A Rule-Based Querying and Updating Language for XML

Wolfgang May
Institut für Informatik
Universität Freiburg
Germany
may@informatik.uni-freiburg.de

DBPL Workshop
Frascati, 9.9.2001

LANGUAGES

- Addressing/Selection:
  XPath
- Querying and Transformation
  - XSLT (Transformation): XPath
  - XML-QL (1998)
    - XML Patterns
    - Problem: dereferencing only supportable by joins
    - no notion of XML's axes
  - Quilt/XQuery (1999/2000)
    - FLWR-Clauses
    - XPath
- Generation
  - instantiating XML Patterns
- W3C XML Query Requirements/Data Model/Algebra
- no XML update language
  "Updating XML" @ SIGMOD 2001
- XML data integration: MIX, based on XMAS/XML-QL

Design Decisions

- experiences with F-Logic for semi-structured data and data integration
- extend XPath
- XPath-Logic: describing and reasoning in XML structures
- Horn Fragment: XPathLog
  declarative rule-based language with bottom-up semantics
  use XPath for *updating* and *generating* XML
- graph model, overlapping trees, multiple parents

TOPICS OVERVIEW

- XPathLog as an XML Database Programming Language:
  DBPL '01
- Considerations on the Data Model: FMLDO/FMII '01
  independent from the programming language

  Data Integration

  - objects of different sources represent the same real-world object

  ⇒ Fusing objects, merging their properties

  - not compatible with XML Data Model (DOM, XML Query Data Model)
  - graph data model
- Application in "intelligent" data integration: KRDB '01
- Implementation: LoPiX
  VLDB Demonstration Track

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>Belgium</name>
     <population year=“95”>951580</population>
   </city>
   :
 </country>

 <country car_code=“D” capital=“cty-Berlin”
          memberships=“org-eu org-nato . . . ”>
   :
 </country>

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

 <organization id=“org-nato” seat=“cty-Brussels” . . . >
  :
 </organization>
  :
</mondial>
```

XPathLog by 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.

XPathLog by Examples

Metadata: Tag Variables and Schema querying

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

:

XPath-Logic: Syntax

- XPathLog: Horn Fragment of XPath-Logic

Extends the XPath syntax

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
- quantifiers

DATA MODEL: XTREEGRAPH

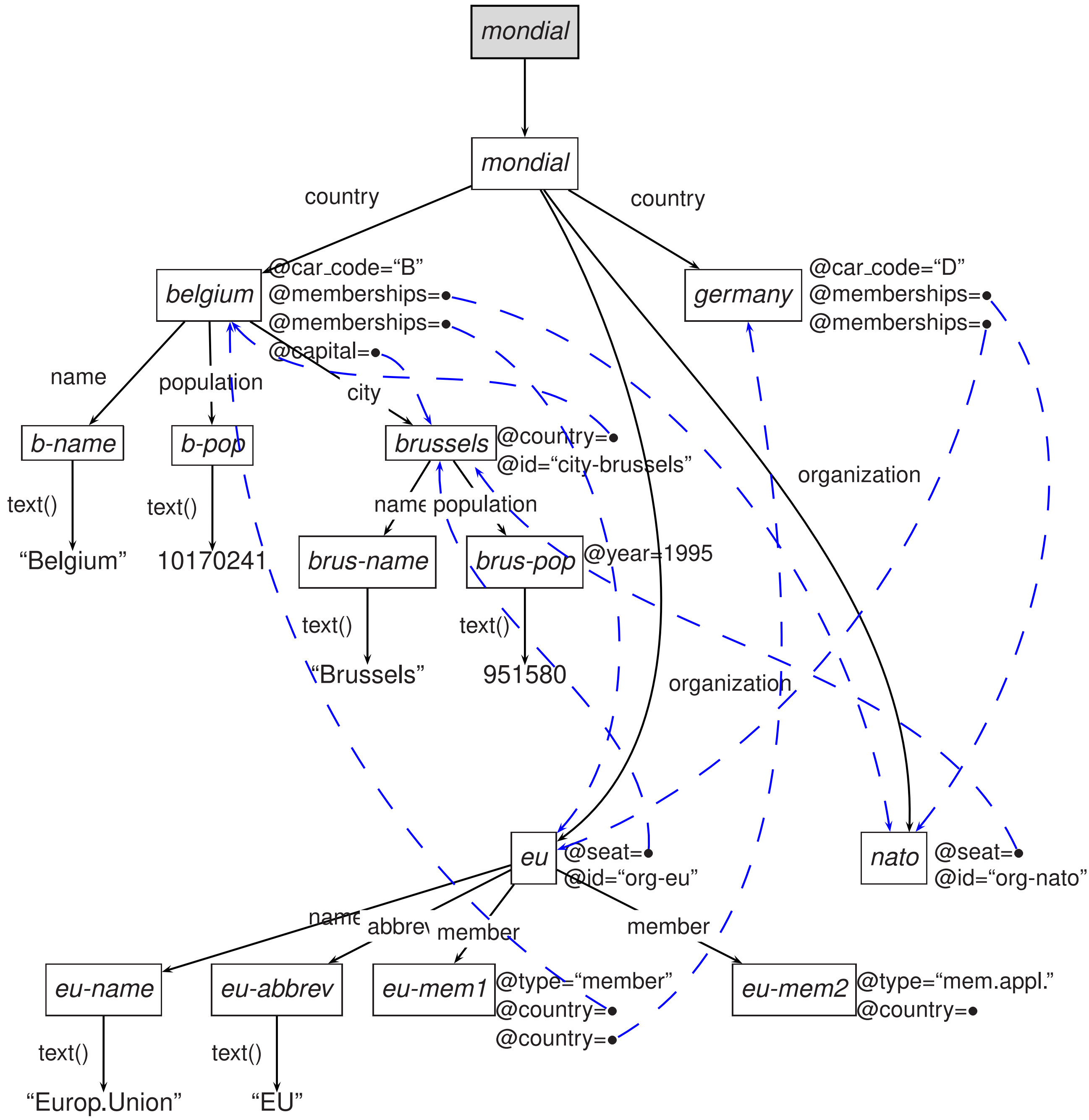
- extends DOM/XML Query Data Model
- element nodes,
- subelement relationship,
- attributes:
  - multivalued attributes (`NMTOKENS`, `IDREFS`) are split
  - reference attributes (`IDREF`) are resolved

$\Rightarrow$ stores only child and attribute axis (DOM-style)

- $\mathcal{A}_{\mathcal{X}}(\mathsf{child}, x)$: list of ($name$,$element$) pairs, including ($text$,$string$).
- $\mathcal{A}_{\mathcal{X}}(\mathsf{attribute}, x)$: set of ($name$,$value$) and ($name$,$element$) pairs

- Mapping: XML instance $\rightarrow$ canonical XTree
- may become a graph through updates:
  $element$.INSERT $contents$ {before|after} $child$

EXAMPLE: MONDIAL XTREEGRAPH



FORMAL SEMANTICS

XPath-Logic: model-theoretic semantics, extends semantics for XPath by P.Wadler (1999)

$$\mathcal{S}_{\mathcal{X}} : \text{Ref\_Exprs} \to (\mathcal{V} \cup \mathcal{L} \cup \mathcal{N})^{\mathbb{N}}$$
$$\mathcal{Q}_{\mathcal{X}} \subseteq \text{Pred\_Exprs} \times \mathcal{V} \times \text{Var\_Assignments}$$

Theorem 1
For variable-free expressions (i.e., XPath expressions) without

- *navigating along reference attributes and*
- *splitting `NMTOKENS` attributes*

the semantics coincides with the one given in [wadler-99]: For every such XPath expression $expr$,

$$\mathcal{S}_{\mathcal{X}}(expr) = \mathcal{S}[[expr]](x)$$

(for arbitrary x) where $\mathcal{S}[[expr]]$ is as defined in [wadler-99] and enumerated wrt. document order.

- join variables restrict the result set.

XPathLog Rules

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

- Evaluation of rule bodies = queries
- Constructive semantics for XPathLog atoms in rule heads

Head: definite XPathLog atoms:

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

Bodies/Queries: Answer set semantics

Annotated result list

(i) a result list, and

(ii) with every element of the result list, a list of variable bindings is associated.

Example

$expr$ =

//organization→O[member/@country[@car_code→C and
name/text()→N]]
/abbrev/text()→A.

results in

list((“UN”, {(O/*un*, A/“UN”, C/“AL”, N/“Albania”),
(O/*un*, A/“UN”, C/“GR”, N/“Greece”),
: }),
(“EU”, {(O/*eu*, A/“EU”, C/“D”, N/“Germany”),
(O/*eu*, A/“EU”, C/“F”, N/“France”),
: }),
:)

SEMANTICS

- Algebraic semantics

Extends $\mathcal{S}$ and $\mathcal{Q}$:

$$\begin{aligned}
\mathcal{SB}_{\mathcal{X}}: \quad & (\text{Ref_Exprs} \times \text{Var_Bindings}) \rightarrow \text{AnnotatedResults}^{\mathbb{N}} \\
& (\text{Axes} \times \mathcal{V} \times \text{Ref_Exprs} \times \text{Var_Bindings}) \\
& \qquad\qquad\qquad\qquad \rightarrow \text{AnnotatedResults}^{\mathbb{N}} \\
\mathcal{QB}_{\mathcal{X}}: \quad & (\text{Literals} \times \text{Var_Bindings}) \rightarrow \text{Var_Bindings} \\
& (\text{Pred_Exprs} \times \mathcal{V} \times \text{Var_Bindings}) \rightarrow \text{Var_Bindings}
\end{aligned}$$

- Left-to-Right propagation of variable bindings (*sideways information passing strategy*):

  $\mathcal{SB}_{\mathcal{X}}($*axis,node,refExpr,*$Bdgs)$

  mirrors the generation of answer sets by algebraic evaluation: $Bdgs$ may contain bindings for free variables in *refExpr*:

  – $Bdgs$ serves as join variables

  – $Bdgs$ is completed by evaluating *refExpr*

- evaluate negation as a relational “minus” operator: *exclude* some bindings

CORRECTNESS

Theorem 2

For every XPathLog expression $expr$,

$$\mathrm{Res}(\mathcal{SB}_{\mathcal{X}}(expr)) = \bigcup_{\beta \in (\mathcal{V} \cup \mathcal{L} \cup \mathcal{N})^{\mathsf{free}(expr)}} \mathcal{S}_{\mathcal{X}}(expr, \beta)$$

More detailed, for all $x \in \mathcal{V} \cup \mathcal{L} \cup \mathcal{N}$,

$$(x \in \mathrm{Res}(\mathcal{SB}_{\mathcal{X}}(expr)) \text{ and } \beta \in \mathrm{Bdgs}(\mathcal{SB}_{\mathcal{X}}(expr), x)) \Leftrightarrow x \in \mathcal{S}_{\mathcal{X}}(expr, \beta)$$

SEMANTICS OF RULE HEADS

Constructive semantics of XPath expressions

- Definite XPathLog atoms:
  - use only the child and sibling axes
  - no negation, function applications, aggregation, and *proximity position predicates*

"/" and "[. . .]" act as constructors:

- $host[property{\rightarrow}value]$ modifies $host$
- $host/property\ remainder$
  inserts new element $host/property$ which satisfies $remainder$
- $property$ of the form
  - child::$name$
  - child(i)::$name$
  - preceding/following-sibling::$name$
  - preceding/following-sibling(i)::$name$
  - attribute::$name$

$\Rightarrow$ unambiguous insertions

SEMANTICS OF RULE HEADS

Attributes

C[@datacode→"de"], C[@memberships→O] :-
  //country→C[@car_code="D"],
  //organization→O[abbrev/text()→"EFTA"].

⇓

<country $\boxed{\text{datacode="de"}}$ car_code="D"
  memberships="org-eu org-un $\boxed{\text{org-efta}}$ . . . ">
  ⋮
</country>

- C: host element
- O: target of reference
- extends $\mathcal{A}_{\mathcal{X}}(\mathtt{attribute}, \textit{germany})$

Semantics of Rule Heads

Create “free” elements

/country[@car_code→“BAV”].

⇓

<country car_code=“BAV”> </country>

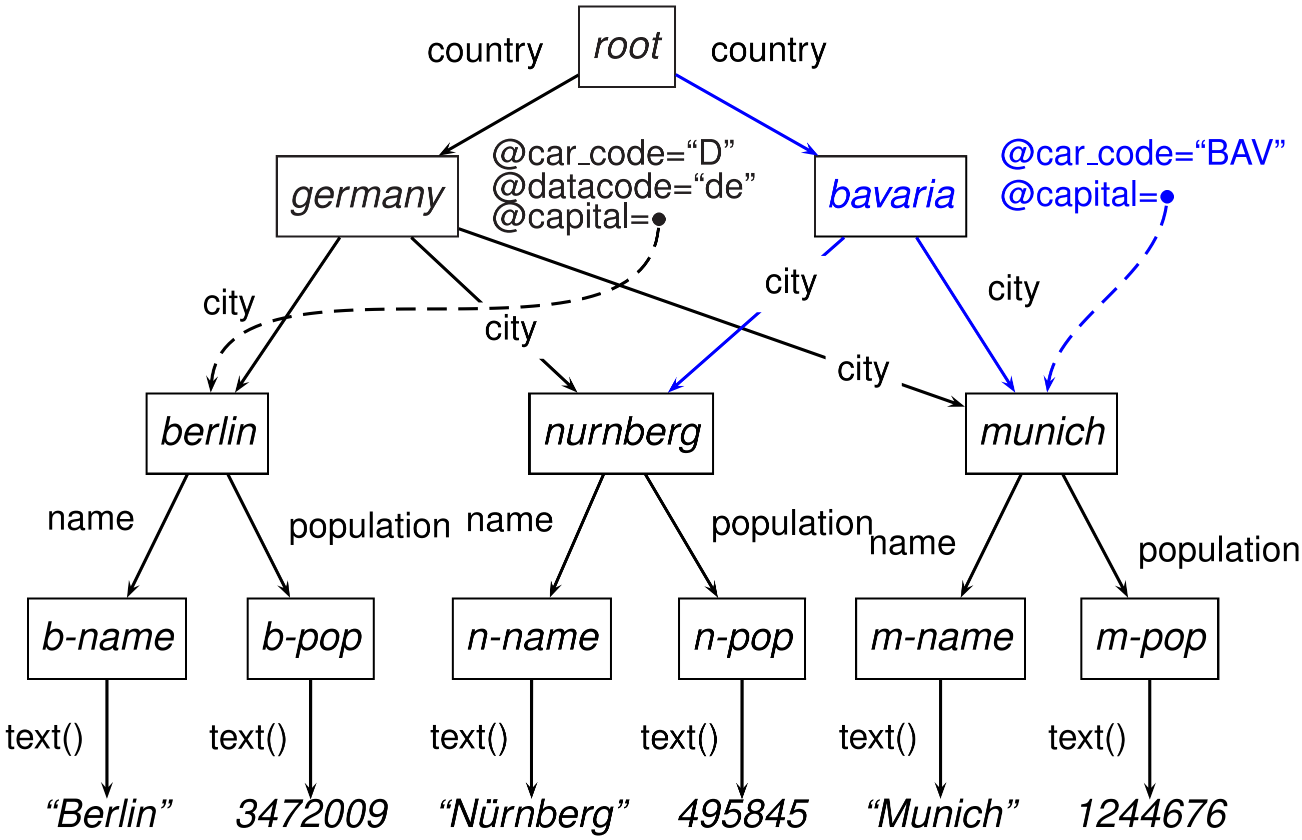
SEMANTICS OF RULE HEADS

Add 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.
- extends $\mathcal{A}_{\mathcal{X}}(\mathrm{child}, \textit{bavaria})$ with $(\textit{munich}, \mathrm{city})$ and $(\textit{nurnberg}, \mathrm{city})$.
- crucial for efficient in-place restructuring and integration

Example: Linking



- elements may have multiple parents

Semantics of Rule Heads

Generation of Elements by Path Expressions

C/name[text()→"Bavaria"] :-

//country→C[@car_code="BAV"].

⇓

<country car_code="BAV" capital="city-munich">
<city>. . . </city>
<city>. . . </city>
<name>Bavaria</name>
</country>

Atomized:

C[name→_N], _N[text()→"Bavaria"] :-

root[descendant::country→C], C[@car_code="BAV"].

FORMAL SEMANTICS OF RULE HEADS

- bottom-up semantics with T_P-operator
  - Atomize complex expressions in the head (only *definite* expressions) into atoms of the forms

    $node[axis{::}nodetest{\rightarrow}\mathsf{X}]$ and $node[axis(i){::}nodetest{\rightarrow}\mathsf{X}]$

    and extend $\mathcal{A}_{\mathcal{X}}(\mathsf{child}, node)$ and $\mathcal{A}_{\mathcal{X}}(\mathsf{attribute}, node)$
- Negation: (user-defined) stratification

XPATHLOG: EXTENSIONS

- XPathLog supports class hierarchy and non-monotonic value inheritance
- Signatures: “lightweight” signature formalism:

  country[@car_code⇒string].

  country[@area⇒numeric].

  country[@capital⇒city].

  country[city⇒city].

  used for defining tree projections of the internal database

Integration: "Three-level" model

- "Warehouse" strategy, "global-as-view"

access multiple sources

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

merge data from different sources

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

"export" layer: result trees views defined as projections

- root node
- signature

RESULTS

- declarative semantics for generating XML with XPath
- powerful, flexible language
  - metadata/schema querying
  - specialized data integration operations
    (e.g., element creation, element fusion, synonyms)
- first available implementation of XML updates
- implementation: LoPiX (using major components of Florid)
- practicability: case study
- graph data model suitable & necessary for integration
- extension concepts
  (classes, signatures)

Contents