# LoPiX: Logic Programming in XML

Databases & Information Systems Group · Institut für Informatik · Universität Freiburg · Germany
http://www.informatik.uni-freiburg.de/~may/lopix · contact: may@informatik.uni-freiburg.de

## Project Overview

- Experiences with F-Logic/Florid in integration of semistructured data [Flo98, Flo99]

### Goals

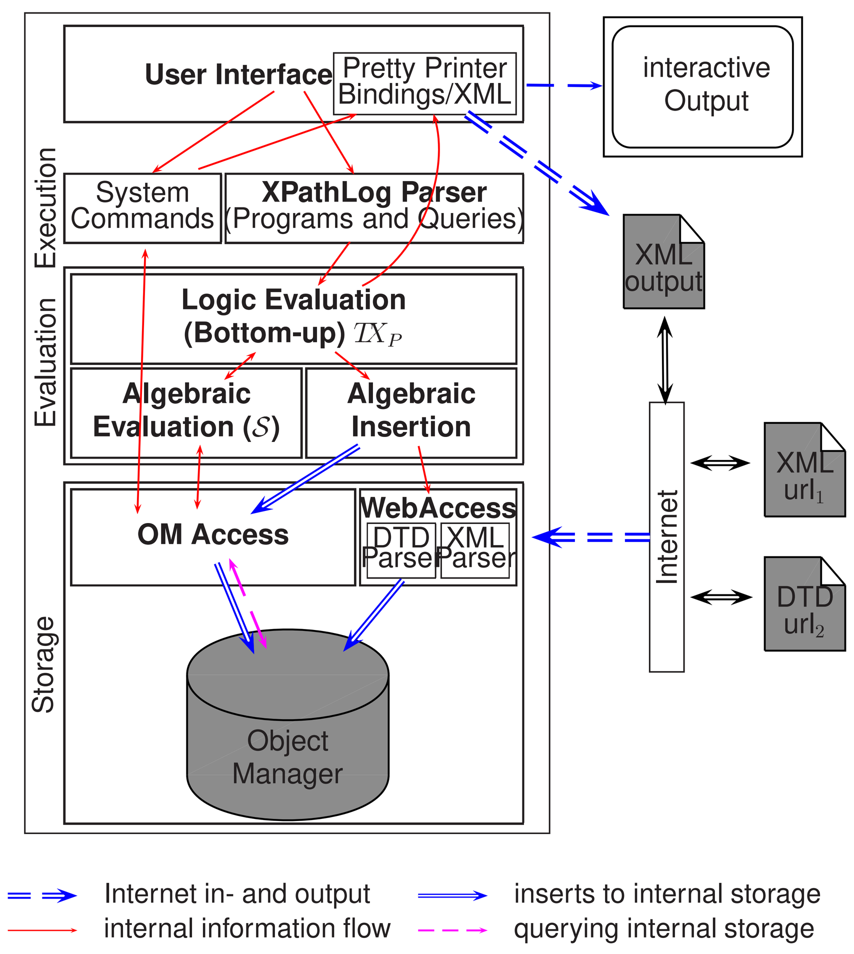
- design & implement a programming language for XML: querying, manipulation, *integration*

### Results

- analysis of XML, its languages, and data model(s)
- design own framework ...
- ... end up with a non-W3C data model and using basic W3C language elements:
- extend XPath
- XPathLog:
  declarative rule-based language with bottom-up semantics
- XTreeGraph:
  graph model, overlapping trees, multiple parents, tailored to XML Data Integration
- Implementation: LoPiX
- Case Study: Mondial[Mon01]
- The complete report: [Lop01a]

## System Architecture

- object-oriented modular design
- exchangeable storage components




## History

- since 1988: F-Logic, an early “semistructured” data model with a declarative rule-based language (Kifer, Lausen SIGMOD’89 and Kifer, Lausen, Wu JACM’95)
- 1992-1996: Design and implementation of the research prototype Florid 1.0; language extensions and optimizations.
- 1997-1999: research on mapping HTML data to F-Logic,
  Florid2.0 as a Web-enabled version; Data integration from the Web [Flo98, Flo99]
- 1999/2000: application to XML data (Florid 3.0/FloXML)
- 2000/2001: migration to “native” XML, design of XPathLog, implementation of LoPiX based on Florid3.0.
  LoPiX still provides features known from F-Logic such as class hierarchy, signatures, and non-monotonic inheritance as extensions to XML data.

Thanks to E. Behrends, J. Frohn, R. Himmeröder, P.-Th. Kandzia, G. Lausen, B. Ludäscher, D. Malheiro, P. J. Marrón, T. Pferdekämper, C. Schlepphorst, M. Seilnacht, H. Uphoff, and T. Westmann.

## Publications

[Lop01a] W. May. XPath-Logic and XPathLog: A Logic-Based Approach for Declarative XML Data Manipulation, 2001. Available from `http://www.informatik.uni-freiburg.de/~may/lopix/`.

[Mon01] W. May. Information Integration in XML: The Mondial Case Study, 2001. `http://www.informatik.uni-freiburg.de/~may/lopix/lopix-mondial.html`.

[XPL01a] W. May. A Rule-based Querying and Updating Language for XML. *Workshop on Databases and Programming Languages (DBPL 2001)*.

[XPL01b] W. May and E. Behrends. On an XML Data Model for Data Integration. *Workshop on Foundations of Models and Languages for Data and Objects (FMLDO 2001)*.

[XPL01c] W. May. Integration of XML Data in XPathLog. *CAiSE Workshop Data Integration over the Web (DIWeb’01)*, 2001.

[XPL01d] W. May. A Framework for Generic Integration of XML Sources. *Workshop on Knowledge Representation meets Databases (KRDB 2001)*.

[Flo98] B. Ludäscher, R. Himmeröder, G. Lausen, W. May, and C. Schlepphorst. Managing Semistructured Data with Florid: A Deductive Object-Oriented Perspective. *Information Systems*, 23(8):589–612, 1998.

[Flo99] W. May, R. Himmeröder, G. Lausen, and B. Ludäscher. A Unified Framework for Wrapping, Mediating and Restructuring Information from the Web. *Workshop on the WWW and Conceptual Modeling (WWWCM)*, Springer LNCS 1727, 1999.

## Language: XPathLog

for details see [XPL01a]

- Extends the XPath syntax with *variable Bindings*
- Declarative, rule-based language with bottom-up semantics
  head($V_1,\ldots,V_n$) :- body($V_1,\ldots,V_n$)
- Queries: extends the semantics for XPath given by P.Wadler (1999)
- Constructive semantics for XPathLog atoms in rule heads

### XPath-Logic *reference expressions* are XPath *location paths*

```
[0]   ReferenceExpr ::= AbsLocPath | ConstLocPath
[2b]  ConstLocPath ::=  constant "/" RelLocPath
                      | variable "/" RelLocPath
```

### Extend *LocationSteps*

```
[4] Step ::= AxisSpec NodeTest Pred*
           | AxisSpec NodeTest Pred* "->" Var Pred*
           | AxisSpec Var Pred*
           | AxisSpec Var Pred* "->" Var Pred*
```

- navigation by dereferencing `IDREF` attributes
- Predicates over reference expressions

### Rule Heads: definite XPathLog atoms

- use only the child and sibling axes
- no negation, disjunction, function applications, and *proximity position predicates*

## XPathLog: Examples

### Pure XPath expressions

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

### Output Result Set

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

### Additional Variables

?- //country[name/text()→N1 and @car_code→C]//city/name/text()→N2.
N2/“Brussels” C/“B” N1/“Belgium”
:

### Local Variables

?- //country[name/text()→N1]//city[population/text()→_P] /name/text()→N2, _P > 100000.

### Dereferencing

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

### Navigation Variables

?- //Type→X[name/text()→“Monaco”].
Type/country X/*country-monaco*
Type/city X/*city-monaco*

### Schema Querying

?- //city/N.
N/name
N/population
:

## XML Information Integration

for details see [XPL01c, XPL01d]

- integration of databases: semantics-driven integration process
- objects of different sources represent the same real-world object

⇒ Fusing objects, merging their properties

- synonyms, ontologies
- not compatible with XML Data Models (DOM, XML Query Data Model)

## Data Model: XTreeGraph

for details see [XPL01b]

- Graph data model, extends DOM/XML Query Data Model
- Supports updates
- Tailored to data integration

⇒ multiple, overlapping trees

## Three-Level Integration

- multiple sources which provide tree structures (basic layer)
- merge data from different sources; “internal” layer: XTreeGraph
  - fuse elements/merge subtrees ⇝ overlapping trees
  - add subelement links ⇝ overlapping trees
  - define synonyms for properties
- “export” layer: define result tree views by projections

## Case-Study in Data Integration: Mondial

### CIA World Factbook Organizations

- organizations
- memberships

```
<!ELEMENT orgs (organization*)>
<!ELEMENT organization
   (member_names*)>
<!ATTLIST organization
   abbrev CDATA #REQUIRED
   name CDATA #REQUIRED
   established CDATA #IMPLIED
   seatcountry CDATA #IMPLIED
   seatcity CDATA #IMPLIED>
<!ELEMENT member_names
   (#PCDATA)>
<!ATTLIST member_names
   type CDATA #REQUIRED>
```

### CIA World Factbook Countries

- continents
- countries

```
<!ELEMENT cia
   (continent*, country*)>
<!ELEMENT continent EMPTY>
<!ATTLIST continent
   name CDATA #REQUIRED>
<!ELEMENT country
   (ethnicgroups*, religions*,
    languages*, borders*)>
<!ATTLIST country
   name CDATA #REQUIRED
   continent CDATA #IMPLIED
   total_area CDATA #IMPLIED
   population CDATA #IMPLIED
      :
   capital CDATA #IMPLIED>
<!ELEMENT religions (#PCDATA)>
<!ATTLIST religions
   name CDATA #REQUIRED>
(analogous ethnicgroups and
 languages)
<!ELEMENT borders (#PCDATA)>
<!ATTLIST borders
   country IDREF #REQUIRED>
```

### Global Statistics

- countries, continents
- administrative divisions, cities

```
<!ELEMENT gs (continent*, country*,
   city*, province*)>
<!ELEMENT continent EMPTY>
<!ATTLIST continent
   name CDATA #REQUIRED>
<!ELEMENT country (name+)>
<!ATTLIST country
   id ID #REQUIRED
   capital IDREF #IMPLIED
   population CDATA #IMPLIED
   continent IDREF #IMPLIED
   main_cities IDREFS #IMPLIED
   adm_divs IDREFS #IMPLIED>
<!ELEMENT province EMPTY>
<!ATTLIST province
   id ID #REQUIRED
   name CDATA #REQUIRED
   country IDREF #REQUIRED
   capital IDREF #IMPLIED
   population CDATA #IMPLIED
   area CDATA #IMPLIED>
<!ELEMENT city
   (population*, name+)>
<!ATTLIST city
   id ID #REQUIRED
   country IDREF #REQUIRED
   province IDREF #IMPLIED>
<!ELEMENT population (#PCDATA)>
<!ATTLIST population
   year CDATA #REQUIRED>
```

### Auxiliary

- country codes
- german and english names

```
<!ELEMENT codes (country*)>
<!ELEMENT country
   (name,name,name)>
<!ATTLIST country
   id CDATA #REQUIRED
   car_code CDATA #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ATTLIST name
   language CDATA #REQUIRED>
```

### TERRA

- uses German names
- countries, administrative divisions, cities
- mountains, islands, waters

```
<!ELEMENT terra
   (country*, province*, city*,
    (mountain, desert, island,
     river, lake, sea)*)>
<!ELEMENT country
(encompassed*)>
<!ATTLIST country
   id ID #REQUIRED
   name CDATA #REQUIRED
   code CDATA #REQUIRED
   area CDATA #REQUIRED
   population CDATA
      #REQUIRED
   capital CDATA #REQUIRED>
<!ELEMENT encompassed
   (#PCDATA)>
<!ATTLIST encompassed
   continent CDATA
     #REQUIRED>
```

```
<!ELEMENT province (#PCDATA)>
<!ATTLIST province
   id ID #IMPLIED
   name CDATA #IMPLIED
   abbrev CDATA #IMPLIED
   country CDATA #IMPLIED
   pop CDATA #IMPLIED
   capital CDATA #IMPLIED>
<!ELEMENT city (province*)>
<!ATTLIST city
   name CDATA #REQUIRED
   country CDATA #REQUIRED
   population CDATA #IMPLIED
   longitude CDATA #IMPLIED
   latitude CDATA #IMPLIED>
<!ELEMENT river (located*)>
(similar: lake, river, sea,
 desert, island, mountain)
<!ELEMENT located EMPTY>
<!ATTLIST located
  country_code CDATA #REQUIRED
  province_id CDATA #REQUIRED>
```

- The CIA World Factbook XML database is derived from the data provided by the CIA at `http://www.odci.gov/cia/publications/pubs.html`.
- The GlobalStatistics XML database is derived from the *Global Statistics* Data at `http://www.stats.demon.nl` collected by Johan van der Heijden.
- The TERRA XML database is derived from the TERRA database of the *Institut für Programmstrukturen und Datenorganisation der Universität Karlsruhe*.

VLDB 2001 Demonstration Track, Rome, 11.-15.9.2001

# LoPiX: Logic Programming in XML
# Case-Study in Data Integration: Mondial

## The Source DTDs

### CIA World Factbook Organizations

- organizations
- memberships

```
<!ELEMENT orgs (organization*)>
<!ELEMENT organization
   (member_names*)>
<!ATTLIST organization
   abbrev CDATA #REQUIRED
   name CDATA #REQUIRED
   established CDATA #IMPLIED
   seatcountry CDATA #IMPLIED
   seatcity CDATA #IMPLIED>
<!ELEMENT member_names
   (#PCDATA)>
<!ATTLIST member_names
   type CDATA #REQUIRED>
```

### CIA World Factbook Countries

- continents
- countries

```
<!ELEMENT cia
   (continent*, country*)>
<!ELEMENT continent EMPTY>
<!ATTLIST continent
   name CDATA #REQUIRED>
<!ELEMENT country
   (ethnicgroups*, religions*,
    languages*, borders*)>
<!ATTLIST country
   name CDATA #REQUIRED
   continent CDATA #IMPLIED
   total_area CDATA #IMPLIED
   population CDATA #IMPLIED
        :
   capital CDATA #IMPLIED>
<!ELEMENT religions (#PCDATA)>
<!ATTLIST religions
   name CDATA #REQUIRED>
(analogous ethnicgroups and
 languages)
<!ELEMENT borders (#PCDATA)>
<!ATTLIST borders
   country IDREF #REQUIRED>
```

### Global Statistics

- countries, continents
- administrative divisions, cities

```
<!ELEMENT gs (continent*, country*,
   city*, province*)>
<!ELEMENT continent EMPTY>
<!ATTLIST continent
   name CDATA #REQUIRED>
<!ELEMENT country (name+)>
<!ATTLIST country
   id ID #REQUIRED
   capital IDREF #IMPLIED
   population CDATA #IMPLIED
   continent IDREF #IMPLIED
   main_cities IDREFS #IMPLIED
   adm_divs IDREFS #IMPLIED>
<!ELEMENT province EMPTY>
<!ATTLIST province
   id ID #REQUIRED
   name CDATA #REQUIRED
   country IDREF #REQUIRED
   capital IDREF #IMPLIED
   population CDATA #IMPLIED
   area CDATA #IMPLIED>
<!ELEMENT city
   (population*, name+)>
<!ATTLIST city
   id ID #REQUIRED
   country IDREF #REQUIRED
   province IDREF #IMPLIED>
<!ELEMENT population (#PCDATA)>
<!ATTLIST population
   year CDATA #REQUIRED>
```

### Auxiliary

- country codes
- german and english names

```
<!ELEMENT codes (country*)>
<!ELEMENT country
   (name,name,name)>
<!ATTLIST country
   id CDATA #REQUIRED
   car_code CDATA #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ATTLIST name
   language CDATA #REQUIRED>
```

### TERRA

- uses German names
- countries, administrative divisions, cities
- mountains, islands, waters

```
<!ELEMENT terra
   (country*, province*, city*,
    (mountain, desert, island,
     river, lake, sea)*)>
<!ELEMENT country
(encompassed*)>
<!ATTLIST country
   id ID #REQUIRED
   name CDATA #REQUIRED
   code CDATA #REQUIRED
   area CDATA #REQUIRED
   population CDATA
      #REQUIRED
   capital CDATA #REQUIRED>
<!ELEMENT encompassed
   (#PCDATA)>
<!ATTLIST encompassed
   continent CDATA
      #REQUIRED>
<!ELEMENT province (#PCDATA)>
<!ATTLIST province
   id ID #IMPLIED
   name CDATA #IMPLIED
   abbrev CDATA #IMPLIED
   country CDATA #IMPLIED
   pop CDATA #IMPLIED
   capital CDATA #IMPLIED>
<!ELEMENT city (province*)>
<!ATTLIST city
   name CDATA #REQUIRED
   country CDATA #REQUIRED
   population CDATA #IMPLIED
   longitude CDATA #IMPLIED
   latitude CDATA #IMPLIED>
<!ELEMENT river (located*)>
(similar: lake, river, sea,
 desert, island, mountain)
<!ELEMENT located EMPTY>
<!ATTLIST located
   country_code CDATA #REQUIRED
   province_id CDATA #REQUIRED>
```

- The CIA World Factbook XML database is derived from the data provided by the CIA at `http://www.odci.gov/cia/publications/pubs.html`.
- The GlobalStatistics XML database is derived from the *Global Statistics* Data at `http://www.stats.demon.nl` collected by Johan van der Heijden.
- The TERRA XML database is derived from the TERRA database of the *Institut für Programmstrukturen und Datenorganisation der Universität Karlsruhe*.

## Excerpts of the Sources

Part I: CIA Organizations, CIA Countries and GlobalStatistics

### CIA World Factbook — Organizations

```
<organization
   abbrev="EU"
   name="European Union"
   established="07 02 1992"
   seatcity="Brussels"
   seatcountry="Belgium">
   <member_names type='member'>Belgium</>
   <member_names type='member'>Belgium</>
   <member_names type='member'>Germany</>
        :
</organization>
```

### CIA World Factbook — Countries

```
<continent id='europe'
   name='Europe'/>

<country id='cia-country-Germany'
   continent='Europe'
   name='Germany'
   total_area='356910'
   population='83536115'
   capital='Berlin'>
   <ethnicgroups name='German'<95.1</>
   <ethnicgroups name='Italians'<0.7</>
        :
   <religions name='Roman Catholic'<37</>
   <religions name='Protestant'<45</>
   <languages name='German'<100</>
   <borders country='cia-country-Austria'>784</>
   <borders country='cia-country-Belgium'>167</>
        :
</country>

<country id='cia-country-Belgium'
   continent='Europe'
   name='Belgium'
   total_area='30510'
   population='10170241'
   capital='Brussels'>
   <ethnicgroups name='Fleming'>55</>
   <ethnicgroups name='Walloon'>33</>
   <borders country='cia-country-France'>620</>
   <borders country='cia-country-Germany'>167</>
        :
</country>
```

### Global Statistics

```
<continent id='europe'
   name='Europe'/>

<country id='gs-country-ger'
   capital='cty-gs-country-ger-1'
   population='83536115'
   continent='europe'
   main_cities='cty-gs-country-ger-1
                  :
               cty-gs-country-ger-50
                  :
               cty-gs-country-ger-85'
   adm_divs='prov-gs-country-ger-1
                  :
             prov-gs-country-ger-10
                  :
             prov-gs-country-ger-16'
   <name>Germany</name>
</country>

<city id='city-gs-country-ger-1'
   country='gs-country-ger'
   province='prov-gs-country-ger-1'>
   <population year='95'>3472009</>
   <name>Berlin</name>
</city>
   :
<city id='city-gs-country-ger-50'
   country='gs-country-ger'
   province='prov-gs-country-ger-10'>
   <population year='95'>588482</>
   <name>Stuttgart</name>
</city>
   :

<province
   id='prov-gs-country-ger-1'
   name='Berlin'
   country='gs-country-ger'
   capital='cty-gs-country-ger-50'
   population='3472009'
   area='889'/>
   :
<province
   id='prov-gs-country-ger-10'
   name='Baden Wurttemberg'
   country='gs-country-ger'
   capital='cty-gs-country-ger-2'
   population='10272069'
   area='35742'/>

<country id='gs-country-bel'
   capital='cty-gs-country-bel-1'
   population='10170241'
   continent='europe'
   main_cities='cty-gs-country-bel-1
                cty-gs-country-bel-2
                  :
                cty-gs-country-bel-11'
   adm_divs='prov-gs-country-bel-1
             prov-gs-country-bel-2
                  :
             cty-gs-country-bel-9'
   <name>Belgium</name>
</country>

<city id='cty-gs-country-bel-1'
   country='gs-country-bel'
   province='prov-gs-country-bel-1'>
   <population year='95'>951580</>
   <name>Brussels</name>
</city>
<city id='cty-gs-country-bel-2'
   country='gs-country-bel'
   province='prov-gs-country-bel-2'>
   <population year='95'>459072</>
   <name>Antwerp</name>
</city>
   :

<province
   id='prov-gs-country-bel-1'
   name='Brabant'
   country='gs-country-bel'
   capital='cty-gs-country-bel-1'
   population='2253794'
   area='3358'/>
<province
   id='prov-gs-country-bel-2'
   name='Antwerp'
   country='gs-country-bel'
   capital='cty-gs-country-bel-2'
   population='1610695'
   area='2867'/>
   :
```

Explanation:

- double lines: to be fused
- lines: to be linked
- dashed lines: existing IDREFs

## First Integration Step

### The Program

Synonyms

```
gs:population = population.
gs:country = country.
gs:province = province.
gs:capital = capital.
gs:area = area.
gs:year = year.
gs:name = name.
cia:name = name.
cia:borders = border.
cia:ethnicgroups = ethnicgroups.
cia:religions = religions.
cia:languages = languages.
cia.population = population.
cia:total_area = total_area.
orgs:abbrev = abbrev.
orgs:name = name.
orgs:established = established.
```

Continents from GS

```
result[continent →C] :- gs/gs:continent→C.
```

Organizations from CIA

```
result[organization →O] :- orgs/orgs:organization→O.
```

Fusing CIA and GS Countries

```
result[country→C1], C1 = C2 :-
    cia/cia:country→C1[@cia:name→N],
    gs/gs:country→C2[gs:name/gs:text()→N].
C[@name→N] :- result/country→C/@cia:name→N.
```

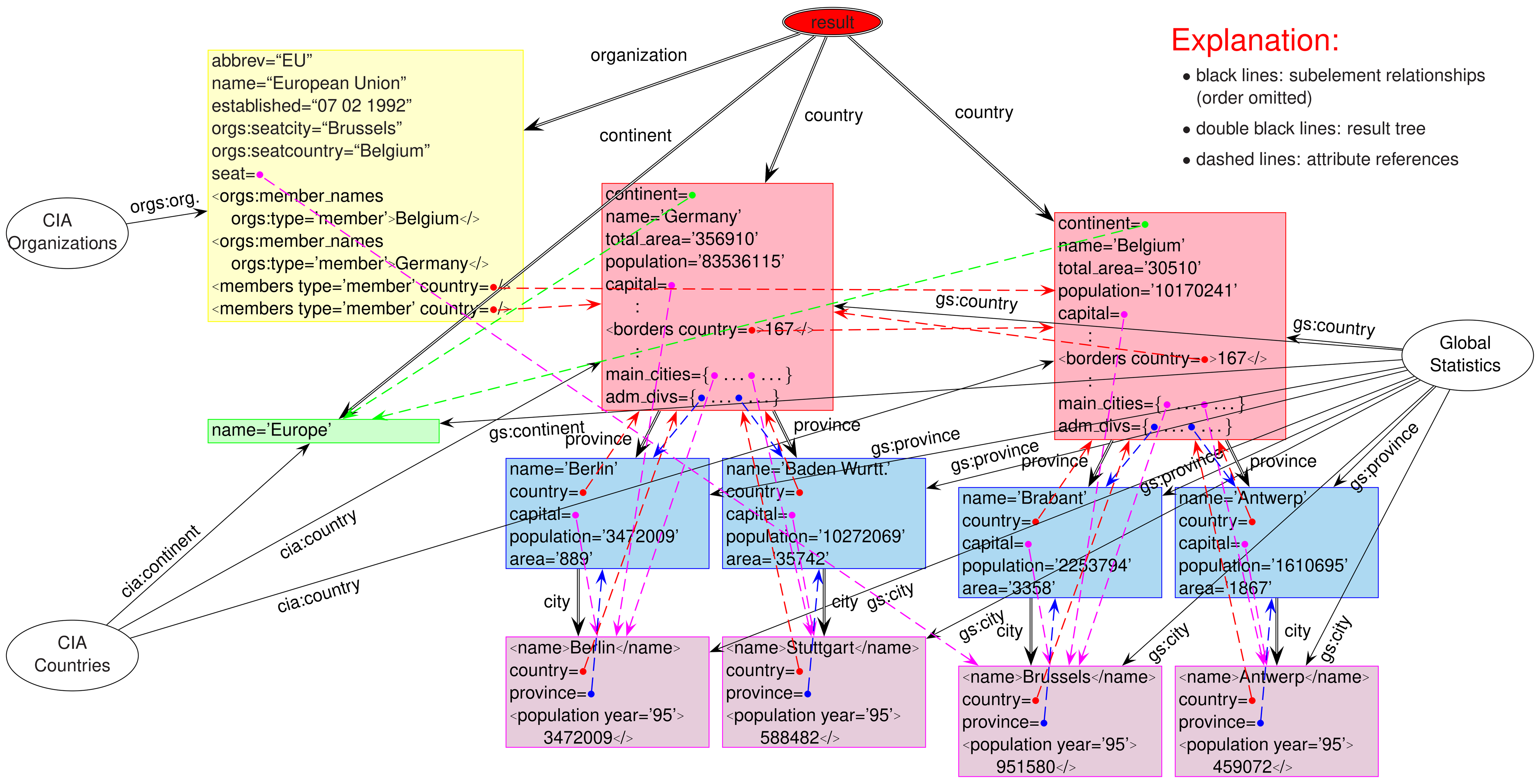
Organization seats

```
O[@seat→Cty] :-
   result/organization→O[@orgs:seatcity→N and
                         @orgs:seatcountry→CN],
   result/city→Cty[@name→N and
                   @country/@name→CN].
```

Organization Memberships

```
O/members[@type→T and @country→C] :-
   result/organization→O
    /orgs:member_names[@orgs:type→T]
     /orgs:text()→CN,
   result/country→C[@name→CN].
```

### Resulting XTreeGraph




Explanation:

- black lines: subelement relationships (order omitted)
- double black lines: result tree
- dashed lines: attribute references

## Continuing the Integration

Subsequent steps integrate the auxiliary country information, adding the car code and german names of the countries. Both are then used for integrating the TERRA data source which uses german names. The full program and the resulting XML file can be found at `http://www.informatik.uni-freiburg.de/~may/lopix`.