Creation, Population and **Preprocessing** of **Experimental Data Sets** for **Evaluation** of **Applications** for the **Semantic Web**

G. Frivolt, J. Suchal, R. Veselý,
P. Vojtek, O. Vozár, M. Bieliková

Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology

Motivation

- Lack of suitable data sets for experimental evaluation of semantic web oriented applications (faceted browser)
- Preserve as much as possible information from original data sources
- Existing data sets miss (or contain sparse) meta-data

Goals

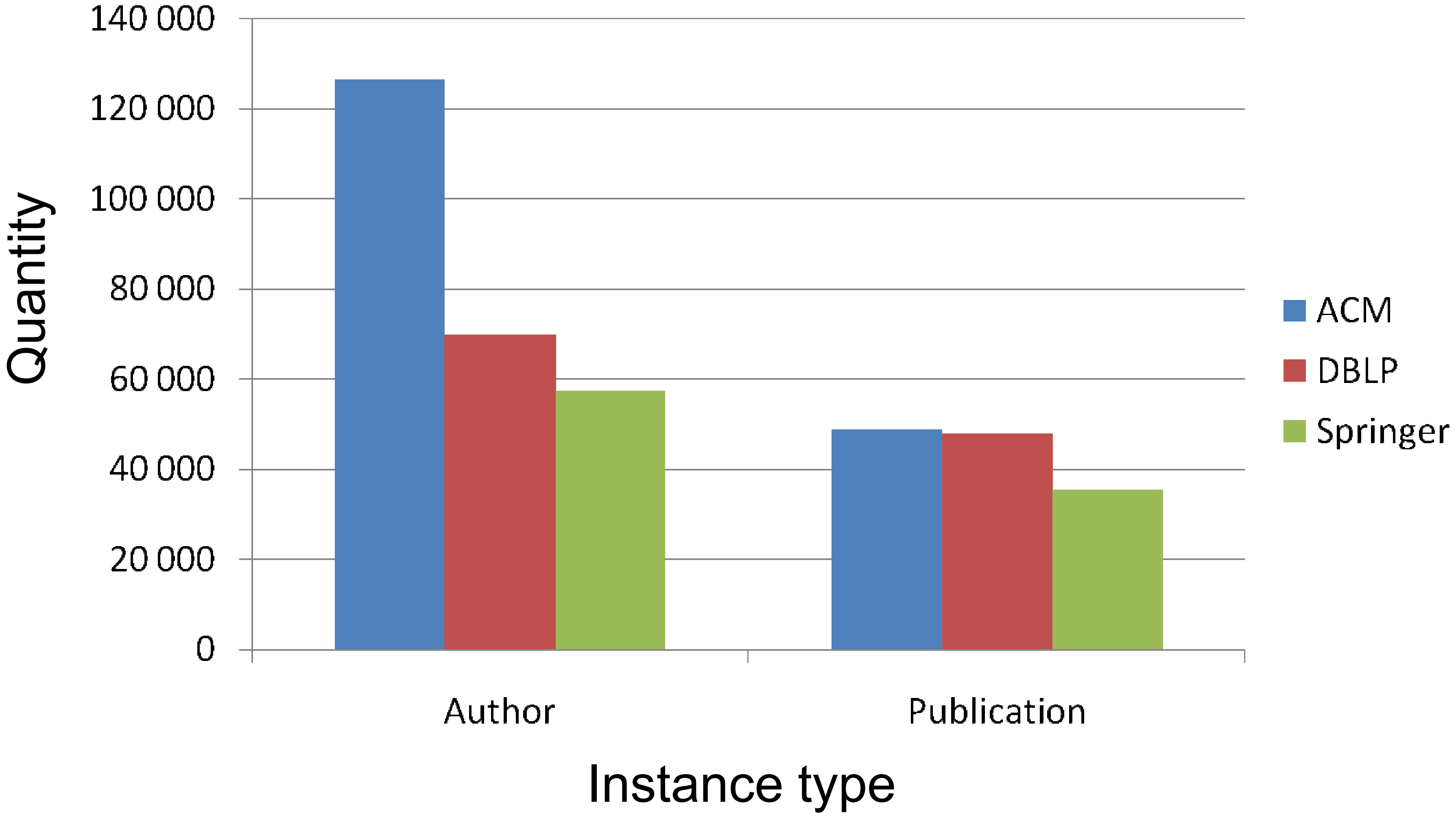
- Project MAPEKUS[1]
  - create semantic layer over digital libraries
  - background for inferencing
  - analysis of social networks
- Improve quality of obtained data
  - identify duplicated and malformed data
- Provide visual navigation in data set

[1] http://mapekus.fiit.stuba.sk

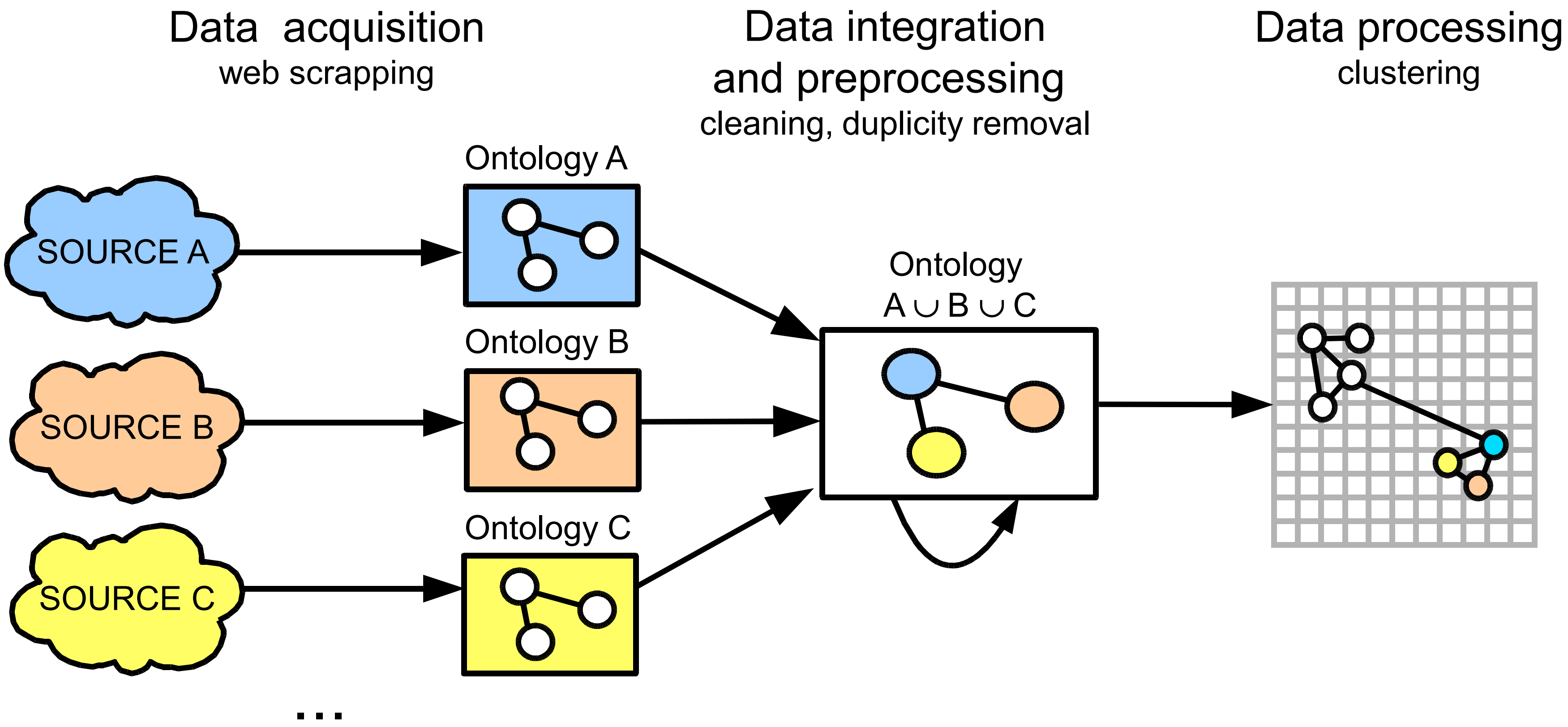
Domain Description

- Data from scientific publications domain
- Digital libraries:
  - **ACM** www.acm.org
  - **Springer** www.springer.com
- Meta-data repository:
  - **DBLP** www.informatik.uni-trier.de/~ley/db/

Domain Description



Data Process Flow

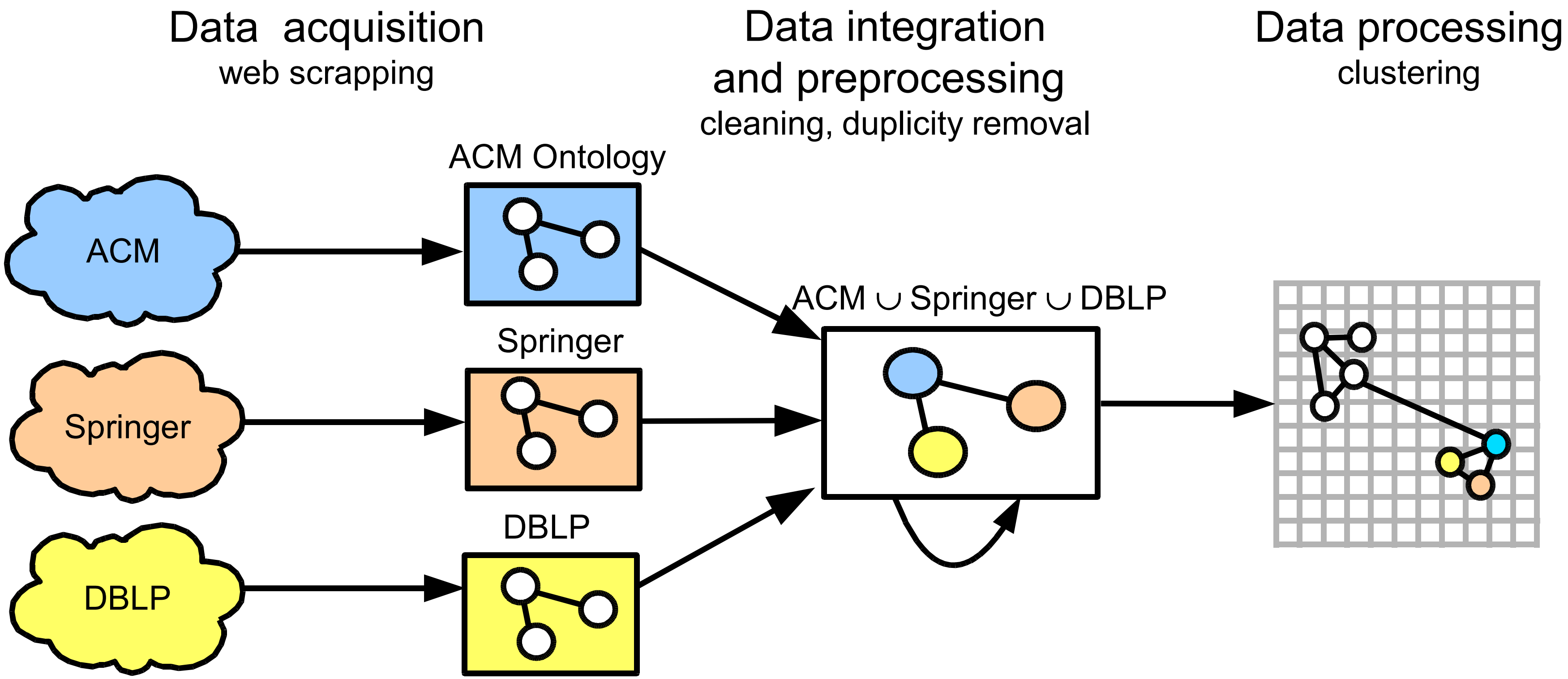


Data acquisition
web scrapping

Data integration and preprocessing
cleaning, duplicity removal

Data processing
clustering

Data Process Flow

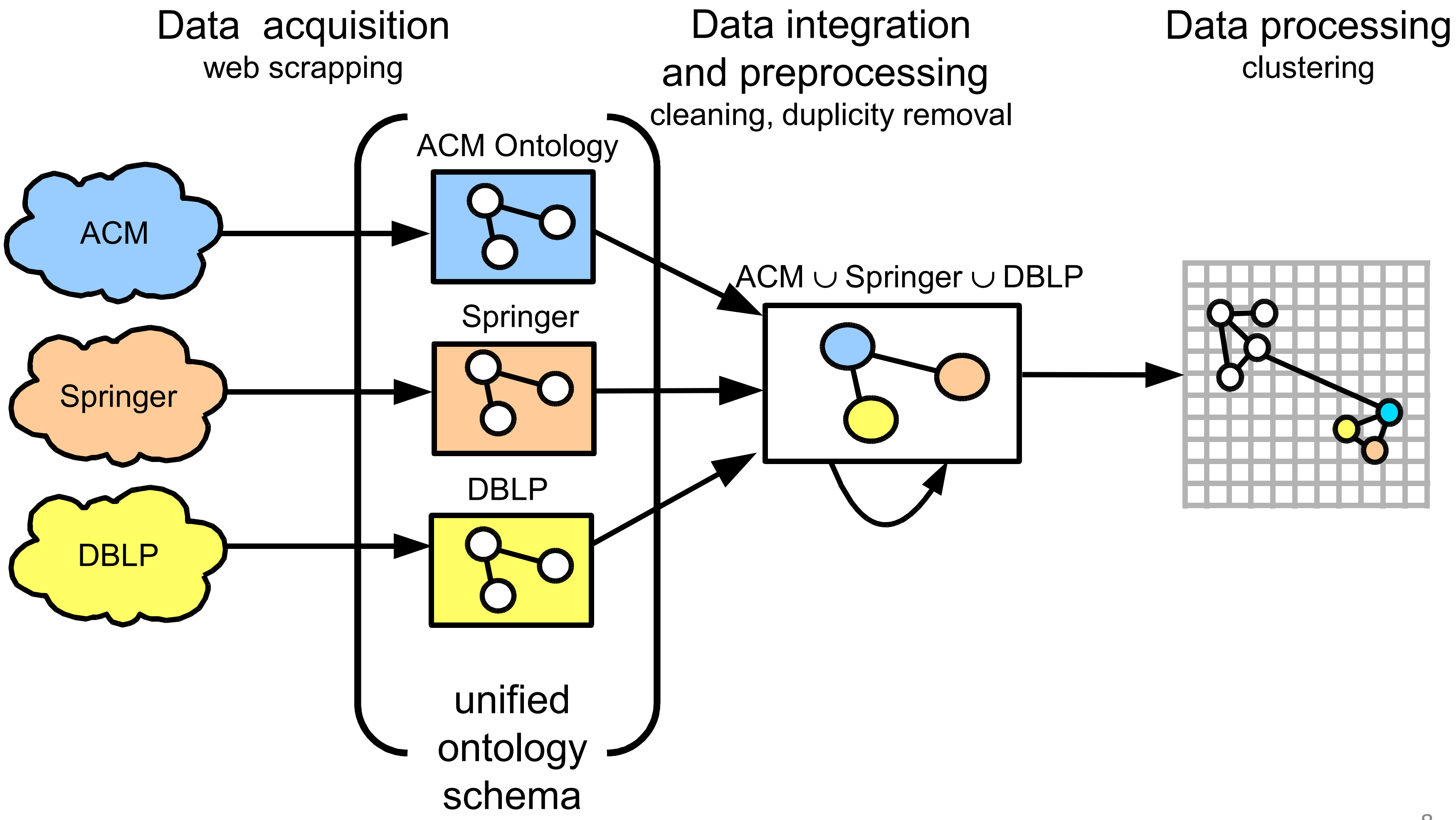


Data acquisition
web scrapping

Data integration and preprocessing
cleaning, duplicity removal

Data processing
clustering

Data Process Flow



Data Acquisition

- How did we gather data?
  - wrapper induction by giving positive and negative examples of patterns on the web pages
- Wrapper induction exploits machine learning techniques for generalization of patterns
  - XPath based learning of patterns
  - generalization of patterns' attributes using Bayesian networks
- Gathered data stored in structured form in an ontological repository

Data Acquisition

- Wrapped data (depends on data source):
  - publication instances: name, abstract, year
  - publication categories, topics and keywords
  - authorship relation
  - **isReferencedBy** and **references** relations between publications

Data Preprocessing

- Why to clean data?
  - inconsistencies:
    - in source data (name misspelling, diacritics)
    - inconsistencies created during wrapping process
    - source integration (same author in two sources) – relevant for social networks of authors
- Non-invasive cleaning
  - tagging inconsistent data (without removal)

Data Preprocessing

Single-pass instance cleaning

- Cleaning in the scope of one instance (without relations)
- Set of filters, each filter for particular purpose:
  - correcting capital letters in names and surnames
  - separating first names and surnames
- One pass through all instances – linear time complexity

Data Preprocessing
Duplicate identification

- Combination of two methods:
  - comparison of data properties (e.g., author names, publication titles)
  - comparison of object properties (e.g., coauthors, references, relations between author and publication)

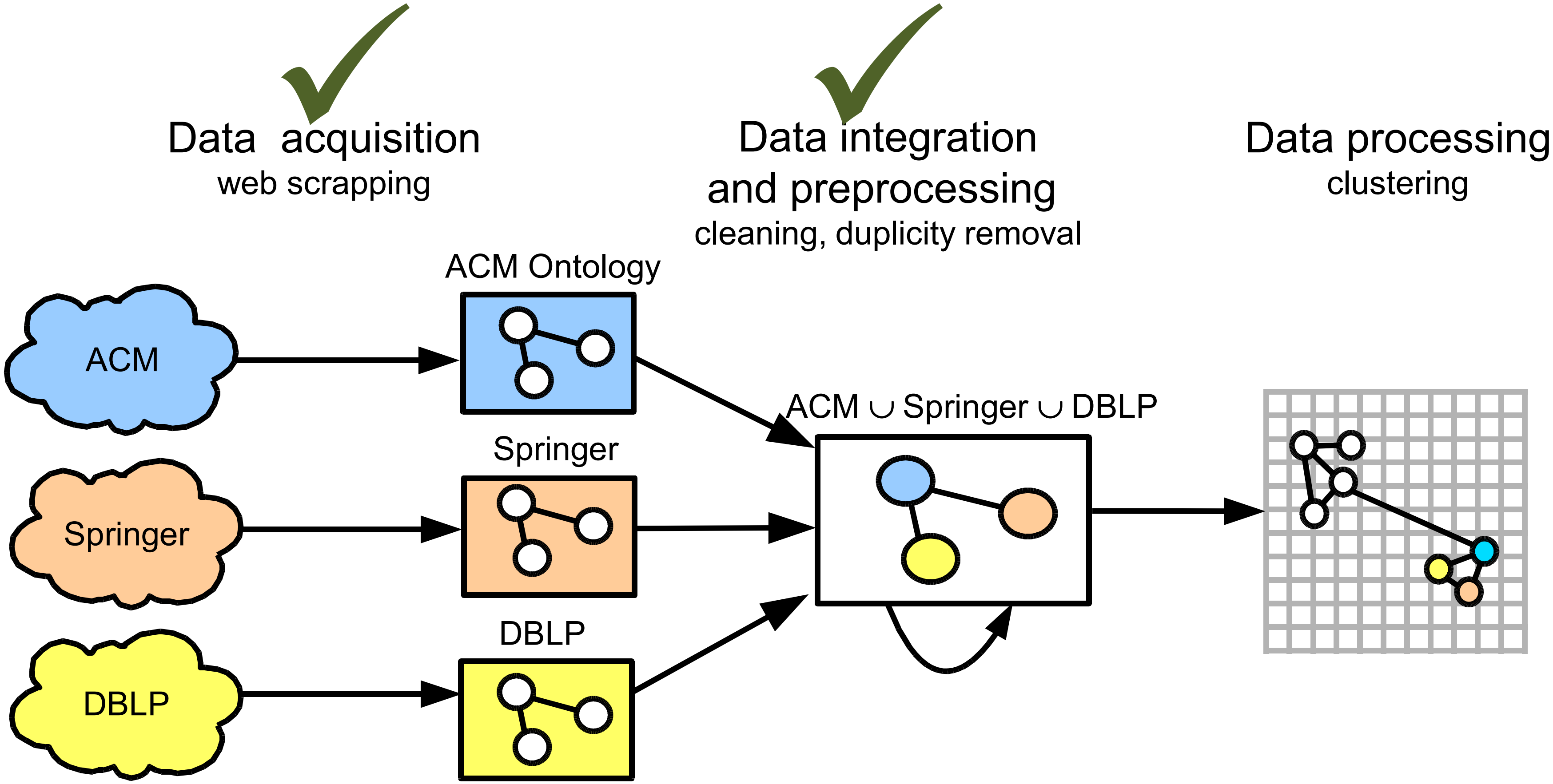
Duplicate Identification
Data properties comparison

- using standard string metrics like
  - Levenstein distance
  - Monge-Elkan
  - N-grams
- special string metrics
  - distance of different characters on keyboard
  - name metrics, considering abbreviations
    (J. Smyth = John Smyth)

Duplicate Identification
Object properties comparison

- Object properties comparison
  - for each object property the similarity is computed from number of matches
  - for example: similarity of two authors depends on number of conjoint co-authors

Data Process Flow



✓ Data acquisition
web scrapping

✓ Data integration
and preprocessing
cleaning, duplicity removal

Data processing
clustering

ACM

Springer

DBLP

ACM Ontology

Springer

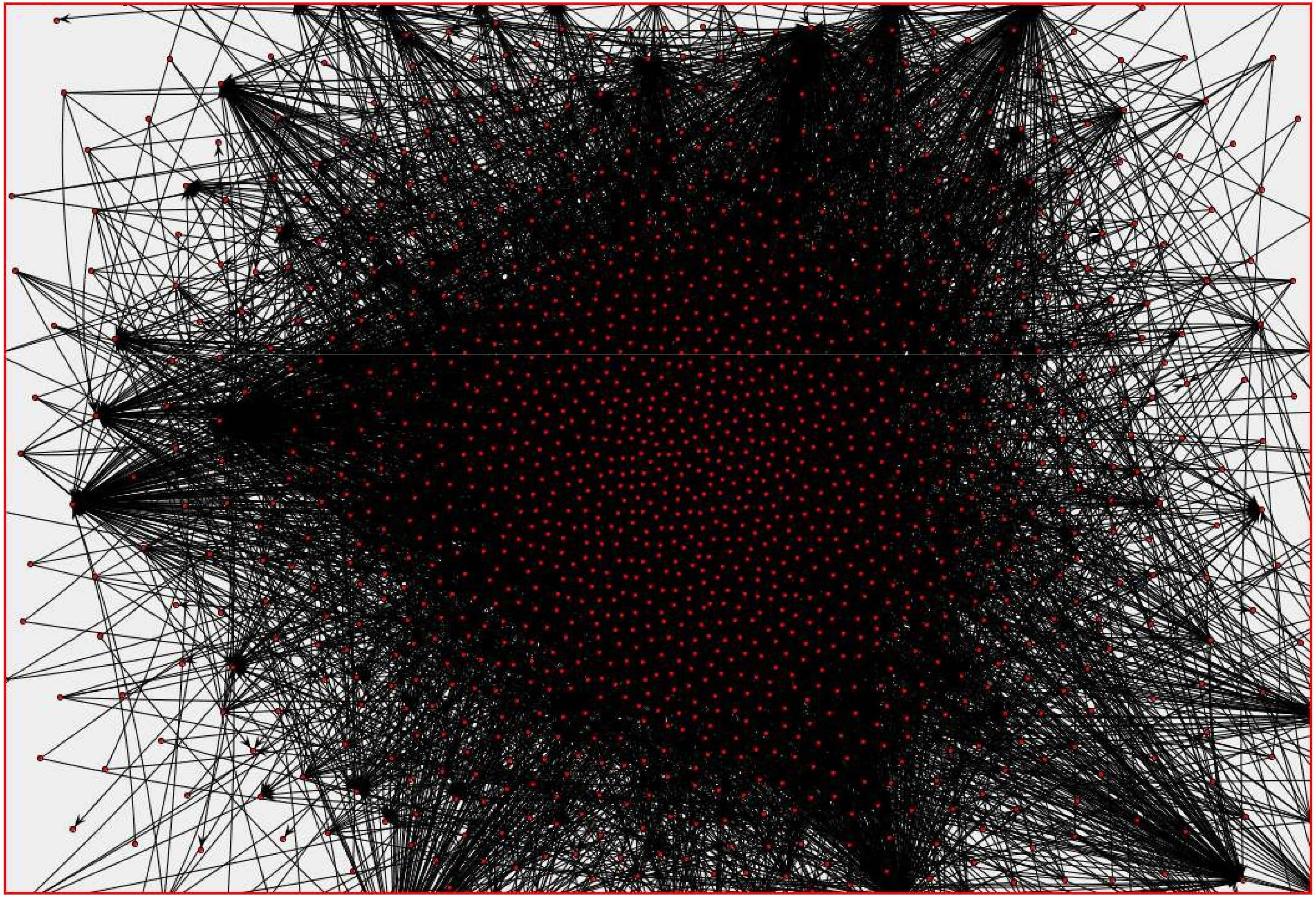
DBLP

ACM ∪ Springer ∪ DBLP

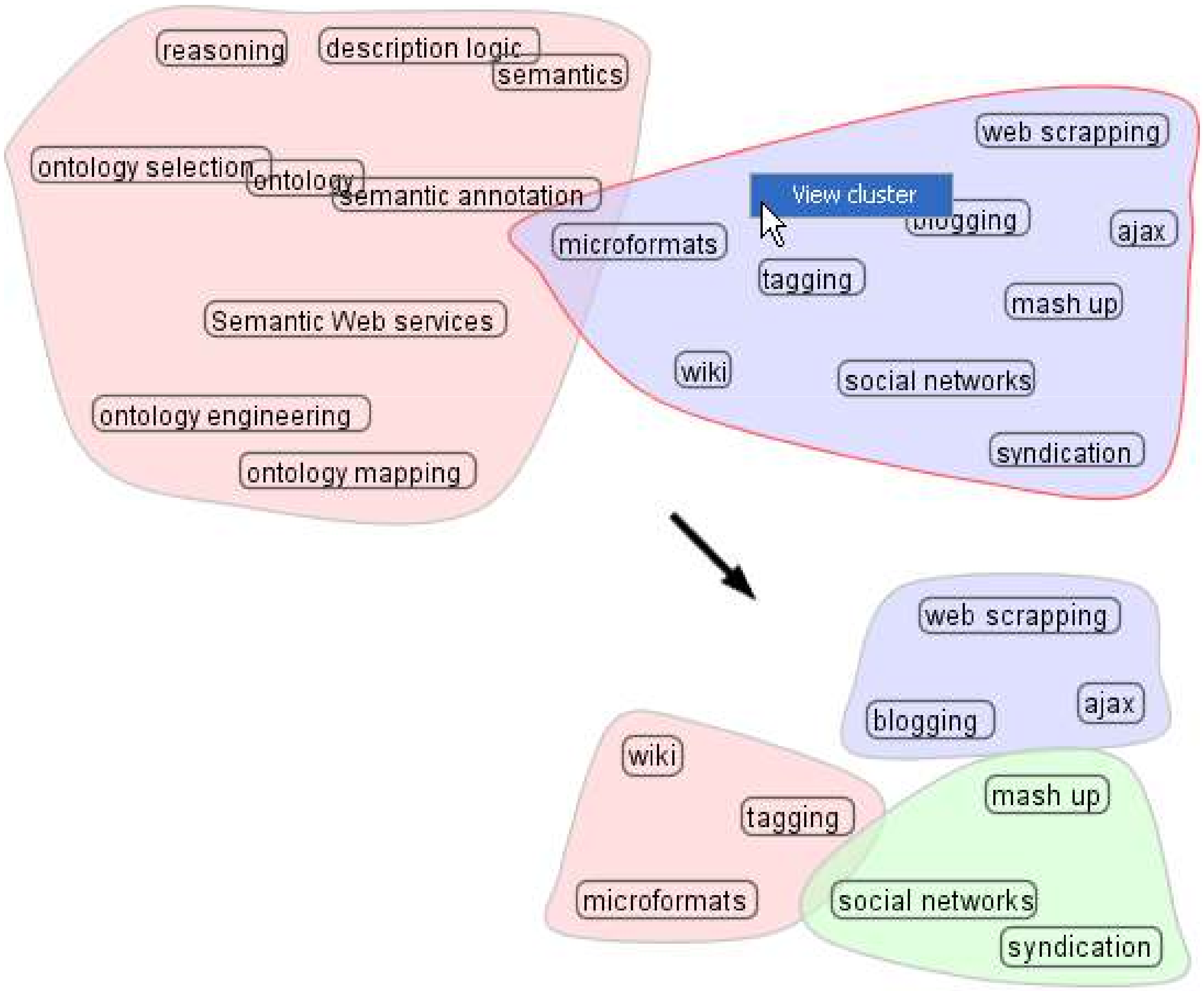
Graph Clustering

- Graph extraction from ontology
  - preparation for clustering
- Hierarchical clustering
  - clustering methods from JUNG library
  - layers generated using bottom-up approach
- Results stored in relational database
  - speed and simplicity

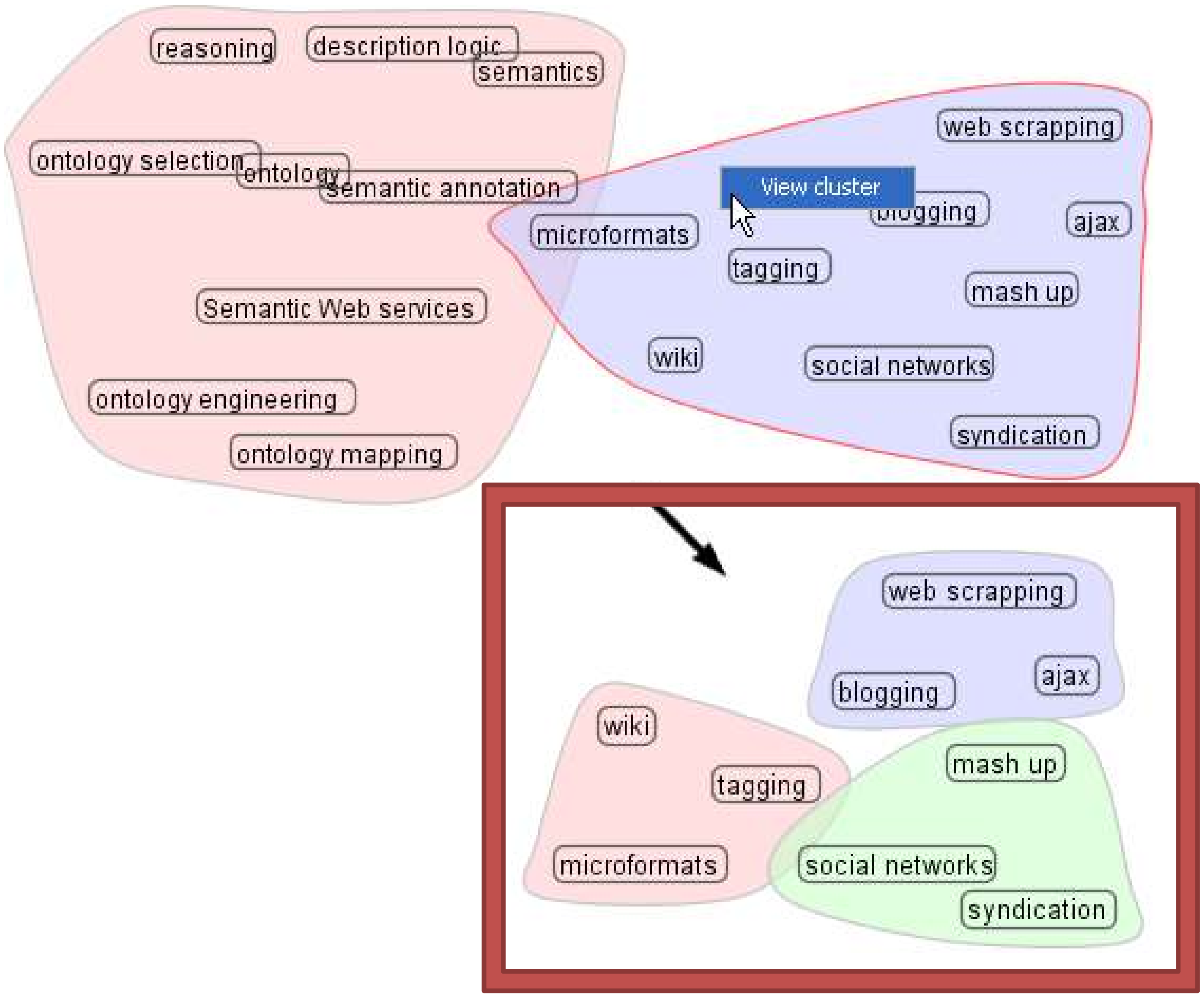
ACM visualization (1500 publications)



Graph Clustering



Graph Clustering

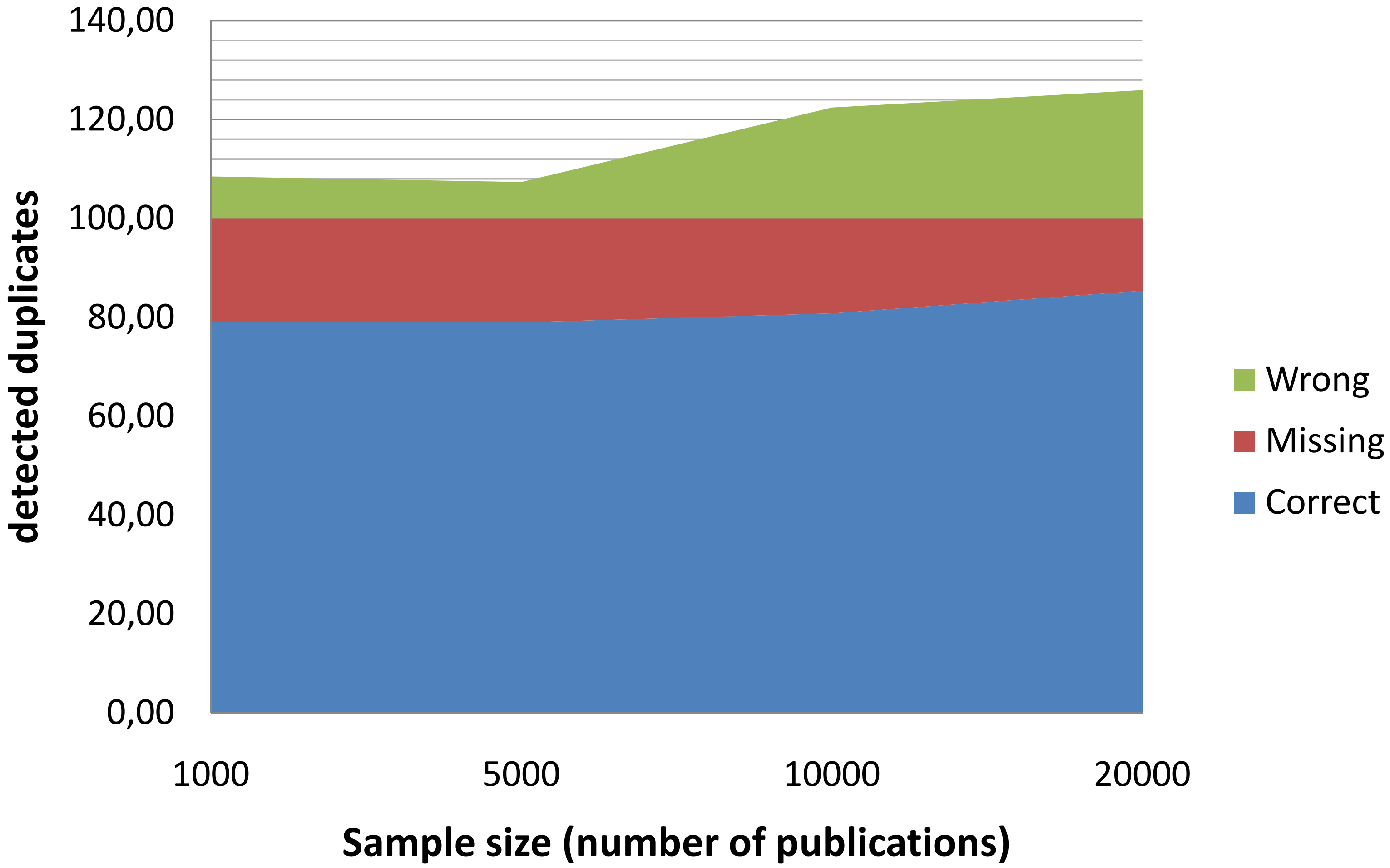


Evaluation – Identification of Duplicates

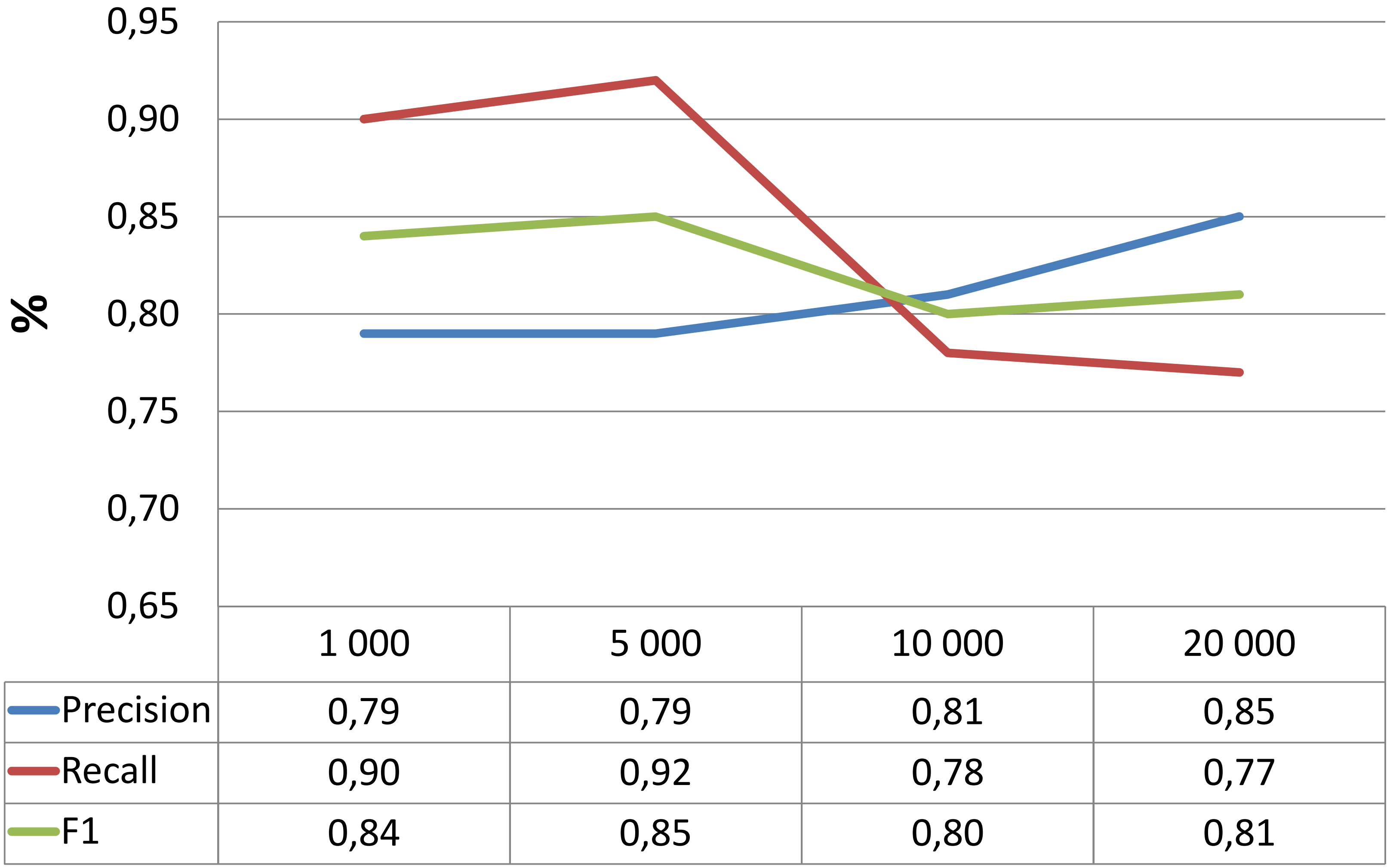
- Sampling (publications count) :

  1 000, 5 000, 10 000, 20 000
- 10 runs for each sample size
- Injected 100 generated duplicities
- All data from DBLP
- Duplicities already present in DBLP were ignored

Evaluation – Identification of Duplicates

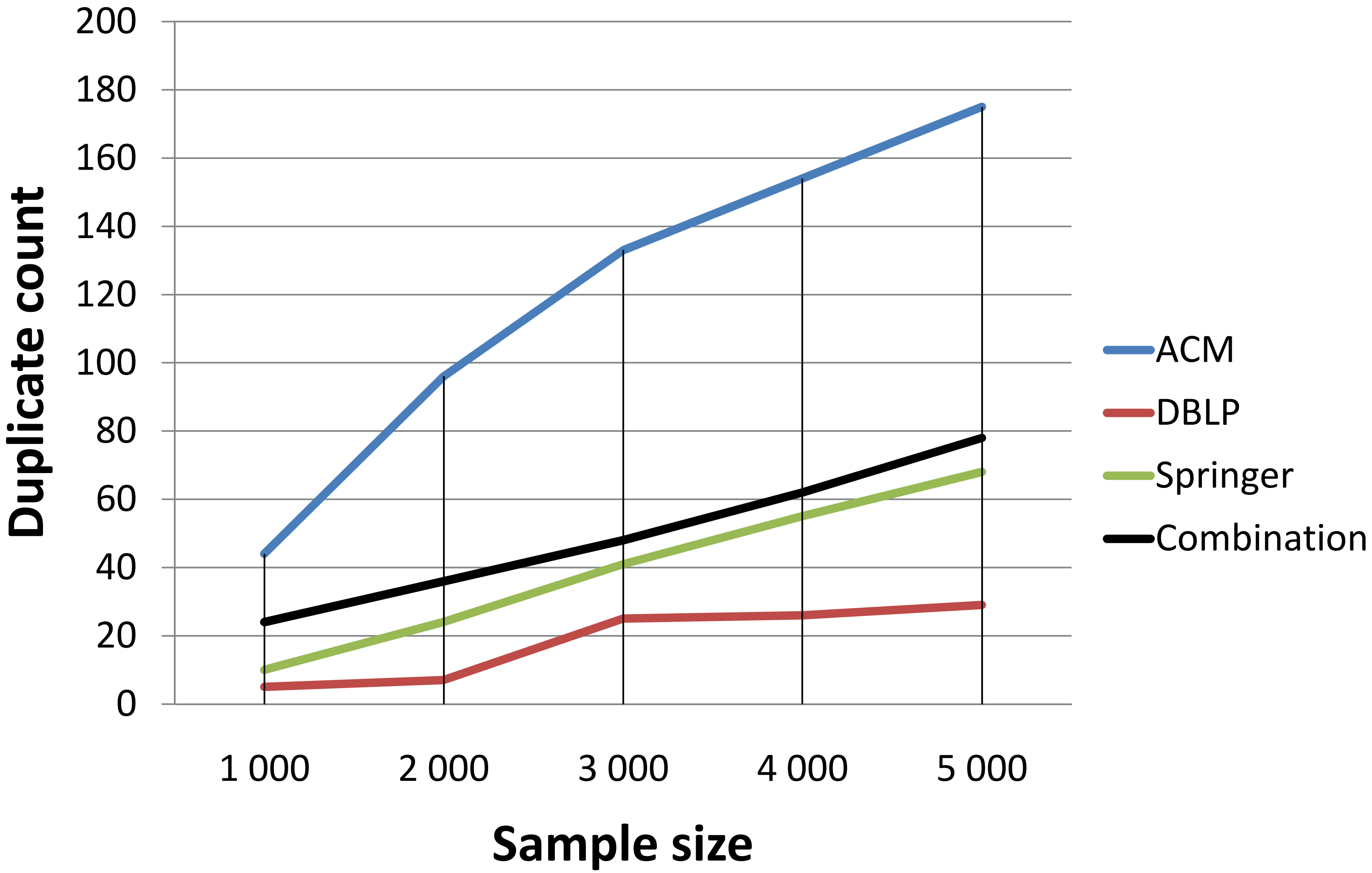


Evaluation – Identification of Duplicates



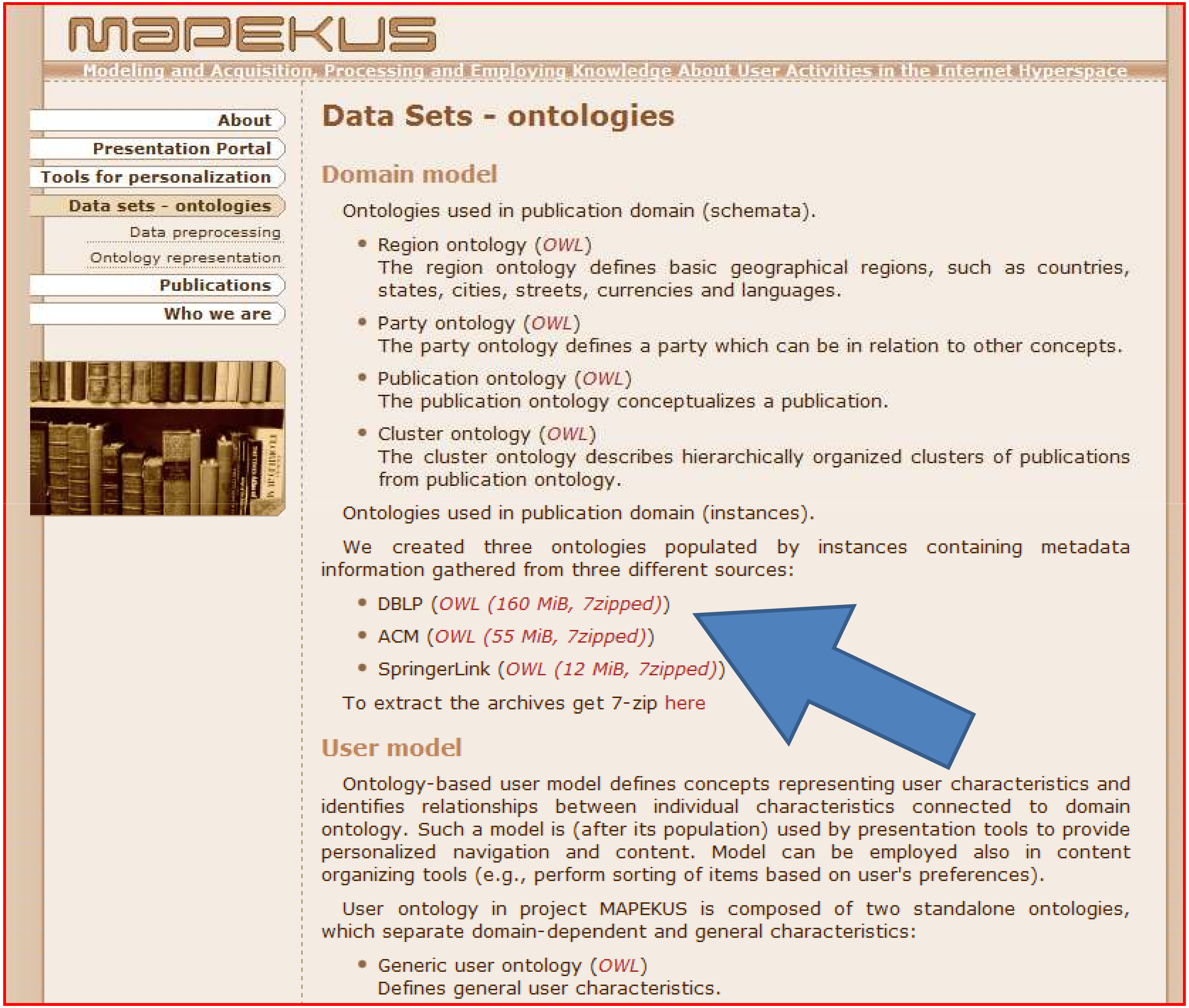
	1 000	5 000	10 000	20 000
Precision	0,79	0,79	0,81	0,85
Recall	0,90	0,92	0,78	0,77
F1	0,84	0,85	0,80	0,81

Evaluation – Identification of Duplicates in Real Data



Conclusions

- ACM, Springer and DBLP data sources were:
  - obtained via web scrapping
  - stored in meta-data preserving format (OWL)
  - available online: **http://mapekus.fiit.stuba.sk**
- Data evaluation:
  - data cleaning (duplicity identification)
  - case study of data set processing – cluster-based visual navigation



Modeling and Acquisition, Processing and Employing Knowledge About User Activities in the Internet Hyperspace

Data Sets - ontologies

Domain model

Ontologies used in publication domain (schemata).

- Region ontology (*OWL*)
  The region ontology defines basic geographical regions, such as countries, states, cities, streets, currencies and languages.
- Party ontology (*OWL*)
  The party ontology defines a party which can be in relation to other concepts.
- Publication ontology (*OWL*)
  The publication ontology conceptualizes a publication.
- Cluster ontology (*OWL*)
  The cluster ontology describes hierarchically organized clusters of publications from publication ontology.

Ontologies used in publication domain (instances).

We created three ontologies populated by instances containing metadata information gathered from three different sources:

- DBLP (*OWL (160 MiB, 7zipped)*)
- ACM (*OWL (55 MiB, 7zipped)*)
- SpringerLink (*OWL (12 MiB, 7zipped)*)

To extract the archives get 7-zip here

User model

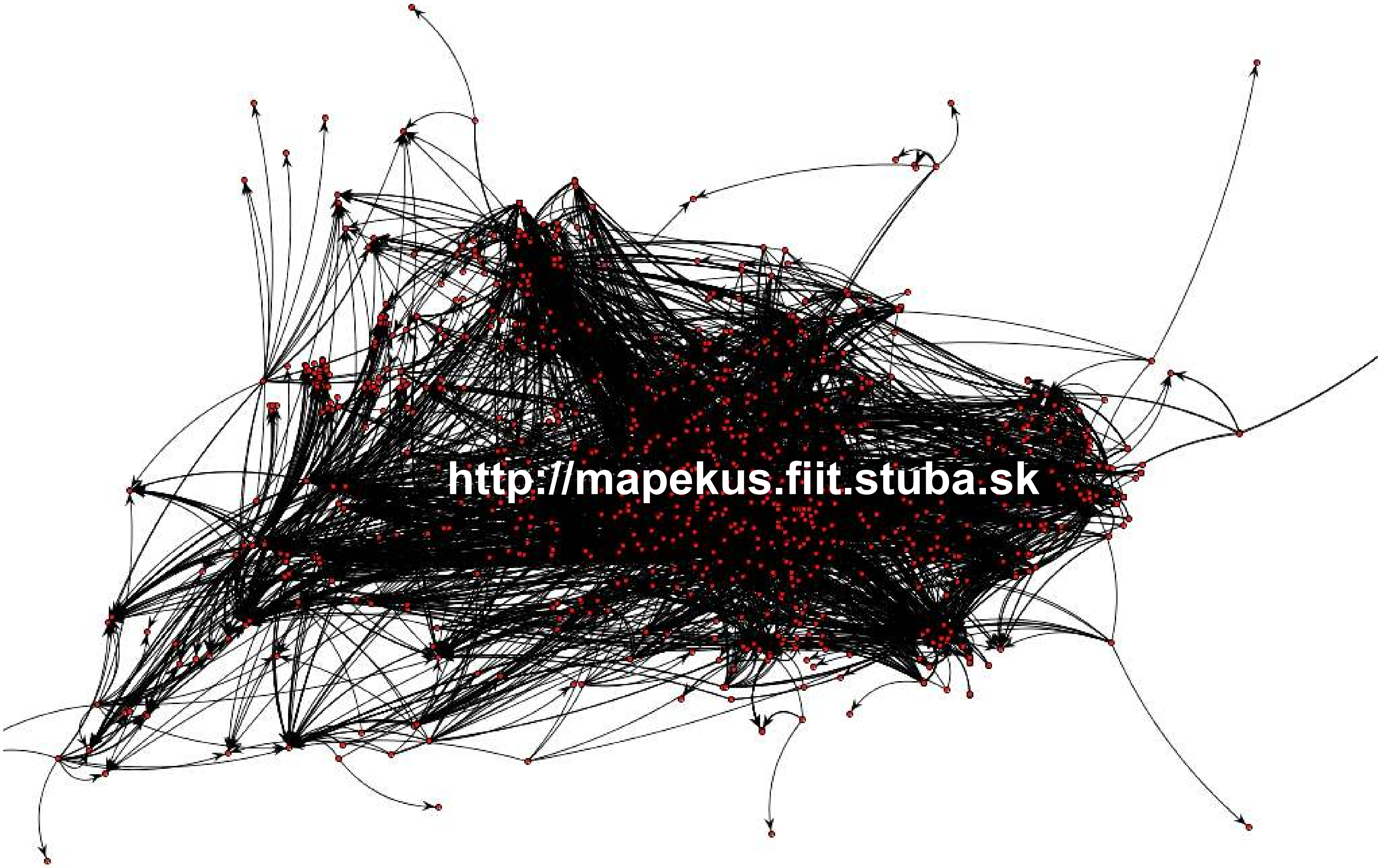
Ontology-based user model defines concepts representing user characteristics and identifies relationships between individual characteristics connected to domain ontology. Such a model is (after its population) used by presentation tools to provide personalized navigation and content. Model can be employed also in content organizing tools (e.g., perform sorting of items based on user's preferences).

User ontology in project MAPEKUS is composed of two standalone ontologies, which separate domain-dependent and general characteristics:

- Generic user ontology (*OWL*)
  Defines general user characteristics.

Future Work

- Make available integrated and cleaned ontology
  - add to this “pack” also cluster-based visual navigator of data
- Create smaller, focused data set in specialized sub-domains for experimental reasons:
  - software engineering
  - user modeling



http://mapekus.fiit.stuba.sk