

Viewing a World of Annotations through AnnoVIP

K. Karanasos I. Manolescu S. Zoupanos

Leo group
INRIA Saclay-Île-de-France and LRI, Université de Paris Sud-11

Codex Project Meeting, 19/3/2010

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE



INRIA

centre de recherche **SACLAY - ÎLE-DE-FRANCE**



Overview

- Distributed Hash Tables (DHT)
- ViP2P
- AnnoVIP

We are not alone!

Asterios Katsifodimos

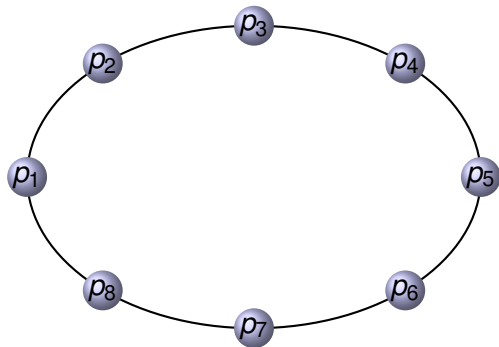
Jesús Camacho-Rodríguez

Alin Tilea

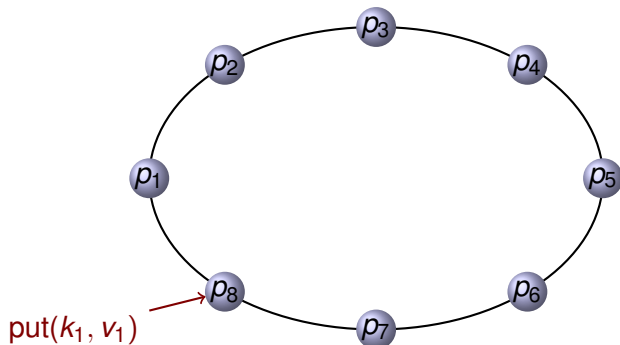
Martin Goodfellow

- Distributed Hash Tables (DHT)
- ViP2P
- AnnoVIP

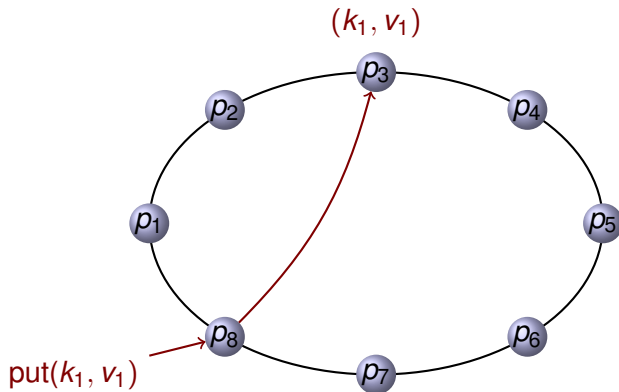
Distributed hash tables



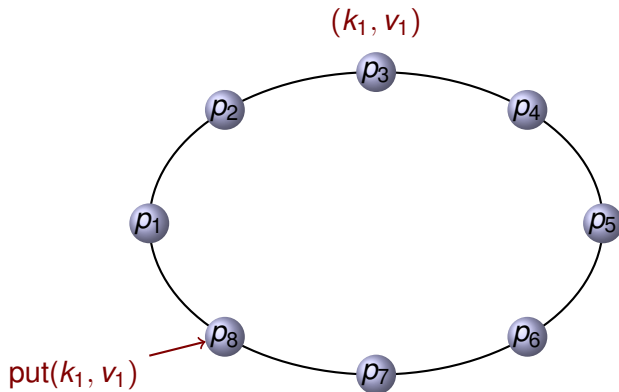
Distributed hash tables



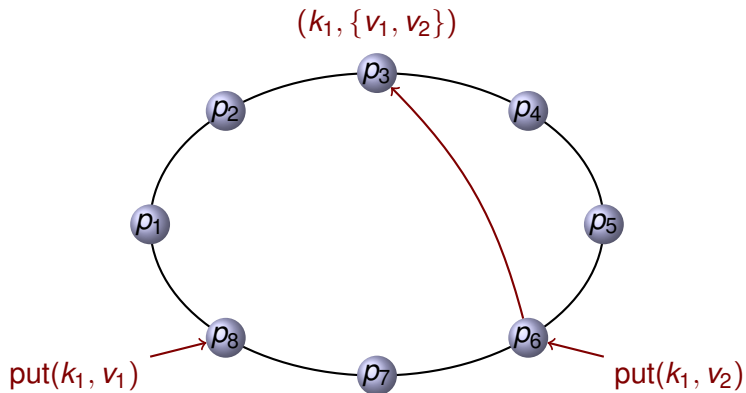
Distributed hash tables



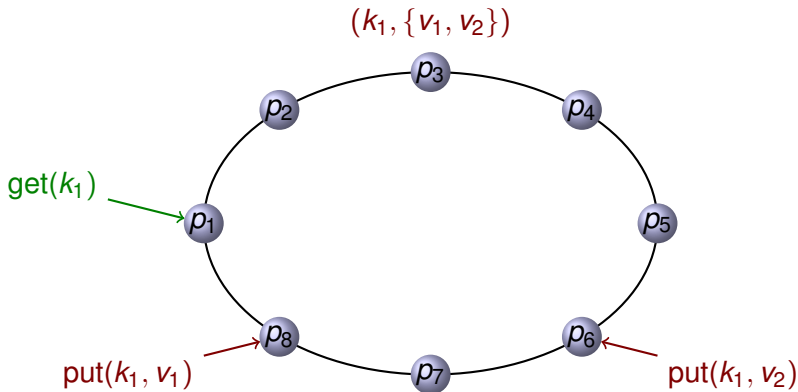
Distributed hash tables



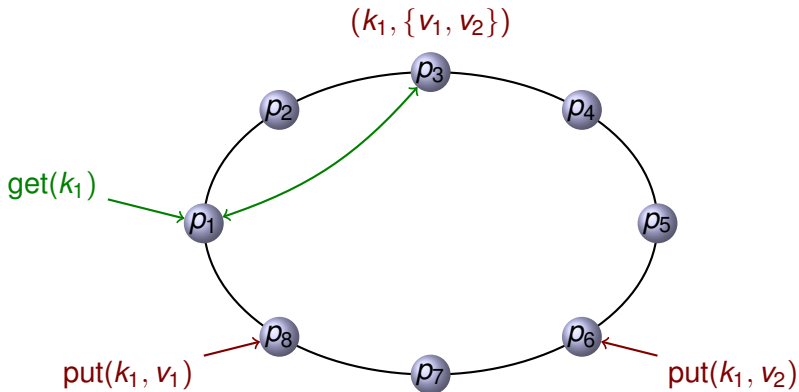
Distributed hash tables



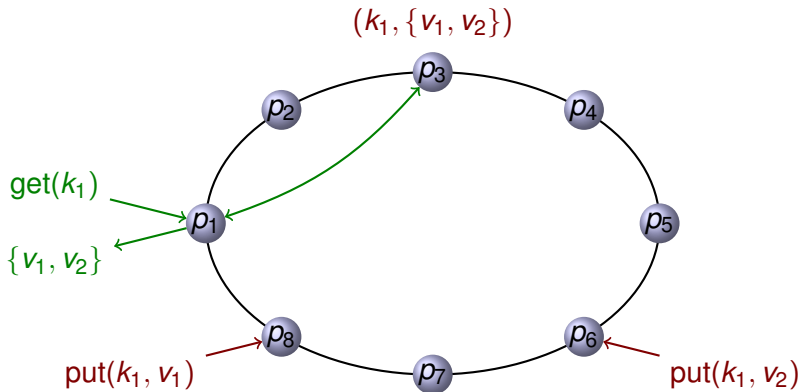
Distributed hash tables



Distributed hash tables

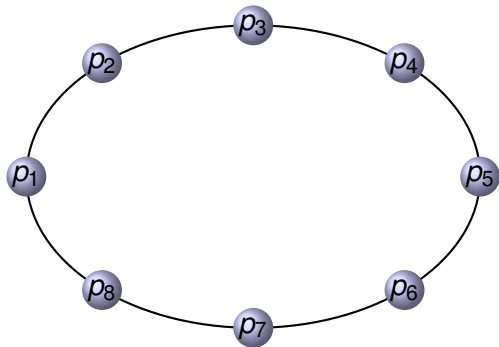


Distributed hash tables

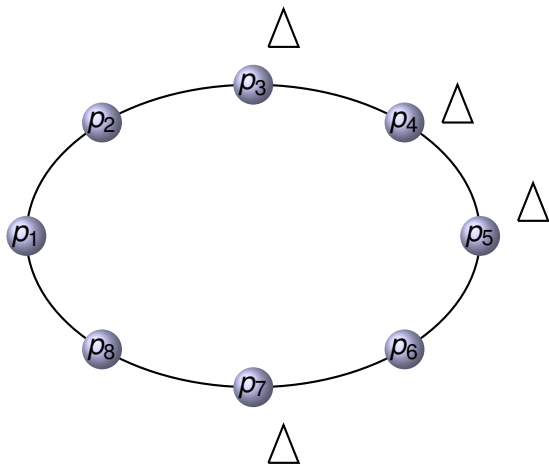


- Distributed Hash Tables (DHT)
- **ViP2P**
- AnnoVIP

ViP2P: views in peer-to-peer



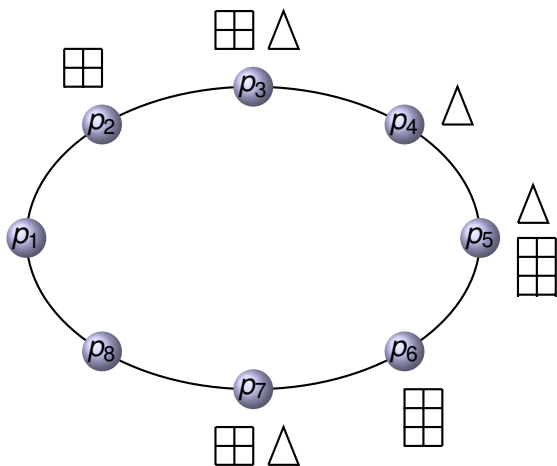
ViP2P: views in peer-to-peer



The peers may store:

- documents

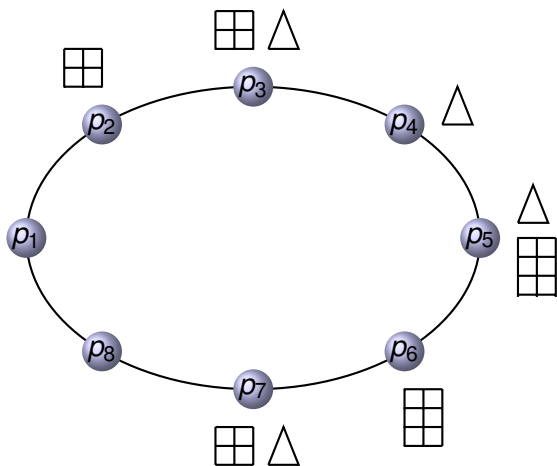
ViP2P: views in peer-to-peer



The peers may store:

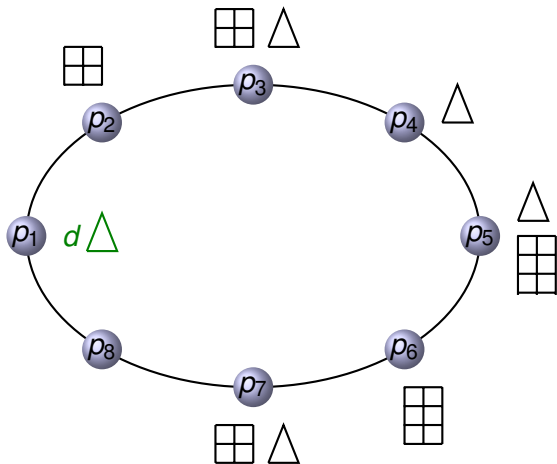
- documents
- views

ViP2P: views in peer-to-peer



View materialization

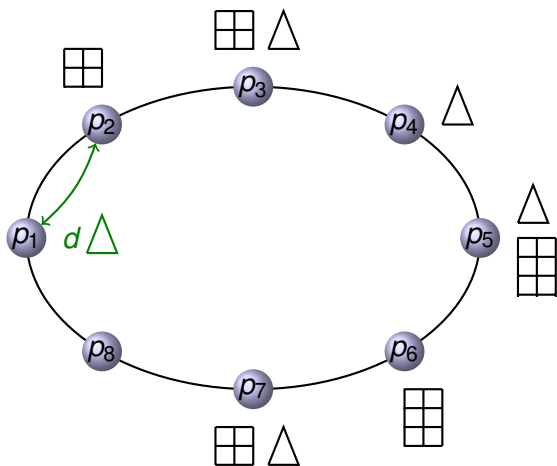
ViP2P: views in peer-to-peer



View materialization

When d arrives:

ViP2P: views in peer-to-peer

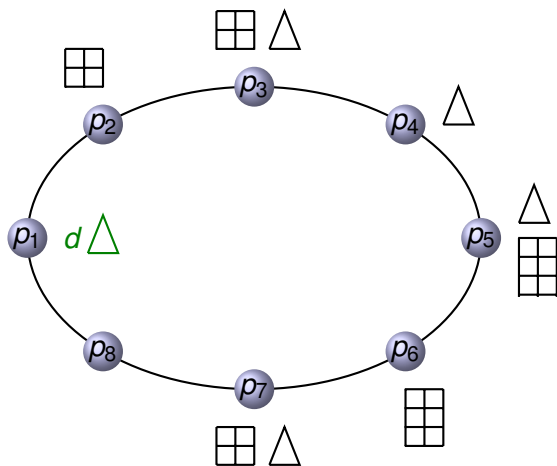


View materialization

When d arrives:

- search view definitions for which $v_i(d) \neq \emptyset$

ViP2P: views in peer-to-peer

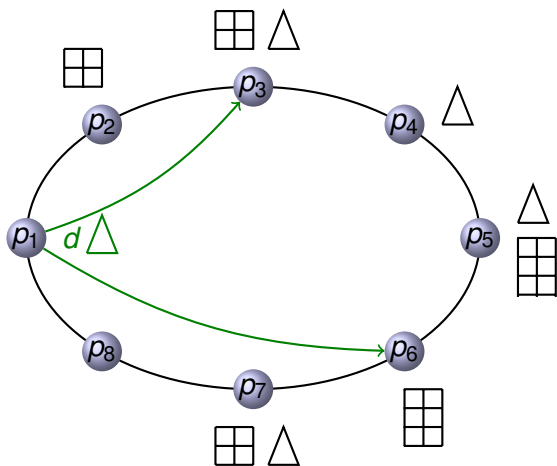


View materialization

When d arrives:

- search view definitions for which $v_i(d) \neq \emptyset$
- compute $v_i(d)$

ViP2P: views in peer-to-peer

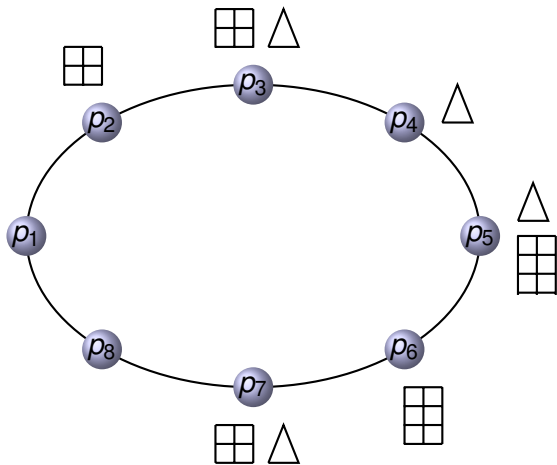


View materialization

When d arrives:

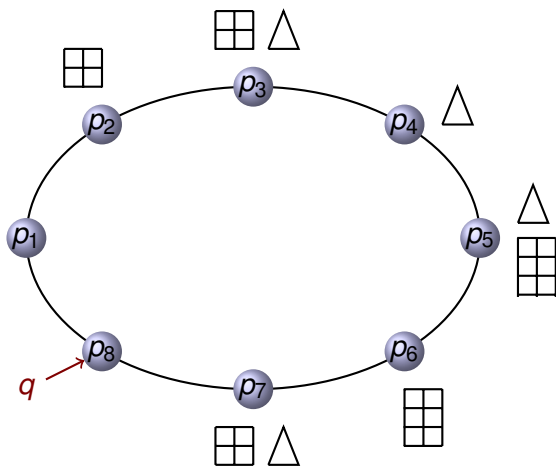
- search view definitions for which $v_i(d) \neq \emptyset$
- compute $v_i(d)$
- send results

ViP2P: views in peer-to-peer



Query answering

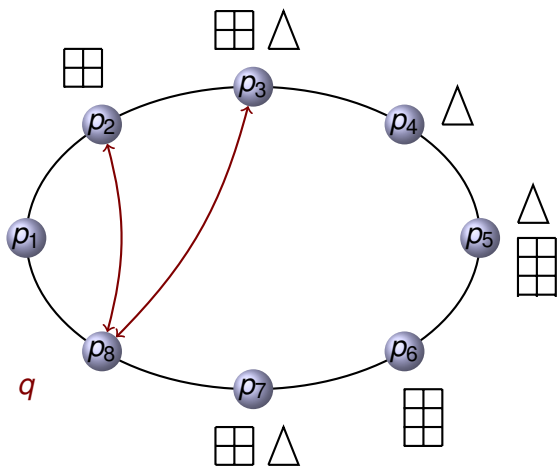
ViP2P: views in peer-to-peer



Query answering

When q arrives:

ViP2P: views in peer-to-peer

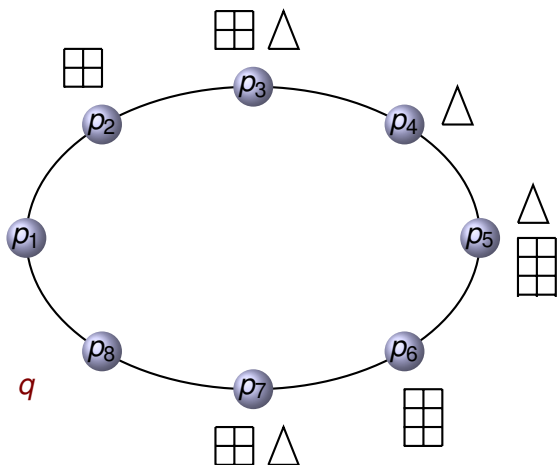


Query answering

When q arrives:

- view definition lookup

ViP2P: views in peer-to-peer

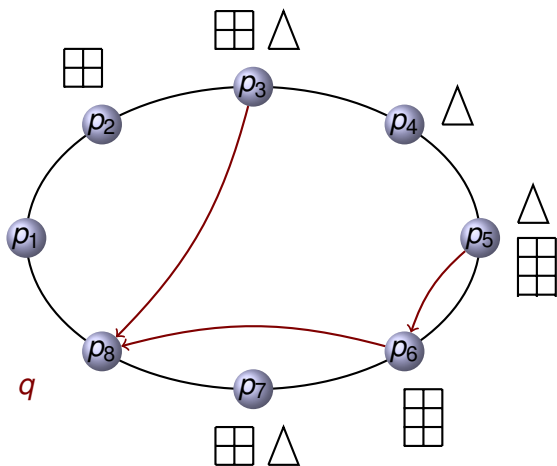


Query answering

When q arrives:

- view definition lookup
- rewriting

ViP2P: views in peer-to-peer



Query answering

When q arrives:

- view definition lookup
- rewriting
- execution of physical plan

Representing our queries and views.

- Each pattern node carries a label (element name or attribute name or word).
- Each pattern edge correspond to a /- or a //-relationship between nodes.
- A node may be **annotated with** zero or more among the following **labels**: *id*, *cont* and *val*.
- A node may be **annotated with a predicate** of the form $[val = \underline{c}]$.

ViP2P: Algebraic rewriting & operators

Let q be a query and $\mathcal{V} = \{v_1, v_2, \dots, v_k\}$ a set of views.
A **rewriting** of q using \mathcal{V} is an algebraic expression

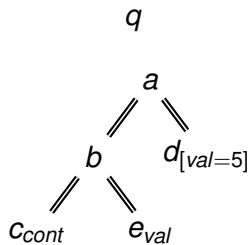
$$e(v_1, v_2, \dots, v_k)$$

such that $e(\mathcal{D}) = q(\mathcal{D})$ for any document set \mathcal{D}

Algebra operators

$scan(v)$	\times
$\pi_{cols}(op)$	$\pi^o(op)$
$sort_{cols}(op)$	$\sigma_{cond}(op)$
$nav_{i,np}(op)$	evaluates np over the $cont$ attribute $op.i$

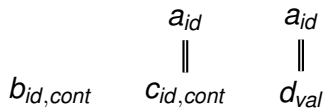
ViP2P: Rewriting example



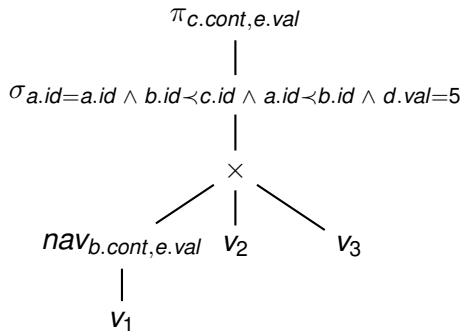
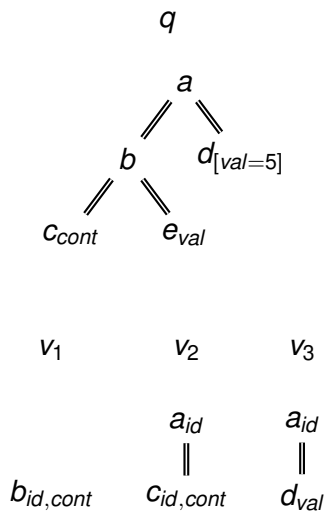
v_1

v_2

v_3

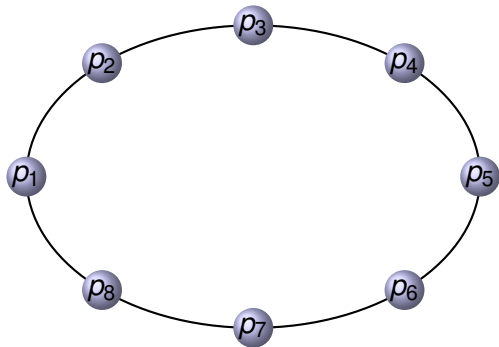


ViP2P: Rewriting example

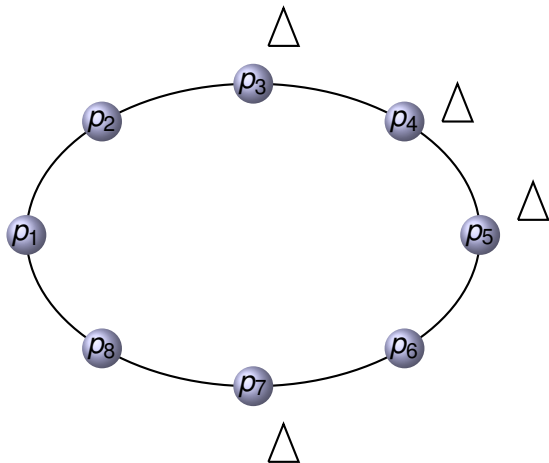


- Distributed Hash Tables (DHT)
- ViP2P
- AnnoVIP

AnnoVIP: Annotations in ViP2P



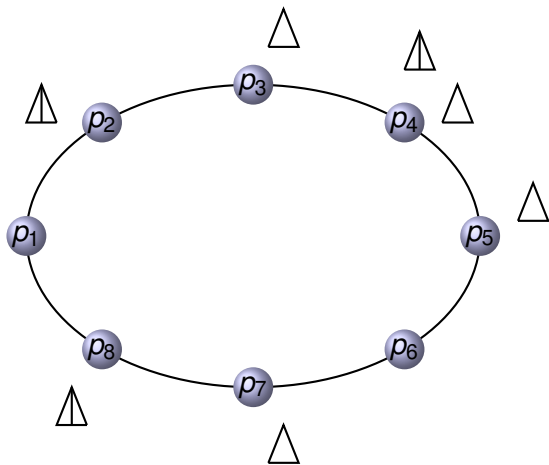
AnnoVIP: Annotations in ViP2P



The peers may store:

- documents (XML)

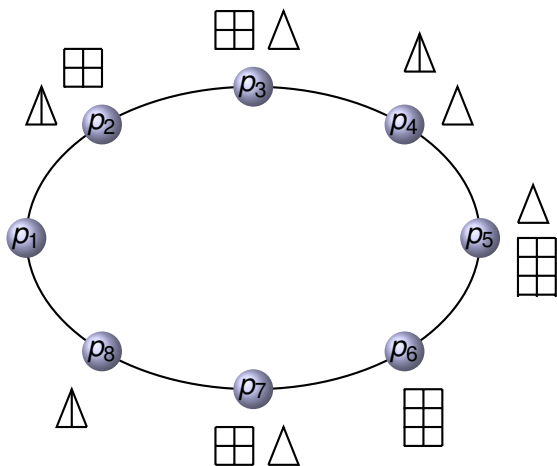
AnnoVIP: Annotations in ViP2P



The peers may store:

- documents (XML)
- annotations (RDF)

AnnoVIP: Annotations in ViP2P



The peers may store:

- documents (XML)
- annotations (RDF)
- views

AnnoVIP: Overview of the new features

- Extended pattern dialect for views and queries (**tree patterns with value joins** - subset of XQuery).

AnnoVIP: Overview of the new features

- Extended pattern dialect for views and queries (**tree patterns with value joins** - subset of XQuery).
- Modified **view materialization**.
 - each tree pattern of a view may contain data from different documents
 - each document is published independently

AnnoVIP: Overview of the new features

- Extended pattern dialect for views and queries (**tree patterns with value joins** - subset of XQuery).
- Modified **view materialization**.
 - each tree pattern of a view may contain data from different documents
 - each document is published independently
- Extended **rewriting algorithm** to deal with the new pattern dialect.

AnnoVIP: Overview of the new features

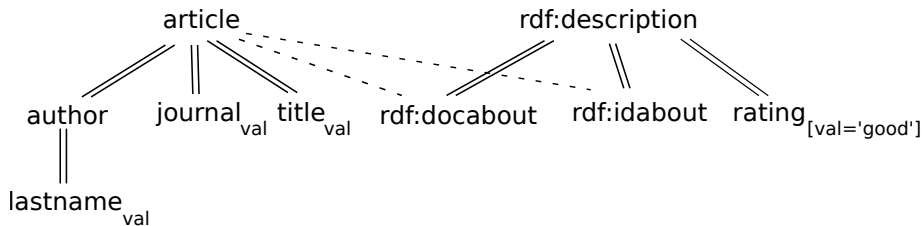
- Extended pattern dialect for views and queries (**tree patterns with value joins** - subset of XQuery).
- Modified **view materialization**.
 - each tree pattern of a view may contain data from different documents
 - each document is published independently
- Extended **rewriting algorithm** to deal with the new pattern dialect.
- Support for **namespaces** (needed especially for annotations)

AnnoVIP: Overview of the new features

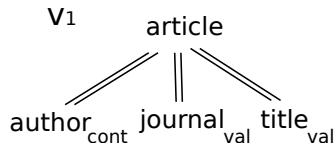
- Extended pattern dialect for views and queries (**tree patterns with value joins** - subset of XQuery).
- Modified **view materialization**.
 - each tree pattern of a view may contain data from different documents
 - each document is published independently
- Extended **rewriting algorithm** to deal with the new pattern dialect.
- Support for **namespaces** (needed especially for annotations)
- Graphical User Interface (which you are about to see...)

Rewriting examples

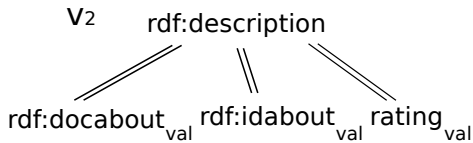
q



V1

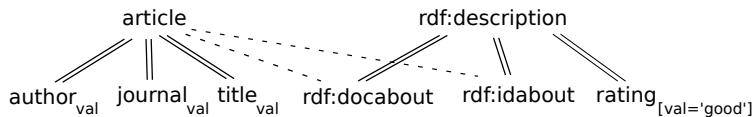


V2

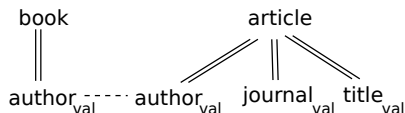


Rewriting examples

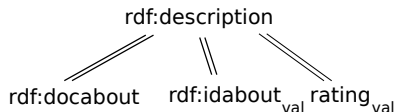
q₂



v₃

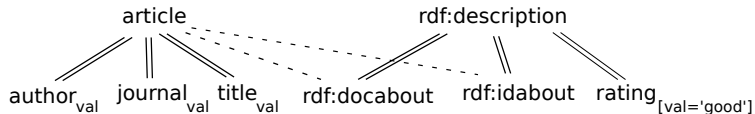


v₄

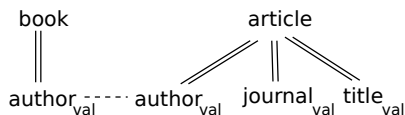


Rewriting examples

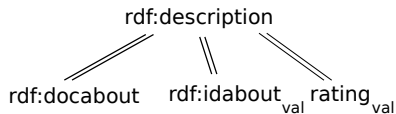
q₂



v₃



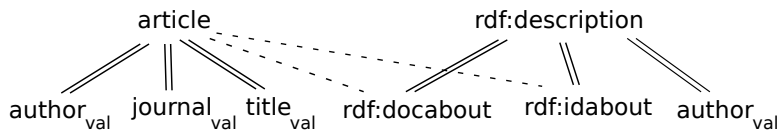
v₄



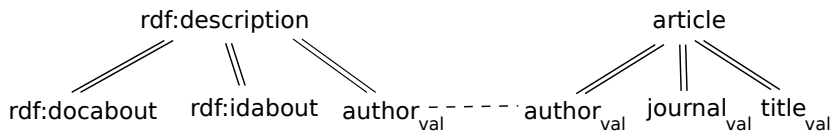
No rewriting!

Rewriting examples

q₃

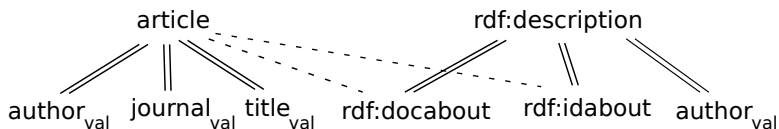


v₅

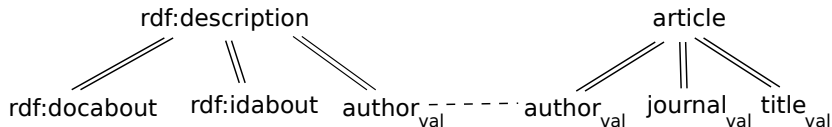


Rewriting examples

q₃

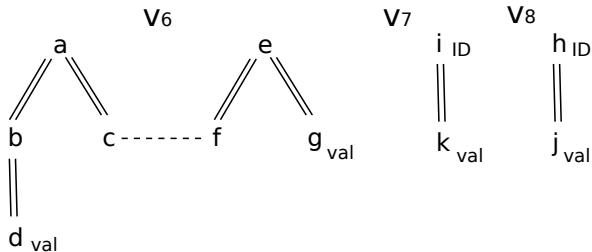
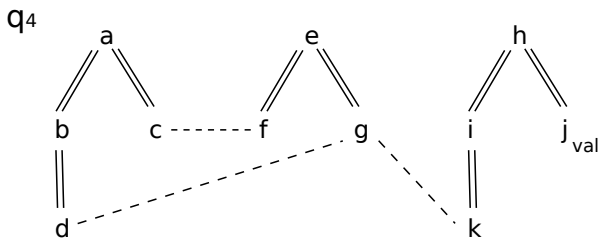


v₅



No rewriting!

Rewriting examples



Thank you