Using Trackback to Support Citation Notification Services

Brian Matthews, Katherine Portwin, Catherine Jones,

e-Science Centre, Science and Technology Facilities Council (STFC)

Bryan Lawrence

National Centre for Atmospheric Science (NCAS) British Atmospheric Data Centre

b.m.matthews@rl.ac.uk

Introduction

The notion of Linkbacks between websites has been developed to enable authors to keep track of who is linking to their pages. Authors receive notifications when other authors link to their pages, which is especially useful in the blogging community. One of the most well-known of several protocols which has been developed for Linkbacks is Trackback which has been developed within the blogging system, MoveableType.

A similar problem to tracking links between blog articles is found in repositories of research outputs, such as archives of academic publications or research data. Publications use citations to refer to one another, but repositories are typically not informed when there are external publications referring to their content. This is useful information in judging the value of a paper.

This paper considers the use of Trackback to support automatic notification to propagate citation information between repositories to provide a simple light-weight peer-to-peer service which is as widely applicable as possible and can be implemented quickly. The original Trackback protocol has been adapted to the

requirements of citation notification via a richer set of metadata fields which can be sent with the data using RDF, and a richer interaction between the peers.

This work was carried out within the *Citation, Location and Deposition in Discipline & Institutional Repositories (CLADDIER)* project[1] which has been investigating the issue of linking publications held in institutional repositories to the underlying data held in specialist repositories by developing the theme of citations, not only for publications but also for datasets. A summary of the project was presented at the UK e-Science All-Hands Meeting (see *[Matthews et. al. 07a],* especially the presentation), and the current paper summarises the data linking report *[Matthews et. al. 07b].* Further discussions on the topic are to be found on the CLADDIER site Wiki (http://claddier.badc.ac.uk/trac) , and on Bryan Lawrence's Blog (http://home.badc.rl.ac.uk/lawrence/blog)

Linking data and publications

Many academic institutions now maintain *Institutional Repositories* to record, archive and disseminate their research output, typically documents produced for formal publication. Further, there are a number of repositories which collect, store and distribute *data* arising from experiments or observations. Whilst such data could be archived in institutional repositories, they are often collected into subject based repositories to take advantage of discipline specific expertise and to maintain a close relationship with the relevant research community.

The existence of repositories allows the relationship between research publication and research data to be explored. From a publication it would be useful for readers to track back to primary data to inspect for themselves the quality of the data and the validity of the conclusions drawn, possibly running analyses of their own, and for authors to track forward to find further experimental studies influenced by the results which they reported. From a dataset, it would be useful to track back to discover the context the data was generated from the publications used to justify its collection, and to track forward to find the resulting

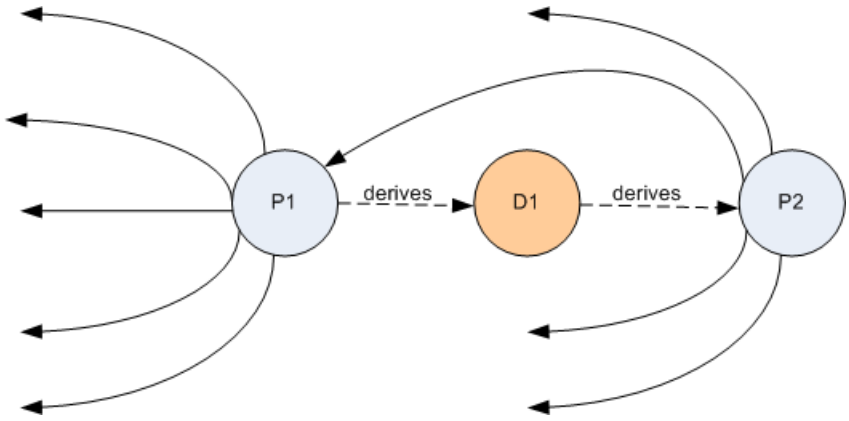
---

[1] http://CLADDIER.badc.ac.uk/ The partners are the University of Reading, the University of Southampton, the Science and Technology Facilities Council, and a data repository: the NCAS British Atmospheric Data Centre (BADC). The project was funded by the UK's Joint Information Systems Committee (JISC).

publications reporting analysis of the data, not only the primary analysis of the data generator, but also secondary analyses by other users of the dataset.

At first sight, this issue of cross-citation of data and publication seems straightforward. There are long established conventions for referencing publications from other publications – we can simply extend these to cover the case of citing datasets. Citation indexes and websites such as Citeseer (http://citeseer.ist.psu.edu/) track cross-citations. However, on closer consideration, a number of issues arise which need to be addressed.

Traditional publishing uses one directional "*backwards citation*" which is entered by the author, and typically such citations will only reference publications. This situation is illustrated in Figure 1. In this figure, P1 and P2 are papers, and D1 a dataset, and all un-labelled arrows represent backward citations. In theory, the chain of backward citations could be followed until the beginnings of formal academic publishing in the 17th century or beyond. Datasets are typically not cited even though the work in the paper (and thus the paper itself) derives from the generation and analysis of that data set, although it is frequently the case that the "inspiration" papers behind the data set are cited in the resulting paper.



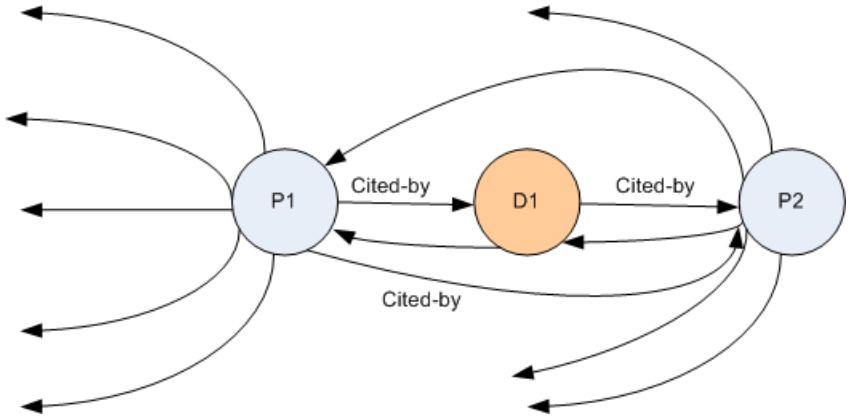
trad-citation.png

Figure 1: Traditional Publication Citation

Data archives wish to track who has been using data resources and thus want to keep track of "*forward citations*" ("cited-by" links) – they may be informed of a citation from a personal communication, or from a usage report for example. Once a data archive has recorded a paper as arising from a particular dataset,

then the backward citation from the paper to the data set can be added; this is not necessarily added by the author, but rather by the repository managers.

Thus we wish to move from the traditional situation as in Figure 1, to situation in Figure 2 where forward citation ("cited-by") links are added, as well as citations to and from datasets.



cross-citation.png

Figure 2: Data and Publication Cross-Citation

We identified three components which are required to support the cross-citation of data and publications were: a standard convention for citing datasets in repositories such as BADC; mechanisms in repositories for recording backward and forward citations; and a *Citation Notification Service* which notifies holders of the targets of citations that a source object is citing them. The Citation Notification is the subject of this paper.

Citation Notification

In order to complete the desired cross-citation graph as in Figure 2, we need to populate the repositories with cross-citations. In particular, we need to inform repositories that their entries have been cited (that is a backward citation in a source) so that they can add "cited-by" entries (that is the corresponding forward citation). We propose that this citation would be undertaken by **a Citation Notification Service**. A number of principles were adopted in the design of the citation notification service.

1. The notification should be on a Peer-2-Peer basis.

2. The service should be generic across different types of repositories and repository software.

3. The service should be generic across different digital object types (e.g. data and publications) and metadata formats.

4. The notification should exchange appropriate metadata on the citation from citation source holders to citation target holders.

5. The notification system should not determine what the target repository does with the notification of the citation.

6. The mechanism should fail gracefully in the event that a target does not exist or does not recognise the notification.

7. The mechanism should identify the sender of the notification and defend against bogus notifications of citations.

Further, it was seen as desirable if existing off-the-shelf tools and mechanisms could be adapted to build on existing established practice and save on development effort.

There a number of different potential architectures to support citation notification. We considered a number of these architectures, including: using a intermediate Broker (mediated push); Linkback (peer-to-peer push); a Publish/Subscribe protocol such as RSS (peer-to-peer pull); a central Harvester service such as supported by Crossref - http://www.crossref.org (mediated pull); and a hybrid Linkback with Registration (Broker/Linkback Hybrid)

Approaches using publish/subscribe were attractive as a powerful mechanism where by repositories can notify citations asynchronously and with little effort, and it has been noted that the JISC SWORD project is basing its Deposit API on ATOM (see http://www.ukoln.ac.uk/repositories/digirep/index/SWORD). However, the Linkback approach was chosen because:

- Linkback is a pure peer-to-peer approach which can be maintained by the repositories themselves without relying on third party services and a business models which sustain them (unlike broker, harvester or hybrid models).

- Linkbacks can be received from any other repository without the source or target repository knowing of each other's existence in advance (unlike

publish/subscribe model which requires the target to know of the likely sources of citations).

- Existing well-known and well-defined simple protocols for notification of article cross-referencing used in the Blogging community, including well-known problems (e.g. spamming), with simple and easy-to use software.

A number of Linkback specifications exist (for a summary see http://en.wikipedia.org/wiki/Linkback), including Trackback, Pingback and Refback. **Trackback** is a simple "framework for peer-to-peer communication" supported by blogging tools such as *MoveableType* (http://www.movabletype.org/). It has a relatively simple metadata transmission in its simplest form, but has a straightforward mechanism for extension of the metadata as it uses the POST mechanism. Problems with Spamming are well-known and mechanisms can be added to mitigate this problem. Consequently, Trackback was chosen as basis the experimental Citation Notification Service.

Introduction to Trackback

Trackback was first released as an open specification in August 2002. The original trackback specification can be found at: http://www.sixapart.com/pronet/docs/trackback_spec . Essentially, Trackback involves sending a "ping" request over HTTP POST messages, saying "*resource A has a link to (cites) resource B"*. We give a brief description of the original two stage Trackback protocol. Trackback uses a two stage protocol. In simple terms, once a Repository B discovers a citation URI in a source object B1 referring to a target object A1 in repository A, it:

1. Repository B accesses A1's catalogue entry via a conventional HTTP call.
2. Within the returned page Repository B looks for a "*TrackBack URL*". If it fails to find one the protocol stops.
3. If it finds a Trackback URL, it accesses that URL via a HTTP POST, delivering a URL to the resource B1, together with the title of the source object.
4. Repository A can then augment the metadata of A1 with the reference to B1, thus completing the cross-reference.

Thus the protocol involves two requests:

1. Sender makes a GET request for the URL for the linked-to resource. Receiver returns the HTML page including embedded Trackback URL.

2. Sender makes a POST request (sending a “ping”) to the Trackback URL.

This simple protocol is thus well-suited for the task of adding cross-references. The TrackBack mechanism is proven to be robust and simple to implement. The original Trackback specification defines a simple 'Ping' message which has the simple contents of a URL of the citing resource, the title of citing resource, and a short excerpt of citing resource.

Using Trackback for a Citation Notification Service

Summary of functionality

For the purposes of citation notification, we propose to extend the protocol to carry arbitrary metadata so that it can provide a P2P mechanism for inter-repository communication. A summary of functionality supported by the Trackback Citation Notification Service is as follows:

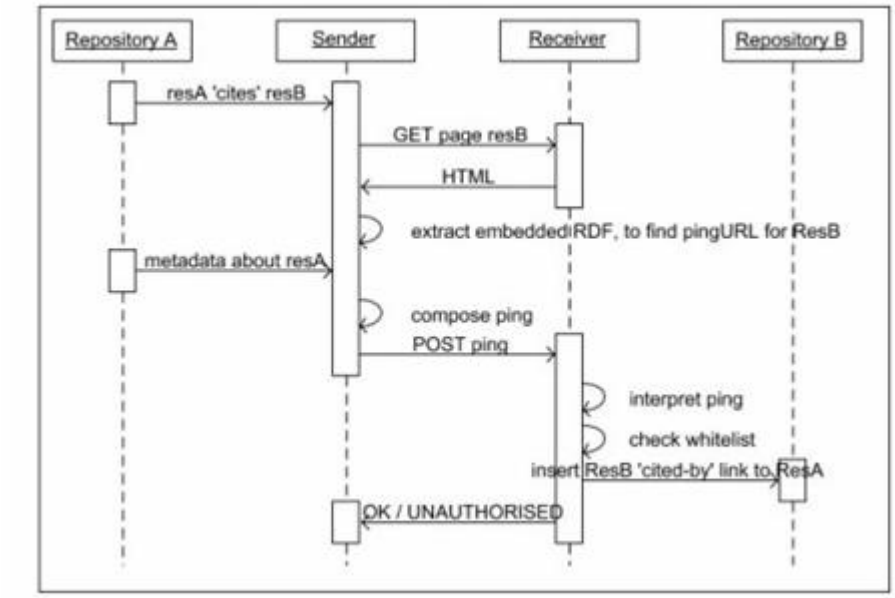
1. **Send** citation ‘ping’ messages, including metadata on the source of the citation, from a source repository to repositories which host resources cited by resources in the source repository. This is supported by a *Sender Service* which generates the Trackback pings at some point after the submission of the citation.

2. **Receive** citation ‘ping’ messages from other repositories, extract and read the citation metadata, and enter them in the target repository as cross-citations including any links. This is supported by a *Receiver Service*.

3. Operate a ‘**whitelist**’ system to reduce the possibility of bogus notifications (“spam”).

The sequence of request and responses including our extensions are given in trackback_sequence_diagram.jpg

Figure 3. Further extensions to the functionality and protocol were also specified, though not necessarily implemented.



trackback_sequence_diagram.jpg

Figure 3: Extended Trackback Sequence Diagram

The major extension required to use Trackback as a citation notification service is to greatly extend the metadata sent across the protocol within the HTTP POST request. This is a conservative extension; the citation metadata is appended to the end of the POST message, and a non-citation aware Trackback system would ignore this extra content.

Thus, the sender service first accesses the provided URL for the catalogue entry for the target of the citation. The receiving repository embeds RDF metadata in the catalogue resource page, which includes a unique `trackback:ping` URL for the given resource. This follows the Trackback specification, although we extend this by adding additional Dublin Core fields for `dc:indentifier` (giving the URI of the target object) and `dc:title` giving the title of the resource. These are used for logging the transaction.

An exemplar of the RDF metadata which is embedded in the target object is given below. Note the Trackback namespace, and the embedded Trackback Ping URL.

```
<rdf:RDF
```

```
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:trackback=
  ”http://madskills.com/public/xml/rss/module/trackback/” >
      <rdf:Description
          rdf:about="http://www.foo.com/archive.html#foo"
          dc:identifier="http://www.foo.com/archive.html#foo"
          dc:title="Foo Bar"
          trackback:ping="http://www.foo.com/tb.cgi/5" />
  </rdf:RDF>
```

Once the Trackback URL has been identified, the sender service composes a HTTP POST message following the format containing a number of key-value pairs with standard names for the keys as given as follows.

```
HTTP POST to http://trackbackurl-for-resourceB
  &title=[title of source object]
  &url=[URL of source object]
  &excerpt=[excerpt from source object]
  &blog_name=[website of source repository]
  &metadata=[rich metadata about source object]
  &metadataformat=[format of metadata]
  &type=[type of cross-reference being notified]
```

The first four keys in the figure are fields used in a standard Trackback call. We defined several extensions. In addition to the key-value pairs above, pings contain three additional keys: `metadata`, `metadataformat`, and `type`.

The `metadata` key gives the metadata of the source object as a text block. This could be encoded in the Dublin Core Guidelines for Bibliographic Citation *[Apps 2005]*, which is used as a default metadata format, or any other format as supplied by the source repository. The `metadataformat` key specifies the format of the metadata supplied. The `type` key is a field added to support communicating the nature of the cross-reference being notified.

For example, the metadata key can contain the metadata in Dublin Core Bibliographic Citation format as below.

```
<rdf:RDF
   xmlns:dcterms="http://purl.org/dc/terms/"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:dc="http://purl.org/dc/elements/1.1/" >
 <rdf:Description rdf:about="http://theidentifier">
   <dc:title>Best article ever</dc:title>
   <dc:creator>F Flinstone</dc:creator>
   <dc:creator>J Bloggs</dc:creator>
   <dcterms:issued>2007</dcterms:issued>
   <dcterms:isPartOf>urn:ISSN:1234-5678</dcterms:isPartOf>
   <dcterms:bibliographicCitation>&amp;ctx_ver=Z39.88-2004
         &amp;rft_val_fmt=info:ofi/fmt:kev:mtx:journal
         &amp;rft.jtitle=Lecture Notes in Computer Science
         &amp;rft.volume=1
         &amp;rft.issue=11
         &amp;rft.spage=111
    </dcterms:bibliographicCitation>
  </rdf:Description>
</rdf:RDF>
```

This metadata can then be processed by the target repository; typically, the target repository will enter a citation to the source object in the catalogue entry of the target object, thus completing the cross-citation.

Using Whitelists

A well known problem with the Trackback protocol is that of Spam; if the ping is accepted unconditionally, then the target repository can receive bogus messages and enter them in the catalogue entry of the target object. This may not arise from a legitimate citation within another repository, but from any other sender implementing the protocol. This may include incidental or accidental links (e.g. mentions of papers in Blog entries) or malicious links (e.g. sending advertising and links to unrelated web pages to be displayed in catalogue entries).

To avoid this problem, a mechanism is needed to identify and accept Trackbacks from legitimate sources of citations. The approach which was undertaken was the use of *Whitelists*. Upon receipt of a ping, repositories check the IP address of the sending repository against a "whitelist" of trusted partners, a step which is added in trackback_sequence_diagram.jpg

Figure 3. Whitelists are maintained in a simple RDF file with information about known repositories names and IP addresses; an example whitelist file in this format is given below. If the sending host is not on the whitelist, the receiving host returns a 403 unauthorised error.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:wl ="http://epubs.cclrc.ac.uk/vocab/trackback/">
    <wl:repository rdf:about="http://testserver.claddier">
        <wl:hostname>Claddier Testserver</wl:hostname>
        <wl:ipaddress>130.246.142.155</wl:ipaddress>
    </wl:repository>
    <wl:repository rdf:about="http://test.home">
        <wl:hostname>Test Home</wl:hostname>
        <wl:ipaddress>130.246.242.92</wl:ipaddress>
    </wl:repository>
</rdf:RDF>
```

This method helps to prevent both malicious and inadvertent spam, without the use of a central registry. Repositories can exchange information about their IP addresses outside the protocol and thus maintain a level of confidence in the notifications that they are receiving. However, this is an area where further consideration is needed as IP addresses may not be the most appropriate mechanism for checking the identity of repositories. The whitelist we propose could be extended to accommodate other means of identifying legitimate sources, such as domain name, repository contact, or public encryption key.

Further Extensions to Trackback

Once the P2P communication between repositories is established, it can be exploited further to provide additional functionality. Thus a number of further extensions to Trackback were identified and partially implemented.

Reverse Trackback

In the conventional Trackback protocol, the sender sends metadata to the receiver, as a ping (Figure 3), and there is no exchange in the other direction – the metadata which the sender holds about the citation must be entered by the user, or in some other way.

However, the protocol can be augmented by using it to send metadata about the target (receiver) to the source (sender) – a *Reverse Trackback*. This extension involves enhancing the embedded RDF to enable the sender to gather data as a side effect of sending a ping. This metadata can then be use to correct or enhance the metadata held about the target within the citation.

To implement this extension, two additions to the protocol are required:

- Overload the RDF snippet embedded in the receiver page, to include metadata about the receiver resource.
- Sender reads the full RDF snippet, and uses the metadata to enhance its own (sparse) record.

This extension has a very positive result on the user experience; a user entering source resource A which includes a citation to target B formerly needed to give metadata about the citation – at minimum, a title for display on the link, and ideally several other fields. With this extension the user needs only to enter the URL for B and the Trackback system can retrieve the rest of the metadata and enter it accurately and more completely (assuming that the metadata held in the target repository has a more complete record of the target resource).

Multiple Metadata Formats

As discussed above, the `&metadataformat` key of the extended Trackback protocol allows the sender of the ping to specify the format of the metadata being transmitted. Further, this extension allows the metadata in a ping to be sent in an alternative schema, with the ping *receiver* being able to specify a list of supported schemas. The default encoding for metadata sent in the ping is Dublin Core, based on "Guidelines for Encoding Bibliographic Citation Information in Dublin Core Metadata *[Apps 2005]*. The Scholarly Works (Eprints) Application Profile *[Allinson et. al. 2007]* was also selected to demonstrate the concept of multiple schemas. Three extensions to the protocol are required:

- Receiver specifies list of supported `<trackbackx:preferredSchemas>` in embedded RDF.
- Sender reads this list while visiting the target repository to collect Trackback URI, and chooses the best format based on the receiver's preferences and its own capabilities.
- Sender sends a ping using that format, and includes HTTP POST parameter of `metadataformat=[x]`.

The vocabulary used is indicated in the snippet RDF as follows:

```
<rdf:Description rdf:about="http://epublicns02"
           dc:type = "http://purl.org/dc/dcmitype/Service"
           dc:title = "Epubs trackback receiver service.">
 <trackbackx:preferredSchemas>
  <rdf:Seq>
   <rdf:li>
     <trackbackx:MetadataSchema
           rdf:about="http://purl.org/dc/elements/1.1/" />
   </rdf:li>
   <rdf:li>
     <trackbackx:MetadataSchema
           rdf:about="http://purl.org/eprint/epdcx/"/>
   </rdf:li>
```

```
    </rdf:Seq>
   </trackbackx:preferredSchemas>
  </rdf:Description>
```

Citation Type

The current implementation of Trackback Citation Notification assumes that the sender is notifying that the source object has a conventional, backward citation to the target object ("*cites*") and the receiver should respond by inserting a forward citation to the source into the target ("*cited-by*"). However, the protocol is neutral to the nature of the citation and can be used to notify targets of *forward* citations. Indeed, a use case is that of a data centre which is informed of a paper which uses the data, inserts the forward citation to the paper into its records, and wishes to notify the holder of the paper that the data should be cited.

In order for the receiver repository to react appropriately, the type of the citation also needs to be communicated. Thus the Trackback protocol is extended with an additional `type` key field in the HTTP POST. This can have values as in Table 1.

`backward`	The citation being notified is a backward citation. This is the default if the type key is omitted.
`cites`	Same as backward
`forward`	The citation being notified is a forward citation.
`cited-by`	Same as forward
`copy`	The sender is notifying that it holds a copy of an object which is held by the receiver.

Table 1: Available Citation Type Values

Note that the copy value can be used to communicate that the same object is held in multiple repositories. This illustrates that the type value can be used to notify of other kinds of cross-references.

De-Duplication

As currently implemented, if a source entry is edited then all the Trackbacks are initiated from its citations again, which could lead to duplicate Trackback entries. This should be handled in the receiver. The sender should be dumb and not judge the nature of the update, and just notify information and the receivers should distinguish how to react to the Trackback. The receiver could de-duplicate by just checking the existing list of URLs for a match with this URL to see if it has been seen before. Alternatively, the target may not care how much of a change there has been and just accepts an update and overwrites the first ping. Thus handling duplicates is outside of protocol.

Anti-Trackback

The current Trackback specification only deals with adding notification of citations. However, errors may occur in citations, with the wrong target object or target URL, or as a paper evolves, citations may be removed. In order to be a robust peer-to-peer protocol, senders should be able to notify receivers of the removal of a citation, and thus give the opportunity to remove the cross-citation, a process which might be described as "Anti-Trackback". It is proposed that this is handled by further overloading of ping POST message with the additional key `action` which can take one of the values `insert`, `update` or `delete` (with `insert` as the default if omitted).

Problems with the protocol

There are some known problems with the citation notification method which need further investigation. Some of these have been considered amongst the extensions above, but have not been implemented or need further consideration.

- Detecting bogus notifications. This has already been discussed as a potentially serious problem (it has impacted the use of Trackback in blogs quite severely), and needs a simple yet robust solution, which whitelists only partially address.

- Multiple citations from the same object in the same repository. As also discussed above, when a citation is edited, the Trackbacks are likely to be resent. Repositories need to handle such resending appropriately.

- Retraction of notifications. A mistaken or misdirected citation should to be retracted; again this is discussed above and a mechanism proposed.

- Detecting and entering citations with quality metadata. The process of detecting and entering citations is a difficult one. Entering them by hand is a tedious and highly error prone operation, whilst as discussed above, automatic methods are heuristic and can produce misidentified results. Further, whilst citation between papers is well-established, citations to and from data sets are not typically entered, nor well-represented within institutional repositories.

- Identifying different citations to the same object. It is frequently the case that the same digital object is stored in different places (e.g. the institutional repositories of multiple authors). If each sends a trackback ping to the target of a citation within the object, then the target repository could have a misleading picture of the citation status of the target; it could over measure the citation of the target. The target should try to identify that the digital objects are the same, and record the citation appropriately, although providing links to the multiple copies would be useful.

- The protocol is reliant on URLs being provided and a Trackback URL being detected in the page returned. This restricts Trackback to those citations which provide URLs (not that many do). The protocol would become more widely applicable if it could work through standard identifiers (e.g. DOI, ISSN) and some redirection service.

  Also the URLs the protocol uses are to "RDF-embeddable" pages (e.g. usually HTML or perhaps XML). This would in most systems usually mean the catalogue entry of the target object rather than the digital object itself; links in traditional publications are frequently given to digital object itself, and the catalogue entry is not considered.

Future Work

To date, there are two repositories which have implemented the Trackback notification service. The ePubs publication repository of STFC (http://epubs.cclrc.ac.uk) has implemented both a trackback sender and receiver service, while the British Atmospheric Data Centre has implemented only a receiver; this receiver handles the receipt of a trackback message by sending an

email message to the data centre administrator notifying the centre of a data citation. Initial experiments show promise, although it would require a critical mass of notification enabled repositories to test the notion out thoroughly.

The notion of a peer-to-peer communication protocol which can be used to propagate metadata between a loosely coupled federation of repositories of digital objects is a powerful one and applications are not limited to those discussed in the extensions above. There are a number of other possible extensions and uses of the protocol which have been identified.

- Tracking of who is requesting the Trackback URL and who is pinging. This could be done by generating a different Trackback URI for every invocation. This could aid the detection of bogus notification.

- As already discussed, forward citations can also be notified within the same protocol.

- If we extend the notion of notification to tracking copies of a paper stored in different repositories, we can establish co-citations between copies.

- Digital objects which cite each other are not necessarily all submitted together. Often someone submits a paper but data is not yet public until a later date. Trackback can be used to add the forward citation.

- Tracing a neighborhood of related work. Once the cross-citation network has been established across repositories as in Figure 2, then this can be used to analyse the space. Related papers and data can be traced by crawling over this network, revealing the neighborhood of interest of the current object, even though they might not cite them directly. Thus a form of highly directed search engine could be developed.

- Forming citation indices. Also, by crawling over the cross-citation network as in Figure 2, citation analysis could be undertaken, identifying how many citations a paper has, and identifying particularly influential object in the network – and including data as well as papers.

The role of the protocol within a research process should also be considered. We took the view that the digital objects are deposited within repositories independently and a post-hoc process of citation notification takes place. However, the protocol is neutral to this usage, and this may not be the only scenario in which can be usefully used. Other projects consider the research

process from a practitioner's point of view, with the data and publication arising from the process; a subject of potential research is how the Trackback protocol can be exploited in that scenario.

Bibliography

[Allinson et. al. 2007] Julie Allinson, Pete Johnston and Andy Powell, Dublin Core Application Profile for Scholarly Works (ePrints) , 2007. http://www.ukoln.ac.uk/repositories/digirep/index/Eprints_Application_Profile

[Apps 2005] Ann Apps. Guidelines for Encoding Bibliographic Citation Information in Dublin Core Metadata. Dublin Core Metadata Initiative Citation Working Group Recommendation, 2005. http://dublincore.org/documents/dc-citation-guidelines/

[Matthews et. al. 07a] Brian Matthews, Katherine Bouton , Jessie Hey, Catherine Jones, Sue Latham, Bryan Lawrence, Alistair Miles, Sam Pepler, Katherine Portwin. Cross-linking and referencing data and publications in Claddier. Proc. UK e-Science 2007 All Hands Meeting, 10-13 Sep 2007, UK e-Science 2007 All Hands Meeting, 10-13 Sep 2007. http://epubs.cclrc.ac.uk/work-details?w=37696

[Matthews et. al. 07b] Brian Matthews, Katherine Portwin, Catherine Jones, Bryan Lawrence. Recommendations for Data/Publication Linkage. JISC CLADDIER Project Report III, Nov. 2007. http://claddier.badc.ac.uk/trac/attachment/wiki/WikiStart/Report_III_RecommendationsForDataLinking-final.doc