PRATA: A System for XML Publishing, Integration and View Maintenance

Gao Cong, Wenfei Fan, Xibei Jia and Shuai Ma

gao.cong@ed.ac.uk, wenfei@inf.ed.ac.uk, x.jia@sms.ed.ac.uk, sma1@inf.ed.ac.uk

Abstract: We present PRATA, a system that supports the following in a uniform framework: (a) XML publishing, *i.e.,* converting data from databases to an XML document, (b) XML integration, *i.e.,* extracting data from multiple, distributed databases, and integrating the data into a single XML document, and (c) incremental maintenance of published or integrated XML data (view), *i.e.,* in response to changes to the source databases, efficiently propagating the source changes to the XML view by computing the corresponding XML changes. A salient feature of the system is that publishing, integration and view maintenance are *schema-directed*: they are conducted strictly following a user-specified (possibly recursive and complex) XML schema, and guarantee that the generated or modified XML document conforms to the predefined schema.

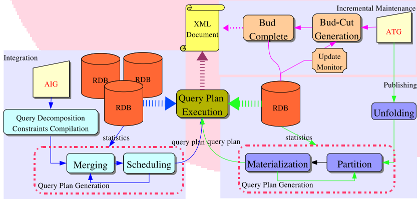

Figure 1. PRATA System Architecture

PRATA consists of three main modules:

- Schema-Directed XML Publishing
- XML Integration
- Incremental Maintenance of XML Views

To our knowledge, PRATA is the first and the only system that is capable of supporting all of these.

Schema-Directed Publishing

- This module allows users to specify mappings from a relational database schema R to a predefined XML schema D, via a GUI and in a novel language *Attribute Translation Grammar* (ATG) that we proposed in [2].
- The ATG approach for publishing relational data in XML is given as follows, by using a simplified example taken from the IUPHAR (International Union of Pharmacology) Receptor Database [4].

Source relational schema R_0:

```
chapters(chapter_id, name)
receptors(receptor_id, chapter_id, name, code)
refs(ref_id, chapter_id, year, title)
cite(ref_id, receptor_id)
```

Target DTD D_0:

```
<!ELEMENT db         (family*)>
<!ELEMENT family*    (name, receptors, references )>
<!ELEMENT references (reference*)>
<!ELEMENT reference  (title, year)
<!ELEMENT receptors  (receptor*)>
<!ELEMENT receptor   (name, receptors)>
/* #PCDATA is omitted here.  */
```

ATG σ_0:

Semantic Attributes: /*omitted*/

Semantic Rules:

```
db → family*
$Q_1$:  $family ← select  chapter_id, name  from  chapters
family → name, receptors, references
  $fname = ($family.name),            $references = ($family.chapter_id),
  $receptors = (0, $family.chapter_id, ∅)
receptors → receptor*
$Q_2$:  $receptor ← case $receptors.tag of
    0:        select  receptor_id, name, $receptors.ids
              from    receptors
              where   chapter_id = $receptors.id
    1:        select  a.receptor_id, a.name, $receptors.ids
              from    receptors a, cite b, cite c
              where   b.receptor_id = $receptors.id and
                      b.ref_id = c.ref_id and
                      b.receptor_id <> c.receptor_id and
                      a.receptor_id = c.receptor_id and
                      a.receptor_id not in $receptors.ids
receptor → name, receptors
  $rname = ($receptor.name),
  $receptors = (1, $receptor.receptor_id, $receptor.ids ∪ $receptor.receptor_id)
references → reference*
$Q_3$:  $reference ← select   title, year
                       from     refs
                       where    chapter_id = $references.chapter_id
reference → title, year
  $year = ($reference.year),     $title = ($reference.title)
A → S   /* A is one of name, title, year */
  $S = ($A.val)
```

- Given an ATG σ_0 and an IUPHAR database instance I of R_0 as above, the system automatically generates an XML document (view) $\sigma_0(I)$ of I such that $\sigma_0(I)$ is guaranteed to conform to the given DTD D_0 as above.
- We successfully generate the XML document of the whole IUPHAR database using ATG grammar.
- One can write ATG "programs" easily with basic knowledge of SQL and DTD.

XML Integration

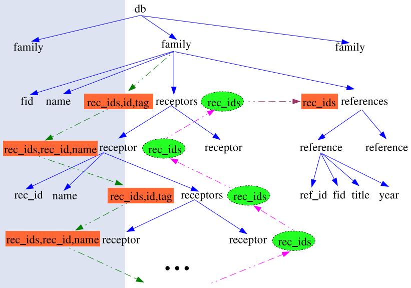

Figure 2. An example of XML report

- Dashed arrows in Figure 2 represent information flows, which contain top-down, bottom-up, and sideway information passing, thus top-down methods such as ATGs [1] do not work any more in this case.
- In data integration environment, constraints, such as keys and foreign keys, and distributed quries become normal requirements.
- *Attribute Integration Grammar* [1] extends the support for ATGs.
- Given an AIG and source databases, this module extracts data from the distributed sources and produces an XML document that is guaranteed to conform to the given DTD D and satisfy predefined XML contraints Σ.

Incremental Maintenance

- This module maintains published XML viewsc $\sigma(I)$ based on our incremental computation techniques developed in [3].
- In response to changes ΔI to the source database I, this module computes the XML changes ΔT to $\sigma(I)$ such that $\sigma(I \oplus \Delta I) = \Delta T \oplus \sigma(I)$, while minimizing unnecessary recomputation. The operator $\oplus$ denotes the application of these updates.
- The performance is proportional to the size of the updates instead of the whole view.

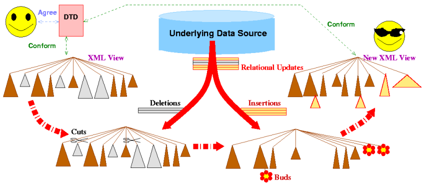

Figure 3. Incremental XML View Maintenance

Features

Taken together, PRATA has the following salient features, which are beyond what are offered by commercial tools or prototype systems developed thus far.

- Schema conformance
- Automatic validation of XML constraints
- Integration of multiple, distributed data sources
- Incremental updates
- Novel evaluation and optimization techniques
- Friendly GUIs.

Current Status

The current implementation of PRATA fully supports:

- (a) schema-directed XML publishing.
- (b) incremental maintenance of XML views of a single source, based on novel evaluation and optimization techniques.
- (c) for XML schemas, it allows generic (possibly recursive and non-deterministic) DTDs, but has not yet implemented the support for XML constraints.

We are currently implementing:

- (a) XML integration
- (b) incremental maintenance of XML views of multiple sources.

References

[1] M. Benedikt, C. Y. Chan, W. Fan, J. Freire, and R. Rastogi. Capturing both types and constraints in data integration. In *SIGMOD*, 2003.

[2] M. Benedikt, C. Y. Chan, W. Fan, R. Rastogi, S. Zheng, and A. Zhou. DTD-directed publishing with attribute translation grammars. In *VLDB*, 2002.

[3] P. Bohannon, B. Choi, and W. Fan. Incremental evaluation of schema-directed XML publishing. In *SIGMOD*, 2004.

[4] IUPHAR. Receptor Database. `http://www.iuphar-db.org`.