Adding Time to Linked Data: A Generic Memento proxy through PROV

Miel Vander Sande, Sam Coppens, Ruben Verborgh,
Erik Mannens, and Rik Van de Walle

Ghent University – iMinds – Multimedia Lab
Gaston Crommenlaan 8 bus 201, B-9050 Ledeberg-Ghent, Belgium
firstname.lastname@ugent.be

Abstract. Linked Data resources change rapidly over time, making a valid consistent state difficult. As a solution, the Memento framework offers content negotiation in the datetime dimension. However, due to a lack of formally described versioning, every server needs a costly custom implementation. In this poster paper, we exploit published provenance of Linked Data resources to implement a generic Memento service. Based on the W3C PROV standard, we propose a loosely coupled architecture that offers a Memento interface to any Linked Data service publishing provenance.

1 Introduction

Linked Data defines data to be published as resources on the web, uniquely identified by persistent URIs. However, the state of these resources changes rapidly over time, which causes inconsistencies in the links between them. This is of great concern to enterprises maintaining their data archives extensively. Requesting a consistent state of resources at a given point in time is crucial for recovery, analytics and administration purposes. A popular solution is the Memento framework [3]. It provides access to prior versions of web resources through datetime negotiation over HTTP. With a fixed datetime, clients can access a consistent state valid at that time. Unfortunately, every server needs a custom implementation, since there is no uniform way of exposing the relations between the different stored versions.

In Linked Data, *provenance* has been a hot research topic for years. It formally describes where the current resource state originates from. Recently, the W3C Provenance Working Group released the PROV [2] standard, enabling the publication of provenance in a uniform way. Based on uniform provenance, any Linked Data server could feed the Memento framework in a loosely coupled way.

In this poster paper, we extend the Memento framework with provenance discovery and exploit it to enable generic access. We redefine Memento as an independent service, compatible with any Linked Data server publishing provenance. First, we shortly introduce the Memento framework (Section 2) and propose an extended architecture (Section 3). Next, we describe our approach for generic provenance-based content negotiation (Section 4). Finally, we add conclusions and some future work (Section 5).

2 Overview of Memento

Memento is an HTTP framework for accessing prior archived versions of web resources, based on a given datetime. It defines three types of resources:

- Original Resource R_i: an existing resource on the web, of which prior versions are desired.
- Memento $M_{i,j}$: encapsulated prior states of the requested R_i.
- Timegate G_i: a resource supporting content negotiation in the datetime dimension. When requested, it decides on a best matching $M_{i,j}$, where t_j is the given datetime.

An architectural overview is given in Figure 1. The resources R_i, G_i and M_i are connected through hypermedia. When R_i is requested, the HTTP response contains a *Link* header, pointing at G_i. When G_i is requested, it responds with its *Location* header set to the selected $M_{i,j}$ resource. When M_i, j is requested, the response holds link headers to its related R_i and to the previous, the next, the first and the last M_i.

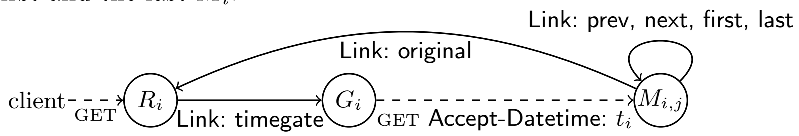

Fig. 1. Hypermedia connecting Original Resource R_i, Timegate G_i and Memento $M_{i,j}$

3 Extended architecture

Memento needs a tailored implementation on every server, due to lacking formal and uniform descriptions about the structure and locations of the different archives. In our approach, we will use provenance of web resources to create a generic Memento implementation as demonstrated in Figure 2. We define two independent types of services: a *Linked Data Service* LDS and a *Generic Memento Proxy* GMP. The former publishes a Linked Data resource L_i and their provenance in PROV P_i, which describes the archives $A_{i,j}$. The latter provides the functionality of the Memento framework, providing access to the resources R_i, $M_{i,j}$ and G_i. The decision logic in G_i now depends on P_i to select a matching $M_{i,j}$, which is explained in Section 4. This results into a loose coupling between LDS and GMP, thus making publishing provenance the only requirement for adding Memento functionality. This reduces costs and effort, while enabling other applications of provenance.

4 Generic time-based content negotiation

When requested, the timegate module decides on a $M_{i,j}$ based on the value t_i from the *Accept-DateTime* header and the descriptions in P_i. We use the semantic reasoner EYE [1] to implement our decision logic, which has three main advantages. First, PROV descriptions can be directly requested as RDF, using the PROV-O[1] ontology, which is natively supported. Second, we can describe our

[1] http://www.w3.org/TR/prov-o/

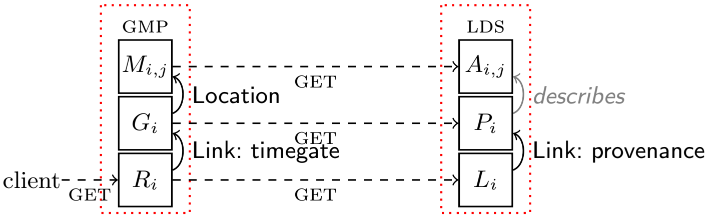

Fig. 2. The Generic Memento proxy GMP and the Linked Data service LDS are loosely coupled, creating a generic architecture

```
@prefix prov:  <http://www.w3.org/ns/prov#>.
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#>.
<Resource> prov:wasRevisionOf <Resource/1>; prov:wasGeneratedBy :rev3.
:rev3 prov:endedAtTime "2012-04-18T14:30:00Z"^^xsd:dateTime.

<Resource/1> prov:wasRevisionOf <Resource/2>; prov:wasGeneratedBy :rev2.
:rev2 prov:endedAtTime "2012-04-15T14:30:00Z"^^xsd:dateTime.

<Resource/2> prov:wasRevisionOf <Resource/3>; prov:wasGeneratedBy :rev1.
:rev1 prov:endedAtTime "2012-04-11T12:30:00Z"^^xsd:dateTime.
```

Listing 1. Provenance Record for `http://example.org/Resource` in PROV-O

logic in only a few compact N3 rules. Third, it can easily be extended with more complex logic if desired later. We can devide the approach into two main steps: discovery of provenance and selecting the memento.

Discovery of provenance Before any decisions can be made, the provenance has to be retrieved. As descibed by the PROV-AQ note[2], a link header pointing to the PROV description is added to an HTTP response. The module will lookup the resource L_i and dereference the URI in the header. As stated above, we will request the provenance in RDF. An example is given in Listing 1.

Selecting the Memento Once the provenance is retrieved, we decide on a memento $M_{i,j}$ using semantic reasoning. The resulting rules are demonstrated in Listing 2. First, we identify all mementos. Each revision is linked to its predecessor using the predicate *prov:wasRevisionOf*, forming a chain of revisions with R_i as endpoint. Relying on the transitive property defined in OWL logic, the relation between Ri and $M_{i,j}$ is derived by adding the triples on lines 5 and 6.

Next, we select a version valid at a given datetime *[line 8]*. The predicate *prov:wasGeneratedBy* links each version to an instance of *prov:Activity*, whose *prov:endedAtTime* predicate indicates the initiation of validity. The rule starts with extracting the defined datetime *[line 10]* and creating a finite list of occuring datetimes. This list is composed by finding all datetimes *[line 11]* that occur on or before the requested datetime *[line 13]*. The valid version occurs on the latest datetime in that list *[lines 15 and 16]*, and is added to the response *[line 18]*. In addition, we define analogue rules to select the first, the last, the next and the previous memento as well, since links to all of these resources are required. The complete rule file can be found here: `http://goo.gl/dz13UN`.

[2] `http://www.w3.org/TR/2013/NOTE-prov-aq-20130430/`

```
@prefix prov:  <http://www.w3.org/ns/prov#>.
@prefix pred:  <http://www.w3.org/2007/rif-builtin-predicate#>.
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#>.
@prefix e:  <http://eulersharp.sourceforge.net/2003/03swap/log-rules#>.
prov:wasRevisionOf rdfs:subPropertyOf :memento.
:memento a owl:TransitiveProperty.

:request :datetime "2012-04-11T12:30:00Z"^^xsd:dateTime
{
  :request :datetime ?req_datetime.
  [] e:findall (?datetime {
    ?rev prov:endedAtTime ?datetime .
    (?datetime ?req_datetime) pred:dateTime-less-than-or-equal true.
  } ?datetime_list) .
  ?datetime_list e:max ?current_datetime.
  ?current prov:endedAtTime ?current_datetime.
} => {
  :response :memento ?current.
}.
...
```

Listing 2. N3Logic selects the Memento valid at a specific Datetime

After the rule execution, the derived result can be mapped directly to the response. We add a *Location* header pointing at $M_{i,j}$ and add the necessary Memento-specific *Link* headers.

5 Conclusion and Future Work

Many enterprises publish their data archives as Linked Data. This requires access to a consistent state of all resources. The Memento framework solves this, but requires a costly custom implementation on each server. In the described approach, we avoid these costs by extending the framework using provenance descibed in PROV. We proposed a loosely-coupled architecture, where a Memento server can operate independently. We explained how semantic reasoning can implement the decision logic in a quick and scalable way.

In a related project, we have created *r&wbase* [4], a triple version control approach for triple stores. In future work, the capabilities of existing interfaces (e.g., SPARQL, Linked Data Platform and the Graph Store Protocol) to access different versions will be investigated. The approach described in this paper simplifies datetime-based version selection for these interfaces.

References

1. J. De Roo. Euler proof mechanism, 1999–2013. Available at `http://eulersharp.sourceforge.net/`.
2. P. Groth and L. Moreau. PROV-Overview: An Overview of the PROV Family of Documents. W3C Working Group Note, 2013.
3. H. Van de Sompel, M. L. Nelson, R. Sanderson, L. Balakireva, S. Ainsworth, and H. Shankar. Memento: Time travel for the Web. *CoRR*, abs/0911.1112, 2009.
4. M. Vander Sande, P. Colpaert, R. Verborgh, S. Coppens, E. Mannens, and R. Van de Walle. R&Wbase: git for triples. In *Proceedings of the 6th Workshop on Linked Data on the Web*, May 2013.