LDS^3: Applying Digital Preservation Principals to Linked Data Systems

David Tarrant and Les Carr
School of Electronics and Computer Science
University of Southampton
Southampton
UK
SO17 1BJ
davetaz,lac @ecs.soton.ac.uk

Data publishing using semantic web and linked data techniques enables the sharing of detailed information. Importantly this information is shared using common standards and vocabularies to enable simple re-use. In the digital preservation community, an increasing number of systems are adopting linked data techniques for sharing data, including the PRONOM and UDFR technical registries. In many systems, only current information is being shared. Further, this information is not being described with data relating to who and when it was published. Such basic metadata is seen as essential in all digital preservation systems, however has been overlooked to a large extent when publishing linked data. This failing is partly due to there being very few specifications, reference implementations and verification systems in place to aid with publishing this type of linked data. This publication introduces the Linked Data Simple Storage Specification, a solution that enables careful curation linked data by following a series of current best practise guidelines. Through construction of a reference implementation, this work introduces how historical information can be referenced and discovered in order to build customisable alerting services for risk management in preservation systems.

1. INTRODUCTION

Data, or to use another term, knowledge is the foundation for progression in society. Knowledge is key to making informed decisions that hopefully, on reflection, are correct. This principal is particularly true in the field of digital preservation and archiving where a key opportunity exists to automate the sharing of knowledge for the good of the entire community. The most common form of knowledge exchange within the digital preservation community is via registries ([7],[18],[8]). Moving on from simple fact based registries, such systems have evolved with the aim of sharing process information [1] to the point where it is now possible to share work-flows [10].

The automated sharing knowledge via the web is an area of research that has seen huge interest over the past decade, partly driven by the vision for a Semantic Web [4]. In this vision, knowledge comes together with reasoning such that informed decisions can be made on a persons behalf. This is a field of study which brings modern techniques together with years of Artificial Intelligence research [12].

The idea of publishing self describing data on the web, that could be read and understood by computers became the key driving principal for what is now known as Linked Data. Berners-Lee outlines a 5-star guide for publishing linked data on the web [3], a guide that has been followed successfully by many communities ([6],[13],[17]) including in the field of digital preservation [9].

The P2-Registry prototype [18] took advantage of the ability to harvest, manipulate and reason over linked data available from many sources to help make informed decisions regarding preservation actions. Data from PRONOM and DBpedia (the linked data version of wikipedia) was imported and aligned using a series of simple ontologies. This lead to huge increases in the amount of knowledge available to answer questions relating to specific digital preservation problems including: “What tools can open a particular file?”, and “How do I migrate this file to JP2000?”.

The original P2-Registry prototype has been utilised successfully by many preservation systems to help users make important decisions ([2],[19]). In addition many other linked-data related projects have began in the area of digital preservation, most notably the PRONOM data is now available directly from the National Archives (UK) as linked data [9].

While the amount of linked-data becoming available from various sources is becoming much greater, there still exists many problems in managing this data and deploying the correct architectures. Further challenges are then faced in understanding what information is available, establishing trust of this information and separating historical and current information.

While these problems exist within both the UK government data (where PRONOM is hosted) and P2-Registry system, they are not unique in these systems. In the years following the initial effort on the P2 system, many efforts have been made in the wider community to tackle the problems with understanding, trust and provenance resulting the in production of many best practise guidelines. In this publication, we present LDS^3, the successor to P2 that follows a number of these best practices to provide a simple system which automates and assists with the process of publishing data to maintain integrity, trust and full historical information. Further to this, the LDS^3 system also enforces strict

data curation policies, meaning any hosted datasets should be easy to understand, query and re-use.

LDS^3 supports a publication-based named graph model to re-connect data indexed for querying to the actual source data. Further LDS^3 removes the concern from the user about version and temporal data, much like version control systems do for computer code, enabling users to directly upload and manipulate documents containing the important data. The LDS^3 reference implementation extends a number of freely available and well supported software libraries. This is done with a lightweight shim that simplifies and streamlines the process of managing linked data. At the same time as implementing the LDS^3 specification, this shim also incorporates authentication services using OAuth2 to allow the management of data to be restricted.

This publication presents both the LDS^3 specification and related reference implementation. Further a number of exemplar use cases, similar to that presented in the P2-Registry work, are introduced to demonstrate the benefits of the new capabilities available. Specifically, one of these capabilities looks at how historical information can be queried to provide automated alerting services when expected behavioural change.

The remainder of this paper is structured as follows. Section 2 recaps the P2-Registry and related work from the wider community, introducing many of the efforts being made to produce best practice guidelines for managing trust, authenticity and history of data on the web. Sections 3 and 4 introduce the LDS^3 specification and reference implementations addressing how some of these best practise guidelines have been applied to produce a specification for managing data.

Section 5 looks at the problem with changing data in the digital preservation community. By continuing the P2-Registry work, this section looks at the risks to changing characterisation data and outlines how LDS^3 can be used to build alerting services to information about risks related to change in this type of data. Before concluding the broader implications for LDS^3 type systems are introduced demonstrating how LDS^3 supports discovery and querying of historical data.

This paper concludes by looking at the applications of LDS^3 and possible future work. This section looks at how the P2-Registry has now been enhanced with temporal data without changing the existing API and available services. LDS^3 provides an exemplar for publishing persistent datasets that provide valuable information needed to establish trust,. By extending the use of such services beyond the preservation community, this will in turn enable easier data preservation in the future.

2. LINKED DATA TODAY

Berners-Lee's original vision for the Semantic Web became a vision for the future of automated computing in which information is not only discoverable and transferable, but also fully understood. Further, this information enables the generation of new knowledge through complex reasoning and other inferencing techniques. Essentially the web and http would be used as the location, storage and transport methods for knowledge. Artificial Intelligence methods would be required to assist with trust, proof and the understanding of the data.

While the semantic web is still a vision, some of the barriers to seamless knowledge exchange are being lowered. Sharing of knowledge starts with the sharing of data; facts that can be used in other contexts. The web has encouraged the sharing of information, however this has typically been via the embedding of data in web pages (using HTML). The drawback of this technique is that HTML is designed as a human readable format and not one to be used for automated exchange of understandable data. In order to move to a web of machine readable, open data requires a new way to expose data.

The benefits of sharing data have been seen in many applications [5]. Many services have opened up their data using formats such as XML, JSON and simple CSV, following the 5-star principals of linked data [3]. Exposing data under an open licence in this way achieves between two and three stars. The 4th star calls for the data to be shared in the RDF format, using URIs for identifiers, such that data can be easily discovered over the web and then used in a standards compliant way. Once the data is exposed as 4-star Linked Data, techniques from the Semantic Web can be used to align datasets from disparate sources, leading to a greater breadth of knowledge being available. 5-star Linked Data is that which is already aligned and linked in some way to other available 4 and 5-star linked datasets.

The idea of the P2-Registry was to expose the benefit of creating 5-star linked data for the digital preservation community. This was achieved through the linking of the PRONOM data to that exposed by DBpedia (the data endpoint for wikipedia). At the time the PRONOM data was not exposed as Linked Data, thus translating the XML data into RDF with URIs was necessary. This was required in order to get to a point where semantic web techniques could be used to align and link to the data from DBpedia.

Figure 1 shows the use of the RDF Schema vocabluary to connect two PRONOM identifiers (two versions of the PDF file format) to the DBpedia identifier for Portable Document Format. As DBPedia does not contain entries for each version of PDF, these links state that each PRONOM identifier is a subClass of the file format. In the case where a direct mapping could be found, i.e. for software URIs, then the sameAs predicate can be utilised from the Web Ontology Language (OWL) ontology.

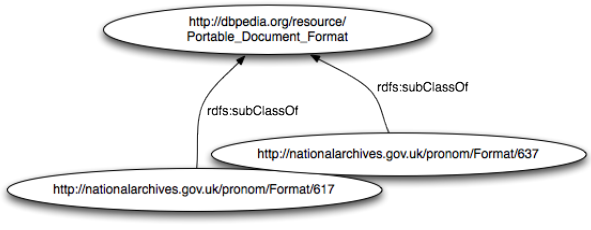

Figure 1: Associating PRONOM data with DBPedia data

The benefit of this simple link is easy to see when asking questions about the software tools available to read and write PDF files. With only the PRONOM data being used, the number of available tools was found to be 19. With the alignment to DBpedia (as shown in Figure 1), this number jumps to 70. Thus one connection (from PDF 1.4 to DBPedia) results in a near 4 fold increase in available data.

Since the P2-Registry work, the PRONOM data has been made available by The National Archives (UK) as 4-star linked data [9], with the 5th star (linking to other content) something of great interest. This work was enabled through the push in the UK for releasing of government data as 4 and 5-star linked data, something for which there is traction and now a substantial number of datasets available. International efforts have also been pushing to make raw data available in similar ways [8].

The publishing of linked data is just a single step towards fulfilling the promise of the semantic web. The problem is that the current methods for publishing and managing linked data fall short when looking at the full intention of the semantic web. Current publishing methods don't guarantee understanding, trust is not easy to establish and provenance information is also hard to find. Problems with establishing trust can be explained by analysing current publication and dissemination methods to discover that linked data is often only made available in a way disconnected from its source. When the source of the data is located, a process not made easy by current systems, it is still not clear how current and valid this data is, and what previous state the information held.

In the years following the initial effort on the P2 system, many efforts have been made in the community to tackle the problems with understanding, trust and provenance of linked data. This has the resulted in the production of many best practise guidelines that are discussed in this section.

2.1 Publishing Linked Data

Publishing of linked data starts with knowledge modelling, the process of taking existing data and deciding how to serialise this into a linked data format, typically RDF. Take the following axiom of information:

```
<David_Tarrant> worksFor <University_of_Southampton>
```

While this is a valid triple, on its own no clue is given about the validity of this information, something normally established by looking at the information source (e.g. this publication). Once discovered, questions like "how old is this information?" and "who published this information", can be answered easily. However in linked data (using RDF or SPARQL), it is not clear how to find the source of such information.

This was realised as problem by early linked data systems, examples of which include triple-stores. Such systems would store a fourth piece of information detailing the location from which the information originated so it could be easily updated. While systems designed to index and store linked data realised this need, it is still not fully realised by systems that expose this data, as was the case in the P2-Registry.

Many active linked-data systems utilise storage and indexing systems as their only dissemination mechanism, often with an accompanying SPARQL (RDF Query Language) endpoint. While this allows the data to be re-sliced to answer queries, this results is a disconnection between the exposed data and the original sources. In the P2-Registry, answers to queries consisted of data from two data sources (PRONOM and DBpedia), resulting in this same disconnection problem.

Moving from a triple based RDF model to that of a quad, means that named graphs (term for the quad), can be used to provide source information. Named Graphs can be used in two ways, either to express publication information or for representation information [17]. Using named graphs to express publication information allows the connection back to the original source (here termed as publication). Representation information relates more directly to the result of combining data, e.g. the source of a query and data about the query endpoint. There is value in both uses, especially as it may be required to keep a record of where the data was discovered (or queried from) as well as the locations for the original sources of that data.

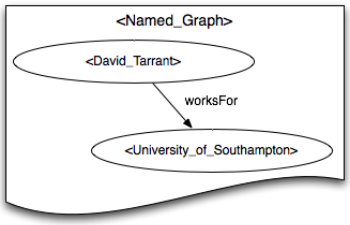

Figure 2: Encoding a triple with a named graph (a quad)

Figure 2 shows and example of the previous triple now represented with a quad. In the case of this representation it has been chosen to represent the named graph as a document that represents the source of the triple. Equally this document might convey information relating to many subjects (in this case people) and their related information.

Taking this forward, Figure 2 also indicates that the `<Named_Graph>` can also be the subject of information, thus allowing triples to be included in this named graph that describe itself. It is this data that can including facts like the author, publisher and publication time.

Exposing the named graph in queries immediately allows separation of data sources, allowing data from PRONOM to be differentiated from that produced via wikipedia. Knowing the exact source of the data allows any user to retrieve the original data from it's source (rather than the query endpoint) in order to verify the information and establish some level of trust. Additionally, techniques such as Public Key Identifiers (PKI) can also be used at this point to further verify that the data received is authentic [14].

Using named graphs for publication data clearly has its benefits, but requires that a user be able to retrieve the original data for inspection, not via an index of the data. While

sources for information, e.g. RDF, can be easily hosted on web servers directly, this process relies on the user to keep these documents up to date and properly annotating them with information relating to the time and place of publication. As well as combining indexing and query services, the main role of LDS^3 (as a Simple Storage Service) is to provide hosting services for the source of data. LDS^3 enforces the use of named graphs to represent publication data and will automatically annotate data that it is hosting with the publisher and publication time data, meaning that the author does not have to worry about these aspects. Providing both storage and indexing services means that LDS^3 is able to easily keep the two services in synchronisation while allowing users easy access to the source documents that were used to build the index.

2.2 Versioning Linked Data

The publication of linked data is typically a two-stage process involving the initial creation and subsequent importing of data into a linked data endpoint. It is this endpoint that provides fast access to the latest version of information using direct export or query functionality [17]. Further, such endpoints all include functionality for managing data indexes and ability to apply simplistic semantic reasoning. These systems were the early adopters of named graphs, using this information to allow data to be updated and overwritten, allowing the index to only return the most up to date (and thus valid) results. This is perfectly acceptable as the majority of queries are asking for current data. With many systems regarding the data endpoint as the only way to access data, finding previous information can be a significant challenge.

The problem with versioning resources is not necessarily applicable to all resources, for example statistical data intrinsically relies on temporal and contextual data to justify its own results. On the semantic web, such data would be referred to as an information resource. On the other hand data about a University, or Person, is an example of a non-information resource, where the main requirement is to discover current information. [17] (also discussed on Jeni Tennison's blog[1]) examines the problem with versioning information and non-information resources. One of the main conclusions is that it should be possible (not necessarily easy) to discover the previous state of non-information resources.

One technique for versioning linked data relating to non-information resources is to use publication named graphs these. Tennison recommends combining named graphs with cool URIs [16], making it very easy to see that versioning is being used. Further these URIs can be used to relate versions together, as it demonstrated in the example below:

```
<http://data.ac.uk/doc/{resource}/{version-2}>
   dct:replaces <http://data.ac.uk/doc/{resource}/{version-1}>
   dct:published ''2012-05-09 14:00:00+01:00''
   dct:author <http://id.ecs.soton.ac.uk/person/9455>

 <http://id.southampton.ac.uk>
   foaf:Name ''University of Southampton''
```

Here the resource name and versioning scheme can be be freely defined by the publisher, such that schemes such as simple version numbers can be used, or perhaps the date of publication is embedded in the named graph URI. Importantly, by using already available technologies, it is possible to navigate easily between versions of a named graph that (potentially) contain information relating to an Information Resource published by the same author, akin to editions of a book.

By separating storage from indexing, LDS^3 automatically creates and manages versions of named graphs submitted by authors. This way all previous versions of a named graph, containing all original data are available from storage, with the latest version available directly from the index. LDS^3 adopts a combination of Globally Unique Identifiers (GUID) and date stamps to generate the named graph URIs and versions of this URIs respectively. This also allows a user to ask for a GUID (without a date) and be re-directed automatically to the latest version.

In the field of digital preservation, people have for many years been talking about registries as the source for information. However these registries contain the same flaws due to the lack of temporal and provenance information. Historically (before digital), a register is a book in which records are kept, thus the authoritative source of information may well be a page in this book and cited in the same way as traditional journals. Each register would have its own version information and publication date. As registers have become digital, it has become very easy to duplicate and move data around and simply overwrite old data, loosing the versioning and authoritative information related to the original publisher. In part this is due to the lack of clarity on what is the source of data, and what is simply a representation built from some index (or registry). Using named graphs effectively re-introduces versioned registries, where a much greater level of granularity is possible.

3. THE LDS^3 SPECIFICATION

The Linked Data Simple Storage Specification[2] outlines a mechanism for assisted publication of linked data. By taking influences from many existing systems, LDS^3 and accompanying reference implementation enables the management and exposure of large scale datasets. The LDS^3 specification utilises the named graph as a publication reference and requires any compliant server to automatically augment incoming data with further information relating to both the time and author responsible for the publication. All requests to publish data must be authenticated in a secure manner before data is augmented and URIs returned to the requestor.

The LDS^3 specification takes many influences from existing specifications, most notably the AtomPub[11] and SWORD2 [3] specifications. These existing specifications focus on the publishing of web and scholarly resources respectively. LDS^3 compliments these specifications while focussing on data publication and providing services to help with the curation and automated tracking of versions.

[1]Versioning (UK Government) Linked Data - http://www.jenitennison.com/blog/node/141

[2]LDS^3 Specification - http://www.lds3.org/Specification

[3]SWORD2 (Simple Web-service Offering Repository Deposit) Specification - http://swordapp.org/sword-v2/sword-v2-specifications/

The most important influence from both the AtomPub and SWORD2 specifications is the reference to CRUD (Create, Retrieve, Update and Delete) for managing resources. Each create request will also be processed to generate specific objects (and related URIs) within the LDS[3] system.

The process of creating a resource is shown by Figure 3 where data is HTTP POST'ed to the servers Data Submission endpoint. The server handles the request in the standard HTTP based based way and simply returned the location of the created resource. Further to this location, the server also returns the *edit-iri* that can be used to update and delete the document.

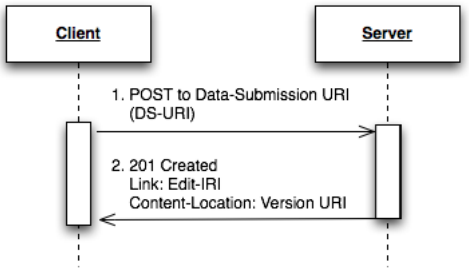

Figure 3: Submitting a new named graph to LDS[3]

For authentication, LDS[3] requires that all requests be signed using the same technique as employed by Amazon's Simple Storage Service (S3)[4]. This key based authentication mechanism works by users signing parts of the HTTP request with their private key. With only the request part of the transaction being signed, the process of authentication does not require bi-directional communication, meaning no loss in performance.

3.1 Managing resources with LDS[3]

An implementation of LDS[3] is intended to be deployed directly on the web server hosting the data URIs (e.g. starting `id.data.ac.uk`). This way the LDS[3] implementation can directly serve requests for information and non-information resources as well as the named graphs. Information relating to resources is likely to be sourced from many documents, thus requests for a resource will be handled via the index of the latest data. Named graphs, both current and previous versions can be provided directly from disk, avoiding the need for a data index.

While specifications for handling data indexes are well defined, LDS[3] compliments existing systems by also handling the publication of the named graphs, annotating these and storing them for indexing and provisioning to other systems and users. The LDS[3] specification dictates that resources (e.g. People, Universities or File Formats) cannot be directly created, updated or deleted. Each resource has to be described in a published document (named graph). This paradigm is similar to that of traditional publishing, where the trust of information is to some degree established by looking at the Book, Journal or Proceedings in which the data was published. By limiting users to only being able to publish and update documents, the LDS[3] enforces a model of versioning and provenance on resources. These graphs are thus being used as the publication mechanism rather than as presenting representational information.

Figure 2 shows how one document can be used to describe a resource. Here the LDS[3] endpoint is hosting data at `http://data.opf.org/`, with non-information resources having a prefix of `http://data.opf.org/id/`. Note that in Figure 2 the named graph URI is an example URI. When an LDS[3] server receives a correctly formatted an authenticated request, a unique URI must be created for the document. This URI should consist of two parts, one to identify the document series (the aforementioned *edit-iri*), the other for the version of the document. It is recommended to use a GUID for the *edit-iri* and append a version or date to this URI as the location of this particular version of the document.

Taking Figure 2 from before, the server then fills in (or changes) the document URI to the new URI and annotates it with data pertaining to who published the document, when and which (if any) documents it replaces. This results in a new document being generated similar to that shown by Figure 4.

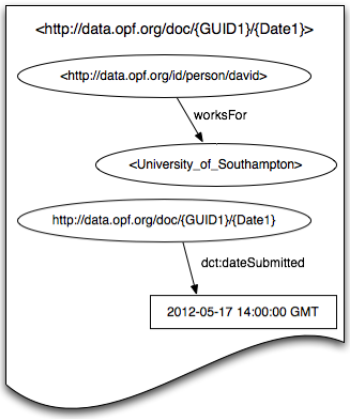

Figure 4: Document including LDS[3] annotation

Figure 4 shows that the named graphs are stored under their own document prefix `http://data.opf.org/doc/` with GUID and Data used as the suffix's (not shown here in order to save space). As well as the submitted data from Figure 2, the LDS[3] system has annotated the graph to make it self describing, adding the date when the graph was submitted. It is this exact same mechanism that is used to add further annotations and links to previous versions of the document.

Once the data has been annotated and stored, the *edit-iri* (or GUID only URI) is returned to the user along with a final representation of the annotated document. The final representation of the document is accompanied with the HTTP Content-Location header which defines the exact location on the server of the document, in this case the full document URI including GUID and Data suffixes. The *edit-iri* is communicated to the user using an HTTP Link header

[4]Signing and Authenticating REST Requests - http://docs.amazonwebservices.com/AmazonS3/latest/dev/RESTAuthentication.html

(as shown in Figure 3), it is this URI that can be used to submit new versions of the document as well as retrieve the latest version.

Using named graphs in this way allows many users to submit data relating to the same URI whilst retaining the separation of who submitted what information. Correspondingly, as users can only manipulate documents, they are only able to delete the data that they added, and not directly manipulate the resource URI. It is a combination of these factors that mean LDS[3] is able to provide enough information to enable the establishment of trust in the data. Allowing users to annotate their own named graphs and by enforcing versioning, allows the easy discovery of provenance information.

4. REFERENCE IMPLEMENTATION

In order to aid the deployment of LDS[3], a reference implementation has been developed. Rather than start from fresh the reference implementation ties together many existing libraries. The only new piece of development involved the creating of a shim to handle authentication, requested operations and document annotation.

The authentication module requires that users register in order to obtain a key-pair. It is expected that this key-pair be used by the users client in order to upload a series of documents, much like handling of objects in Amazon S3. Each key-pair remains linked to a single user account, but each user can have several key-pairs. To avoid building a user management system, the LDS[3] reference implementation contains an OAuth2 [15] module, allowing any OAuth2 compatible authentication service to be used.

Once a user has a key-pair, documents can be submitted to the Data Submission IRI (DS-IRI). Each received request is verified before a new GUID is generated and added to the document. Annotation is performed using this Graphite library[5] before storing the resultant document on disk and calling the index process to update the query endpoint. To index and allow querying of the data, the reference implementation recommends use of a quad store (such as 4store). Currently the LDS[3] reference implementation only indexes the latest data, handling old versions of is discussed in section 6.

With the index in place and data injested, the major requirement is to expose the datasets and make available a version of the Linked Data API[6] to make the data usable. In order to achieve this, the Puelia-PHP library has been chosen and themed with the data.gov.uk style. data.gov.uk utilises the exact same set of libraries as LDS[3], thus streamlining the functionality and mechanisms for publishing datasets, something also handled using Puelia-PHP.

Puelia-PHP is an application that handles incoming requests by reading a dataset configuration file to discover how to serve the request. Each dataset configuration outlines the URI pattern to match and how to query for the data from a SPARQL endpoint. The advantage with this type of deployment is that Puelia-PHP can gather different datasets from many SPARQL endpoints, spreading the hardware and processing requirements for hosting billions of items of data. Further the data is then cached to enable fast delivery for future requests. Finally, Puelia-PHP provides multiple serialisations of the data including JSON, XML, CSV alongside HTML and RDF.

As Puelia-PHP is designed to query data from a SPARQL endpoint and then serialise this into a new representation, the ability to retrieve the original named graph is not available. To counteract this, the LDS[3] reference implementation recommends that Puelia-PHP be patched to enable retrieval of named graphs from either the precise document URL (.rdf), dated URI or related edit-IRI (both content negotiated). Since resource URIs cannot be directly edited, the use of a representational named graph here is ideal.

Although Puelia-PHP does provide an excellent and well supported implementation of the linked-data API, it currently lacks the ability to expose named graph information. This is due to the challenges in exposing non-native named graphs that are linked to non-information resources. The linked data API specifies that systems should be able to query many indexes to location information from many sources and aggregate this into a new named graph (a representation named graph). Options exist to simply use this new named graph to point to all the existing named graphs, resulting in a meta-aggregation that doesn't directly describe the object the user asked about. Further you can envisage infinite meta-aggregations, making the process of retrieving any piece of information a painful one.

```
SELECT * WHERE {
Graph ?graph { ?subject ?predicate ?object }
}
```

While SPARQL supports the direct retrieval of named graph information (as showsn by the query above) it is the serialising of this information, into formats including RDF, that is the challenge. Not being able to serialise the data back to RDF doesn't mean that it cannot be used however and many other visualisation tools, including DISCO[7] and MARBLES[8] enable the browsing of quad based information. It is hoped that in the near future that this level of browsing capability can be bought to Puelia-PHP.

5. LEARNING FROM THE PAST

Historical records consist of two important pieces of information: facts about the environment at the time and decision data about choices made based upon interpretation of these facts. Example facts might include file format identification information (at the time), while process information outlines the actions, or provenance data, related to how these facts was used. It is the facts that inform the process, neither piece of data is useful without the other. Another way to look at facts, is to refer to them as non-information resources, while your processes are examples of information resources. Non-information resources (facts) can change over time, so only keeping the latest information means that the

[5]Graphite - http://graphite.ecs.soton.ac.uk/

[6]The Linked Data API -http://code.google.com/p/linked-data-api/

[7]Disco Hyperdata Browser - http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/

[8]Marbles - http://marbles.sourceforge.net

information resources (processes) become a lot less useful.

A good example of non-information resources in the field of digital preservation is identification data. Many file format identification tools exist, each under continuous development as new formats and format types become available. Due to the dynamic nature of file formats, there is a high risk of miss-identification. This is particularly true with formats which re-use the zip and xml standards for packaging. Additionally there is a chance that older formats may get re-classified if a newer format is very similar. These are all high preservation risks, and ones that require services to inform people of change.

As part of the European project looking at Scalable Preservation Environments (SCAPE), an LDS^3 implementation is being set up to store results of running a number of identification tools over a wide ranging corpora of exemplar data. By collecting this data over time, it will be possible to observe the changing behaviour of the tools and any potential risks to the identification process each version of a tool might introduce. For example, a number of the DROID signature files wrongly identify the Microsoft Word docx format, while other miss-identify PDF. Such information is currently only available to those running their own experiments, or via a few forums and mailing lists. There is currently no method for auto discovery of this information. By using LDS^3 to store data relating to these experiments, it is possible to discover these risks and report on them automatically using Preservation Watch services (also being developed within the SCAPE project).

By gathering results from experiments, data from many sources, and combining this with temporal data. LDS^3 has the capability to enhance the previous risk analysis work by being able to present evidence relating to how results have changed over time. Figure 5 shows the components of the preservation watch service for characterisation change, with an LDS^3 system collecting the results ready for analysis and publication.

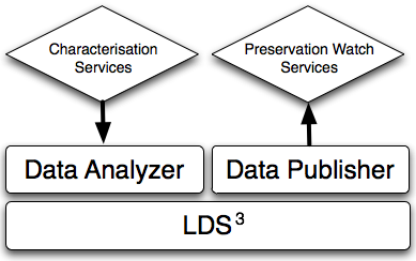

Figure 5: LDS^3 and preservation watch services

This system involves many of the components being developed as part of the SCAPE project being developed by many different parties. LDS^3 plays an important role in being a persistent store for published and usable datasets. By using widely available standards and technologies means that the many different parts of the system can be worked on independently to produce a usable solution for preservation practitioners.

When complete it is envisaged that the preservation watch services will produce a series of customisable alerts tailored for each individual user. If the users interest is in preserving multimedia content, then received alerts can be customised to only be relevant to this type of material. Most importantly though, each user will have the ability to trace the complete provenance of each alert, including the decision process and the facts that informed this alert. Further this can be done at any point in time, thus decisions made today, can be analysed again in the future without loss of information.

6. THE DATA TIME-MACHINE

The real appeal from this provenance information comes from what can be done with it, firstly and most obviously the clock can easily be turned back to discover the previous state of any named graph. As demonstrated by Figure 6, this can then be combined with the user interface to create a clear view of the data against time.

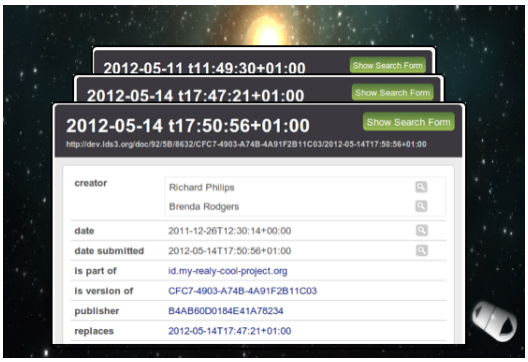

Figure 6: "Time-Machine" interface for LDS^3

This "Time Machine" style interface for linked data, shown by figure 6 working with LDS^3, allows the retrieval of any named graph from any point in time. This can also be achieved by using the Memento API [20] directly on the LDS^3 server. The request below shows an example request for a document (e.g. that from Figure 4) at a specific point in time. Note that the only extension to the normal HTTP request is the addition of the "Accept-Datetime" header as defined by the Memento specification [21].

```
GET /doc/GUID1 HTTP/1.1
Date: Mon, 14 May 2012 15:55:15 GMT
Accept: application/rdf
\textbf{Accept-Datetime: Thu, 21 Jan 2012 04:00:00 GMT}
Host: data.example.org
```

In addition to providing access to static documents from the past, by maintaining a few indexes of named graphs and their relation to resources it is possible to rebuild an index as it looked at any point in time. This allows full SPARQL queries to be executed on the data as it existed at this point. This capability represents a breakthrough for retrieving the previous state of a resource. All current web archives are very static in nature, showing content conforming to how the harvesting service retrieved it. Being able to completely re-query the index as it looked at a specific time is a major improvement on this technology.

7. CONCLUSION AND FUTURE WORK

Exposing linked data specifically about digital preservation has already been shown to have large benefits for the community. The P2-Registry work demonstrated how a simple relation between two existing datasets results in a four fold increase in results for a query. However these results are always considered current and come without any provenance information relating to the origin of each individual result. A four fold increase in data is only possible if many people can describe the same (or similar) objects on the web and without provenance information it is impossible to establish trust in such distributed data. Additionally, if decision processes are made based upon this data, without access to historical information, it is challenging to review such decisions again in the future.

By focussing on identification information, this publication presented a scenario in which the historical nature of identification information is not known. A major problem if a file format is wrongly identified. Such a change could cause serious consequences if process information is affected and called into question.

In order to address the challenges of provenance, versioning and trust, this publication introduced the Linked Data Simple Storage Specification (LDS^3) and related reference implementation. LDS^3 enforces the use of named graphs for publication of data related resources on the web, e.g. file format data. It is these named graphs that can be directly annotated with additional data including author, publisher and date of publication. Further, by using a combination of Globally Unique Identifiers (GUIDs) and time stamps in the URI scheme, LDS^3 provides automatic versioning of data.

LDS^3 provides an HTTP CRUD based interface enabling the secure management of fully annotated and versioned linked data. The LDS^3 reference implementation, written as a shim, uses many existing and well supported libraries to perform data management, annotation and indexing. One such library, Puelia-PHP (used by the UK Government open data project), is used as the primary user interface with a quad-store backing the SPARQL endpoint.

As well as an LDS^3 endpoint being created to store results of identification experiments, enabling the provisioning of preservation watch services, the existing P2-Registry system will be upgraded. This will enable sources of data to be discovered, allowing users to separate wikipedia data from that delivered by PRONOM. As the P2-Registry system was also based on the linked data principals, the user facing functionality and API does not change, it simply gets upgraded with new functionality designed to enable the establishing of trust and validity of data.

Having ingested fully annotated and versioned data. The LDS^3 reference implementation applies parts of the Memento protocol to enable resources to be retrieved as they existed at specific points in time. Additionally it was demonstrated how these can be displayed in a "Time Machine" like user interface. Future work will investigate the possibility of allowing SPARQL queries to be performed over whole datasets, enabling fully dynamic query of semantically annotated datasets at any point in their history.

8. REFERENCES

[1] B. Aitken, P. Helwig, et al. The planets testbed: Science for digital preservation. *Code4Lib Journal*, 2008.

[2] C. Becker, H. Kulovits, et al. Plato: a service oriented decision support system for preservation planning. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, 2008.

[3] T. Berners-Lee. Linked data. *w3c Design Issues*, 2006.

[4] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 2001.

[5] C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2009.

[6] C. Bizer, J. Lehmann, et al. Dbpedia-a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2009.

[7] A. Brown. Automating preservation: New developments in the pronom service. *RLG DigiNews*, 2005.

[8] U. Center. Unified digital format registry (udfr). 2012.

[9] R. Fisher. Linked data pronom. *National Archives Labs*, 2011.

[10] C. Goble and D. De Roure. myexperiment: Social networking for workflow-using e-scientists. In *Proceedings of the 2nd workshop on Workflows in support of large-scale science*, 2007.

[11] P. Hoffman and T. Bray. Atom publishing format and protocol (atompub). *IETF, RFC 5023*, 2006.

[12] I. Horrocks, P. Patel-Schneider, and F. Van Harmelen. From shiq and rdf to owl: The making of a web ontology language. *Web semantics: science, services and agents on the World Wide Web*, 2003.

[13] G. Kobilarov, T. Scott, et al. Media meets semantic web: How the bbc uses dbpedia and linked data to make connections. *The Semantic Web: Research and Applications*, 2009.

[14] E. Rajabi, M. Kahani, et al. Trustworthiness of linked data using pki. In *World Wide Web Conference (www2012)*, 2012.

[15] D. Recordon and D. Hardt. The oauth 2.0 authorization framework. *IETF*, 2011.

[16] L. Sauermann, R. Cyganiak, and M. Völkel. Cool uris for the semantic web. *W3C Interest Group Note, 3rd Decemeber 2008*.

[17] J. Sheridan and J. Tennison. Linking uk government data. *Statistics*, 2010.

[18] D. Tarrant, S. Hitchcock, and L. Carr. Where the semantic web and web 2.0 meet format risk management: P2 registry. *International Journal of Digital Curation*, 2011.

[19] D. Tarrant, S. Hitchcock, et al. Connecting preservation planning and plato with digital repository interfaces. In *7th International Conference on Preservation of Digital Objects (iPRES2010)*, 2010.

[20] H. Van de Sompel, M. Nelson, et al. Memento: Time travel for the web. 2009.

[21] H. Van de Sompel, M. Nelson, and R. Sanderson. Http framework for time-based access to resource states: Memento. *Internet Engineering Task Force*, 2010.