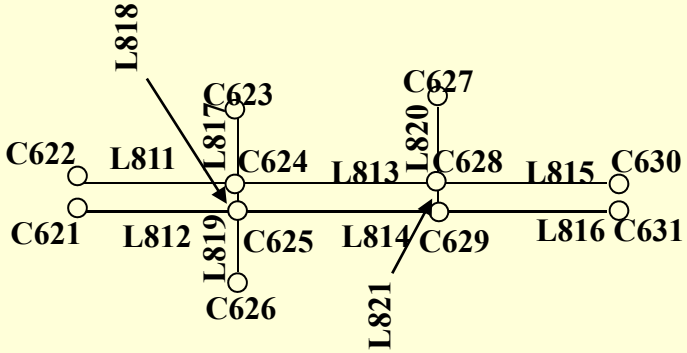
LEVEL 2



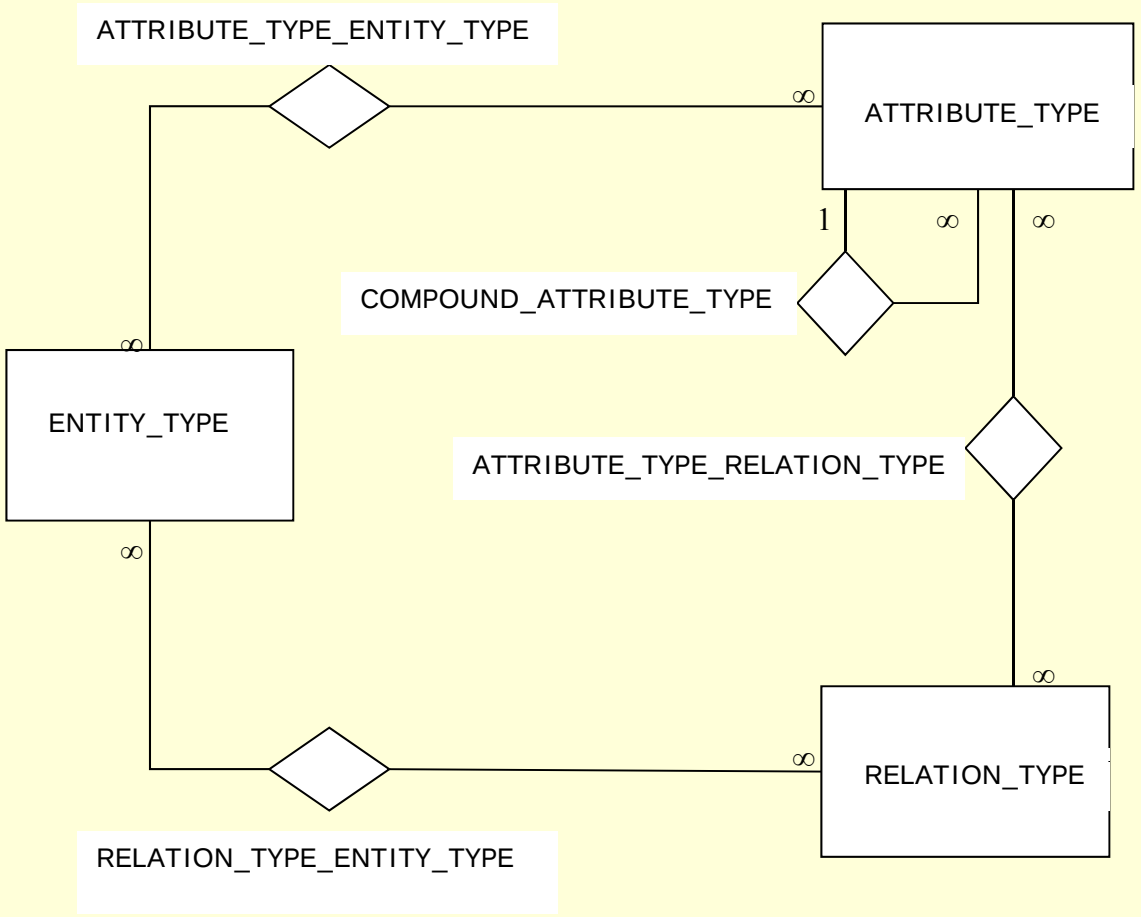


LEVEL 1

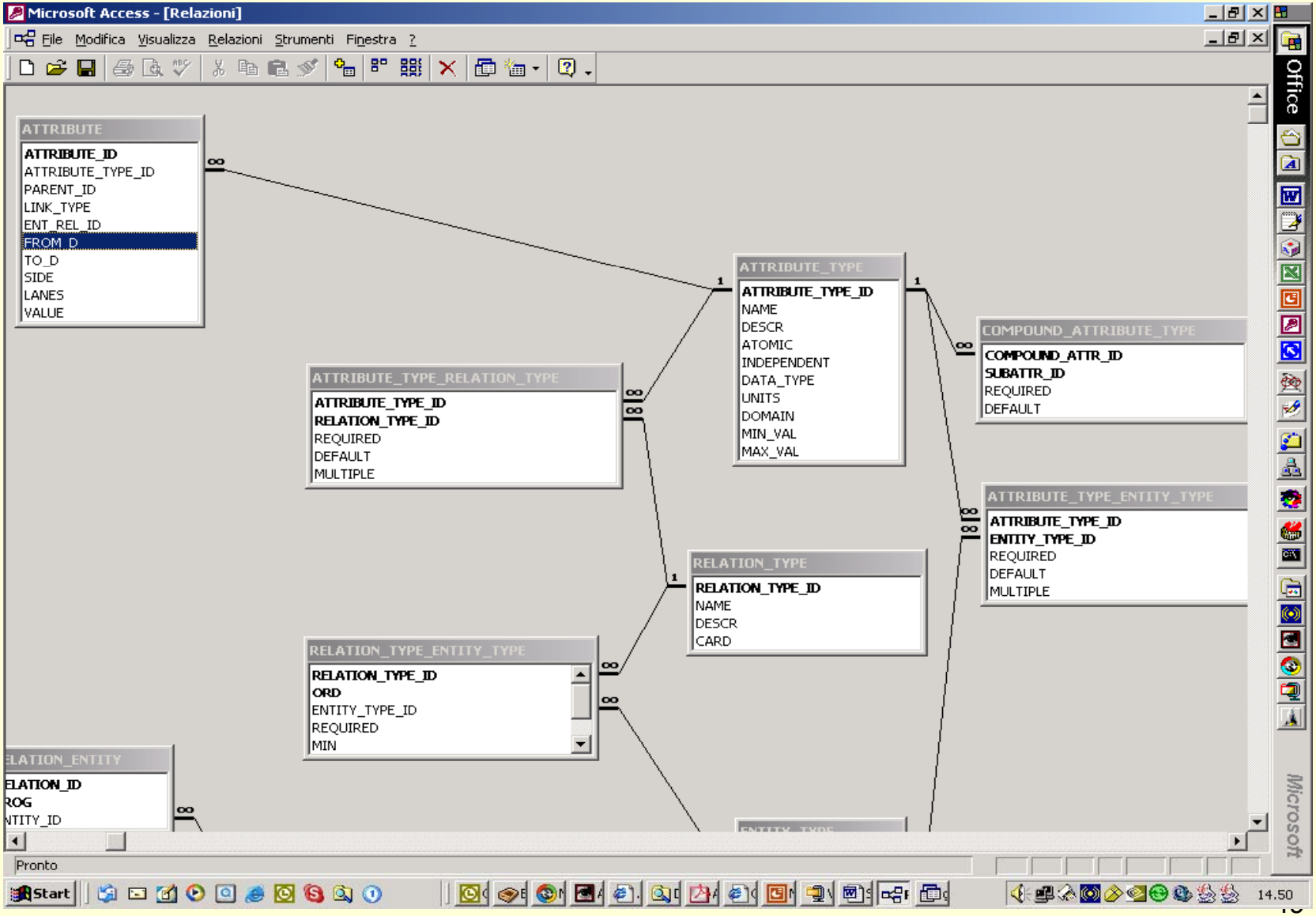
LEVEL 0



META-ENTITY DEFINITION



GDF META-SCHEMA

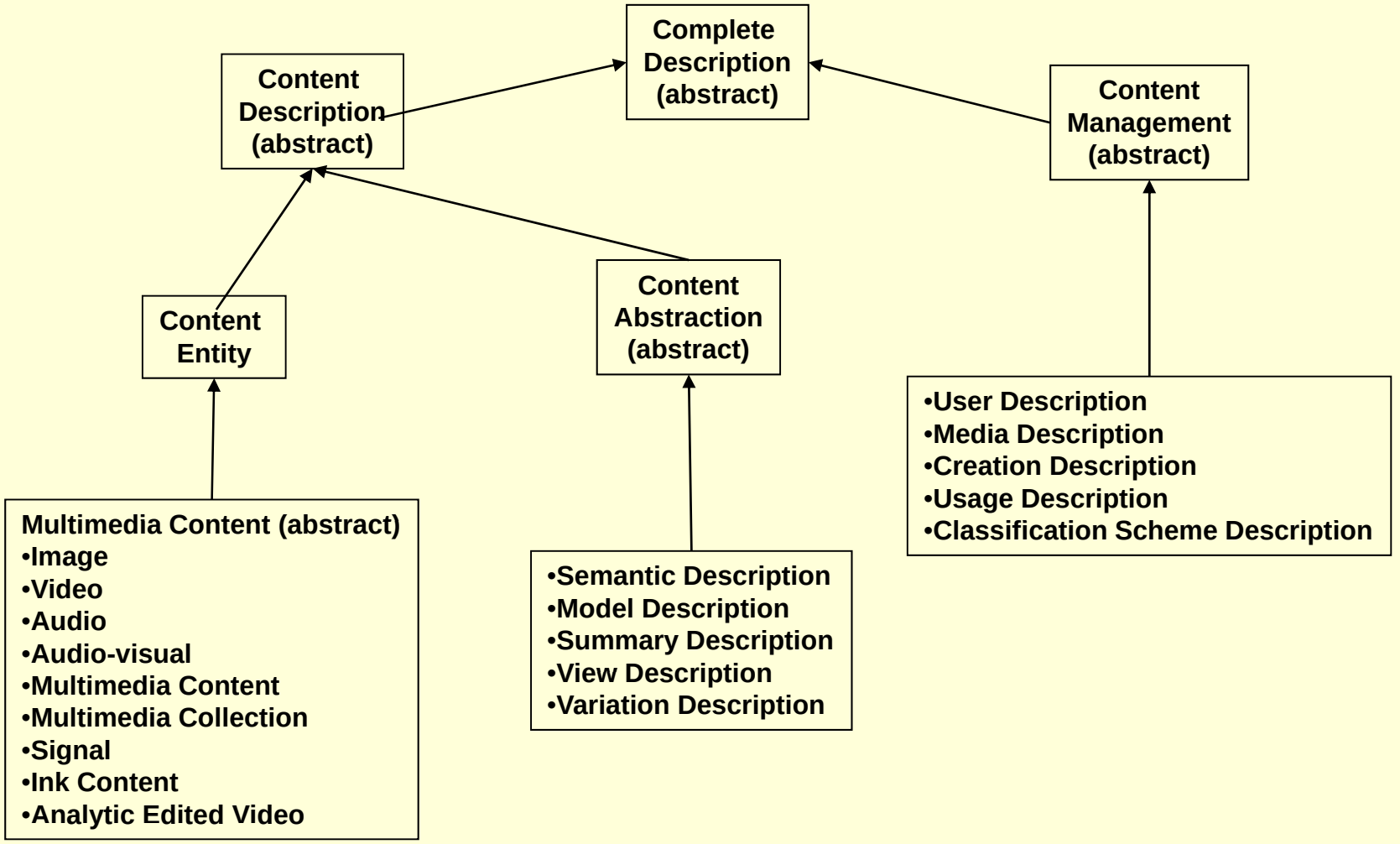


AN ORTHOGONAL APPROACH: A STANDARD FOR METADATA

MULTIMEDIA CONTENT DESCRIPTION INTERFACE (MPEG-7) http://www.tilab.com/mpeg

- PROVIDES A SET OF TOOLS FOR DESCRIBING MULTIMEDIA CONTENT
- XML-SCHEMA BASED
- NOT BASED ON SPECIFIC APPLICATION DOMAINS, IT ENABLES EASY EXCHANGE AND REUSE OF MULTIMEDIA CONTENTS
  - SPEECH → TEXT
  - IMAGES ←→ TEXT
- IT IS APPLIED IN REAL-TIME AS WELL AS IN NON-REAL-TIME SITUATIONS:
  - OFF-LINE CONTENT STORAGE
  - ON-LINE CONTENT STORAGE
  - STREAM (NO PERMANENT STORAGE)

MPEG-7 TOP LEVEL ELEMENTS



Da: Martinez, Koenen, Pereira