Collaboratively Patching Linked Data
A Patch Repository for Linked Datasets

Magnus Knuth, Johannes Hercher, and Harald Sack
Hasso Plattner Institute, University of Potsdam
USEWOD Workshop @ WWW 2012
April 17, 2012 - Lyon, France
Outline

- Introduction
- Patch Request Ontology
- Architecture / Workflow
- Use Case
  - WhoKnows?
  - Patch Repository
- Outlook
Problem

- the Web of Data is noisy
- dataset not under employers control
- erroneous Data distributed to multiple local data stores
- missing error correction propagation mechanisms
  - for commits, for updates

- examples from DBpedia:
Goals

- a solution that addresses Linked Data error corrections
- an ontology to describe error corrections for Linked Datasets
- a framework to collect Linked Data Patches in a collaborative way
  - explicitly involving data consumers
- a process to propagate Linked Data Patches over multiple local stores
“a simple vocabulary to share data about erroneous triples”

http://purl.org/hpi/patchr
A Patch Request

repo:Patch_15 a pro:Patch ;
pro:hasUpdate [  
a guo:UpdateInstruction ;
guo:target_graph <http://dbpedia.org/> ;
guo:target_subject dbp:Oregon ;
guo:insert [  
  dbo:language dbp:English_language ]
] ;
pro:hasAdvocate repo:Player_25 ;
pro:appliesTo <http://dbpedia.org/void.ttl#DBpedia> ;
pro:status "active" ;
pro:hasProvenance [  
a prv:DataCreation ;
prv:performedBy repo:WhoKnows ;
prv:involvedActor repo:Player_25 ;
prv:performedAt "..."^^xsd:dateTime ] .
Architecture / Workflow

- **Public Dataset (Original)**
- **Patch Repository**
- **Patch Request**
- **Local Copy**

**Flow**:
1. User creates patch
2. User applies (SPARQL Update) to local copy
3. Patch Request is processed by the repository
4. Duplicated data is retrieved and applied

**Roles**:
- User
- Repository

**Uses**
- Uses local copy
- Applies to patch repository
- Retrieves patch

**Concepts**
- Duplication
- Patching
- RDMS
- Linked Data
Workflow

1. find an error
   - by human user or automatic algorithm
2. create a patch
3. commit to (central) repository
   - there should be one responsible repository for each dataset
   - if patch preexists: one more vote
4. other users / dataset providers retrieve patches from repository
   - via SPARQL query
   - customizable to individual requirements
5. apply updates to local dataset
   - easy transformation of patch request to SPARQL Update query
Use Case: *WhoKnows?*

- GWAP generates multiple choice questions from DBpedia facts
- Player identifies wrong triples if the question (or desired answer) makes no sense
- Generating patch from user vote

**DEMO**
http://141.89.225.43/game.html
Patch Repository

- list most recent / most popular patches, individual filtering
- show patches for individual resources

Filtered Patch Requests

<table>
<thead>
<tr>
<th>action</th>
<th>advocates</th>
<th>criticsisers</th>
<th>subject</th>
<th>predicate</th>
<th>object</th>
<th>graph</th>
<th>last time</th>
</tr>
</thead>
<tbody>
<tr>
<td>delete</td>
<td>5</td>
<td>1</td>
<td>dbo:Mexico</td>
<td>dbo:regionalLanguage</td>
<td>dbp:Languages_of_Mexico</td>
<td><a href="http://dbpedia.org/">http://dbpedia.org/</a></td>
<td>2012-03-07 16:37:03</td>
</tr>
<tr>
<td>insert</td>
<td>3</td>
<td>1</td>
<td>dbo:102nd_Intelligence_Wing</td>
<td>dbo:country</td>
<td>dbp:United_States</td>
<td><a href="http://dbpedia.org/">http://dbpedia.org/</a></td>
<td>2012-02-27 09:39:41</td>
</tr>
</tbody>
</table>

DEMO
http://141.89.225.43/patchr/browse.php
LOD Benefits

- collecting patches from crowdsourcing or algorithmic data curation systems
- providing patches for replicated Linked Datasets
  - improving data quality
  - measuring data quality
- sustainability (Use Case: DBpedia): fix errors at their source, i.e. Wikipedia
Outlook

- effective synchronization of patches
- further standardization
- dataset quality evaluation
- API to submit patches
  - validity checking
- advanced trust and access control mechanisms
  - rating patches (vote up/down)
  - provide feedback (comments)
  - reputation management
- pingback to inform dataset providers
Thanks for your attention!

http://purl.org/hpi/patchr-repository