Towards Interoperable Preservation Repositories: Repository Exchange Package Use Cases and Best Practices

Joseph G. Pawletko; New York University; New York, New York / USA; Priscilla Caplan; Florida Center for Library Automation; Gainesville, Florida / USA

Abstract

Towards Interoperable Preservation Repositories (TIPR) is a project funded by the Institute of Museum and Library Services to create and test a packaging format for inter-repository Archival Information Package (AIP) exchange. The result of this work is the Repository eXchange Package (RXP), a lightweight packaging format that uses the Metadata Encoding and Transmission Standard (METS) schema to encode the RXP structure and the Preservation Metadata: Implementation Strategies (PREMIS) schema to encode digital provenance. The RXP can accommodate heterogeneous AIP structures and can be easily generated and ingested by heterogeneous preservation environments while maintaining an unbroken chain of digital provenance. These attributes make the RXP useful in a variety of preservation contexts: succession, disaster recovery, software migration, diversification, and specialized content processing. The TIPR project and the RXP format have been described in papers delivered at iPRES 2009 and 2010, DLF Spring 2009, and the NIST US Digital Preservation Interoperability Framework Workshop. This paper focuses on how the RXP can be used in each of the different contexts listed above, describes potential issues for each scenario, and recommends best practices for creating, interpreting and ingesting RXP packages. It also explores limitations of the RXP and modifications that have been suggested by other parties. Finally, it outlines decisions that transfer partners must negotiate in order to support successful exchange of content between repositories.

Introduction

Preservation repositories need to exchange ortransfer archival information packages (AIPs)with one another. The canonical case arises whena repository can no longer store and preserve content due to changing preservation priorities or a lack of funding and must transfer the at-risk content to a new custodial organization for safe keeping [1]. This scenario, called "succession", may be repeated many times. Other repository exchange scenarios include disaster recovery, software migration, and specialized content processing.

In each scenario there is no guarantee that the exchanging repositories are compatible. The repositories may usedifferent repository software, have different AIP structures, or enact different preservation policies.This potential incompatibility can be addressed through the use of a common information package transfer format, and inter-repository agreements that address logistical and custodial issues.

In response to these needs, the Toward Interoperable Preservation Repositories Project (TIPR), funded by the Institute for Museum and Library Services (IMLS), developed the Repository eXchange Package (RXP) and a set of discussion topics and questions to help define the boundaries of inter-repository exchange agreements.

RXP Structure

The Repository eXchange Package (RXP) is a lightweight packaging format designed for transferring AIPs between repositories while maintaining an unbroken chain of digital provenance. An RXP consists of METS and PREMIS documents, a files/ subdirectory, and an optional signature file. Figure 1 shows the RXP structure.

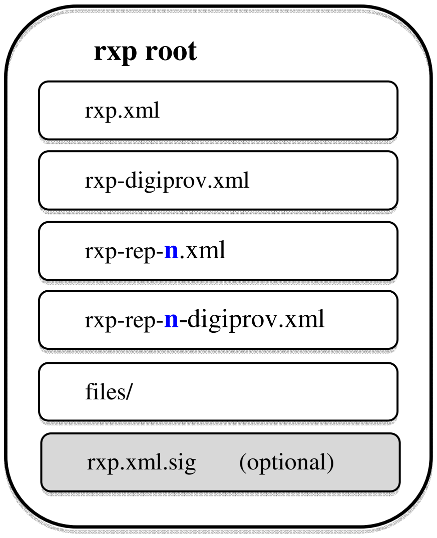

Figure 1.*RXP structure*

- **rxp root/**is the parent directory of the rxp.xml file and is the base of the RXP hierarchy. All file URLs in the RXP are relative to the rxp root.
- **rxp.xml** is a METS descriptor that includes RXP sender information and references to rights documents, and representation descriptors
- **rxp-digiprov.xml** is a PREMIS document that contains digital provenance information specific to this RXP
- **rxp-rep-n.xml** is a METS document that describes the sender's representation(s) in the RXP. There is one rxp-rep-n.xml file for each of the sender's representations packaged in the RXP, i.e., rxp-rep-1.xml, rxp-rep-2.xml, …
- **rxp-rep- n-digprov.xml** is a PREMIS document that contains the digital provenance for the representation described in rxp-

rep-n.xml. There is one rxp-rep-n-digiprov.xml file for each of the sender's representations packaged in the RXP, i.e., rxp-rep-1-digiprov.xml, rxp-rep-2-digiprov.xml, …

- **files/** is a directory that contains the sender's representation files

Repositories capable of reading and writing RXPs should be able to exchange RXPs with other RXP-capable repositories.In an RXP exchange the repository that generates an RXP is called the **source** repository and the repository that receives and ingests an RXP is called the **target** repository.For additional information on the RXP structure, please see [2, 3, 4].

RXP Use Cases

Preservation repositories may use RXPs to exchange AIPs in the following scenarios: succession, disaster recovery, software migration, diversification, and specialized content processing. This list is not exclusive. It is hoped that other RXP use cases will emerge over time.

Succession

The Trustworthy Repositories Audit & Certification: Criteria and Checklist (TRAC:CC)states that trustworthy repositories should have a formal succession plan in place in the event that the repository ceases to operate or changes its scope [5]. The formal succession plan may include logistical details regarding AIP transfer. Source repositories can package AIPs as RXPs and transfer the RXPs to successor repositories. The successor repositories can convert the RXPs into SIPs, ingest the SIPS, ingestand update the associated digital provenance, provide notification of receipt, and formally take custody of the content.

Disaster Recovery

RXPs are useful in disaster-recovery scenarios. For example, two geographically distributed preservation repositories, PR1 and PR2, establish a storage-mirroring agreement in which each repository agrees to allocate 20 TB of storage to store the partner's AIPs, i.e., PR1 stores 20 TB of PR2 AIPs and PR2 stores 20 TB of PR1 AIPs. PR1 and PR2 package their AIPs as RXPs send the RXPs to the partner repository for ingest. Each repository then processes the RXPs as in the Succession case. Should PR1 experience a disaster, PR2 can export PR1's AIPs as RXPs and transmit the RXPs to PR1. As part of the RXP-generation process, PR2 exports all of the digital provenance events that occurred while the PR1 RXPs were in PR2 custody. Thus PR1 can restore their AIPs while maintaining an unbroken chain of digital provenance.

Software Migration

RXPs can facilitate AIP migration from onepreservation repository implementation to another. For example, a repository is upgrading its repository software implementation and needs to migrate all of the AIPs from the current system to the new system. All of the AIPs in the current system can be exported as RXPs, and then all of the RXPs can be ingested into the new system. This scenario is an example of technological, and short-term temporal, interoperability.

Temporal Interoperability

RXPs can also address longer-term temporal interoperability and disaster recovery challenges. For example, in a conventional filesystem-backup environment, the backup system replicates the files as they appear on disk. If a preservation repository uses an opaque file renaming strategy that distributes AIP files across different directories then any disaster recovery from backup tape is dependent upon the repository implementation at the time the backup tapes were created. If, however, a repository first packages its AIPs as RXPs and writes the RXPs to backup-tape then the AIPs can be recovered by any preservation repository implementation able to read the RXP format. This use case may be beneficial to preservation repositories that write their AIPs directly to tape.

Diversification

RXPs can facilitate storage of AIPs in technologically heterogeneous repositories. Those in the OAIS Producer role may request that particularly valuable content be stored in technologically heterogeneous preservation repositories. Storing AIPs in technologically heterogeneous repositories reduces the risk that a software bug in one repository software application will result in the loss or corruption of all copies of an AIP. The RXPcreation,transfer, and ingest processes in this case are the same as those outlined above.

Specialized Content Processing

RXPs can facilitate specialized content processing. For example, PR1 has multiple AIPs that contain files in an obscure format. PR2 has obscure file format migration expertise. PR1 enters into a repository exchange agreement that supports the following workflow:

1. PR1 exports AIPs containing files in obscure format as RXPs
2. PR1 transfers RXPs to PR2
3. PR2 converts the RXPs into SIPs
4. PR2 migrates the obscure format files
5. PR2 ingests the SIPs with the migrated files
6. PR2 updates the digital provenance as appropriate
7. PR2 exports the AIPs as RXPs
8. PR2 transfers the RXPs back to PR1 for processing and ingest

Note that step 5 may not be necessary as long as the digital provenance can be updated appropriately.

RXP Best Practices

During the RXP testing phase, the TIPR partners identified various best practices related to RXP generation and exchange. The RXP specification uses the term "SHOULD" to indicate recommended practices [6].While some recommended practices are self-explanatory, other recommendations are discussed below.

The RXP can accommodate heterogeneous AIP structures. Some AIPs may only contain one representation, while other AIPs may contain multiple representations. In the multi-representation case exchanging repositories may wish to use the ORDER attribute in the rxp.xml structMap to indicate the order in which representations were created. Some repositories only preserve the firstand last-best digital object representations so the ORDER values may not be sequential and the highest ORDER attribute value may not correspond to the highest number **n** in rxp-rep-**n**.xml/rxp-rep-**n**-digiprov.xml.Figure 2 shows an excerpt from an rxp.xml filegenerated by a PR that only stores first and last-best representations. The AIP has been forward migrated three times, so the last-best representation ORDER number is 4.

```
<fileGrp>
<file ID="file-0"
        CHECKSUM="a9bbe4118f0caf1facf9484094108fb2e2ed26d3"
        CHECKSUMTYPE="SHA-1">
<FLocat LOCTYPE="URL" xlink:href="rxp-rep-1.xml"/>
</file>
<file ID="file-1"
        CHECKSUM="51753590f1a7d12b0dc833448f09c168987811b2"
        CHECKSUMTYPE="SHA-1">
<FLocat LOCTYPE="URL" xlink:href="rxp-rep-2.xml"/>
</file>
</fileGrp>

...

<structMap>
<div>
<div ORDER="1">
<fptr FILEID="file-0"/>
</div>
<div TYPE="ACTIVE" ORDER="4">
<fptr FILEID="file-1"/>
</div>
</div>
  ...
```

Figure 2.*rxp.xml structMap div ORDER vs. rep-**n** values*

Given that there are only two representations in the RXP, the highest value **n** in rxp-rep-**n**.xml/rxp-rep-**n**-digiprov.xml is 2.

The METS and PREMIS RXP xml documents should explicitly state the schemas under which the documents were created, e.g., xsi:schemaLocation="http://www.loc.gov/METS/ http://www.loc.gov/standards/mets/version19/mets.xsd". This ensures that, over time, RXP files are validated using the appropriate schema versions.

To maintain an unbroken chain of digital provenance it is important to bind PREMIS object identifiers to their associated digital objects. In each rxp-rep-**n**.xml file there are two METS fileGrp elements. One fileGrp (USE='METADATA') stores information about the rxprepresentation metadata files. The other fileGrp stores information for representation files located under the files/ directory. RXP creation software should store the PREMIS objectIdentifier for each representation file in the METS file elementOWNERID attribute in order to associate the representation file in the files/ subdirectory with the digital provenance for that file. Figure 3 shows this relationship.

In addition to the RXP generation recommendations listed above, there are also recommended transmission and validation methods. When transmitting RXPs between repositories it is critical that all files arrive intact. For this purpose, the TIPR partners used the BagIt packaging formatduring the RXP exchange tests [7]. Upon receipt, the bags were validated using standard bag validation tools [8, 9]. After passing bag validation, the RXPs themselves were validated using metadata schema documents and Schematron files [10, 11, 12]. The TIPR partners created a publicly available Schematron file for each required RXP XML file, and METS profiles have been created and registered with the Library of Congress for each of the RXP METS files [13, 14].After validation, the RXP was converted into a SIP and ingested into the target repository. During this phase the digital provenance was also processed.

```
Excerpt from "rxp-rep-1.xml" (METS document)

<fileGrp USE='METADATA'>
<file ID="TIPR-RMD"
        CHECKSUM="51753590f1a7d12b0dc833448f09c168987811b2"
        CHECKSUMTYPE="SHA-1">
<FLocat LOCTYPE="URL" xlink:href="rxp-rights.xml"/>
</file>
...
</fileGrp>
...
<fileGrp>
<file CHECKSUM='863cc6cdd35d1eadc8e43a8dc589a19cbb09645a'
        CHECKSUMTYPE='SHA-1' ID='file-1'
OWNERID='daitss-test://E00001996_NQF0Z3/file/1'
 SIZE='3793'>
<FLocat LOCTYPE='URL'
xlink:href='files/E00001996_NQF0Z3/wave.xml' />
</file>
...

</fileGrp>
...

-------------------------------------------------------
Excerpt from "rxp-rep-1-digiprov.xml" (PREMIS document)

...
<objectIdentifier>
<objectIdentifierType>URI</objectIdentifierType>
<objectIdentifierValue>daitss-test://E00001996_NQF0Z3/file/1
</objectIdentifierValue>
</objectIdentifier>
...
```

Figure 3.*Link file to provenance through OWNERID*

It is important to consider how identifiers are assigned during ingest. If the target repository uses the same digital object identifier values as the source repository then tracing digital provenance for a given object is straightforward. If, however, each target repository assigns new identifiers to digital objects in an RXP then this "alias" event must be tracked or the chain of digital provenance could be broken. Figure 4 shows the structure of an alias event as encoded in PREMIS [15]. The alias events form a linked list that allows repositories to identify all of the preservation events related to a particular digital object across multiple repositories.

RXP Limitations/Suggested Modifications

The RXP specification should evolve over time. The RXP was initially envisioned as only a transfer package format with limited use outside of an RXP exchange, however the flexibility of the RXP design has led some to consider using RXPs as AIPs [16]. This objective is hinderedby the current RXP prohibition of descriptive metadata (DMD) at the RXP root. The TIPR partners are considering updating the RXP spec to remove the DMD restriction to support the RXP-as-AIP use case.

Transfer Partner Decisions

Successful inter-repository exchange depends on many variables. Although the RXP specification defines the transfer package format, there are additional logistical and custodial questions that must be addressed. The TIPR project partners have drafted a set of topics and questions that exchanging repositories should review and answer before beginning RXP transfers [17].The use cases outlined in this paper may help further narrow the scope of the inter-repository agreement discussions.

The exchanging repositories should record their decisions in an inter-repository agreement after consensus has been reached.

The inter-repository agreement, at a minimum, must document the details of RXP creation, transfer logistics between source and target repositories, actions performed at the target repository upon RXP receipt, treatment of ingested RXP, rights and permissions, financial arrangements, and legal arrangements.

```
<?xml version="1.0" encoding="UTF-8"?>
<eventxmlns="info:lc/xmlns/premis-v2">
<!-- event identifier -->
<eventIdentifier>
<eventIdentifierType>
      URI</eventIdentifierType>
<eventIdentifierValue>
test:/7fecafa0/event/1</eventIdentifierValue>
</eventIdentifier>
<eventType>alias</eventType>
...
<!--the OLD name for the object -->
<linkingObjectIdentifier>
<linkingObjectIdentifierType>
      URI</linkingObjectIdentifierType>
<linkingObjectIdentifierValue>
info:xyz/noid/5e43f71
</linkingObjectIdentifierValue>
<linkingObjectRole>source</linkingObjectRole>
</linkingObjectIdentifier>

<!--the NEW name for the object -->
<linkingObjectIdentifier>
<linkingObjectIdentifierType>
      URI</linkingObjectIdentifierType>
<linkingObjectIdentifierValue>
test:/7fecafa0/file/0</linkingObjectIdentifierValue>
<linkingObjectRole>alias</linkingObjectRole>
</linkingObjectIdentifier>
</event>
```

Figure 4.*alias event encoded in PREMIS*

RXP creation options

The structure of a valid RXP can differbetween pairs or groups of repository exchange partners. The exchanging repositories need to select the schema versions for the RXP xml files, and decide which optional RXP files and parameters are in scope for their particular RXP exchanges. The following topics and questions are adapted from this document [17].

RXP transfer logistics

After the RXP structure has been defined, the repositories must decide how RXPs will be transferred.

- Will the RXPs be wrapped in "bags"?
- Will the bags be further packaged and compressed before transfer?
- Will the RXPs be transferred across the network or by disk?
- If across the network, what protocol will be used?
- What credentials are required for successful transfer?
- When transferring multiple RXPs, what mechanisms will be used to ensure that all RXPs have been transferred?

These issuesshould be decided prior to RXP exchange.

Target repository actions upon RXP receipt

Exchanging repositories need to agree upon RXP processing upon receipt. These actions will differ based on the "exchange scope". Exchange scope refers to the degree to which the target repository has to "understand" the contents of an RXP. The TIPR partners have identified three exchange scopes: shallow, deep, and enhance.

In a shallow exchange the target repository preserves the received representations with minimal modification. The target repository updates digital provenance metadata, but nothing else. This allows the target repository to return content as received, as in the Disaster Recovery and Diversification use cases.

In a deep exchange, the target repository fully comprehends and takes full custody of the received content. The deep exchange scope is applicable to the Succession and Software Migration use cases.

In an enhance exchange, the target repository enhances a representation or representations and returns the enhanced entity to the source repository. The enhance exchange scope is applicable to the Specialized Content Processing use case.

In addition to exchange scope, other questions that need to be answered include:

- How will RXP receipt be acknowledged?
- How quickly will RXPs be ingested into the target repository?
- How will the content producer be notified upon successful ingest?
- What criteria will be used to reject an RXP at the target repository?

Post-ingestRXP handling

The repositories must agree upon post-ingest handling of the ingested RXPs.

- Which metadata from the received RXP will be stored?
- RXPs are likely to contain metadata from the source AIP. How should this metadata be handled? Should the metadata be saved as-is, or mapped into the target-repository format?
- What preservation strategies will be applied to the ingested RXPs, e.g., normalization, format migration?
- Some repositories store all intermediate representations of a digital object while others only store two: the original representation and the last-best representation. How should the intermediate representations be managed?
- Can ingested RXPs be deleted?

Rights and permissions

Rights and permissions for the exchanged RXPsmust be specified.

- Are all RXPs subject to the same terms, or do terms vary by RXP?
- Are there package-specific, representation-specific, and/or object-specific rights statements?
- How will the rights-information be encoded?
- What happens if the target repository is unable to comply with the rights statements?

Financial arrangements

Repositories must also work out financial arrangements. Storage is not free and there are additional costs associated with running and managing any repository system.

- Who will pay for the costs incurred by the target repository,the source repository or the content owner?
- What is the schedule of charges and terms and conditions of payment?

Legal arrangements

Finally, repositories must agree upon the legal arrangements for RXP processing.

- What is the standard contract between the target repository and its users? How does this impact the content received from the source repository?
- Is the source repository protected from any copyright infringement committed by the target repository?
- What liability does the source repository have for viruses or flaws in the transferred materials?

Conclusion

The TIPR Repository eXchange Package (RXP) is useful in a variety of preservationscenarios. In each scenario, exchanging repositories need to explore the boundaries of the exchange. The TIPR Project has developed a list of questions and topics that may help exchange partners define inter-repository logistical and custodial obligations. Theseobligationsshould be documented in an inter-repository agreement.

RXP creation best practices, RXP limitations, and new RXP uses cases have emerged since the inception of the TIPR Project. Best practices should continue to be documented and the RXP specification should evolve to accommodate the needs of the preservation community.

Acknowledgements

The TIPR project is supported by a grant from the Institute of Museum and Library Services. The authors thank William Kehoe, Francesco Lazzarino, Rasan Rasch, and Marly Wilson for their work on the TIPR project.

References

[1] Blue Ribbon Task Force on Sustainable Digital Preservation and Access, Sustainable Economics for a Digital Planet: Ensuring Long-Term Access to Digital Information, pg 74, (2010). Retrieved March 18, 2011, from http://brtf.sdsc.edu/biblio/BRTF_Final_Report.pdf

[2] Priscilla Caplan,“Repository to Repository Transfer of Enriched Archival Information Packages,” D-Lib Magazine v.14 no.11/12, (2008).Retrieved March 18, 2011, fromhttp://www.dlib.org/dlib/november08/caplan/11caplan.html

[3] Priscilla Caplan, “Towards Interoperable Preservation Repositories (TIPR),” (2010). Retrieved March 18, 2011, from http://ddp.nist.gov.workshop/papers/03_08_Caplan_TIPR.pdf

[4] Priscilla Caplan, William Kehoe, Joseph Pawletko “Towards Interoperable Preservation Repositories,” International Journal of Digital Curation, v.5, no.1, pg. 34 (2010). Retrieved March 18, 2011, fromhttp://www.ijdc.net/index.php/ijdc/article/view/145/207

[5] TRAC:CCThe Center for Research Libraries & OCLC, “Trustworthy Repositories Audit & Certification: Criteria and Checklist”, (Chicago, IL, 2007). Retrieved March 18, 2011, from http://www.crl.edu/sites/default/files/attachments/pages/trac_0.pdf

[6] TIPR Project Partners, “Repository eXchange Package (RXP) Spec,” (2010). Retrieved March 18, 2011, from http://wiki.fcla.edu:8000/TIPR/21

[7] A.Boyko, J. Kunze, J. Littman, L. Madden, B. Vargas, “The BagIt File Packaging Format,” (2009). Retrieved March 18, 2011, fromhttp://www.cdlib.org/inside/diglib/bagit/bagitspec.html

[8] Library of Congress, computer code “Library of Congress – Transfer Tools,” (2011). Retrieved March 18, 2011, fromhttp://sourceforge.net/projects/loc-xferutils/

[9] Francesco Lazzarino, computer code “Ruby Library and Command Line tools for bagit,” (2010).Retrieved March 18, 2011, fromhttps://github.com/flazz/bagit

[10] Jerome McDonough, et. al., XML schema “METS: Metadata Encoding and Transmission Standard,” (2010).Retrieved March 18, 2011, fromhttp://www.loc.gov/standards/mets/mets.xsd

[11] Ray Denenberg, XML schema“PREMIS Preservation Metadata Schema,” (2011).Retrieved March 18, 2011, fromhttp://www.loc.gov/standards/premis/premis.xsd

[12] Rick Jelliffe, “The Schematron,” (n.d.).Retrieved March 19, 2011, from http://xml.ascc.net/resource/schematron/schematron.html

[13] Francesco Lazzarino, computer code “TIPR RXP Tools,” (2010). Retrieved March 19, 2011,from https://github.com/flazz/rxp

[14] Library of Congress, “Registered Profiles,” (2010).Retrieved March 19, 2011, from http://www.loc.gov/standards/mets/mets-registered-profiles.html

[15] Francesco Lazzarino, untitled alias event example, (n.d.).Retrieved March 19, 2011 from http://daitssfs.fcla.edu/alias-event.xml.html

[16] Peter Van Garderen, “ARCHIVEMATICA: Using Micro-Services and Open-Source Software to Deliver a Comprehensive Digital Curation Solution,” iPRES 2010, pg. 145. (2010).Retrieved March 19, 2011, from http://publik.tuwien.ac.at/files/PubDat_191968.pdf

[17] TIPR Project Partners, “Inter-repository Agreement” draft, (2010). Retrieved March 19, 2011, from http://wiki.fcla.edu:8000/TIPR/36

Author Biography

JosephPawletkois a Software Systems Architect in New York University’s Digital Library Technology Services group where he works on content processing and preservation issues.

Priscilla Caplan is Assistant Director for Digital Library Services at the Florida Center for Library Automation, where she oversees the Florida Digital Archive.