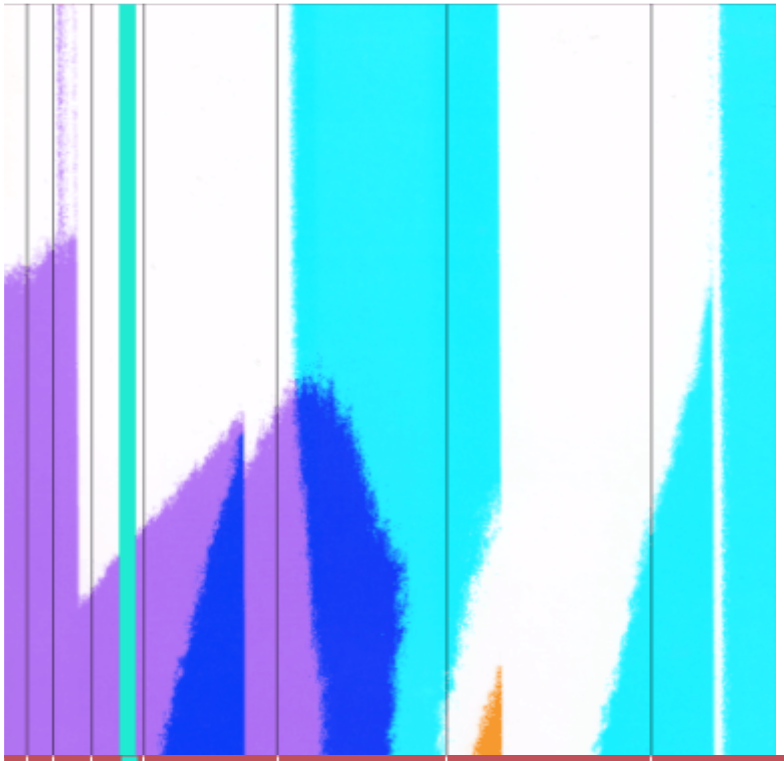


A Test Platform for the INEX Heterogeneous Track

Date

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

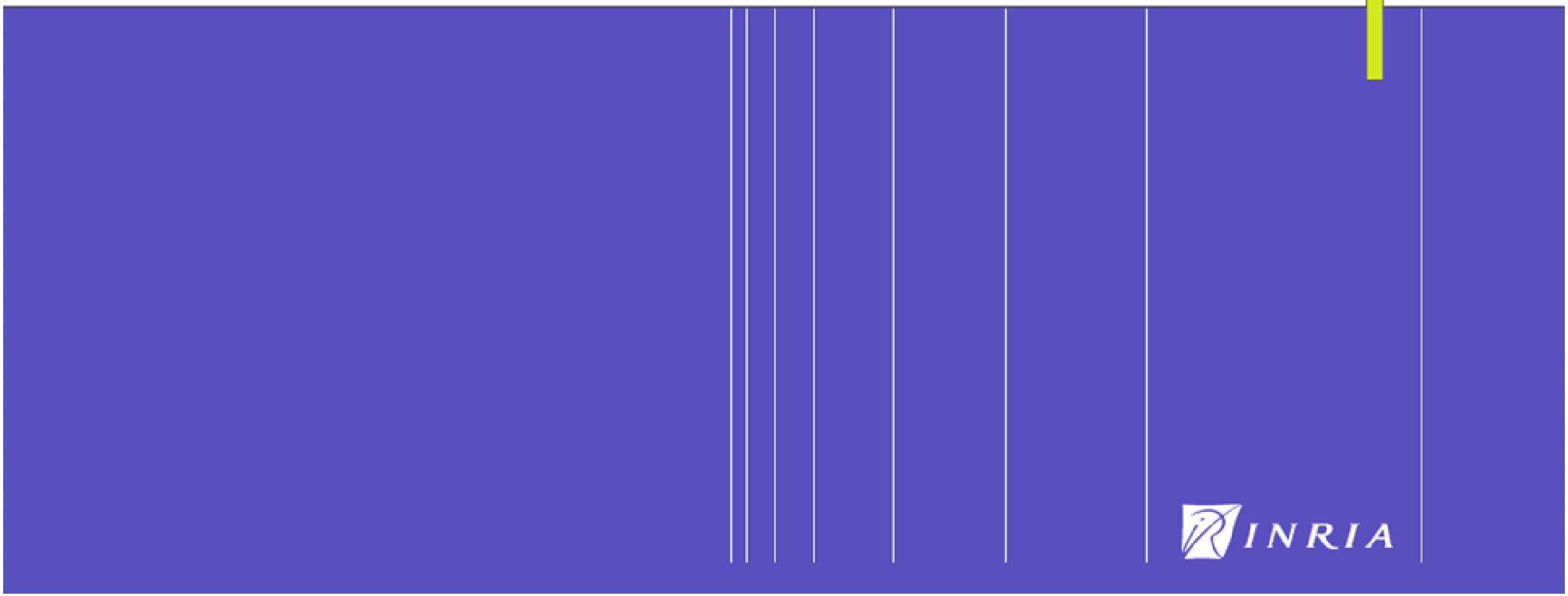
INRIA

Serge Abiteboul (INRIA Futurs, Gemo group, France)
Ioana Manolescu (INRIA Futurs, Gemo group, France)
Benjamin Nguyen (Univ. Versailles, PRiSM lab, France)
Nicoleta Preda (INRIA Futurs, Gemo, France)

Plan

- Our work during INEX 2004
  - Visualizing heterogeneous XML corpora: XSum
  - Unified DTD for the het-track corpora + mappings
  - Approach for (distributed) querying of heterogeneous sources: possible test platform for the het-track
- Perspectives
  - Remaining work on semantic integration
  - Platform deployment
  - Feedback ?

Visualizing heterogeneous XML corpora: XSum



INEX 2004 heterogeneous collection

All data sources related to CS bibliography

Berkeley	Bib Duisburg	CompuScience
DBLP	HCI	QMUL

(tried also documents from the RF track, will see)

We focus on CAS topics

What kind of structures do we have ?

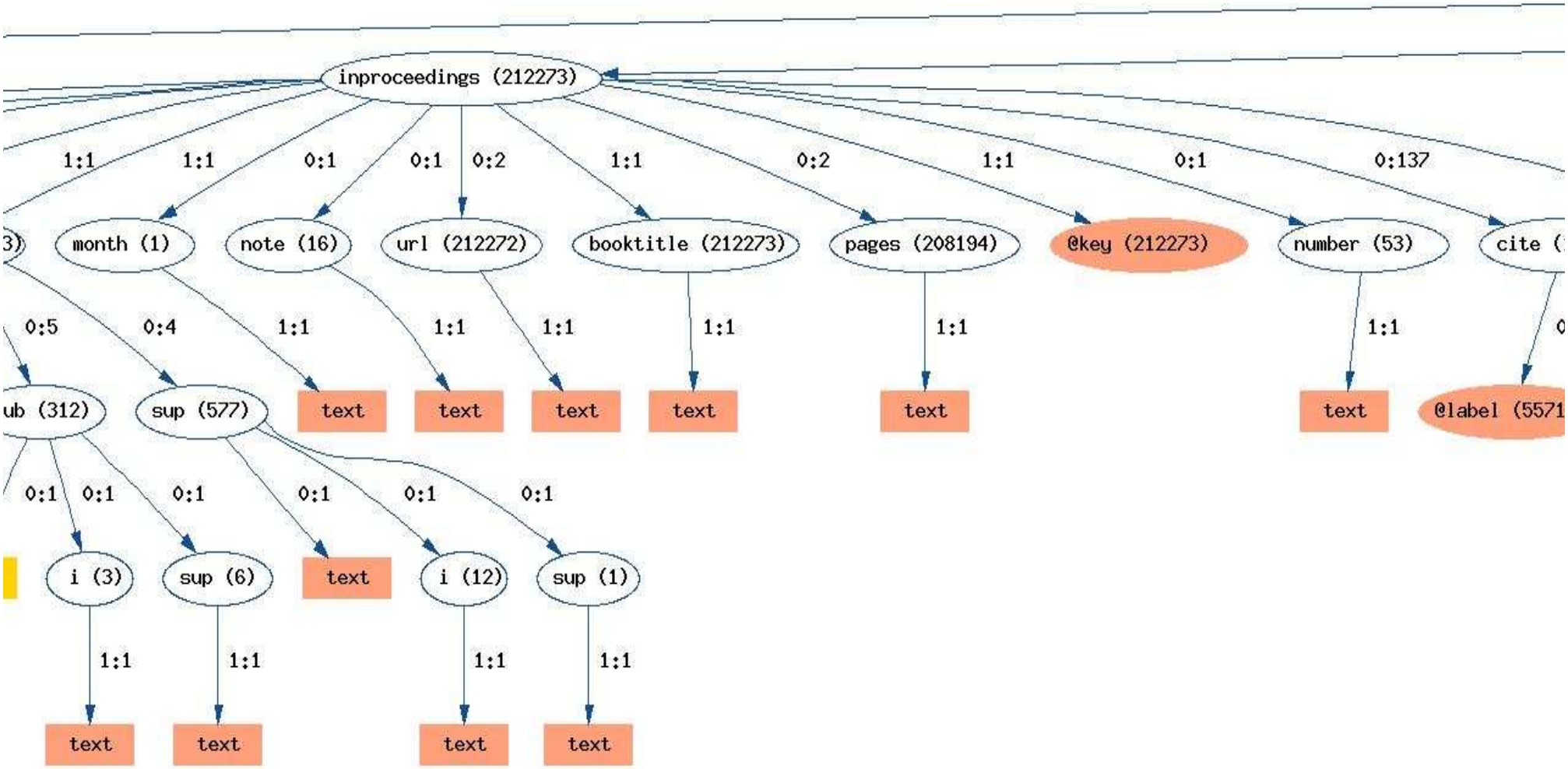
Some files have DTDs

Berkeley DTD: about 700 element types, Bib Duisburg: 56, CompuScience 60, DBLP: 40, HCI: 40, QMUL: 30

Trying to grasp the complexity: extracted XSum from our previous prototype XQueC, added new functionalities

An image is worth a thousand words

DBLP path summary produced by XSum



XSum path summaries

+ No need of a DTD/XML Schema

One node for each path in the incoming XML document (strong DataGuides)

Information:

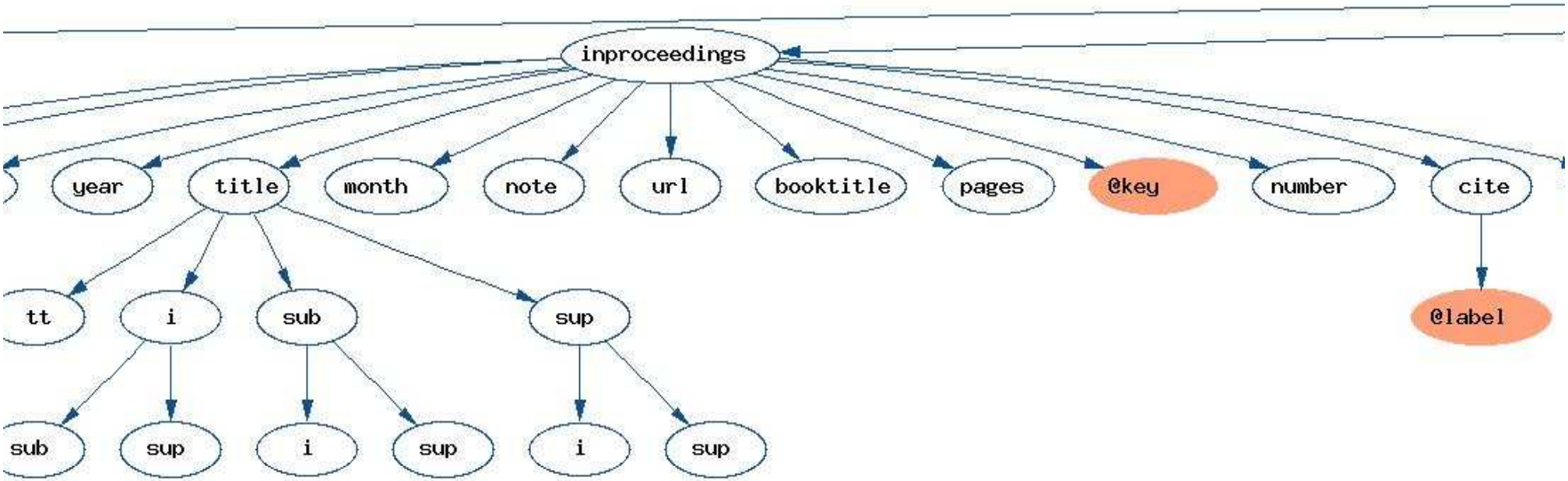
+ Structure

Inferred leaf data types

Cardinalities

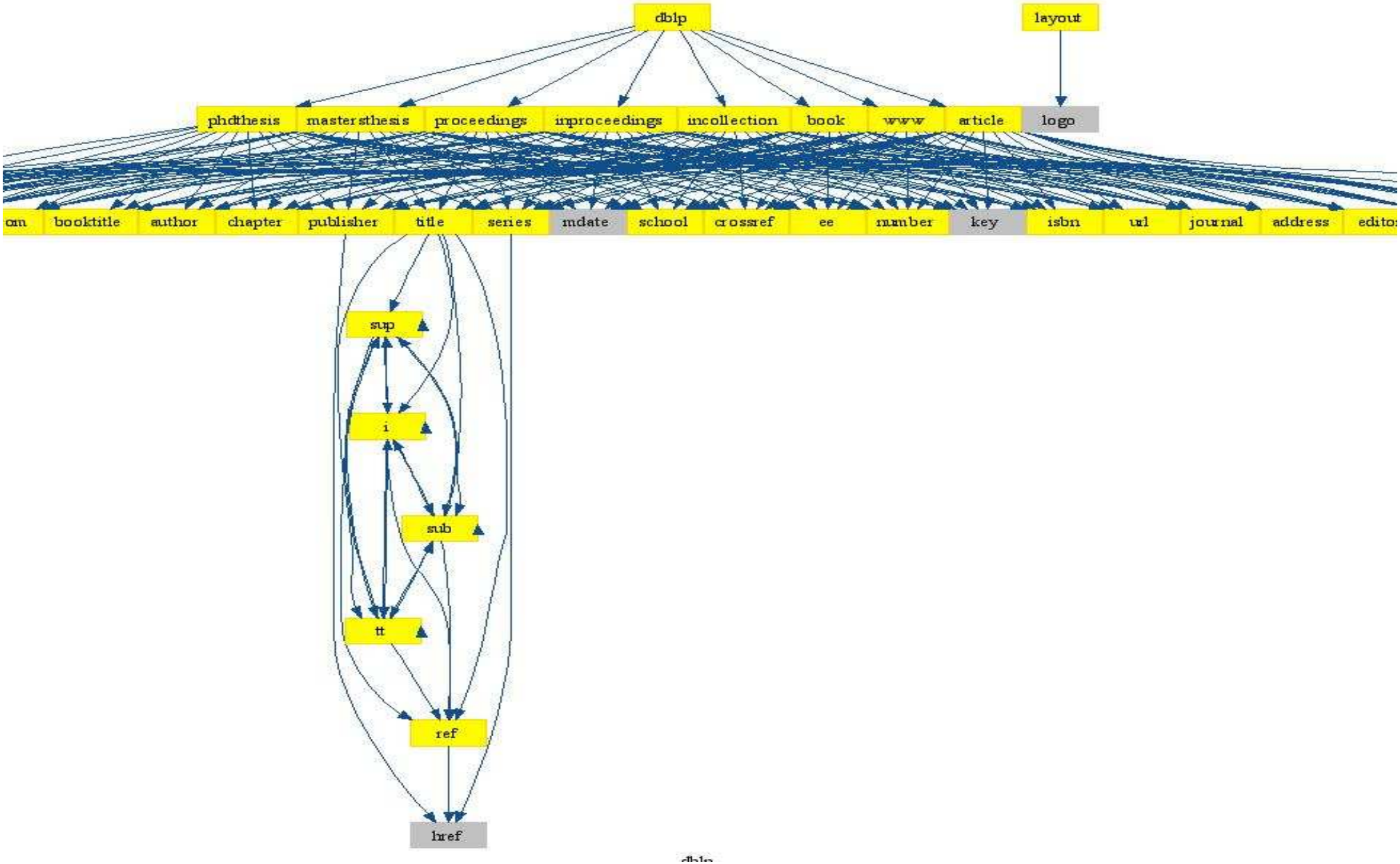
- Complex XML documents lead to very large (typically, wide) path summaries

Simpler path summaries



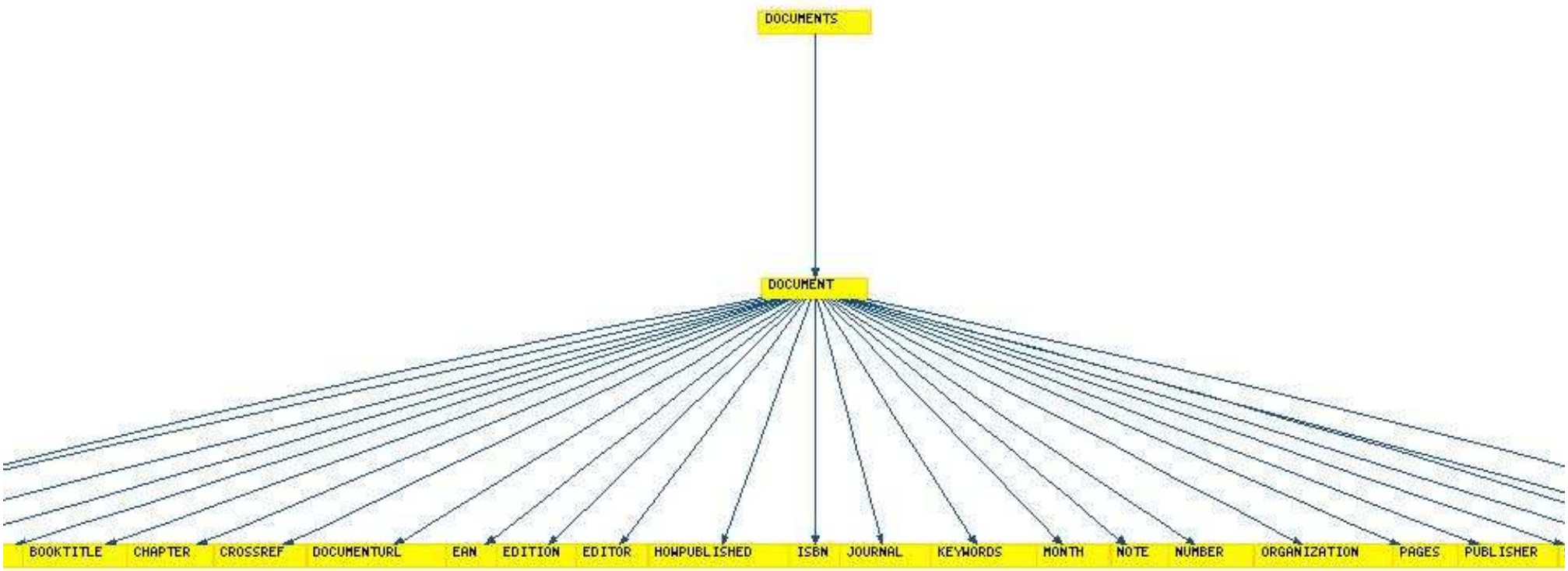
Another idea: DTD graphs

DBLP DTD graph produced by XSum:



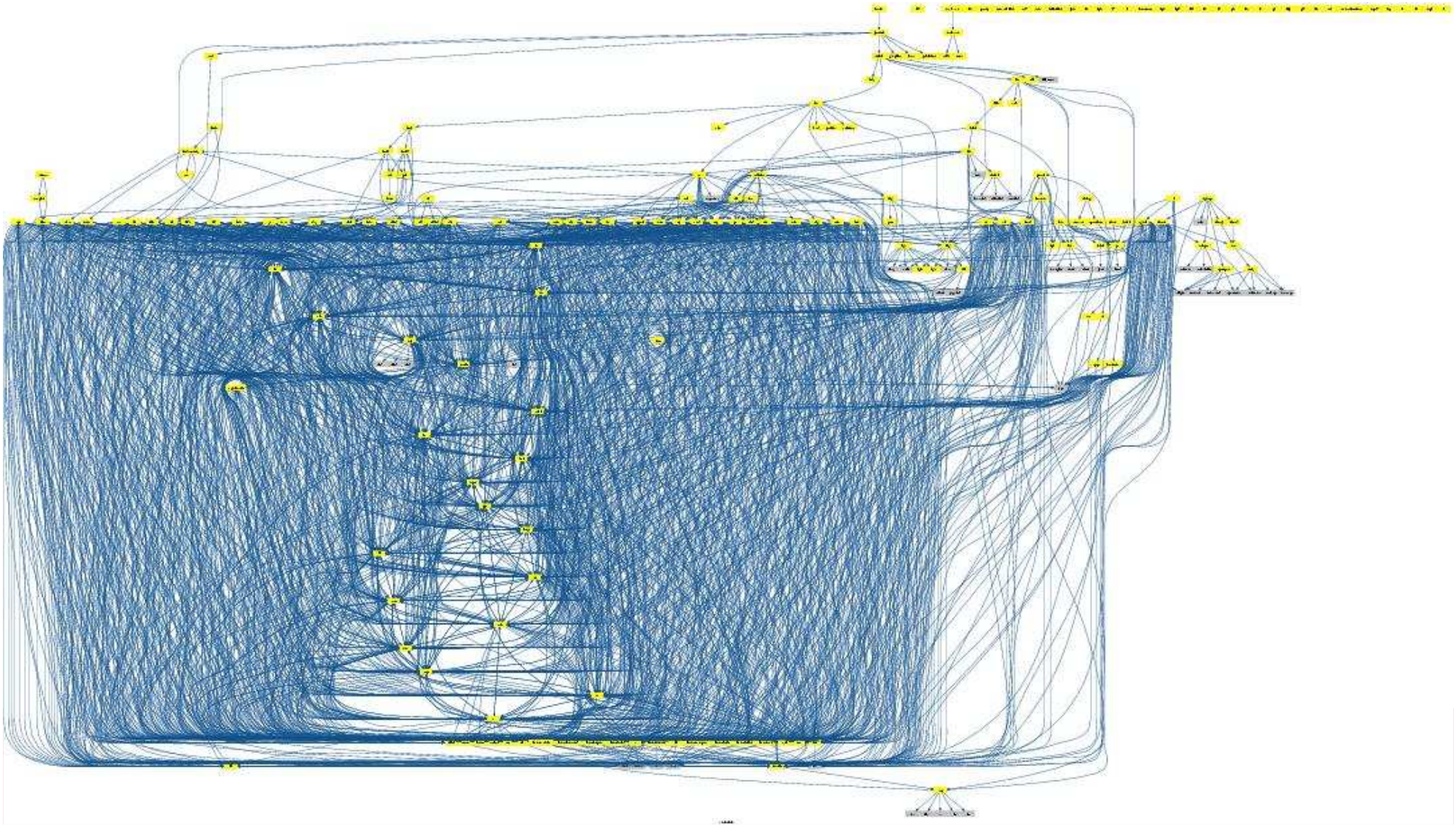
DTD graphs, another example

DTD graph for the QMUL data source:



DTD graphs, another example

DTD graph extracted from xmlarticle.dtd (RF track):



DTD graphs

One node for each type in the DTD

If DTD unavailable, we derive one from the data

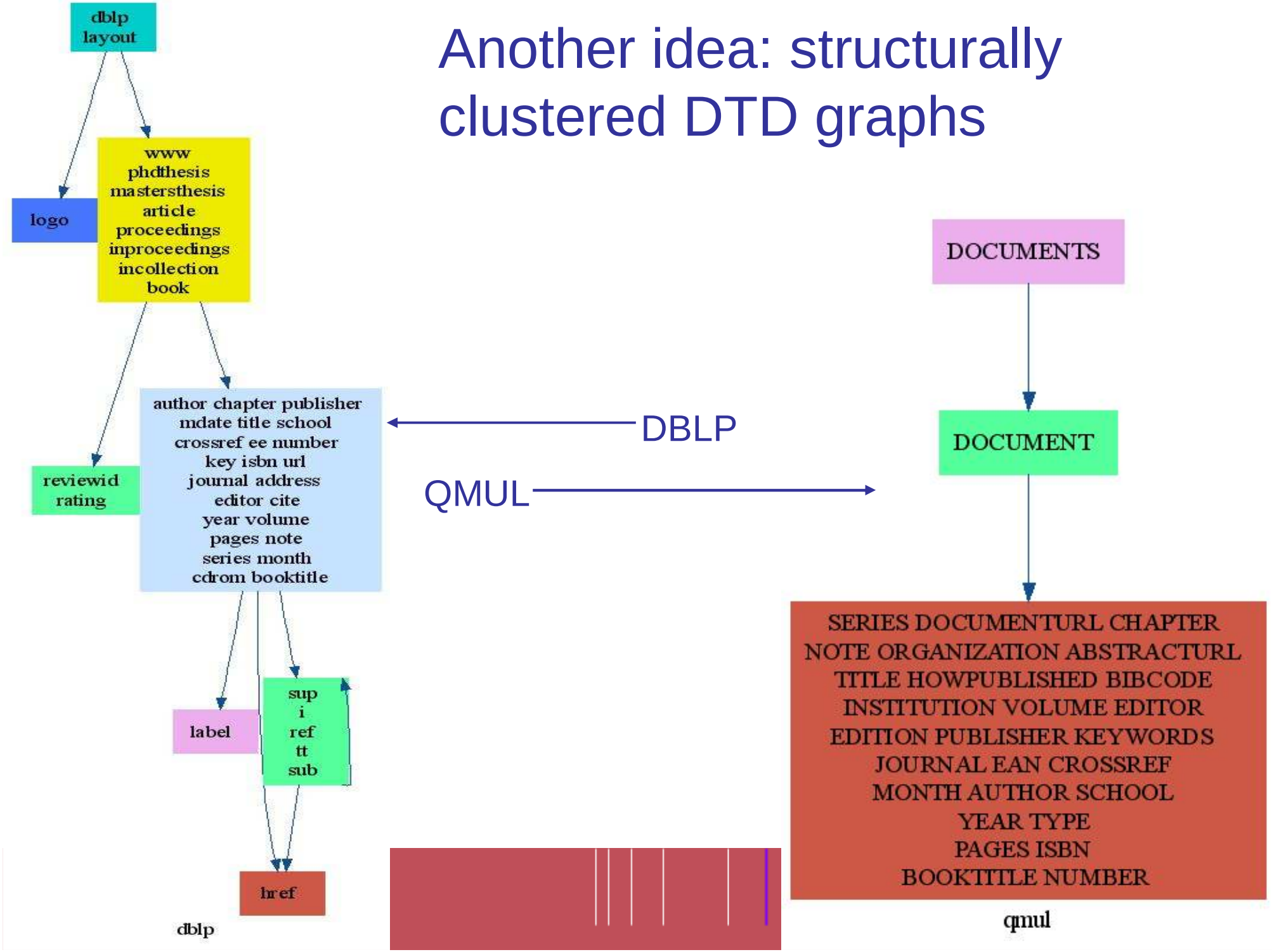
Information:

Structure

+ Fewer nodes

- Many edges wrt nodes. Edge crossings make graphs unredable. No way to avoid these crossings.

Another idea: structurally clustered DTD graphs



Structurally clustered DTD graphs

Collect in a cluster all DTD types having the same set of parents

Appearing in the same contexts

One node for each structural cluster

Information:

Structure (clusters)
Some information loss wrt DTD graph

Level-by-level DTD structure

Isolates mutual recursion in individual clusters (~ strongly connected components)

Our experience with XSum

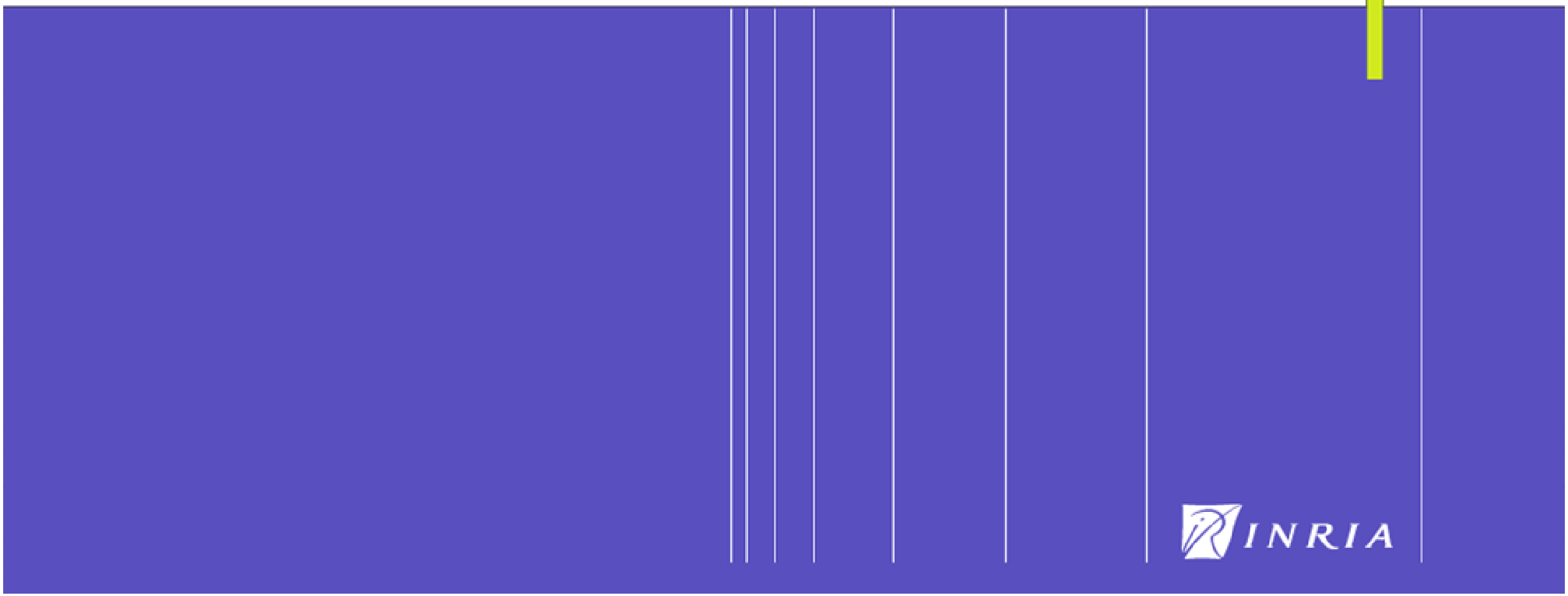
Structurally clustered XML graphs are most useful

They require some understanding

XSum: Java-based, freely available (Web site, mailing list, CD)

Currently investigating INRIA legalese to make it really open source

Unified DTD for het-track collection



Approach for building an unified DTD + mappings

Gather all individual types from the 6 DTDs

Build semantic clusters of types from various DTDs

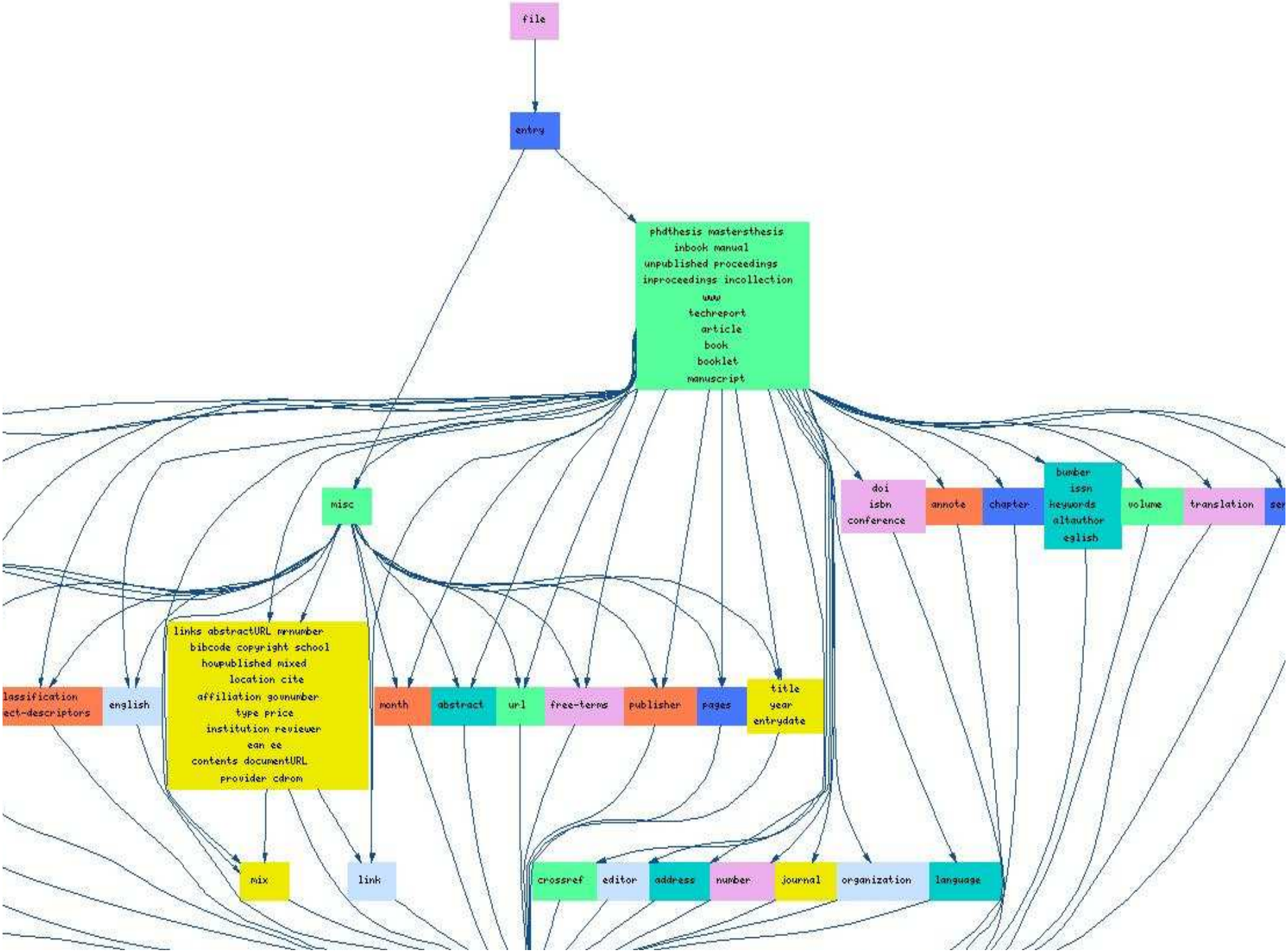
- Use WordNet to compute similarity between type names

Construct one type in the unified DTD for each semantic cluster

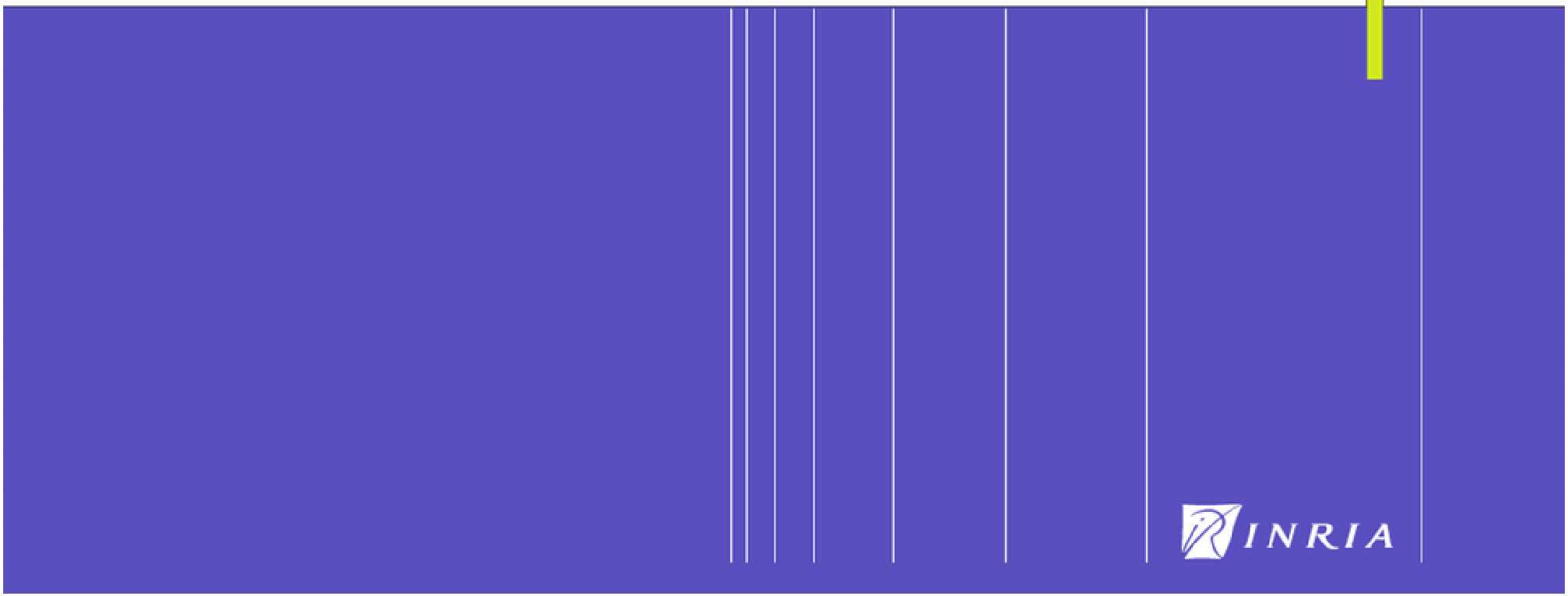
- Manual intervention to judge similarity threshold

Create edges in the unified DTD resulting from the edges in the individual DTDs

Create one mapping tSource isA tUnified for each source type tSource and resulting semantic cluster tUnified


file
entry
phdthesis mastersthesis
inbook manual
unpublished proceedings
inproceedings incollection
www
techreport
article
book
booklet
manuscript
misc
doi
isbn
conference
annote
chapter
bumber
issn
keywords
altauthor
eglish
volume
translation
links abstractURL mrnumber
bibcode copyright school
howpublished mixed
location cite
affiliation govnumber
type price
institution reviewer
ean ee
contents documentURL
provider cdrom
lassification
ect-descriptors
english
month
abstract
url
free-terms
publisher
pages
title
year
entrydate
mix
link
crossref
editor
address
number
journal
organization
language

Running CAS topics on the het-track collection

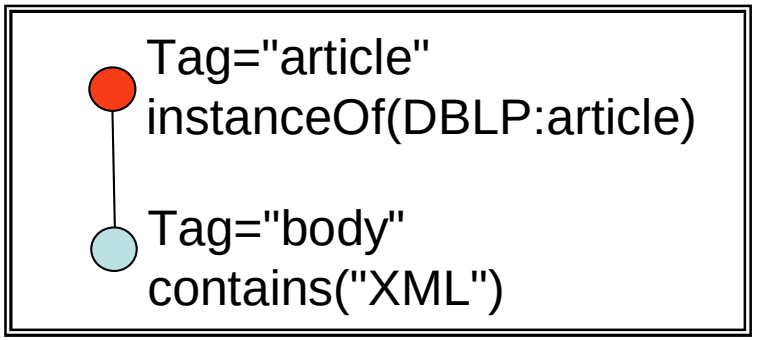


Structure and semantic queries in KadoP

KadoP: integration platform for XML documents and semantic information

Documents: dblp.xml, dxf2.xml
Tags: article, DBLP:article
Words: XML, database,
Concepts: DBLP:article, DBLP:title, QMUL:DOCUMENT
Relationships: DBLP:article IsA Unified:article
DBLP:title partOf DBLP:article
DBLP://article instanceOf DBLP:article

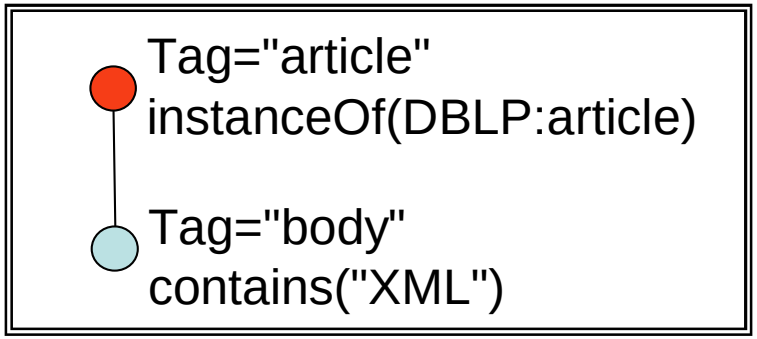
Queries over all data sources:

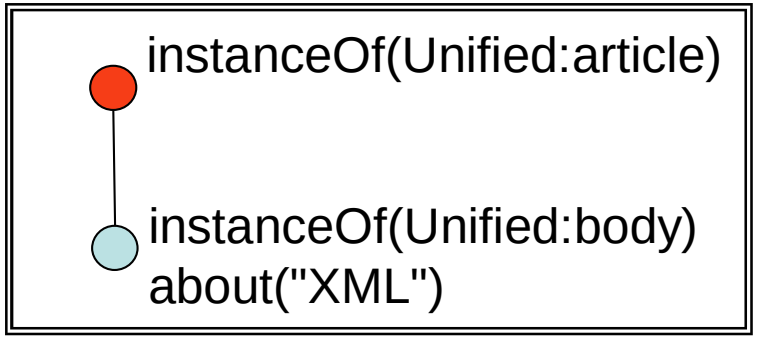


Solving CAS topics over the het-track collection

KadoP: integration platform for XML documents and semantic information

Queries over all data sources:





Look for x instanceOf Unified:article → nothing
Look for y IsA Unified:article → DBLP:article, QMUL:article...
Look for z instanceOf y → dblp.xml//article, qmul.xml//article...

Solving CAS topics over the het-track collection

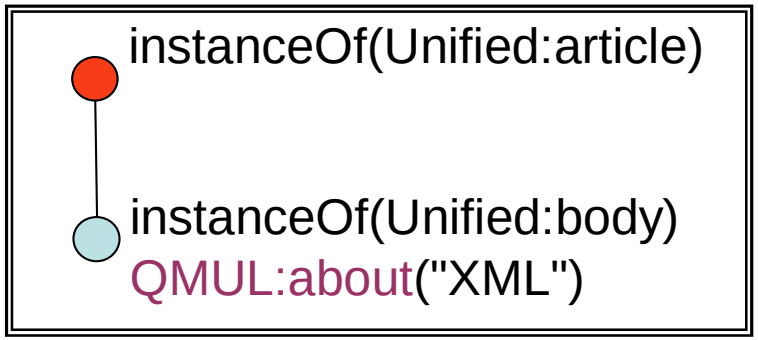
Possible scenarios

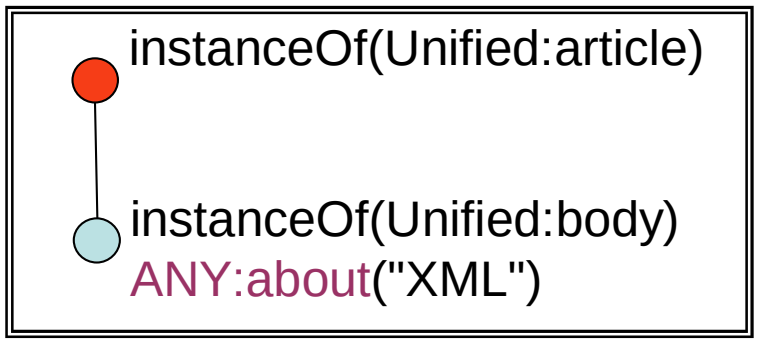
- Distributed data sources
- Centralized data sources

The about( ) predicate

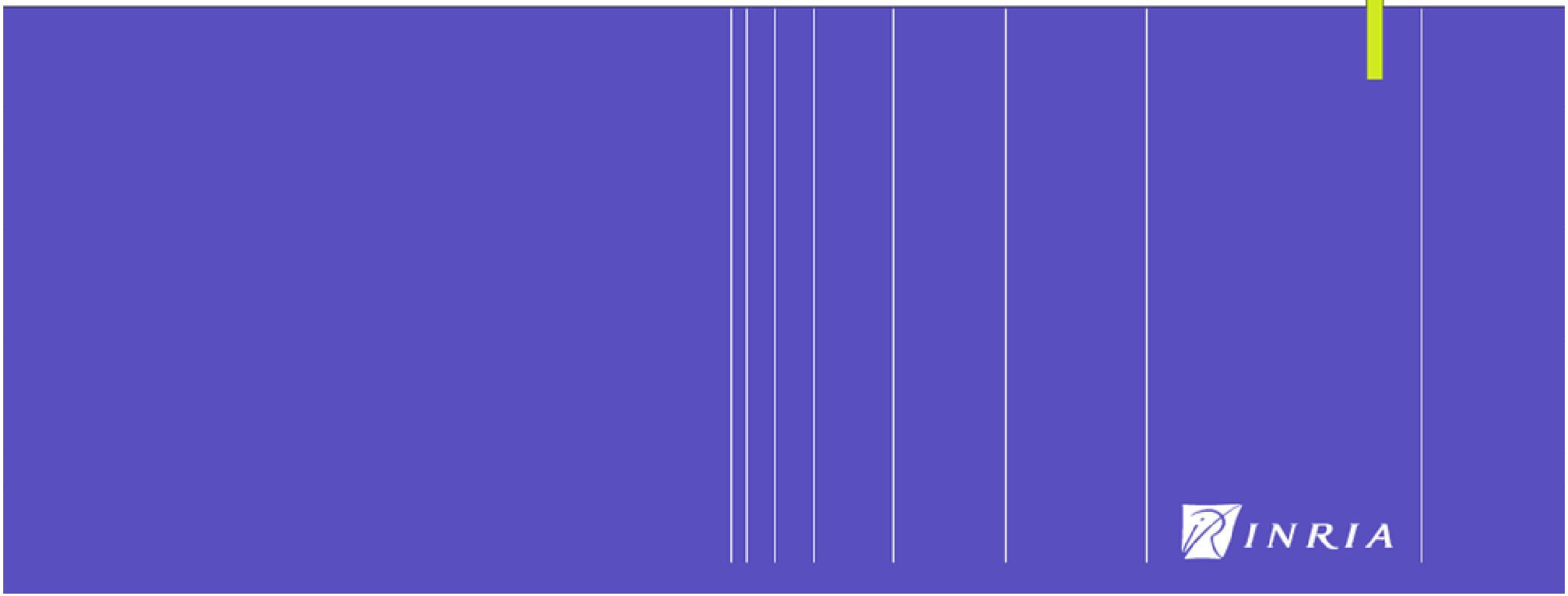
- Can be plugged in as a Web service

Possible queries:

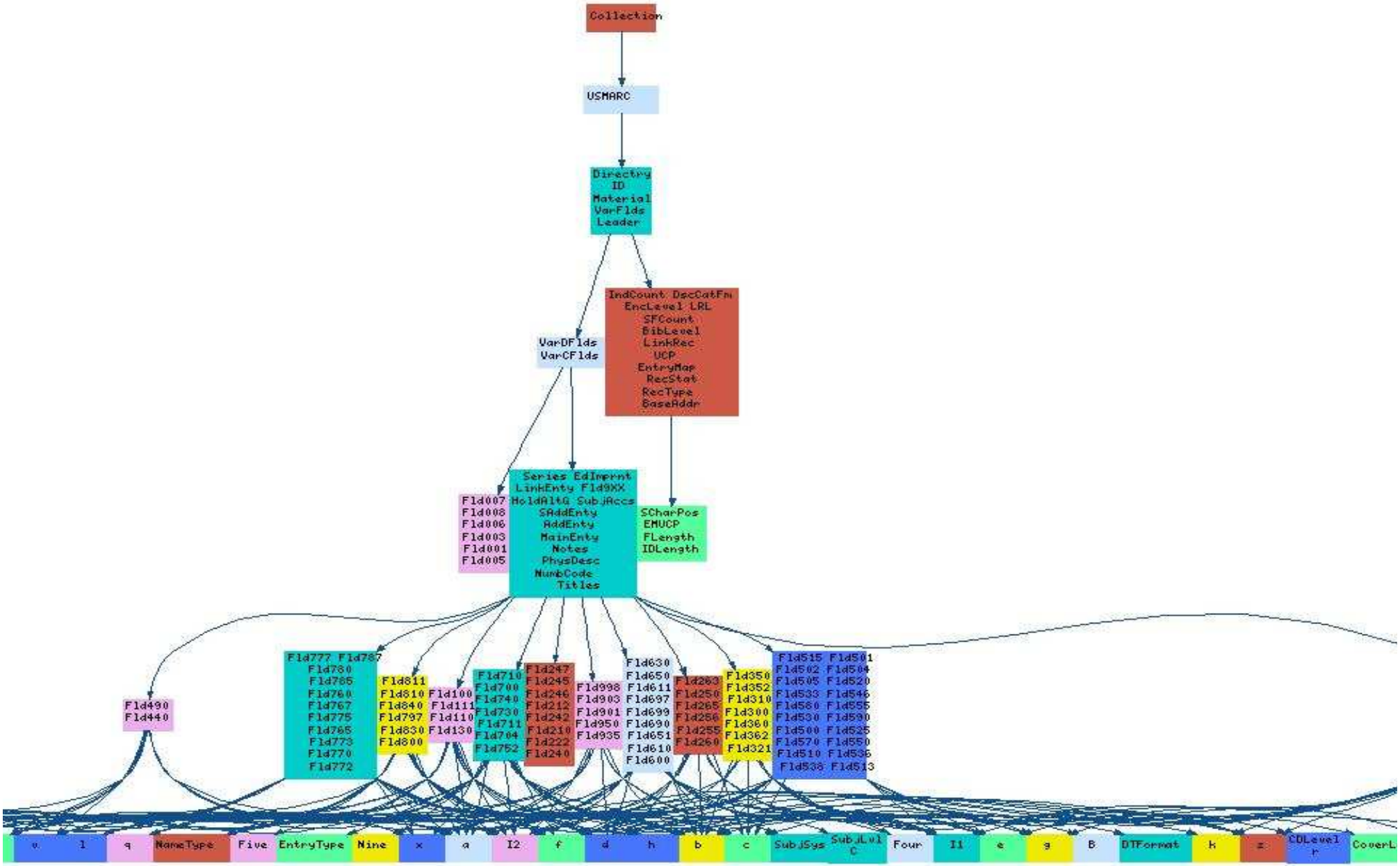




Perspectives



Semantic integration of the Berkeley data set



The Berkeley data set

The semantics is in comments preceding the type

Derived DTD has 200 types, original DTD has 700

Structural clusters:

- Sometimes group logically similar types
- More often, group types of the same provenance

Few types completely lack semantics

KadoP platform deployment

Must decide on

- Deployment architecture: single server ? per-participant ?
- Interface(s) for about()
- Other issues

Thank you