Integrating Linked Metadata Repositories into the Web of Data

Gofran Shukair[1], Nikolaos Loutas[1], and Vassilios Peristeras[2]

[1] DERI, NUI Galway
fistname.lastname@deri.org
[2] European Commission, Directorate-General for Informatics, Interoperability Solutions for European Public Administrations
vassilios.peristeras@ec.europa.eu

Abstract. The heterogeneity of the environment in which data exchange happens is creating many challenges during the execution of digital public services across Europe. Lack of agreement and guidance on the meaning and format of information to be exchanged between Member States are the main stumbling block. Semantic interoperability is jeopardized by different interpretations of the information exchanged between people and applications, thus hampering the effective and efficient creation of new European public services. In an attempt to promote the use of common models and standards, governments develop e-Government metadata repositories to store semantic data models, taxonomies, codelists and reference data making them openly available for reuse. The term semantic asset has been devised to refer to these types of resources. However, these repositories differ in their scope, target group, implementation technologies and end-user interfaces. Although, the semantic content they include can often be reused, their physical isolation and the heterogeneity of the assets hamper the reusability of common concepts and cross-repository search. To deal with these challenges, this paper specifies a generic Exchange API that capitalizes on Linked Data technologies and REST in order to enable the semantic integration of distributed metadata repositories, thus creating a flexible federation which facilitates cross-repository querying and consequently enhances semantic asset reusability.

Keywords: e-Government, semantic assets, Linked Data, ADMS, REST, federation of semantic asset repositories

1 Introduction

As part of their everyday operation, governments need to seamlessly exchange information and cooperate in the provision of public services. In this context, semantic interoperability is recognized as one of the major enablers of e-Government and is perceived as an essential precondition for open, flexible delivery of e- Government services [2]. Interoperability issues can arise especially when the agreement on the meaning of concepts is needed [14]. The European Interoperability

Framework (EIF) emphasizes the importance of semantic interoperability to enable organizations to process information from external sources in a meaningful manner, allowing data elements to be exchanged and understood in the same way [1]. In order to facilitate semantic interoperability, governments worldwide develop standards to support their numerous functions, including provision of e-Government services to citizens and efficient exchange of information among different agencies [8]. These standards include reusable data models, schemata, taxonomies and codelists. The term semantic asset has been devised to refer to these types of resources.

In an effort to house and manage semantic assets centrally and make them accessible for developers, administrators, and project managers working on the creation of public services; governments develop central metadata repositories, i.e., an online system to host semantic assets and allows users to access these assets along with their metadata descriptions. Examples of such repositories include Digitaliser.dk[3] which hosts semantic assets used by the Danish government and Joinup[4] (formerly known as Semic.eu) which lists assets generated and used across Europe.

Existing semantic asset repositories use heterogeneous terminologies and data models and provide different access mechanisms. The absence of a standard way to represent the content of semantic asset repositories hampers any effort to utilize the data across repositories essential to enable cross-repository search and data analysis tools for instance. To address this problem, the Asset Description Metadata Schema (ADMS), a common meta-model for semantic assets is developed to enable repository owners to publish their assets metadata in machine-readable format following the Linked Data guidlines[5].

However, a recent survey of the metadata management efforts across Europe indicated the importance of the federation of the semantic asset repositories to boost asset reuse across Europe as common practice. This survey was conducted by the Community of European Semantic Asset Repositories (CESAR)[6] during a workshop in Brussels, March 2012. The majority of the participants stated that a federation of semantic asset repositories increases the visibility of semantic assets, making them more accessible and therefore stimulating their reuse [4]. In order to build this federation, a technology-neutral agreement to exchange data between these heterogeneous repositories and the federation is needed. To address this, we specify the ADMS Exchange API and we use it to implement an extensible federation of semantic asset repositories. This federation consumes Linked Data from four major existing semantic asset repositories and provides a single interface to enable users to search, select, and obtain semantic assets from different repositories.

The remainder of this paper is organized as follows: Section 2 introduces background information about ADMS vocabulary and the federation approach.

[3] http://www.digitaliser.dk
[4] http://joinup.ec.europa.eu/
[5] http://www.w3.org/DesignIssues/LinkedData.html
[6] http://joinup.ec.europa.eu/community/cesar/description

In section 3, we present the RESTful ADMS Exchange API specification. Section 4 demonstrates our pilot implementation of the federation. Section 5 discusses related efforts on web services and APIs and highlights few federated repositories. Finally, section 6 concludes the paper and discusses our future research plan.

2 Background

In this section, we introduce the ADMS vocabulary and we discuss our federation approach and its characteristic.

2.1 The Asset Description Metadata Schema (ADMS)

ADMS[7] is a common vocabulary to describe semantic interoperability assets making them easier to search and discover once shared through the forthcoming federation of asset repositories. It is an initiative of the ISA[8] programme of the European Commission [3]. Figure 1 illustrates the main concepts and properties of ADMS[9]. It has three main concepts:

1 A *Semantic Asset Repository* is a system or service that provides facilities for storage and maintenance of descriptions of Semantic Assets and Semantic Asset Distributions, and functionality that allows users to search and access these descriptions.
2 A *Semantic Asset* is an abstract entity that reflects the intellectual content of the asset and represents those characteristics of the asset that are independent of its physical embodiment.
3 A *Semantic Asset Distribution* represents a particular physical embodiment of a Semantic Asset. A Distribution is typically a downloadable computer file (but in principle it could also be a paper document) that implements the intellectual content of an Asset.

Explaining the ADMS design and development methodology is not in the scope of this paper. However, it is worth mentioning that ADMS is the result of a collaborative standardization effort carried out in the context ADMS Working Group[10], i.e., a hybrid community formed to review and stabilize the ADMS model consisting of 33 persons from standardization bodies, academia, IT industry, European Institutions and European public administrations. We were members of this group and participated actively in the design of ADMS. Version 1.0 of ADMS was recently published that recommends RDF and XML distributions of the model. The RDF representation of ADMS (referred to as ADMS/RDF henceforth) reuses existing vocabularies as far as possible and is aligned with the DCAT vocabulary recently published by the W3C Government Linked Data Working Group[11].

[7] http://joinup.ec.europa.eu/asset/adms/home
[8] http://ec.europa.eu/isa/
[9] The full RDF model can be accessed at http://joinup.ec.europa.eu/asset/adms/release/100
[10] https://joinup.ec.europa.eu/asset/adms/document/adms-working-group
[11] http://www.w3.org/TR/vocab-dcat/

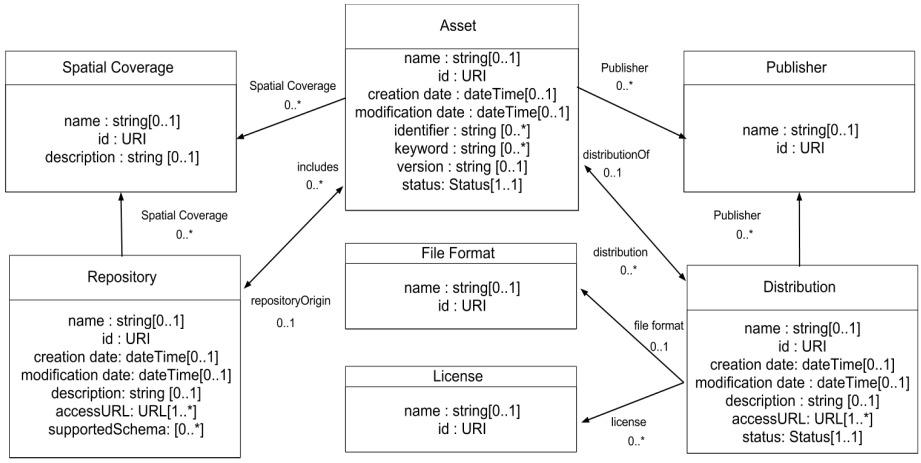

Fig. 1. ADMS Overview : conceptual model

2.2 Federation of Semantic Asset Repositories

ADMS provides the data model to describe repositories contents and integrating them directly with the Web of Data, but this leaves many open challenges when building a federation. These include accessing and querying individual repositories and efficiently retrieving updated content without having to retrieve the whole content. Generally, in data integration over distributed sources, we can distinguish two broad classes of approaches [6]:

a Virtual Data Integration (Distributed Query Processing), where queries are evaluated against the distributed sources by splitting the query in appropriate sub-queries and combining the results from the remote sources.
b Data Warehousing, where all data is collected in advance, preprocessed and stored in a central database, and queries are evaluated against this central database.

In practice, the first approach is more demanding as integrated systems are very likely to be heterogeneous using different database systems and structure the data using different schemas making it hard to query them efficiently, where Data Warehousing approach allows coping with heterogeneity of sources technologies and provides optimized performance by avoiding network overload in query processing, given that individual repositories do not change frequently [10]. For these reasons, we choose to build our integration solution using Data Warehousing approach. We use ADMS as the common data model between the integrated repositories and we specify the ADMS Exchange API to: *(i)* seamlessly exchange the semantic assets metadata from different repositories, *(ii)* keep the federation of semantic assets repositories up to date. The following sections provide detailed descriptions of these two main building blocks.

3 RESTful ADMS Exchange API

In this section, we introduce the design principles of the ADMS Exchange API and its specification.

3.1 API Design principles

Following Linked Data guidelines, repositories owners should maintain dereferenceable URIs for their repository description and ADMS can be used to provide the meaningful exchange of data. However, this leaves the challenge of efficient access to the data not fully addressed. In particular, keeping the federation up to date by retrieving the whole content of individual repositories (via URI dereferencing). Additionally, repositories containing a large number of assets might choose not to provide the full list of their content as a one block so that some sort of pagination is needed. Therefore, we define a simple API that is easy to implement by each individual repository and yet sufficient to address these issues.

The RESTful ADMS Exchange API, is built to (i) retrieve ADMS descriptions of semantic assets from distributed repositories and (ii) keep the federation data up to date. This API is a client-server approach governed by the following principles:

1 REST architectural style, i.e., decoupling the relation between the client and the server side.
2 Hypermedia as the engine of application state (HATEOAS), i.e., a client interacts with a network application entirely through hypermedia provided dynamically by application server [9].
3 Linked Data representation, i.e., universal interface using HTTP methods, universal addressing and access scheme using URIs and simple extensible data model with RDF.

3.2 API Specification

In this section, we propose a simple specification that orchestrates the communication between the API client (the federation system) and the server (each individual repository participating in the federation). The federation of semantic assets repositories needs to harvest the repositories descriptions and their assets metadata and keep the assets synchronized with their original source.

Table 1 summarizes the API specification indicating two main resources: *repository* and *assets collection* and their associated HTTP method. Both resources are read-only and support at least RDF and HTML representations, through content negotiation, to serve humans and machines consumers.

Table 1. REST API Specification

Resource	HTTP Method	Description
repository	GET	a representation of the repository using native ADMS
assets	GET	a representation of a collection of assets

Using ADMS, each repository publishes its assets metadata online making them available for harvesting. The repository can simply host its ADMS description along with its assets metadata under the repository resource URI, e.g.,*<repositoryBaseURI>/adms/repository*. Dereferencing this URI returns data that can be handled by the client. However, repositories containing hundreds or even thousands of assets will result in a very big data to be efficiently transmitted in a single HTTP response. To address this, we define another resource called Assets Collection, e.g., *<repositoryBaseURI>/adms/assets*, representing assets included in the repository. The URI of the assets collection can be indicated as an RDF property in the repository resource description. If this is the case, assets metadata are no longer returned in the response when calling the repository URI and instead the URI of the assets collection is returned as a part of the repository description guiding the client or any API consumer to the assets metadata location.

The client (federation) starts harvesting by requesting the repository resource from the server (repository) to get its metadata. The federation processes the repository metadata file. If the assets metadata are included, the federation populates the assets into its triple store, otherwise, the federation calls for the assets collection to get the assets metadata. Defining the assets collection resource is optional and dedicated for large repositories that might have thousands of assets.

Furthermore, the API supports a paging technique as described in [12]. When the client calls assets collection URI of a big repository the server will redirect the call - using a HTTP-303 See Other - to the first page of assets, e.g., *<AssetsCollectionURI>?firstpage*. A simple RDF pattern can be returned to guide the client to the rest of the assets following the HATEOAS principle, i.e., Hypermedia as the engine of application state (see Listing 1).

Listing 1. RDF pattern for paging technique for a huge assets collection : *<AssetsCollectionURI>?firstpage* (based on [12])

```
@prefix rdf:<http://www.w3.org/1999/02/22−rdf−syntax−ns#>.
@prefix bp:<http://open−services.net/ns/basicProfile#>.

<http://example.org/netW/nw1/assetCont> a bp:Container;
<http://example.org/netW/nw1/assetCont?firstPage> a bp:Page;
    bp:pageOf <http://example.org/netW/nw1/assetCont>;
    bp:nextPage <http://example.org/netW/nw1/assetCont?p=2>.
```

In the case of updates request, there is no need to get all the assets in the repository. We only need the assets updated or added after the last harvesting

date. To address this, our API supports a *date* parameter that can be associated with assets collection URI, e.g., *<AssetsCollectionURI>?date=date_of_last_call*. When the client request updates, the server will return all assets added and/or updated after the date indicated in the call. We created a simple vocabulary to describe the assets collection resource called the ADMS API Vocabulary[12] consisting of the *admsapi:AssetsCollection* resource and the *admsapi:assets* property connecting the *adms:Repository* resource with its *admsapi:AssetsCollection*. Listing 2 shows an example of the RDF data returned when de-referencing small repository resource URI, where ADMS descriptions of the assets are included in the same response along with the repository description. Listing 3 shows of the RDF data returned when de-referencing big repository resource URI, where assets are no longer included in the response and the URI of assets collection is indicated (see line 3 in Listing 3).

Listing 2. ADMS repository representation - small repository

```
1  @prefix : <http://www.w3.org/ns/adms#> .
2  @prefix dct: <http://purl.org/dc/terms/> .
3  @prefix rdf: <http://www.w3.org/1999/02/22−rdf−syntax−ns#> .
4  @prefix rdfs: <http://www.w3.org/2000/01/rdf−schema#> .
5
6  <http://vocab.deri.ie> a :SemanticAssetRepository;
7      dct:description "DERI Vocabularies";
8      dct:hasPart <http://vocab.deri.ie/void>,
9                  <http://vocab.deri.ie/search>;
10     :accessURL "http://vocab.deri.ie/neologism/adms";
11     :supportedSchema "1.0" .
12 <http://vocab.deri.ie/void> a :SemanticAsset;
13     dct:coverage <http://sws.geonames.org/2963597/>;
14     dct:isPartOf <http://vocab.deri.ie>;
15     dct:language <http://dbpedia.org/resource/English_language>;
16     dct:title "Vocabulary of Interlinked Datasets (VoID)";
17     :interoperabilityLevel <http://example.org/adms/voc/interopLevel/Semantic>;
18     :status <http://example.org/adms/voc/status/Published>;
19     :distribution <http://vocab.deri.ie/void#ttl>.
```

Listing 3. ADMS repository representation - big repository

```
1  @prefix admsapi: <http://vocab.deri.ie/admsapi#> .
2  <http://vmudi205.deri.ie:8080/digitaliser/adms/repository> a :SemanticAssetRepository;
3    dct:title "Digitaliser (Denmark)";
4    admsapi:assets <http://vmudi205.deri.ie:8080/digitaliser/adms/assets>;
5    :accessURL "http://vmudi205.deri.ie:8080/digitaliser/adms/repository";
6    :supportedSchema "1.0".
```

[12] http://vocab.deri.ie/admsapi

4 Pilot implementation

This section describes an implementation of a federation of repositories based on ADMS and the data exchange API described in section 3.2. As we emphasize later on, this pilot implementation is fully functional. It is built using real-world semantic asset repositories and accesses their actual content. In order to include a semantic asset repository that implements the API to the federation the following process has to be followed:

1 The repository manager registers the repository URI through a simple UI.
2 The federation of semantic asset repositories requests the repository content at the registered URI.
3 The repository responds with the ADMS/RDF repository description, optionally partitioning long lists into pages as defined in the API specification.
4 Periodically, the federation manager checks the individual repositories for updates supplying the last update date it has of each repository.
5 The repository responds with recently added and/or updated ADMS/RDF descriptions of semantic assets.
6 The federation system updates its triple store accordingly.

To evaluate this scenario, we first implemented the client side in a form of online portal to register and harvest the repositories according the API specification and then storing their descriptions in a triple store. We used Sesame[13] to run the triple store to store the harvested data. Additionally, a faceted browsing functionality is provided to explore and search for semantic assets. Then we implemented the server side of the API for each of the four selected repositories. The following four semantic asset repositories partake in the federation :

a Digitalisr.dk, which is an e-Government metadata repository from Denmark. It hosts e-Government-related documents, software, technical specifications and standards, and XML schemata.
b DERI vocabularies[14] is a URI space for RDF Schema vocabularies and OWL ontologies maintained at DERI[15] from Ireland.
c ESD Standard Service Lists[16] from the UK serves as an Metadata registry (MDR) where controlled lists, cross-references and data interchange standards for use by the public sector are stored and maintained.
d Joinup[17] (formerly Semic.eu) is an initiative led by the European Commission through the ISA program to foster the reuse of syntactic and semantic assets across Europe.

We chose these repositories because they all host reusable semantic assets but they differ in their target group, size and implementation technologies. For example, Digitaliser has thousands of assets and it is targeting government agencies,

[13] http://www.openrdf.org/
[14] http://vocab.deri.ie/
[15] http://www.deri.ie
[16] http://standards.esd.org.uk/
[17] http://joinup.ec.europa.eu

while DERI vocabularies has less than 60 assets and it is targeting academic audience. Moreover, some of them support ADMS inherently, while others do not.
Table 2 illustrates the difference between these repositories in terms of number of assets, the mechanism they offer to access their assets metadata and their support for ADMS.

Table 2. Repositories partaking in the pilot implementation of the federation

	Digitaliser.dk	DERI Vocabularies	ESD Standards	Joinup
Country	Denmark	Ireland	UK	EU
Number of assets	>8000	60	91	516
Access mechanism	REST	URIs	SPARQL endpoint	-
ADMS Support	No	Yes	No	No

In order to cope with these differences, we built custom wrappers to serve the API client. In the case of Digitaliser.dk, a wrapper for its REST API was built. It requests the REST API to get the assets represented in the digitaliser metamodel, maps them to ADMS and then publishes them in ADMS/RDF format on-the-fly[18]. DERI Vocabularies inherently support ADMS under its repository URI : `http://vocab.deri.ie/neologism/adms`, so there was no need to implement a wrapper. This URI can be registered in the federation and it can be harvested by the client. Both Digitaliser.dk and DERI Vocabularies wrappers return the latest feed of their assets when they are requested by an API client. For other cases, We implemented a configurable wrapper of ADMS/RDF SPARQL endpoint to preform as API server[19]. The ESD repository offers access to its metadata descriptions through a SPARQL endpoint with data represneted in SKOS[20] and Joinup has no access mechanism. For these repositories, we mapped their metamodels manually, then we converted a sample data into RDF[21] and stored each of them in a triple store. We configured our SPARQL wrapper for both of them. Any semantic asset repository that maintain a SPARQL endpoint of its ADMS data can use this wrapper and participate in the federation.

The running pilot of the federation is available at `http://vmudi205.deri.ie:8080/apiclient/home`. Figure 2 illustrates the federation's user interface.

[18] A demonstration of the repository resource can be accessed at: `http://vmudi205.deri.ie:8080/digitaliser/adms/repository.rdf`.
[19] See github repository https://github.com/gofranshukair/ADMS-API-Prototypes
[20] http://www.w3.org/2004/02/skos/
[21] 8 assets from ESD and 49 from Semic.eu

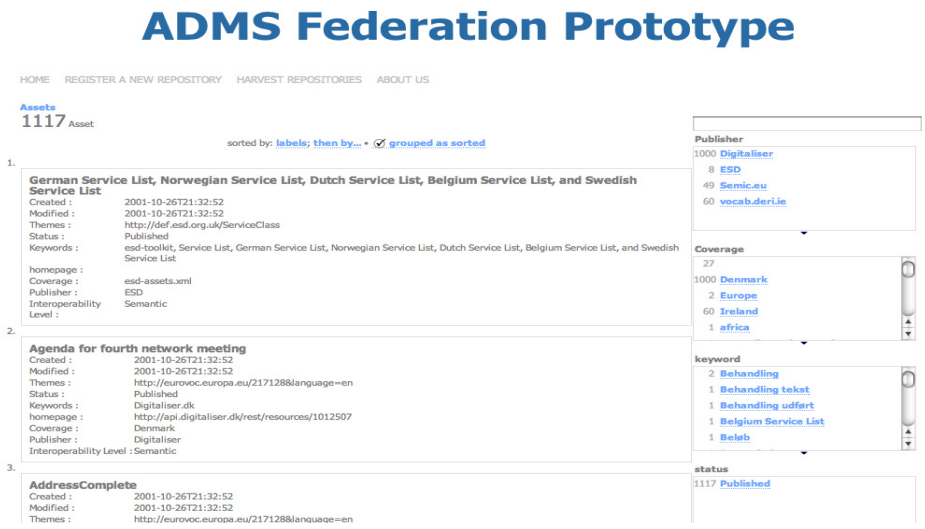

Fig. 2. ADMS Federation Prototype

5 Related Work

In this section, we study a few Web Service descriptions, APIs and federated repositories initiatives that can be compared to our work.

Web Service descriptions and APIs : SOAP [13] and Web Service Description Language (WSDL) [7] are popular Web service technologies. The design idea behind these technologies is to use HTTP as a transport protocol for API calls, and the API completely hides the resources which are handled by the application [15]. These two approaches leads to a tightly coupled systems that contradict with the Semantic Web nature, where resources (identified by URIs) represented in open and well-known vocabularies, making them self-descriptive to avoid misinterpretations. Our proposed API suggests publishing resources openly following Linked Data principles. Using native ADMS as a representation vocabulary without any application-specific annotations. In this way, any application that understands ADMS can consume the data.
Moreover, the Open Data Protocol (OData)[22] is a Web protocol for querying and updating data of a variety of applications, services, and stores. OData does this by applying and building upon Web technologies such as HTTP, AtomPub and JSON. In metadata repositories integration, we adopted RDF and Linked Data principles to query, update and integrate data.
Recently, the Linked Data Platform (LDP) Working Group [23] was launched to produce a W3C Recommendation for HTTP-based (RESTful) application integration patterns using read/write Linked Data. In our proposed approach, we de-

[22] http://www.odata.org/
[23] http://www.w3.org/2012/ldp/wiki/Main_Page

liberately try to conform the Basic Profile Resources (BPRs)[24] recently submitted to W3C, where each repository resource is provided in application/rdf+xml representation and explicit *rdf:type* for HTTP-GET requests.

Federated repositories : Other semantic asset federated repositories do exist, e.g., Open Ontology Repository (OOR) Initiative [5] and Oyster [11]. However, they follow different approaches. For example Oyster offers a user-driven approach where each user has its own local repository of asset metadata and also has access to the information of others repositories, thus creating a virtual decentralized repository. The Open Ontology Repository (OOR) Initiative encourages ontology developers to share their ontologies (assets) in the system and offers interfaces for ontology lifecycle management, alignment..etc. Both approaches expect owners to publish their assets in the federation by themselves while ours is a distributed approach based on the current Web infrastructure and requires no changes to repositories existing publishing workflows.

6 Conclusion and Future Work

Many technical, strategic and organizational challenges have to be faced before the federation becomes a reality, since the vast majority of semantic asset repositories have not foreseen the need to exchange data with external systems. As discussed in this paper, solutions are already emerging with the ISA efforts to raise awareness of the importance of the obscure semantic assets and to encourage sharing and reusing them across borders and sectors.

In this work, we proposed and implemented a technology-neutral solution that can be built to provide the bridge between the local repositories and the federation. Our proposed approach capitalizes on the current Web architecture to allow repositories to publish their assets metadata in ADMS independently from any consuming application. By exploiting the ADMS Exchange API, the federation harvests ADMS data from distributed repositories and keep its triple store up to date. It is worth mentioning, that our approach has been submitted to the ADMS working group and is currently being considered to become the defacto approach for publishing ADMS data to the federation.

Creating mapping between ADMS and the repositories metamodels is a great challenge per se. In this vein, we plan to automate it, e.g., through a mapping tool based on Google refine[25], thus making it easier and more efficient for repositories to publish ADMS data and then use the ADMS API to transfer them to the federation. Moreover, the federation should be able to harvest metadata from RDFa sources, because the increasing adoption of RDFa will probably encourage many repositories to support RDFa representations of their semantic assets. Additionally, once ADMS is adopted by a significant number of repositories, a scalable automatic crawling of all ADMS sources and automatic synchronization by the federation are needed to eliminate the manual registration and update requests.

[24] http://www.w3.org/Submission/2012/SUBM-ldbp-20120326/#bpr-HTTP_GET
[25] http://code.google.com/p/google-refine/

Acknowledgements

We'd like to thank Richard Cyganiak for the valuable feedback and disscussion on the API design. This work was funded in part by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2) and in part by Granatum EU FP7 ICT under Grant No. 270139.

References

1. European Interoperability Framework for pan-European e-government services. Technical report, Brussels, 2004.
2. Harnessing ICT to promote smart, sustainable & innovative Government . *The European eGovernment Action Plan 2011-2015*, Dec. 2010.
3. Asset Description Metadata Schema Specification - Version 1.0 . `https://joinup.ec.europa.eu/asset/adms/release/100`, Mar. 2012.
4. CESAR Workshop Survey Results. `https://joinup.ec.europa.eu/sites/default/files/CESAR_Feedback_Survey_Results_Report_1.pdf`, Mar. 2012.
5. K. Baclawski and T. Schneider. The open ontology repository initiative: Requirements and research challenges. In *Proceedings of Workshop on Collaborative Construction,Management and Linking of Structured Knowledge*, ISWC 2009, Oct. 2009.
6. P. A. Bernstein and L. M. Haas. Information integration in the enterprise. `http://classes.soe.ucsc.edu/cmps277/Winter10/Papers/bernstein-haas-cacm08.pdf`, 2008.
7. R. Chinnici, M. Gudgin, J. Moreau, and S. Weerawarana. Web Services Description Language (WSDL) Version 1.2 Part 1: Core Language. Technical report, June 2003.
8. L. DeNardis. E-Governance Policies for Interoperability and Open Standards. *Policy & Internet*, Oct. 2010.
9. R. T. Fielding. *Architectural styles and the design of network-based software architectures*. PhD thesis, 2000.
10. P. Haase, T. Mathäß, and M. Ziller. An evaluation of approaches to federated query processing over linked data. In *Proceedings of the 6th International Conference on Semantic Systems*, I-SEMANTICS '10, New York, NY, USA, 2010. ACM.
11. R. Hartmann, J.and Palma, Y. Sure, M. C. Suárez-figueroa, P. Haase, A. Gómez-pérez, and R. Studer. Ontology metadata vocabulary and applications. In *International Conference on Ontologies, Databases and Applications of Semantics*. Springer, 2005.
12. A. L. Hors and M. Nally. Using read / write Linked Data for Application Integration : Towards a Linked Data Basic Profile. *Linked Data on the Web (LDOW2012)*, 2012.
13. N. Mitra and Y. Lafon. SOAP Version 1.2 Part 0: Primer (Second Edition), Recommendation REC-soap12-part0-20070427. Technical report, Apr. 2007.
14. V. Peristeras, N. Loutas, S. K. Goudos, and K. Tarabanis. A conceptual analysis of semantic conflicts in pan-European e-government services. *Journal of Information Science*, May 2008.
15. E. Wilde. Putting things to REST. Technical report, UC, Berkeley, Nov. 2007.