



DSNotify: Handling Broken Links in the Web of Data

Niko Popitsch, University of Vienna / Austria

niko.popitsch@univie.ac.at

Joint work with Bernhard Haslhofer

bernhard.haslhofer@univie.ac.at

April 30, 2010

WWW 2010 Conference

Raleigh, North Carolina, USA

Outline

- Introduction and problem definition
- Related work and solution strategies
- DSNotify
 - Usage scenarios and design
 - Core algorithm
 - Evaluation
- Summary & Discussion
- References

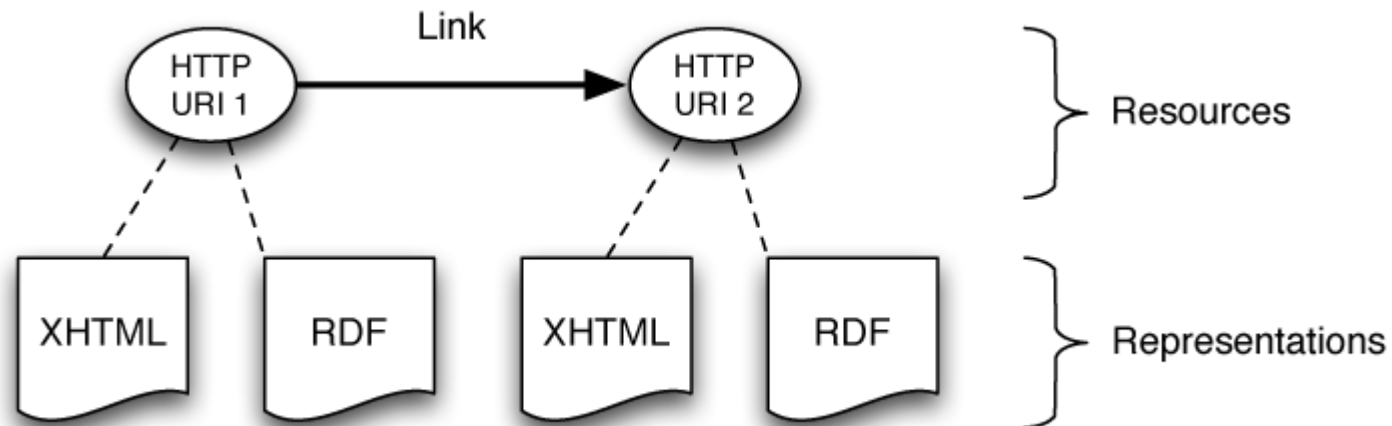




image by TBL / Hans Rosling

Linked Data Principles (short version):

- (1) use **HTTP URIs** to identify resources,
- (2) deliver **meaningful representations** (e.g., RDF, XHTML) when these are dereferenced
- (3) **link** to other resources



BBC - Music - Green Day - Mozilla Firefox


File Edit View History Bookmarks Tools Help

http://www.bbc.co.uk/music/artists/084308bd-1654-436f-ba03-df6697104e19

BBC - Music - Green Day

Green Day

Formed 1989.



Nigel Crane/Redferns

Biography

Green Day is an American rock trio formed in 1987. The band has consisted of lead vocalist and guitarist Billie Joe Armstrong, bassist and backing vocalist Mike Dirnt, and drummer Tré Cool for the majority of its existence. The band is credited as one of the two main bands, along with The Offspring, who put the punk revival into process.


Green Day was originally part of the punk scene at 924 Gilman Street in Berkeley, California. The band's early releases for independent record label Lookout! Records earned it a grassroots fanbase. Nevertheless, its major label debut Dookie (1994) became a breakout success and eventually sold over 10 million copies in the U.S. and 15 million worldwide. As a result, Green

MOST PLAYED ON BBC RADIO 1

Latest Tracks Played On The BBC

Basket Case BBC 6 Music Shaun Keaveny 29/03/2010	▶
Know Your Enemy BBC 6 Music Jon Richardson 28/03/2010	▶
East Jesus Nowhere BBC Radio 1 Dev 23/03/2010	▶
Welcome To Paradise BBC 6 Music Lauren Laverne 18/03/2010	▶
American Idiot BBC 6 Music Steve Lamacq 17/03/2010	▶

Audio Previews From Latest Album Review



21st Century Breakdown	
1 Song of the Century	▶
2 21st Century Breakdown	▶
3 Know Your Enemy	▶

Done

\$ curl -H "Accept: application/rdf+xml" http://www.bbc.co.uk/music/artists/084308bd-1654-436f-ba03-df6697104e19

Links within the data source

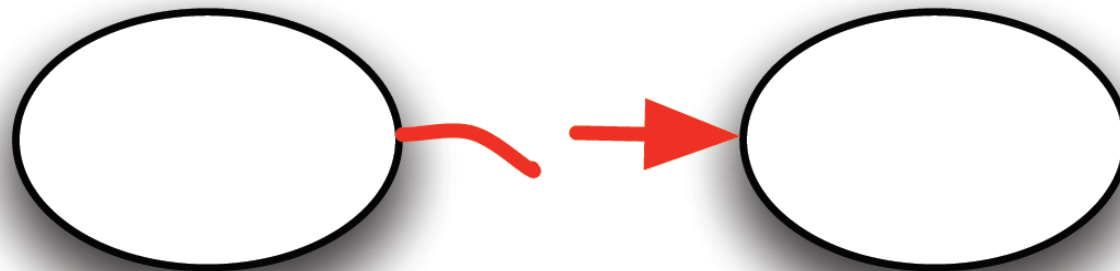
```
[...]
<mo:member rdf:resource="/music/artists/5d06fe54-485a-4a07-b506-5f6f719448cb#artist" />
<mo:member rdf:resource="/music/artists/f332a312-e95b-4413-b6cc-1762a5a6a083#artist" />
<mo:member rdf:resource="/music/artists/0dcee02c-5d2c-4f5c-9d60-d58a4df32d9e#artist" />
[...]
```

RDF links between data sources

```
[...]
<owl:sameAs rdf:resource="http://dbpedia.org/resource/Green_Day" />
<mo:musicbrainz rdf:resource="http://musicbrainz.org/artist/084308bd-1654-436f-ba03-df6697104e19.html" />
[...]
```

```
[...]
<mo:MusicArtist rdf:about="/music/artists/084308bd-1654-436f-ba03-df6697104e19#artist">
  <rdf:type rdf:resource="http://purl.org/ontology/mo/MusicGroup" />
  <foaf:name>Green Day</foaf:name>
[...]
```

Problem: links can break



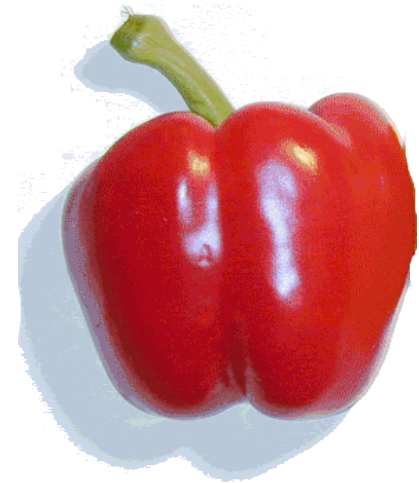
Ignore broken links ? Not a good idea !

Broken links on the Web are annoying for humans

- but alternative paths may be used:
 - search engines, URL manipulation, alternative information providers, etc.

Much harder for machines in a Web of Data !

- reduced data accessibility
- data inconsistencies



Avoid broken links ? Great!

But hard to achieve in the Web environment...

- Solution strategies that solve problem only partially:
 - Relative references
 - embedded links
 - redundancy
- Solution strategies that are not commonly applicable:
 - Versioned/static collections
 - regular (predictable) updates
 - dynamic links
 - indirection services (PURLs, DOIs)



Solve the problem (1/2) : Notification

Notification strategy:

- Data source “knows” about the events that are taking place
- Notifies clients
- Client may then check their links and fix the broken ones

Current Activities:

- **WOD-LMP [Volz et al. 2009]**
- **Triplify Linked Data Update Log [Auer et al. 2009]**
- **PubSubHubbub / sparqlPuSH**
- <http://groups.google.com/group/dataset-dynamics>
- ...



Solve the problem (2/2): Detect and correct

Detect and correct

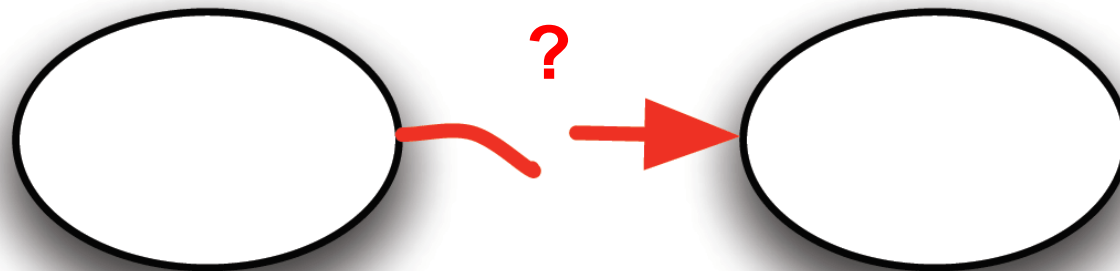
- If notification is not applicable
- Clients detect broken links and try to fix them



Current activities:

- **Robust hyperlinks [Phelps & Wilensky 2000]** – Web documents
- **PageChaser [Morishima et al. 2009]** – Web documents
- **DSNotify** – aims at becoming a general framework for fixing broken links
- ...

What events cause the problem ?



Events that potentially lead to broken links

Broken links due to **deletion** events



- A **deletion** event takes place at time t when a resource had (dereferencable) representations at $t-\Delta$ but has none at time t
- Vice versa: **create** event
- Easy to detect

Events that potentially lead to broken links

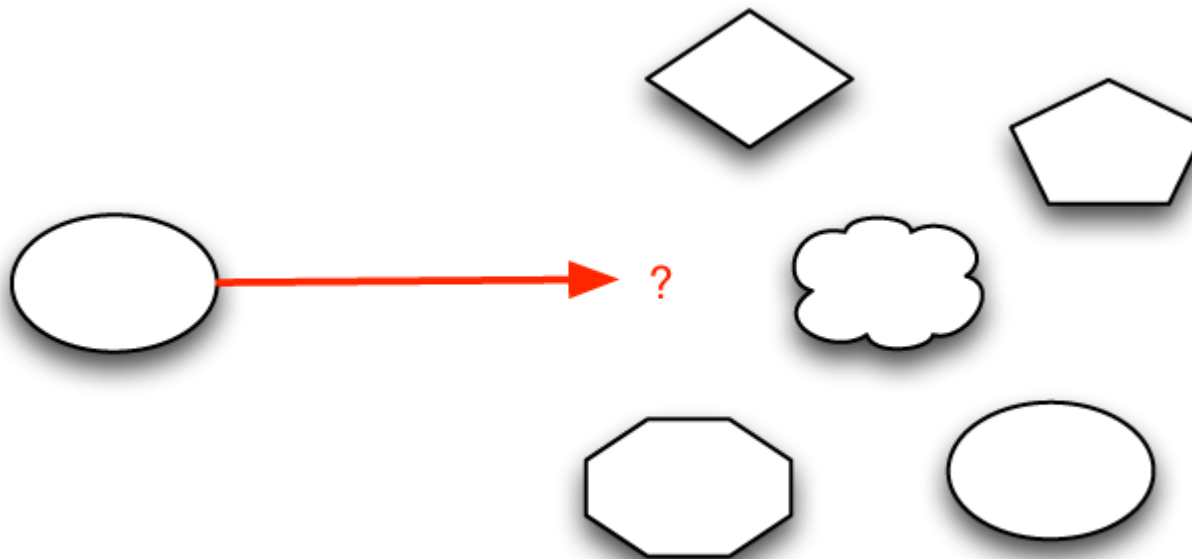
Broken links due to **update** events



- An **update** events takes place at time t when a resource had different representations at $t-\Delta$ compared to the ones at time t
- Resource **updates** resulting in representations with different meaning (**semantic drift**) may lead to **semantically broken links**
- **Hard to detect, open problem**

Events that potentially lead to broken links

What about **move** events ?



Events that potentially lead to broken links

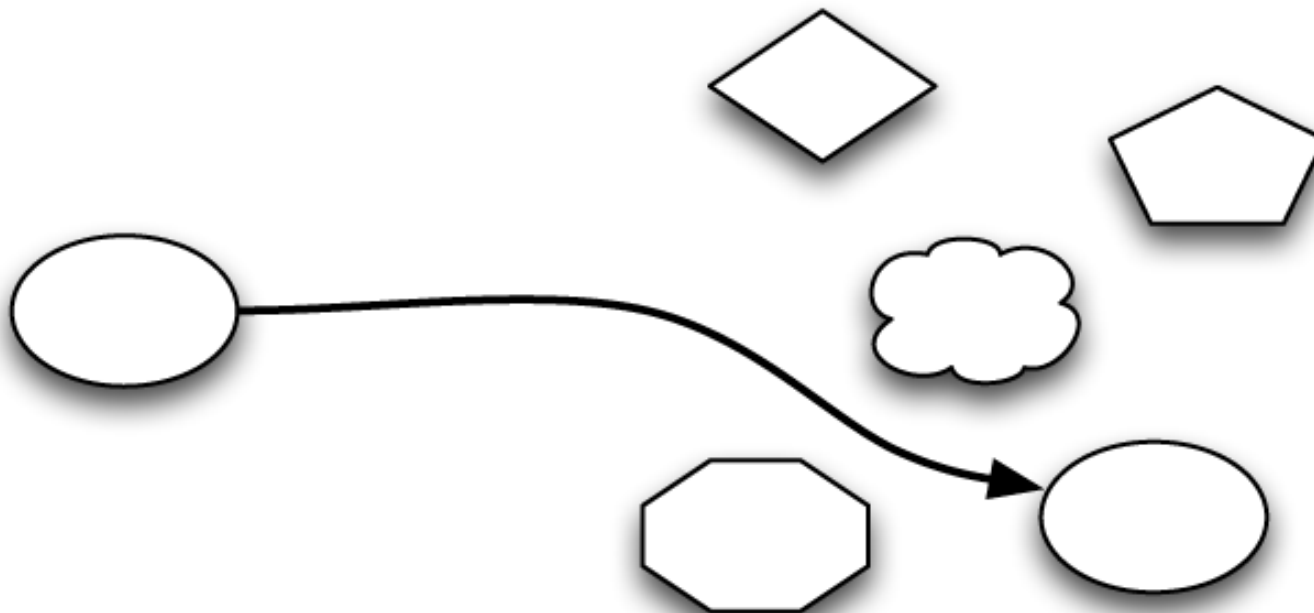
What about **move** events ?



- A **move** event from **a** to **b** takes place at time **t** when
 - There were no representations of **b** at time **t-Δ**
 - There are no representations of **a** at time **t**
 - The representations of **a**_{t-Δ} are more similar to the ones of **b**_t than to the ones of any other considered resource at time **t**
 - The calculated similarity between them is > than a **threshold**
- **Instance matching problem!**

Events that potentially lead to broken links

The core algorithm of DSNotify detects **move events** based on resource **similarities**

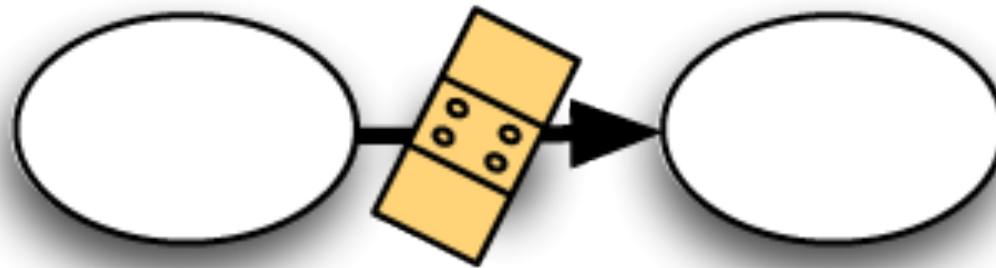


Changes in DBpedia

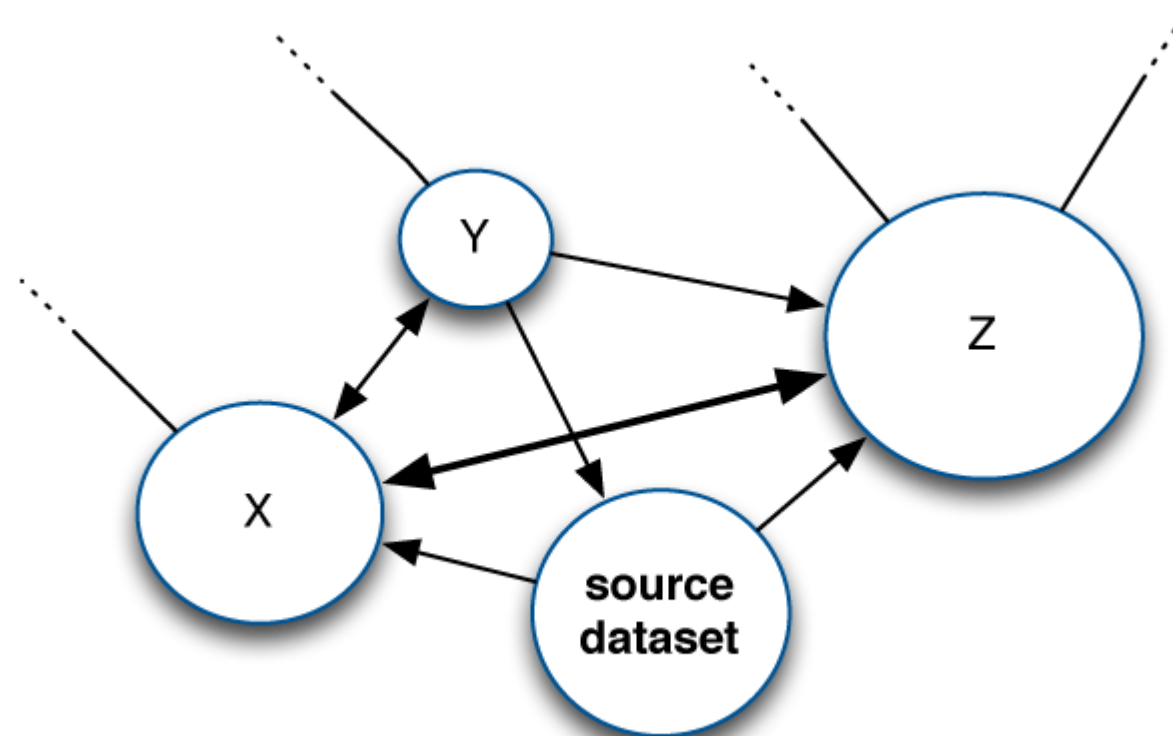
Class	Snapshot 3.2	Snapshot 3.3	Moved	Removed	Created
Person	213,016	244,621	2,841	20,561	49,325
Place	247,508	318,017	2,209	2,430	70,730
Organisation	76,343	105,827	2,020	1,242	28,706
Work	189,725	213,231	4,097	6,558	25,967

Resources that were moved/removed/created between the DBpedia snapshots 3.2 (October 2008) and 3.3 (May 2009)

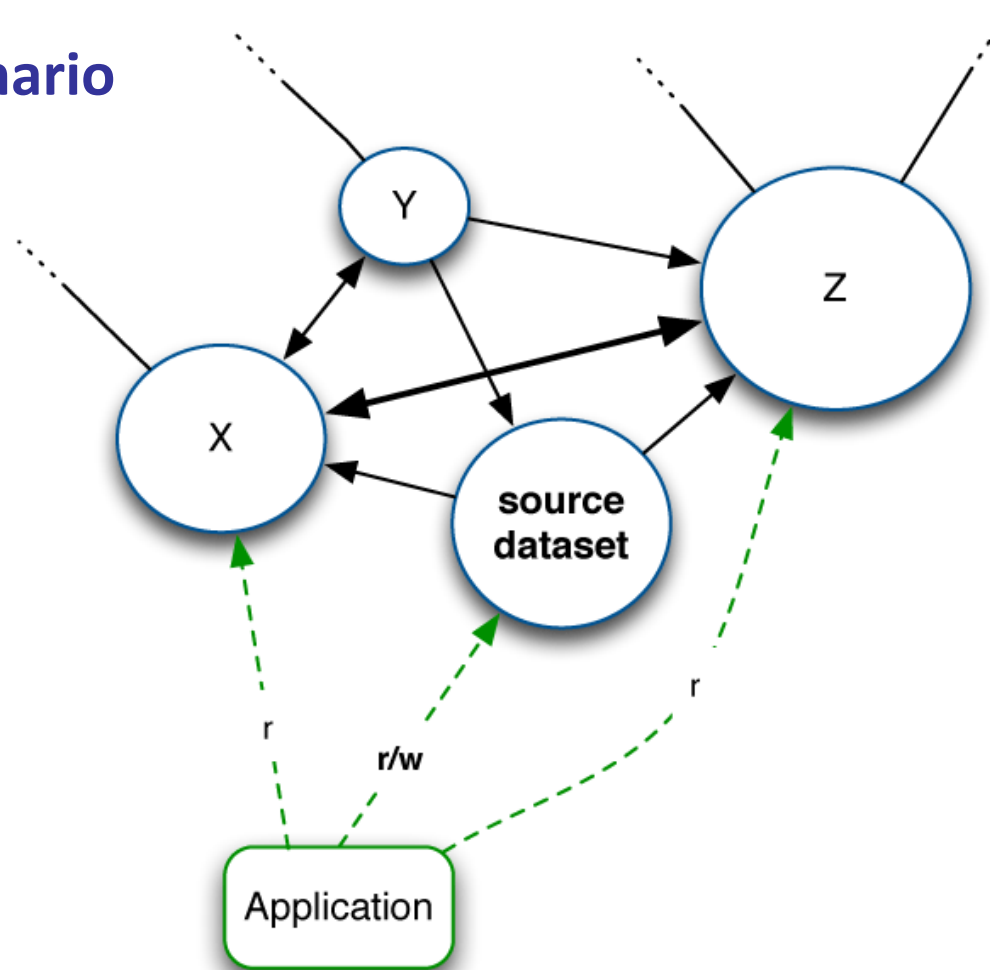
PART 3 : DSNotify



Usage Scenario

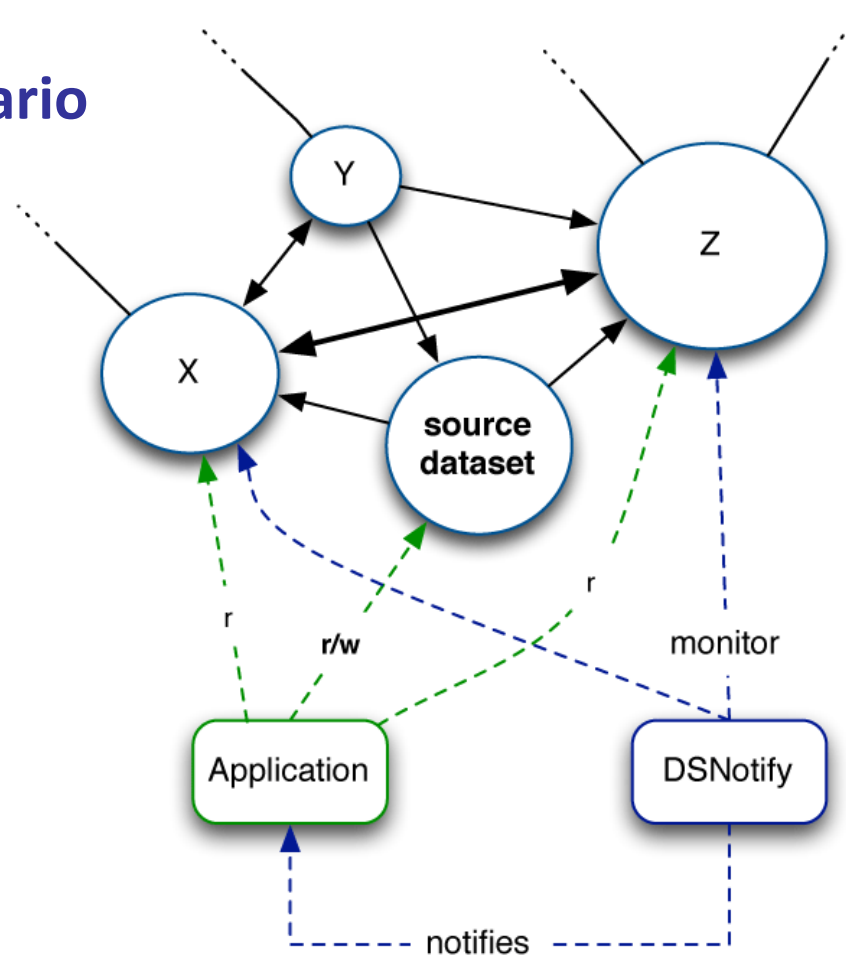


Usage Scenario



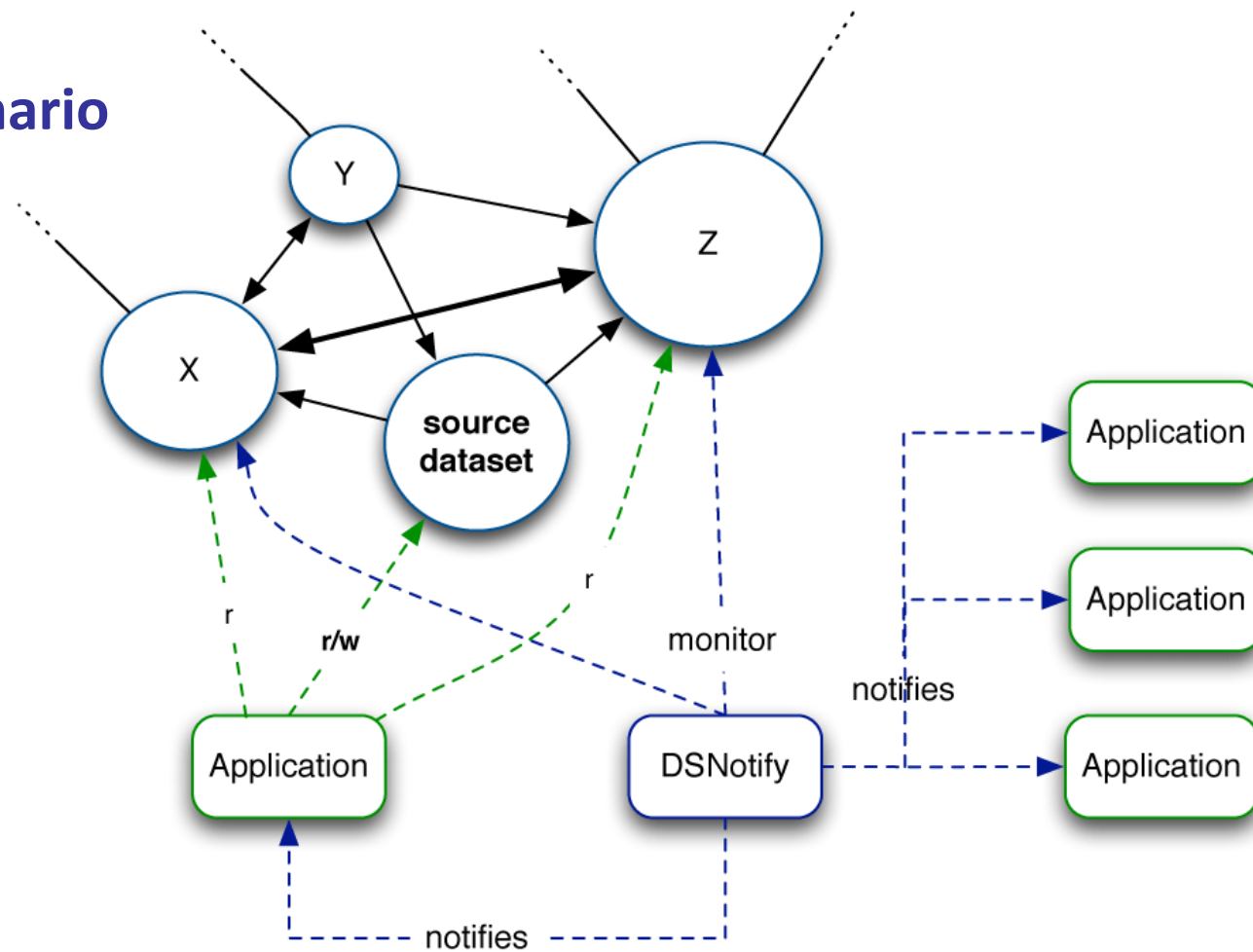
- Application that consumes various LD sources and may update a “source dataset”

Usage Scenario



- DSnotify is an **add-on** for applications that want to preserve **high link integrity** in **their data**

Usage Scenario



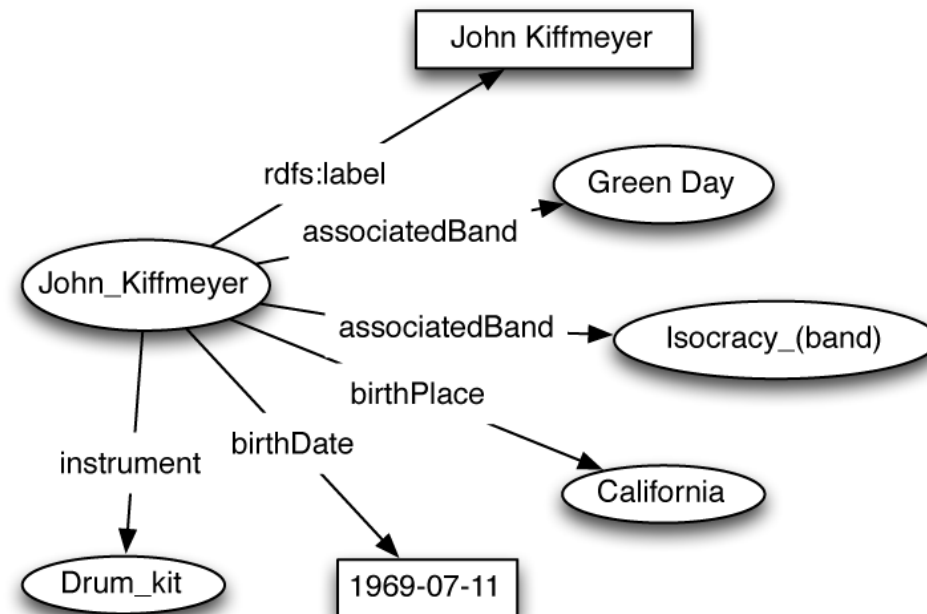
- Other actors (applications) might also be interested in these events

General Approach

- Periodically access linked data sources
- Extract **features** from **resource representations**
- Combine them to **comparable feature vectors** (FV)
- Store them in 3 indices
 - 1st index represents the current state of the monitored data
 - 2nd index stores items that became **recently** unavailable
 - 3rd index stores archived feature vectors
- Periodically access index 1+2 and **log detect events**
- Periodically **update indices** 1-3

From Resource to Feature Vector

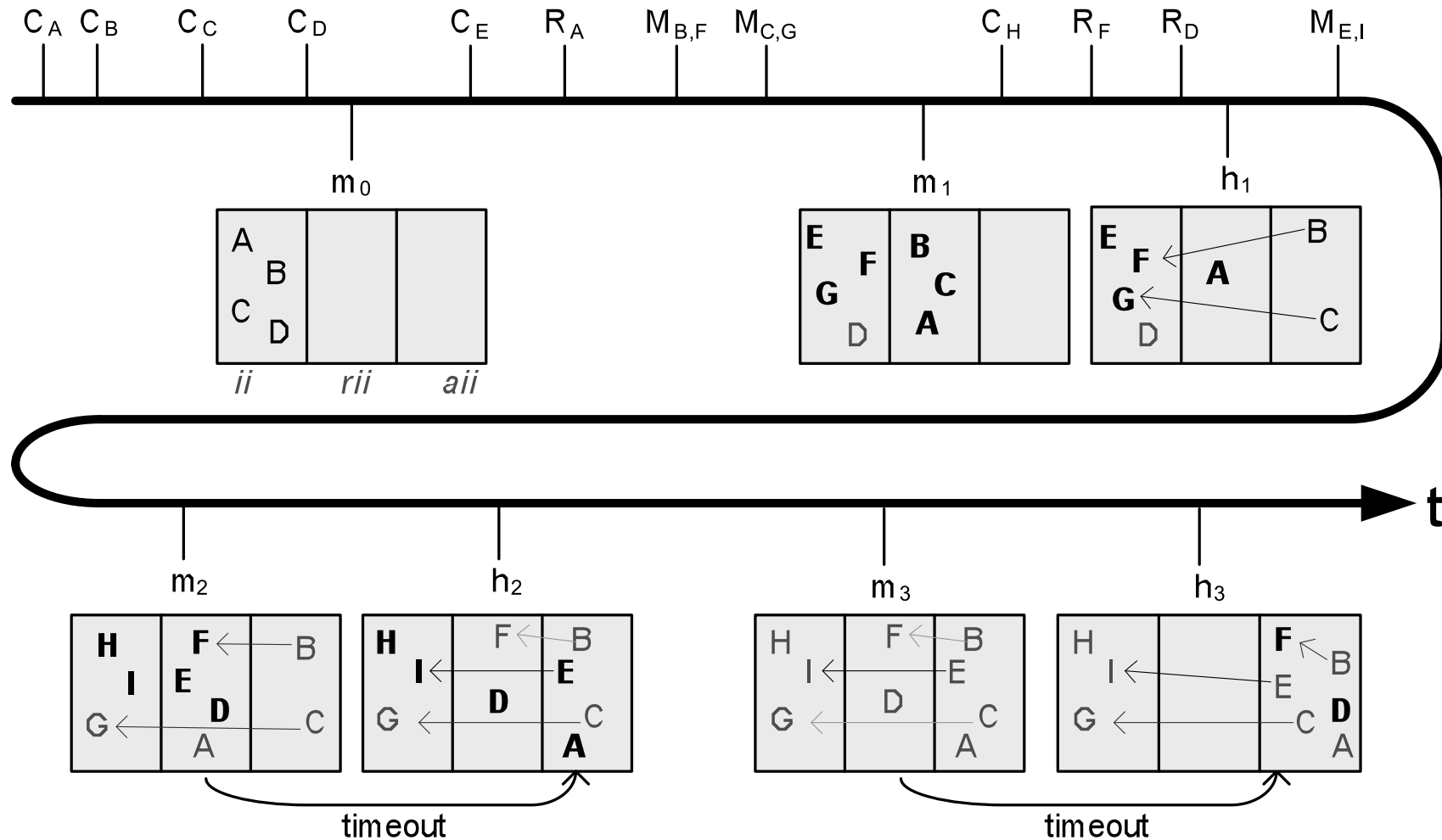
- Both, **data type and object properties** supported
- Feature influence is **weighted**
- Some are used in **plausibility checks**
 - RDFHash over all features



$$\left(\begin{array}{ll} \text{rdfs:label (0.6)} = & \text{John Kiffmeyer} \\ \text{associatedBand (0.2)} = & \text{http://dbpedia.../Green Day,} \\ & \text{http://...Isocracy_(band)} \\ \text{birthPlace (0.1)} = & \text{http://.../California} \\ \text{birthDate (0.1)} = & \text{1969-07-11} \\ \text{RDFHash (-)} = & \text{3E4F123D...} \end{array} \right)$$

Move Event Detection

- **Pair wise comparison** using a vector space model
 - Feature comparison e.g., using Levenshtein similarity.
- It is sufficient to compare **recently added** and **recently removed** feature vectors !
- **Two thresholds** for comparing the similarity between FVs representing created and removed items:
 - **lower threshold**: select predecessor candidates
 - consider URI of added FV as possible new URI of resource represented by removed FV
 - **upper threshold**: decidable by DSNotify?
 - decide whether such a candidate can be automatically selected or whether human user has to be asked for assistance.



C_i, R_i and $M_{i,j}$ denote *create*, *remove* and *move* events of items i and j . m_x and h_x denote *monitoring* and *housekeeping* operations respectively.

Resulting Data Structures

- DSNotify constructs three data structures:
 - An **event log** containing all events detected by the system
 - A log containing all “**event choices**” DSNotify cannot decide on and
 - a **linked** structure of feature vectors constituting a **history** of the respective items.
- Accessible via
 - Linked data interface
 - Java interface
 - XML-RPC

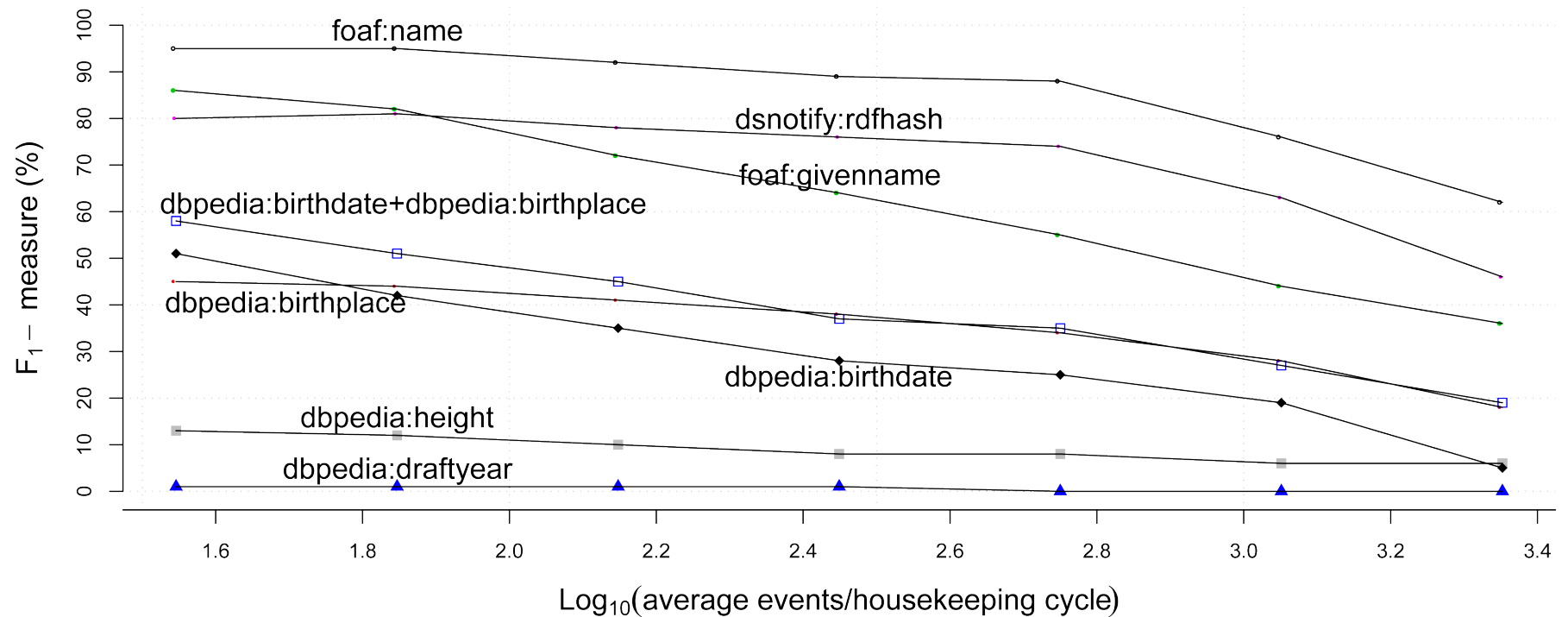


Evaluation

- Core questions:
 - Does DSNotify work with **real data** ?
 - How does **housekeeping frequency** affect its **effectiveness** ?
- Used data:
 - Data from **DBpedia** (8380 events) and **IIMB** (10 x 222 events) were used
 - Hand-picked features based on **coverage** and **entropy** in the data sets
- Results:
 - **Housekeeping frequency** and **data source dynamics** determine the number of FV-pairs that have to be compared (**scalability**)
 - Number of FV comparisons as well as **coverage** and **entropy** of indexed features influence **accuracy** of method

Evaluation - Results

$$F_{\beta} = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$



- Influence of data source agility and housekeeping frequency on the accuracy of the DSNotify algorithm

Discussion

- Broken links are a **considerable problem** in a Web of Data
- The broken link problem is partly a **special case of the instance matching problem**
- **DSNotify** is an **event-based** approach to this problem:
 - DSNotify can be used as an **add-on** for data sources that want to **preserve link integrity** in their data
- **We cannot “cure” the Web of Data** from broken links (but at least alleviate the pain a bit :)

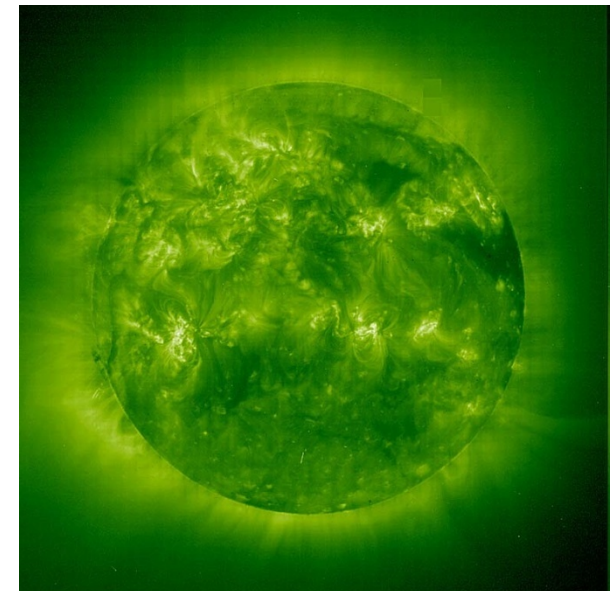
Current and Future Work

- Scalability issues, evaluation
- Automatic feature selection (parameter estimation)
- Event algebra: high-level composite events
- Evaluation with other data sources (e.g., file system)
- Dataset dynamics:
vocabularies, protocols, formats
- ...

Thank You !

niko.popitsch@univie.ac.at

<http://www.dsnotify.org>



Images: NASA / NSSDC

References and Related Work

- H. Ashman. Electronic document addressing: dealing with change. *ACM Comput. Surv.*, 32(3), 2000.
- F. Kappe. A scalable architecture for maintaining referential integrity in distributed information systems. *Journal of Universal Computer Science*, 1(2):84–104, 1995.
- A. Morishima, A. Nakamizo, T. Iida, S. Sugimoto, and H. Kitagawa. Bringing your dead links back to life: a comprehensive approach and lessons learned. In *HT '09: Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 15–24, 2009.
- T. A. Phelps and R. Wilensky. Robust hyperlinks cost just five words each. Technical Report UCB/CSD-00-1091, EECS Department, University of California, Berkeley, 2000
- J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Discovering and maintaining links on the web of data. In *8th International Semantic Web Conference*, 2009.
- A. Hogan, A. Harth, and S. Decker. Performing object consolidation on the semantic web data graph. In *Proceedings of the 1st I3: Identity, Identifiers, Identification Workshop*, 2007
- A. Ferrara, D. Lorusso, S. Montanelli, and G. Varese. Towards a benchmark for instance matching. In *Ontology Matching (OM 2008)*, volume 431 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008
- C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3), 2009
- S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, and D. Aumüller. Triplify: light-weight linked data publication from relational databases. In *WWW '09*, New York, NY, USA, 2009. ACM
- W. Y. Arms. Uniform resource names: handles, purls, and digital object identifiers. *Commun. ACM*, 44(5):68, 2001.