Digital Preservation Interoperability through Preservation Actions Registries

Matthew Addis[1], Justin Simpson[2], Jon Tilbury[3], Jack O'Sullivan[3], Paul Stokes[4]

[1]Arkivum (UK), [2]Artefactual (Canada), [3]Preservica (UK), [4]Jisc (UK)

ABSTRACT

Current digital preservation systems such as Archivematica and Preservica lack a common and consistent way to describe and execute preservation policies and actions at a technical level. Archivematica's Format Policy Register (FPR) and Preservica's Linked Data Registry (LDR) both define what tools and rules to use when doing digital preservation and whilst these two approaches aim to solve similar problems, they are not interoperable. There is no way to share technical information between the two solutions and for users to share their experience on what approaches to use in different contexts and why. This hinders implementation and execution of digital preservation in an interoperable way within the digital preservation community. This paper presents new work by Artefactual, Arkivum, Preservica and Jisc on how Preservation Action Registries (PAR) could be used to capture and share technical best practice for the preservation of digital objects in the form of a corpus of machine-readable recommendations. The registry's data model defines what tools can be used for different digital object formats, what properties can be extracted or measured and what preservation actions can be taken. The model includes contextual and historical information on the reasons why recommendations are being given. The information is versioned and includes the tool parameters and software environments needed to execute different preservation actions so they are directly 'executable' by preservation systems. Our model makes registry content accessible through APIs using a distributed set of registries rather than a single canonical source. In this way, information can be made available from a range of sources including central registries (moderated and curated) or by exchange directly between trusted peer institutions or systems. Preservation systems such as Archivematica and Preservica are able to publish to and consume content from these registries. The work is supported by Jisc as part of the Research Data Shared Service initiative and we believe is the first time that vendors, national service providers and end-users have all come together in this way. The PAR approach should engender greater confidence in digital preservation, provide users with more flexibility and knowledge sharing opportunities, and accelerate the adoption of digital preservation in new sectors.

KEYWORDS

Digital preservation; technical registry; system interoperability; knowledge sharing; preservation tools; preservation policies; preservation rules; research data; shared service

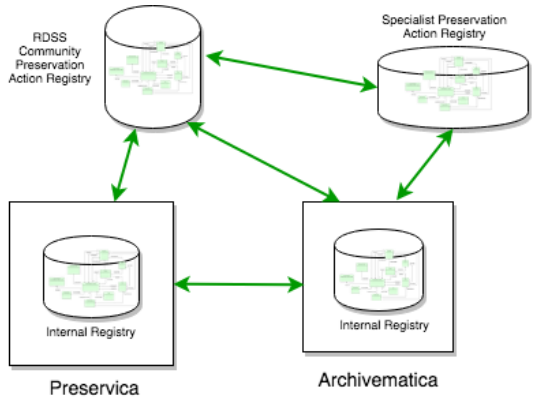
Conference Theme Addressed

The paper describes how Preservation Actions Registries support the sharing of technical information on digital preservation rules/tools within a community and between preservation systems. This addresses the conference themes of 'Mapping out sustainable digital preservation approaches and communities' and 'Technological infrastructure'

Introduction

The objective of our work on Preservation Action Registries (PAR) is to allow Archivematica's[7] Format Policy Register (FPR) [10] and Preservica's [8] Linked Data Registry (LDR) [11] to be combined and extended in order to create a shared registry approach that can used by both the preservation systems within the Jisc Research Data Shared Service (RDSS) [9] and other preservation systems.

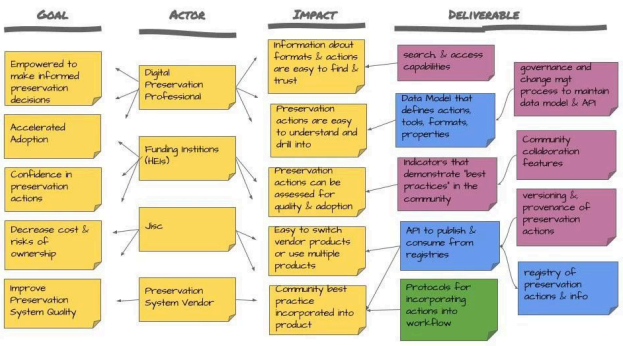
The key feature of our approach is to decentralise machine-readable knowledge of preservation actions so this knowledge can be shared within a community rather than being baked into a single digital preservation solution or be under the control of a single supplier or institution.



Exchange of preservation action information between preservation systems and registries

This enables direct exchange between preservation systems, e.g. so specific users can share their preservation policies with each other, and also supports the publication and sharing of preservation information with a wider community. The community can include using general-purpose community registries (e.g. a registry managed by Jisc for the RDSS) or discipline specific registries (e.g. a registry for preservation actions on the data types seen in a specific discipline such as creative arts, life sciences etc. that are provided by or embody knowledge from domain experts, e.g. AV specialists or providers of scientific instruments).

The benefits of this approach include RDSS users being able to make more efficient use of their time and resources as well as having more confidence when doing digital preservation. The goals, stakeholders, benefits and the work that needs to be done to deliver it are shown in the impact map.



PAR Impact Map

The work on the PAR is ongoing. The first stage of the work has started to address some core questions that include: (a) What are the base entities of the PAR? Entities include Formats, e.g. as identified using PUIDs such as PROMOM IDs, Tools that can act on Formats (e.g. DROID, veraPDF, JHOVE, FFMPEG, ImageMagick), the Actions that tools perform (e.g. identify, verify, extract properties, migrate), and the Properties that are inputs/outputs of these actions (e.g. checksum, number of pages, image height, image width, viewing duration, and (b) What is the data model for the PAR? How do we describe the above entities in machine understandable terms and how can we make registry entries 'executable' by preservation systems? For example, how do we describe preservation tools in enough detail to enable them to be installed and run in an automated way as possible (e.g. what is the data model for name, owner, licence, executable name, version, software dependencies, parameters, and run time environment)?

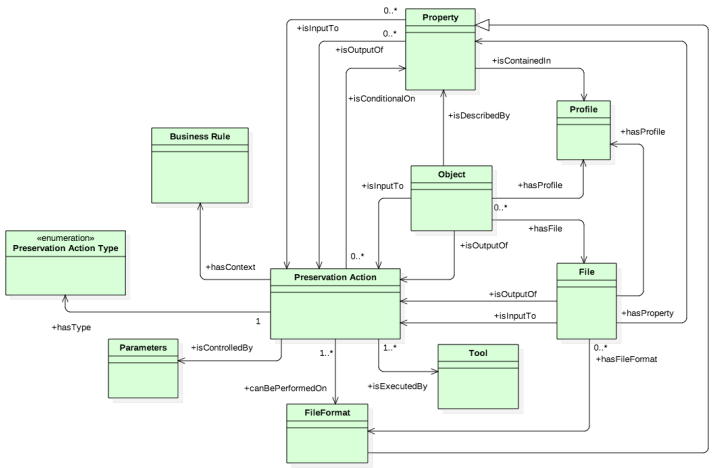
Overview of the PAR conceptual model

In developing our PAR conceptual model we align with existing models where it makes sense, including Jisc's Canonical Data Model (CDM)[1], the Portland Common Data Model (PCDM)[2] and the PREMIS model of events, objects and environments [3]. Likewise, we build on existing models for describing workflows including general purpose models of command line tools (Common Workflow Language Command Line Tool Description)[4], task execution (GA4GH Task Execution Schema)[5] and the Open Service Broker API [6]. Our objective is to use compatible models and standards with rather than trying to re-invent the wheel. Our conceptual model is shown below.

Preservation Actions are processes that are run as part of performing digital preservation (e.g. generate and check checksums, convert a file from format A to format B, extract properties X,Y,Z from a file). **Preservation Actions** are classified according **Preservation Action Types** including fixity, identification,

characterisation, validation, migration/normalisation, and rendering, e.g. as defined by PREMIS event types [13].

A **Preservation Action** acts upon an input pcdm:**Object** or an input pcdm:**File**. An Object may itself contain one or more Files or an **Object** might only contain metadata (e.g. a METS document containing technical and descriptive metadata). A **Preservation Action** may create an output **Object** (e.g. an AIP that contains multiple Files and metadata etc.) or a **Preservation Action** may create a single output **File** (e.g. a TIFF image created from normalising an input JPEG image).



PAR Conceptual Model

A **Preservation Action** is executed using one or more **Tools** (e.g. md5sum, DROID, JHOVE, ffmpeg etc.). A tool could be run on a command line or it could be wrapped in some other way e.g. as a Web Service or in a Docker Container. Performing a Preservation Action may be as simple as invoking a tool, i.e. running a command line, or it could be more complicated, e.g. using an asynchronous webservice over a REST API. The approach we take is to abstract/decouple the specific tools/parameters/execution models used to execute digital preservation from the preservation action(s) that are being achieved.

A **Preservation Action** may take **Properties** as inputs (e.g. checksum validation takes in a checksum for the file that is being checked). A **Preservation Action** may also create/extract **Properties** about a **File** or an **Object**, for example, generating a checksum, identifying file format, extracting video resolution and bitrate etc. A **File Format** (e.g. as defined by a PRONOM PUID) is one example of a Property that is associated with a File. **File Formats** are an explicit entity in the model because of the lynch-pin role that formats play in digital preservation strategies and are hence are a first-class object. A set of **Properties** form a **Profile** for an **Object** or a **File**. Profiles can include various types of metadata (descriptive, technical, provenance etc.) and may be at the level of individual Files (e.g. size, checksum, file format) or an Object as a whole (e.g. an audiovisual asset). For example, Technical Application Profile [12] is used for Files in PCDM and Resource Type profiles are used for Objects in the Jisc RDSS CDM [1]. Profiles provide a way to group together metadata fields that are not common to all types of Object or File without having to make these fields part of the core model. This is the approach used in Dublin Core Application Profiles [14] and is common in digital preservation where general purpose standards do not cover the detail needed for specific types of content, e.g. objectCharacteristicsExtension in PREMIS [3].

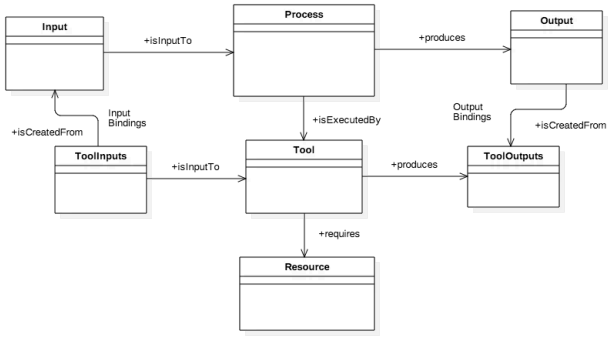
Preservation Action	par:GenerateMD5checksum	Preservation Action Type	premis:fixity
		Input Type	pcdm:File
		Output Type	par:Property
Tool	par:toolRegistryName	COPTR	
	par:toolRegistryKey	Md5sum Unix Command	
	par:toolVersion	md5sum (GNU coreutils) 8.25	
	par:toolDocumentation	https://man.cx/md5sum	
Property (Output)	par:propertyType	premis:messageDigest	
	premis:messageDigestAlgorithm	MD5	
	xsd:anyURI	http://id.loc.gov/vocabulary/preservation/cryptographicHashFunctions/md5.html	

Simplified example of a Preservation Action (generating a MD5 checksum on a File)

A **Preservation Action** is controlled/configured using a set of **Parameters** (e.g. what level of compression to use for video migration/normalisation, what checksum algorithm to use when fixity checking, what signatures to use for file format identification). A **Preservation Action** may also be conditional on input **Objects** or **Files** having particular **Properties** (e.g. validation can only be done for Files that have PDF as their File Format because the only tool available is veraPDF). These constraints help stop attempts to perform actions where tools aren't suited for the purpose. **File Format** is a common constraint, but there may be others, e.g. Tools may only work for limited file sizes, or metadata validation may only be possible for certain schemas.

Context around a **Preservation Action** is captured in **Business Rules**. This allows a series of statements to be made about by organisations or individuals about preservation actions. This is about expressing preferences, best practice, ordering, experience etc. This adds additional context that goes above and beyond the basic constraints expressed in the rest of the model. For example, the Business Rules entity provides a place for people to say that, for example, mediainfo is better for characterisation than ffprobe for a given file format. It also gives a place for people to say what variations or tweaks might be needed or to provide more detail than can be sensibly encoded in the PAR. Examples of business rules might be: "I use Parameters X on Tool Y to perform Action Z", "Tool A is preferable to Tool B for Format F", "First try Tool P and if that doesn't give a good result then try Tool Q", "I only do Action A for Format B", "I extract Property P using Tool Q", "I found Tool T didn't work properly when validating Format C", "My order of preference/priority when using multiple tools for file format identification is X then Y then Z".

At this stage of the project, the scope of describing/modelling software in the PAR is restricted to software that is used to *perform* specific preservation of files such as file format validation, format migration/normalisation or simple viewing/rendering of file content. In other words, software is not *the subject of* digital preservation. In future work we would like to extend the model to include preservation of software and environments that *is the subject* of preservation, e.g. preservation of software applications that are used to create, interact with and reuse digital content. This includes use of techniques such as emulation and containerisation as part of capturing/preserving the original environment for digital content and maintaining this environment over time. There is already interesting work in this area, e.g. ReproZip [16], Singularity [17], Encapsulator [15] and Research Objects [18] that we would seek to build upon.



PAR preservation action execution model

Preservation Action Execution Model

The approach of defining a combination of Parameters/Actions/Tools that have some form of Input and produce some form of Output allows us to align with the Common Workflow Language (CWL) approach to describing Command Line Tools (inputs, outputs, parameters, process, command line tool) and the GA4GH Task Executor model (inputs, outputs, executor, environment). The Preservation Action in the conceptual model becomes a Process in CWL that uses a Tool for its execution.

A **Process** has an **Input** (e.g. an Object plus some Parameters) and it produces an Output (e.g. some Properties and some logging information). A **Process** is executed by a **Tool**. A Tool requires **Resources** in order to be runnable (e.g. memory, compute, disk, libraries, OS etc.). The Tool has **ToolInputs** (e.g. command line options specifying input file locations and any parameters that determine what the tool does). The **Input** is converted to the **ToolInput** through an Input Binding, e.g. a given part of the Input (e.g. File to be processed) is set as a particular command line option (e.g. -i /inputfiles/myfile) for the tool The **Tool** produces **ToolOutput** (e.g. log files, output files,

exit codes) and this is used to create the Output of the Process through an Output Binding (e.g. 'success' parameter = True is set in the Output the tool exit code = 0 in the ToolOutput). By using Bindings of the Input and Output of a Process to the specific way information is passed to and from a Tool, it is possible to bind to different implementations of the same process, e.g. a Web Service, a Command Line Tool, or a Docker Container.

Expected outcomes

By allowing the rapid exchange of factual information and policy recommendations between vendors, domain experts and novice and expert practitioners this initiative will accelerate the development and distribution of best practice leading to improved collaboration, reduced duplication and the automated application of internationally agreed preservation activities.

Conclusions

This paper has presented initial work on Preservation Action Registries as a basis for community sharing of technical knowledge on how to perform preservation actions in practice. The PAR approach supports machine-readable exchange of detailed information on digital preservation in terms of the actions, tools, formats, properties and business rules. The use of open standards and multiple registries allows for a community of preservation users, service providers and vendors to exchange both technical specifications of digital preservation actions and supporting context of when to use these actions, who has implemented them and what results were achieved. The next stages of the work include: developing a working PAR prototype; integrating this with Archivematica and Preservica; and defining and using a specific set of use cases commonly found in the digital preservation world (e.g. file format identification, characterisation and migration/normalisation) as test cases for evaluating the approach and demonstrating the benefits.

Acknowledgements

The work presented in this paper is being supported by Jisc as part of the Research Data Shared Service.

References

[1] Jisc Canonical Data Model (CDM). https://github.com/JiscRDSS/rdss-canonical-data-model
[2] Portland Common Data Model (PCDM). https://github.com/duraspace/pcdm/wiki
[3] PREMIS. https://www.loc.gov/standards/premis/
[4] Common Workflow Language (CWL). http://www.commonwl.org/v1.0/CommandLineTool.html
[5] GA4GH Task Execution Schema (TES). https://github.com/ga4gh/task-execution-schemas/
[6] Open Service Broker API. https://www.openservicebrokerapi.org/
[7] Archivematica digital preservation solution. https://archivematica.org/
[8] Preservica digital preservation solution. https://preservica.com/
[9] Jisc Research Data Shared Service (RDSS). https://www.jisc.ac.uk/rd/projects/research-data-shared-service
[10] Archivematica Format Policy Registry (FPR). https://www.archivematica.org/en/docs/fpr/
[11] Preservica Linked Data Registry (LDR) .
https://pdfs.semanticscholar.org/98bd/5a5ac26c40438b8af978021bdca8123f1988.pdf
[12] PCDM Technical Metadata Profile.
https://wiki.duraspace.org/display/samvera/Technical+Metadata+Application+Profile
[13] PREMIS event types. http://id.loc.gov/vocabulary/preservation/eventType.html
[14] Dublin Core Application Profile. http://dublincore.org/documents/profile-guidelines/
[15] Encapsulator. https://arxiv.org/abs/1803.05808
[16] ReproZip. https://www.reprozip.org/
[17] Singularity. https://singularity.lbl.gov/
[18] Research Objects. http://www.researchobject.org/