ICS

WW

中华人民共和国文物保护行业标准

WW/T XXXXX—XXXX

文物元数据访问协议

Access Protocol of Cultural Relic Metadata

（征求意见稿）

（本稿完成日期：2017-03-17）

XXXX－XX－XX 发布　　XXXX－XX－XX 实施

中华人民共和国国家文物局　　发布

目　　次

前　　言

本标准起草单位：华中师范大学。

本标准撰写人：高劲松。

文物元数据访问协议标准规范

1 范围

本标准规范规定了文物元数据访问应遵循的协议和要求，对文物元数据进行访问时应采取的协议进行详细描述。本标准规范在OAI-PMH2.0的基础上进行修改采用，适用于Dublin Core元数据以及文物元数据核心元素集及专门元素集的访问与获取。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 25100-2010 信息与文献都柏林核心元数据元素集
OAI-PMH2.0 http://www.openarchives.org/OAI/openarchivesprotocol.html
ISO8601:2000 数据存储和交换形式·信息交换·日期和时间的表示方法
GB/T 7408-2005 数据元和交换格式·信息交换·日期和时间表示法
ISBN-10: 0321480910The Unicode Standard

3 术语和定义

下列术语和定义适用于本标准规范。

3.1 文物

文物是人类在历史发展过程中遗留下来的遗物、遗迹，具有历史、艺术、研究价值的东西。

3.2 文物元数据

关于文物信息资源或数据的一种结构化的数据。

3.3 核心元素（core element）

使用频率高的、共性的、可用于不同类型的信息资源描述的元数据元素。文物元数据核心元素具体定义参见《文物核心元数据标准》。

3.4 收集器（harvester）

收集器(harvester)是一个客户端应用程序，由服务提供商(service provider)操作，发布OAI-PMH请求，并从仓储（repositories）中获取元数据。

3.5 仓储（repository）

仓储（repository）是一种可被访问的网络服务器，由数据提供者（data provider）管理，将元数据发布给收集器（harvester）。资源的元数据可以通过仓储进行发布。

3.6 资源（resources）

资源（resources）是数据对象或者是有元数据说明的资料。

3.7 条目（item）

条目(item)是仓储的基本组织单元,一个条目是用来存储或动态生成单个资源的容器(container)，该资源可以有多种形态，每种资源形态都可以通过OAI-PMH协议以记录（records）的形式获得。

3.8 记录（record）

记录（record）是具有特定数据格式的元数据，使用OAI-PMH从条目里请求元数据的响应时，记录以XML编码的字节流（XML-encoded byte stream）的形式返回。通过条目唯一标识符之间的整合，记录可以被唯一标记。metadataPrefix标识了记录中元数据格式和该记录的时间戳。

3.9 唯一标识符（unique identifier）

唯一标识符（unique identifier）可以在仓储内唯一标识一个条目（item），OAI-PMH请求中所使用的唯一标识符用于从条目中提取元数据。唯一标识符的形式必须与URI（统一资源标识符）的语法保持一致。各个团体可以开发团体特定的URIschemes进行跨仓储协调使用。该scheme的唯一标识符的组成不能与公共的URI scheme重合，除非该scheme的标识符属于公共的scheme。具体的命名规则参见《文物元数据唯一标识符规范》。

3.10 集合（set）

集合（set）它是为获取方便而设定的条目组，不是必须具备的。仓储可以将条目组成集合。集合可以是单层的，比如一个简单的列表，或者有层次结构的。多层次架构可以具有不同的、独立的顶层节点。

3.11 选择性获取(selective harvesting)

选择性获取（selective harvesting）用于限制收集器对仓储内元数据部分的请求。OAI-PMH支持两种标准化的选择性获取：选择性获取和时间戳（Datastamps），以及选择性获取和集合元素（set membership）。

4 文物元数据访问协议

目前在元数据访问检索获取方面应用较为普遍的协议为OAI-PMH和Z39.50协议，经过详细的比较分析，本标准规范采用OAI-PMH协议进行文物元数据的访问（具体参见附录1）。

OAI-PMH协议采用了基于HTTP协议请求和XML格式响应的方式，这使其可以和目前的Web方式很好地结合，是一个应用较为普遍的元数据访问收割机制。多个服务提供者可以从多个数据提供者那里收割元数据，这在某种程度上确保了元数据信息的广泛传播。OAI-PMH已经从最初的应用于电子出版组织发展到可共享的各种信息资源的领域，目前已注册的数据提供者的仓储达到2843个，其中涉及到图书馆、高校、博物馆等多个领域。

为了标准的统一化和更广泛的适用性，OAI-PMH把Dublin Core(DC)作为互操作的标准元数据，但是由于DC的15个元素不能完全满足不同类型部门的需求，现在采用OAI协议的多数机构都是通过对DC进行扩展来达到自身的特殊要求， 因此OAI-PMH除了支持DC元数据外，也可以支持其它学科领域的元数据标准，只要这些元数据可以用XML描述，并有相应的XML Schema。同时，OAI-PMH又支持并鼓励应用多种元数据格式，数据提供者可以自由决定采用的元数据标准，只要它们可以用XML编码传输。因此OAI-PMH

支持课题组提出的文物元数据的格式，文物元数据以20个元素作为核心元素，另外还提出了专门元素以及元素修饰词等。

在文物方面，文物信息仓储是网络服务器上OAI协议的HTTP请求提交的对象，文物仓储内存储了各类文物资源的条目，每个条目又可能有多条文物元数据记录，这组成了文物OAI-PMH协议的数据提供者。用户可通过服务提供者利用OAI-PMH协议的Request和Response命令对文物元数据记录或资源进行访问、检索与收割，如图1所示。

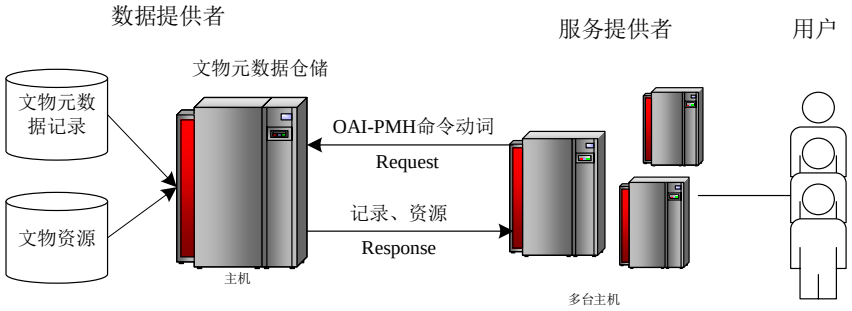

图1　文物元数据 OAI-PMH 协议访问模型

5　文物元数据 OAI-PMH 协议原则

OAI-PMH协议是以元数据收割为架构基础的首要协议。该协议的数据提供者为其他提供者准备数据，服务提供者可以使用特定的表示法在元数据schema中使用这些数据。

5.1　三方原则

OAI-PMH协议主要由数据提供者和服务提供者组成，在功能上，该协议允许把描述各种资源的元数据集中起来，因此还存在集成者，这组成了三方协议原则。通过收集元数据提供分布式资源的访问，并利用它们满足特殊服务需求，这是一个“第三方元数据模型”。

OAI-PMH定义了建立在元数据架构上的各种角色。数据提供者以一种或多种描述格式使其元数据被获取。从常见的帮助文档中，可获取的记录可以匹配一个标准的XML schema。OAI仓储（数据提供者机器上的元数据仓储）中存在各种不同格式的元数据，服务提供者通过发布名为harvester的程序来进行访问，并收集它所需格式的元数据，以满足各种不同的要求。服务提供者将处理所收集的元数据，并提供基于这些元数据的服务。

集成者从各种数据提供者那里收集元数据，并使它们在OAI仓储中可用，经再处理提高自身质量后，在此基础上构建增值服务。该集成者可以保证元数据具有较高质量，确保元数据的可存储性并形成一套标准化的体系。

5.2　收割原则

OAI-PMH 只管理数据传输，不支持查询功能，因此它不是一个交叉协议。但是，可以通过在服务提供者的服务器上获取帮助使其可在OAI仓储内进行查询。交叉搜索功能可以作为OAI架构的互补功能被添加和使用。由于修改、创建和跟踪已删除的条目都有时间戳，收集的数据包括从上一次收割之后的所有添加、修改、删除的记录数据。获取的内容不仅包含所有的数据，而且包括从上一次收割后的仓储修改的信息。仓储记录的时间戳是收割过程所依赖的关键元素。因此，在仓储的更新过程中应充分修改记录，从而使收集器可以了解自最后一次收割后的实际修改。调度程序可以实现对元数据收割的定期修改。调

度时间应取决于仓储的类型和更新频率。如果数据传输量太大，那么该协议允许定义传输数据的数量，进而划分转移的元素集，以此执行收割的相关步骤。OAI-PMH是基于HTTP的协议，因此附加功能可用于传输数据，例如，数据加密和数据压缩。

5.3 应用原则

在OAI架构中，内容提供者必须作为数据提供者，同时可选择性地作为服务提供者或集成者。内容提供者创建的所有类型的信息会进行交换和公开，无论是访问描述位置还是其他相关信息，如网站上发布的事件、专业文章、门户网站描述、术语工具和索引等。这些信息为内容提供者在制定帮助文件和目录时起到重大作用。

（1）跨机构和跨域服务

当数据产生在两个不同的机构时，即使这些机构正在使用类似的元数据方案，这些数据也是异构的。当访问各种类型目录时，会面临互操作的问题，如对共同访问点的定义，显示搜索结果的方式，提供一致查询结果的方式，显示相同格式化内容的必要性等。所有类型的资源可以通过一个单一的接口进行访问，无论它们覆盖一个特定区域（区域门户）还是一个特定主题。提供 Web 资源的传统门户网站可以允许用户在指定的网站进行深入信息搜索和检索。

（2）数据重用

跨域服务可能不会利用所有博物馆的信息，其中只有一部分可能作为重用数据以提供资源检索的案例。另一方面，跨域服务可能需要定义额外信息去充分挖掘文物资源、管理元数据、保存元数据、用于数据定位的特定元数据，从而将文物信息连接到地理信息系统。

（3）数据交换

通常，XML已经成为不同数据库间交换数据的标准化方式，OAI-PMH基于XML格式和HTTP协议，已成为元数据交换的标准化方式。鉴于OAI协议能够收集相关资源，因此可以促进元数据和资源之间的交换。

5.4 服务原则

（1）数据共享

内容提供者可根据特定的服务需求或者共享需求提供数据开放服务。数据共享是为仓储建立可获取元数据的一种开放的模式，利于并鼓励收集器尽可能广泛地获取数据。这并不排除控制仓储访问和记录收集数据的收集器应遵循通用的共享政策。

（2）数据公开

数据公开是在基于全球社区范围内的项目中，为仓储建立的特定使用服务。数据公开模式基于一个明确定义的机构，构成一个以服务为导向的建立仓储的方式。它必须作为内容提供者的主要方式满足基于 OAI 的服务。然后，提出“使自身元数据可用”的真正的政策。

（3）异步式原则

OAI架构具有异步性，而原始系统的仓储和服务也可能是异步的（如原始元数据存储于数据库中），这会使原始系统更新和服务数据更新之间产生延迟。最大的挑战在于能否尽可能确保仓储与原始系统更

新保持同步，能否尽可能确保服务和仓储之间保持同步，这种架构很难应用于频繁的数据修改中。但是一般来说，文物目录的修改频率并不高，因此文物资源可以应用OAI架构。

6 OAI-PMH 协议请求与响应

OAI-PMH 协议通过 6 个命令动词进行请求与响应，服务提供者利用这 6 个命令动词检索数据提供者提供的元数据。

命令动词的参数有三种类型：

- required，每一命令必须（must）包含该参数。
- optional，每一命令可能（may）包含该参数。
- exclusive，每一命令可能（may）包含该参数，但必须（must）是唯一的参数（除了动词参数）。

6.1 GetRecord

GetRecord用于从仓储中检索单独的元数据记录。GetRecord的Required参数指定了条目的标识符，以及从条目中获取的记录的元数据格式。根据仓储所跟踪的deletions级别，如果由metadataPrefix指定的元数据格式不能从仓储中和指定的条目中得到时，可以返回带有值为“deleted”的status属性的头部信息。

参数（Arguments）

- Identifier ：必要参数，指明了仓储中条目的唯一标识符。
- MetadataPrefix ：必要参数，指定了元数据格式的 metadataPrefix，该格式包含在返回记录的 metadata 部分中。对于 identifier 参数标识的条目，如果由 metadataPrefix 确定的元数据格式可以从条目中发出，则仅返回记录。仓储所支持的元数据格式和特定记录的元数据格式可以用 ListMetadataFormats 请求检索得到。

错误和异常情况

- badArgument — 该请求中包含非法的参数或缺少必要的参数。
- CannotDisseminateFormat —参数 metadataPrefix 指定的元数据格式不是 identifier 指定条目支持的格式。idDoesNotExist — 仓储不支持或存在不合法的 identifier 参数的值。

6.2 Identify

Identify用于检索有关仓储的信息。仓储也可以使用Identify动词返回一些额外的描述性信息。

参数（Arguments）

- 无

错误和异常情况

- badArgument — 请求包含非法的参数。

响应格式

响应必须包含以下元素中的一个实例。

- repositoryName：人类可读的仓储名称；
- baseURL：仓储的 base URL；
- protocolVersion：仓储支持的 OAI-PMH 的版本；
- earliestDatestamp：保证仓储中记录变化、修改或删除的所有时间戳的最小限度的一个 UTC 格式的 datetime。仓储不能使用低于 earliestDatestamp 元素指定的时间戳。earliestDatestamp 必须

表明仓储支持的最佳粒度。

- deletedRecord：仓储支持记录删除的方式。在 deletion 部分定义了删除记录的值包括 no、transient、persistent。
- Granularity：仓储支持的最佳收割粒度。在 ISO8601 中定义了时间粒度的合理的值是 YYYY-MM-DD 和 YYYY-MM-DDThh:mm:ssZ。

响应还必须包含以下元素的一个或多个实例。

- adminEmail：仓储管理员的 e-mail 地址。

响应可以包含以下可选元素的一个或多个实例。

- compression：仓储支持的压缩编码。建议采用 RFC2616 描述的 HTTP1.1 中的 14.11 部分的 Content-Encoding。compression 元素不应该包括隐含的 identity 编码。
- description：描述仓储的可扩展机制。例如，为响应 identify 请求，description 容器应该使用集合级元数据。每一个 description 容器必须附有描述容器结构的 XML Schema 的 URL。

示例

请求：

```
http://memory.loc.gov/cgi-bin/oai?verb=Identify
```

响应：

下述示例中 Identify 请求的响应包含三个 description 容器：

- oai-identifier 容器符合 http://www.openarchives.org/OAI/2.0/oai-identifier.xsd 的 XML Schema。仓储使用该模式选择符合特定格式的条目的唯一标识符。这个标识符的格式在 oai-identifier.xsd 的 XML Schema 中进行了解释。
- eprints 容器符合 http://www.openarchives.org/OAI/1.1/eprints.xsd 的 XML Schema。该模式已与 OAI e-print 机构商定，并包含机构的仓储的指定信息。
- friends 容器符合 http://www.openarchives.org/OAI/2.0/friends.xsd 的 XML Schema。仓储利用该模式使收集器通过其它仓储的 base URLs 指向其它仓储。建议使用 friends 容器，可以支持收集器发现仓储的网络位置。

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
        http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2002-02-08T12:00:01Z</responseDate>
<request verb="Identify">http://memory.loc.gov/cgi-bin/oai</request>
<Identify>
<repositoryName>Library of Congress Open Archive Initiative
                Repository 1</repositoryName>
<baseURL>http://memory.loc.gov/cgi-bin/oai</baseURL>
<protocolVersion>2.0</protocolVersion>
<adminEmail>somebody@loc.gov</adminEmail>
<adminEmail>anybody@loc.gov</adminEmail>
<earliestDatestamp>1990-02-01T12:00:00Z</earliestDatestamp>
<deletedRecord>transient</deletedRecord>
<granularity>YYYY-MM-DDThh:mm:ssZ</granularity>
<compression>deflate</compression>
<description>
<oai-identifier
       xmlns="http://www.openarchives.org/OAI/2.0/oai-identifier"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=
          "http://www.openarchives.org/OAI/2.0/oai-identifier
       http://www.openarchives.org/OAI/2.0/oai-identifier.xsd">
<scheme>oai</scheme>
<repositoryIdentifier>lcoa1.loc.gov</repositoryIdentifier>
<delimiter>:</delimiter>
```

```
    <sampleIdentifier>oai:lcoa1.loc.gov:loc.music/musdi.002</sampleIdentifier>
    </oai-identifier>
    </description>
    <description>
    <eprints
            xmlns="http://www.openarchives.org/OAI/1.1/eprints"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://www.openarchives.org/OAI/1.1/eprints
            http://www.openarchives.org/OAI/1.1/eprints.xsd">
    <content>
    <URL>http://memory.loc.gov/ammem/oamh/lcoa1_content.html</URL>
    <text>Selected collections from American Memory at the Library
    of Congress</text>
    </content>
    <metadataPolicy/>
    <dataPolicy/>
    </eprints>
    </description>
    <description>
    <friends
             xmlns="http://www.openarchives.org/OAI/2.0/friends/"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/friends/
            http://www.openarchives.org/OAI/2.0/friends.xsd">
    <baseURL>http://oai.east.org/foo/</baseURL>
    <baseURL>http://oai.hq.org/bar/</baseURL>
    <baseURL>http://oai.south.org/repo.cgi</baseURL>
    </friends>
    </description>
    </Identify>
</OAI-PMH>
```

6.3 ListIdentifier

ListIdentifier是ListRecords的缩写形式，仅返回头部（headers）信息而不是记录本身。ListIdentifier的可选（Optional）参数允许基于集合成员和/或时间戳的头部进行选择性获取。依据仓储对deletions的支持，如果请求中指定的与参数匹配的记录已经被删除，那么返回的头部（header）中可以有一个值为“deleted”的status属性。

参数（Arguments）

- from：可选参数，值为 UTC 格式的 datetime，指定了基于时间戳的选择性获取的时间下限。
- until：可选参数，值为 UTC 格式的 datetime，指定了基于时间戳的选择性获取的时间上限。
- metadataPrefix：可选的参数，指定了返回那些与 metadataPrefix 值相匹配的元数据格式，或者已经被删除的头部（基于仓储对 deletions 属性的支持）。仓储所支持的元数据格式或特定条目的元数据格式可以用 ListMetadataFormats 请求返回。
- set：带有 set Spec 值的可选参数，指定了选择性获取的集合标准（set criteria）。
- resumptionToken：独有参数，其值是由前一个 ListIdentifiers 请求返回的流控制标志的值，用于处理一个不完整列表。

错误和异常情况

- badArgument — 请求中包含非法参数或缺少必要参数。
- badResumptionToken — resumptionToken 参数的值无效或过期。
- cannotDisseminateFormat — 仓储不支持 metadataPrefix 参数的值。
- noRecordMatch — from、until 和 set 参数请求的记录不存在。

- noSetHierarchy — 仓储不支持该集合。

6.4 ListMetadataFormats

ListMetadataFormats用于在仓储中检索可获得的元数据格式。可选参数限制了该请求对某个特定条目元数据格式的获取。

参数（Arguments）

- Identifier：可选参数，指定条目的唯一标识符，而且该条目必须有有效的元数据格式。如果这个参数被忽略，那么响应将返回仓储所支持的所有的元数据格式。要注意的是，一个元数据格式被仓储所支持，并不意味着该元数据格式就能被仓储的所有条目发布。

错误和异常情况

- badArgument — 请求中包含非法参数或缺少必要参数。
- idDoesNotExist — 仓储不支持或存在不合法的 identifier 参数的值。
- noMetadataFormats — 指定条目中没有可用的元数据格式。

6.5 ListRecords

ListRecords用于从仓储中获取记录。可选的参数允许基于集合（set）成员和/或时间戳选择性获取记录。依据仓储对deletions的支持，如果与请求中参数匹配的记录已经被删除，那么返回的头部（headers）可以有一个值为“deleted”的status属性。有“删除”状态的记录不能表达成元数据形式。

参数（Arguments）

- from：可选参数，值为 UTC 格式的 datetime，指定了基于时间戳的选择性获取的时间下限。
- until：可选参数，值为 UTC 格式的 datetime，指定了基于时间戳的选择性获取的时间上限。
- set：带有 set Spec 值的可选参数，为选择性获取指定了集合标准（set criteria）。
- ResumptionToken：独有参数，其值是由前一个 ListIdentifiers 请求返回的流控制标志的值，用于处理一个不完整列表。
- MetadataPrefix：必要参数（当参数 resumptionToken 被使用时除外），定义了应该包含在返回记录中由 metadataPrefix 指定的格式。返回记录的条目应该由 metadata Prefix 指定的元数据格式进行发布。仓储支持的元数据格式和特定条目的元数据格式可以利用 ListMetadataFormats 请求获取。

错误和异常情况

- badArgument — 请求中包含非法参数或缺少必要参数。
- badResumptionToken — resumptionToken 参数的值无效或过期。
- cannotDisseminateFormat — 仓储不支持 metadataPrefix 参数的值。
- noRecordMatch — from、until、set 和 metadataPrefix 参数请求的记录不存在。
- noSetHierarchy — 仓储不支持该集合。

6.6 ListSets

Listsets用于返回仓储的集合结构（set structure），有利于选择性获取。

参数（Arguments）

- resumptionToken：独有参数，其值是前一个 ListSets 请求返回的流控制标志的值，用于处理一个不完整列表。

错误和异常情况

- badArgument — 请求中包含非法参数或缺少必要参数。
- badResumptionToken — resumptionToken 参数无效或过期。
- noSetHierarchy — 仓储不支持该集合。

示例

请求：

```
http://purl.org/alcme/etdcat/servlet/OAIHandler?
verb=ListSets
```

响应：

该响应表明仓储没有集合层级。

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
        http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2001-06-01T19:20:30Z</responseDate>
<request verb="ListSets">
         http://purl.org/alcme/etdcat/servlet/OAIHandler</request>
<error code="noSetHierarchy">This repository does not
support sets</error>
</OAI-PMH>
```

7 OAI-PMH 协议特征

7.1 HTTP 内嵌的 OAI-PMH 请求

7.1.1 HTTP 请求格式

OAI-PMH 请求在一定程度上可以表示为 HTTP 请求。OAI-PMH 请求必须使用 HTTP GET 或 POST 方法。POST 的优势是对参数的长度没有任何限制。仓储必须同时支持 GET 和 POST 方法。对于所有请求可以设置一个单一的 baseURL。baseURL 指定因特网的主机和端口，以及一个可选的 HTTP 服务器路径，该 HTTP 服务器可用作仓储。仓储将其 baseURL 作为 Identity 响应中 baseURL 元素的值。需要注意的是，任何路径都是由仓储的 HTTP 服务器的配置决定的。

除了 baseURL，所有请求都包含一个关键字参数（keyword argument）列表，它以键值对（key=value）的形式存在。参数可能会以任何顺序排列，多个参数必须以符号[&]隔开。每个 OAI-PMH 请求必须至少有一个键值对（key=value），用来指明通过收集器发布的 OAI-PMH 请求：

- key 是字符串‘verb’；
- value 是一个协议定义的 OAI-PMH 请求。

附加的键值对的数量和性质取决于独立请求的参数。

7.1.1.1 HTTP GET 中 URL 的 OAI-PMH 请求编码

GET 请求的 URLs 将关键字参数附加到 baseURL 后，用一个问号[?]标记分开。例如，对于一个仓储，若其 baseURL 为 http://an.oa.org/OAI-script，则其 GetRecord 请求的 URL 应该为：

http://an.oa.org/OAI-script?verb=GetRecord&identifier=oai:arXiv.org:hep-th/9901001&metadataPrefix=oai_dc

由于 URIs 的特殊字符必须要编码，上面的 GET 请求的 URL 的正确形式是：

http://an.oa.org/OAI-script?verb=GetRecord&identifier=oai%3AarXiv.org%3Ahepth%2F9901001&metadataPrefix=oai_dc

7.1.1.2 HTTP POST 中 OAI-PMH 的请求编码

OAI-PMH请求的关键字参数存在于HTTP POST的信息内容中。请求的内容类型（Content-Type）必须为application/x-www-form-urlencoded。例如，利用POST方法提交上述同样的请求，使用同样的baseURL，POST的格式是：

```
POST http://an.oa.org/OAI-script HTTP/1.0
Content-Length: 82
Content-Type: application/x-www-form-urlencoded

verb=GetRecord&identifier=oai%3AarXiv.org%3Ahepth%2F9901001&metadataPrefix=oai_dc
```

7.1.1.3 OAI-PMH 请求中关键字参数的特殊编码字符

在一些环境中，URIs的语法规则限制许多字符（保留字符）的特殊作用，如果以任何其他方式使用这些字符，则必须将这些字符变为转义字符，即百分号%加十六进制的字符编码表示。这些保留字符包括：

符号	URI 的作用（URI role）	转义序列
/	路径组件分隔符	%2F
?	查询组件分隔符	%3F
#	片段标识符	%23
=	名称/值分隔符	%3D
&	在查询组件中的参数分隔符	%26
:	主机端口分离器	%3A
;	命名空间分隔符	%3B
	空格字符	%20
%	转义指示器	%25
+	转义空间	%2B

当保留字符在使用时，其作用同上述URI角色对应，但必须用各自的转义序列来代表。在OAI-PMH中，当保留字符为请求键值对（key=value）的value部分时，则保留字符必须编码。上述编码规则对OAI-PMH的请求的GET和POST都适用。

7.1.2 HTTP 响应格式

请求响应格式为带有HTTP头字段的HTTP响应。

7.1.2.1 内容类型

对于所有OAI-PMH请求，其返回内容类型（Content-Type）必须是text/xml形式。

7.1.2.2 状态代码

OAI-PMH 错误通过 HTTP 状态代码（Status-Codes）加以区分。由于 OAI-PMH 使用 HTTP 作为传输层，实现 OAI-PMH 的服务器必须遵循 HTTP 状态定义代码（HTTP status code definitions），并通过这些状态代码报告 HTTP 传输层状态。HTTP 结果状态码可以应用到 OAI-PMH 请求的响应中，比如将 302，503 作为“200 OK”的补充。

- 302——允许仓储暂时重定向 OAI-PMH 请求到另一个仓储中。临时仓储的 URI 应该在 HTTP

响应的 Location 字段。

- 503——服务不可用，要求执行“Retry-After”并进行等待。收集器在尝试另一个 OAI-PMH 请求之前，需要等待“Retry-After”执行的时间。

7.1.3 响应压缩

在 OAI-PMH 请求中，响应压缩是可选的。OAI-PMH 请求的响应压缩在 HTTP 级别上进行处理操作，并有以下限制：

- OAI-PMH 请求在指定响应压缩的偏好参数时，收集器需包括一个接收编码（Accept-Encoding）标头。
- 没有设定接收编码（Accept-Encoding）标头的收集器将只能收到未压缩的响应。
- 当一个请求包含一个可接受的编码标头时，编码列表必须包含标识符(无压缩)编码(非零qvalue)。
- 仓储必须支持 HTTP 标识符编码。
- 在 Identify 响应中，除了“identity”元素，仓储需通过添加“compression”元素到 Identify 响应中，来表达其他可支持的编码。

7.2 XML 响应格式

对 OAI-PMH 请求的所有响应必须是具有正确格式的 XML 实例文档。XML 编码必须使用 Unicode 的 UTF-8 表示。XML 编码使用字符引用而不是实体引用。字符引用可以将 XML 响应作为独立文档，可以不依赖外部文档的实体声明进行操作。对于 OAI-PMH 请求的所有响应，其 XML 数据必须通过 XMLSchema 验证。

对 OAI-PMH 请求的响应具有以下共同点：

1.第一个输出的标签是一个 XML 的声明，其版本为 1.0，编码为 UTF-8，格式为<?xml version="1.0" encoding="UTF-8" ?>。

2.其余的内容被封装在名为 OAI-PMH 的根元素中。OAI-PMH 元素必须具有以下三个属性，用来定义响应的其他部分的 XML 的命名空间，以及验证 Schema 的位置：

- xmlns - 它的值必须是 OAI-PMH 的 URI 命名空间（http://www.openarchives.org/ OAI/2.0/）。
- xmlns：xsi - 它的值必须是 XML Schema 的 URI 命名空间（http://www.w3.org/2001/XMLSchema-instance）。
- xsi： schemaLocation - 成对出现，其中第一部分是 OAI-PMH 的 URI（由 XML 命名空间规范定义）命名空间（http://www.openarchives.org/OAI/2.0/），第二部分是为响应的验证 schema 的 URL（http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd）。

3.对于所有的响应，根元素的前两个子元素是：

- responseDate - 表示发送该响应的时间和日期，其数据类型是 UTCdatetime，必须以 UTC 描述。
- request -表示产生响应的协议请求。生成 request 元素的规则如下：
  - request 元素的内容必须始终是协议请求的 baseURL；
  - 对于 request 元素的唯一有效属性是协议请求的 key = value 键值对的 keys。该属性值必须是那些 key = value 键值对相应的 values 值；
  - 在产生此响应的请求没有导致错误或异常状态的情况下，request 元素的属性和属性值必须与协议请求中 key = value 键值对相匹配；

- 在产生这个响应的请求导致 badVerb 或 badArgument 的错误情况下，仓储必须只返回协议请求的 baseURL，在这些情况下则不能提供属性。

4.根元素的第三个子元素可以是：

- 在发生错误或异常情况下必须使用的错误元素；
- 对应于 OAI-PMH 请求命令动词、具有相同名称的元素。

上述GetRecord请求的成功响应示例如下：

```
<?xml version="1.0" encoding="UTF-8" ?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
        http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2002-05-01T19:20:30Z</responseDate>
<request verb="GetRecord" identifier="oai:arXiv.org:hep-th/9901001"
         metadataPrefix="oai_dc">http://an.oa.org/OAI-script</request>
<GetRecord>
<record>
     ...
</record>
</GetRecord>
</OAI-PMH>
```

7.3 UTCdatetime

日期和时间统一使用ISO8601编码，并在整个协议中以UTC表示。当时间被包括在内时，必须使用特殊UTC指示符（"Z"）。UTC表示日期，但没有指定时区标志。例如，1957-03-20T20：30：00Z是UTC 1957年3月20日下午8点30分。UTCdatetime在协议请求和协议回复中都使用。

7.3.1 协议请求的 UTCdatetime

在ListIdentifiers和ListRecords命令请求中，可选参数from和until的值表示为时间戳（Datestamps），该时间戳利用ISO8601编码，并以UTC表示。时间戳参数用于说明“基于时间戳的选择性收割”。这些参数支持“完整日期”，并使用ISO8601中定义的“完整日期加时，分，秒”的粒度，其正确格式是YYYY-MM-DD和YYYY-MM-DDThh：mm：ssZ。这两个参数必须具有相同的粒度。所有的仓储都必须支持YYYY-MM-DD。支持YYYY-MM-DDThh：mm：ssZ的仓储，应该在Identity响应中明确表明。如果收集器使用了比仓储支持粒度更小的粒度请求，那么就会产生错误。

7.3.2 协议响应的 UTCdatetime

对于响应 ListIdentifiers、GetRecord 和 ListRecords 请求而返回的记录，时间戳出现在返回记录的头部。这些时间戳使用 ISO8601 编码，以 UTC 表示，并必须根据仓储支持的最适合的粒度去表述。时间戳的值必须与“基于时间戳的选择性收割”的规则相对应。

每个协议响应包括 responseDate 元素，它必须是以 UTC 表示的响应的时间和日期，使用 ISO8601“完整日期加时，分，秒”中定义的变量进行编码，其格式为 YYYY-MM-DDThh：mm：ssZ。

协议回复中的resumptionToken可包括一个可选的参数expirationDate,它以UTC表示,使用ISO8601“完整日期加时，分，秒”中的变量进行编码，其格式为YYYY-MM-DDThh：mm：ssZ。

7.4 metadataPrefix 和元数据 Schema

OAI-PMH 支持从仓储中以多种元数据格式的方式发布记录。ListMetadataFormats 请求返回仓储中所有可用的元数据格式列表，每一个仓储都具有以下属性：

- metadataPrefix——定义发送给仓储的 OAI-PMH 请求中元数据格式的字符串。metadataPrefix

由所有有效的 URIunreserved 字符组成。metadataPrefix 参数用于对记录进行检索的 ListRecords， ListIdentifiers 和 GetRecord 请求，或者用于包含由 metadataPrefix 定义的元数据信息的记录或头部；

- 元数据 schemaURL——用于测试元数据格式有效性的 XMLschema 的 URL；
- XML 命名空间——元数据格式的通用标识符。

ListRecords 和 GetRecord 返回的每一个记录中的元数据必须遵守 XML 命名空间规范的要求。这意味着元数据部分的根元素必须包含 xmlns 属性，属性值为元数据的 XML 命名空间 URI。根元素还必须包含一个 xsi：schemaLocation 属性，其属性值包括用于元数据验证的 XMLschema URL。此 URL 必须与 metadataPrefix 的元数据 schema 的 URL 匹配，metadataPrefix 以参数的形式包含在 ListRecords 或 GetRecord 请求中（从 metadataPrefix 到元数据 schema 的映射由仓储对 ListMetadataFormats 请求的响应来定义）。

基于互操作性的目的，仓储必须无条件支持都柏林核心元数据（Dublin Core）。因此，该协议保留 metadataPrefix ‘ oai_dc’，以及无任何使用限制的 Dublin Core 元数据 schema 的 URL（为 http://www.openarchives.org/OAI/2.0/oai_dc.xsd ），相应 XML 命名空间的 URI 是 http://www.openarchives.org/OAI/2.0/oai_dc/。

metadataPrefix 术语 ' all '被留作将来使用，当前不应该使用此 metadataPrefix 术语。

7.5 流控制

许多 OAI-PMH 请求返回离散实体列表，例如 ListRecords 请求命令返回记录（records）列表，ListIdentifiers 请求命令返回头部信息（headers）列表，ListSets 请求命令返回集合（sets）列表，这些列表统称为请求列表（list requests）。在某些情况下，这些列表可能很大，因此在一系列请求与反馈中，可能需要对这些列表进行区分。列表区分主要包括：

- 仓储可能利用不完全列表（incomplete list）和恢复标志（resumptionToken）回复请求；
- 为了反馈完全列表（complete list），收集器（harvester）需要以 resumptionTokens 为参数发布一个或多个请求。完全列表包含连接请求序列的不完全列表（incomplete lists），被称为请求序列列表（list request sequence）。

流控制和恢复标志（resumptionToken）如下：

- 恢复标志（resumptionToken）的唯一用途如下：
  - 仓储必须包含 resumptionToken 元素作为每个响应的一部分，其中每个响应包括一个不完全列表；
  - 为了获取完全列表的下一部分，下一个请求必须使用 resumptionToken 元素值，作为请求的 resumptionToken 参数的值；
  - 响应包括不完全列表，其中必须包含一个空的 resumptionToken 元素；
  - 收集器对 resumptionToken 的其他使用不合理时，必须返回 error。
- 当使用 resumptionToken 时，不完全列表必须包含完整实体，例如从 ListRecords 请求的不完全记录列表中，返回的所有单个记录必须是完整的。
- resumptionToken 的格式在 OAI-PMH 中没有定义，因此收集器应该明确定义。
- 协议没有定义“不完全(incompleteness)”这个语义。因此，收集器不应该假定不完全列表中的成员符合选定的标准（如时间排序等）。
- 在返回 resumptionToken 后续请求的 URL 之前，收集器必须对特殊字符进行编码。

以下可选属性可能同 resumptionToken 本身一起，作为 resumptionToken 元素的一部分：

- expirationDate——表示当 resumptionToken 不再有效时显示的 UTCdatetime。

- completeListSize——表示完全列表基数。由于在仓储的一个列表请求序列中，completeListSize 的值可能变化，因此 completeListSize 的值可能只是对完全列表的实际基数的估计，并且在一个列表请求序列过程中可以修改。
- cursor——目前为止返回的完全列表元素数（cursor 始于 0）。

结合HTTP传输层设施,流控制机制提供了一些基本工具,仓储利用这些工具对其收集接口进行配置。机构在使用OAI-PMH时可能需要更多额外的工具 ，这样不仅可以用于仓储的收割接口，而且可用于各仓储收割的元数据。

7.5.1 resumptionTokens 的幂等性

使用 resumptionTokens 的仓储，必须允许收集器恢复不完全列表的请求序列，通过最新的 resumptionToken 重新发布请求列表。这样做的目的是让收集器从网络或其他错误中恢复，否则这些错误可能意味着列表请求序列必须再次开始。利用 resumptionToken 重新发布的请求列表在以下情况出现：

仓储中没有变化时。列表请求序列返回的完全列表没有变化的情况下，当最新的请求列表重新发布时，仓储必须返回同样的不完全列表，例如最新的未失效的 resumptionToken。

仓储中发生变化时。列表请求序列返回的完全列表有变化的情况下，这些变化（如仓储的改变或删除）发生在请求时间戳范围内的记录写入或写出时，对于使用 resumptionToken 的不完全列表请求而言，严格的幂等性不是必需的。反之，对于重新请求响应而返回的不完全列表，必须在最初请求列表的时间戳范围内给出未发生改变内的所有记录。

对于重新响应请求而返回的不完全列表，可能包含最初请求的时间戳范围内或范围外的记录。在仓储未发生实质改变的情况下，仓储可能返回badResumptionToken错误，表明收集器应该重新启动请求序列列表。

7.6 错误和异常情况

在发生错误或异常情况时，仓储必须注明OAI-PMH的错误，通过在响应中包含一个或多个 error元素，建立区别于HTTP 状态代码的OAI-PMH代码。当error元素足以指示错误或异常情况的存在时，仓储应报告请求处理出现的所有错误或异常。每个error元素必须具有一个代码属性，还可以具有一个自由文本字符串值，提供易于人类读写的有关错误信息，如下表所示。

错误代码	描述	可用命令动词
badArgument	该请求包括非法参数，缺少必要的参数，包括重复参数，或参数值存在非法的语法。	所有命令动词
badResumptionToken	resumptionToken 参数值无效或已过期。	ListIdentifiers ListRecords ListSets
badVerb	命令动词的参数值不是合法的 OAI-PMH 动词，动词参数丢失，或动词参数重复。	N / A
cannotDisseminateFormat	通过 metadataprefix 参数值确定的元数据格式不受项目或仓储的支持。	GetRecord ListIdentifiers ListRecords
idDoesNotExist	identifier 参数的值在仓储中是未知或非法的。	GetRecord ListMetadataFormats
noRecordsMatch	from，until，set 和 metadataPrefix 参数值的组合在一个空的列表中。	ListIdentifiers ListRecords
noMetadataFormats	没有可用于指定条目的元数据格式。	ListMetadataFormats

noSetHierarchy	仓储不支持该集合。	ListSets ListIdentifiers ListRecords

下面例子展示了存在非法动词参数情况下的错误处理，从这里开始所有请求的 URLs 将具有可读性。

请求：

```
http://arXiv.org/oai2?
verb=nastyVerb
```

响应：

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
        http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2002-05-01T09:18:29Z</responseDate>
<request>http://arXiv.org/oai2</request>
<error code="badVerb">Illegal OAI verb</error>
</OAI-PMH>
```

下面例子展示了仓储不支持集合的情况下 ListSets 请求的错误处理。

请求：

```
http://arXiv.org/oai2?
verb=ListSets
```

响应：

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
        http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2002-05-01T09:18:29Z</responseDate>
<request verb="ListSets">http://arXiv.org/oai2</request>
<error code="noSetHierarchy">This repository does not
support sets</error>
</OAI-PMH>
```

8 文物元数据 XML Schema

OAI-PMH 协议基于文物元数据 XML Schema 对文物元数据进行访问。文物元数据核心元素集借用 XML Schema 定义了如何使用 XML 描述文物信息，可以有效地对各种信息编目，并且易于修改、查询和使用。

文物元数据核心元素集在信息描述和定义中的应用，使语义 Web 得以实现。通过使用一种统一的标准来描述数据，同时基于精心设计并经过实践检验的解决方案，文物元数据核心元素集可以详细描述其他 XML 文档中的数据，从而可以在不同来源之间有效地交换和比较信息。

文物元数据核心元素集的命名域可直接引用《文物数字对象唯一标识符规范》，文物元数据核心集的XML Schema文档所在地址具体参见《文物元数据XML Schema》。

附 录 A
（规范性附录）
OAI-PMH 应用于文物元数据访问的可行性分析

1. 元数据访问协议调研现状

经调研，目前在元数据访问检索获取方面应用较为普遍的协议为 OAI-PMH 和 Z39.50 协议，经过详细的比较分析，本标准规范采用 OAI-PMH 协议进行文物元数据的访问。下面是 OAI-PMH 协议与 Z39.50 具体的比较分析。

OAI 协议起源于电子出版界的电子文档共享，它提供了基于元数据的、简单易行的互操作框架，是一种轻量级的协议，目前应用越来越广泛。Z39. 50 起源于图书馆界，最初是针对图书馆机读目录（Machine-Readable Catalog，简称 MARC）数据库共享而开发的标准，现在其主要应用领域仍然是图书馆的联机书目检索服务。在元数据标准上，Z39.50 协议采用的 MARC 元数据著录信息的准确度高，但字段繁琐重复，结构复杂，数据处理要求高，操作难度大、效率差，只有专业编目人员才能使用。其技术的复杂性提高了标准使用的门槛，显然在描述电子资源方面有一定的局限性；而 OAI 协议采用的 Dublin Core 相对简单，其内核只是一个很小的应用集合，而且经过几年的发展和修正，已经能很好地对大多数电子资源进行描述，该标准实施起来比较容易。通过 OAI 协议与 Z39.50 的比较分析，在文物资源方面，更适于选择轻量级的 OAI-PMH 协议。

表A.1 OAI 协议与 Z39.50 协议比较

	OAI	Z39.50
起源	电子出版界	图书馆界
应用领域	数字资源访问	书目数据共享
互操作程度	轻量级	重量级
元数据标准	具有 XML Schema 的元数据	MARC
体系结构	集中式检索	分布式检索
网络层次	基于 HTTP	基于 TCP/IP

2. 对 METS 的支持

METS（Metadata Encoding and Transmission Standard）是使用 XML 技术对元数据进行编码和表达文献结构的标准，主要由描述性元数据、管理性元数据、文件组、结构图、操作行为等部分组成，每部分又可以进一步模块化。在 OAI-PMH 中采用更加复杂、有表现力、准确描述数字资源的元数据格式可以有效地解决资源的准确收割和同步问题。而 METS 则是基于能够准确表示数字对象的更加复杂的格式来描述资源，这些格式允许表达资源相关的各种二次信息，包括描述元数据、管理元数据和结构元数据，它们也允许以传值或传址的方式明确传递标识符，包括资源本身。目前已有很多项目开始探讨将复杂对象格式与 OAI-PMH 相结合，以解决资源收割问题。数字资源所有相关的信息在被组织为 METS 结构后，将在传输的过程中被封装为 OAI 格式。METS 文档将作为 OAI 的元数据信息模块中的内容。OAI 除了提供元数据的模块外，还包括互操作相关的信息。在 OAI-PMH 和 METS 结合的项目中，北航和 HP 合作的中国数字博物馆项目，由北航计算机系、HP 实验室、多家博物馆和图书馆合作的中国数字博物馆项目（The China DigitalMuseum Project），将基于 Dspace 系统探讨数字资源的长期保存问题，并利用网格技术实现各大数字博物馆的互联互通和资源共享。在项目中拟采用 OAI-PMH 和 METS 相结合的方式，实现中心系统从多个分布式系统准确收割数字资源以及它们之间添加或者修改数字资源的紧密同步。

3. 对 LIDO 的支持

LIDO 是一个获取收割的 XML Schema，其主要目的在于传递元数据，使各种在线服务，从一个组织的在线数据库集合到聚合资源的门户网站上，并且包括公开、共享以及链接网络的数据。它的主要优

势在于能够代表关于博物馆对象的所有范围的描述性信息，它能够应用于所有类型的对象，如艺术、文化、技术和自然科学，并且支持多语种操作环境。在 LIDO 的处理工具中，OAICatMuseum 软件可以促进数据提供者（提供数据-博物馆的服务器）和数据收割者（接收数据-信任伙伴的服务器）之间的机器与机器之间的对话。OAICatMuseum 以 CDWA Lite XML 格式交流收集条目信息，并且基于 OAI-PMH 支持的 DC 核心元数据。

4. OAI-PMH 协议技术特色

OAI 协议最大的特色是通过相对简单的、独立于应用程序外的元数据收割协议，来实现异构分布元数据资源之间便捷的互操作。具体而言包括以下几点：

（1）职能分工明确，提高互操作效率

OAI 协议将元数据互操作的主体划分为数据提供者(Data Provider，DP)和服务提供者(Service Provider，SP)两类。数据提供者是拥有数字资源的主体，他只负责将自身的元数据用 XML 编码，以统一的都柏林核心集(DC)揭示，并以记录(Record)形式存储在支持 OAI 协议的服务器上，形成 OAI 仓储(Repository)。服务器在接收到数据需求方提出的获取数据请求后，判断是否满足请求，并给予回应。服务提供者是 OAI 机制中的数据收割方，也是面向用户提供数字化服务的主体。它通过 OAI 协议提供的指令动词(Verb)，向一个或多个数据提供方的 OAI 服务器发送元数据获取请求，定期进行收割聚合(aggregate)后在本地存储。在实际操作中，一个机构既可以是 DP（Data Provider），同时也可以是向其他 DP 收割数据的 SP（Service Provider），可以具备双重身份。OAI 协议将互操作的主体从职能上作了明确划分，双方各司其职，提高了互操作的效率。由于 SP 将数据收割至本地进一步加工建库，并开发各种数据库和平台，向用户提供多种增值服务。因此与分布式检索相比，本地检索提高了服务的响应速度和质量。

（2）独立于应用程序，降低互操作成本

由于 OAI 协议只关注数据的发布与获取，使元数据的互操作与平台无关，这大大简化了协议的配置与实施，降低了协议的门槛与执行的成本。DP 提供的数据仓储本身独立于应用平台，使 SP 在数据收割时不必投入大量的精力进行分布式检索，绕开了复杂的应用程序，简化了互操作的过程，提高了效率。作为数据收割者的 SP，可以将更多的力量投入到应用系统的开发中，利用收割来的数据为用户提供更好的增值服务。正因如此，OAI 协议使电子资源的互操作变得更为简单、灵活和高效。

（3）提供元数据筛选功能，实现同步更新

为了便于 SP 有选择地收割数据，OAI 协议提供了基于时间戳(Datestamp)和数据集(Set)的两种元数据筛选途径。OAI 仓储中的每条记录都有惟一标识符(URI)用以识别记录，并且标有时间戳用以标明记录创建、修改时间。SP 可以根据时间戳获取某段时间内新增加或修改过的数据。同时，为了便于针对资源内容进行选择性存取，DP 可以将元数据按不同主题划分成若干组，并以层式架构表示，以节点(node)作为各分类的区分，每一个节点即为数据集。这样，SP 就可以根据数据集的不同主题，选择在某一主题的数据集内获取数据。但是它的主题划分并非按照分类法的严格细分，只是学科领域或资源形式上的划分。由于协议的元数据筛选功能只是为了解决元数据的同步更新问题，为了简化操作起见，它只提供简单的查询功能，不支持关键词、截词、位置算符等复杂匹配，也不提供分布式检索机制。

（4）基于主流技术，易于推广和完善

OAI 协议基于 http 技术，使用 XML 来描述资源，用 XML 编码的字节流来应答收割请求，提供结构化的信息。它选用发展成熟的、适于揭示多学科领域信息资源的都柏林核心集(DC)为必须支持的元数据格式，同时支持对其他多种元数据格式的查询。这种采用 Internet 主流技术的方案，使 OAI 协议具备较好的通用性，更易于实现，便于广泛应用，随着主流技术的不断更新，OAI 协议的基础将不断完善，有利于保持较强的生命力。

5. OAI-PMH 协议应用于文物的典型案例

OAI 协议推出以后，得到了许多涉及到数字资源的组织和机构的支持与响应，也有一些著名机构的研究项目采用了 OAI 协议。下面列举几个比较有影响力的项目。

（1）Illinois

2001 年 7 月，在 Andrew W. Mellon 基金会的资助下，伊利诺伊大学香槟分校承接了一个文物领域的项目，该项目致力于测试 OAI-PMH 协议在构建、检索及发现文物信息资源服务方面的有效性。Illinois 项目已经集成了 38 个博物馆、数字图书馆及其他机构的元数据仓储的超过二百万条元数据记录，这些记录混合了数字信息及模拟信息的主要内容。

（2）Getty

Getty 博物馆提出了针对艺术品的元数据标准 CDWA，并提出了技术层面的数据互操作标准 CDWA Lite。CDWA Lite 是基于 CDWA 和 CCO 制定的描述艺术类作品和物质文化类作品的 XML Schema。该模式的目的是描述艺术作品和文物资源的核心记录的格式，并基于 CDWA 和 CCO 的数据元素和指导方针。该项目中使用 OAI 协议对联合目录和其他仓储的 CDWA Lite 记录进行获取和访问。

（3）ATHENA

ATHENA项目致力于为Europeana提供文物资源内容，其最终目的是将遍布欧洲各地的利益相关方和内容所有者连接起来，将促进新的数字内容加入Europeana，制定评估和集成的标准与工具，以致更多用户可以体验欧洲丰富的文物资源。在ATHENA项目中，OAI-PMH协议提供了获取工具及多种模块或平台之间的互操作机制，例如使用OAI-PMH技术标准及获取工具，Europeana的数据提供者或者任意第三方门户，可以通过OAI-PMH命令或服务请求检索服务提供者的元数据记录。ATHENA元数据的摄入工具能够管理异构的元数据记录的集合，同时可以公开元数据模式间的映射和转换服务。为了拓宽利用OAI-PMH协议的ATHENA元数据获取工具的功能，OAI-PMH命令动词应该在领域特定数据层实施，并且通过互操作机制使元数据获得开放。

参 考 文 献

[1]The Open Archives Initiative Protocol for Metadata Harvesting [EB/OL]. [2015-07-31].http://www.openarchives.org/OAI/openarchivesprotocol.html.
[2]Category for the Description of Work of Art (CDWA) [EB/OL].[2015-8-1] http://www.getty.edu/research/publications/electronic_publications/cdwa/index.html.
[3]Muriel Foulonneau.Open Archives Initiative – Protocol For Metadata Harvesting Practices of cultural heritage actors[R].France,2001.
[4]Kostas Pardalis, Nikos Simou, KostasRaftopoulos, et al. Report on the integration of the plug-in with the Europeana portal[R].Europe, 2011.
[5]Breeding, Marshall.Understanding the Protocol for Metadata Harvesting of the Open Archives Initiative[J]. Computers in Libraries, 2002, 22 (8): 24-29.
[6]Registered Data Providers[EB/OL].[2015-9-25].http://www.openarchives.org/Register/Browse Sites.
[7]LIDO[EB/OL].[2015-10-09].https://en.wikipedia.org/wiki/LIDO.
[8]What is LIDO[EB/OL].[2015-10-09]. http://network.icom.museum/cidoc/working-groups/lido/what-is-lido/.
[9]OAICatMuseum 1.0 [EB/OL].[2015-10-10].http://www.oclc.org/research/activities/Oaicat museum.html.
[10]Sarah L. Shreeves, Joanne S. Kaczmarek, Timothy W. Cole. Harvesting cultural heritage metadata using the OAI Protocol [J] .Library Hi Tech, 2003,02:159 -169.
[11]Category for the Description of Work of Art CDWA Lite[EB/OL].[2015-10-12].http://www.getty.edu/research/publications/electronic_publications/cdwa/cdwalite.html.
[12]张晓林. 元数据研究与应用[M]. 北京:北京图书馆出版社,2002.
[13]肖珑,申晓娟. 国家图书馆元数据应用总则规范汇编[M]. 北京:国家图书馆出版社,2011.
[14]GB/T 25100-2010,信息与文献都柏林核心元数据元素集[S]. 北京:中国标准出版社,2010.
[15]GB/T 7408-2005, 数据元和交换格式信息交换日期和时间表示法[S].
[16]ISO8601:2000,数据存储和交换形式信息交换日期和时间的表示方法[S].
[17]牛振东,朱先忠.OAI-PMH 协议应用指南[R].北京:国家图书馆,2004-05.
[18]牛振东,朱先忠,赵春宇.OAI 元数据获取协议综述报告[R].北京:国家图书馆,2003-03.
[19]陈凌.中国高等教育数字图书馆技术标准与规范[S].北京:CALIS 管理中心,2004.
[20]CADAL 20803-2012,数字资源服务协议和接口标准第三部分:资源检索协议规范[S].CADAL 项目管理中心,2012.
[21]张海涛,郑小惠,张成昱. 数字图书馆的互操作性研究:Z39.50 和 OAI 协议的比较[J]. 现代图书情报技术,2003,02:13-15.
[22]马蕾. 元数据及其封装标准 METS 研究[J]. 情报杂志,2002,02:56-57.
[23]程妍妍. 基于 METS 的电子文件元数据封装研究[J]. 湖北档案,2011,07:11-14.
[24]曾婷,张成昱. 基于 OAI-PMH 和复杂对象格式的资源收割机制探讨[J]. 现代图书情报技术,2005,11:14-18+23.
[25]齐华伟,王军. 元数据收割协议 OAI-PMH[J]. 情报科学，2005，03:414-419+425.
[26]郭少友. 基于 OAI-PMH 的信息资源整合[J]. 大学图书馆学报，2005，03:16-18.