

OAI-PMH Protocol over TAPIR Applicability Document

Kevin Richards
richardsk@landcareresearch.co.nz
20 August 2007

Introduction

OAI-PMH is a widely used generic metadata harvesting protocol for querying datasets available on the Internet. Web site is available at <http://www.openarchives.org/OAI/openarchivesprotocol.html>.

TAPIR is based on a similar idea but provides more flexible xml schema mapping functionality, and advanced searching. However it would be useful, or at the minimum, an interesting experiment, to send and receive OAI-PMH requests and responses using TAPIR service providers.

This document outlines the discoveries and conclusions that came out of such an investigation.

The investigation involved work with the .NET TAPIR provider (available at <http://sourceforge.net/projects/tapirdotnet>), where enhancements were made to this implementation of the TAPIR protocol, allowing OAI-PMH messages to be handled.

Basically the messages being returned from OAI-PMH requests can be represented as TAPIR output models based on the OAI-PMH protocol schema (<http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd>). The majority of the effort to achieve this work was with implementing a handler for processing OAI-PMH requests.

Minimum Implementation

A minimum implementation of the OAI-PMH protocol was used for this investigative work.

This includes handling the messages:

- ListMetadataFormats
- ListIdentifiers
- GetRecord

Other protocol messages should be handled for a complete implementation, including:

- Identify
- ListRecords
- ListSets

Only a single metadataFormat was handled in this work. The OAI-PMH specification allows for a data source to have provide the data via multiple metadata xml formats. For this investigation, a single group of TAPIR concepts was used to handle the metadataFormat value, and these concepts were mapped to Fixed Values, for example "TaxonName", <http://rs.tdwg.org/ontology/voc/tapir/structure/TaxonName/TaxonName.xsd> for the metadata format prefix and namespace.

OAI-PMH TAPIR Concepts and Messages

The required subset of OAI-PMH TAPIR concepts that were used for this investigation resulted in the following list:

identifyRepository = <http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:Identify/oai:repositoryName>
identifyUrl = <http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:Identify/oai:baseURL>
identifyVersion = <http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:Identify/oai:protocolVerion>
identifyEmail = <http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:Identify/oai:adminEmail>
identifyDescription = <http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:Identify/oai:description>

metadataFormatPrefix = http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:ListMetadataFormats/oai:metadataFormat/oai:metadataPrefix
metadataFormatSchema = http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:ListMetadataFormats/oai:metadataFormat/oai:schema
metadataFormatNamespace = http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:ListMetadataFormats/oai:metadataFormat/oai:metadataNamespace
getRecordIdentifier = http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:GetRecord/oai:record/oai:header/oai:identifier
getRecordDatestamp = http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:GetRecord/oai:record/oai:header/oai:datestamp
getRecordMetadata = http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:GetRecord/oai:record/oai:metadata
listIdsIdentifier = http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:ListIdentifiers/oai:header/oai:identifier
listIdsDatestamp = http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:ListIdentifiers/oai:header/oai:datestamp

The output model structure used to handle OAI-PMH messages was simply the OAI-PMH schema.

Output model mappings were defined for each message.

For the **ListMetadataFormats** message:

```

<?xml version="1.0" encoding="UTF-8" ?>
<request
  xmlns="http://rs.tdwg.org/tapir/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://rs.tdwg.org/tapir/1.0
    http://rs.tdwg.org/tapir/1.0/schema/tapir.xsd">
  <header>
    <source sendtime="2005-11-11T12:23:56.023+01:00">
      <software name="tapir_client.aspx" version="1.0"/>
    </source>
  </header>
  <search count="true" start="0" limit="20" envelope="false">
    <outputModel>
      <structure>
        <xs:schema location="http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema"
          xsi:schemaLocation="http://www.w3.org/2001/XMLSchema
            http://www.w3.org/2001/XMLSchema.xsd"/>
      </structure>
      <indexingElement path="/OAI-PMH/ListMetadataFormats/metadataFormat"/>
      <mapping>
        <node path="/OAI-PMH/request/@verb">
          <literal value="ListMetadataFormats"/>
        </node>
        <node path="/OAI-PMH/responseDate">
          <variable name="date"/>
        </node>
        <node path="/OAI-PMH/ListMetadataFormats/metadataFormat/metadataPrefix">
          <concept id="http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:ListMetadataFormats/oai:metadataFormat/oai:metadataPrefix"/>
        </node>
        <node path="/OAI-PMH/ListMetadataFormats/metadataFormat/schema">
          <concept id="http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:ListMetadataFormats/oai:metadataFormat/oai:schema"/>
        </node>
        <node path="/OAI-PMH/ListMetadataFormats/metadataFormat/metadataNamespace">
          <concept id="http://www.openarchives.org/OAI/2.0/oai:OAI-PMH/oai:ListMetadataFormats/oai:metadataFormat/oai:metadataNamespace"/>
        </node>
      </mapping>
    </outputModel>
  </search>
</request>
  
```

```
    </outputModel>
  </search>
</request>
```

For the **ListIdentifiers** message:

```
<?xml version="1.0" encoding="UTF-8" ?>
<request
  xmlns="http://rs.tdwg.org/tapir/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://rs.tdwg.org/tapir/1.0
    http://rs.tdwg.org/tapir/1.0/schema/tapir.xsd">
  <header>
    <source sendtime="2005-11-11T12:23:56.023+01:00">
      <software name="tapir_client.aspx" version="1.0"/>
    </source>
  </header>
  <search count="true" start="0" limit="20" envelope="false">
    <outputModel>
      <structure>
        <xs:schema location="http://www.openarchives.org/OAI/2.0/OAI-
PMH.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema"
          xsi:schemaLocation="http://www.w3.org/2001/XMLSchema
http://www.w3.org/2001/XMLSchema.xsd"/>
      </structure>
      <indexingElement path="/OAI-PMH/ListIdentifiers/header"/>
      <mapping>
        <node path="/OAI-PMH/request/@verb">
          <literal value="ListIdentifiers"/>
        </node>
        <node path="/OAI-PMH/responseDate">
          <variable name="date"/>
        </node>
        <node path="/OAI-PMH/ListIdentifiers/header/datestamp">
          <concept id="http://www.openarchives.org/OAI/2.0/oai:OAI-
PMH/oai:ListIdentifiers/oai:header/oai:datestamp"/>
        </node>
        <node path="/OAI-PMH/ListIdentifiers/header/identifier">
          <concept id="http://www.openarchives.org/OAI/2.0/oai:OAI-
PMH/oai:ListIdentifiers/oai:header/oai:identifier"/>
        </node>
      </mapping>
    </outputModel>
  </search>
</request>
```

For the **GetRecord** message: (the [OAIPMH_IDENTIFIER] is replaced with the identifier being requested in the GetRecord message)

```
<?xml version="1.0" encoding="UTF-8" ?>
<request
  xmlns="http://rs.tdwg.org/tapir/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://rs.tdwg.org/tapir/1.0
    http://rs.tdwg.org/tapir/1.0/schema/tapir.xsd">
  <header>
    <source sendtime="2005-11-11T12:23:56.023+01:00">
      <software name="tapir_client.aspx" version="1.0"/>
    </source>
  </header>
  <search count="true" start="0" limit="20" envelope="false">
    <outputModel>
```

```

    <structure>
      <xs:schema location="http://www.openarchives.org/OAI/2.0/OAI-
PMH.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xsi:schemaLocation="http://www.w3.org/2001/XMLSchema
http://www.w3.org/2001/XMLSchema.xsd"/>

    </structure>
    <indexingElement path="/OAI-PMH/GetRecord/record"/>
    <mapping>
      <node path="/OAI-PMH/responseDate">
        <variable name="date"/>
      </node>
      <node path="/OAI-PMH/request/@verb">
        <literal value="GetRecord"/>
      </node>
      <node path="/OAI-PMH/GetRecord/record/header/identifier">
        <concept id="http://www.openarchives.org/OAI/2.0/oai:OAI-
PMH/oai:GetRecord/oai:record/oai:header/oai:identifier"/>
      </node>
      <node path="/OAI-PMH/GetRecord/record/header/datestamp">
        <concept id="http://www.openarchives.org/OAI/2.0/oai:OAI-
PMH/oai:GetRecord/oai:record/oai:header/oai:datestamp"/>
      </node>
      <node path="/OAI-PMH/GetRecord/record/metadata/any">
        <concept id="http://www.openarchives.org/OAI/2.0/oai:OAI-
PMH/oai:GetRecord/oai:record/oai:metadata"/>
      </node>
    </mapping>
    </outputModel>
    <filter>
      <equals>
        <concept id="http://www.openarchives.org/OAI/2.0/oai:OAI-
PMH/oai:GetRecord/oai:record/oai:header/oai:identifier"/>
        <literal value="[OAI-PMH_IDENTIFIER]"/>
      </equals>
    </filter>
  </search>
</request>

```

If the above TAPIR messages are sent to a TAPIR provider, which has been mapped to required concepts in the OAI-PMH schema, the responses will be valid OAI-PMH responses.

The remaining functionality required to provide an OAI-PMH service is to handle OAI-PMH requests and return the valid responses. This was simply done by handling OAI-PMH request URLs and “forwarding” the messages to the TAPIR provider using the request structures shown above. The VB code developed to achieve this is shown later in this document.

The following example OAI-PMH URLs were handled:

1. <http://example.org/TapirDotNET/tapir.aspx/test?verb=ListMetadataFormats>
2. <http://example.org/TapirDotNET/tapir.aspx/test?verb=ListIdentifiers>
3. <http://localhost/TapirDotNET/tapir.aspx/test?verb=GetRecord&identifier=urn:lsid:example.org:record:123>

OAI-PMH and TAPIR

The following information was noted during the implementation of OAI-PMH over TAPIR.

- A literal value is used to define the response’s OAI-PMH/request/@verb value
- The **variable** element “date” is used for the /OAI-PMH/responseDate response element

- The xs:any elements must be handled as they are used within the OAI-PMH protocol. One place this xml type is used is for a generic “placeholder” for the metadata returned by GetRecord call.
- The OAI-PMH GetRecord response requires “embedded” xml, so an “xml” data type was introduced. When the mapping has this data type then the data is not encoded when added to the response (as we need the value to be actual XML).
- Functionality was added to the provider implementation to map to “LSID data”, ie data returned from an LSID resolution call (which is probably RDF/XML). This is used for the metadata element of the GetRecord response.
- TAPIR developers note : avoid setting the IndexingElement to an element that is within the set of elements that make up the main (indexed) element – eg do not make the indexing element /OAI-PMH/ListIdentifiers/header/identifier, make it /OAI-PMH/ListIdentifiers/header
- The following code was used to handle OAI-PMH requests. Any URL that contain a query parameter “verb” was directed to this handler.

```

public static void HandleOAIPMHRequest ()
{
    string verb = HttpContext.Current.Request["verb"];

    string id = HttpContext.Current.Request["identifier"];
    string body = "";

    if (verb == "GetRecord")
    {
        StreamReader rdr = new
StreamReader(TpConfigManager.TP_OAIPMH_DIR + "\\oai_GetRecord_call.tpl");
        body = rdr.ReadToEnd();
        rdr.Close();

        body = body.Replace("[OAIPMH_IDENTIFIER]", id);
    }
    else if (verb == "ListIdentifiers")
    {
        StreamReader rdr = new
StreamReader(TpConfigManager.TP_OAIPMH_DIR +
"\\oai_ListIdentifiers_call.tpl");
        body = rdr.ReadToEnd();
        rdr.Close();
    }
    else if (verb == "ListMetadataFormats")
    {
        StreamReader rdr = new
StreamReader(TpConfigManager.TP_OAIPMH_DIR +
"\\oai_ListMetadataFormats_call.tpl");
        body = rdr.ReadToEnd();
        rdr.Close();
    }

    string dsa = HttpContext.Current.Request.Params["dsa"];

    string url =
HttpContext.Current.Request.Url.GetLeftPart(UriPartial.Authority) +
HttpContext.Current.Request.Path + "/" + dsa;

    WebRequest http_request = WebRequest.Create(url);
    http_request.Method = "POST";
    http_request.ContentType = "text/xml";

    Byte[] b = System.Text.Encoding.UTF8.GetBytes(body);
    http_request.ContentLength = b.Length;

    Stream s = http_request.GetRequestStream();

```

```

        s.Write(b, 0, b.Length);
        s.Close();
        s.Flush();

        WebResponse res = http_request.GetResponse();

        StreamReader respRdr = new
StreamReader(res.GetResponseStream());

        string result = respRdr.ReadToEnd();
        respRdr.Close();

        HttpContext.Current.Response.ContentType = "text/xml";
        HttpContext.Current.Response.Write(result);
        HttpContext.Current.Response.Flush();
    }

```

Conclusions and recommendations

The OAI-PMH protocol is a harvesting protocol that allows a client to harvest a data source by requesting a list of Identifiers, then retrieving the records for these identifiers. Generic search capabilities are not supported over OAI-PMH.

OAI-PMH is a non-TDWG standard that is being used in a wide variety of fields, with a variety of data sources. It is useful to allow for the interoperability of TDWG standards and data with the other standards, institutions and data sources that are not specifically within the TDWG domain.

With relatively few modifications to TAPIR provider implementations OAI-PMH messages can be handled by a TAPIR data source.

No modifications to the TAPIR protocol specification are required to handle a simple implementation of OAI-PMH.

A simple set of OAI-PMH TAPIR concepts was used for this investigation. It would be very useful to create a CNS record of the entire list of TAPIR concepts in the OAI-PMH schema.

To create valid OAI-PMH responses, the TAPIT header must be omitted from a response.

Incremental harvesting was not implemented in this work. This may prove challenging with TAPIR as an OAI-PMH response returns a “resumption token” to the client, which is then returned to the data provider in subsequent calls. This would require the TAPIR provider software to “remember” these tokens and returned the next set of data for that request. The amount of effort to achieve this needs to be investigated.