A Digital Resource Harvesting Approach for Distributed Heterogeneous Repositories

Yang Zhao and Airong Jiang

Tsinghua University Library
Beijing, China 100084
{zhaoyang, jiangar}@lib.tsinghua.edu.cn

Abstract. OAI-PMH has been widely adopted as a simple solution for harvesting the metadata of different repositories automatically. Harvesting digital resources described by the metadata is outside of the scope of the OAI-PMH data model. However, there are some growing needs to make resources, not only metadata, harvestable by an interoperable manner in the distributed heterogeneous environments. In this paper, we present the new approach of digital resource harvesting, which uses Message Queue-based communication mechanism as the datastream transfer method, and ensures the request and response message specification built on METS during the course of data transfer. The approach can harvest digital resources solely or synergically with OAI-PMH. A case study about this approach applied in CALIS_ETD digital library will be introduced in the end.

Keywords: OAI-PMH; Message Queue; Digital Resource Harvesting; METS.

1 Introduction

The Open Archives Protocol for Metadata Harvesting (OAI-PMH) has been widely adopted as a simple and powerful solution for metadata harvesting. There are many digital library systems and projects to use OAI-PMH to harvest metadata held by different repositories into central systems as a basis for building the value-added services, e.g., NDLTD, OAIster, NSDL, arXiv, etc. However, there are some growing needs to make resources, not only metadata, harvestable by an interoperable manner in the distributed heterogeneous environments. These needs are motivated by two major use cases. One is mainly for resource discovery in order to use content itself for providing value-added services of central systems, such as making full-text from different repositories searchable, or building browsing interfaces of high-quality thumbnail images. Another is mainly for resource preservation in central systems, such as harvesting digital contents from different repositories to the trusted central systems charged with storing and preserving safety copies of the contents. Both use cases have been discussed in the context of digital library projects, such as JISC FAIR in UK, DARE in Netherlands, DINI in Germany, NDIIPP in USA and so on [1].

Although OAI-PMH does not say anything about how to harvest digital resources described by the metadata, resource harvesting is associated with metadata harvesting

S. Sugimoto et al. (Eds.): ICADL 2006, LNCS 4312, pp. 71 – 80, 2006.

to some extent. For example, we can harvest resources according to the returned information of metadata harvesting or expand the scope of descriptive metadata to be more than just DC and similar bibliographic formats in order to be compatible with existing OAI-supported repositories. In nature, resource harvesting is more complex than metadata harvesting because of the complexity of digital resources that include many kinds of digital file formats (i.e., PDFs, GIFs, TIFFs, AVIs, etc.) or the ordered combination of many files (such as an E-book is made of many TIFF files).

There are some existing methods for indirectly harvesting digital resources. For example, the reference [1] puts forward to harvest resources within the OAI-PMH framework by means of complex and expressive metadata formats (i.e. SCORM, MPEG-21 DIDL or METS, etc.) to represent digital objects by embedding a base64 encoding or the network location of digital resources inside the wrapper XML document. However, it is difficult for a simple HTTP-based request-response to solve large datastream harvesting or transfer failures caused by the network congestion. References [2-3] emphasize on harvesting the network location of digital resources within the DC metadata record by some DC elements such as dc.format, dc.relation or dc.identifier. A separate process outside the scope of OAI-PMH collects the described resources from their network location. But this method does not provide the general mechanism for describing and gathering resources from their network location.

In order to make digital resource harvesting general and compatible with widely deployed OAI-supported repositories, and tackle many complicated problems caused by resource harvesting, we put forward an alternative approach to harvest digital resources in this paper. The proposed approach is based on METS (Metadata Encoding and Transmission Standard) that possesses sufficiently rigorous semantics to unambiguously express and describe both simple digital objects (consisting of a single datastream) and compound digital objects (consisting of multiple datastreams), which represent digital resources from different repositories. And the approach also discusses request-response communication mechanism based on MQ (Message Queue) between Data Providers (DPs) and Service Providers (SPs) for improving the security and efficiency of datastream exchange and transfer. The rest of the paper is as follows: Section 2 gives an overview of the approach. Section 3 introduces the implementation of the approach in the CALIS-ETD digital library. Section 4 gives the conclusion and future works.

2 An Overview of the Approach About Digital Resource Harvesting

Digital resource harvesting is concerned with some complicated problems, such as how to describe compound digital objects and their relationship, how to deal with transfer of large datastream representing digital resources and so on. Considering such complexity, the proposed approach discusses some key points about resource harvesting. We use MQ as the data transfer method in order to improve safety of data transfer and solve large datastream transfer. Also we ensure the message content specification during data transfer by MQ mechanism, which includes request message specification based on 5 verbs defined by CALIS (the China Academic Library & Information System) technology workgroup, and response message specification built on METS as complex object formats for accurately describing digital resources.

Like the OAI-PMH, we define two classes of participants including data providers (DPs) and service providers (SPs). DPs administer repositories that exposing digital resources. SPs harvest digital resources from DPS as a basis of building value-added services. MQ (Message Queue) applications are deployed and implemented between SPs and DPs in order to provide the two-way communication between them. Figure1 shows the architecture of digital resource harvesting. 1) When SP sends messages (request for resource harvesting) to DP, the MQ manger in DP puts request messages on message queue. 2) The message processing thread in DP is called whenever there is a new message on message queue, receives messages from the top of queue and orderly processes messages on queue one at a time.3) According to request messages, thread creates response message queue by communicating with metadata repository and digital object server in DP. Each response message consists of a XML document format based on METS encoding schema that can describe and encapsulate digital resources and theirs metadata. 4) Finally, the message processing thread in SP read response messages and put digital resources and their metadata harvested into central metadata repository and digital object server. In the approach, message reception and message processing are decoupled and receiving a message takes very little time, even when processing the message may take significant time. This improves application responsiveness and guarantees that all messages are received.

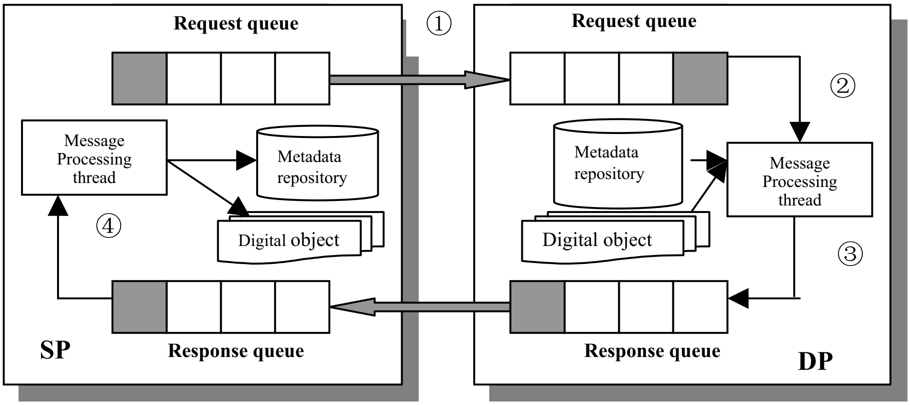


Fig. 1. Architecture of digital resource harvesting

2.1 Message Queue Transfer Mechanism

The message queue is reliable and asynchronous communication technology that enables applications on different systems to communicate with each other. With the message queue middleware software (such as OpenJMS or MSMQ), the process of building message queue applications between senders and receivers of message is simple and convenient. The application in senders uses the open API to create message on queue by allocating local memory and adding information to the message, such as timeout values, name of response queues and destination queues etc. The messages are sent through open API. The application in receivers uses the open API with queue identification information to receive and handle messages. MQ technology has following main features:

1) Synchronous or asynchronous communication: The applications can send request messages whether the receiving systems are available or not.
2) Reliable message transfer: MQ enables applications on different systems to communicate with each other even if systems and networks occasionally fail. MQ, using disk-based storage mechanisms and log-based recovery techniques, can ensure that messages get delivered as soon as connections are restored or applications and machines are restarted.
3) Advantages over transferring very large message body.

From the above analysis, we think that MQ is a reliable and easy-deployed message transfer mechanism. The applications on different systems can conveniently realize the communication based on messages by the open API of the MQ middleware software. MQ mechanism is a very good solution to solve the complexity of content transfer of digital resources caused by large datastream size or network failures and so on. In order to realize the standardization and interoperability of digital resource harvesting, we need to regulate and specify message content on the queue messages.

2.2 The Message Content Specification in Message Queue

A message is a unit of information or data that is sent from a process running on one computer (e.g., SP) to other processes running on the different computers (e.g., DPs) on the network. A message consists of header, properties and body. The message header contains values used for routing and identifying messages. The message properties provide additional information about data sent between SPs and DPs, for example, which processes create it, the time it is created etc. The message body contains data content of communications. In the approach, the message body mainly includes request and response content of resource harvesting between DPs and SPs. We specify and standardize content of the message body in the architecture of digital resources harvesting.

5 verbs were defined by CALIS for request of digital resource harvesting in message body. According to request, response content in message body is XML datastream based on METS schema to describe digital resources and their metadata. Table1 lists the functions of 5 verbs and their relationship with OAI-PMH, which can harvest digital resources solely or synergically with OAI-PMH. The former 3 verbs will cooperate with OAI metadata harvesting. Harvester of digital resources in SPs create request according to the information that OAI harvesters return, such as metadata datestamp or MetaID. The latter 2 verbs will lonely establish request of resource harvesting by digital object's ObjID or datastamp, and do not need to cooperate with harvesting based on OAI [4]. A MetaID can uniquely identify a digital resource and may include several ObjIDs, each of which represents the different files consisting of

a digital resource. For example, a scanned E-book has a MetaID and many ObjIDs to represent JPG files corresponding to each page of an E-book. For existing OAI framework, we only need add module of digital resource harvesting, rather than modify the existing OAI repository deployment.

Figure 2 demonstrates the request and response content in message body. The request in SP will send one of the 5 verbs to DP according to requirement of harvesting. DP processes request and send response messages to SP, which will handle harvested METS document. METS provides the expressive and accurate mechanism for representing both simple digital objects or compound digital objects, describing a variety of information pertaining to the datastream, such as descriptive, administrative and structural metadata, etc, and containing datastream by value embedding a base64 encoding of datastream or by reference embedding the network location of datastream inside the wapper XML document. So message response based on METS can provide the useful standard for harvesting and gathering of digital objects between DP and SP [5].

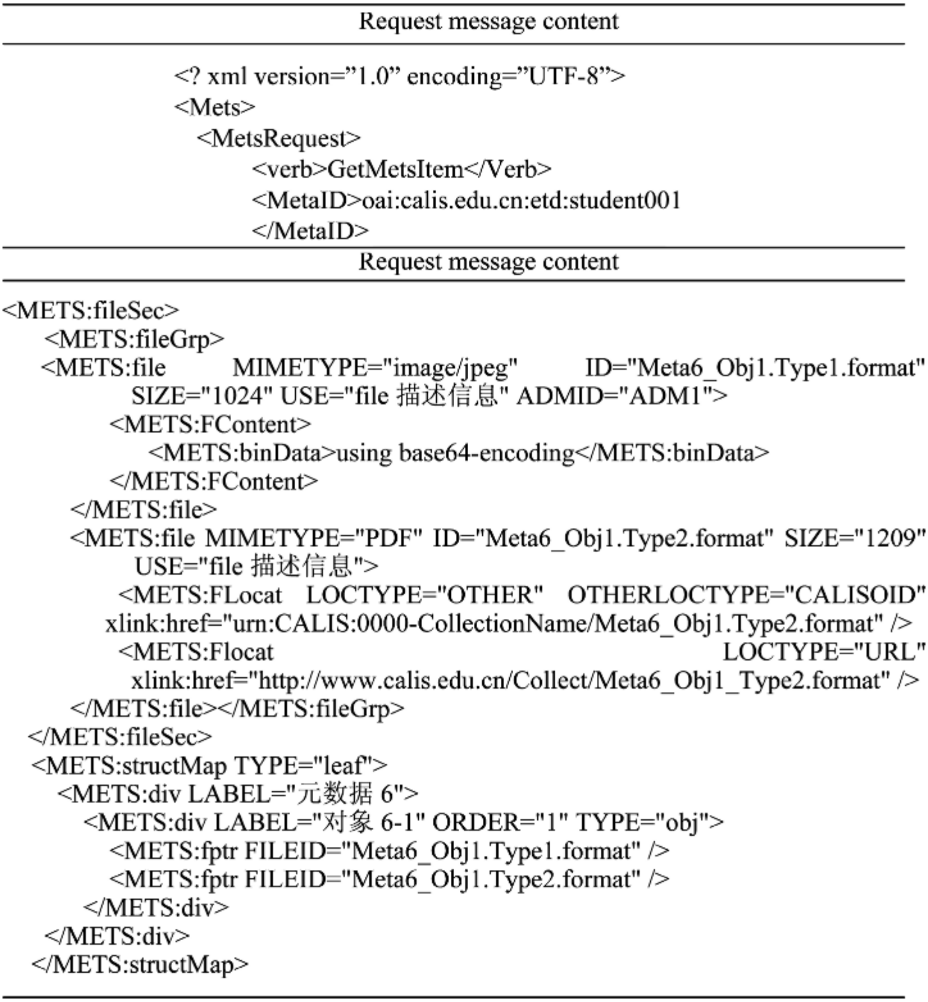
Table 1. 5 verbs for request message of digital resource harvesting

Verbs	Description of functions	Relationship with OAI
GetMetsItem	Get a digital object according to a metadata MetaID	Cooperating with OAI harvesting
GetMetsItems	Get a set of digital objects according to a set of metadata MetaIDs	
GetMetsItemByDate	Get a set of digital objects according to the specified datestamp bound	
GetObjMetsItem	Get a digital object according to a digital object's ObjID	Independently finishing harvesting tasks
GetObjMetsItems	Get a set of digital objects according to a set of digital object's ObjIDs	

3 A Case Study: Application of the Proposed Approach in CALIS-ETD Digital Library

3.1 General Information About CALIS-ETD Digital Library [6]

CALIS-ETD digital library is national digital library project funded by CALIS, and aims at making the electronic thesis and dissertation (ETD) resources become more readily and more completely available and speeding up technology and knowledge sharing. It is a distributed digital library system that consists of central CALIS-ETD system as SP and ETD resource repositories as DPs distributed in the member universities. The central CALIS-ETD system will centrally manage the ETD metadata or digital resources related to ETDs (such as first 16 pages of full-text ETDs, some technical datasheet, audio or video of ETDs, etc.) harvested from member universities. We cannot harvest full-text ETDs into central system because of the copyright restriction. The full-text search engine in the central system can abstract the index of ETDs (such as first 16 pages of full-text ETDs, etc.) harvested and enable them and metadata to be searchable. By the OPENURL or URN resolver, users in the central system can link obtained records to their corresponding full-text ETDs in member universities, whose access right is respectively controlled by each member university. We use the proposed approach based on the MQ transfer mechanism and message content specification based on METS to realize ETDs harvesting between the central system and 77 member universities of CALIS-ETD project.



Request message content

```
<? xml version=”1.0” encoding=”UTF-8”>
<Mets>
  <MetsRequest>
        <verb>GetMetsItem</Verb>
        <MetaID>oai:calis.edu.cn:etd:student001
        </MetaID>
```

Request message content

```
<METS:fileSec>
    <METS:fileGrp>
    <METS:file       MIMETYPE="image/jpeg"       ID="Meta6_Obj1.Type1.format"
            SIZE="1024" USE="file 描述信息" ADMID="ADM1">
          <METS:FContent>
              <METS:binData>using base64-encoding</METS:binData>
          </METS:FContent>
      </METS:file>
      <METS:file MIMETYPE="PDF" ID="Meta6_Obj1.Type2.format" SIZE="1209"
            USE="file 描述信息">
           <METS:FLocat   LOCTYPE="OTHER"   OTHERLOCTYPE="CALISOID"
          xlink:href="urn:CALIS:0000-CollectionName/Meta6_Obj1.Type2.format" />
           <METS:Flocat                                        LOCTYPE="URL"
            xlink:href="http://www.calis.edu.cn/Collect/Meta6_Obj1_Type2.format" />
      </METS:file></METS:fileGrp>
   </METS:fileSec>
   <METS:structMap TYPE="leaf">
      <METS:div LABEL="元数据 6">
        <METS:div LABEL="对象 6-1" ORDER="1" TYPE="obj">
          <METS:fptr FILEID="Meta6_Obj1.Type1.format" />
          <METS:fptr FILEID="Meta6_Obj1.Type2.format" />
        </METS:div>
     </METS:div>
   </METS:structMap>
```

Fig. 2. The request and response content in message body

3.2 Module Design for Digital Resource Harvesting in Member Universities as DPs

To make ETD repositories in the member universities support resource harvesting, it is necessary to add the module of resource harvesting in order to support the message queue and METS message response mechanism. Figure 3 shows the main module for accomplishing functions of resource harvesting as DPs.

1) The listener thread will listen the message request queue from harvesters, resolve the request and put MetaIDs into the MetaID global List container in the memory queue manager. MetaID global List container will orderly get a MetaID from list and submit the MetaID to METS package creating thread, which transfer MetaID to the interface of repositories (databases) and digital object servers.

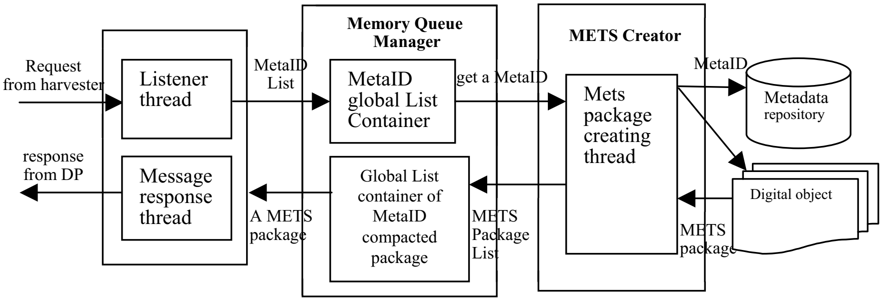


Fig. 3. The main module of digital resource harvesting as DP

2) METS package creating thread will create METS encoding datastream according to the returned information from repositories and digital object servers, which will be compacted for lessening the size of datastream. Each of compacted package of METS encoding datastream will be put in the global List container of MetaID compacted package in the memory queue manager. The message response thread will orderly send the METS compacted package representing digital resources as response messages to SP [4].

Most of member universities choose one of four types of commercial ETD repository systems that the project recommends as their local ETD repositories, which not only need to finish basic functions of managing ETDs, such as submitting, checking, cataloging, searching ETDs, etc, but also need to support OAI-PMH for metadata harvesting and the proposed approach for resource harvesting. For existing earlier ETD repositories that does not support resource harvesting, it is convenient to upgrade them to realize resource harvesting by installing plug-in because module of resource harvesting is independently designed and deployed [6].

3.3 Harvester Design in the Central System as SPs

The central system of CALIS_ETD digital library is developed in Java, using JDBC for database connectivity to ORACLE data source. The web interface is accomplished using Java servlets. Figure 4 shows main module of harvester. The storage layer centrally manages metadata and digital resources harvested from repositories in member universities by ORACLE Databases and digital object servers. The business logic layer is the core of harvester, which including several modules. DP registration module is charged with managing and maintaining the registration information of DPs in the member universities. Log module records log information of harvesting, such as how many records are harvested in certain time bound, what errors happened about harvesting, etc. The schema check module will examine correctness of METS packages, which will be put into the database by the digital object storage management module if passing the check. There are three methods to finish harvesting tasks according to the requirement. Message Queue server along with message queue middleware software (such as MSMQ or JMS MQ) will lonely finish digital resource harvesting. OAI-driven interface will work together with OAI harvester and finish resource harvesting according to returned information from OAI harvester. FTP service interfaces as the supplementary method will be used when harvester in SP and digital resource harvesting module in DPs cannot connect or digital resource harvesting module of DP breaks down and so on. The representation logic layer will realize management and configuration of harvester by the web interface.

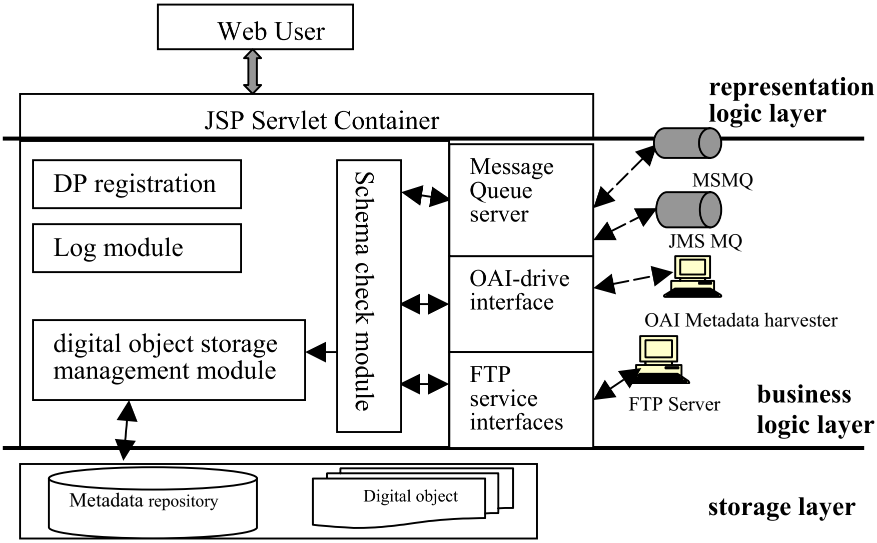


Fig. 4. The main module of harvester

The web configuration interface of harvester for creating tasks of resource harvesting is shown in Figure 5. The window of interface is divided into three parts. In the upper part, we can configure the time bound of harvesting tasks, automatic setup time of harvest tasks**,** running counts of each harvest task or timeout bound of harvesting. In the middle part, we may configure information related to DPs, such as repository names, IP addresses or service ports of DPs, etc. The lower part focuses on the information associated with MQ configuration, such as message queue name, MQ type and port of MQ service, etc.

3.4 Performance Evaluation of Harvesting

We take two phases to test the proposed approach and improve performance of resource harvesting. In the first phases, the main aim is to test feasibility of the approach by test programs to test prototype systems designed according to the proposed approach. The test uses the metadata and digital objects conforming to the specification required by the project, and is limited within small areas, which can ignore influence of network congestion or interruption. During the test, the approach is gradually amended and improved. In the second phases, we firstly choose about 20 member universities with better experience of managing ETDs and upgrade their existing ETD repositories or installing new ETD systems for supporting resource harvesting. Within two months, we have successfully harvested ETDs up to 60,000 records from repositories of about 20 member universities. The proposed approach is proved to be feasible. Of course, many problems are encountered during the course of real harvesting, such as unexpected interruption of harvest tasks or timeout error, poor data quality for lacking of better data validation mechanism, performance of harvesting partly influenced by capability of ETD repositories in DPs, frequent backend Oracle database down caused by synchronous running multi-tasks for harvesting, etc. So we need to gradually improve the performance of harvester in SP and ETD repositories in DPs according to problems that we have encountered.

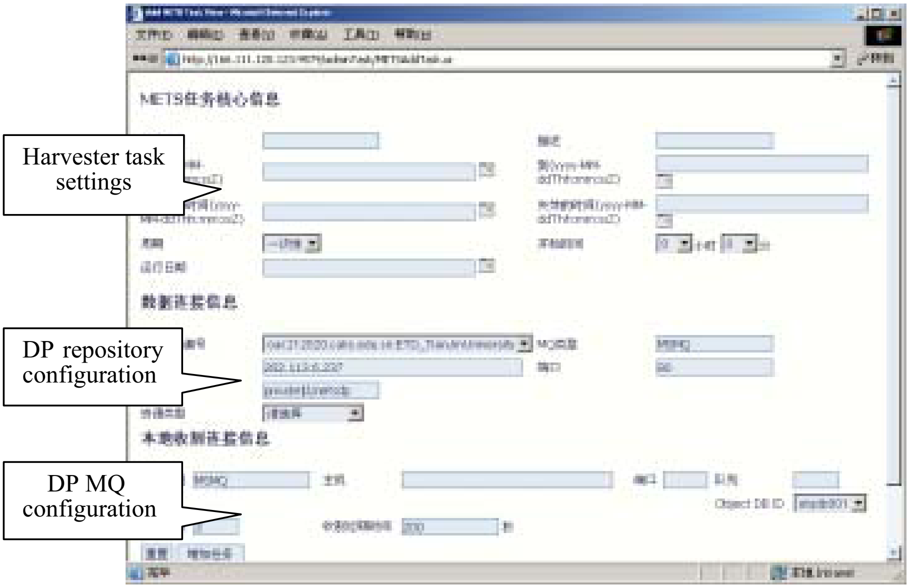


Fig. 5. The web configuration interface of harvester

4 Conclusion and Future Works

In this paper, we first analyze shortcomings of some existing methods of digital resource harvesting and put forward to an alternative approach about resource harvesting. The approach is based on the Message Queue transfer mechanism to ensure the security, reliability and efficiency of datastream transfer. We also specify and standardize message request specification including 5 verbs, and message response specification built on the METS encoding to describe complex digital resources, their metadata and relationship between them. Because of the flexibility and scalability of METS, the approach supports any types of digital resources from any distributed heterogeneous repositories. For example, the CALIS special resource digital library project also uses the approach to harvest the special Chinese resource from repositories of the member universities, such as rarebooks, ancient atlas, rubbings ancient genealogy and chorography and so on. And the approach, which is compatible with the well specified and widely applied OAI-PMH, make its deployment simple and general for existing OAI-PMH implementations.

Our work will continue in following some aspects. The first aspect will focus on improving performance of harvester, such as error warning, harvest interruption handle, detailed log and statistics analysis of harvested data and so on. The second aspect will be concerned with developing data quality check tool to verify and enhance the data quality of harvested digital resources.

References

1. Herbert Van de Sompel, Michael L. Nelson, Carl Lagoze, Simeon Warner. Resource Harvesting within the OAI-PMH Framework., D-Lib Magazine, December 2004
2. Extensible Repository Resource Locators (ERRoLs) for OAI Identifiers, http://www.oclc.org/research/projects/oairesolver/default.htm
3. Encoding full-text links in the eprint jump-off page, http://www.rdn.ac.uk/projects/eprints-uk/docs/encoding-fulltext-links/
4. Technical Standards & specifications of CALIS (China Academic Digital Library & Information System). November 2004.
5. Metadata Encoding & Transmission Standard, http://www.loc.gov/standards/mets/
6. Yang Zhao, Airong Jiang: Building a distributed heterogeneous CALIS_ETD digital library. Digital Libraries: International Collarboration and Cross-Fertilization. The 7th International Conference on Asia Digital Libraries. Shanghai, China, Dec 13-17 2004, Springer-Verlag, Berlin Heidelberg New York, 2004, 155-164.