

Denilson Barbosa  
denilson@ucalgary.ca

# **TOWARDS SCHEMA-FREE EXCHANGE AND UPDATE OF XML DATA**

# Why XML?

- Web data means XML and...
  - Web-based **data sharing systems** (e.g., Google base)
- ... XML is everywhere else
  - Microsoft's Office Open XML standard
  - OASIS OpenDocument standard
- DBMS support
- Web Services
- Semantic Web

# Data Management for the Masses

- Several popular online data stores
  - Allow the users to organize their data in any way they wish, while offering pre-defined, customizable schemas from different application domains
  - Provide flexible, easier query paradigms than XQuery mixing querying and searching (**schema-free XQuery** [Li et al.'04])
  - GoogleBase, FreeBase, CraigsLists ...

Domains and Types : Freebase

http://www.freebase.com/view/allDomains

Daily Work Teaching Google Yahoo! SIGMOD08 SIGMOD-CMT Blackboard français denilson.barbosa

freebase<sup>TM</sup>  
alpha

Home Data Apps Discuss Help  
Please sign in or register to contribute

Search

## Domains & Types

**USER'S TYPES**

Other users' types  
 Types you've created

**POPULAR TYPES**

Person 615,551  
City/Town 117,243  
Company 35,091  
Musical Artist 348,096  
Film 33,389  
TV Program 10,386  
Book 18,513  
Computer Game 12,774  
Restaurant 100,141  
Skyscraper 1,361  
Cause Of Death 184  
Aircraft 3,228

**DATA IMPORTS**

Perform a bulk upload  
Have a data set you'd like to load in bulk? Here's how to

Topics in Freebase are organized by type. Types are grouped together into domains. Please explore and contribute! If you are an expert with some technical background and you want to become an administrator, let us know in the [discussion forums](#).

stay up to date  
**JOIN THE DATA MODELING EMAIL LIST**

[browse the list archive](#)

**NEWEST DOMAINS**

Wine  
Martial Arts  
Anime/Manga  
Automotive  
Religion  
Cricket  
Games  
Law  
Time  
Fictional Universes

**RECENT DATA LOADS**

NFL Rosters for 2007/2008  
Company founding data

**Categories**

Arts & Entertainment ▶ Science & Technology ▶  
Society ▶ Time & Space ▶  
Sports ▶ Special Interests ▶  
Products & Services ▶ System ▶  
Money ▶

**Data Mob Projects**

New thes...

Done

Star : Freebase

http://www.freebase.com/view/astronomy/star


Daily Work Teaching Google Yahoo! SIGMOD08 SIGMOD-CMT Blackboard français denilson.barbosa


VIEW SCHEMA


type as such, but they are also typed as what we now know them to be.


NARROW RESULTS


Name  
T


Hipparcos ID  



Constellation  
 alpha


Star  
 Andromeda  
Constellation


Star  
 Centaurus  
Constellation


Star  
 Crux  
Constellation


Star  
 Monoceros  
Constellation

Star  
 Perseus  
Constellation

Star  
 Libra  
Constellation

Star  
 Delphinus  
Constellation


Star  
 Leo  
Constellation

Luminosity  


Filter

Items 1 - 30 of 30+.

**Beta Ursae Majoris**  
*Star, Celestial Object*  
Beta Ursae Majoris ( $\beta$  UMa /  $\beta$  Ursae Majoris) is a star in the constellation of Ursa Major. It also has the traditional name Merak. It is more familiar to northern hemisphere observers as one of the "pointer stars" in the Big Dipper, and a line connecting it with...

**Centaurus**  
*Constellation*  
 Centaurus (, ) is a bright constellation of the southern hemisphere. One of the largest constellations in the sky, Centaurus was one of the 48 constellation listed by Ptolemy, and also counts among the 88 modern constellations. Centaurus contains Proxima Centauri, a red dwarf that is the nearest known... star in the constellation Centaurus. It also has the traditional name Rigel Kentaurus. It has a faint, 14th magnitude companion, Epsilon Sagittarii B, 32 arcseconds distant.

**J Centauri**  
*Star, Celestial Object*  
J Centauri (J Cen) is a star in the constellation Centaurus. It is approximately 355 light years from Earth. J Centauri is a blue-white B-type main sequence dwarf with a mean apparent magnitude of +4.52. It is classified as a Beta Cephei type variable star and its

Done

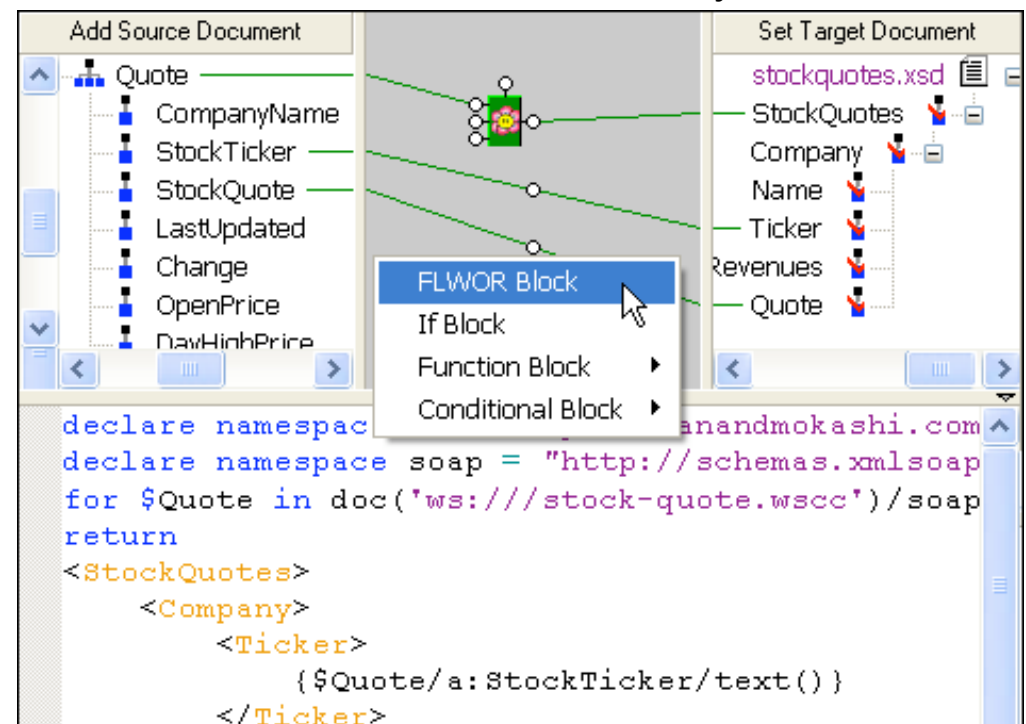
# Data Exchange

- Data exchange consists in taking data structured under a **source** schema and **creating an instance** of a **target** schema that reflects the source data as accurately as possible [Fagin et al.'03]
- Data exchange appears naturally in information integration and Web data management settings

# Data Exchange Mappings

- Mappings are queries
- To define good mappings, one must understand:
  - The source **and** target schemas
  - Which data is where
- These are done with the help of tools

Source: stylusstudio.com



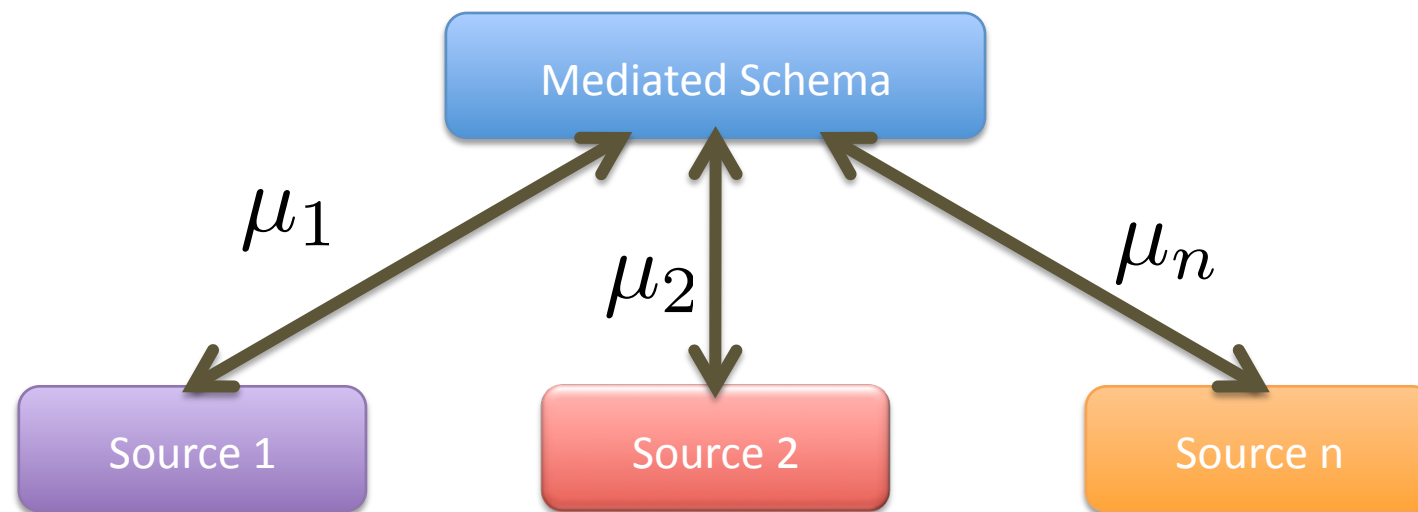
# State-of-the-Art in Data Exchange

- Heavy artillery: DBMS on each end of the mapping, coupled with sophisticated mapping design tools
  - Prototypical example: IBM's CLIO
- **Costly operation:** migrating data between two similar corporate applications **may take several man months** [Sikka VLDB 2006 Keynote]
  - Typical organizations have hundreds of database applications



# Classical Data Integration/Exchange

- Mappings are expressed in a variety of formalisms: SQL, XQuery, LAV, GAV, and GLAV
- Classical notion of **mediated schema**

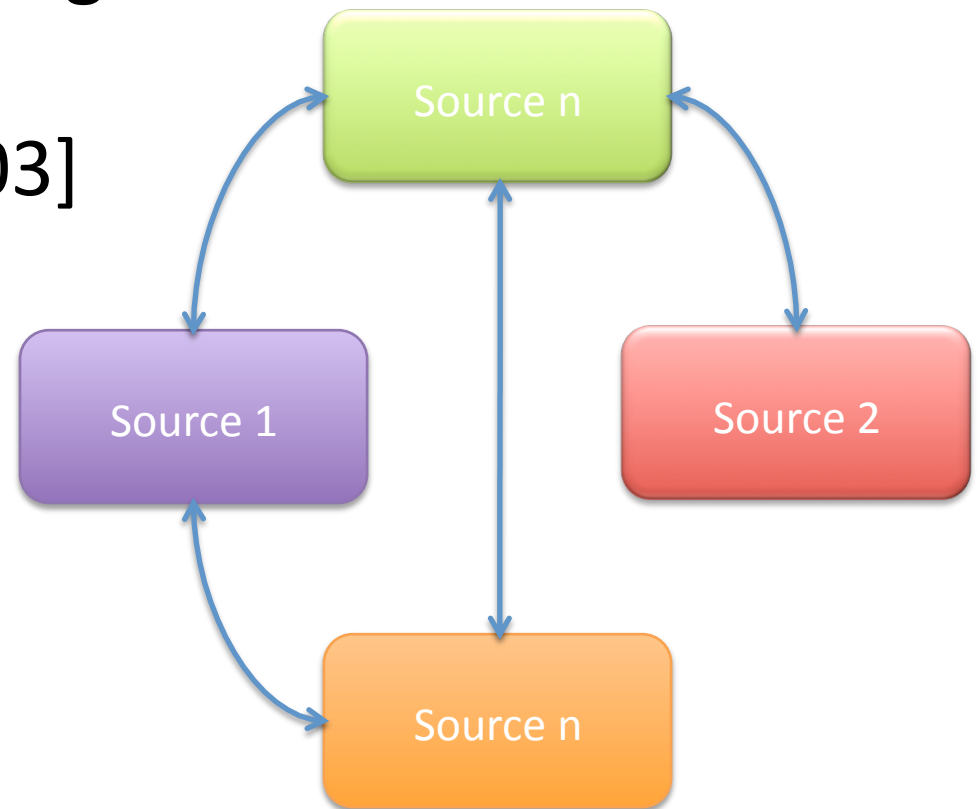


# Classical Data Exchange

- Lack in user support [Doan & Halevy'05]
  - Poor explanations to users about how the mapping works; especially needed when the mapping **doesn't work** as expected
- Lack in formal understanding [Doan & Halevy'05]
  - Most works have focused on tools and prototypes, little attention has been devoted to understanding what these mappings are and what they do

# P2P Data Exchange

- P2P is a promising paradigm for distributed data management [Aberer'03]
- Data sources exchange data amongst them independently of each other
  - No centralized mediated schema



# Data Exchange For the Masses?

- **Most P2P connections are brief, most exchanges are “one-of” affairs**
- Users are not trained in data management (they don't even see the “database” for the most part)
- **Lower cost for data management results in a myriad of schemas** for the same application domain [Halevy et al.'07]
- Needed: better mapping tools, easy to use data manipulation paradigm

# Outline

- In this talk I will briefly discuss ...
- A simple and flexible data exchange framework for XML (FleDEx) [WIDM'07]
- Ongoing work on allowing casual users to update XML documents

# **FLEXIBLE XML DATA EXCHANGE**

# Example: Exchanging Music Data



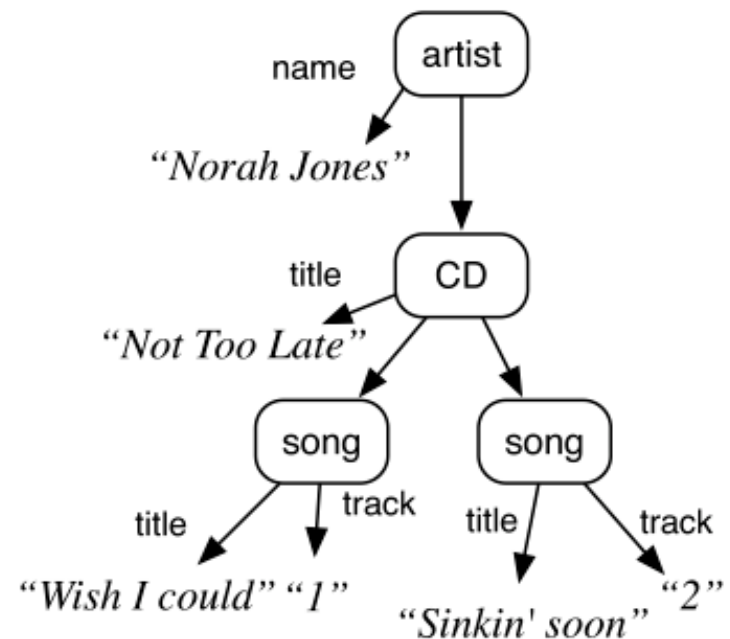
Music grouped by genres



Music grouped by artists/bands

# Data Model

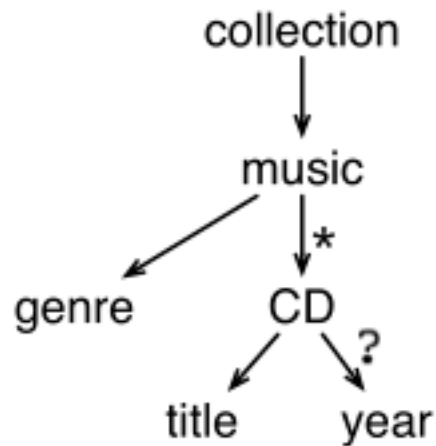
- We capture simple, unordered XML
- Each element label becomes a type (à la DTDs)
- Two kinds of nodes:
  - Entities (or XML inner element nodes)
  - Attributes (XML leaf nodes—elements/attributes)



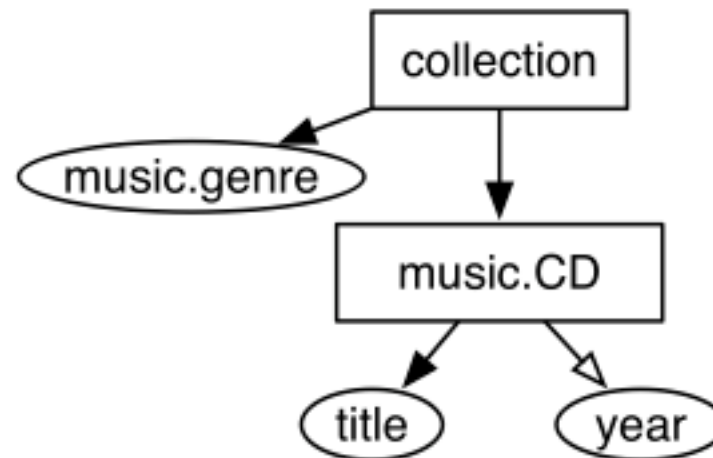


# Abstraction of DTDs

- The schema specifies the nesting of entities and the attributes that describe them



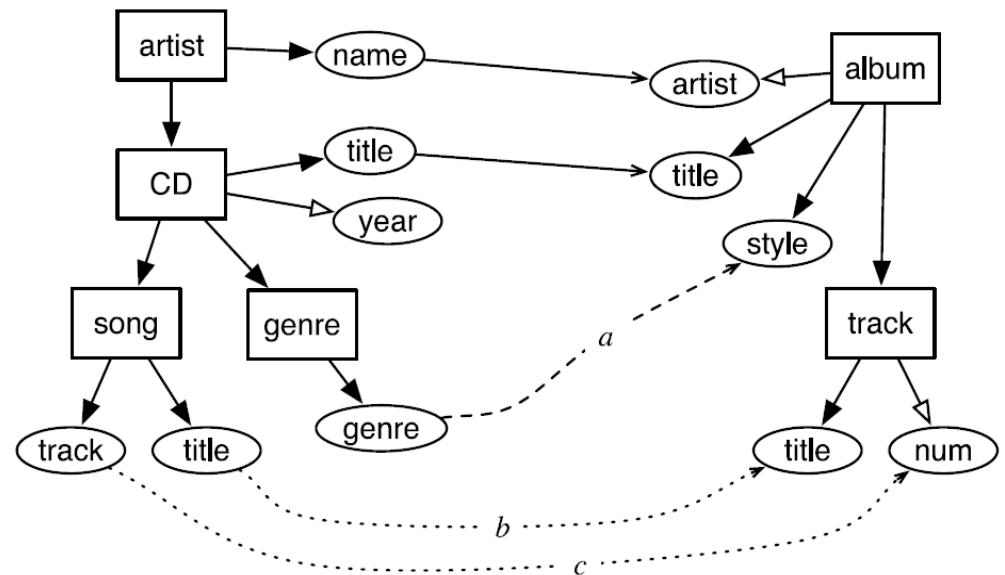
(a) DTD Graph.



(b) *FDM* Schema.

# FleDEx – Flexible Data Exchange

- **STEP1:** Automatic discovery of correspondences between source and target data



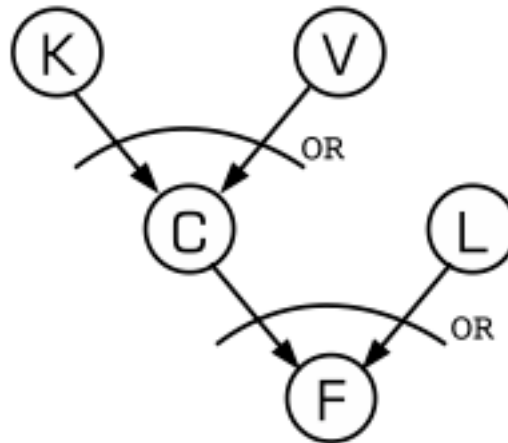
- We consider **all possible pairwise mappings**, and keep those that lead to the better overall similarity scoring
- Combined state-of-the-art matching algorithms

# Matching Types A, B

- L: Label similarity between A,B
- K: Keyword similarity between A,B
  - Takes tf-idf into account
- C: Attribute overlap between A,B
- C: For numeric values, we look into the overlap between their distributions

# Combining Matching Scores

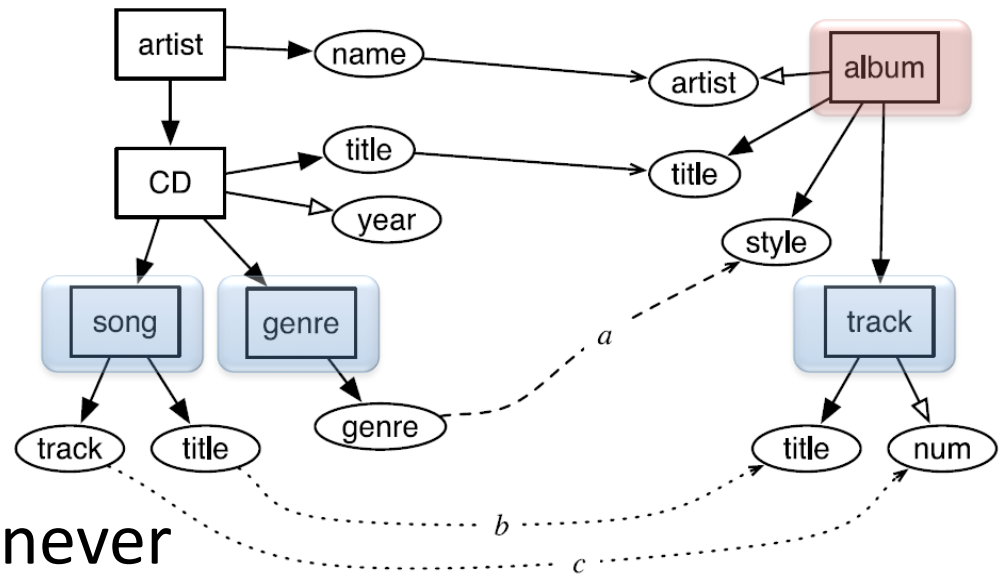
- We use noisy-OR gates [Pearl'88]:



- The final score of a node is high if any (or both) incoming is high

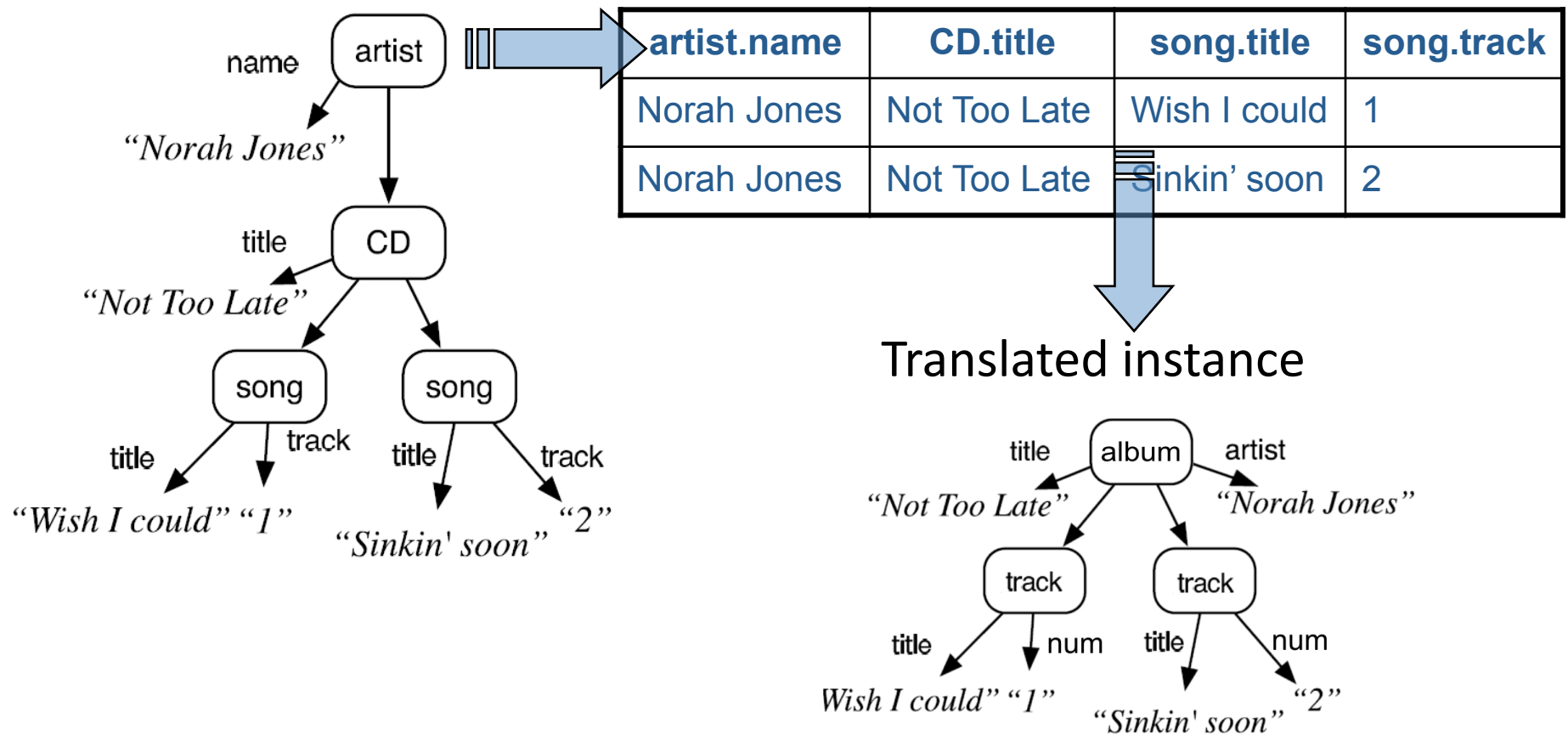
# FleDEx – Flexible Data Exchange

- **STEP2:** consider all candidate mappings
- Choose a suitable one
  - Avoid redundancy whenever possible; **never introduce** redundancy
  - Ex: artist and title must be duplicated for each style to compose a album and all tracks must be repeated for each duplicate album!



# FleDEx – Flexible Data Exchange

- **STEP3:** translate the source data



# FleDEx: Results

- Goal: produce “good” mappings on real data
- Evaluation metric: f-measure (accuracy)
- Real data:

Domain	Source Collection		Target Collection		Overlap
	Entities	Attr.	Entities	Attr.	
Movies	774	77	8,914	19	10
Music	714	40	10,000	4	4
Books	789	5	1,211	19	4
Articles	1,630	6	8,000	13	4

- Details in [WIDM'07]

```

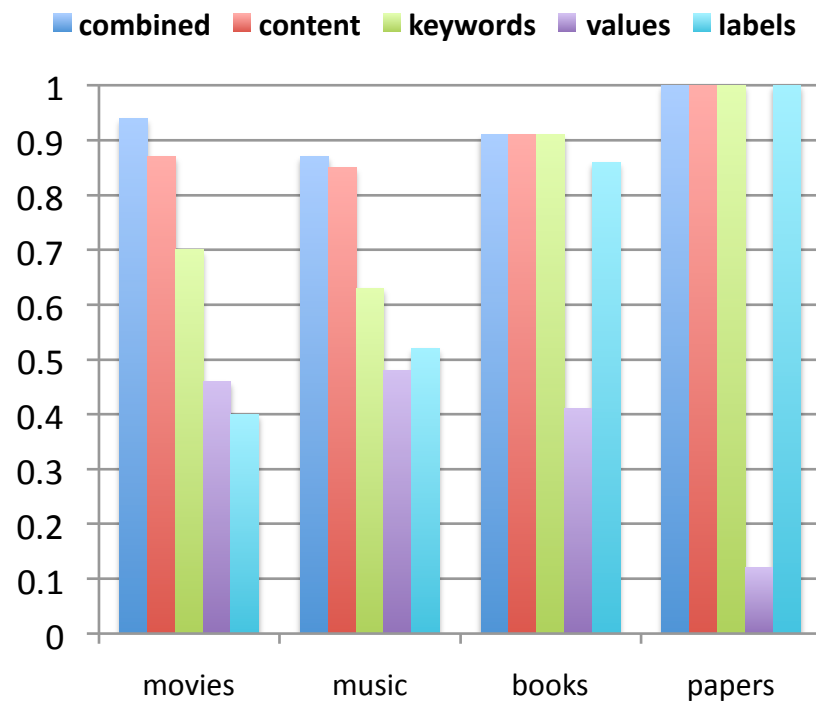
<movie>
  <title>Night of the Living Bread</title>
  <imdb_key>0133313</imdb_key>
  <year>1990</year>
  <rating votes="22">7.1</rating>
  <genre>Short</genre>
  <genre>Comedy</genre>
  <genre>Horror</genre>
  <keyword>parody</keyword>
  <keyword>independent-film</keyword>
  <credits>
    <director>Kevin S. O'Brien</director>
    <cast>
      <cast_member>Vince Ware</cast_member>
      <cast_member>Katie Harris</cast_member>
    </cast>
  </credits>
  <runtimes>
    <runtime country="USA">8</runtime>
  </runtimes>
  <country>United States of America</country>
  <languages>
    <language>English</language>
  </languages>
  <certifications>
    <certification country="USA">Unrated</certification>
  </certifications>
  <user_comment>
    <p>Being someone who lists Night of the Living Dead at number

```

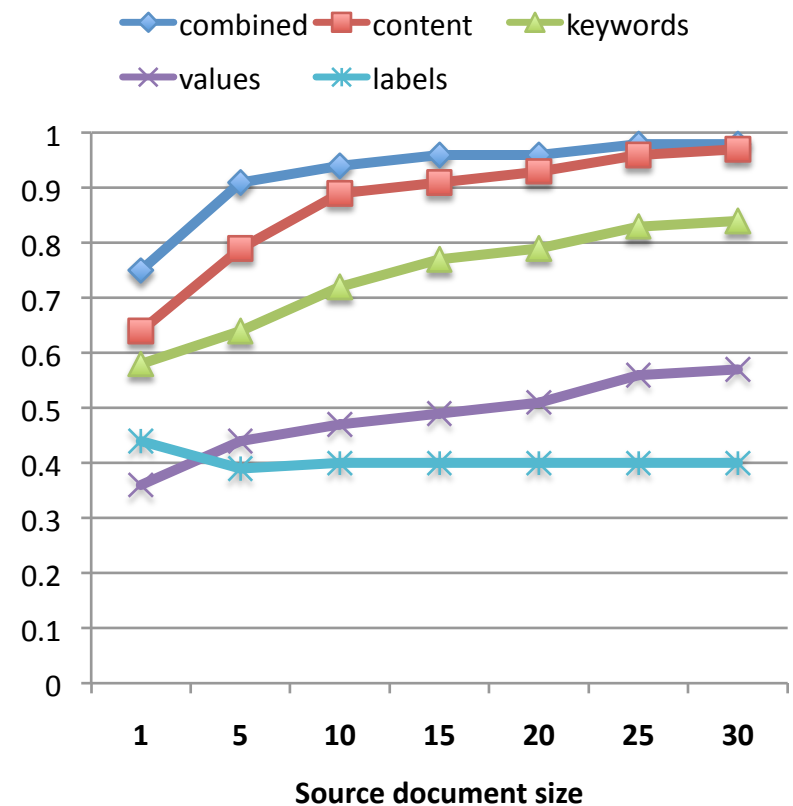


```
<movie>
  <title>
    Africa's Elephant Kingdom
  </title>
  <mid>
    1800020075
  </mid>
  <url>
    http://movies.yahoo.com/movie/1800020075/details
  </url>
  <details>
    <description>
      An elephant family is tracked as they make their way
    </description>
    <production_status>
      Released
    </production_status>
    <genres>
      <genre>
        Documentary
      </genre>
    </genres>
    <running_time>
      40 min.
    </running_time>
    <release_date>1998</release_date>
    <MPAA_rating>
      Not Rated
    </MPAA_rating>
    <distributors>
      <distributor>
        IMAX Corporation
      </distributor>
    </distributors>
```

# Results: Combining Scores

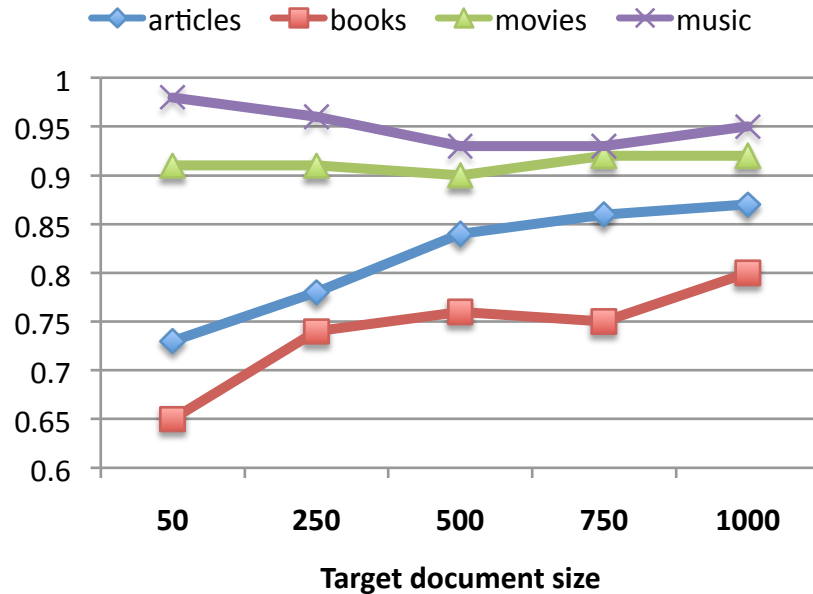


Size=50

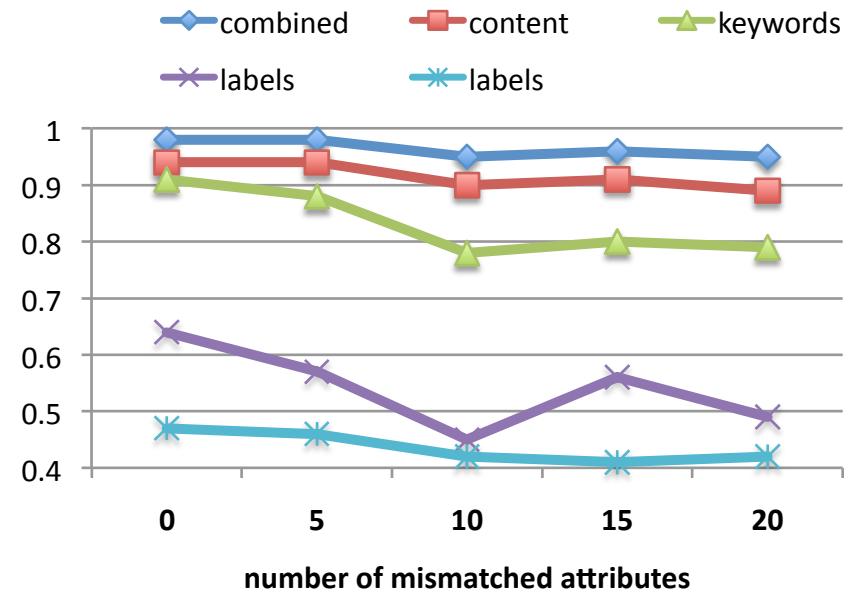


Movies domain

# More Results



Impact of target document size



Resilience to noise

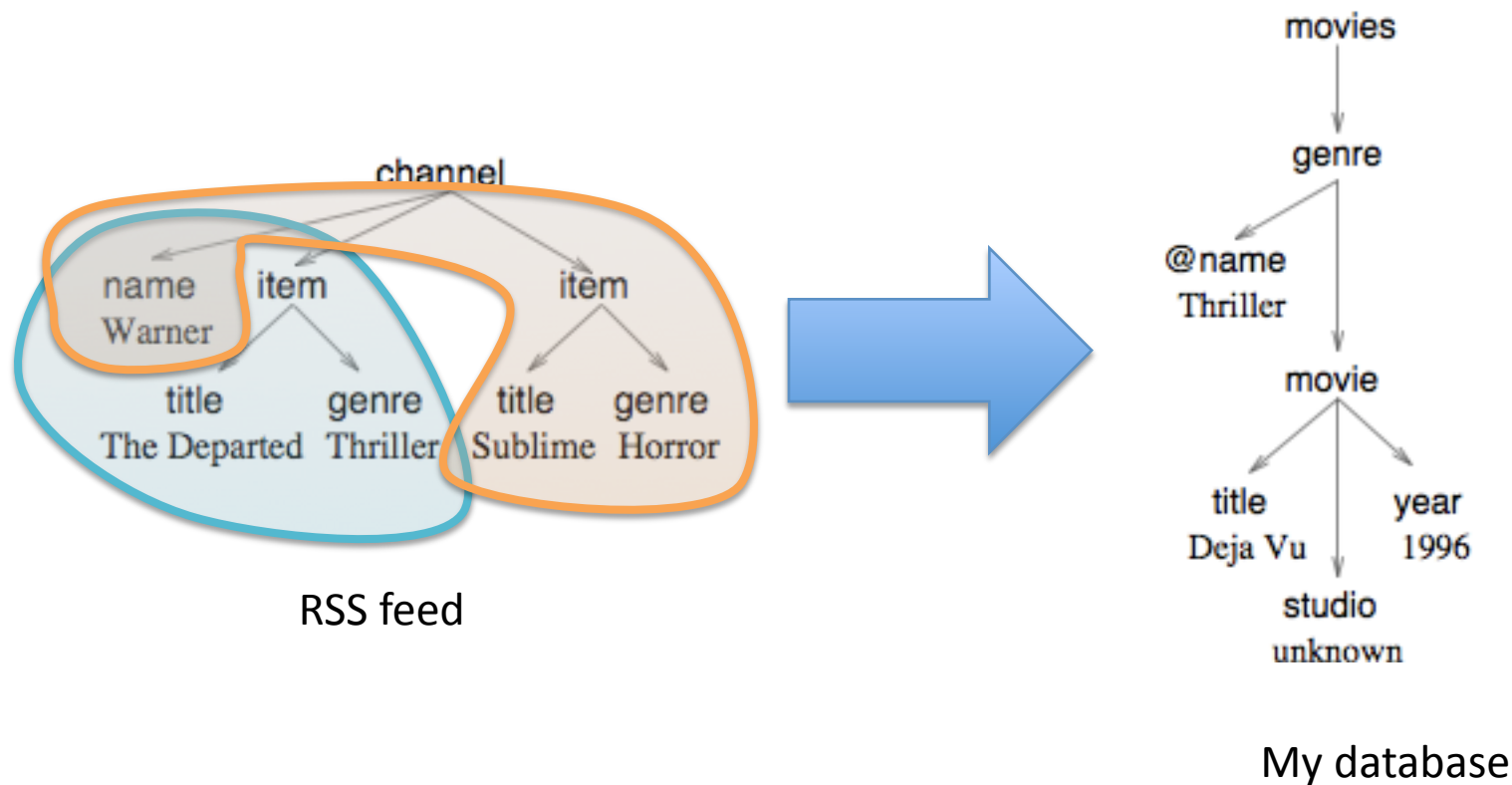
# FleDEx: Summary of Results

- High accuracy in finding mappings even with pretty small databases (as few as 50 objects)
- High accuracy even with low overlap between the source and target data
- High accuracy with noise (i.e., real data that does exist in the source or the target only, but not in both)

# SCHEMA-FREE XML UPDATES

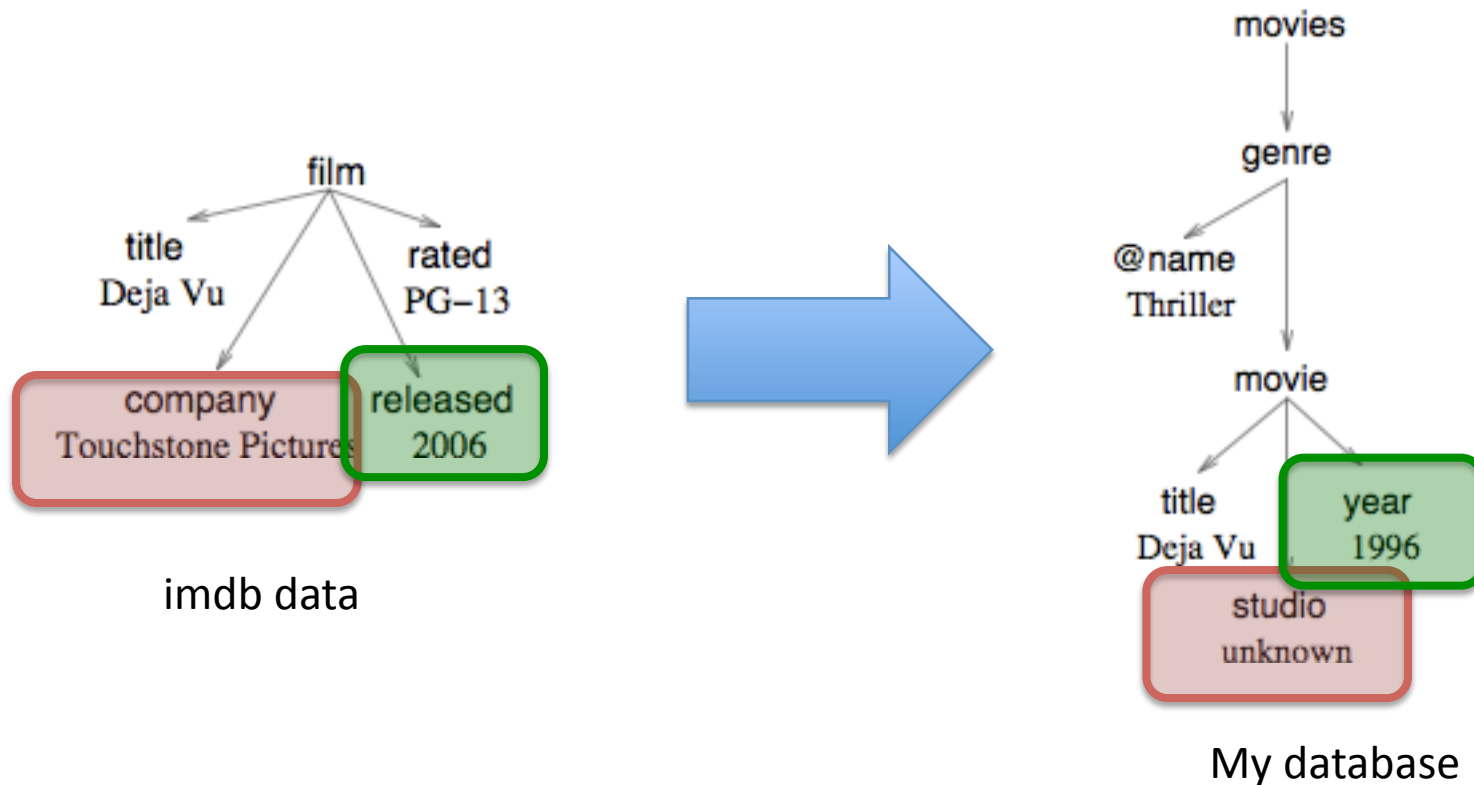
# Updates for the Masses?

- Consider this simple operation: copying movies from one database into another

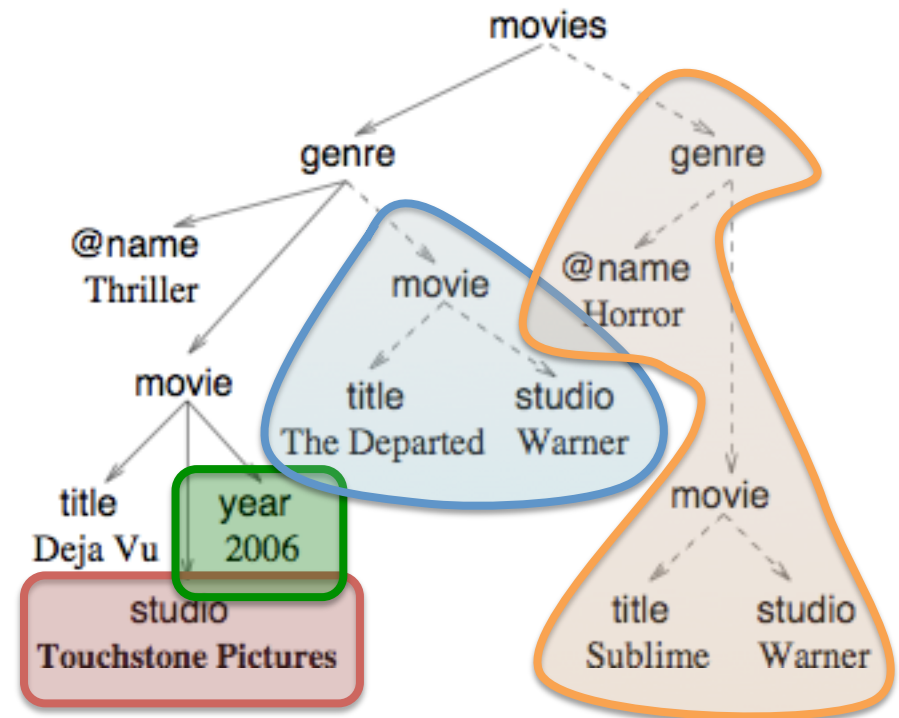
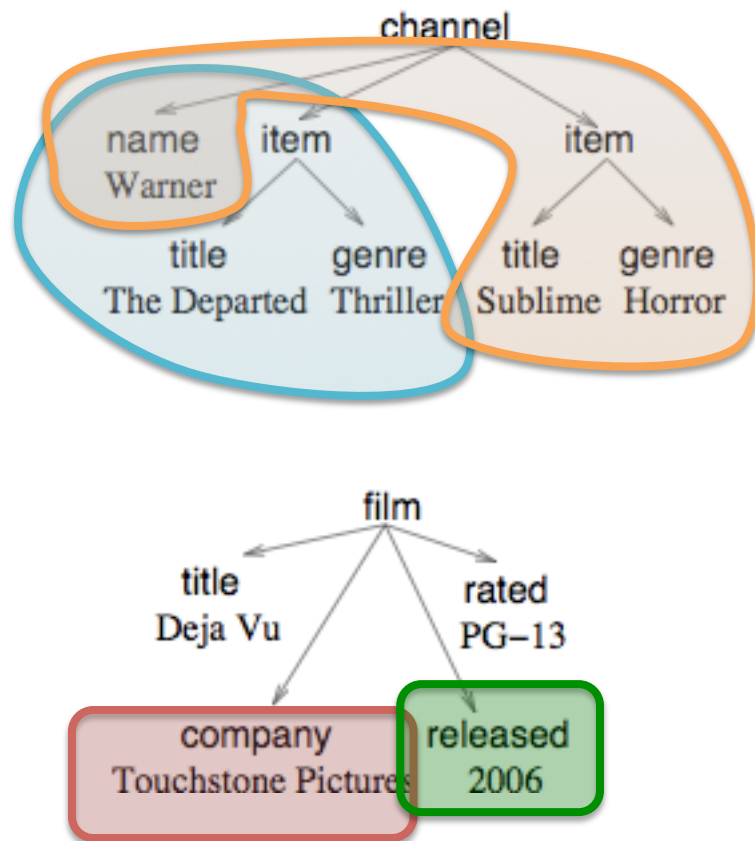


# Updates for the Masses?

- Consider another operation: updating a database with more accurate information



# Updates for the Masses?





# Updates for the Masses?

Inserting  
Movie 1



**SA1: DO INSERT**

```
let $item := doc('input.xml')//item[1]  
return <movie>
```

```
    <title>string($item/title)</title>
```

```
    <studio>string($item/../name)</studio>
```

```
  </movie>
```

```
INTO doc('db.xml')//genre[@name="Thriller"]
```

**SA2: DO INSERT**

```
let $item := doc('input.xml ')//item[2]
```

```
return <genre name="string($item/genre)">
```

```
  <movie>
```

```
    <title>string($item/title)</title>
```

```
    <studio>string($item/../name)</studio>
```

```
  </movie>
```

```
</genre>
```

```
INTO doc('db.xml')//movies
```

Inserting  
Movie 2



# XQuery for Casual Users

- Ideas:
  - Provide IR-style search that takes the structure into account
  - Relax the notion of positional/structural predicate into a full-fledged XML query language
- Methods:
  - Approximate XML Joins [Guha et al.'02]
  - Schema-Free XQuery [Li et al.'04]
  - Approximate XML matching [Amer-Yahia et al.'03]

# Structured Updates

- A **structured update** is a triple  $su=(op, loc, c)$ 
  - $op$ : INSERT-BEFORE, APPEND, REPLACE, DELETE
  - $loc$ : XPath expression over the target database
  - $c$ : new content that goes in the database
- An update program consists of several individual structured updates

# Structured Updates

Inserting  
Movie 1



```
 $u_1 = (\text{APP},$   
      doc('db.xml')//genre[@name = "Thriller"],  
      <movie><title>The Departed</title>  
      <studio>Warner</studio></movie>)
```

Inserting  
Movie 2



```
 $u_2 = (\text{APP},$   
      doc('db.xml')//movies,  
      <genre @name = "Horror">  
      <movie><title>The Departed</title>  
      <studio>Warner</studio></movie>  
      </genre>)
```

# Schema-Free Update Framework

- Mechanism for the user to specify the source and target nodes involved in the operation
  - Point-and-click
  - Copy-and-paste
  - A simpler language
- Update semantics
  - Conservative: avoiding redundancy
- Producing the equivalent structured updates

# Update Language – Examples

- Inserting all the data in the RSS document:

```
INSERT doc('RSS.xml') INTO doc('db.xml')
```

- Inserting a specific movie

```
INSERT doc('RSS.xml')//item[title='The Departed']  
INTO doc('db.xml')genre genre
```

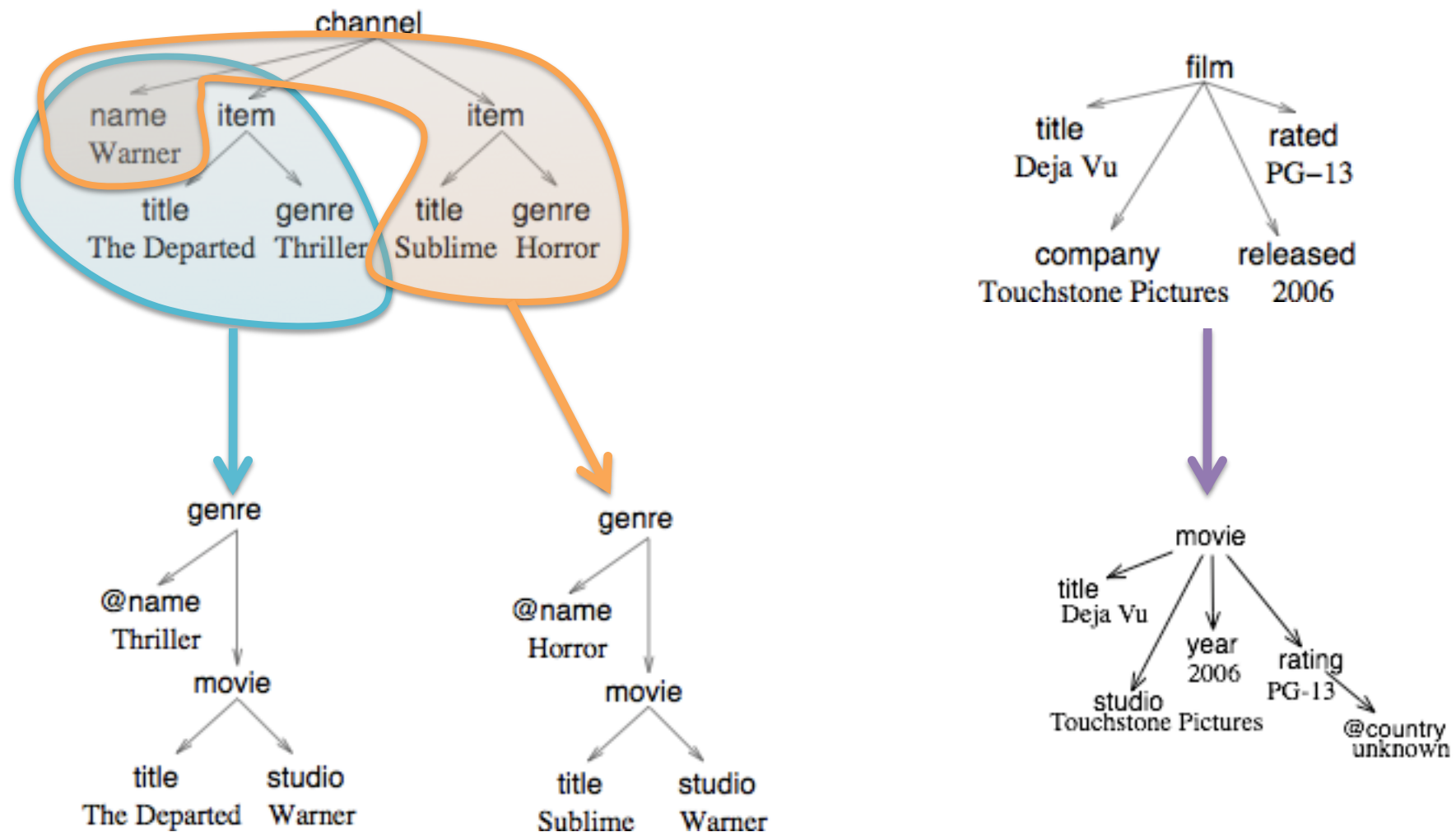
- Updating with the imdb data:

```
UPDATE doc('db.xml') WITH doc('imdb.xml')
```

# Towards Schema-Free Updates

- Goal: translate schema-free update operations into structured updates
- Overview:
  - **Data fitting**: re-format the incoming data (FleDEx)
  - **Anchor discovery**: find which “data items” are already in the database
  - Determine where to effect the update

# Data Fitting: Translating Instances





# Data Fitting – Rewriting Instances

- Once we find the correspondences:
  - We decide **which** data to map and **where** ...
  - We add required elements/attributes

bibitem (author+, editor+, title,  
booktitle, year, pages)

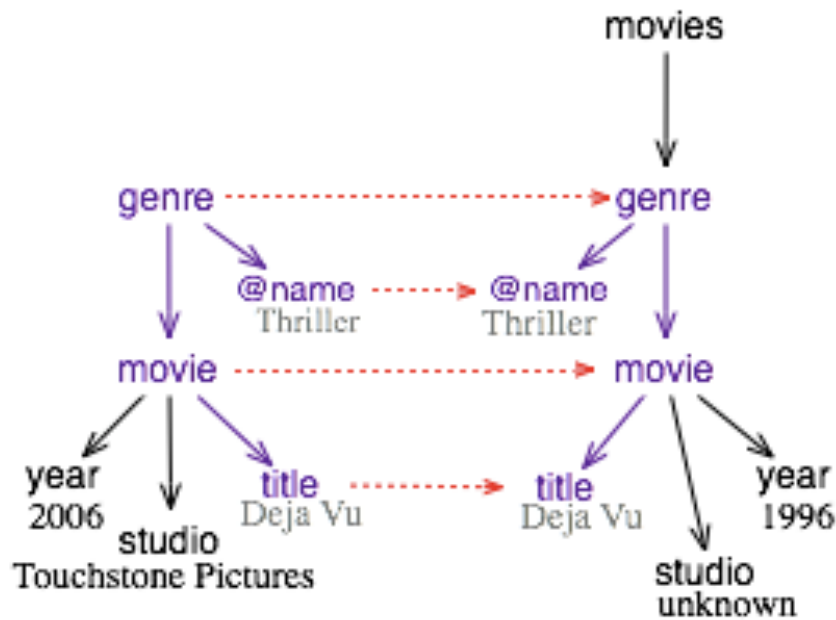
paper (author+, title, proceedings, year, pages)  
article (author+, title, journal, volume, pages)  
proceedings (editor+, title, year, publisher)  
journal (editor+, name, publisher)

```
<bibitem>  
<author>Amihai Motro</author>  
<editor>Won Kim</editor>  
<title>Management of Uncertainty in database Systems</title>  
<booktitle>Modern Database Systems</booktitle>  
<year>1995</year>  
<pages>457-476</pages>  
</bibitem>
```

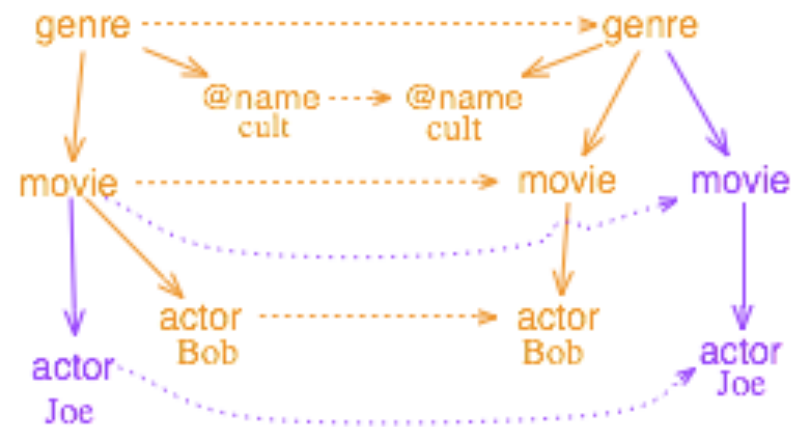
# Anchor Discovery

- **Unambiguous Anchoring**: partial isomorphism between the source and target trees
- **Complete Anchoring**: if a node in the source is anchored, so are all of its ancestors
- Goal: given an XML tree  $t_1$ , determine whether there is a tree  $t_2$  in the database such that  $t_1$  and  $t_2$  “represent the same object”
  - Related work: finding duplicates [Weis et al.'05]

# Ambiguity in Anchoring



Unambiguous anchoring



Ambiguous anchoring

# Anchoring Algorithm

- Input: trees  $t_1, t_2$
- For each tree  $t_j$  in  $t_2$  of the same type as  $t_1$ 
  - Phase 1 (top-down): anchor all pairs of nodes from  $t_1, t_j$  that have the same type
  - Phase 2 (bottom-up): un-anchor pairs of nodes that are not **similar enough**
  - Un-anchor (and remember) all ambiguously anchored nodes

# Node Similarity

- Leaves:  $e$  ,  $a$  are similar if the normalized edit distance of their content low enough ( $<0.3$ )

- Inner nodes:  $sim(e, a) = \frac{w(E_{\approx})}{w(E_{\approx}) + w(E_{\neq})} > \lambda$

$w(\cdot)$  inverse document frequency of the paired values

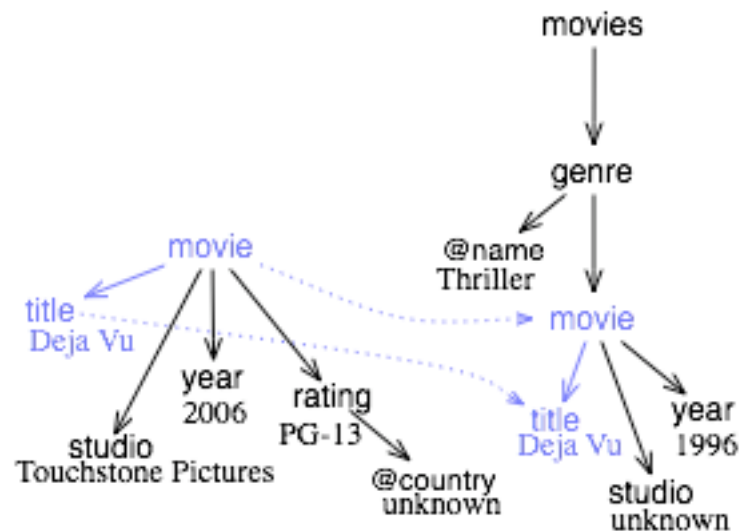
$E_{\approx}$  pairs of leaf nodes that remained anchored

$E_{\neq}$  pairs of leaf nodes with the same type but **not** anchored

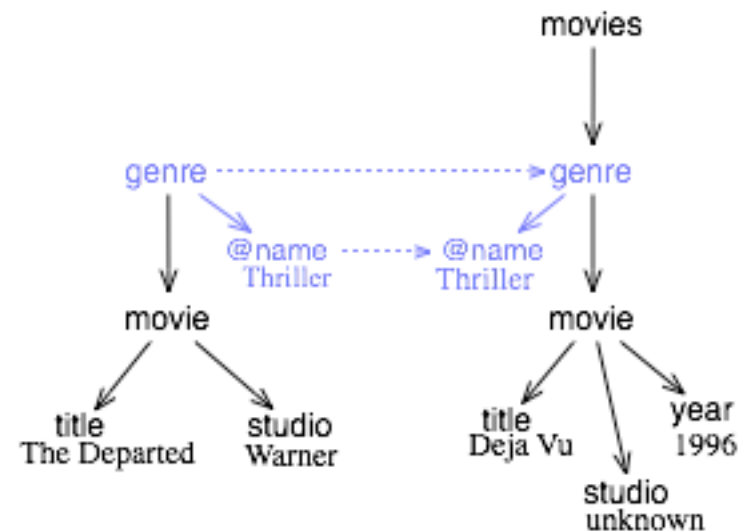
# Conservative Update Semantics

- Updates are performed only with an unambiguous and complete anchoring
- **Insertions:** add all non-anchored nodes
- **Updates:** replace non-anchored nodes that have a corresponding one (i.e., same label)
- **Merges:** update followed by insertion
- **Deletions:** remove anchored nodes from the database

# Semantics: Examples



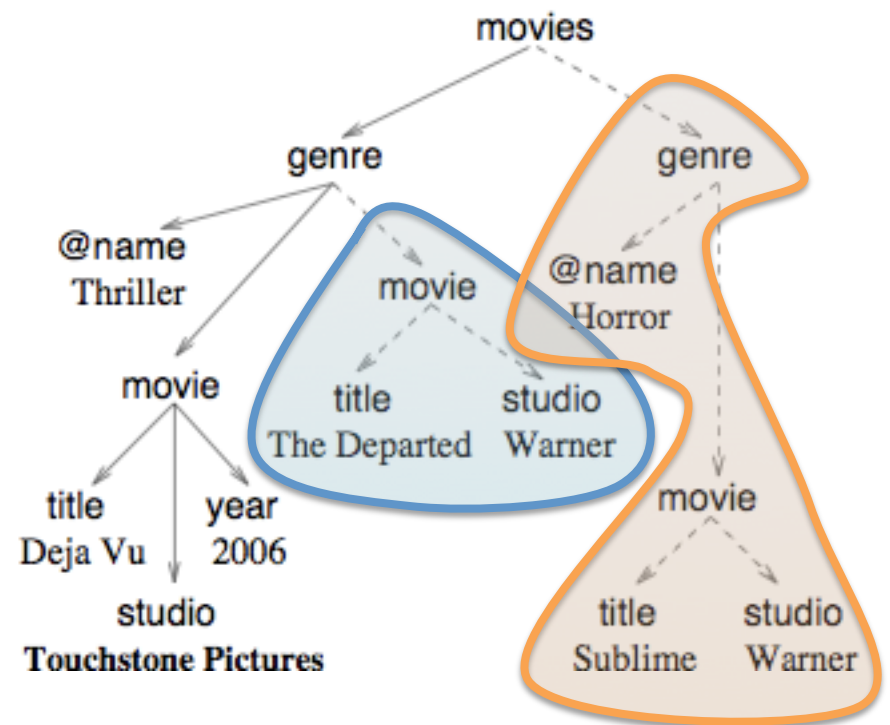
Updating an existing movie



Inserting a new movie

# One Last Step: Where to Insert?

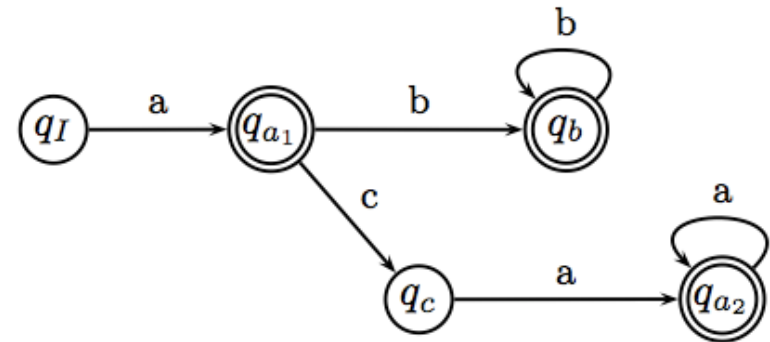
- Problem: we do not want updates that violate the target DTD
  - We need to find all possible insertion points
  - When a tree does not anchor at all, try to find **the place** where other similar nodes are stored





# One Last Step: Where to Insert?

- Idea: build an index that associates positions with labels
- Problem: for arbitrary DTDs, one can't tell the positions without validating the document
- Good news: there are simple DTDs for which this becomes trivial
  - [Arenas&Libkin'03], [ICDE'04], [WIDM'06]



$$l_i \leftarrow a, (b^* \mid (c, a^+))$$

# Experimental Evaluation

- Main test: how many mistakes happen when we try to update a database
- We use real data **only**

Databases	Site	Objects
Movies	<a href="http://imdb.com">http://imdb.com</a>	8,914
Music	<a href="http://musicbrainz.org/doc/Database">http://musicbrainz.org/doc/Database</a>	14,966
Books	<a href="http://dblp.uni-trier.de/xml/">http://dblp.uni-trier.de/xml/</a>	1,211
Articles	<a href="http://dblp.uni-trier.de/xml/">http://dblp.uni-trier.de/xml/</a>	8,000

(a) Test databases.

Source docs	Site	Format
Movies	<a href="http://movies.yahoo.com">http://movies.yahoo.com</a>	HTML
Music	<a href="http://www.pandora.com">http://www.pandora.com</a>	RSS
Books	<a href="http://books.google.com">http://books.google.com</a>	HTML
Articles	<a href="http://www.sigmod.org/record/xml/">http://www.sigmod.org/record/xml/</a>	XML

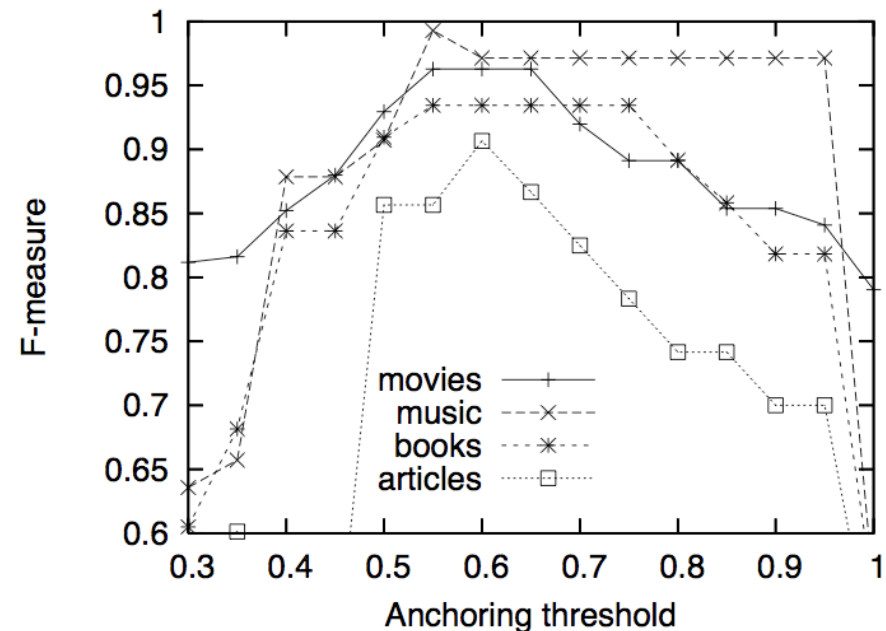
(b) Source documents.

# Accuracy of Data Fitting/Anchoring

- Accuracy = f1-measure

	All	Simple	Complex
movies	0.96	0.97	0.96
music	0.98	0.98	0.98
books	0.95	0.94	0.95
articles	0.95	0.92	0.95

Data Fitting



# Update Correctness – Summary

- Source instance  $S$ , intended updated instance  $I$ , and instance produced by our method  $P$
- Atomic edits needed to convert instances

$$- U_{SI}, U_{SP}$$

- Metric:

$$1 - \frac{(n - c) + (m - c)}{m}$$

$$n = |U_{SP}|, \quad m = |U_{SI}|, \quad c = |U_{SP} \cap U_{SI}|$$

Operations	Existing			New		
	P	R	A	P	R	A
insert	0.99	1	0.99	1	1	1
update	0.55	0.72	0.11	–	–	–
merge	0.88	0.9	0.76	1	1	1
delete	0.98	0.98	0.95	–	–	–

# Conclusion

- Schema-free data management benefits non-expert, casual users
  - Possible because of the “hints” in the data (schema, structure, content)
- State-of-the-art: a long way towards allowing non-experts to store and query data
- We’ve started looking into exchanging and updating Web data
  - Long way to go still

Collaborators in the work mentioned here:

Marcelo Arenas, **Eli Cortez**, Gregory Leighton, Leonid Libkin,  
Alberto O. Mendelzon, **Filipe Mesquita**, Laurent Mignet,  
**Altigran Soares da Silva**, Andrew Smith

# THANK YOU