Building Structured Web Community Portals:
A Top-Down, Compositional, and Incremental Approach

Pedro DeRose

University of Wisconsin-Madison

Joint work with Warren Shen, Fei Chen, AnHai Doan, and Raghu Ramakrishnan

Structured Web Community Portals

Numerous Web communities

- database researchers, movie fans, legal professionals, bioinformatics, enterprise intranets, tech support groups

Increasing interest in managing community data

Structured community portals capture information about community entities and relations

- allow users to query, browse, monitor, mine, etc.

Illustrating Examples

How should we build such portals?

Limitations of Current Solutions

Manual

- e.g., DBLP
- require a lot of human effort

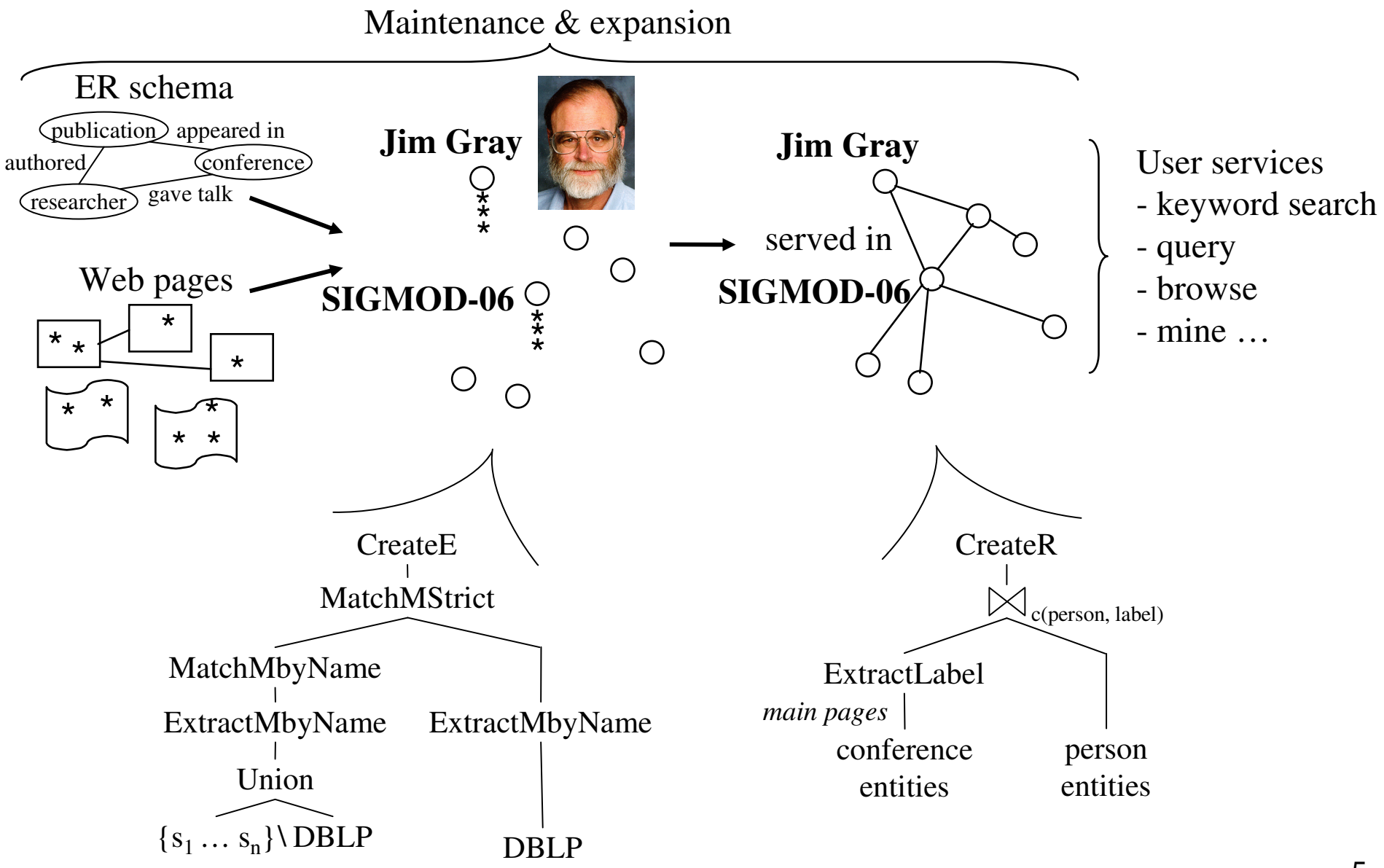
Semi-automatic, but domain-specific

- e.g., Yahoo! Finance, Citeseer
- difficult to adapt to new domains

Semi-automatic and general

- many solutions from the database, WWW, and Semantic Web communities, e.g., Rexa, Libra, Flink, Polyphonet, Cora, Deadliner
- often use monolithic solutions, e.g., learning methods such as CRFs
- require little human effort
- can be difficult to tailor to individual communities

Proposed Solution: A Compositional Approach



Benefits of Our Proposed Solution

Easier to develop, maintain, and extend

- e.g., using our workbench, 2 students × 1 week to create DBLife

Provides opportunities for optimization

- e.g., extraction and integration plans allow for plan rewriting

Can achieve high accuracy with relatively simple operators by exploiting community properties

- e.g., found talks with 88% F_1 by focusing on seminar pages

Rest of the Talk

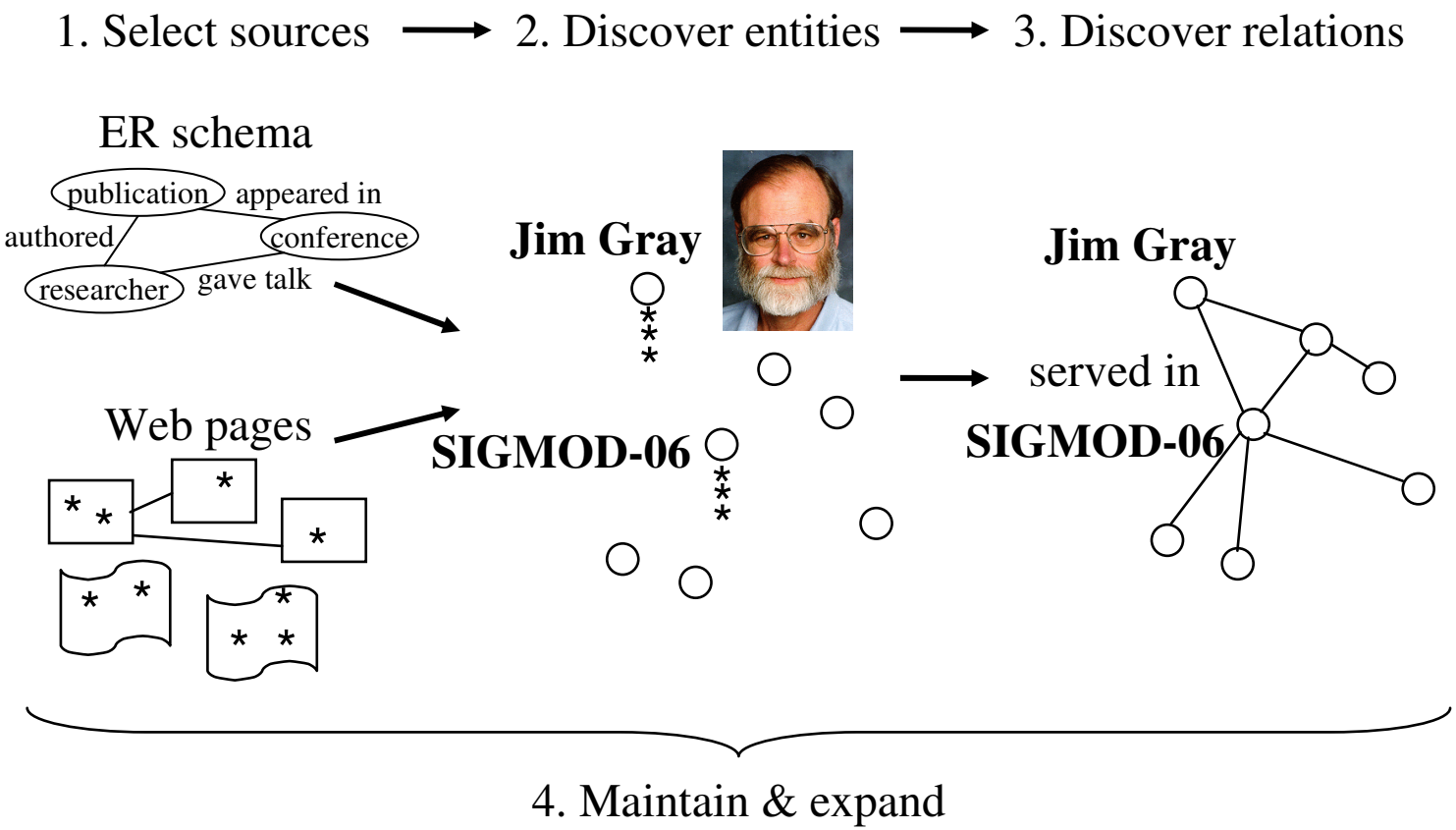
Our initial solution

- key ideas and contrast with current solutions

Cimple 1.0 workbench, DBLife prototype, and experimental evaluation

Future research directions

Workflow Overview



1. Select a Good Initial Set of Sources

Communitites often show an 80-20 phenomenon

- small set of sources already covers 80% of interesting activity

Select these 20% of sources

- e.g., for DB community, sites of prominent researchers, conferences, departments, etc.

Can incrementally expand later

- semi-automatically or mass collaboration

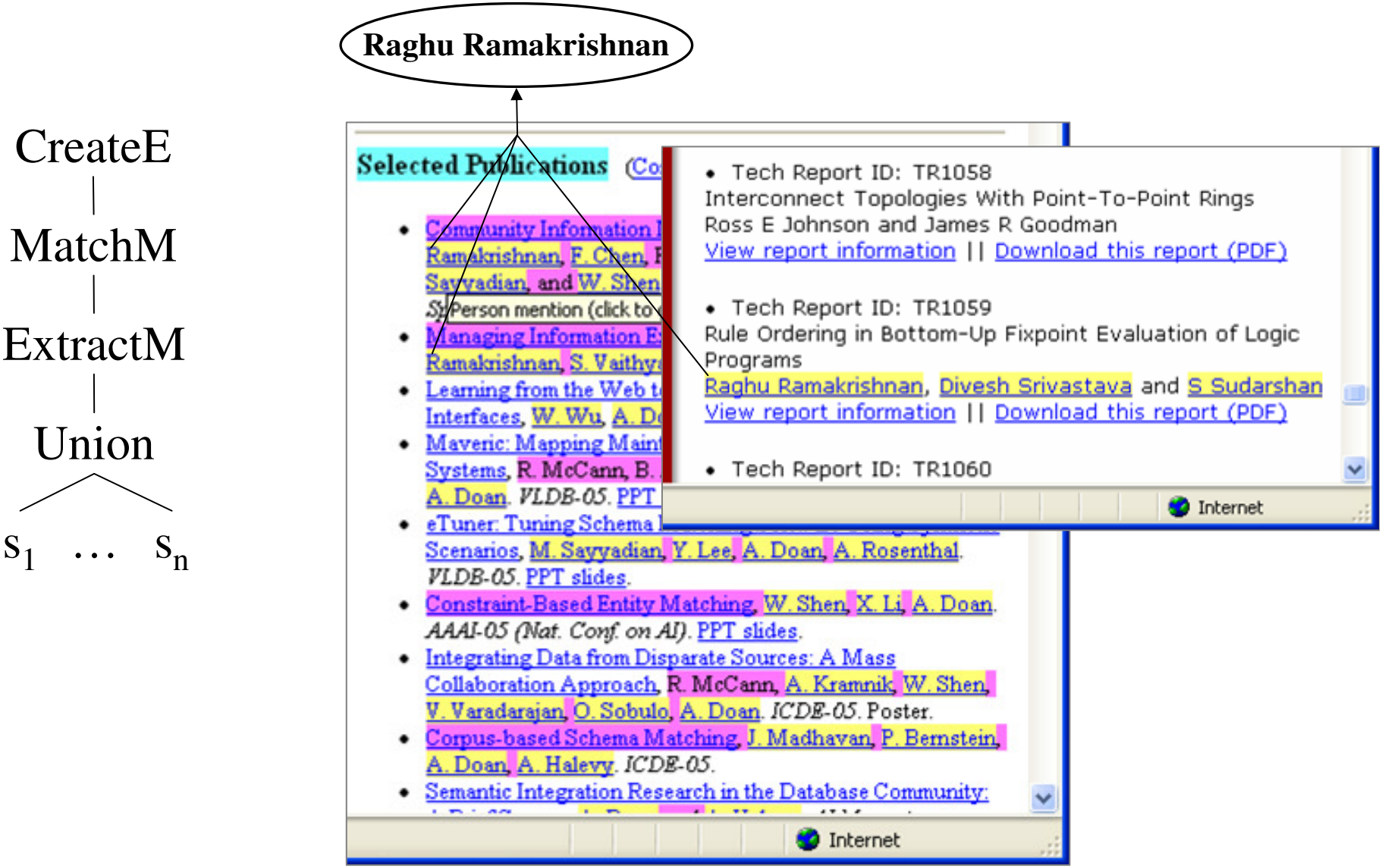
Differs from current solutions

- often select as many potentially relevant sources as possible
- lots of noisy sources, which can lower accuracy

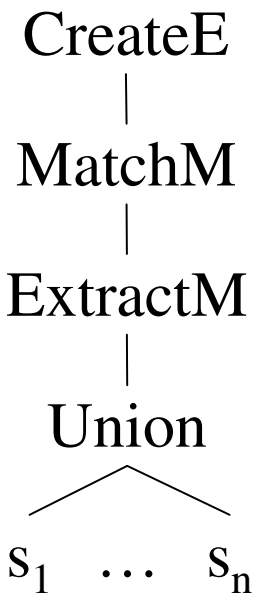
Crawl sources periodically

- e.g., DBLife crawls ~10,000 pages (+160 MB) daily

2. Create Plans that Discover Entities



Simple Solutions in Community Settings



These operators address well-known problems

- mention recognition, entity disambiguation…
- many sophisticated solutions

In community settings, simple solutions can already work surprisingly well

- often easy to collect entity names from community sources (e.g., DBLP)
  - ➡ ExtractMbyName: finds variations of names
- entity names within a community are often unique
  - ➡ MatchMbyName: matches mentions by name
- These simple methods work with 98% F_1 in DBLife

But there are difficult spots…

Handling Difficult Spots

DBLP: Chen Li

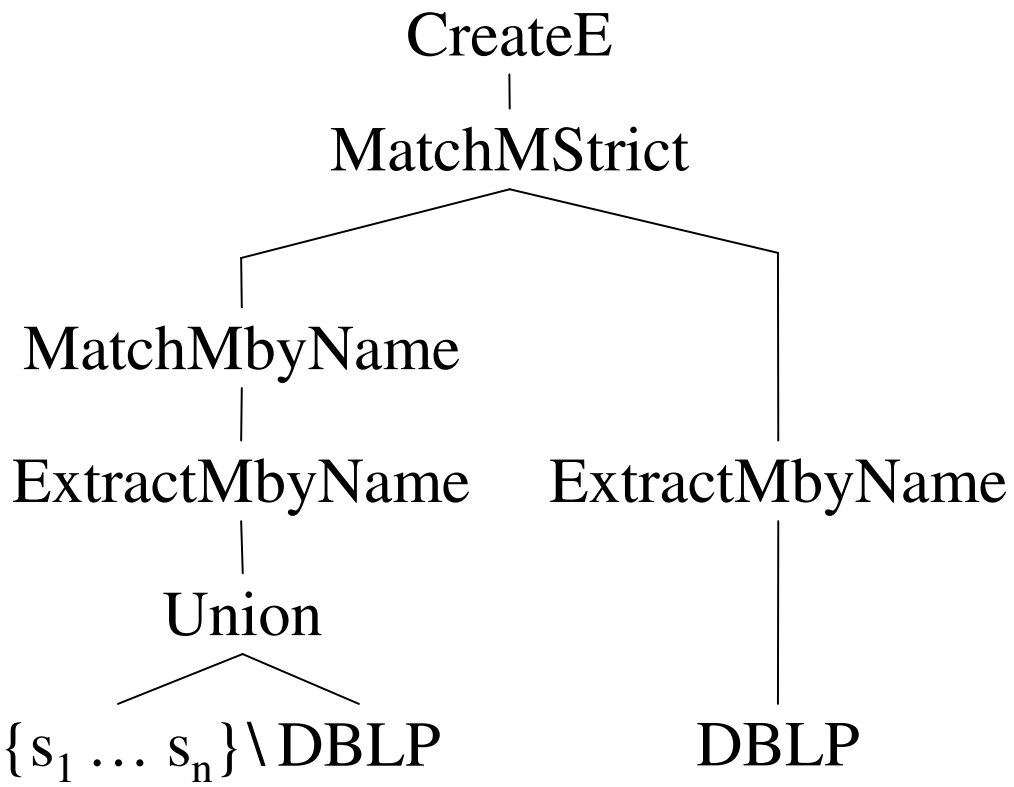
. . .

41. Chen Li, Bin Wang, Xiaochun Yang. VGRAM. VLDB 2007.

. . .

38. Ping-Qi Pan, Jian-Feng Hu, Chen Li. Feasible region contraction. Applied Mathematics and Computation.

. . .



Must decide which operators to apply where

- e.g., stricter operators to more ambiguous data

Provides opportunities for optimization

- See ICDE-07a for a way to optimize such plans

3. Create Plans that Discover Relations

We categorize relations into general classes

- co-occur, label, neighborhood…

Then provide operators for each class

- ComputeCoStrength, ExtractLabels, neighborhood selection…

And compose them into a plan for each relation type

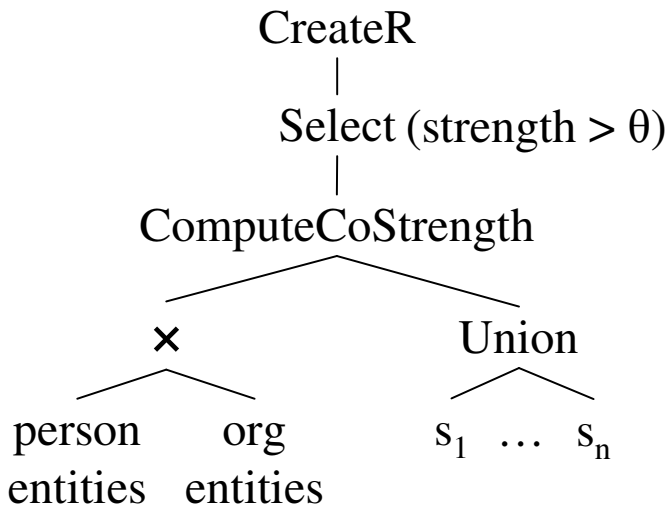
- makes plans easier to develop
- plans are relatively simple to understand
- can easily add new plans for new relation types

Illustrating Example: Co-occur

Find affiliated(person, org) relation

- e.g., affiliated(Raghu, Univ of WI), affiliated(Raghu, Yahoo! Research)
- categorize as a co-occur relation

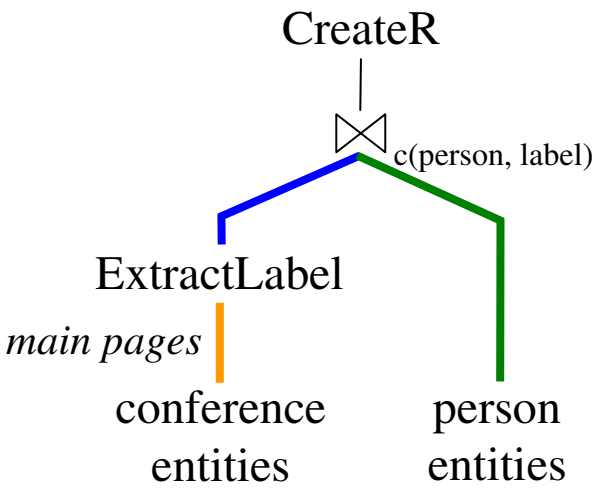
Compose a simple co-occur plan



This plan already finds affiliations with 80% F_1

Illustrating Example: Label

Plan for
served-in(person, conf)



ICDE'07 Istanbul Turkey

General Chair
- Ling Liu
- Adnan Yazici

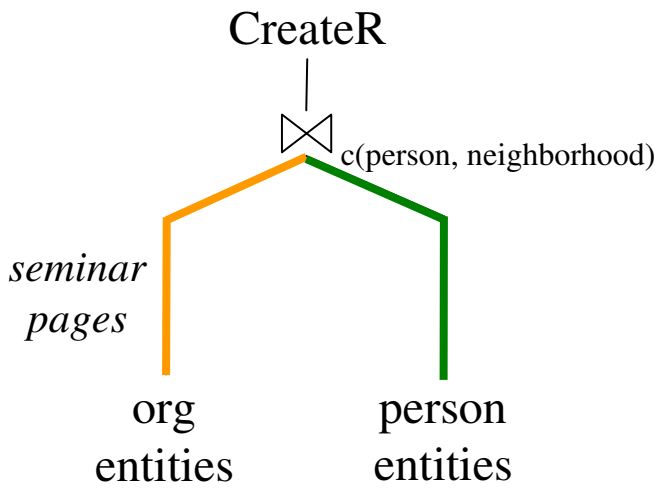
Program Committee Chairs
- Asuman Dogac
- Tamer Ozsu
- Timos Sellis

Program Committee Members
- Ashraf Aboulnaga
- Sibel Adali

…

Illustrating Example: Neighborhood

Plan for
gave-talk(person, venue)



UCLA Computer Science Seminars

Title: Clustering and Classification
Speaker: Yi Ma, UIUC
~~Contact: Rachelle Reamkitkarn~~

Title: Mobility-Assisted Routing
Speaker: ~~Konstantinos Psounis~~, USC
~~Contact: Rachelle Reamkitkarn~~

…

Discovering Relations: Discussion

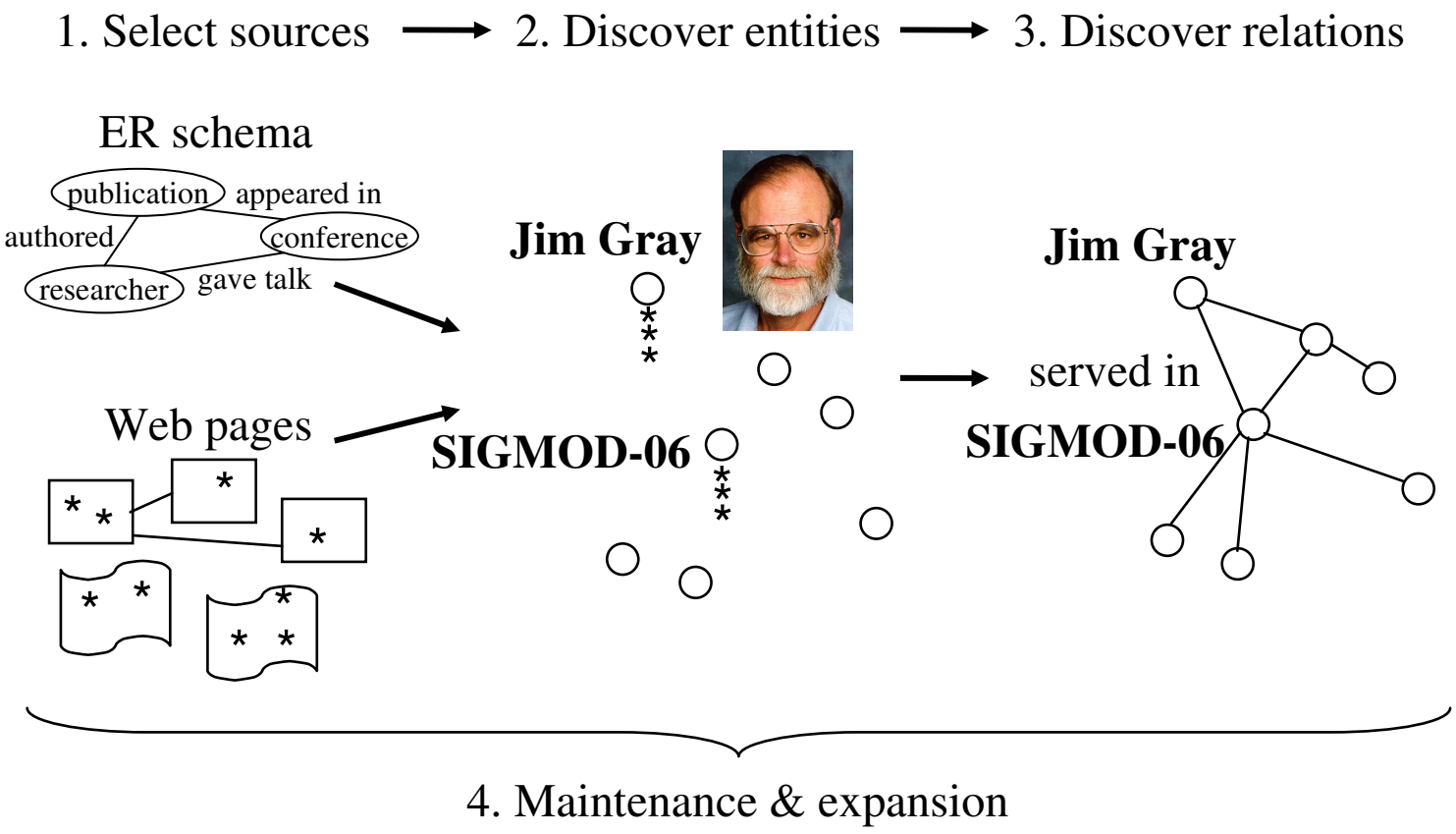
Creating top-down plans allows us to focus on highly relevant sources

- e.g., "gave talk" plan finds talks with 88% F_1

Composing operators into plans provides many opportunities for optimization

- like query plans, can be optimized via re-writing [VLDB-07a]

Generate a Daily ER Graph



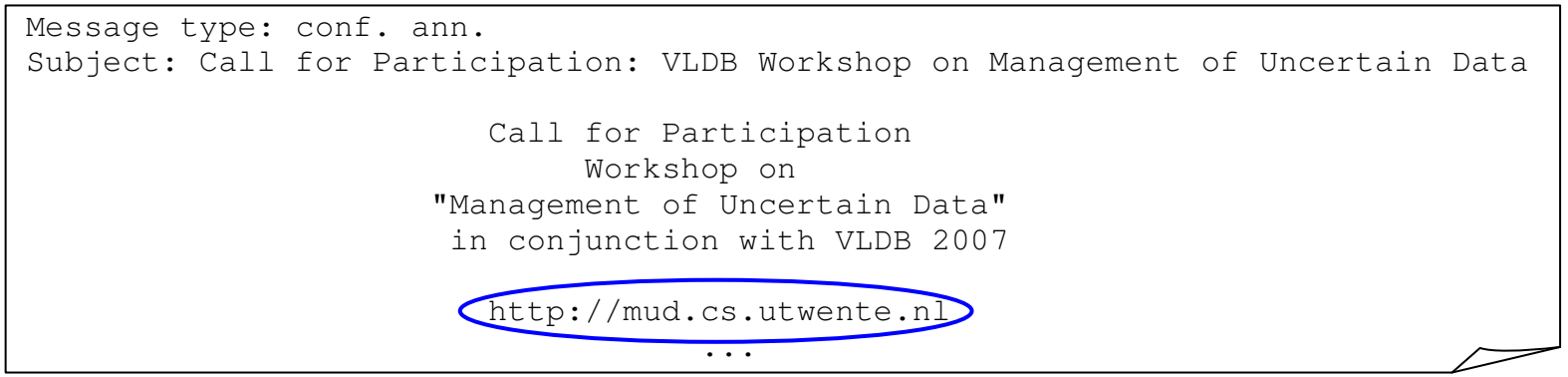
4. Maintain and Expand

Maintenance

- in many cases, core sources move or disappear only rarely
- can keep sources up-to-date with little manual effort

Incremental expansion

- we note that important new sources and entities are often mentioned in certain community sources (e.g., DBWorld)



```
Message type: conf. ann.
Subject: Call for Participation: VLDB Workshop on Management of Uncertain Data

                    Call for Participation
                         Workshop on
                "Management of Uncertain Data"
                 in conjunction with VLDB 2007

                    http://mud.cs.utwente.nl
                              ...
```

- monitor these sources with simple extraction plans

A Compositional Portal-Building Workbench

Cimple 1.0 workbench

- empty portal shell, including basic services and admin tools
  - browsing, keyword search…
- set of general operators, and means to compose them
  - MatchM, ExtractM…
- simple implementation of operators
  - MatchMbyName, ExtractMbyName…
- end-to-end development methodology
  - 1. select sources, 2. discover entities…

Employ Cimple 1.0 to Build DBLife

Initial DBLife (May 31, 2005)	Time
Data Sources (846): researcher homepages (365), department/organization homepages (94), conference homepages (30), faculty hubs (63), group pages (48), project pages (187), colloquia pages (50), event pages (8), DBWorld (1), DBLP (1)	2 days, 2 persons
Core Entities (489): researchers (365), department/organizations (94), conferences (30)	2 days, 2 persons
Operators: DBLife-specific implementation of MatchMStrict	1 day, 1 person
Relation Plans (8): authored, co-author, affiliated with, gave talk, gave tutorial, in panel, served in, related topic	2 days, 2 persons

Maintenance and Expansion	Time
Data Source Maintenance: adding new sources, updating relocated pages, updating source metadata	1 hour/month, 1 person

Current DBLife (Mar 21, 2007)
Data Sources (1,075): researcher homepages (463), department/organization homepages (103), conference homepages (54), faculty hubs (99), group pages (56), project pages (203), colloquia pages (85), event pages (11), DBWorld (1), DBLP (1)
Mentions (324,188): researchers (125,013), departments/organizations (30,742), conferences (723), publication: (55,242), topics (112,468)
Entities (16,674): researchers (5,767), departments/organizations (162), conferences (232), publications (9,837), topics (676)
Relation Instances (63,923): authored (18,776), co-author (24,709), affiliated with (1,359), served in (5,922), gave talk (1,178), gave tutorial (119), in panel (135), related topic (11,725)

DBLife Accuracy

Mean accuracy over 20 randomly chosen researchers

Experiment	Mean Recall	Mean Precision	Mean F_1
Extracting mentions with ExtractMByName	0.99	0.98	0.98
Discovering entities with default plan	1.00	0.96	0.98
Discovering entities with source-aware plan	0.97	0.99	0.98
Finding "authored" relations (DBLP plan)	0.76	0.98	0.84
Finding "affiliated" relations (co-occurrence)	0.85	0.83	0.80
Finding "served in" relations (labels)	0.84	0.81	0.77
Finding "gave talk" relations (neighborhood)	0.87	1.00	0.88
Finding "gave tutorial" relations (labels)	0.90	1.00	0.92
Finding "on panel" relations (labels)	0.95	0.92	0.89

Relatively Easy to Deploy, Extend, and Debug

DBLife has been deployed and extended by a dozen individual developers

- CS at IL, CS at WI, Biochemistry at WI, Yahoo! Research
- development started after only a few hours Q&A

Developers quickly grasped our compositional approach

- easily zoomed in on target components
- could quickly tune, debug, or replace individual components
- e.g., a new student extended ComputeCoStrength operator and added the "affiliated" plan in just a couple days

Lessons Learned

Top-down, compositional, incremental is promising

- relatively easy to develop, maintain, and extend
- provides opportunities for optimization
- relatively simple operators can achieve high accuracy

User feedback may help tremendously

- use mass collaboration to correct and update data
- our current work includes turning DBLife into a wiki

Research Challenges

The overall approach

- right data model? viewpoint? operators? composition?
- declarative solutions? [VLDB-07a]
- right data storage? should we use RDBMS? [VLDB-07b]
- dealing with evolving data? provenance? uncertainty?

Optimization

- run time? accuracy? [ICDE-07a, ICDE-07b, Tech Report 07a]
- distributed computation?

Semantics

- knowledge management? Semantic Web technologies?

User community

- effective user services? context-sensitive services?
- can users contribute data? code? domain knowledge? mashups? and how? [Tech Report 07b]
- can we capture and exploit social interaction?

Conclusions

Building structured Web community portals

– increasingly crucial problem

Proposed a top-down, compositional, and incremental solution

– as embodied by the Cimple 1.0 workbench

Developed the DBLife portal prototype

– shows promising results

– a research/education tool, community service, benchmark

Identified many interesting research challenges

– requires a community effort

– let me know if you would like the DBLife code or data

For more information, query "cimple wisconsin"