Data Manipulation and Integration in XML
or
From F-Logic to XPathLog

Wolfgang May
Universität Freiburg, Germany

Seminar “Rule Markup Techniques”
Dagstuhl, 4.2.–8.2.2002

rev. 5.2.2002

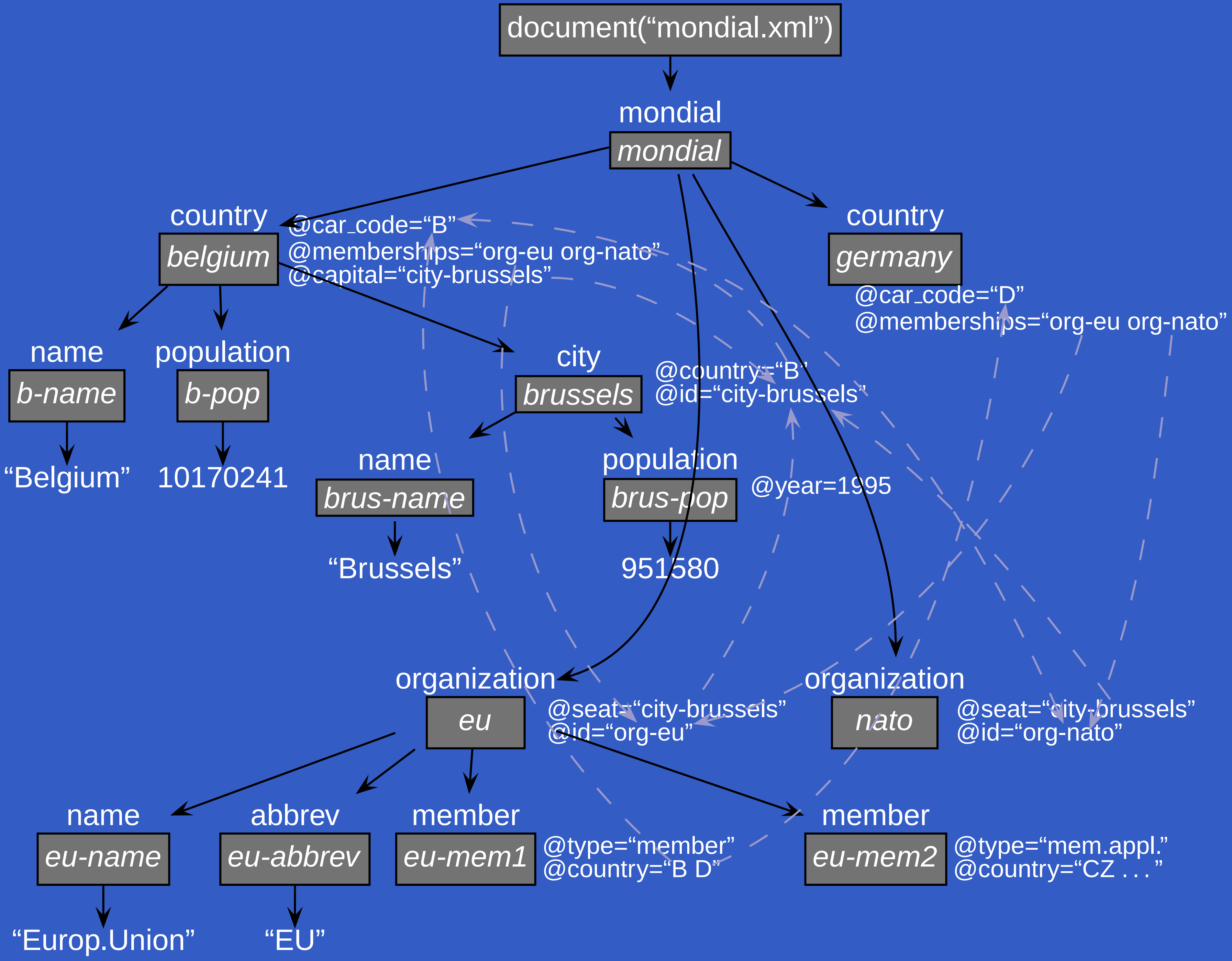
Project Overview

- Project Background
  - Experiences in deductive, object-oriented database languages: F-Logic
  - F-Logic as an early semi-structured/self-describing data model
  - integration of semi-structured data in F-Logic/FLORID
- XML: Internet-wide data format,
  distributed, autonomous sources
- our focus:
  - database applications
  - solid formal foundation of
    - data model,
    - language for querying/manipulation/integration

Example: Mondial

```
<mondial>
  <country car_code=“B”
    capital=“cty-Brussels”
    memberships=“org-eu org-nato ..”>
    <name>Belgium</name>
    <population>
      10170241
    </population>
    <city id=“cty-Brussels”
      country=“B”>
      <name>Brussels</name>
      <population year=“95”>
        951580
      </population>
    </city>
    :
  </country>
```

```
  <organization id=“org-eu”
      seat=“cty-Brussels”>
    <name>Europ. Union</name>
    <abbrev>EU</abbrev>
    <members type=“member”
      country=“GR F E A D I B L . . . ”/>
    <members type=“applicant”
      country=“AL CZ . . . ”/>
  </organization>
  <organization id=“org-nato”
    seat=“cty-Brussels” . . . >
    :
  </organization>
  :
  :
  :
</mondial>
```



document(“mondial.xml”)

mondial

mondial

country

belgium

@car_code=“B”

@memberships=“org-eu org-nato”

@capital=“city-brussels”

country

germany

@car_code=“D”

@memberships=“org-eu org-nato”

name

b-name

“Belgium”

population

b-pop

10170241

city

brussels

@country=“B”

@id=“city-brussels”

name

brus-name

“Brussels”

population

brus-pop

@year=1995

951580

organization

eu

@seat=“city-brussels”

@id=“org-eu”

organization

nato

@seat=“city-brussels”

@id=“org-nato”

name

eu-name

“Europ.Union”

abbrev

eu-abbrev

“EU”

member

eu-mem1

@type=“member”

@country=“B D”

member

eu-mem2

@type=“mem.appl.”

@country=“CZ . . . ”

F-Logic

Frame-based Data Model

mondial[country↠{*belgium, germany, . . .* };
organization↠{*un, eu, nato, . . .* }].
belgium:country[name→“Belgium”; capital→*brussels*;
city↠{*brussels, antwerp, . . .* };
memberships↠{*eu, nato, . . .* }].
brussels:city[name→“Brussels”; country→*belgium*;
population@(95)→951580].
eu:organization[abbrev→“EU”; seat→*brussels*;
members↠{*belgium, germany, . . .* }].

Navigation-based query language

?- *mondial*..organization[abbrev→ON].seat.country[name→CN].

LP-style data manipulation language

Outline

... high expectations what a language should be able to do

- Analysis of general problems and concepts
  - Query languages: addressing, compound queries
  - Data manipulation
  - Data integration
  - Data model
- design of an own framework:
  Non-W3C data model, W3C-language constructs as base
- Implementation: LoPiX
- Perspectives

XML & Friends: State of the Art

- Two querying languages:
  W3C XPath + XQuery vs. XML-QL
- W3C XML Query Requirements/Data Model/Formal Semantics
- Proposals for Update Language Extensions:
  - [A.Halevy@U.Washington] “Updating XML” @ SIGMOD 2001 (for XQuery)
  - [Software AG] XQuery + Updates in QuiP

Addressing: XPath

- Navigation expressions
- Output: “Result set” consisting of XML nodes

/mondial/country/name

/mondial/country/@car_code

/mondial/country[population $>$ 5000000]//city/name/text(),

//city[population[@year $<$ 1990] $>$ 5000000]/name/text()

/mondial/organization[name=“EU”]/@seat$\Rightarrow$city

Querying: XQuery

- influenced by
  - SQL (SFW-clauses, variable bindings)
  - XPath (addressing)
  - XSLT and XML-QL (generation of the result)

FOR *variable* IN *xpath-expr*
LET *additional_variable* := *xpath-expr*
WHERE *filters*
RETURN *xml-expr*

Updates in XML

Generic proposal in [TIHW01]

- *e*.Delete(*member*)
- *e*.Rename(*member*,*name*)
- *e*.Insert(*content*)
- *e*.Replace(*member*, *content*)
- *content*: variable or XML pattern

FOR *variable* IN *xpath-expr*
LET *additional_variable* := *xpath-expr*
WHERE *filters*
apply update method to variable

- if *content* is or contains an already existing XML element?

Data Integration

- Databases: Graph, unordered semantical integration:
  - elements in different sources represent the same object

  ⇒ element/object“fusion”
  - synonyms
  - not compatible with the XML data model

Design Decisions

Data Model: XTreeGraph

- extends the DOM/XML Query Data Model
- database is not a tree, but a graph consisting of overlapping trees
- “crossbreed” between F-Logic and XML Data Model:

Elements	⇔	Objects/Frames
Subelements, Attributes	⇔	Properties/Slots
XTreeGraph/X-Structure	⇔	F-structure

- supports updates and integration operations
- results: XML tree views over this graph

Design Decisions

Language: XPath-Logic and XPathLog

- crossbreed between XPath and F-Logic:
  extend XPath with variable bindings
- declarative rule-based language with bottom-up semantics
  - XPathLog is the Horn fragment of XPath-Logic
  - constructive semantics of XPath expressions in rule heads

XPathLog by Example

- Pure XPath expressions

```
?- //country[name/text() = “Belgium”]//city/name/text().
true
```

- Output result set

```
?- //country[name/text() = “Belgium”]//city/name/text()→N.
N/“Brussels”
N/“Antwerp”
⋮
```

XPathLog by Example

- Additional variables

  ?- //country[name/text()→N1 and
  @car_code→C]//city/name/text()→N2.

  N1/“Belgium”   C/“B”   N2/“Brussels”

  N1/“Belgium”   C/“B”   N2/“Antwerp”

  ⋮

- Dereferencing

  ?- //organization→O[@seat[name/text()→N] =
  members/@country/@capital]

  O/*eu*   N/“Brussels”

  ⋮

XPathLog by Example

- Metadata: navigation variables

  ?- //Type→X[name/text()→“Monaco”].

Type/country	X/*country-monaco*
Type/city	X/*city-monaco*

- Metadata: schema queries

  ?- //city/N.

  N/name

  N/population

  ⋮

Semantics of XPathLog Queries

- Extends the well-known XPath semantics:
  - Result set + variable bindings
- induces algebraic evaluation strategy adaptation of the *Object Algebra* of F-Logic (navigation and filters)

Semantics of XPathLog Queries

- Extends the well-known XPath semantics:
  - Result set + variable bindings
- induces algebraic evaluation strategy
  adaptation of the *Object Algebra* of F-Logic
  (navigation and filters)

- XPathLog is the Horn fragment of XPath-Logic

$$\text{head}(V_1,\ldots,V_n)\text{ :- }\text{body}(V_1,\ldots,V_n)$$

Rule Heads

Constructive semantics of definite XPathLog atoms:

- only *child*, *sibling* and *attribute*-Axis
- no negation, function applications, aggregation, and *proximity position predicates*

“/” and “[. . .]” as Constructors:

- *host*[*property*→*value*] modifies *host*
- *host*/*property remainder*
  generates a new element *host/property*, that satisfies *remainder*
- *property* is *axis::name* or *axis(i)::name*

... again similar as for F-Logic

Rule Heads: Attributes

C[@datacode→“be”], C[@memberships→O] :-
//country→C[@car_code=“B”],
//organization→O[abbrev/text()→“EFTA”].

⇓

<country $\boxed{\text{datacode=“be”}}$ car_code=“B”
memberships=“org-eu org-un $\boxed{\text{org-efta}}$ . . . ”>
⋮
</country>

Generation of “free” Elements

/country[@car_code→“BAV”].

⇓

<country car_code=“BAV”> </country>

Generation of Elements

C/name[text()→“Bavaria”] :- //country→C[@car_code=“BAV”].

⇓

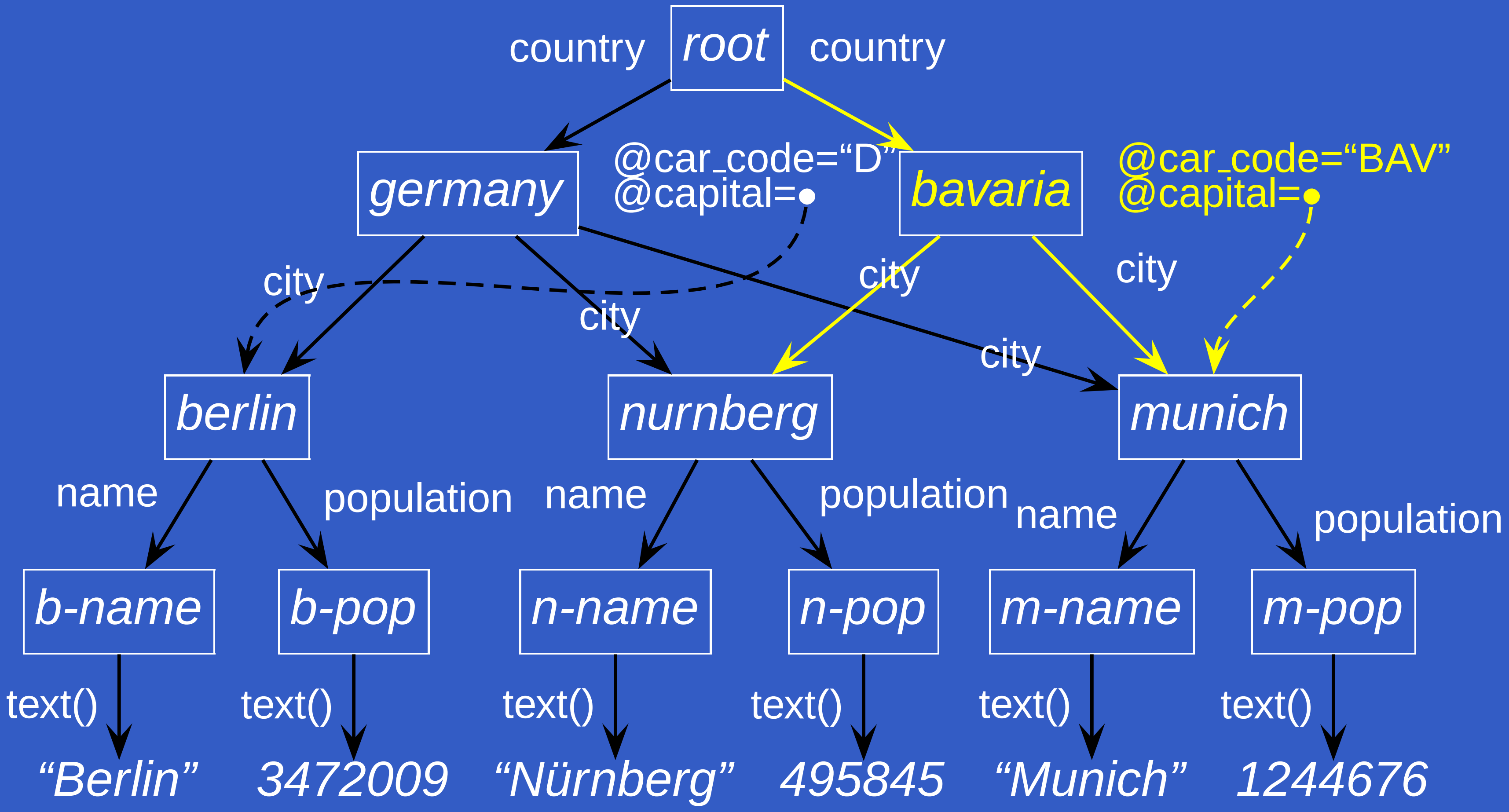
```
<country car_code=“BAV” capital=“city-munich”>
    <city>. . . </city>
    <city>. . . </city>
    <name>Bavaria</name>
</country>
```

Adding Subelement Relationships

```
C[@capital→X and city→X and city→Y] :-
    //country→C[@car_code→“BAV”],
    //city→X[name/text()=“Munich”],
    //city→Y[name/text()=“Nurnberg”].
```

- city elements are linked as subelements
- efficient *in-place* restructuring and integration

Linking



- Elements have multiple parents

Extensions

- class hierarchy
  with nonmonotonic inheritance
- signatures

  country[@car_code⇒string].

  country[@area⇒numeric].

  country[@capital⇒city].

  country[city⇒city].

  - derivable from DTD or XML Schema
  - serves for definition of views/projections

Integration

“Three-level”-Model

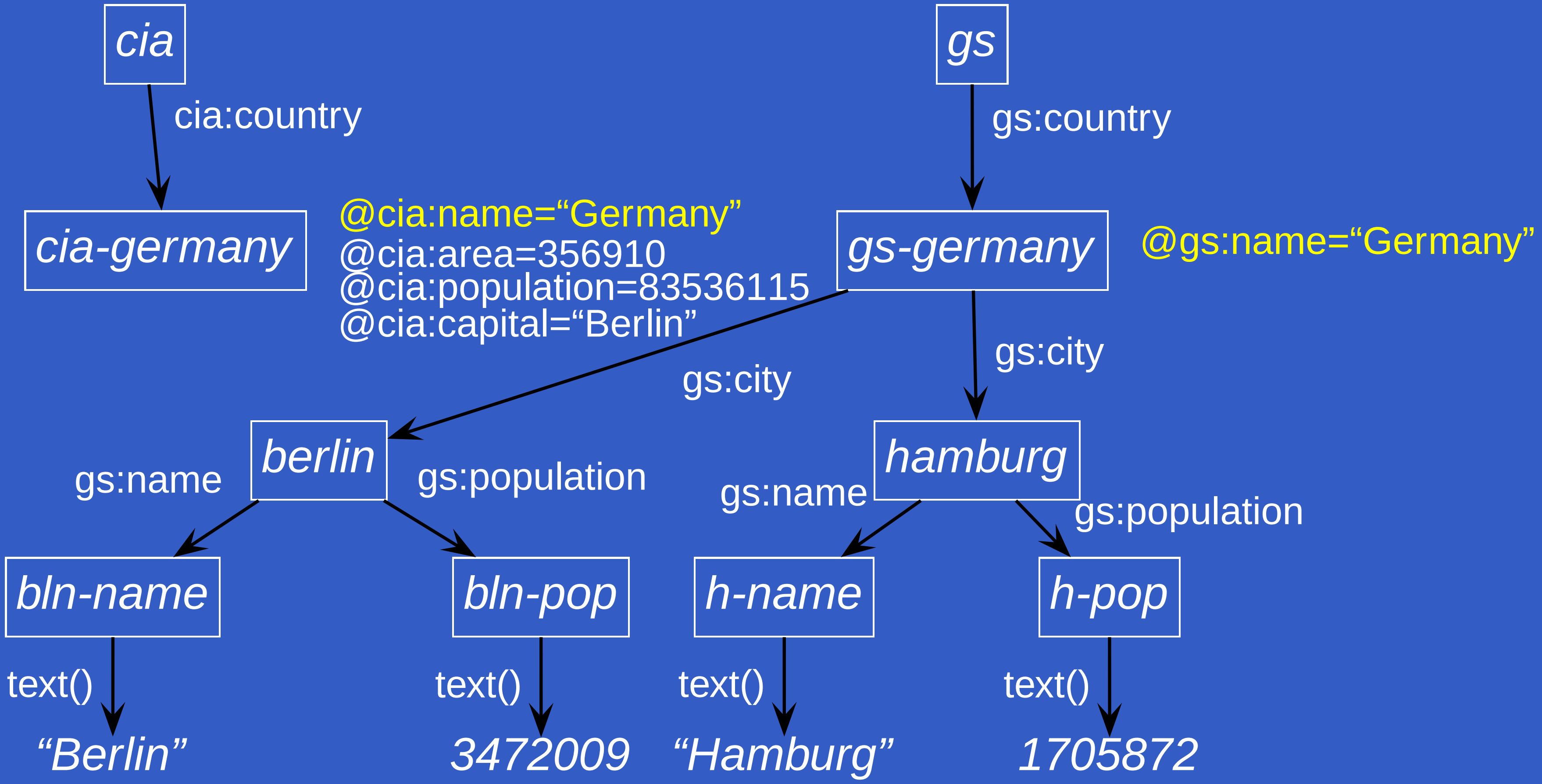
access multiple sources:

- “basic” layer: source(s) provide tree structures,
- optionally with namespaces

Data Integration

Data Sources describing countries:

- cia: name, area, population and capital (by name)
- gs: cities with name, population



Integration

“Three-level”-Model (2)

Merge data from different sources

- “internal” layer: XTreeGraph
  - overlapping trees
  - multiple parents
- fuse elements/merge subtrees
- add subelement links
- generate elements
- synonyms for properties

Synonyms

$$\mathit{namespace{:}name}_1 = \mathit{name}_2$$

```
cia:name = name.          gs:name = name.
cia:area = area.          cia:population = population.
cia:text() = text().      gs:text() = text().
```

- does not generate new element or attribute nodes,
- but “only” additional navigation paths
- order-preserving

Element Fusion

- elements represent the same real-world entity in different sources
- fuse elements into a unified element: $e_1 = e_2$

Resulting element

1. is then an element of *both* source trees
2. collects the attributes of both original elements
3. collects the subelements of both original elements

Element Fusion: Example

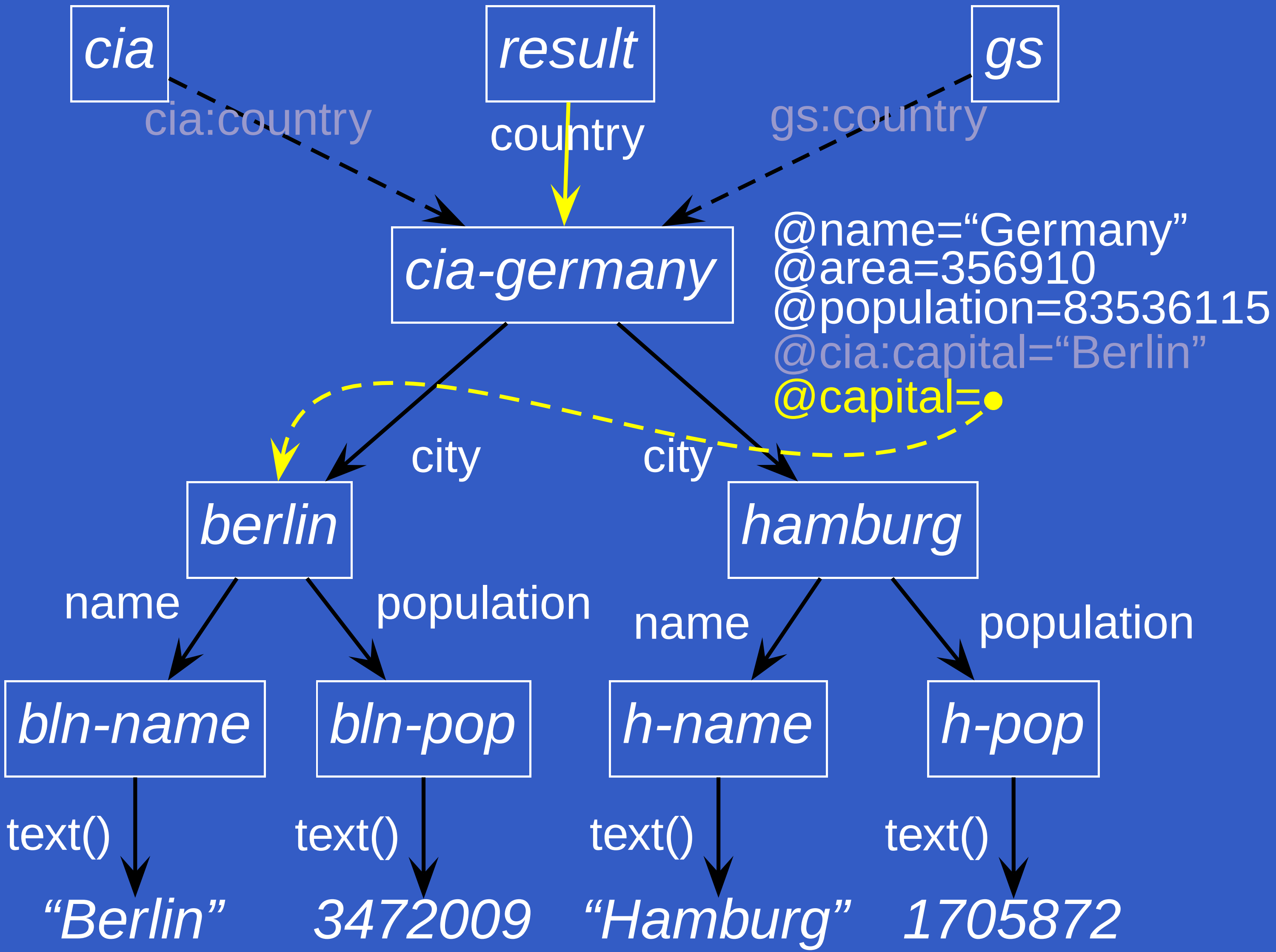
result[country→C1],

C1 = C2 :- *cia*/cia:country[@name→N]→C1,
gs/gs:country[@name→N]→C2.

C[@capital→Cap] :-

result/country→C[@cia:capital→N and
city→Cap[name/text()→N]].

Element Fusion: Example



C[@capital→City] :-

Integration

“Three-level”-Model (3)

Definition of Results:

- “export” layer: XML result trees as views defined by
  - root
  - signature

    mondial[country⇒country].

    country[@name⇒string].

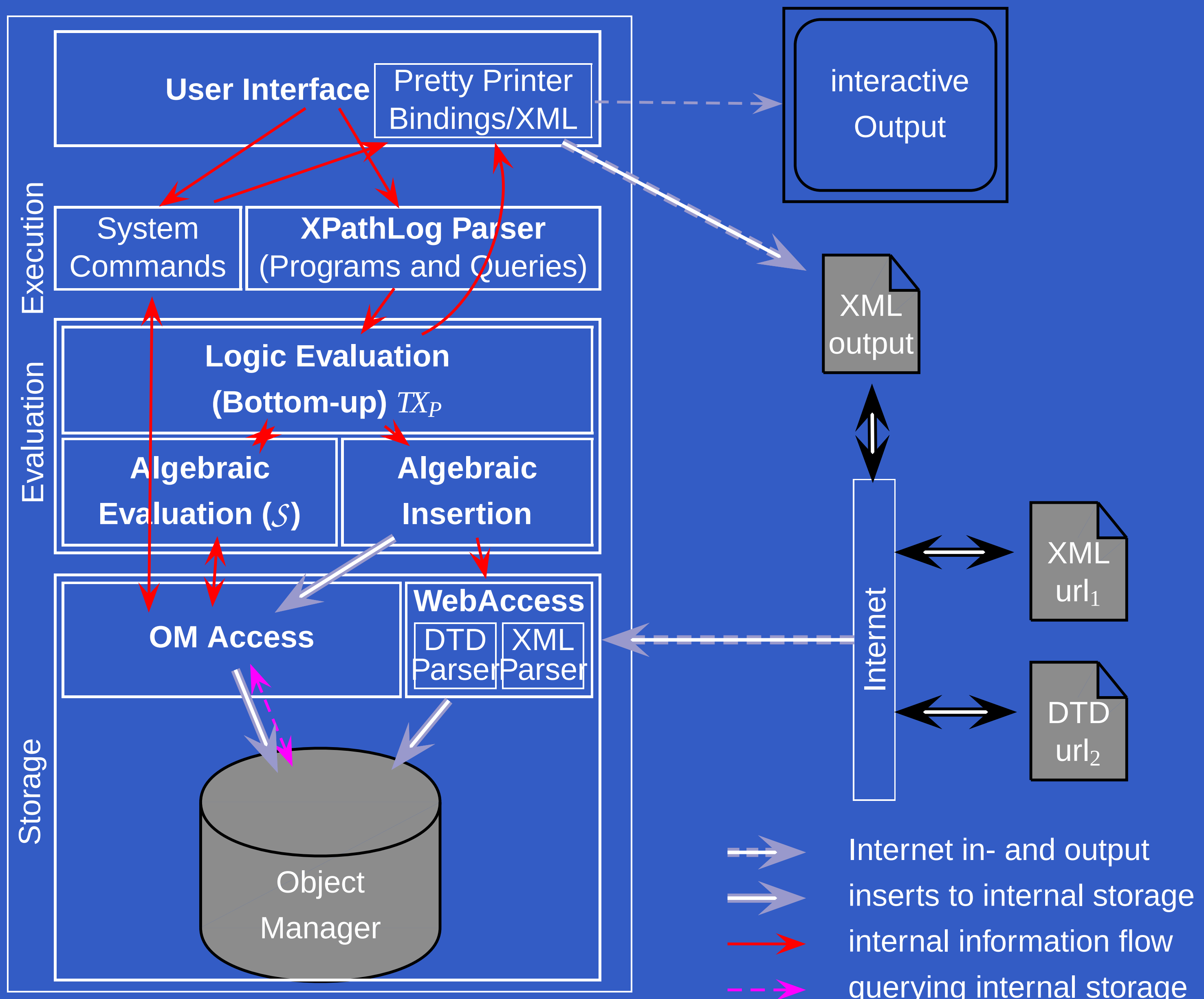
    country[@area⇒numeric].

    country[@population⇒numeric].

    country[@capital⇒city].

    ⋮

Implementation: LoPiX



Execution

Evaluation

Storage

User Interface

Pretty Printer
Bindings/XML

System
Commands

XPathLog Parser
(Programs and Queries)

Logic Evaluation
(Bottom-up) T_{X_P}

Algebraic
Evaluation ($\mathcal{S}$)

Algebraic
Insertion

OM Access

WebAccess

DTD
Parser

XML
Parser

Object
Manager

interactive
Output

XML
output

Internet

XML
url_1

DTD
url_2

Internet in- and output

inserts to internal storage

internal information flow

querying internal storage

Experiences & Conclusion

Case-study “Mondial”

- XPathLog + XTreeGraph
  - powerful, expressive language
  - important: linking, fusing, synonyms
- pure XPathLog: XML q/m/i-language
- full XPathLog with F-Logic features: useful as internal language for powerful reasoning systems
  - use of ontologies ...
  - represent a knowledge base/database as an XTreeGraph
  - changes to the knowledge base result in adaptations of the rule system

Questions ??

- LoPiX:
  `www.informatik.uni-freiburg.de/~may/lopix`
- Mondial:
  `www.informatik.uni-freiburg.de/~may/mondial`