2. PROTOCOLLO DI RACCOLTA OAI-PMH

2.1 INTRODUZIONE

Il presente capitolo ha per oggetto il protocollo di raccolta dei metadati OAI-PMH (Open Archive Iniziative – Protocol Metadata Harvesting), che, nella versione 2.0 attuale, rappresenta un valido documento di riferimento [S2] per gli implementatori di sistemi documentari di tipo open access e, più precisamente, per coloro che implementino i propri open archive OAI compatibili.
In particolare, dopo un accenno alla nascita e all'evoluzione del protocollo, vengono trattati:

- *concetti preliminari di repository, harvester e record di metatadati*: i primi due, come soggetti interessati nella comunicazione tramite il protocollo e il terzo, quale contenitore e mezzo di rappresentazione dell'oggetto di scambio (metadati) tra harvester e repository
- *identificatore unico*: quale mezzo, condiviso tra i vari data provider, per etichettare gli item delle risorse
- *set*: costrutto attraverso il quale si rende possibile l'ampliamento dei servizi a valore aggiunto offerti dagli harvester
- *selective harvesting e partizionamento della risposta*: meccanismi attraverso i quali è possibile effettuare rispettivamente, semplici raccolte selettive di record di metadati basate sui set e/o sui datestamp, per il primo, e suddivisione della risposta da parte del data provider in risposte di dimensioni più piccole, per il secondo

- *richieste, risposte ed errori OAI-PMH*: che permettono a service e data provider di stabilire il processo di comunicazione che garantisce l'interscambio informativo

La conoscenza del protocollo e dei suoi meccanismi si rivela, dunque, di fondamentale importanza per tutti coloro che intendano aderire all'iniziativa diventando essi stessi attori attivi in una nuova forma di comunicazione scientifica in ambito open access.

2.2 PROTOCOLLO OAI-PMH

Come già precedentemente detto nel capitolo introduttivo, uno degli aspetti principali del meeting di Santa Fe del 1999 fu la scelta del protocollo per il *metadata harvesting* (raccolta dei metadati) che permettesse la disseminazione dei metadati da parte dei *Data Provider* e la conseguente possibilità di raccolta da parte dei *Service Provider*.
Nella sua fase di progettazione si tenne conto del modello di Service e Data provider di Repec, delle esperienze dell'UPS (Universal Preprint Server) e del lavoro del gruppo dei partecipanti al meeting.

Il protocollo fu definito in via sperimentale prendendo il nome di *Dienst Protocol* e pose la sua attenzione sulla diffusione degli e-print. Il formato di metadati provvisoriamente adottato fu l'*OAMS* (Open Archieve Metadata Set), corrispondente ad un sottoset del Dubline Core. Da questa fase vennero fuori delle scelte che sarebbero poi state mantenute nelle revisioni successive di quello che divenne il protocollo cosiddetto di harvesting o raccolta.

Furono così stabiliti:

- HTTP – come protocollo di trasporto a cui appoggiarsi per effettuare le richieste di metadati da parte del Service provider utilizzando i metodi GET e POST.
- XML – come mezzo di rappresentazione della risposta da parte del Data Provider.

Successivamente il protocollo passò ad una versione che tenne conto della diffusione generica di *documenti* come "oggetti di scambio informativo" piuttosto che di soli e-print.
Esso fu spinto dai lavori presso la Cornel University e dagli incontri stabiliti dalla Digital Library Federation prendendo il nome di OAI-PMH (Open Archives Iniziative - Protocol Metadata Harvesting) nella versione 1.0. Introdusse, sempre in via sperimentale, la necessità di una *soglia minima di interoperabilità,* ossia un protocollo che fosse facile da implementare da parte dei soggetti aderenti all'iniziativa e che potesse garantire con il minimo sforzo la comunicazione tra archivi.
Così fu scelto come "formato minimo" dei metadati, da implementare da parte dei Data Provider, il *Dublin Core,* lasciando poi l'autonomia di poter in aggiunta usare altri formati di rappresentazione esistenti o di crearne di nuovi che potessero essere codificati in XML.

L'introduzione della nuova tecnica di validazione dei documenti attraverso XML-Schema portò alla revisione dell'OAI-PMH stabilendo così la versione 1.1 sempre in via sperimentale e senza ulteriori novità.

La versione *definitiva* e *stabile* (OAI-PMH v.2) del protocollo si ebbe nel Giugno del 2002.

Il protocollo è ancora in questa versione e conferma gran parte delle scelte precedenti riguardo all' HTTP per il livello transport, al Dublin Core per il formato minimo dei metadati e all' XML per la validazione delle risposte ricevute dal Service Provider.

La novità sostanziale è quella di consentire ai metadati di essere il mezzo di scambio di risorse (resources) tra sistemi, slegandosi dalla natura dell'informazione scambiata.

Tale evoluzione del protocollo può essere brevemente riassunta nella seguente tabella:

	Convenzione di Santa Fe	OAI-PMH v.1.0/1.1	OAI-PMH v.2.0
Protocollo transport	HTTP	HTTP	HTTP
Formato risposta	XML	XML	XML
Stato	Sperimentale	Sperimentale	Stabile
Dati trattati	E-print	Documenti come oggetti	Resource
Formato metadati	OAMS	Dublin Core Unqualified	Dublin Core Unqualified

La definizione di resource è la seguente: *un resource è un generico "oggetto": documento, immagine, servizio, collezione di resource stessi, a cui il metadato si riferisce.*

In tal senso la natura della risorsa, indipendentemente dal fatto che sia fisica o digitale, che sia un documento o un filmato audio, è al di fuori dell'interesse dell'OAI-PMH.

Lo scambio dei metadati non coinvolge più solo risorse informatiche definibili a priori, ma *risorse informative* in generale referenziabili attraverso metadati aventi

un formato appositamente stabilito.

L'OAI-PMH introduce gli elementi portanti che servono a strutturare gli open archives.
Essi sono:

- Repository
- Harvester
- Record
- Identificatore unico (Unique identifier)
- Set

Tali elementi saranno trattati nei paragrafi seguenti.

2.3 REPOSITORY (DATA PROVIDER)

Un *repository* è un server di rete normalmente interfacciato ad un database contenente i record dei metadati.
Il soggetto che si occupa della sua gestione è il Data Provider il quale rende disponibile i metadati ed eventualmente i *fulltext* richiesti dall'harvester del Service Provider.

2.4 HARVESTER (SERVICE PROVIDER)

Un *harvester* è un'applicazione client tramite la quale il Service Provider può inviare delle richieste OAI-PMH al repository del Data Provider per effettuare l'harvesting (raccolta) dei metadati.
In questo modo l'harvester può fornire servizi a valore aggiunto (searching, suddivisione in collezioni, browsing, ecc.) agli utenti che si connettono ad esso.
L'harvester può effettuare una di sei richieste per volta che verranno dettagliatamente trattate al paragrafo 2.9.

2.5 RECORD

L'ultima versione del protocollo OAI-PMH riferisce lo scambio di metadati tra open archive come mezzo di accesso a un *resource* (documento, immagine, servizio, collezione di altri resource) ossia risorse in generale e non solo documenti di eprint.
Un resource all'interno di un repository viene riferito attraverso l'*item* ad esso relativo il quale a sua volta è univocamente individuato attraverso un *identificatore unico*.

L'item ha infatti la funzione di "contenitore di metadati", in quanto può contenere uno o più *record* ognuno dei quali riferito ad un metadato in uno specifico formato.

La figura 1 seguente mostra la relazione intercorrente tra risorse, item e record di metadati:

Figura 1 - Relazione tra resource, item e record

Un record è ritornato al Service Provider, che ne ha fatto richiesta, codificato in un documento XML. Tale documento rappresenta la *response* della corrispondente *request* OAI-PMH che riferisce un item specifico.
Un record si compone dunque delle seguenti parti:

- Header – è la parte iniziale del record e contiene le informazioni unique identifier e datestamp, che insieme al formato del metadato (normalmente indicato nel parametro metadataPrefix di una richiesta, come discusso nel

paragrafo 2.9.3) lo individuano in modo univoco. Esso consiste a sua volta di:

- **identifier** - che corrisponde all'unique identifier dell'item che contiene il record.
- **datestamp** – che rappresenta la data di creazione, modifica o cancellazione del record. Il datestamp, come vedremo, può essere utilizzato nelle request del Service Provider per effettuare un *harvesting selettivo* ossia: un modo per effettuare la selezione dei dati da raccogliere basato sulla data di modifica associata ai record.
- **setSpec -** Zero o più elementi, che indicano l'appartenenza del record a uno o più *set* ossia insiemi di catalogazione dei record che, come vedremo, consentono l'harvesting selettivo basato sui set.
- **deleted** – opzionale, è un campo di stato indicante che il record in questione, ossia il metadato, non è più disponibile. Se supportato dal repository, può avere uno di tre valori predefiniti:
  - *no*: Indica che il repository non rivela l'eventuale stato deleted di un record
  - *persistent*: Il repository tiene traccia in maniera persistente delle cancellazioni dei suoi record e indica l'eventuale stato deleted di un record
  - *transient*: Il repository non garantisce che la gestione di tutti i record cancellati sarà per sempre effettuata: potrà non necessariamente indicare informazioni su un record con lo stato di deleted

- Metadato – è un singolo formato del metadato di un item. Secondo le

specifiche OAI-PMH è necessario che il data provider adotti il Dublin Core come formato minimo per i propri metadati, ma lascia comunque la possibilità di supportare formati multipli per lo stesso item. In alcuni tipi di richieste è possibile specificare il formato di metadato desiderato attraverso l'argomento metadataPrefix trattato più specificatamente al par 2.9.3.

- About (opzionale) – è un campo che può contenere informazioni aggiuntive di utilità per una specifica comunità. L'OAI-PMH non pone alcuna restrizione circa l'uso di tale campo, l'unica premura è quella di fornire dei tag XML opportuni per codificare l'elemento in conformità con l' XML Schema. Alcune informazioni che vengono comunemente associate a tale elemento sono:
    - **diritti di utilizzo** – vengono utilizzati da alcuni repository per aggiungere i termini di utilizzo della risorsa a cui il metadato, reso disponibile, si riferisce.
    - **provenienza** – indica, se il metadato è stato raccolto da un altra fonte, il repository da cui è stato prelevato e quando. Un caso tipico è quello di un'applicazione che funge sia da data che da service provider come ad esempio CDSware.

2.6 IDENTIFICATORE UNICO (UNIQUE IDENTIFIER)

L'identificatore unico permette di referenziare in maniera univoca un item dentro un repository ed è usato nelle richieste OAI-PMH per estrarre i relativi metadati.

È bene precisare che l'identificatore dell'item e quindi dei metadati ad esso associati, non è anche identificatore del resource di partenza in quanto

l'OAI-PMH nulla impone circa l'identificazione di quest'ultimo.
Ad ogni modo, è buona norma indicare all'interno di un metadato che referenzia un resource, un elemento che stabilisca un collegamento tra il metadato stesso e l'identificatore del resource (scelto in maniera arbitraria: URI, URN, URL, ecc.).

Il *formato dell'identificatore* [S5] adottato dall'OAI è quello degli URN (Uniform Resource Name) ossia un URI (Uniform Resource Identifier) che non sia un URL (Uniform Resource Locator).
La sostanziale differenza tra un URN ed un URL è che il primo ha come scopo quello di etichettare in maniera persistente una risorsa, mentre il secondo tiene principalmente traccia della sua posizione nella rete.

2.6.1 Componenti dell'identificatore unico

La definizione dell'identificatore OAI segue la sintassi:

oai-identifier = schema : namespace-identifier : local-identifier

Abbiamo così tre componenti separate dal delimitatore ":" di cui si parlerà tra breve.

2.6.1.1 Schema

La prima componente indica lo *schema* a cui l'URI aderisce e quindi la sintassi utilizzata per la sua composizione.
Nel caso di identificatori di item OAI si avrà:

schema = “oai”

2.6.1.2 Namespace identifier

La seconda componente è il *namespace-identifier*, ossia un identificatore che permette di rendere l’oai-identifier univoco.
Le versioni 1.0 e 1.1 del protocollo adottavano come identificatore di namespace il nome del repository scelto e gestito nel rispetto di maiuscole e minuscole, ed opportunamente *registrato* dall’OAI.
Con l’attuale versione 2.0, le organizzazioni che aderiscono all’iniziativa dell’OAI devono scegliere un nome di dominio “registrato” come namespace-identifier, mantenendo, anche in questo caso, il rispetto delle maiuscole e delle minuscole in maniera permanente.
Questo nuovo approccio permette di evitare la necessità di un servizio di registrazione per gli oai-identifier in quanto un identificatore a cui è associato un domain name registrato “univoco”, diventa esso stesso univoco. Così si avrà:

namespace-identifier = nome di dominio

dove il nome di dominio inizia necessariamente con un carattere alfabetico, mentre i successivi caratteri possono essere alfanumerici.

2.6.1.3 Local identifier

La terza componente è un *local- identifier* costituito da uno o più caratteri di un set ristretto denominato "uric" o "uri characters".
Così si avrà:

local-identifier = uno o più crt. uric

Il set di caratteri uric è stato scelto opportunamente per favorire le caratteristiche di *trascrivibilità* e di *usabilità* sia nei sistemi informatici che in sistemi di comunicazione senza l'ausilio del computer.

Esso corrisponderà a:

uric = crt. riservati | crt. non riservati | sequenze di escape (escaped)

dove al simbolo "|" rappresenta l'operatore logico OR ed ogni carattere potrà essere scelto tra uno dei tre tipi: riservati, non riservati e sequenze di escape.

2.6.1.3.1 Caratteri riservati

Per "riservati", si intendono dei caratteri speciali utilizzati dentro le componenti dell'URI con un ben determinato scopo:

riservati = ";" | "/" | "?" | ":" | "@" | "&" | "=" | "+" | "$" | ","

normalmente tali caratteri vengono utilizzati come delimitatori per le componenti

di un URI (ad esempio il carattere "?" viene utilizzato per indicare l'inizio delle coppie nome-valore che vengono passate ad uno script cgi attraverso il form di una pagina HTML) e necessitano di essere esplicitati da una *sequenza di escape* (vedi paragrafo 2.6.1.3.3) se utilizzati come dati all'interno di una componente che non sia quella del local-identifier. In tal caso infatti, questi caratteri non hanno un significato speciale e possono qui essere usati senza sequenza di escape.

2.6.1.3.2 Caratteri non riservati

Per "non riservati" si intendono tutti i caratteri ammessi all'interno di una componente dell'URI, ma che non sono riservati ad uno specifico scopo. Questo set comprende caratteri alfabetici maiuscoli e minuscoli, cifre decimali ed un limitato set di segni di punteggiatura e simboli:

non riservati = crt. alfanumerici | segni

dove

caratteri alfanumerici = tutte le lettere maiuscole e minuscole dell'alfabeto e le cifre da 0 a 9

segni = "-" | "_" | "." | "!" | "~" | "*" | " ' " | "(" | ")"

essendo tali caratteri non riservati per scopi specifici, essi possono essere utilizzati, per definizione, senza ricorrere ad equivalenti sequenze di escape anche all'interno del componente dell'URI relativo al local-identifier.

2.6.1.3.3 Sequenze di escape

Per "sequenze di escape" si intendono delle opportune terne di caratteri che

permettono di rappresentarne altri con un particolare significato sintattico in una determinata componente dell'URI, oppure di rappresentare caratteri del set di codifica ASCII che non siano stampabili o che non siano permessi.
Ad esempio il carattere ASCII " " di spazio è non permesso per l'utilizzo in una componente, poiché durante le fasi di trascrizione o trattamento dell'URI da parte di programmi di word-processing, spazi *non significativi* possono essere involontariamente introdotti e spazi *significativi* possono essere tolti; di conseguenza bisognerà utilizzare la sequenza di escape "%20".
Così una sequenza di escape è codificata come una terna consistente nel carattere "%" seguito da due cifre esadecimali che rappresentano il codice del carattere da rappresentare:

sequenza di escape = "%" cfr.hex cfr.hex

dove

cfr.hex = crf.numerica da 0 a 9 | "A" | "B" | "C" | "D" | "E" | "F"

Tutti i caratteri all'interno della componente local-identifier che non appartengono al set dei riservati o a quello dei non riservati, devono essere rappresentati dalla relativa sequenza di escape seguendo la stessa codifica adottata per i caratteri riservati nelle request OAI-PMH.

Per la codifica infatti viene utilizzata la seguente tabella:

Carattere	Sequenza di escape
/	%2F
?	%3F
#	%23
=	%3D
&	%26
:	%3°
;	%3B
(spazio)	%20
%	%25
+	%2B

2.6.2 Esempio di identificatore

Nei paragrafi precedenti si è discusso come si compone l'identificatore OAI, di seguito vediamo il seguente esempio:

oai:xXx.org:math-net/10121-3523%20**AC**

dove

oai è lo schema dell'URI;

xXx.org è il domain name per il namespace-identifier;

math-net/10121-3523 AC è il local-identifier con "/" e "-" ammessi e " " (spazio) con la relativa sequenza di escape.

2.7 SET

È possibile strutturare un record all'interno di un repository in modo tale da indicare l'appartenenza dell'item ad esso associato, ad un cosiddetto *set*.

Un set è quindi un costrutto utilizzato dal Data Provider per raggruppare item sotto una stessa categoria al fine di consentire ai Service Provider di realizzare una raccolta più sofisticata dei metadati basata su di esso.

Il protocollo OAI-PMH considera l'implementazione dei set opzionale da parte dei Data Provider e non dà particolari disposizioni circa il loro utilizzo.
In generale questa forma di catalogazione può essere impiegata dalle comunità di Data Provider per creare le proprie configurazioni di set, adottando eventualmente un *vocabolario prestabilito* per la definizione dei nomi da assegnare loro.
Le comunità dovrebbero anche sviluppare un meccanismo di esposizione di set agli harvester per consentire loro di sapere quali sono quelli con i quali effettuare la raccolta selettiva dei metadati (selective harvesting, vedi paragrafo 2.8). Come si vedrà più avanti (paragrafo 2.9.4.6) ciò può essere fatto implementando la richiesta *ListSets*.

2.7.1 Struttura dei set

La parte iniziale (*header*) della struttura di un record può contenere un campo opzionale e ripetibile setSpec tramite il quale è possibile indicare l'appartenenza

dell'item da esso rappresentato ad un ben determinato set.
L'organizzazione dei set da parte del Data Provider può essere *gerarchica* o più semplicemente consistere in un *elenco*.
Un'organizzazione gerarchica prevede la possibilità di far derivare dei sottoset (set figli) da un set più generale, consentendo così di avere gerarchie multiple con set di livello superiore indipendenti.

Un esempio di organizzazione gerarchica potrebbe essere:

- Scienze
    - Matematica
    - Fisica

dove Matematica e Fisica sono sottoset di Scienze.

Mentre un'organizzazione lineare potrebbe essere:

- Libri
- E-print
- Tesi
- Journal

in cui non sono presenti specializzazioni dei set adottati Libri, E-print, Tesi e Journal.

Ogni record conterrà le seguenti informazioni nella parte relativa all'header:

- *SetSpec* - questo elemento è ripetibile poiché un record può appartenere a più set e contiene il nome con cui identificarli.

Nel caso di semplici elenchi, basterà utilizzare tanti elementi setSpec, quanti sono i set a cui appartiene il record.

Per organizzazioni di tipo gerarchico invece, sarà possibile identificare tutte le componenti (di set) della gerarchia, a partire dal set radice, separate dal carattere ":" similmente ad un percorso di file in ambiente Unix in cui si utilizza il simbolo "/" di separazione delle varie componenti partendo dalla directory radice.

Ovviamente gli identificatori di set non potranno contenere il carattere ":" per evitare confusione.

Se un record appartiene ad un set e ad un suo sottoset (ad esempio scienze e scienze:matematica), si dovrà indicare solamente il sottoset (scienze:matematica), assumendo l'implicita appartenenza anche al set di livello superiore (scienze).

- *SetName* - questo elemento opzionale, ma raccomandabile per i repository che usano i set, serve a specificare una breve stringa utilizzata ad esempio per la visualizzazione del set consentendo eventualmente l'harvesting dei record attraverso il browsing dei set di interesse.

- *SetDescription* - questo elemento opzionale e ripetibile può contenere informazioni, codificate in XML, di comunità di repository, descrittive dei set adottati. Questo descrittore diviene di particolare utilità, quando i set rappresentano delle vere e proprie collezioni prestabilite di item opportunamente catalogati e riconosciuti dalle comunità, le quali possono così realizzare i propri schemi XML per validare le loro collezioni. In tal modo qualunque harvester visualizzando i set di un repository, potrà venire a conoscenza di ulteriori informazioni per offrire servizi di crescente utilità per i suoi utenti. Anche in questo caso l'OAI-PMH nulla impone circa le modalità di utilizzo di questo elemento. Una possibilità potrebbe essere quella di utilizzare un tag simmetrico (es. <description>)

per contenere uno o più tag (anch'essi simmetrici) <setDescription> descrittori degli item di una specifica collezione.

Il set a cui appartiene un item è dunque indicato da un elemento setSpec del record ad esso associato. La codifica XML può utilizzare un tag apposito per contenere una stringa rappresentante il nome univoco del set definito.

Dunque la rappresentazione XML dell'header di un record associato a Matematica e Fisica potrebbe essere:

```
<header>
    <identifier>itemDiMatematicaFisica</identifier>
    <setSpec>Scienze:Matematica</setSpec>
    <setSpec>Scienze:Fisica</setSpec>
    ...
</header>
```

dove, come si può notare, l'item rappresentato appartiene ai sottoset specificati dagli elementi setSpec relativi, ma ovviamente anche al set di livello superiore Scienze il quale non dovrà essere ulteriormente specificato.

2.8 SELECTIVE HARVESTING

Il selective harvesting rende possibile specificare un sottoinsieme di record da raccogliere attraverso una richiesta OAI-PMH.

Il protocollo specifica due semplici criteri per il selective harvesting che può essere così basato sui datestamp dei record oppure sui set a cui gli item possono appartenere.

2.8.1 Selective harvesting basato sul datestamp

L'harvesting selettivo basato sul datestamp consiste nella possibilità di poter limitare il numero di record in risposta ad una richiesta, attraverso l'utilizzo da parte dei Service Provider di due parametri: *from* e *until* che permettono di specificare il range di date entro il quale i record di risposta debbono ricadere.
Il parametro from indica il limite inferiore di tale range, mentre until il limite superiore.
Ovviamente specificando solo from saranno raccolti tutti i record a partire da un certo datestamp assegnatogli, mentre specificando solamente until saranno raccolti tutti i record fino al datestamp indicato.
Il fomato di rappresentazione delle date adottato è l'UTC che secondo l'ISO8601 supporta due tipi di granularità:

- **data completa**: rappresentata come YYYY-MM-DD (es: 2004-03-02)
- **data completa con ore, minuti e secondi**: rappresentata come YYYY-MM-DDThh:mm:ssZ (es: 2004-03-02 T11:49:00Z) che adotta i delimitatori T e Z

Ovviamente, ogni richiesta da parte dell'harvester dovrà indicare argomenti from e until con la stessa granularità.
Tutti i repository devono supportare almeno la granularità YYYY-MM-DD, mentre per i campi from e until devono essere supportate le forme YYYY-MM-DD, YYYY-MM e YYYY [B3]

2.8.2 Selective harvesting basato sui set

I set possono essere usati nelle richieste di harvesting per specificare che solo i record appartenenti ad uno specifico set debbono essere ritornati nella risposta da parte del repository.
Ogni item in un repository può appartenere ad uno, diversi o nessun set, ciò fa capire come effettuare raccolte di record basate solo sui set non implica necessariamente la raccolta di tutti.
Come si vedrà più avanti, il set in base al quale effettuare la raccolta dovrà essere specificato nell'argomento *set* della richiesta.
L'harvester può venire a conoscenza dei set implementati dal repository attraverso la richiesta *ListSets*,(vedi paragrafo 2.9.4.6) a patto che il repository stesso implementi i set e quindi l'interfaccia di esposizione degli stessi.

2.9 RICHIESTE OAI-PMH

Il protocollo per il metadata harvesting definisce sei richieste che il service provider può inviare al repository del data provider, il quale è tipicamente un webserver che si occupa di inviarle al software di gestione.

La figura 2 seguente mostra a grandi linee il processo di scambio tra data e service provider:

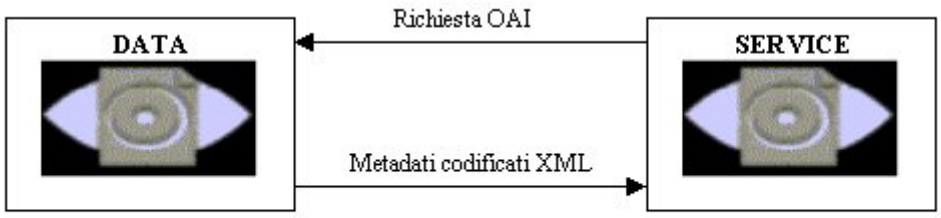

Figura 2 - Comunicazione tra Data e Service provider

Una richiesta OAI-PMH è espressa utilizzando il protocollo HTTP e può essere inviata scegliendo il metodo GET o POST.
Il data provider dovrà implementare l'interfaccia per le richieste in maniera tale da poterle riconoscere e gestire per entrambe i metodi.

Una richiesta sarà composta, indipendentemente dal metodo scelto, da :

- un **URL** di base che specifica l'host e la porta su cui il web server rimane in ascolto
- un parametro o argomento **verb** obbligatorio, il quale costituisce la coppia nome-valore che specifica una delle sei richieste inviata dall'harvester.
- ulteriori parametri aggiuntivi e non obbligatori per tutti i tipi di richiesta, che rappresentano anch'essi delle coppie nome-valore separate dal simbolo "&".

2.9.1 Metodo Get di richiesta

Con questo metodo le coppie nome-valore che rappresentano la richiesta vengono accodate all'URL del data provider a seguire il simbolo "?" e sono separate tra loro dal simbolo "&" come di seguito:

http://xXx.org/OAI-script?verb=ListIdentifiers
&from=2000-01-15
&metadataPrefix=oai_dc

in questo caso il nome o verbo della richiesta è ListIdentifiers ed i parametri aggiuntivi sono from e metadataPrefix il cui significato si specificherà in seguito al paragrafo 2.9.4.5.

2.9.2 Metodo Post di richiesta

A differenza del metodo Get, il Post trasporta i parametri che costituiscono la richiesta OAI-PMH nel corpo di un POST-MESSAGE inviato tramite richiesta HTTP. In questo caso il formato della richiesta sarà:

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

verb=ListIdentifiers&from=2000-01-15&metadataPrefix=oai_dc

Come si può notare, la richiesta costituita dai parametri viaggia nel corpo del messaggio ed è staccata dall'URL.

2.9.3 Parametro metadataPrefix

Come detto in precedenza, i data provider possono supportare diversi formati di metadati per la disseminazione dei record ed in particolare devono adottare il Dublin Core per garantire la soglia minima di interoperabilità.
Le richieste del protocollo inoltrate dagli harvester per la raccolta di uno o più record di metadati devono specificarne il formato, permettendo al repository del

data provider di selezionarlo tra quelli a disposizione. L'argomento "metadataPrefix" in una richiesta OAI-PMH permette proprio di specificare la coppia nome-valore rappresentante il formato del o dei record da raccolgliere.
Il valore da assegnare a metadataPrefix è una stringa che può essere composta utilizzando solamente i caratteri ammissibili per gli URI: valgono perciò le considerazioni fatte in precedenza per gli identificatori unici (vedi paragrafo 2.6).
Le richieste che necessitano di tale argomento sono quelle utilizzate per la raccolta dei record o degli header dei record.
Allo stato attuale del protocollo è possibile specificare un solo formato di metadato per richiesta, si ritiene che sviluppi futuri permetteranno l'assegnazione del valore "all" a metadataPrefix, consentendo quindi richieste per raccolte di formati multipli di record.
Le comunità di data provider devono adottare soluzioni per la condivisione delle stringhe che è possible assegnare a metadataPrefix; l'OAI-PMH non dà particolari indicazioni, ma auspica che tutti i data provider implementino l'interfaccia per la richiesta "ListMeatadataFormats", discussa più avanti (vedi paragrafo 2.9.4.3), per esporre i formati di metadati da loro supportati.

2.9.4 Verbi OAI-PMH

Le richieste del protocollo OAI-PMH vengono spesso chiamate "verbi" poiché specificate per mezzo di un argomento obbligatorio "verb" seguito dagli eventuali parametri aggiuntivi.
Di seguito sono esposti tutti e sei i possibili verbi di richiesta e i parametri che possono o devono essere specificati.

2.9.4.1 GetRecord

Questo verbo consente di recuperare un singolo record di metadati dal repository di un data provider.

La sintassi delle richiesta è la seguente:

http://xXx.org/OAI-script?verb=GetRecord
&identifier=oai:xXx.org:math-net/10121
&metadataPrefix=oai_dc

Gli argometi obbligatori da specificare nella richiesta sono:

- **identifier** – per indicare l'identificatore unico (Unique Identifier) dell'item tramite il quale il record può essere raccolto.
- **metadataPrefix** – per indicare il formato in cui il metadato, rappresentato nella parte metadata del record, deve essere restituito.

  Questo argomento permette di scegliere tra tutti i record contenuti in un item, quello con formato di metadato desiderato dall'harvester del service provider. Nel caso in cui il formato richiesto non sia più disponibile per l'item in questione o non supportato più dal repository potrà accadere una delle seguenti situazioni:

  - se supportate le cancellazioni di record, potrà essere ritornato l'header del record con relativo campo di stato "deleted"
  - altrimenti verrà restituito un errore ad indicare la totale mancanza di supporto per il formato richiesto.

Le possibili condizioni di errore a seguito della richiesta sono:

- *badArgument* – indica che la richiesta contiene argomenti illegali oppure uno o entrambi gli argomenti obbligatori **identifier** e metadataPrefix sono mancanti.
- *cannotDisseminateFormat* – indica che il formato specificato nell'argomento **metadataPrefix** non è supportato per l'item specificato nella richiesta.
- *idDoesNotExist* – indica che l'identificatore specificato nell'argomento **identifier** è sconosciuto per il repository del data provider al quale la richiesta è stata rivolta.

2.9.4.2 Identify

Questo verbo permette di ricavare informazioni in merito al repository del data provider al quale la richiesta viene inoltrata. Tutti i repository conformi al protocollo devono indicare alcune informazioni obbligatorie e possono decidere autonomamente se aggiungerne altre nella risposta.

La sintassi della richiesta è la seguente:

http://xXx.org/OAI-script?verb=Identify

La richiesta non prevede argomenti oltre all'unico obbligatorio **verb** per indicarne il tipo.

L'unica possibile condizione di errore a seguito della richiesta è:

- *badArgument* – indica che la richiesta contiene argomenti illegali e non necessari poiché l'unico richiesto è verb.

Le informazioni che, opportunamente codificate, devono essere restituite dal repository sono:

- *repositoryName* – una stringa indicante il nome del repository che allo stato attuale del protocollo non necessita di obbligatoria registrazione dall'OAI, anche se tale pratica è raccomandata per chi voglia pienamente aderire all'iniziativa.
- *baseURL* – l'URL che specifica l'host internet del repository del data provider
- *protocolVersion* – la versione del protocollo supportata dal repository es. v.1.0, v.1.1 oppure v.2.0.
- *earliestDatestamp* – rappresenta la data al di sotto della quale non possono essere i datestamp dei record del repository. In tal modo si garantisce un limite inferiore per la data di ultima modifica o cancellazione dei record, facendo così comprendere all'harvester che non potrà fare richiesta di record che vanno oltre tale limite. L'informazione dovrà essere rappresentata con la granularità più accurata supportata per le date.
- *deletedRecord* – indica se e come il repository supporta lo stato di cancellato per il corrispettivo campo facoltativo nella struttura dei record discussa precedentemente. Come visto infatti, i valori che l'elemento deleteRecord può assumere sono *no*, transient o *persistent*.
- *granularity* – indica la granularità per le date più accurata supportata dal repository del data provider. I valori possibili, come precedentemente visto, sono:

  YYYY-MM-DD

YYYY-MM-DDThh:mm:ssZ.

- *adminEmail* – indica l'indirizzo e-mail dell'amministratore del repository. Tale campo può essere ripetuto nel caso vi siano più responsabili.

La risposta può includere il seguente argomento opzionale e ripetibile:

- *compression* – indica un tipo di compressione supportata dal repository. Maggiori approfondimenti sono presenti nelle LineeGuida Data Provider al Cap. 3.

2.9.4.3 ListMetadataFormats

Questo verbo è utilizzato dal service provider per conoscere o i formati di metadati supportati dal repository del data provider oppure i formati disponibili per uno specifico item.
È importante osservare che:
il fatto che un repository possa supportare un numero N di formati, non implica che tutti gli item contengano N record. È cioè possibile che uno o più formati non siano disponibili per alcuni item del repository.

La sintassi della richiesta è la seguente:

http://xXx.org/OAI-script?verb=ListMetadataFormats
&identifier= oai:xXx.org:math-net/10121

o semplicemente

http://xXx.org/OAI-script?verb=ListMetadataFormats

La richiesta prevede un solo argomento opzionale:

- **identifier** – specifica l'identificatore unico dell'item del quale si vogliono conoscere i formati disponibili. Se omesso, la risposta fornirà tutti i formati di metadato supportati dal repository.

Le possibili condizioni di errore a seguito della richiesta sono:

- *badArgument* - indica che la richiesta contiene uno o più argomenti illegali.
- *idDoesNotExist* - indica che l'identificatore specificato nell'argomento identifier è sconosciuto per il repository del data provider al quale la richiesta è stata rivolta.
- *noMetadataFormats* – indica che non ci sono formati disponibili per l'item specificato.

2.9.4.4 ListRecords

Questo verbo è utilizzato per raccogliere più record dal repository del data provider e adoperando gli argomenti opzionali rispettivamente from e until, o set, è possibile realizzare una raccolta selettiva (selective harvesting) dei metadati basata sui datestamp o sui set.
Da notare come la raccolta selettiva può essere basata su entrambi i criteri.

La sintassi della richiesta è la seguente:

http://xXx.org/OAI-script?verb=ListRecords

&from=2001-04-15

&until=2004-03-26

& set=Scienze:Fisica

&metadataPrefix=oai_dc

oppure

http://xXx.org/OAI-script?

verb=ListRecords&resumptionToken='ccxb'

nel caso si utilizzi l'argomento resumptionToken.

L' unico argomento obbligatorio da specificare nella richiesta è:

- **metadataPrefix** – per indicare il formato in cui il metadato, rappresentato nella parte metadata dei record, deve essere restituito.
  Questo argomento permette all'harvester del service provider di scegliere tra tutti i record contenuti in un item, quelli con formato di metadato desiderato.
  Dipendentemente dal fatto che il repository del data provider supporti le cancellazioni di record, potranno essere restituiti uno o più header per i record con valore "deleted" del relativo campo di stato. Essi dunque sono stati cancellati e la relativa parte meadata non è più resa disponibile.

Gli argometi opzionali sono:

- **from** – che indica il limite inferiore per la data della raccolta selettiva dei record basata sul datestamp discussa in precedenza al paragrafo 2.8.1.
- **until** – che invece indica il limite superiore di quanto detto per l'argomento from precedente.
- **set** – che indica il set a cui devono appartenere i record da raccogliere. Tali set, come discusso nel paragrafo sui set (vedi paragrafo 2.7), sono

specificati nella parte header dei record attraverso il campo setSpec ad indicarne l'appartenenza e permettono di poter effettuare raccolte selettive basate su essi; come discusso nel paragrafo 2.8.2.

L'argomento "esclusivo" (poiché la sua presenza esclude tutti gli altri possibili argomenti eccetto quello obbligatorio verb) che è possibile utilizzare nella richiesta è:

- **resumptionToken** – a cui è associato il valore generato dal data provider a seguito del partizionamento della risposta di una ListRecords precedente come discusso nel paragrafo partizionamento della risposta e nel capitolo 3: Linee Guida Per il data provider.

Le possibili condizioni di errore a seguito della richiesta sono:

- *badArgument* - indica che la richiesta contiene uno o più argomenti illegali.
- *badResumptionToken* – indica che il valore precedentemente assegnato all'argomento resumptionToken non è valido oppure è terminato il tempo di validità dello stesso.
- *cannotDisseminateFormat* – indica che il valore assegnato a metadataPrefix è un formato non supportato dal repository del data provider.
- *noRecordsMatch* – indica che non esiste alcun record del repository corrispondente alla selezione effettuata tramite gli argomenti: from, until, set e metadataPrefix.
- *noSetHierarchy* – indica che il repository non supporta i set e quindi nessun set è stato definito per esso.

2.9.4.5 ListIdentifiers

Questo verbo funziona come ListRecords, ma in modo più restrittivo in quanto permette la raccolta dei soli header di record del repository tralasciando le parti metadata e about. Adoperando gli argomenti opzionali rispettivamente from e until, o set, è possibile realizzare una raccolta selettiva (selective harvesting) dei metadati basata sui datestamp o sui set.
Da notare come la raccolta selettiva può essere basata su entrambi i criteri.

La sintassi della richiesta è la seguente:

http://xXx.org/OAI-script?verb=ListIdentifiers
&from=2001-04-15
&until=2004-03-26
& set=Scienze:Fisica
&metadataPrefix=oai_dc

oppure

http://xXx.org/OAI-script?
verb=ListIdentifiers&resumptionToken='ccxb'

nel caso si utilizzi l'argomento resumptionToken.

L' unico argomento obbligatorio da specificare nella richiesta è:

- **metadataPrefix** – specifica che gli header dovrebbero essere ritornati solo se il formato dei record che li contengono corrisponde al valore di tale argomento. Dipendentemente dal fatto che il repository del data provider supporti le cancellazioni di record, potranno essere restituiti uno o più header per i record con valore "deleted" del relativo campo di stato. Essi

dunque sono stati cancellati e la relativa parte meadata non è più resa disponibile.

Gli argometi opzionali sono:

- **from** – che indica il limite inferiore per la data della raccolta selettiva degli header basata sul datestamp discussa in precedenza al paragrafo 2.8.1.
- **until** – che invece indica il limite superiore di quanto detto per l'argomento from precedente.
- **set** – che indica il set a cui devono appartenere i record degli header da raccogliere. Tali set, come discusso nel paragrafo sui set (vedi paragrafo 2.7), sono specificati nella parte header dei record attraverso il campo setSpec ad indicarne l'appartenenza e permettono di poter effettuare raccolte selettive basate su essi; come discusso nel paragrafo 2.8.2.

L'argomento "esclusivo" (poiché la sua presenza esclude tutti gli altri possibili argomenti eccetto quello obbligatorio verb) che è possibile utilizzare nella richiesta è:
resumptionToken – a cui è associato il valore generato dal data provider a seguito del partizionamento della risposta di una ListIdentifiers precedente come discusso nel paragrafo partizionamento della risposta e nel capitolo 3.

Le possibili condizioni di errore a seguito della richiesta sono:

- *badArgument* - indica che la richiesta contiene uno o più argomenti illegali.
- *badResumptionToken* – indica che il valore precedentemente assegnato all'argomento resumptionToken non è valido oppure è terminato il tempo

di validità dello stesso.

- *cannotDisseminateFormat* – indica che il valore assegnato a metadataPrefix è un formato non supportato dal repository del data provider.
- *noRecordsMatch* – indica che non esiste alcun record del repository corrispondente alla selezione effettuata tramite gli argomenti: from, until, set e metadataPrefix.
- *noSetHierarchy* – indica che il repository non supporta i set e quindi nessun set è stato definito per esso.

2.9.4.6 ListSets

Questo verbo fa si che l'harvester venga a conoscenza dei set implementati dal repository del data provider, a patto che il repository stesso implementi i set e quindi l'interfaccia di esposizione degli stessi. L'OAI-PMH pur non dando restrizioni sull'implementazione dei set, raccomanda ai data provider che li implementano di mettere a disposizione questa richiesta.

La sintassi della richiesta è la seguente:

http://xXx.org/OAI-script?verb=ListSets

oppure

http://xXx.org/OAI-script?verb=ListSets&resumptionToken='ccxb'

nel caso si utilizzi l'argomento resumptionToken.

La richiesta non prevede argomenti opzionali o obbligatori, ma l'unico argomento "esclusivo" (poiché la sua presenza esclude tutti gli altri possibili argomenti eccetto quello obbligatorio verb) che è possibile utilizzare è:

- **resumptionToken** – a cui è associato il valore generato dal data provider a seguito del partizionamento della risposta di una ListSets precedente come discusso nel paragrafo partizionamento della risposta e nel capitolo 3: Linee Guida Per il data provider.

Le possibili condizioni di errore a seguito della richiesta sono:

- *badArgument* - indica che la richiesta contiene uno o più argomenti illegali.

- *badResumptionToken* – indica che il valore precedentemente assegnato all'argomento resumptionToken non è valido oppure è terminato il tempo di validità dello stesso.
- *noSetHierarchy* – indica che il repository non supporta i set e quindi nessun set è stato definito per esso.

2.10 RISPOSTE OAI-PMH

Nel paragrafo 2.9 sono stati discussi i verbi e quindi le richieste del protocollo OAI-PMH. A seguito di tali richieste il protocollo prevede che il data provider restituisca all'harvester una risposta codificata nel formato XML.
Dunque, tutte le risposte definite nel protocollo consistono in corrispondenti documenti XML validabili attraverso uno schema XML opportunamente predisposto dall'OAI e reperibile al seguente indirizzo:
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd.
Le risposte OAI-PMH possono rappresentare risposte vere e proprie alle richieste del protocollo GetRecord, Identify, ListRecords, ListIdentifiers, ListSets e ListMetadataFormats, precedentemente viste, oppure condizioni d'errore o eccezione codificate in XML.
I paragrafi seguenti descrivono più in dettaglio quanto appena detto.

2.10.1 Formato di risposta HTTP

Come già accennato in precedenza, il protocollo OAI-PMH si basa, al livello transport, sul protocollo HTTP. Dunque le richieste e le risposte sono formattate secondo l'HTTP e con gli appropriati campi di intestazione.

Il content-type ritornato da tutte le richieste OAI-PMH dovrà essere text/xml.

Infine il protocollo consiglia di specificare i codici di stato HTTP indipendentemente dagli errori OAI-PMH (vedi paragrafo 2.10.9). In tal modo i repository devono conformarsi alle definizioni dei codici di stato presenti nelle specifiche del protocollo HTTP rfc 2616 [S5] e possono impiegarli in aggiunta al codice di stato *200 OK*.
Ad esempio si possono implementare i codici di stato seguenti:

- 302 – permette al repository di rediregere una richiesta OAI-PMH ad un altro repository.
- 503 – indica che il servizio non è disponibile e specifica un periodo di tempo prima del quale non bisogna ritentare.

2.10.2 Record nel formato XML

Precedentemente si è visto come un record di metadati deve essere strutturato secondo le specifiche del protocollo.
Di seguito viene presentata la relativa rappresentazione XML di tale record:

```
<header>
    <identifier>oai:xXx.org:cs/0112017</identifier>
    <datestamp>2002-02-28</datestamp>
    <setSpec>cs</setSpec>
    <setSpec>math</setSpec>
</header>
<metadata>
…
</metadata>
<about>
 <provenance
```

```
      xmlns="http://www.openarchives.org/OAI/2.0/provenance"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/provenanc
      e
      http://www.openarchives.org/OAI/2.0/provenance.xsd">
        <originDescription harvestDate="2002-02-02T14:10:02Z"
        altered="true">
       <baseURL>http://the.oa.org</baseURL>
       <identifier>oai:r2.org:klik001</identifier>
       <datestamp>2002-01-01</datestamp>
 <metadataNamespace>
 http://www.openarchives.org/OAI/2.0/oai_dc/
 </metadataNamespace>
 </originDescription>
 </provenance>
</about>
```

dove il record è diviso, come precedentemente detto al par.2.5, nelle tre parti header, metadata e about, rappresentate dai corrispondenti tag XML.

Le informazioni presenti nella parte header indicano che il record è associato ad un item il cui identificatore unico è **oai:xXx.org:cs/0112017** (tag identifier) e tale item appartiene ai set **cs** e **math** (tag setSpec). Infine il record è stato modificato l'ultima volta il 28-02-2002 (tag datestamp).

All'interno del tag corrispondente all'elemento metadata dovrebbe essere presente la codifica XML del metadato vero e proprio, ma poiché non rilevante per lo scopo dell'esempio (quello di vedere la struttura generale di un record) è stata volutamente omessa.

Infine, all'interno del campo opzionale about sono presenti diverse informazioni di utilità e di arricchimento del record stesso (non del metadato). Di particolare rilevanza sono le informazioni presenti all'interno dell'elemento

originDescription che indicano rispettivamente la base URL del data provider da cui il record è stato raccolto, il datestamp e l'identificatore unico che il record possedeva all'interno del repository prima di essere stato raccolto.

2.10.3 Risposta alla richiesta GetRecord

Viene di seguito rappresentata la risposta, codificata in XML, al verbo GetRecord visto al paragrafo 2.9.4.1:

```
<?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>2002-02-08T08:55:46Z</responseDate>
  <request verb="GetRecord" identifier="oai:arXiv.org:cs/0112017"
     metadataPrefix="oai_dc">http://arXiv.org/oai2</request>
  <GetRecord>
  <record>
    <header>
      <identifier>oai:arXiv.org:cs/0112017</identifier>
      <datestamp>2001-12-14</datestamp>
      <setSpec>cs</setSpec>
      <setSpec>math</setSpec>
    </header>
    <metadata>
       <oai_dc:dc
         xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_d
         c/
          http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
          <dc:title>Using Structural Metadata to Localize Experience
          of
```

```
          Digital Content</dc:title>
          <dc:creator>Dushay, Naomi</dc:creator>
          <dc:subject>Digital Libraries</dc:subject>
          <dc:description>With the increasing technical
                         sophistication of
                         both information consumers and providers,
                         there is
                         increasing demand for more meaningful
                         experiences of digital
                         information. We present a framework that
                         separates digital
                         object experience, or rendering, from
                         digital object storage
                         and manipulation, so the rendering can be
                         tailored to
                         particular communities of users.
          </dc:description>
          <dc:description>Comment: 23 pages including 2
             appendices,
            8 figures</dc:description>
          <dc:date>2001-12-14</dc:date>
         </oai_dc:dc>
    </metadata>
</record>
</GetRecord>
</OAI-PMH>
```

Tale esempio costituisce il documento XML completo di risposta contenente la parte della dichiarazione XML e dello spazio dei nomi [B3]

2.10.4 Risposta alla richiesta Identify

Viene di seguito rappresentata la risposta, codificata in XML, al verbo Identify visto al paragrafo 2.9.4.2:

```
<?xml version="1.0" encoding="UTF-8"?>
…
<!—Qui la parte relativa alla dichiarazione dello spazio dei nomi,
presente al paragrafo 2.10.3 è stata volutamente omessa-->
<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/eprin
        ts
        http://www.openarchives.org/OAI/1.1/eprints.xsd">
       <content>
         <URL>http://memory.loc.gov/ammem/oamh/lcoa1_content.html</UR
         L>
         <text>Selected collections from American Memory at the
         Library
         of Congress</text>
       </content>
       <metadataPolicy>
       ...
       <metadataPolicy/>
       <dataPolicy>
       ...
       <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/frien
       ds/
       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>
```
…

2.10.5 Risposta alla richiesta ListMetadataFormats

Viene di seguito rappresentata la risposta, codificata in XML, al verbo GetRecord visto al paragrafo 2.9.4.1:

```
<?xml version="1.0" encoding="UTF-8"?>
…
  <ListMetadataFormats>
   <metadataFormat>
     <metadataPrefix>oai_dc</metadataPrefix>
     <schema>http://www.openarchives.org/OAI/2.0/oai_dc.xsd
     </schema>
     <metadataNamespace>
       http://www.openarchives.org/OAI/2.0/oai_dc/
     </metadataNamespace>
   </metadataFormat>
   <metadataFormat>
     <metadataPrefix>olac</metadataPrefix>
     <schema>
       http://www.language-archives.org/OLAC/olac-0.2.xsd
     </schema>
     <metadataNamespace>http://www.language-archives.org/OLAC/0.2/
     </metadataNamespace>
   </metadataFormat>
   <metadataFormat>
     <metadataPrefix>perseus</metadataPrefix>
     <schema>http://www.perseus.tufts.edu/persmeta.xsd</schema>
     <metadataNamespace>http://www.perseus.tufts.edu/persmeta.dtd
     </metadataNamespace>
   </metadataFormat>
 </ListMetadataFormats>
…
```

2.10.6 Risposta alla richiesta ListRecords

Viene di seguito rappresentata la risposta, codificata in XML, al verbo GetRecord visto al paragrafo 2.9.4.1:

```
<?xml version="1.0" encoding="UTF-8"?>
…
 <ListRecords>
  <record>
    <header>
      <identifier>oai:arXiv.org:hep-th/9901001</identifier>
      <datestamp>1999-12-25</datestamp>
      <setSpec>physics:hep</setSpec>
      <setSpec>math</setSpec>
    </header>
    <metadata>
    …
    </metadata>
    <about>
      <oai_dc:dc
        xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc
        /
        http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
        <dc:publisher>Los Alamos arXiv</dc:publisher>
        <dc:rights>Metadata may be used without restrictions as long
        as
        the oai identifier remains attached to it.</dc:rights>
      </oai_dc:dc>
    </about>
  </record>
  <record>
    <header status="deleted">
      <identifier>oai:arXiv.org:hep-th/9901007</identifier>
      <datestamp>1999-12-21</datestamp>
    </header>
  </record>
```

```
 </ListRecords>
…
```

2.10.7 Risposta alla richiesta ListIdentifiers

Viene di seguito rappresentata la risposta, codificata in XML, al verbo GetRecord visto al paragrafo 2.9.4.1:

```
<?xml version="1.0" encoding="UTF-8"?>
…
  <ListIdentifiers>
     <header>
         <identifier>oai:arXiv.org:hep-th/9801001</identifier>
         <datestamp>1999-02-23</datestamp>
         <setSpec>physic:hep</setSpec>
     </header>
     <header>
         <identifier>oai:arXiv.org:hep-th/9801002</identifier>
         <datestamp>1999-03-20</datestamp>
         <setSpec>physic:hep</setSpec>
         <setSpec>physic:exp</setSpec>
     </header>
     <header>
         <identifier>oai:arXiv.org:hep-th/9801005</identifier>
         <datestamp>2000-01-18</datestamp>
         <setSpec>physic:hep</setSpec>
     </header>
         <header status="deleted">
         <identifier>oai:arXiv.org:hep-th/9801010</identifier>
         <datestamp>1999-02-23</datestamp>
         <setSpec>physic:hep</setSpec>
         <setSpec>math</setSpec>
     </header>
     <resumptionToken expirationDate="2002-06-01T23:20:00Z"
        completeListSize="6"
        cursor="0">xxx45abttyz</resumptionToken>
```

```
  </ListIdentifiers>
…
```

2.10.8 Risposta alla richiesta ListSets

Viene di seguito rappresentata la risposta, codificata in XML, al verbo GetRecord visto al paragrafo 2.9.4.1:

```
<?xml version="1.0" encoding="UTF-8"?>
…
 <ListSets>
  <set>
    <setSpec>music</setSpec>
    <setName>Music collection</setName>
  </set>
  <set>
    <setSpec>music:(muzak)</setSpec>
    <setName>Muzak collection</setName>
  </set>
  <set>
    <setSpec>music:(elec)</setSpec>
    <setName>Electronic Music Collection</setName>
    <setDescription>
      <oai_dc:dc
       xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
       xmlns:dc="http://purl.org/dc/elements/1.1/"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/

       http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
       <dc:description>This set contains metadata describing
        electronic music recordings made during the 1950ies
       </dc:description>
      </oai_dc:dc>
    </setDescription>
  </set>
```

```
  <set>
    <setSpec>video</setSpec>
    <setName>Video Collection</setName>
  </set>
 </ListSets>
```
…

2.10.9 Condizioni di errore OAI-PMH

L'OAI-PMH richiede esplicitamente che i repository indichino e quindi implementino gli errori relativi ad eventi particolari che possono avvenire a seguito delle richieste inviategli. Questi errori sono indipendenti dai codici di stato discussi nel paragrafo 2.10.
La rappresentazione deve essere, anche per gli errori, nel formato XML della risposta e l'elemento *error* della stessa avrà un attributo *code* per indicare un codice tra quelli prestabiliti dal protocollo; infine l'elemento error può contenere fra i suoi tag una stringa di testo descrittivo il cui contenuto è scelto liberamente dal data provider dato che l'OAI-PMH non dà alcuna limitazione in tal senso.

Di seguito una tabella descrittiva delle otto condizioni di errore previste dal protocollo:

Codice errore	Descrizione	Richieste interessate
BadArgument	La richiesta inoltrata include argomenti illegali, o mancano argomenti obbligatori, o c'è un argomento ripetuto, oppure i valori assegnati agli argomenti seguono una sintassi scorretta.	Tutte
badResumptionToken	L'argomento resumptionToken ha un argomento non valido o oltre la data di scadenza (expiration date).	ListIdentifiers ListRecords ListSets
BadVerb	Il valore dell'argomento verb non corrisponde ad uno dei sei stabiliti dal protocollo, o l'argomento verb è mancante oppure l'argomento verb è ripetuto.	Tutte con verb errato, mancante o ripetuto
cannotDisseminateFormat	Il formato di metadato indicato come valore dell'argomento metadataPrefix non è supportato per l'item specificato o in generale dal repository del data provider.	GetRecord ListIdentifiers ListRecords
idDoesNotExist	Il valore dell'argomento identifier è sconosciuto o illegale per il repository.	GetRecord ListMetadataFormats
noRecordsMatch	Non esiste alcun record del repository corrispondente alla selezione effettuata tramite gli argomenti: from, until, set e metadataPrefix.	ListIdentifiers ListRecords
noMetadataFormats	Non ci sono formati disponibili per l'item specificato nella richiesta.	ListMetadataFormats
noSetHierarchy	Il repository non supporta i set	ListSets ListIdentifiers ListRecords

Di seguito due esempi di richieste errate con relativa gestione per la risposta:

Esempio 1

http://xXx.org/OAI-script?verb=VerboNonOaiCompatibile

La richiesta in questo caso contiene un valore per l'argomento verb non compatibile con il protocollo.
Nella risposta seguente si omettono alcune parti (definizione namespace) già discusse nel pragrafo 2.10.

```
<?xml version="1.0" encoding="UTF-8"?>
…
  <responseDate>2004-04-01T12:52:29Z</responseDate>
  <request>http://xXx.org/OAI-script</request>
  <error code="badVerb">Questo non è un verbo OAI compatibile</error>
…
```

Esempio 2

http://xXx.org/OAI-script?verb=ListSets

```
<?xml version="1.0" encoding="UTF-8"?>
…
 <responseDate>2002-05-01T09:18:29Z</responseDate>
 <request verb="ListSets">http://xXx.org/OAI-script</request>
 <error code="noSetHierarchy">Questo repository non supporta i
 Set</error>
...
```

In questo caso l'elemento error della risposta evidenzia, con una stringa di testo libero, che i set non sono supportati dal repository.

2.10.10 Partizionamento della risposta

Le richieste ListRecords, ListIdentifiers, ListSets del protocollo OAI-PMH ritornano delle liste di elementi; rispettivamente record, header e set. Esse vengono classificate come *richieste di lista* (list requests).
In questi casi quindi, la risposta da parte del repository del data provider è costituita da un elenco di elementi che potrebbe essere troppo grande da restituire in una sola volta a chi ha inoltrato la richiesta. Il data provider allora può decidere di dilazionare tale elenco attraverso una serie di richieste e risposte in modo da non sovraccaricare il proprio repository. Tale soluzione viene detta "partizionamento della risposta".

Il partizionamento utilizza i concetti di "lista incompleta" (incomplete list) e *resumptionToken*.
Il numero massimo di record che il repository può ritornare a seguito di una delle richieste d'elenco è stabilito a priori. Ammettiamo che una richiesta ritorni un numero di record che superi questo massimo consentito, in tal caso il repository ritornerà come risposta una lista incompleta di record esattamente del numero massimo prestabilito; in più ritornerà un resumptionToken, ossia il valore che l'harvester potrà assegnare all'argomento relativo della prossima richiesta per ottenere una successiva lista incompleta di record. Questo flusso di richieste e risposte di liste incomplete prosegue fino a che il repository non arriva all'ultima disponibile. In questo caso allora l'harvester capirà che quella è l'ultima risposta poiché gli verrà restituito un elemento resumptionToken vuoto.

L'esempio seguente mostra il caso di una richiesta gestita con il partizionamento:
l'harvester di un service provider fa una richiesta ListRecords al repository del data provider di tutti i record a partire dal 28-03-2004 con formato Dubline Core. Il repository ha un totale di 267 record che rispondono ai requisiti di data e formato, ma ne può ritornare solo 100 per volta.

La figura 3 seguente mostra le sequenze della transazione tra data e service provider:

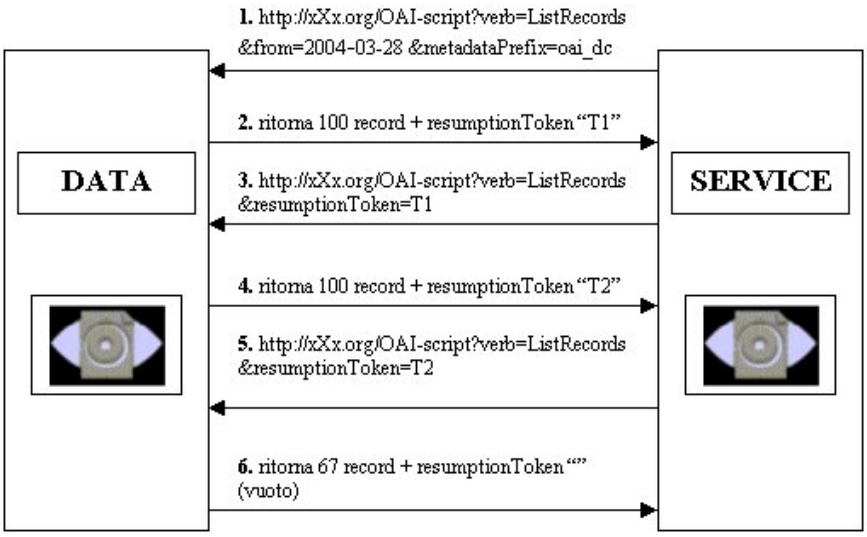

Figura 3 - Transazione tra Data e Service provider con partizionamento della risposta

dove:

1. il service provider invia una richiesta di tipo ListRecords al data provider, chiedendo di inviargli tutti i record a partire dal 28 Marzo 2004 e nel formato Dublin Core
2. il data provider, che adotta il meccanismo del partizionamento, invia solo 100 record al service provider ed un elemento resumption token con valore "T1" per far capire che ci sono ancora record da raccogliere

3. il service provider invia una seconda richiesta ListRecords, specificando il parametro resumptionToken con valore “T1”
4. il data provider invia altri 100 record ed un nuovo resumption token con valore “T2”
5. il service provider invia una seconda richiesta ListRecords, specificando il parametro resumptionToken con valore “T2”
6. il data provider ritorna 67 record e, questa volta, un resumption token vuoto (valore “”), ad indicare che i record della risposta sono terminati

BIBLIOGRAFIA

[B3] – Buttà Basilio, *Metadati negli Open Archive: tecniche di conversione dei formati dei record nel formato MARC21 utilizzato da CDSware*, tesi di laurea in informatica, Università degli Studi di Messina, A.A. 2003-2004 (relatore Puccio L., correlatore De Robbio A.).

SITOGRAFIA

[S2] - http://www.openarchives.org/OAI/openarchivesprotocol.html

(Documento ufficiale della versione 2.0 del protocollo OAI-PMH di raccolta dei metadati)

[S5] - http://www.ietf.org/rfc/rfc2616.txt

(RFC (Request For Comment) che presenta le specifiche tecniche del protocollo HTTP 1.1)