Preservation as a Service for Trust (PaaST)

Functional and Data Requirements for Digital Preservation

Kenneth Thibodeau, Daryll Prescott, Richard Pearce-Moses, Adam Jansen, Katherine Timms, Giovanni Michetti , Luciana Duranti, Corinne Rogers, Larry Johnson, John R. Butler, Courtney Mumma, Vicki Lemieux, Sarah Romkey, Babak Hamidzadeh, Lois Evans, Joseph Tennis, Shyla Seller, Kristina McGuirk, Chloe Powell, Cathryn Crocker, Kelly Rovegno

April 24, 2018

Version 1.0

Abstract

Preservation as a Service for Trust (PaaST) presents functional and data requirements for digital preservation. The requirements reflect the findings of the first and second InterPARES collaborations that in the digital realm, preservation of information necessarily extends either to the output of an information object intended for human use in a form suitable for human consumption or, in the case of something intended to be run on a computer, to reloading it on a computational platform.[1] Preservation would be futile if a preserved information object could not be accessed in its proper form. The maintenance over time of the bit streams that are the basis for either the runtime versions or human-readable output are necessary, but insufficient for digital preservation. Regardless of how well the bitstreams are maintained, the information objects that are the targets of preservation can be corrupted if the wrong hardware, software, or processing is used to instantiate them. This document extends the earlier InterPARES findings in two ways. First, InterPARES research has focused on digital records, but the PaaST requirements are formulated to be applicable to the preservation of virtually any type of digital information, not just records. Second, while earlier treatment of preservation in InterPARES was abstract and conceptual, the PaaST requirements are articulated to support implementation and even the production of software for preservation The requirements are supplemented, in the chapters that follow, by a domain model and explicative text.

This document includes a glossary of special terms used, followed by narrative chapters that explain what is involved in implementing the requirements and the functional capabilities they support. It begins with an overview of the document and explanation of the use of the Unified Modeling Language to supplement the narrative. Chapter 2 is the glossary. Chapter 3 identifies the principal parties involved in digital preservation and their roles and specifies how users are involved in carrying out preservation activities. Chapter 4 describes the basic classes of objects involved in digital preservation. Chapter 5 presents an overview of the information needed to carry out and manage preservation. Chapter 6 describes the major functional capabilities supported by PaaST requirements. Chapter 7 describes how to evaluate the success of digital preservation. Chapter 8 discusses the special case of preserving authentic records. Chapter 9 summarizes the capabilities defined in the requirements, and chapter 10 contains the requirements.

[1] InterPARES 1. Preservation Task Force Report. http://interpares.org/book/interpares_book_f_part3.pdf. InterPARES 2.

Table of Contents

List of Figures

List of Tables

Introduction[2]

The emergence of the Cloud presents special challenges to the preservation of digital information. Just as the Cloud is not a specific technology, or even a family or configuration of technology, the primary challenges it poses for digital preservation are not technological. Rather, the challenges stem from the loss of control over, and even knowledge of, what hardware and software are used and how they are used. Insufficient knowledge and control create a basic issue of trust: how can we trust preservation in the Cloud without adequate knowledge or control over how it is accomplished?

Even before technology service providers began offering capabilities labeled as 'Cloud,' researchers in the first InterPARES project recognized that digital preservation requires a "Chain of Preservation" that ensures that digital records survive uncorrupted.[3] Any break in knowledge of how digital information has been preserved could make it impossible to assert that what remains is what it should be. The Chain of Preservation is realized by implementing controls that ensure that the requirements for preservation are satisfied throughout the life of the information being preserved. The Chain of Preservation is reflected, after the fact, in data that demonstrate that these requirements have been satisfied. The Preservation as a Service for Trust (PaaST) project, an initiative of the InterPARES Trust,[4] defines a comprehensive set of functional and data requirements that support preservation of digital information regardless of the technologies used or who uses them. The requirements are intended to enable authentic digital preservation in the Cloud; nevertheless, the requirements are valid in other scenarios as well, including in-house preservation and situations where digital preservation includes both in-house and contracted services. Moreover, PaaST requirements are applicable to cases that include

- heterogeneity in the types of information objects being preserved;
- variety in applicable directives, such as laws, regulations, standards, policies, business rules, and contractual agreements,
  - including varying conditions of ownership, access, use, and exploitation;
- variation in institutional arrangements and relationships between or among the parties involved;
- a wide spectrum of circumstances as possible from best practices to worst cases.

PaaST builds on earlier InterPARES findings and addresses the issue of trusting preservation, in the Cloud and in other contexts, by providing for the articulation and implementation of criteria for

[2] Terms that are specific to PaaST are capitalized and are defined in the Glossary.

[3] Ibid.

[4] InterPARES Trust (ITrust 2013-2018) is a multi-national, interdisciplinary research project funded by a Social Sciences and Humanities Research Council of Canada Partnership grant. ITrust explores issues concerning digital records entrusted to the Internet, with the goal of generating the theoretical and methodological frameworks to develop local, national and international policies, procedures, regulations, standards and legislation, in order to ensure public trust grounded on evidence of good governance, a strong digital economy, and a persistent digital memory. ITrust is a research partnership that comprises over fifty universities and organizations, national and multinational, public and private, in North America, Latin America, Europe, Africa, Australasia, and Asia. Its researchers are experts in archival science, records management, diplomatics, law, information technology, communication and media, journalism, e-commerce, health informatics, cybersecurity, information governance and assurance, digital forensics, computer engineering, and information policy. https://interparestrust.org.

successful preservation and using these criteria as the basis for reporting and evaluation of both actions taken to preserve digital information and the state of the objects being preserved.

The PaaST requirements support the preservation of authentic records and enable evaluation and documentation of success in doing so; nevertheless, the requirements are formulated with sufficient flexibility to enable adaptation of the criteria for success to cases where information objects are not preserved as records.

> *The difference between preserving an information object as a record or not can be illustrated in the case of a data set of scientific observations. If the data set is preserved as a record, then criteria such as the accuracy, precision, truth and validity of the data values are irrelevant to the success of preservation. The data must retain the values they had in the context in which the data set was used as a record, and be able to be instantiated consistent with the manner in which they were presented in that context. In contrast, if the data were to be preserved to support further scientific research, then the criteria of data quality would be critical in the decision whether to preserve the data set, but the way the data is organized, and even the specific expression of data values might be changed (e.g., from imperial to metric units of measurement) to better serve research purposes.*

PaaST requirements enable adaptation by allowing for variation in the specification of what is required to identify an information object in accordance with different purposes of preservation and different characteristics of the objects being preserved. These specifications can be formulated in different PaaST implementations as “PermanentFeatures.” A PermanentFeature is an attribute or an operation of an object that must remain unaltered for preservation to be deemed successful.

The PaaST requirements supplement the Open Archival Information System (OAIS) Reference Model.[5] OAIS is a conceptual framework for digital preservation that describes, in a technologically neutral manner, the activities and the information that are necessary for preservation. Effectively, it has defined the universe of discourse for digital preservation in a variety of contexts around the world. As a reference model, OAIS neither specifies a design or an implementation nor prescribes or even recommends any specific technology for preservation. The standard recognizes that implementations might organize functionality differently. PaaST requirements supplement OAIS in that they are intended to be directly implementable in software. Nonetheless, like OAIS, PaaST does not specify what technology should be used. Rather, it defines what the technology must be able to do.

The focus on implementation entails differences from OAIS in the in the way functional requirements are organized and, consequently, in the data required to support the functions. The focus on implementation also leads PaaST to have a narrower scope than OAIS in two basic respects. First, OAIS covers both analog and digital information, but the PaaST requirements are limited to preserving digital information. Second, the PaaST requirements address only what can be implemented in computer systems; therefore it has a narrower range of functions than OAIS, excluding the OAIS activities of Administration and Preservation Planning. PaaST also does not address what OAIS calls “Common Services,” generic computational capabilities that should be available in all platforms and are not specific to preservation. PaaST also does not address the Designated Community, which is the OAIS designation for the target group of consumers. PaaST assumes that the needs, interests, and capabilities of the Designated Community are addressed in plans, policies and procedures and that, to the extent these need to be addressed in preservation, they are expressed and implemented as PreservationRules. Another important difference is that OAIS is normative, describing what should be done from a best practices perspective. In contrast,

[5] International Organization for Standardization. (2012). 14721 Space Data And Information Transfer systems -- Open Archival Information System -- Reference model. (No. ISO 14721:2012). Geneva, Switzerland:

PaaST requirements are empirically oriented, aiming to accommodate cases that are far from ideal, as well as supporting best practices.

The PaaST requirements are neutral with respect to preservation policies and methods. For example, technological obsolescence can be addressed by normalizing data formats, by successive format migrations or by migrating software using emulators. PaaST allows any or all of these methods to be used. The requirements related to such methods allow different methods to be prescribed in different cases, but specify generating and collecting data about their application both to determine their effect on the information being preserved and to evaluate the methods. Similarly, the PaaST requirements provide for collection and use of a broad range of data about the information that is preserved, but in particular implementations, some of these requirements could be waived. Preservation policies and specific management decisions can be implemented in PaaST as executable controls.

1. Preservation as a Service

The benefits of Cloud Computing are attracting more and more organizations to acquire computing capabilities as a service. Both Cloud service providers and those who analyze the market report impressive growth to date[6] and predict substantial growth in the future.[7] Use of Cloud services for preservation entails loss of knowledge and control over the technologies used. However, the lack of complete control and knowledge is not unique to the Cloud. In fact, it is inevitable in digital preservation. Given continuing and open-ended change in the technology, it is inevitable that the hardware, software and storage media used in digital preservation will be replaced over time. Use of different and often unspecified technology also characterizes delivery of preserved information over the Internet even in a limited period of time.

The approach embodied in the PaaST requirements takes this predicament as a given. The requirements extend the perception that digital preservation entails the use of different technologies over time to enable different, independent technologies to be used for digital preservation both simultaneously and sequentially and, most importantly, reliably. PaaST requirements aim at the preservation of information that is digitally encoded, not at the preservation of technology. This objective requires some degree of independence of the preserved information objects from the technology used in preservation. Without such independence, it would not be possible to assert that any information object that had been moved from one type of hardware or software or storage to another had been preserved successfully. We would only be able to say that it had been transitioned, but the transition might have entailed the loss or corruption of features we want to preserve. The key to technology independence is the specification of what properties of an information object must be preserved, called “PermanentFeatures” in the requirements. Nonetheless, while the requirements support the use of different technologies, they do not require it. Some properties of digital information objects may be so tightly dependent on specific technology that the technology must be maintained in an operable state as long as the information objects that depend on it. Nothing in the PaaST requirements prevents this, although the requirements do not provide specific support for any particular technology.[8]

Thus, it is not assumed that the requirements will be implemented in a single preservation system. Rather, to ensure suitability for a wide variety of situations, the requirements are grouped into sets of related functions, called capabilities. Different capabilities may be implemented using methods and tools best suited to their purpose, independently of those used for other capabilities, with the possibility that different capabilities may be performed by different providers; for example, some might be performed in house while others may be acquired under one or more contracts with external parties.

1.1. Where: the Preservation Environment

Rather than speak of a preservation system, PaaST introduces the concept of a Preservation Environment. A Preservation Environment includes both the set of Preservation Targets that are

[6] Jourdain Novet. Amazon Web Services brings in $2.4B in revenue in Q4 2015, up 69% over last year. January 28, 2016. http://venturebeat.com/2016/01/28/amazon-web-services-brings-in-2-4b-in-revenue-in-q4-2015-up-69-over-last-year/.

[7] Synergy Research Group. 2015 Review Shows $110 billion Cloud Market Growing at 28% Annually. January 7, 2016. https://www.srgresearch.com/articles/2015-review-shows-110-billion-cloud-market-growing-28-annually.

[8] It should be noted that reliance on a particular technology in digital preservation necessarily limits the potential for access to the preserved information and increasingly over time.

preserved under the same Preservation Rules and the technological infrastructures and tools used in their preservation. The Preservation Environment may include separate, different and independently managed hardware and software used by different Preservation Service Providers. The capabilities offered by a single provider are called the Local Preservation Environment.

The information that is preserved in a Preservation Environment , as well as the information generated or used in carrying out preservation, may be organized in a taxonomy of classes. A class is a group of objects that have at least one thing in common. The PaaST taxonomy provides an overall consistent framework for digital preservation and allows customization in different Preservation Environments. Accordingly, its classes are defined at general levels. In any Preservation Environment, the taxonomy will need to be extended with more precise subclasses appropriate to the types of materials being preserved, the specific objectives of an implementation, and the preservation and access policies that need to be applied. Classes can be extended ad hoc as required by the needs of different situations. PaaST also describes the possibility of a comprehensive approach to defining classes, articulating Preservation Rules and Conditions, establishing reporting requirements, etc. through a PaaST Profile.

1.2. What: Objects, Classes, Class Hierarchies and Collections

The things referenced in PaaST requirements are either classes, taxonomic arrangements of classes, or associations linking classes.   The classes and their associations are described in the chapters that precede the requirements. These descriptions are supplemented with Unified Modeling Language (UML) class diagrams. The diagrams offer a more structured and precise view of what is involved in implementing the requirements than can be done readily using natural language. The diagrams present views of the PaaST domain model. The domain model describes the real world entities involved in digital preservation and their relationships. The model includes one class of entities, DigitalComponents, that can exist independently of a Preservation Environment. DigitalComponents are the bit streams that encode the content and form of the things being preserved and enable their instantiation. The domain model reflects the perspective of archival science and the results of other studies in the InterPARES project and related work, including standards and models for digital preservation. The domain model does not represent nor presume any particular solution to the requirements for digital preservation.

This document adopts UML definitions of ‘class’ and related terms. UML defines a class as “a set of objects that share the same specifications of features, constraints, and semantics.”[9] An object is an instance of a class. The Features, constraints and semantics of an object are specified in the description of its class. A Feature is either an attribute or an operation of an object.   An association is a relationship between classes that specifies how instances of classes are related to each other.

PaaST Requirements center around three classes, PreservationTarget, PreservationAction and PreservationManagementInformation. A PreservationTarget is something that is preserved. A PreservationAction is an action that impacts the preservation of one or more PreservationTargets. PreservationManagementInformation is information needed to carry out digital preservation, describing the actions taken and their impacts on PreservationTargets.

PaaST Requirements enable users to define additional classes and sets to accommodate different policies, objectives and conditions. Furthermore, the PaaST domain model does not prescribe Features of the classes it includes. Appropriate Features are described and occasionally

[9] Object Management Group. OMG Unified Modeling Language (OMG UML), Superstructure. Version 2.4. January, 2011. http://www.omg.org/spec/UML/2.4/Superstructure. p. 51.

recommended in the text, but leaving them unspecified in the domain model provides maximum flexibility in adapting the model to different objectives, needs and policies.

1.3. How: The Requirements

PaaST requirements are organized into "services," groups of related capabilities. At the top, they are grouped under two major headings that parallel the two basic classes, Preservation Action Services and Preservation Management Capabilities. Preservation Action Services prescribe actions directly related to preserving PreservationTargets. Preservation Management Capabilities prescribe actions that ensure that the performance of Preservation Actions achieves the objectives of preservation. Preservation Management Services both produce and use PreservationManagementInformation. The execution of Preservation Action Services frequently involves calls on Preservation Management Capabilities.

Requirements for information technology are commonly expressed as "shall" statements. PaaST requirements facilitate implementation in software by following the norm of having each "shall" statement define one and only one thing the technology must do. PaaST requirements, however, are expressed in the form, "shall provide the capability to…." This phrasing makes implementing the requirements optional. The Preservation Director has the option of deciding which requirements are implemented in any Preservation Environment.

1.4. The PaaST Domain Model

The requirements and the explanatory narrative include numerous specialized terms, which are defined in the glossary. Most of these terms are names of classes, or activities in the PaaST domain model. The domain model enriches the understanding of the requirements by identifying the most important entities and concepts needed in digital preservation and specifying their relationships. It is intended to supplement the requirements and thus provide a fuller picture of what is involved in digital preservation. It is not intended as a data model to guide technical solutions to satisfy the requirements.

While it is extensive, the domain model is articulated at a fairly high level of generalization. A Preservation Environment will require both extension and greater specification of both the requirements and the model. The requirements explicitly support extensibility in several ways. Requirements for class management provide for defining subclasses of the classes described herein, as well as for introducing other classes that are not specializations of any of the current classes. PaaST also adopts the mechanism of attributable object from the Object Management Group's Records Management Services specification, enabling differentiation at the level of individual instances of classes.[10]

The domain model uses the Unified Modeling Language (UML) specification, version 2.4.[11] This document follows the UML convention of formulating names of classes in Upper Camel Case, capitalizing each word in the name and not separating words with spaces. Similarly, it follows the UML convention expressing attribute names in Lower Camel Case, where the first word in an attribute name starts in with lower case and the second and later words are capitalized with no spaces between words.

[10] Attributable objects are described in section 10.6.3, Attributable Objects.

[11] UML Superstructure Specification. v. 2.4.

Several UML class diagrams are included in the following chapters to complement the narrative. A brief description of the UML icons used in the diagrams is provided in Appendix 1. These diagrams are included for illustration and, thus, do not constitute a complete expression of the domain model or even of the model elements that appear in the diagrams. The diagrams do not show navigability of associations because the model is a domain model, while navigability is an implementation issue.[12] Also, the diagrams do not show operations and they include attributes only when needed to illustrate statements in the related requirements or text. Nevertheless, all classes and actions in the domain model are included and defined in the glossary.

The class diagrams indicate multiplicity. Multiplicity refers to the numbers of instances of one class that are or may be related to instances of another class. Multiplicity is shown at the ends of a line associating two classes. Thus, a one-to-one relationship is indicated by a “1” at both ends of the association. If no number is shown at an end of an association line, the multiplicity is undetermined, except in the case of a generalization relationship because, by definition, every instance of a specialization is also an instance of the generalization.

[12] Gonzalo Génova. Semantics of Navigability in UML Associations. https://gonzalogenova.files.wordpress.com/2015/02/semantics-of-navigability.pdf

2. Glossary

The PaaST Glossary includes definitions of all classes of objects referenced in PaaST requirements, the types of capabilities and actions used to implement the requirements, the parties involved in preservation, and special terms, such as ‘Preservation Environment,’ used to describe the approach to preservation adopted in PaaST. Terms related to classes of objects, capabilities, actions, parties and special terms defined in the glossary are capitalized in the following text. Conforming to the UML 2.0 specification, names of classes that consist of more than one word have no spaces between the words, and names of attributes that consist of more than one word also do not have spaces between the words; however, the first letter in the name of an attribute is not capitalized. Furthermore, ‘class,’ ‘instance,’ and ‘object,’ though used frequently, are not defined in the Glossary, but are used as specified in UML.

Terms defined in other documents, notably standards, that are capitalized in those documents, retain their capitalization, but are either preceded by the name or acronym of the other document or shown in quotation marks with the source document referenced in a footnote.

Term	Definition
Access	A capability that enables an Actor to receive PreservationTargets, HeuristicInformation or PreservationManagementInformation, or to exercise interactive functionality that implements one or more PaaST requirements
AccessData	PreservationActionData about performance of an Access capability.
Access Service	The set of capabilities that enables an Actor to receive information resources or to interact with the Preservation Environment.
Actor	A person or a system that interacts with a Preservation Environment using any capabilities defined in the PaaST requirements.
ActualRole	A PreservationRole that an Actor exercises when a PreservationRule is applied.
ArchivalPermanentFeature	A specialization of PermanentFeature that belongs to a PreservedRecord or PreservedRecordCollection.
AreaOfActivity	The sphere of acts or actions in which a RecordsCreator used a Record.
Assessment	A Preservation Management activity that involves the inspection of one or more PreservationTargets or parts thereof and the evaluation of PreservationManagementData.
AssessmentStatus	TargetState data describing the results of performing a Preservation Assessment.

Term	Definition
AttributableClass	A class that may be associated with a DataProfile that defines one or more attributes which may be present in some, but not necessarily all instances of the class.
AttributableObject	An instance of an AttributableClass.
AuthorizedRole	A PreservationRole an Actor is authorized to play in application of a PreservationRule
Benchmark value	The value of a PermanentFeature of a PreservationTarget that is used to determine if the expression of the Feature in a BinaryEncoding preserves the PermanentFeature. If the Feature is an attribute, the benchmark value is the value of the attribute. If the Feature is an operation, the benchmark value is either a postcondition that should be true when the invocation of the operation completes successfully, or the value of a parameter returned by the operation.
BinaryEncoding	A specification of how an IntellectualEntity is digitally encoded and how the bit strings that contain that encoding should be processed in order to instantiate the IntellectualEntity in a form that enables it to convey information as intended.
Capability	An action or set of actions performed to satisfy one or more PaaST Requirements. *Note*: A Capability may act on any data object, DigitalComponent or PreservationManagementInformation. Capabilities may be combined into Services.
Collection	A specialization of IntellectualEntity consisting of an IntellectualEntity that contains other IntellectualEntities.
ComponentDescription	A part of a BinaryEncoding that associates a DigitalComponent with an IntellectualEntity, describing how the component is used in the BinaryEncoding to instantiate the IntellectualEntity the encoding represents.
ComponentPermanentFeature	A specialization of PermanentFeatureExpression that describes how a PermanentFeature is expressed in a DigitalComponent.
ContentComponent	A DigitalComponent that encodes all or part of the content of an IntellectualEntity. It may also contain other types of data, such as data related to the instantiation of the content or data identifying related Software- or InstantiationObjects. In the case of a self-executing file, it might contain software.
ContentComponentDescription	A specialization of ComponentDescription that describes a ContentComponent.
ControllingObject	An object that constrains the execution of a PreservationRule either by a requirement that it exist or not exist or that one or more of its Features serve as parameters or conditions for rule execution.

Term	Definition
DataFormatRegistry	A specialization of a DigitalTypeRegistry describing the encoding of data files.
DataProfile	A profile of attribute definitions that may apply to AttributableObjects under organizational, ad hoc, or de jure standards or conventions.
DataProfileAttributeDefinition	A member of a DataProfile that describes an attribute that is applicable to a specific AttributableClassType.
DigitalComponent	A bit string that is part of a BinaryEncodings and is used to instantiate the IntellectualEntity the encoding represents.
DigitalTypeRegistry	A database describing types of DigitalComponents
DigitalTypeRegistryEntry	A description of a single type of DigitalComponent in a DigitalTypeRegistry.
Document	A persistent and indivisible unit of information that has a fixed form and stable content.
HeuristicAssociation	An association class that associates an instance of HeuristicInformation with a PreservationTarget.
HeuristicInformation	HumanReadableInformation that supports the discovery, understanding, evaluation or use of one or more PreservationTargets.
HumanReadableIntellectualEntity	A specialization of IntellectualEntity in a form that is intelligible to a person.
Information Management	Capabilities that support creating and maintaining PreservationManagementInformation, about controls, objects, processes, and actors. PaaST capabilities include categorizing information, extracting data from information sources, generating data from the execution of a preservation action, associating data with IntellectualEntities or their BinaryEncodings, and collecting data generated in inspection and verification of objects.
Initial Source	A Party that negotiates a Preservation Agreement with a Preservation Director designating Preservation Targets to be preserved and the terms of their preservation and access.
Inspection	An assessment activity that involves examination, by automated methods or human observation, of one or more PreservationTargets.
InstantiationComponent	A DigitalComponent that specifies aspects of the structure, semantics or presentation of an IntellectualEntity and that is applied, by a SoftwareComponent, to one or more ContentComponents to instantiate the IntellectualEntity.
InstantiationComponentDescription	A specialization of ComponentDescription that describes an InstantiationComponent.

Term	Definition
InstantiationObjectRegistry	A specialization of DigitalTypeRegistry describing types of data objects that may function as InstantiationComponents.
IntegrityFeature	A PermanentFeature of a PreservationTarget related to the identification and existence of the elements that constitute the PreservationTarget or to the fact that each of them remains whole and unaltered.
IntellectualEntity	A artifact that is intended to communicate information. *Note*: 'Artifact' as used here includes objects that are directly or indirectly created by humans, such as data that capture details of human interactions with social media or online systems, data generated by environmental sensors and the outputs of artificial intelligence systems.
Item	A specialization of IntellectualEntity that does not contain any other IntellectualEntity.
Local Preservation Environment	The Preservation Environment used by a single Preservation Service Provider.
MachineReadableIntellectualEntity	A specialization of IntellectualEntity in a form suitable for processing by a computer.
ManagedRecord	A Record that has attributes derived from a records management system in which it was kept.
ManagementSet	A set of objects created to support the accomplishment of one or more management objectives. The members of a ManagementSet may belong to different classes.
Manifestation	A concrete, physical implementation of an IntellectualEntity capable of communicating information as originally intended.
ManifestationDescription	A description of how the DigitalComponents in a BinaryEncoding should be processed to generate a Manifestation of the IntellectualEntity the BinaryEncoding represents.
ManifestPermanentFeature	A specialization of PermanentFeatureExpression that is expressed in a Manifestation.
NoticeOfIntentToSubmit	A message from a Submitter to a Preservation Service Provider informing the Provider of the Submitter's intention to transmit a Submission.
Party	A person or organization acting in, or authorized to act in, a specified capacity.
PermanentFeature	A Feature of a PreservationTarget that must remain invariant.
PermanentFeatureExpression	A description of how a PermanentFeature is exhibited in a Manifestation or a DigitalComponent of a PreservationTarget. The description must enable verification of the PermanentFeatureExpression.

Term	Definition
PreservationAction	An action that implements one or more PaaST Requirements and either directly impacts or verifies the preservation of one or more PreservationTargets. It may consist of a single step or multiple steps.
PreservationActionData	PreservationManagementData describing a PreservationAction, the Parties involved, the PreservationTargets and HeuristicInformation that are produced, altered, deleted or used in the action, and the results or outcomes of the action, including any problems encountered and their resolution.
PreservationActionReport	A PreservationManagementDocument describing the performance of one or more PreservationActions.
PreservationActionSet	A set of PreservationTargets defined to facilitate executing a PreservationAction or Actions collectively on all members of the set.
Preservation Agreement	A Preservation Commitment voluntarily established between an Initial Source and a Preservation Director to preserve certain Preservation Targets and specifying terms and conditions applicable to their preservation.
Preservation Assessment	An Inspection and/or an evaluation. A Preservation Assessment involves one or more PreservationActions that assesses the actual status of one or more instances of PreservationManagementInformation, DigitalComponent, ManifestationDescription, Manifestation or compares related PreservationManagementData, or compares PreservationManagementData with the DigitalComponent or Manifestation they describe.
PreservationAssessmentData	PreservationActionData related to the performance of a PreservationAssessment
PreservationAssessmentReport	A PreservationManagementDocument describing the results of a PreservationAssessment
Preservation Change	A Preservation Action that produces a new BinaryEncoding of a PreservationTarget.
PreservationChangeData	PreservationActionData about execution of a Preservation Change.
Preservation Change Service	Capabilities for managing, implementing, documenting and evaluating Preservation Changes.
PreservationCollection	A specialization of Collection whose members are PreservationTargets.
Preservation Commitment	An agreement, obligation or intention to preserve one or more PreservationTargets.

Term	Definition
PreservationDescription	TargetState data that describe the preservation of a PreservationTarget other than as defined in Assessment-, Storage- and SubmissionStatus.
Preservation Director	A party that has authority to direct the preservation of digital materials.
Preservation Environment	An implementation of PaaST requirements, including the set of PreservationTargets that are preserved under the same PreservationRules, documented in a comprehensive body of PreservationManagementInformation, together with the technological infrastructures and tools used in their preservation. The Preservation Environment may include separate, different and independently managed hardware and software used by different Preservation Service Providers. Separate implementations are called local Preservation Environments.
Preservation Intention	A Preservation Commitment unilaterally articulated by a Producer or Initial Source to preserve one or more PreservationTargets. A Preservation Intention may or may not identify the Preservation Director responsible for deciding how preservation is carried out or the Preservation Service Provider responsible for carrying out preservation.
PreservationItem	An intellectual entity that contains no other intellectual entities and is designated for preservation.
PreservationManagementData	PreservationManagementInformation in the form of structured data about PreservationTargets, PreservationActions, PreservationRules and Actors.
PreservationManagementDocument	PreservationManagementInformation in the form of a human readable object whose contents are about preservation or relevant to preservation management.
PreservationManagementInformation	Information used to manage preservation, including structured data and other information about PreservationTargets, Binary Encodings, DigitalComponents, Actors, and Preservation Actions.
PreservationNetwork	A set of objects related to the preservation of a PreservationTarget, including its BinaryEncodings, HeuristicInformation that supports its comprehension and use, other PreservationManagementInformation about it, and any other PreservationTarget associated with the PreservationTarget that is the focus of the network.
Preservation Obligation	A Preservation Commitment imposed by a Preservation Director on a Producer or Initial Source over whom the Preservation Director has authority.
PreservationProblem	A condition that fails to satisfy one or more Criteria for Success in the implementation of a PreservationRule.

Term	Definition
PreservationRole	An enumeration that identifies the modes in which Actors participate in, or may be authorized to participate in, application of PreservationRules
PreservationRule	An executable business rule that constrains implementation of a capability articulated in one or more PaaST requirements and governs one or more objects and/or actions.
Preservation Service	A group of related capabilities that accomplish digital preservation, including bringing Preservation Targets into and under the control of a Preservation Environment, ensuring Preservation Targets are stored in a way that guarantees their integrity and accessibility, implementing changes in Preservation Targets or in the technology needed to operate on and provide access to them, and providing copies of Preservation Targets and access to information about them to authorized parties.
Preservation Service Contract	An agreement between a Preservation Service Provider and a Preservation Director for the provision of one or more preservation services.
Preservation Service Provider	A party that performs one or more preservation services.
PreservationStorage	The part of a Preservation Environment where DigitalComponents are stored.
Preservation Storage Service	Capabilities that ensure that Preservation Targets are not lost or corrupted in storage and remain retrievable from storage.
PreservationStorageData	PreservationActionData about performance of a Preservation Storage capability.
PreservationTarget	An IntellectualEntity that is designated for preservation.
PreservationTargetComponentDescription	A specialization of ComponentDescription that describes how a DigitalComponent is part of a BinaryEncoding of a PreservationTarget and, if the component expresses a PermanentFeature, specifies how it is expressed in the component.
PreservationTargetData	PreservationManagementData about a PreservationTarget.
PreservationTargetManifestation	A specialization of Manifestation that implements a PreservationTarget and exhibits all ManifestPermanentFeatures that are expressed in the Manifestation.

Term	Definition
PreservationTargetManifestationDescription	A specialization of ManifestationDescription that describes how a PreservationTarget is instantiated in a RuntimeVersion or a Rendering. PreservationTargetManifestationDescription differs from the generic ManifestationDescription in that it describes the ManifestPermanentFeatures of the PreservationTarget that are exhibited in the RuntimeVersion or Rendering.
PreservedRecord	A PreservationTarget that is a Record.
PreservedRecordCollection	A PreservationTarget that is a RecordAggregate.
ProblemHandling	Capabilities for recognizing problems related to objects, processes, parties and information; for characterizing and capturing data about problems, rating the severity of problems, assigning responsibility for problem resolution, receiving responses to problem reports, and tracking attempts to resolve problems.
ProblemHandlingInstruction	A business rule that specifies what is to be done when a problem is identified in executing a PreservationRule.
ProblemHistoryData	PreservationActionData related to the discovery, handling and resolution of a problem related to a Preservation Action, PreservationTarget, or PreservationManagementInformation.
ProblemReport	A document identifying and describing a problem discovered in the execution of a Preservation Action.
ProblemResolutionReport	A document describing actions taken in response to an identified problem.
Producer	A Party that created or originally owned or controlled one or more PreservationTargets.
Record	A type of IntellectualEntity that was made or received in the course of a practical activity as an instrument or a by-product of such activity, and set aside for action or reference.
RecordAggregate	A Collection of Records established and maintained by a RecordsCreator.
RecordCategory	An association class that associates a ManagedRecord with a RecordAggregate.
RecordsCreator	A person or organization that produces or acquires Records in the course of a practical activity.
RecordDescription	A specialization of HeuristicInformation that contributes to the understanding of Records, RecordAggregates, or their context of use.
RecordsManagementData	A specialization of PreservationManagementData that describes one or more Records or RecordAggregates.

Term	Definition
RecordsManagementDocument	A specialization of PreservationManagementDocument that describes one or more Records or RecordAggregates.
Rendering	A specialization of a Manifestation that outputs a HumanReadableIntellectualEntity; for example, on a display device, audio speaker, or in hard copy.
RenderingDescription	A specialization of ManifestationDescription that describes a Rendering.
Reporting	Capabilities that support producing, sending, and managing reports about objects, processes and problems.
Requirement	A specification of a capability that a Preservation Environment must have (functional) or a non-functional property of that environment needed for the purpose of preservation.
RuledAction	A PreservationAction whose execution is governed by a PreservationRule.
RuledObject	An object governed by a PreservationRule.
RuleDomain	The identification of the classes of objects and actions involved in application of a PreservationRule, other objects that function as controls in application of the rule, and the Actor(s) authorized to participate in rule execution, and their AuthorizedRole(s).
RuleResult	An output or outcome of a RuleTask that is significant for determining if the rule was applied successfully.
RuleTask	An action that implements a PreservationRule and produces one or more RuleResults.
RuntimeVersionDescription	A specialization of ManifestationDescription that describes a RuntimeVersion.
RuntimeVersion	A specialization of Manifestation that is produced by loading the DigitalComponents in a BinaryEncoding into a computer for processing or output.
RuntimePermanentFeature	A specialization of ManifestPermanentFeature describing how a PermanentFeature is expressed in a RuntimeVersion.
Service	A set of related Capabilities.
Set	A group of objects that have a whole/part relationship where the whole has different Features than the part and, if there is more than one class of part, the different parts have different Features. In UML, the classes that belong to a set have either an aggregation or composition association.

Term	Definition
SoftwareComponent	A DigitalComponent containing software that operates on ContentComponents and any associated InstantiationComponents in the same BinaryEncoding to instantiate an IntellectualEntity.
SoftwareComponentDescription	A specialization of ComponentDescription that describes a SoftwareComponent.
SoftwareRegistry	A specialization of DigitalTypeRegistry describing types of SoftwareComponents.
StorageStatus	TargetState data describing the storage of DigitalComponents
Submission	A transfer of one or more IntellectualEntities and Submission Information to a Preservation Service Provider.
SubmissionData	A specialization of PreservationManagementData describing a Submission or its contents.
SubmissionDocument	A PreservationManagementDocument describing a Submission and its contents.
SubmissionInformation	Information about a Submission or about all or part of its contents.
Submission Process	A PreservationAction related to the ingest of one or more PreservationTargets into a Preservation Environment, including review of a NoticeOfIntentToSubmit, review of SubmissionInformation, review of PreservationTargets included in a Submission, and determination whether to accept a Submission.
SubmissionProcessingData	PreservationActionData about performance of a Submission Process.
Submission Processing Service	Capabilities for perfoming Submission Processes.
SubmissionSet	A set of related IntellectualEntities, Binary Encodings, and DigitalComponents transferred to a Preservation Environment together.
SubmissionStatus	A specialization of TargetState comprising data about the authorization, Submitter, transmission, contents or conditions of a Submission, exclusive of of data about the processing of a Submission, which is a subclass of PreservationActionData.
Submitter	A party that sends one or more Submissions to a Preservation Environment.
SuccessCondition	A property that a RuledObject should have or a postcondition of a RuledAction that should be true in order to satisfy a SuccessCriterion of a PreservationRule. The specifics of the SuccessCondition are determined by applying the SuccessCriterion to the RuleDomain of the PreservationRule.

Term	Definition
SuccessCriterion	A standard, principle or reference point used to determine whether an execution of a PreservationRule satisfies the rule.
TargetDescription	A specialization of PreservationTargetData that describes the persistent attributes and operations of a PreservationTarget. TargetDescription has specializations: PreservationTarget, BinaryEncoding, ComponentDescription, ManifestationDescription, PermanentFeature, PermanentFeatureExpression, and HeuristicInformation.
TargetProblem	A specialization of TargetState describing a problem with a specified PreservationTarget
TargetState	A specialization of PreservationTargetData that describes the state of a PreservationTarget at a given time. TargetState comprises the set of values of the attributes of the PreservationTarget, its parts and its associations with other objects that may change over time. TargetState has specializations SubmissionStatus, StorageStatus, AssessmentStatus, VerificationStatus, TargetProblem and PreservationDescription.
UniquenessFeature	One of a set of PermanentFeatures that uniquely distinguishes a PreservationTarget from any other.
Verification	One or more PreservationActions that determine whether a PreservationTarget has been preserved successfully
VerificationData	PreservationActionData related to the performance of a Verification.
VerificationStatus	PreservationStateData about a PreservationTarget output from a Verification.

3. Who's Involved

Preservation involves persons or organizations, acting in different capacities. PaaST distinguishes the parties involved according the capacities in which they act. The same person or organization may act in more than one capacity and, conversely, different persons or organizations may act in a given capacity.

3.1. Capacities

PaaST defines six capacities, as follows:

- Producer is a person or entity that created or originally owned or controlled one or more PreservationTargets. The Producer's role as Producer ends once the PreservationTarget has been produced. The Producer might not be involved in the decision that an IntellectualEntity should be a PreservationTarget. That decision is made by the Initial Source. The person or organization who was the Producer, nevertheless, could act in one or more other capacities.
- Initial Source has possession of and/or control over Preservation Targets before they are sent to a Preservation Environment; has authority to decide or agree to what is to be preserved and the terms under which they will be preserved.[13]
- Submitter sends Submissions and related information to a Preservation Environment.
- Preservation Director directs the preservation of digital materials.
- Preservation Service Provider has and provides technological capabilities (infrastructures, applications and tools) that satisfy PaaST requirements.
- Access Client accesses automated capabilities in the Preservation Environment. Access encompasses access to functionalities that implement PaaST requirements, as well as to materials being preserved and information about them.[14] An Access Client could be an outside party but it could also be the party that is the Preservation Director or the Preservation Service Provider, or an employee or other representative of one of them.

A Preservation Director decides which PaaST requirements are implemented in any Preservation Environment. This flexibility enables different approaches to fulfilling requirements, for example by performing some PreservationActions in-house while contracting with one or more Preservation Service Providers for other capabilities.

In a scenario where preservation is accomplished in the Cloud, one or more Initial Sources reach Preservation Agreements with a Preservation Director who accepts responsibility for preservation of digital information assets. The Preservation Director then enters into a Preservation Service Contract with a Preservation Service Provider who provides the technological resources necessary for preservation and carries out preservation activities as directed by the Preservation Director and,

[13] An Initial Source may be only indirectly involved in activities that address PaaST requirements. An Initial Source may establish an agreement with a Preservation Director for preservation of its records or other information objects it has. The Preservation Director may independently contract with one or more Preservation Service Providers to fulfill some or all of the requirements for preservation.

[14] An *Access Client* may also be only indirectly involved in PaaST activities. If the *Preservation Director* is an archival institution, it may have a system or service for responding to *Access Clients* who desire access to any of the records it preserves. The digital records preserved under contract with a *Preservation Service Provider* may constitute only a subset of the archives' holdings and the archives' access system may be the direct interface for all *Access Clients*, passing requests and responses between clients and the *Preservation Service Provider*.

presumably as specified in the Preservation Service Contract. However, an Initial Source could retain authority over the materials it wants preserved and thus also act as a Preservation Director. Also, a Preservation Director could provide some preservation services in house and only contract with a Preservation Service Provider for a limited set of preservation activities. Alternatively, the Preservation Director might rely on Preservation Service Providers for all preservation services, but rely on one provider for basic ongoing services, such as processing of Submissions and storage of Preservation Targets, while awarding other contracts for specialized services, such as format conversion.

Submitters and Access Clients have secondary roles in that their rights and responsibilities could be specified in Preservation Agreements or contracts without their involvement. For example, when the Initial Source is a government agency, its Preservation Service Agreement with the government archives might provide for several operating units to transmit Submissions of specified series of records for preservation. Each of these units would be a Submitter. Similarly, the Preservation Director's contract with a Preservation Service Provider might designate classes of Access Clients who may request and receive copies of Preservation Targets, and it may limit their rights by defining access restrictions for privacy, proprietary or other reasons.

3.2. Actor

An Actor is a role played by an entity that interacts with a Preservation Environment using any capability defined in the PaaST requirements. An Actor will often be a person but could be an external system or application.

> *For example, where similar Submissions from a single source occur repeatedly, an external system might be programmed to transmit them automatically. The external system would be an Actor in the Submission.*

A person acting in one or more of the capacities defined in section 3.1, may perform actions that would qualify them as Actors, but many actions entailed by these capacities are external to the Preservation Environment or do not entail the use of PaaST capabilities.   Such actions include defining, implementing, managing, overseeing and evaluating preservation, rather than carrying it out.

In contrast, Actor is a class in the PaaST domain model and is specifically referenced in the requirements. Consistent with the UML specification, "an actor does not necessarily represent a specific physical entity but merely a particular facet (i.e., "role") of some entity that is relevant to the specification of its associated use cases. Thus, a single physical instance may play the role of several different actors and, conversely, a given actor may be played by multiple different instances."[15]

There are four predefined roles an Actor may have in activities that implement PaaST requirements.

- Performer: an entity actively involved in executing one or more preservation capabilities,
- Authorizer: an entity whose approval is needed for an action to be executed in specified circumstances.
- Problem Resolver: an entity responsible for resolving a problem that occurs in preservation activities, and
- Approver: an entity who determines if the outcome of an action, including resolution of any problems, is acceptable.

[15] UML Superstructure Specification, v.2.4, pp. 604-5.

Other roles might be defined to satisfy particular circumstances, in accordance with the AttributableClass concept described in section 9.5.3.

4. What's Involved: PaaST Classes

Classes in the PaaST domain model represent preservation entities, activities, norms, and management information. Each of these groups has one or two central classes. Preservation entities comprise the things that are preserved and constituent objects used in preserving them. The central class of preservation entities is IntellectualEntity, an object that is intended to communicate information. IntellectualEntity is a generalization that has two main specializations, PreservationTarget and HeuristicInformation, which define different roles of IntellectualEntities in a Preservation Environment. Preservation activities are things done for the purpose or as a consequence of preservation. Preservation activities includes two central classes: PreservationAction and Actor. A PreservationAction is something that is done for preservation. An Actor is a person or external system or application involved in carrying out a PreservationAction.[16] Preservation norms control what is preserved, what actions are performed and the conditions that govern them. Preservation norms include two central classes, Requirement and PreservationRule, which customizes Requirements in a specific Preservation Environment. PreservationManagementInformation is generated, updated and used in actions. PreservationManagementInformation is a general class that contains many subclasses. Figure 1, Preservation Overview, is a UML class diagram that models the basic relationships among these central classes. All of these classes are discussed in greater detail in subsequent sections.

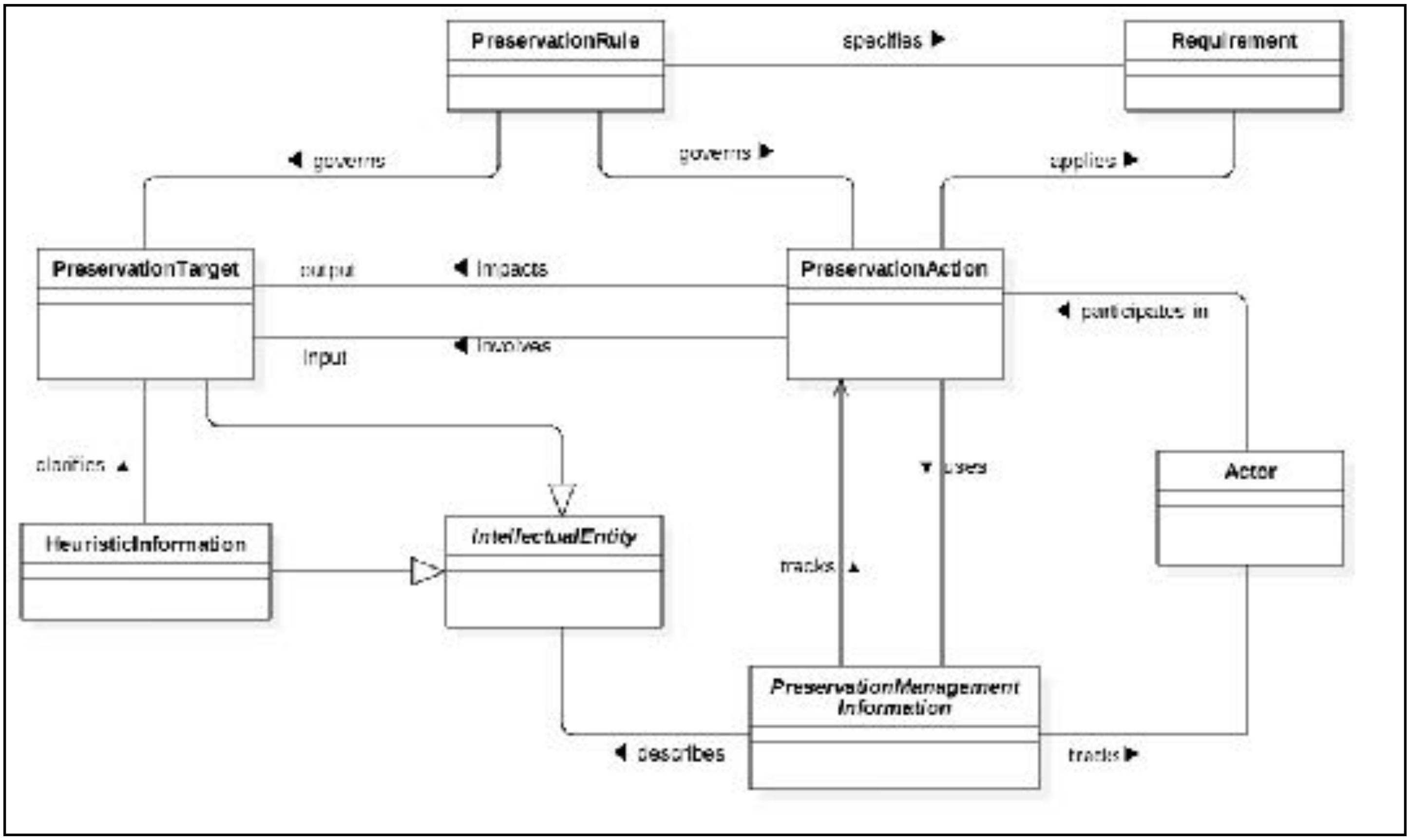


Figure 1, Preservation Overview

[16] "External" here refers to a system, application, or device that is not integrated into the technological tools set normally used for preservation in a Local Preservation Environment. An external tool may be under separate administrative control or may be one used ad hoc within a Preservation Environment.

Apart from the computing platforms on which preservation is carried out, everything involved in digital preservation is some type of information object. As mentioned in section 1.2, PaaST Requirements involve two key classes of information objects, the things that are preserved, designated as PreservationTargets, and the information needed to preserve them, labeled PreservationManagementInformation. Preservation is accomplished by means of a third basic class, Preservation Action.

The Requirements provide the basis both for the actions that carry out digital preservation and the objects involved. A PreservationAction applies one or more Requirements. A Requirement is a specification of a capability that a Preservation Environment must have (functional) or a non-functional property of that environment needed for the purpose of preservation. PaaST Requirements are specifically formulated such that an implementation must provide the desired capabilities, but the use of the capabilities depends on positive affirmation by the Preservation Director; moreover, the implementation may be governed by one or more PreservationRules that specify when and how Requirements apply. As explained in the Introduction, Paast Requirements are formulated to support preservation of a broad range of digital information, for different purposes and in different contexts. The Requirements can be more precisely specified for different types of objects, purposes and contexts by PreservationRules established by the Preservation Director. A PreservationRule is a business rule that governs execution of one or more Preservation Actions and/or the existence or state of one or more PreservationTargets. A rule can specify a class or subsets of a class of objects to which a Requirement applies, conditions under which the Requirement should be applied, and the criteria for judging success. For example, a PreservationRule could specify how a Requirement related to submitting PreservationTargets applies in the case of PreservationTargets from a particular Initial Source, such as identifying the Submitters who are authorized to send the PreservationTargets. Any Preservation Environment will have many PreservationRules and different Preservation Environments may have substantially different rules. PreservationRules support both consistency and optimal automation in implementing the capabilities articulated in PaaST requirements. A PreservationRule can trigger the execution of a PreservationAction. Such a rule should identify the circumstances when the capability is necessary or advantageous and prescribe what should be done when those circumstances occur.

Many different PreservationRules could specify a given Requirement. Similarly, a PreservationRule could specify more than one Requirement.

> *For example, a PreservationRule could identify both a set of PreservationTargets authorized for Submission and the Submitters authorized to send them.*

The focus of digital preservation is the class, PreservationTarget. A PreservationTarget is an IntellectualEntity that is designated to be preserved for a length of time that entails PreservationActions. In a Preservation Environment, an IntellectualEntity is a description of an artifact that is intended to communicate information. An IntellectualEntity is instantiated in a Manifestation.

Figure 1 also includes a second role of IntellectualEntity, HeuristicInformation. HeuristicInformation is human-readable information that assists people to understand and interpret the information conveyed by a PreservationTarget. These two roles of IntellectualEntity are not mutually exclusive. An IntellectualEntity could be both a PreservationTarget and HeuristicInformation. A HeuristicInformation object that is not designated as a PreservationTarget might be modified to better serve the purpose of clarification, and the modified version could replace the earlier one. But if the HeuristicInformation was itself designated as a PreservationTarget, it should not be modified in any way that would affect its preservation.

As a general guide, HeuristicInformation objects that were created in the same context as the PreservationTargets they clarify should be designated as PreservationTargets and associated with the other targets they clarify in a PreservationCollection (q.v. section 4.6). Whether HeuristicInformation that was created after the fact or by different Producers should also be designated as PreservationTarget depends on a variety of factors, such as the relationships of the Producers of the two types of IntellectualEntities, the relationships of the activities in which they were produced, and the characteristics of the Collection or PreservationNetwork to which the clarified IntellectualEntity belongs.

A logical approach to specifying the different Features of IntellectualEntity, PreservationTarget and HeuristicInformation is that Features related to the form and content of an IntellectualEntity be specified in instances of that class; the context and purpose for designating an IntellectualEntity as a PreservationTarget, specifically with a view to how they constrain or otherwise impact PreservationActions related to that entity be specified in the class, PreservationTarget; while Features that identify what PreservationTargets HeuristicInformation clarifies and how it does so be included in that class.

A PreservationAction is an action that implements one or more PaaST Requirements and either directly impacts or verifies the preservation of one or more PreservationTargets. An impact is a change of state of a PreservationTarget or one or more of its parts. Actions that produce state changes include initiation, alteration, and deletion. Initiation is either creation, such as when a data file is created by migration from an earlier format, or introduction, which happens when an existing object is submitted to a Preservation Environment from outside. A PreservationTarget that undergoes a change of state as a result of a PreservationAction is an output of that action. A PreservationAction may also involve a PreservationTarget as input.

For example, when a format migration is performed, the action uses an existing version of a PreservationTarget as the source for generating a new version in the desired format. If the new version is created solely in response to a request from an outside party, the older version is involved, but not impacted by the process. In contrast, if creation of the new version were accompanied by changing the status of the older version to obsolete, then the older version is an input to the PreservationAction and the outputs include both the new version and the obsoleted older version.

Thus, the same PreservationTarget may be both input and output in a PreservationAction, but that is not necessarily the case.

Other examples where a PreservationTarget is both input and output in the same PreservationAction occur in Submission Processing. A PreservationTarget that is included in a Submission is an input to Submission Processing. If no problems are found in Submission Processing, the object changes its state from received to accepted. Thus it is also output.

An example where a PreservationTarget is an input but not an output could occur when the PreservationAction is to verify that a Collection is complete. An action that simply checks that Items that should be part of the Collection are present in the Preservation Environment, uses the Items as inputs, but its output is only the verification that the Collection is complete. The output of the action is additional data supporting the assertion of completeness, but the completeness of the Collection is not altered by the action.

PaaST requirements define other actions besides PreservationActions, notably those that specify information processing without directly impacting or involving any PreservationTarget.

For example, when a Preservation Director contracts with a Preservation Service Provider, the Preservation Service Provider should create a database identifying the PreservationTargets that will be *preserved. While the database will contain data about the PreservationTargets, inputing the data does not directly impact any of them.*

Similarly, when an Actor performs a search to identify PreservationTargets of interest, the search results will be about PreservationTargets, but the search action in and of itself does not impact any PreservationTarget.

Given the focus of this document on preservation, it touches on actions that are not PreservationActions only incidentally in connection with PreservationActions or when needed in describing PreservationManagementInformation.

An Actor is a person or an external system that interacts with a Preservation Environment in the implementation of any capabilities defined in the PaaST requirements. Actors may participate in Preservation Actions in various roles. When a possible role is authorized by a PreservationRule, it is designated as an AuthorizedRole. The actual participation of an Actor in a PreservationAction is an ActualRole. When an authorization exists, the ActualRole should be the same as the AuthorizedRole.

PreservationManagementInformation is information used to manage preservation, including structured data and other information about IntellectualEntities, PreservationTargets, HeuristicInformation, their parts, Actors, and Preservation Actions, as shown in Figure 1. New or revised PreservationManagementInformation is generated in PreservationActions and existing PreservationManagementInformation may be used as input to those actions. An Actor may access PreservationManagementInformation either for the sole purposes of obtaining information or when participating in a PreservationAction.

Three of the classes in Figure 1, PreservationTarget, IntellectualEntity and PreservationManagementInformation, are abstract classes, indicated by their names being italicized. None of these classes is sufficiently specified to enable instantiation. All objects in these classes are members of a specialization of the more general class.

4.1. Types of IntellectualEntity

IntellectualEntities can be differentiated on the basis of their intended recipients. A HumanReadableIntellectualEntity is one that is intended for humans, while a MachineReadableIntellectualEntity is one that is intended to be processed by a computer. A natural language text is an archetypical example of a HumanReadableIntellectualEntity and software is an archetypical MachineReadableIntellectualEntity. IntellectualEntity is an abstract class. Every instance of an IntellectualEntity is either a MachineReadableIntellectualEntity or a HumanReadableIntellectualEntity. Both of these specializations are themselves abstract classes which are instantiated only in their subclasses. Figure 2 includes examples of specializations of both according to genre or form. These examples are for purposes of illustration. In many Preservation Environments specializations would probably be more extensive and more specific. They might make use of existing taxonomies or schemes for HumanReadableIntellectualEntities[17] or MachineReadableIntellectualEntities.[18]

In contrast, in a repository with a narrow range of holdings, such as a data center holding observational data from a scientific specialty, the overall range of specializations might be narrower, but might include more precise subclasses.

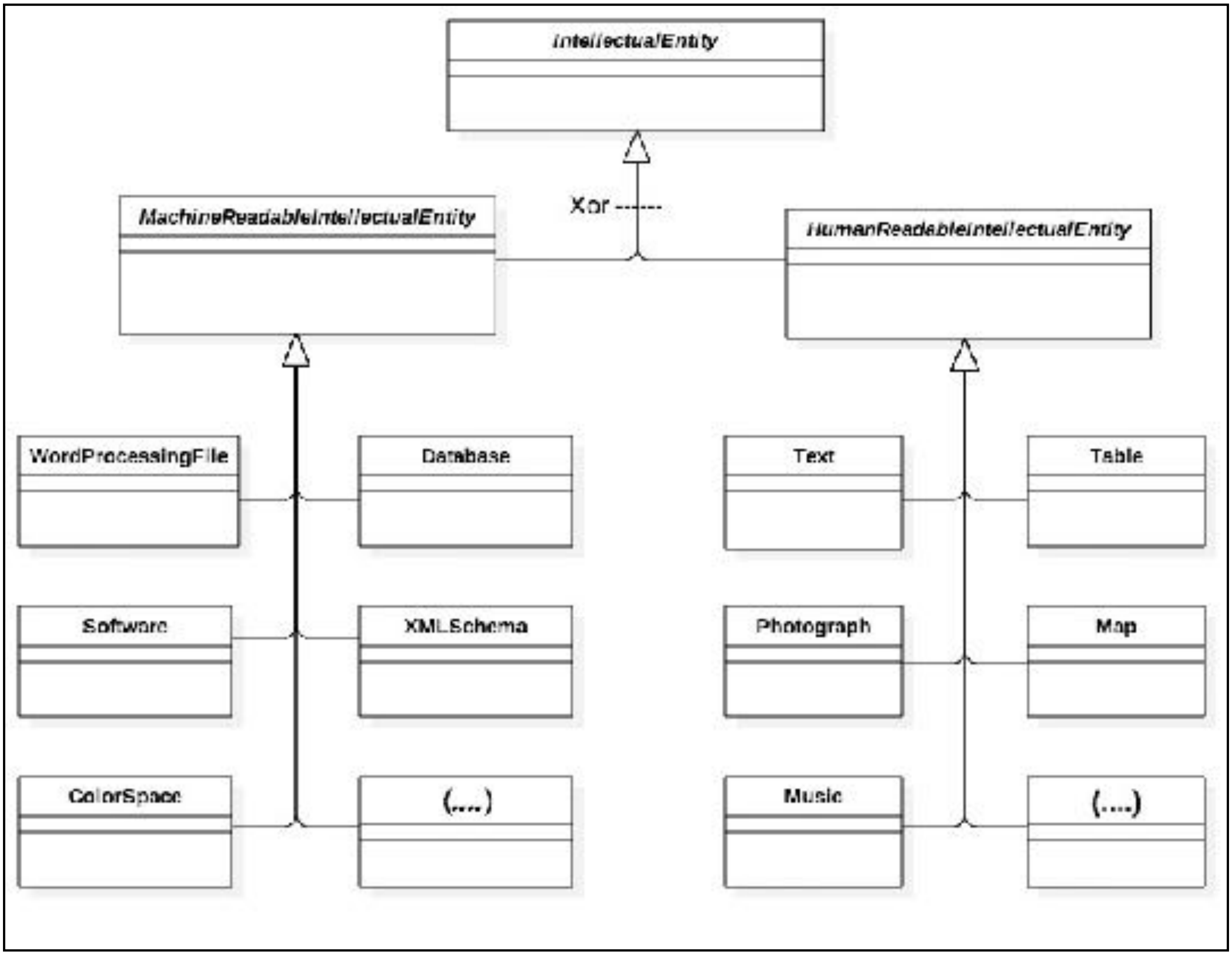


Figure 2, Types Of IntellectualEntity

The subtypes of MachineReadableIntellectualEntity shown in Figure 2 include word processing file, database, software, XML Schema, Dynamic Load Library, and undefined others, suggested by the class icon whose name is "…." The subtypes of HumanReadableIntellectualEntity shown include text, table, photograph, map, music and, again, undefined others.

[17] Getty Research Institute. Art & Architecture Thesaurus® Online: Visual and Verbal Communication. http://www.getty.edu/vow/AATHierarchy?find=&logic=AND&note=&subjectid=300264552. Library of Congress. Genre/Form Headings at the Library of Congress. https://www.loc.gov/catdir/cpso/genreformgeneral.html.

[18] Internet Assigned Numbers Authority (IANA), Media Types. http://www.iana.org/assignments/media-types/media-types.xhtml. National Archives (UK), PRONOM. https://www.nationalarchives.gov.uk/PRONOM/Default.aspx.

These examples clearly illustrate that the distinction between human readable and machine readable does not depend on genre or content, nor on being encoded in digital form, but relates to the intended recipients.

A human readable display of a textual document could be generated from a word processing file. A table could be generated as a database view, and so on. On first glance, MachineReadableIntellectualEntity seems to include some specializations, such as software and color space, that can only be processed by a computer. While this is true in their native formats, software code can be printed as textual instructions and color spaces can be displayed in graph form.

Furthermore, there are software programs, such as optical character recognition, that can extract and analyze data from objects intended for human consumption. The distinction between human and machine readable, then, does not reflect what is possible, but what is intended. A color space, for example, enables the Rendering of a color photograph, but the color space itself is designed to enable a computer to generate multicolored output.

It would be possible to define multiple subsets of HumanReadable- and MachineReadableIntellectualEntities that are orthogonal to each other; that is, each subset is determined on a basis that is independent of the basis of any other. Some possibilities include classifying IntellectualEntities according to:

- *the types of data they express: text, cartographic, photographic, audio/visual, etc.*
- *the forms of materials containing the information: books, manuscripts, databases, websites, etc.*
- *the sources that produced the information. For example, in a data center providing environmental data the subclasses of IntellectualEntity could be defined according to whether the data originated in land based stations, satellites, radar, weather balloons, et al.*

A taxonomy of types of IntellectualEntities in each Preservation Environment should be appropriate to the PreservationTargets and HeuristicInformation in the environment.

4.2. IntellectualEntity Composition

The class, IntellectualEntity, describes artifacts intended to convey information. The Features of an IntellectualEntity identify the artifact and describe how it conveys information to the intended human or computer receiver. For all IntellectualEntities, those Features include the types of information or data that comprise the content and how the content is structured, as well as the functionality involved in communicating information. PaaST Requirements, numbered 5.2.1*, support further refinement of the Features for different specializations of IntellectualEntity, as indicated in the description of Figure 2, Types of IntellectualEntity, above.

A BinaryEncoding specifies both how an IntellectualEntity is digitally encoded and how the bit strings that contain that encoding should be processed in order to instantiate the IntellectualEntity in a form that enables it to convey information as intended. The same IntellectualEntity could be encoded and even instantiated in several different ways.

For example, a textual document might be created in Word, distributed as a PDF file and displayed in HTML on a web page.

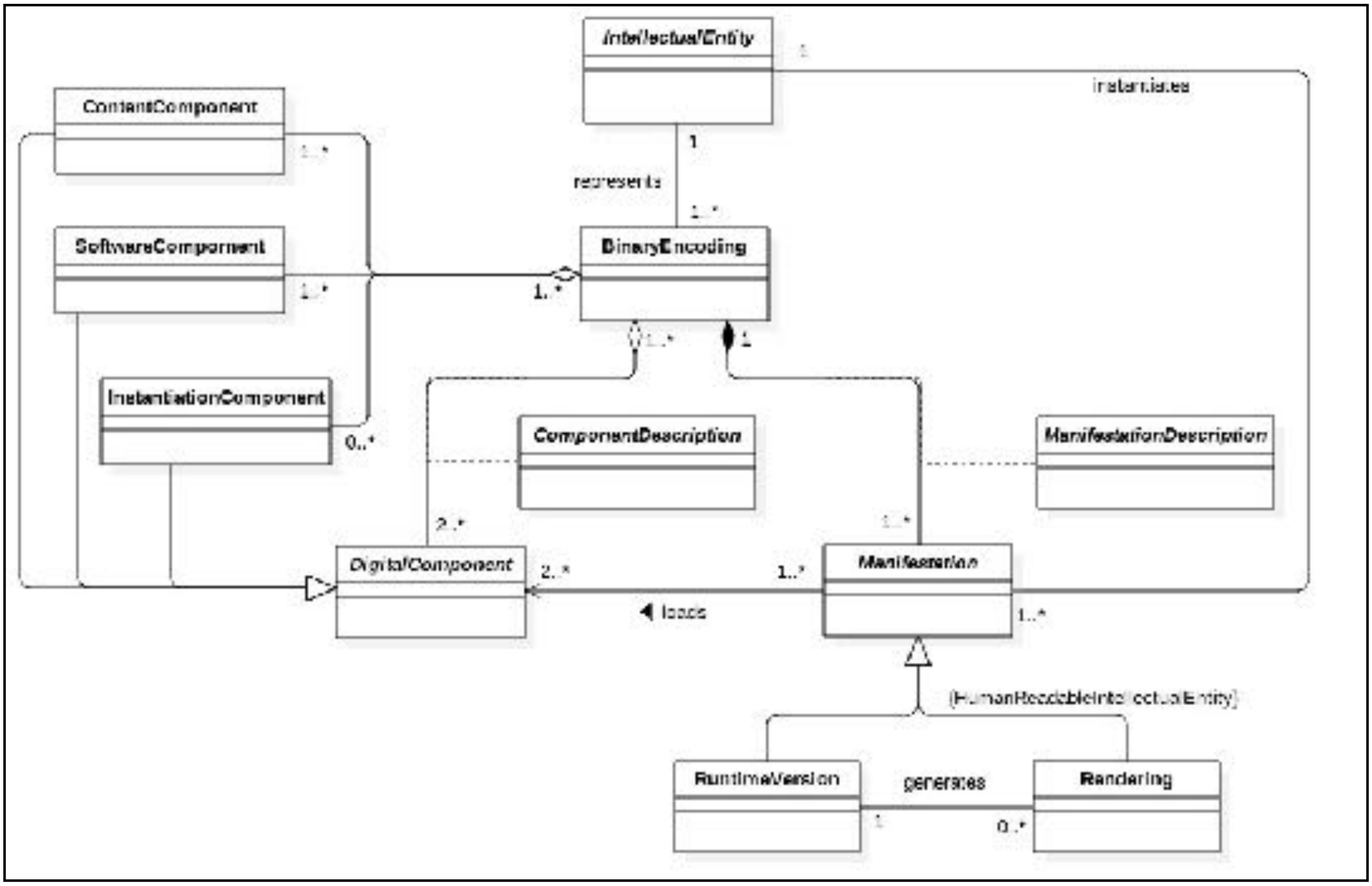
An IntellectualEntity might be a scanned image of a document on paper. The image constitutes a BinaryEncoding of the document. If an image of a text were processed through OCR, the resultant character encoded document would be a second BinaryEncoding.

The manifestations of a relational database might include numerous forms, reports and stored views.

A software program could be compiled into different executable code versions for different platforms.

As shown in Figure 3, Composition of IntellectualEntity, An IntellectualEntity must have at least one BinaryEncoding to be represented in a PreservationEnvironment. As described above, it can have more than one, but a BinaryEncoding can represent only one IntellectualEntity.[19]

Figure 3, Composition Of IntellectualEntity



A BinaryEncoding has two types of parts: Digital Component and Manifestation. A DigitalComponent is an ordered string of bits that encodes numbers, characters, signs, symbols, sounds, images or other graphics in a digital form that is both suitable for storage and appropriate to instantiate an IntellectualEntity. A Manifestation is a concrete, physical implementation of an IntellectualEntity capable of communicating information as originally intended..

DigitalComponent and Manifestation differ from other PaaST classes in that instances of other classes consist only of data about the things they describe. In contrast, DigitalComponents are the primary persistent means of maintaining IntellectualEntities over time and instantiating them when needed. They may be added to or, under appropriate circumstances, deleted from a Preservation Environment, but a Digital Component should never be altered. A DigitalComponent may be, but is not necessarily, stored as a separate file in a computer's file system. Alternatively, a

[19] An instance of an IntellectualEntity may be created in a Preservation Environment before any BinaryEncoding of it is submitted. But the IntellectualEntity cannot be preserved until at least one BinaryEncoding is in the Preservation Environment. Thus, the minimal cardinality of BinaryEncoding is one, even though there may be a period of time when what the encoding is remains to be determined.

DigitalComponent might be stored in a package, such as a TAR or ZIP file, or in a database. What is essential is not how a DigitalComponent is stored, but that each DigitalComponent is uniquely identified, retrievable, correctly associated with Binary Encodings and with other DigitalComponents and appropriately processed to instantiate the IntellectualEntities it encodes. A Manifestation is a physical materialization of an IntellectualEntity. In contrast to a DigitalComponent, a Manifestation is not stored and exists in a Preservation Environment only as long as it is a processor or on an output device.

Each BinaryEncoding should contain a set of compatible DigitalComponents that is sufficient to instantiate the IntellectualEntity it represents in proper form.

DigitalComponent is an abstract class that is instantiated in its three specializations:

- A ContentComponent encodes all or part of the content of an IntellectualEntity. It may also contain data related to the rendering of the content or data identifying related Software or InstantiationComponents. A BinaryEncoding must include one and may include more than one ContentComponent.

  Plain ASCII text and flat files include only content, while formats like PDF-A include data needed for instantiating the IntellectualEntity.

- A SoftwareComponent operates on ContentComponents and any associated InstantiationComponents in the same BinaryEncoding to instantiate an IntellectualEntity. The use of the right software is essential. Regardless of how well DigitalComponents are preserved in storage and transmission, using inappropriate software to instantiate an IntellectualEntity can corrupt it. A BinaryEncoding must include one and may include more than one SoftwareComponents.

  Even when the IntellectualEntity is software, it requires a compatible operating system to load.

- An InstantiationComponent specifies aspects of the structure, semantics or presentation of an IntellectualEntity and is applied, by a SoftwareComponent, to one or more ContentComponents to instantiate the IntellectualEntity. A BinaryEncoding may not include any InstantiationComponent but might include one or more.

  For example, a style sheet determines how an HTML file is displayed on a web page.

  A stored view specifies what data elements are selected from a large and complex database and, possibly, how some of them are processed for display.

A DigitalComponent may be part of many BinaryEncodings representing different IntellectualEntities.

The same SoftwareComponent could operate on any ContentComponent in its native format and an InstantiationComponent, such as a font library, XML schema or style sheet, could apply to many different Content Objects and be used by different Software Objects.

While in many cases a ContentComponent will be unique to an IntellectualEntity , a single ContentComponent could be part of several Binary Encodings. A photograph, for example, could be an IntellectualEntity by itself, but the picture could also be contained in a textual document. In that case, the ContentComponent that encodes the picture would be part of the Binary Encodings of both the textual the photograph itself and the document that includes it.

Thus, the persistence of a DigitalComponent must be independent of the existence of any BinaryEncoding. Therefore, DigitalComponent is associated with BinaryEncoding by aggregation, not composition. At a minimum, the set of DigitalComponents in a BinaryEncoding must include at least two DigitalComponents: one ContentComponent and one SoftwareComponent.

The same DigitalComponent could have different roles in different BinaryEncodings.

> *For example, an XML schema is used as an InstantiationComponent when applied to XML documents, but the schema itself could be an IntellectualEntity in its own right. The two cases would have notably different BinaryEncodings. In the first case, the XML document would be the ContentComponent, software suitable for loading the document would be the SoftwareComponent, and the XML schema would be an InstantiationComponent. In the second case, the XML schema would be the ContentComponent and instantiation would require software capable of rendering the schema in a form suitable for humans to read.*
>
> *A piece of software could be an IntellectualEntity. In the BinaryEncoding of that object, the bitstream that constitutes the software would be a ContentComponent; however, the same software would be a SoftwareComponent in the BinaryEncoding of a data file that was created using the software or otherwise required the software for Manifestation.*

The ambiguities entailed by this situation are resolved by the association class, ComponentDescription. A ComponentDescription associates one DigitalComponent with one BinaryEncoding and specifies its role in that encoding. A ComponentDescription specifies the role that a DigitalComponent plays in a given BinaryEncoding. ComponentDescription is an abstract class that has three specializations corresponding to the three specializations of DigitalComponent: ContentComponentDescription, SoftwareComponentDescription, and InstantiationComponentDescription.

The second type of part of BinaryEncoding is Manifestation. While a BinaryEncoding represents an IntellectualEntity, a Manifestation enables it to be instantiated. A Manifestation is a concrete, physical implementation of an IntellectualEntity. Manifestation is an abstract class that is instantiated in two specializations: RuntimeVersion and Rendering. A RuntimeVersion is produced by loading the DigitalComponents in a BinaryEncoding into a computer for processing and/or output. A MachineReadableIntellectualEntity is fully instantiated in a RuntimeVersion. A Rendering is an output of an IntellectualEntity in a form accessible to humans; for example, on a display device, audio speaker, or in hard copy. A HumanReadableIntellectualEntity must first be loaded in a RuntimeVersion and then output in an appropriate Rendering. Every BinaryEncoding must have at least one RuntimeVersion, but a MachineReadableIntellectualEntity is not intended for human comprehension; therefore, the BinaryEncoding of a MachineReadableIntellectualEntity will have zero Renderings.[20]

A Manifestation is related to only one BinaryEncoding, a logical consequence of the fact that a BinaryEncoding corresponds to only one IntellectualEntity. A BinaryEncoding may have more than one Manifestations.

> *For example, if the IntellectualEntity is a database, the database could be instantiated in different views which select different data elements from the database and/or present them in different forms.*

A Manifestation is related to a BinaryEncoding by composition, indicated by the black diamond at the end of the lines connecting the two classes. If the BinaryEncoding is deleted, it cannot be instantiated.

A ManifestationDescription describes how DigitalComponents are processed to instantiate the IntellectualEntity represented by a BinaryEncoding. It should match ContentComponents and InstantiationComponents with the SoftwareComponents used to process them and, where needed,

[20] It may be possible to display the content of a MachineReadableIntellectualEntity in a human-readable form, but that would not be the form in which the MachineReadableIntellectualEntity could achieve its intended purpose; therefore, it would fail to satisfy the definition of Rendering.

specify relationships between SoftwareComponents. ManifestationDescription is an abstract class whose specializations, RuntimeVersionDescription and RenderingVersionDescription, correspond to the two specializations of Manifestation.

4.3. Collections

An IntellectualEntity may comprise one or more distinct objects. There are two different ways this can happen. The first is an Item, a specialization of IntellectualEntity that does not contain any other IntellectualEntity as such. An Item may be a composite object, as in the case of a textual document that contains a photograph, but the distinct things contained within the Item are described and managed only as parts of the Item, not as separate IntellectualEntities. The second case is another specialization of IntellectualEntity, a Collection; that is, an IntellectualEntity that contains one or more other IntellectualEntities which retain their independent existence and Features. The two specializations and their relationships are depicted in Figure 4, Collections. The relationship of a Collection to its members is shown by an association with IntellectualEntity, rather than Item because a Collection may have other Collections, as well as Items, as members, supporting hierarchical collections.

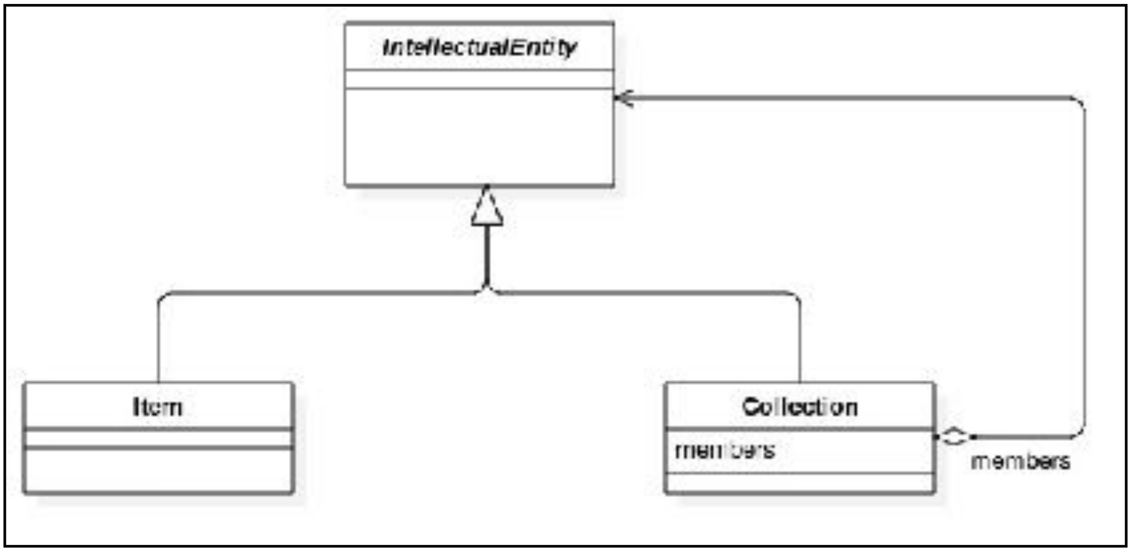


Figure 4, Collections

> *For example, an employee's personnel file may be aggregated with the personnel files of other employees, forming a record series, and this series could be aggregated into a higher level collection that contains other series related to human resource management, such as recruitment files, health and safety programs, training and development, union contracts, etc.*

An Item is an IntellectualEntity that does not contain any other IntellectualEntity. Empirically, an Item could be a composite object, for example a textual document that contained other types of information objects such as photographs or charts; however, in a Preservation Environment, they would be parts of the Item and not distinct IntellectualEntities. (A photograph or other distinct part of an Item could be separately declared as an Item in its own right, but it would then be managed as a different IntellectualEntity, independently of any Item in which a copy of the object is a part.)

The distinction between Item and Collection is independent of the specialization of Machine- and HumanReadableIntellectualEntities and of more precise taxonomies discussed in section 4.1 above.

> *Both Item and Collection could be members of the same specialization of IntellectualEntity. For example, a relational database, a type of MachineReadableIntellectualEntity, contains tables and each table contains tuples. Tuples are Items while the database as a whole and each table is a Collection.*
>
> *Similarly, a professional journal is a Collection that contains volumes and thereunder issues as members, while issues contain articles. All are comprised within the HumanReadableIntellectualEntity subclass of text.*

An IntellectualEntity might not be a member of any Collection. However, an IntellectualEntity could belong to one or more different (i.e., not parts of the same hierarchy) Collections. Membership of an IntellectualEntity in different Collections is possible where the Collections are defined by different criteria.[21]

> *For example, an email can be a member of the Collection of all messages in an employee's email account, and it could simultaneously be a member of a Collection defined on the basis of the business matter addressed in the message. The second Collection could include both messages that do not include the employee as sender or addressee as well as records that are not emails.*

As shown in Figure 4, every Collection has an attribute, members, that identifies the other IntellectualEntities that belong in the Collection. In addition, Collection should have attributes that define what subtype of collection it is:

- Bag: an unordered group whose members are not necessarily unique;
- Sequence: an ordered group whose members are not necessarily unique;
- Set: an unordered group each of whose members is unique; or
- Ordered Set: an ordered group each of whose members is unique.[22]

The attributes of a Collection should also specify the criteria for membership and whether membership is closed or open to the addition of new members.

Figure 4 presents a pattern that needs to be fleshed out in individual cases. The articulation could include defining additional Features of a particular Collection or defining specializations of Item and Collection.

4.4. Preservation Targets

An IntellectualEntity qualified as a PreservationTarget is to be protected from any change that would alter its ability to convey the information it was intended to communicate or, more practically, is to be maintained in a manner that serves the purpose for which it is preserved. The PaaST Requirements provide a verifiable method for determining whether a Preservation Target is preserved as intended through the specification of PermanentFeatures. A PermanentFeature describes an attribute or operation of a PreservationTarget that must remain invariant for preservation to be deemed

[21] The possibility of an Item being a member in several collections strictly does not apply to hierarchically nested Collections because an Item can only belong to the lowest level of the hierarchy. Members of higher level Collections are lower level Collections.

[22] UML Superstructure Specification, v2.4, Table 7.1, p. 131.

successful. Figure 5, PreservationTarget, shows how the articulation of PreservationTargets and PermanentFeatures supports verifiable preservation. The association of PermanentFeatures with a PreservationTarget is one of composition. A PermanentFeature describes an attribute or operation of a PreservationTarget that should not change; therefore, if the PreservationTarget disappeared, so would its PermanentFeatures.

PreservationTarget has the same basic structure as its generalization, IntellectualEntity. A PreservationTarget is represented by one or more BinaryEncodings. The differences between PreservationTarget and IntellectualEntity derive from the PermanentFeatures of the PreservationTarget.

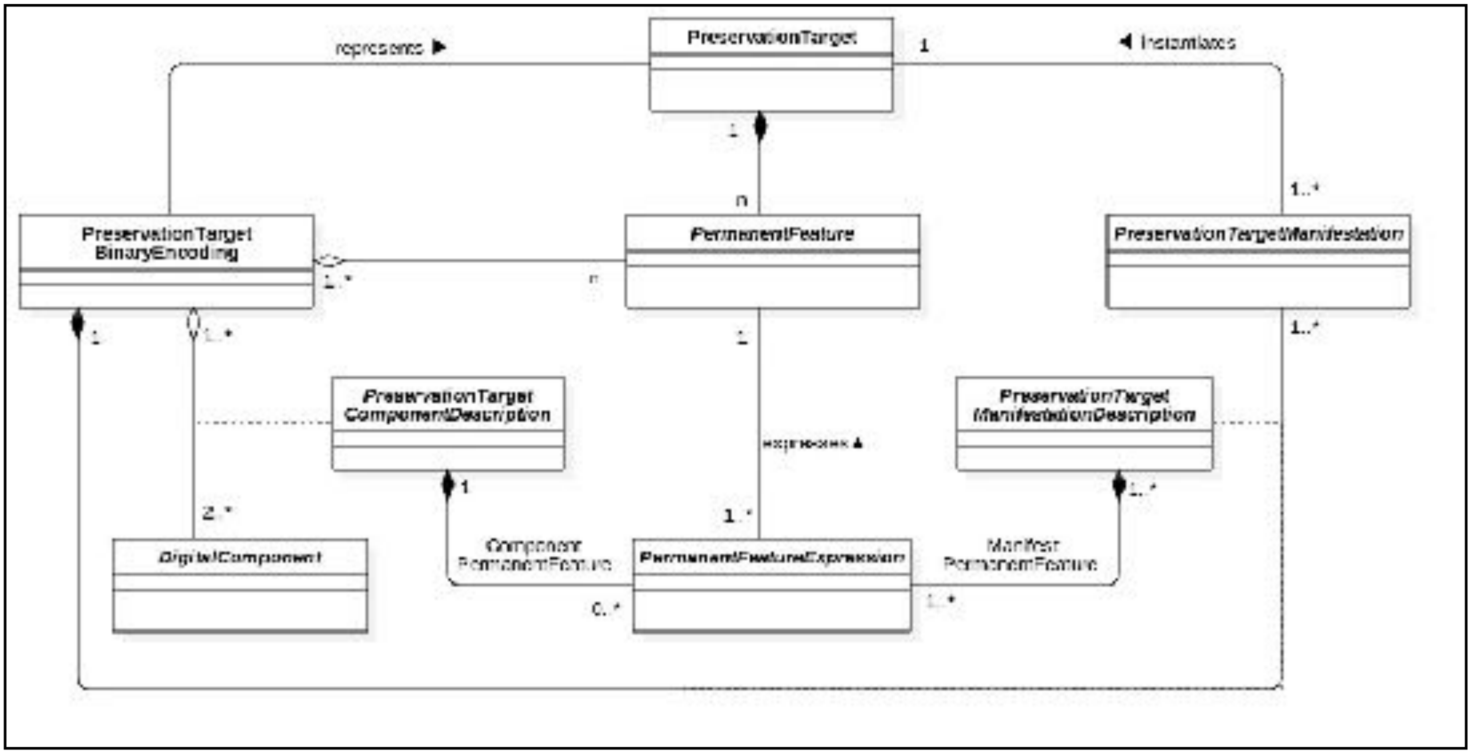

represents ▶
PreservationTarget
1
◀ instantiates
1
n
1..*
PreservationTarget BinaryEncoding
PermanentFeature
PreservationTargetManifestation
1..*
n
1
1..*
1
1..*
PreservationTarget ComponentDescription
expresses ▲
PreservationTarget ManifestationDescription
1
1..*
2..*
1..*
DigitalComponent
Component PermanentFeature
PermanentFeatureExpression
Manifest PermanentFeature
0..*
1..*

Figure 5, Preservation Target

A BinaryEncoding of a PreservationTarget must include every PermanentFeature of the target, as indicated by the same parameter, n, for the multiplicity of PermanentFeatures in an instance of BinaryEncoding, as in the association between PermanentFeature and PreservationTarget. Given that a PreservationTarget must have at least one and may have several BinaryEncodings, each PermanentFeature is part of one or more BinaryEncodings. The association is thus an aggregation, rather than composition.

A PermanentFeature must have at least one expression in a BinaryEncoding. A PermanentFeatureExpression describes how a PermanentFeature is exhibited in a Manifestation or a DigitalComponent of a PreservationTarget. The description must enable verification of the PermanentFeatureExpression. A PermanentFeature may be expressed in a DigitalComponent, a Manifestation or both, as indicated by the roles, ComponentPermanentFeature and ManifestPermanentFeature that associate a PermanentFeatureExpression with either a PreservationTargetComponentDescription or a PreservationTargetManifestationDescription in Figure 5. A PreservationTargetComponentDescription is a specialization of ComponentDescription that describes how a DigitalComponent is part of a BinaryEncoding of a PreservationTarget and, if the component expresses a PermanentFeature, specifies how it is expressed in the component. A PreservationTargetManifestationDescription is a specialization of ManifestationDescription that describes how a PreservationTarget is instantiated in a RuntimeVersion or a Rendering. A PreservationTargetManifestationDescription must identify all the ManifestPermanentFeatures of the PreservationTarget that are exhibited in the RuntimeVersion or Rendering.

> *For example, the identity of the sender of an email message will be expressed in the ContentComponent of the message. It will also be observable in any Rendering of the message. In the ContentComponent, the sender's identity is included in the email header. In the Rendering, the sender's identity is displayed as part of the message. The specifics of the display depend on the client software used; that is, on the SoftwareComponent, not the ContentComponent. Moreover, normally the Rendering will display only part of the header.*

Like every IntellectualEntity, a PreservationTarget contains two or more DigitalComponents and a DigitalComponent may be part of many PreservationTargets. A DigitalComponent is associated with a particular BinaryEncoding of a PreservationTarget by means of the association class, PreservationTargetComponentDescription, which is a specialization of ComponentDescription. The difference between PreservationTargetComponentDescription and its generalization is that a PreservationTargetComponentDescription contains zero or more PermanentFeatureExpressions. Zero is possible because a given PermanentFeature may not be expressed in a particular DigitalComponent.

> *SoftwareComponents that are used to instantiate many PreservationTargets will not express PermanentFeatures that are specific to any of them. Even in the case of PermanentOperations, it may not be possible to verify the Feature in the software that enables the operation. For example, while it might be well documented that a given software product is capable of displaying and rotating a 3-D model through a 360° solid angle, verification that this operation can be performed on a particular model may require user inspection of a Rendering.*

A PermanentFeature that is expressed in a DigitalComponent is not necessarily expressed in every component in a BinaryEncoding.

> *For example, few PermanentFeatures that uniquely identify a PreservationTarget will be expressed in a SoftwareComponent consisting of commercial software or in an InstantiationComponent that is part of a large number of BinaryEncodings.*

Some PermanentFeatures may not be expressed in a verifiable way in any DigitalComponent. In any such case, the PermanentFeature must be expressed in a Manifestation.

Like every IntellectualEntity a PreservationTarget is instantiated by one or more Manifestations. A PreservationTargetManifestation is a specialization of Manifestation that is distinguished from its generalization by the requirement that a PreservationTargetManifestation exhibit PermanentFeatures of the target. As in the general case, a PreservationTargetManifestation is related to BinaryEncoding by composition: if the BinaryEncoding did not exist, it could not have any Manifestation. Similarly, a PreservationTargetManifestation is related to only one BinaryEncoding. Many, if not all, of the PermanentFeatures of a PreservationTarget may be expressed in Manifestation. A PermanentFeature expressed in a PreservationTargetManifestation has the role of ManifestPermanentFeature. A ManifestPermanentFeature is associated with a PreservationTargetManifestation by composition because, if the PreservationTargetManifestation disappeared, it could not express any PermanentFeature. A ManifestPermanentFeature can be part of several PreservationTargetManifestations.

For example, in the case of a database, a common form of Manifestation is a periodic report. All reports generated from the same report specifications have common elements, such as the selection of data elements, format, labels, etc., that would be PermanentFeatures of the report; however, the reports for different periods are different PreservationTargetManifestations because the data elements for different periods are different.

A PreservationTargetManifestation will include several ManifestPermanentFeatures because inevitably every Manifestation would include several PermanentFeatures of the PreservationTarget. Although not shown in Figure 5, ManifestPermanentFeature has two specializations that describe, respectively, how a PermanentFeature is expressed in a RunTimeVersion or a Rendering.

It must be possible to verify that a PermanentFeature is correctly expressed in each locus where it is expressed. The method of identifying and verifying a PermanentFeature depends on where it is expressed, as well as on the characteristics of the Feature itself. Features expressed in DigitalComponents or RuntimeVersions may be testable using automated tools, while Features expressed in Renderings often will require human observation.

If it has been established that the BinaryEncoding that contains the DigitalComponents can be appropriately instantiated, it may be possible to verify the preservation of a PermanentFeature in a BinaryEncoding solely on the basis of testing for it in DigitalComponents in the case of a MachineReadablePreservationTarget. This could also be possible in the case of a HumanReadablePreservationTarget if there is assurance that a PermanentFeature that is appropriately expressed in a DigitalComponent will be reliably and appropriately expressed in a Rendering of the PreservationTarget generated from the BinaryEncoding that contains the DigitalComponent.

A PreservationTarget must have at least three types of PermanentFeatures in order to make it possible to verify that a PreservationTarget has been preserved successfully. Each of the three types may include several PermanentFeatures. The three types are:

- Uniqueness Features: PermanentFeatures that differentiate a PreservationTarget from any other. Many different PreservationTargets can have identical PermanentFeatures, but the set of Permanent Features of a PreservationTarget must uniquely distinguish it from any other.

  For example, all books should have title and author as PermanentFeatures and many books could have the same title or the same author. But if there is more than one book that has the same title and author, then each book should have at least one additional PermanentFeature, such as number or date of the edition, that uniquely distinguishes it from the others.

- Manifestation Features: the characteristics a PreservationTarget should have when instantiated. This requirement commonly will entail several PermanentFeatures, some or all of which may also be PermanentFeatures that uniquely distinguish the PreservationTarget.

  Manifestation Features should include required operations, such as specifications of the types of interactions that should be supported and required results of such operations.

  It is possible that a Manifestation Feature is only expressed in a Manifestation of the PreservationTarget, and not in any of its DigitalComponents, but some Manifestation Features could also be expressed in DigitalComponents.

  An example of a ManifestationFeature that is expressed both in a Rendering and in one or more DigitalComponents is the type font of textual characters. When the text is manifested, the characters should appear in the correct font. The font could be specified in formatting codes included in the

ContentComponent that contains the text. Alternatively, the font could be expressed in an InstantiationComponent, such as a style sheet, that is applied to the ContentComponent at the time of Manifestation.

An example of an Manifestation Feature that is expressed only in a Rendering is the ability to rotate a 3-D model through a 360° solid angle. While the ContentComponent and SoftwareComponent of the model must support this functionality, it is expressed only in Rendering.

Factors such as obsolescence may make it impossible to instantiate a PreservationTarget or to instantiate it with all PermanentFeatures appropriately expressed, but the specification of the PermanentFeatures required for instantiation is logically prior to and independent of the empirical possibility of instantiating a Manifestation. Nevertheless, any failure to express a PermanentFeature correctly should be documented in TargetProblem data related to the PreservationTarget and in ProblemHistoryData related to the PreservationAction in which the failure was discovered. Deciding whether inability to instantiate a PreservationTarget or inability to instantiate one or more PermanentFeatures appropriately is a critical failure of preservation is a matter of policy for the Preservation Director to decide.

- Integrity Features: the elements that make up and are necessary to preserve a PreservationTarget and their interrelationships. Integrity Features identify the essential parts or elements that constitute a PreservationTarget and articulate relationships among the parts that must be preserved.

For example, books are commonly divided into chapters and may be further divided into sections and subsections. Each of these is a constituent part that must be preserved; moreover, the sequence and hierarchical relationships of the parts must also be preserved and implemented in any Rendering.

Similarly, a natural language document might contain heterogeneous types of content, such as photographs, figures or tables. Each of these elements must be preserved. The correct placement of each element in the document and its relationship to specific elements of the text, such as captions and footnotes, are also integrity Features.

Preserving the integrity of a PreservationTarget entails maintaining the integrity of a BinaryEncoding used to instantiate the PreservationTarget and of the objects that constitute the BinaryEncoding. Every DigitalComponent in such a BinaryEncoding must be maintained and be unaltered. The number, order and values of the bits in a DigitalComponent should not change. The relationships among the components, described in PreservationTargetComponentDescriptions, must be maintained. Likewise, every MachineReadablePreservationTarget must have at least one implementable RuntimeVersion and every HumanReadablePreservationTarget must have at least one implementable RuntimeVersion and at least on Rendering. Wherever successful preservation requires multiple Manifestations, such as multiple views on a database, each one must be maintained and implementable.

Once it has been established that a BinaryEncoding satisfies the PermanentFeatures for uniqueness and instantiation of a PreservationTarget and as long as it remains in a steady state, demonstrating the integrity of the BinaryEncoding is a sufficient basis for asserting that the integrity of the PreservationTarget is preserved.

Given that the specific attributes of PermanentFeatures that support integrity, such as the number and identities of DigitalComponents, pertain only to an individual BinaryEncoding, in the case of PreservationChange, the issue of integrity shifts to the PreservationTarget and is resolved by ensuring that all other PermanentFeatures are not altered or lost in the new BinaryEncoding.

As indicated by the preceding discussion, the categories of uniqueness, manifestation and integrity are not mutually exclusive as applied to PermanentFeatures.

A Preservation Director may define generic PermanentFeatures that should be present in all instances of a class.

> *For example, two PermanentFeatures of all correspondence would be the author and addressee.*

Nonetheless, a PermanentFeature that an instance inherits from its class is specific to that instance because it is defined not only by its presence, but also by its value in the instance. Thus, in Figure 5, an instance of PermanentFeature is shown as belonging to only one PreservationTarget. Each PreservationTarget will have many PermanentFeatures, indicated by "n," where n is a number greater than 1, at the PermanentFeature end of the association.[23]

For various reasons, such as overcoming obsolescence, taking advantage of improvements in technology, or responding to the changing needs or capabilities of a Designated Community a Preservation Director might decide to authorize changes in PermanentFeatures. That should be accomplished by creating a new PreservationTarget that is related to an earlier one, but has a different set of PermanentFeatures. Doing otherwise could lead to uncertainty and even to false assumptions that there had been not significant changes over time.

There could be other reasons to create several PreservationTargets that have identical, or largely the same content and form.

> *For example, any document is an IntellectualEntity and, to be preserved, would have PermanentFeatures such as author, title, genre, organization, content, etc, which should be expressed in every instance of the document. However, the same document could be a record in several different contexts; that is, it could be several different records. All of the records would share the PermanentFeatures of the document, but each would have additional PermanentFeatures that are related to its use as a record, such as its provenance, archival bond and records management annotations and that distinguish it from the other records. Each record would be part of a Collection comprising all other records sharing the archival bond. Independently of that aggregation, the relationships among all of the records based on the document, and the document in itself, apart from its use as records could be expressed in a PreservationNetwork focused on the document.*

4.5. PreservationCollections

In many cases, a number of PreservationTargets should be preserved together as a group. This can be accomplished by extending the pattern of Item and Collection by the specializations, IntellectualEntity to PreservationTargets. A PreservationItem is an Item designated for preservation, and a PreservationCollection is a specialization of Collection whose members are PreservationTargets. The relationships between PreservationItem and PreservationCollection parallel those of Item and Collection. A PreservationCollection may contain one or more other PreservationTargets, which can be either Items or Collections.

A PreservationCollection has PermanentFeatures that are proper to it and different than those of individual members. PermanentFeatures of every PreservationCollection should include:

- members: identifying the Items or lower level Collections that belong in the PreservationCollection;

[23] Multiplicity is indicated as "n" rather than the indefinite "*" to support the constraint that a BinaryEncoding must include every PermanentFeature of a PreservationTarget.

- uniqueness: specifying whether each member of the Collection is unique;
- order: specifying the attribute or attributes of the members that determine the order of the Collection, if the Collection is PreservationCollection.

  Examples of the ordering of members of a PreservationCollection include the sequence of volumes and issues of a journal and the arrangement of records in accordance with a files classification systems.

Other Features of a PreservationCollection may be designated as PermanentFeatures depending on circumstances. Attributes that should be considered as candidate PermanentFeatures include:

- numberOfMembers: the count of Items or Collections in a PreservationCollection.

  In the case of hierarchical Collections, the numberOfMembers of a Collection would be the count of Collections at the next lowest level, and the complete hierarchy should include as a PermanentFeature the number of levels and the numberOfMembers in each level.

  A PreservationCollection may grow over time by addition of members. In such cases, the baselineValue of numberOfMembers should be increased after each successful Submission that contains members of the PreservationCollection; furthermore, another PermanentFeature in such cases would be the formula that the number of members in the Collection at any given time should equal the total number ingested in all Submissions up to that time. Consideration should be given to dividing such PreservationCollections into sub-collections. The lower level Collections could correspond to either the members in each successful Submission or the cumulative members up to that point. With either option, the baselineValue of numberOfMembers of each sub-collection would not change.
- membershipCriteria: the criteria that determine membership in the collection. This attribute can be important in uniquely identifying a PreservationCollection. This attribute could supplement numberOfMembers in determining if a PreservationCollection is complete, providing a logical rather than merely empirical basis for the determination. This Feature would also enable determining if a PreservationCollection includes members that do not belong in it.

  For example, if a PreservationCollection consists of a time series of observational data accreted on a monthly basis, any month for which there was no data would be a gap in the series. Similarly, if the membership criteria for a data collection specified a geographic area that contains subdivisions, any subdivision not represented in the collection would signal a gap.

The requirements for PermanentFeatures that establish uniqueness, enable instantiation and enforce composition integrity apply to PreservationCollections as well as PreservationItems. For a PreservationCollection, all of its members and their interrelationships must be maintained, whether the members are PreservationItems or lower level Collections. Composition integrity is transitive over all levels of Collections.

4.6. Heuristic Information

Understanding and using preserved information often requires additional information, such as explanations of its structure and semantics, that helps people to understand it. In PaaST, such information is classified as HeuristicInformation. HeuristicInformation is information that helps people to discover, understand, evaluate or use one or more Preservation Targets. It is distinguished from other IntellectualEntities not by content or form, but by its relationship to PreservationTargets. HeuristicInformation is especially important in preservation because the context in which a PreservationTarget is retrieved and used differs — increasingly with the passage of time — from that in which it was created and used originally. This difference is accentuated in digital preservation by changes in the technology needed to retrieve, use and even to encode PreservationTargets. The evolution of natural language and changes in other means of communication can also impact the interpretation of a PreservationTarget.

HeuristicInformation could elucidate PreservationTargets in several different ways. Catalogs of PreservationTargets relevant to a given area of interest, archival descriptions and tools such as topic maps facilitate discovery. Historical accounts describing the circumstances in which PreservationTargets were created, glossaries of terms used in a PreservationTargets, or descriptions of the data models underlying complex databases, etc. can contribute to understanding. User guides or manuals and explanations of the accuracy, precision, coverage or other qualities of a PreservationTarget help determine if they are appropriate for intended uses.

Figure 6, Heuristic Information, illustrates the relationship between HeuristicInformation and PreservationTargets. While a PreservationTarget can be human or machine readable, HeuristicInformation must be human readable.[24]

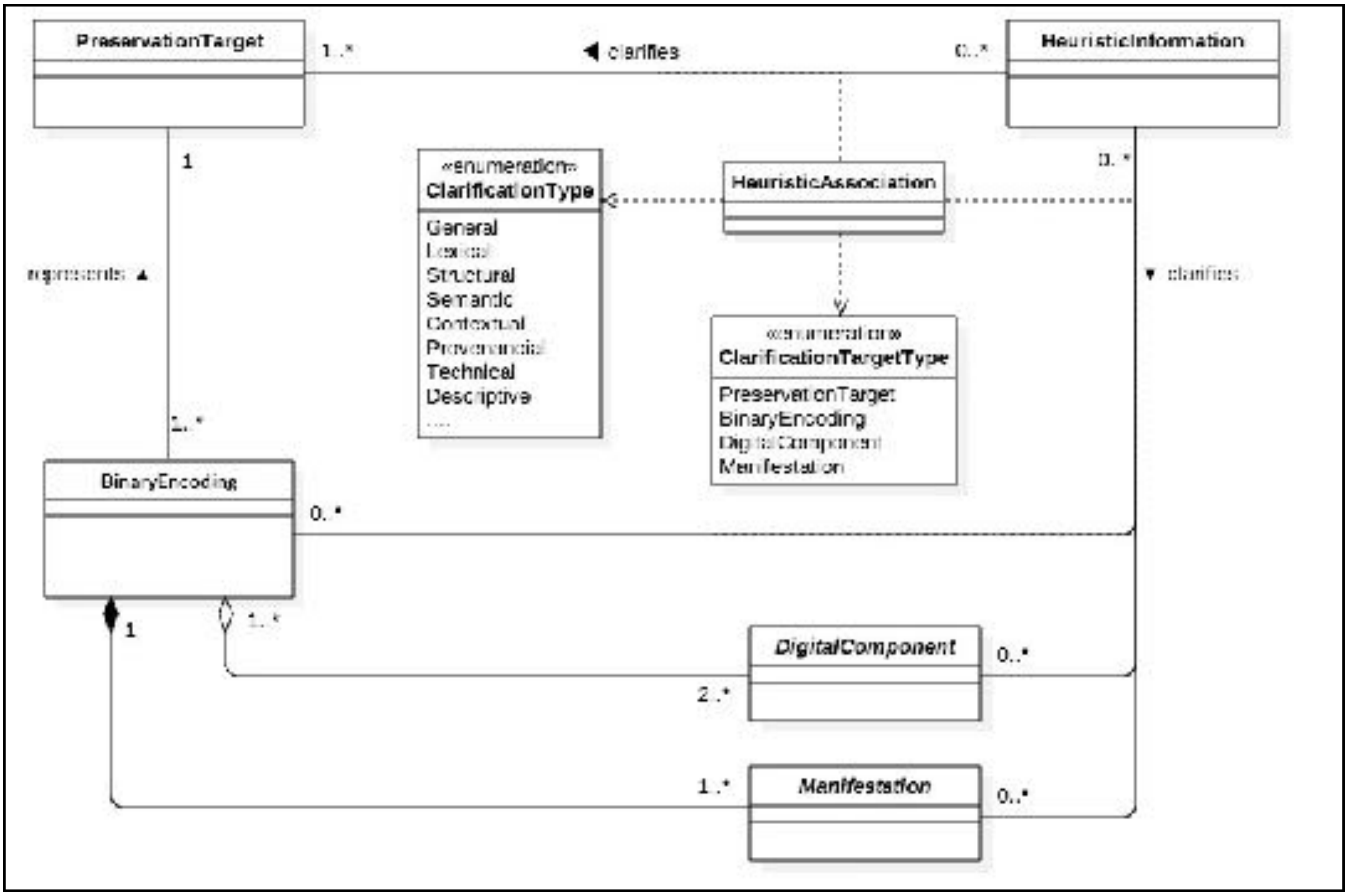


Figure 6, Heuristic Information

[24] HeuisticInformation is a subset of OAIS Representation Information; however, the latter encompasses information that is not human readable. Machine readable Representation Information could be included in a Preservation Environment as PreservationManagementData or, in some cases, as InstantiationComponents.

There could be a PreservationTarget that has no associated HeuristicInformation, but a PreservationTarget could be clarified by many HeuristicInformation objects, and a HeuristicInformation object could clarify many PreservationTargets. Hence, Figure 6 shows that zero or more instances of HeuristicInformation may be related to one or more PreservationTargets.

> *A situation where there are zero HeuristicInformation objects might occur when a library acquires the papers of an author as a donation from the author's estate. The collection might include digital files whose contents and context, initially at least, are unknown and, therefore, cannot be associated specifically with any other documents about the author or the writings.*

> *An example of a PreservationTarget clarified by several instances of HeuristicInformation would be a digital photograph listed in a catalog and also linked to a biography of the photographer and to a text describing the subject of the picture. Its technical properties might be described in another text. Each of these would be a HeuristicInformation object about the photograph.*

> *Conversely, the biography of the photographer could be linked to every photograph by that individual. It would be a HeuristicInformation object associated with several PreservationTargets.*

> *Similarly, a description of a series of records sheds light on every file and record in the series.*

The link between a PreservationTarget and an instance of HeuristicInformation is specified by the association class, HeuristicAssociation. HeuristicAssociation can be characterized according to the categories listed in the enumeration, ClarificationType. As indicated by the ellipsis, ClarificationType could be expanded with additional entries to suit the needs of different situations. The categories in ClarificationType are not mutually exclusive. An instance of HeuristicInformation, might provide both lexical and semantic clarification of a PreservationTarget.

The content of HeuristicInformation may relate specifically to particular elements of a PreservationTarget, such as a BinaryEncoding, DigitalComponent or Manifestation.

> *For example, if the PreservationTarget was a database, HeuristicInformation about it might describe the data model, which would be an InstantiationComponent of the database.*

Any HeuristicInformation that elucidates a BinaryEncoding, DigitalComponent or Manifestation in some sense also clarifies the related PreservationTarget. The link between HeuristicInformation and an element of a PreservationTarget is also specified by an instance of the association class, HeuristicAssociation. An instance of HeuristicInformation could related to more than one of these elements; however, each instance of an HeuristicAssociation associates one instance of HeuristicInformation with only one other object. Thus, an instance of HeuristicInformation that provides technical information about a DigitalComponent is associated with the DigitalComponent by an instance of HeuristicAssociation and it is associated with the PreservationTarget of which the DigitalComponent is part by another instance of HeuristicAssociation. If the same HeuristicInformation elucidated both a BinaryEncoding and a Manifestation, there would be a separate instance of HeuristicAssociation linking the HeuristicInformation to each.

PreservationTarget and HeuristicInformation are not mutually exclusive classes. An IntellectualEntity could be both a PreservationTarget and a HeuristicInformation object. Any HeuristicInformation that originated with the same Producer as a PreservationTarget it clarifies should be designated as a PreservationTarget. But a HeuristicInformation object is not necessarily a PreservationTarget.

> *An example of an IntellectualEntity that would be both a PreservationTarget and a HeuristicInformation object is a guide or manual created to help end users to understand and use a computer application, such as a database. If the database is a PreservationTarget, it would be wise also to preserve the manual created by the database Producer.*

> *An example of a HeuristicInformation object that is not a PreservationTarget might be a history that provides insight into the creation or impact of a PreservationTarget. Because of intellectual property rights, the Preservation Director might not have the option of making the book a PreservationTarget. Furthermore, the Preservation Director might decide to discard the book if a subsequent, improved history were produced.*

HeuristicInformation can be submitted to a Preservation Environment, acquired from a third party independently of any Submission, or generated within the Preservation Environment. Submitted HeuristicInformation might be designated as a PreservationTarget and preserved as part of a PreservationCollection that includes the other PreservationTargets to which it is related.

> *For example, the PreservationCollection for a scientific research project might include the data set generated in the project and also the protocol under which the research was conducted and reports of the results of the research. All three would be Preservation Targets that belong to the same PreservationCollection. The protocol and reports would qualify as HeuristicInformation in relation to the data set.*

Whether HeuristicInformation generated in the Preservation Environment is designated as a PreservationTarget depends on the preservation policies of the Preservation Director. Generated HeuristicInformation might be designated as a permanent record of the Preservation Director. In that case, following the principle of provenance, it would be a member of a PreservationCollection consisting of other records of the Preservation Director, apart from the PreservationTargets it describes.

5. What's Involved: Preservation Management Information

Trusting digital preservation depends on having adequate and appropriate knowledge of what is preserved; what actions are taken with respect to what is preserved; and the outcomes of such actions. This knowledge is especially important in situations, such as preservation in the Cloud, where responsibility for PreservationTargets and PreservationActions is entrusted to others and where the Preservation Director does not have control over what technologies are used or how they are used.

PaaST Requirements provide for the acquisition, production, maintenance, use and evaluation of the knowledge required for preservation in the form of PreservationManagementInformation. Its organization is illustrated in Figure 7, PreservationManagementInformation. An abstract class, PreservationManagementInformation has two specializations: PreservationManagementDocument and PreservationManagementData. A PreservationManagementDocument is a human readable object whose contents are about preservation. PreservationManagementData is structured data about PreservationTargets, PreservationActions, PreservationRules and Actors.

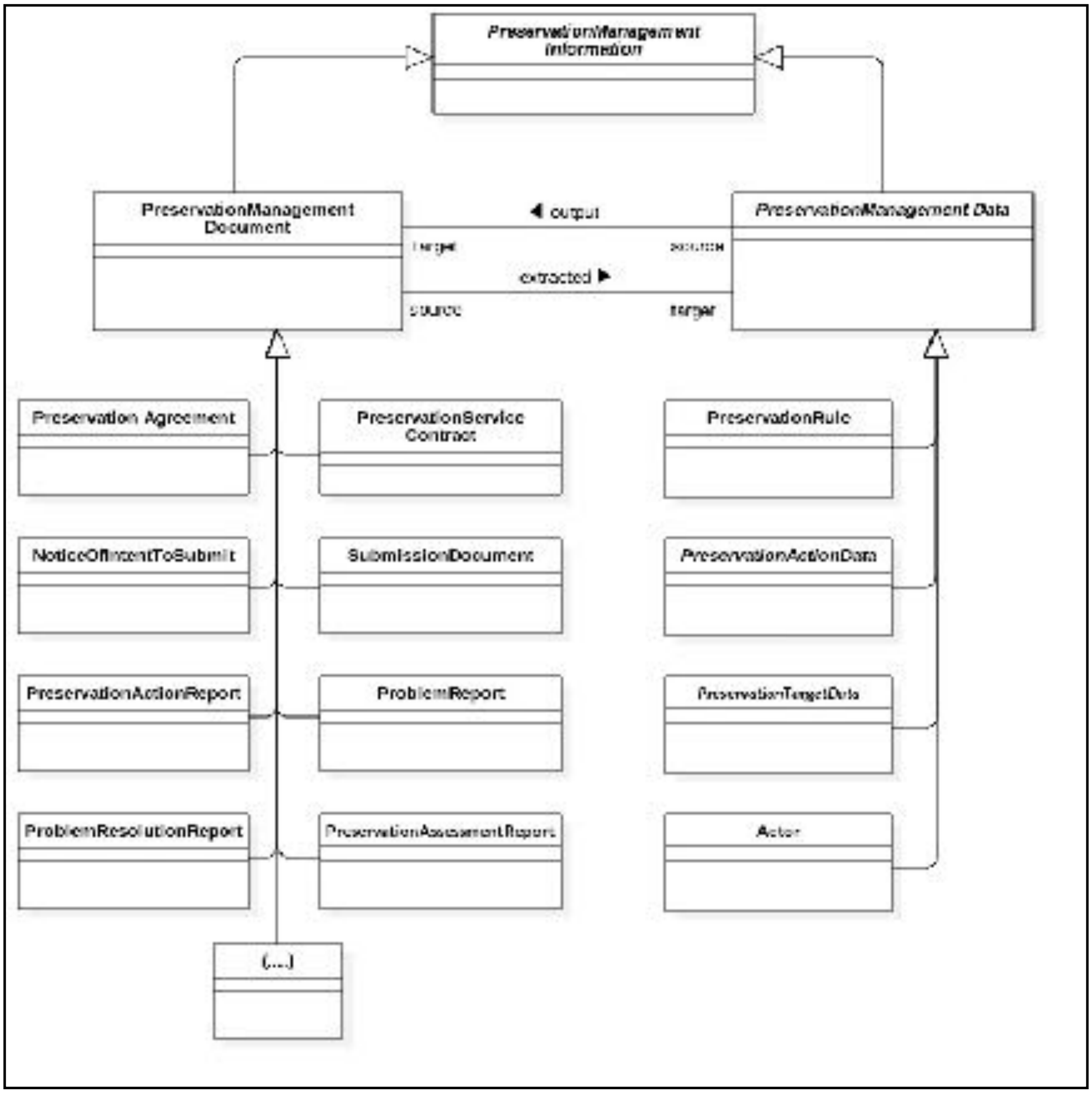


Figure 7, Preservation Management Information

Any Preservation Environment will include management information that does not fall within the scope of PreservationManagementInformation, but is necessary for operations, maintenance, security, etc. Such information is outside of the scope of PaaST. Moreover, PaaST does not specify information requirements related to IntellectualEntities either beyond the basic distinction between machine readable and human readable specializations or apart from their roles as PreservationTargets or HeuristicInformation. Those specifications may be customized in individual implementations.

PreservationManagementData may be extracted from PreservationManagementDocuments and output for inclusion in PreservationManagementDocuments. These possibilities appear in Figure 7 in the two associations between PreservationManagementDocument and PreservationManagementData. Thus, PreservationManagementDocument and PreservationManagementData differ in form according to their intended users, humans and computers, but share content. A PreservationAction generates at least one piece of PreservationManagementInformation concerning what was done and its result. In addition, if any problem occurs in carrying out a PreservationAction, data about the problem and what is done to address it can and should be captured. If a PreservationAction results in a new or altered PreservationTarget, data describing that object would also be produced.

PreservationManagementData has one subclass that qualifies as a core class, PreservationRule. A PreservationRule is a business rule that governs execution of one or more Preservation Actions. It is a specialization of PreservationManagementData, rather than the more general PreservationManagementInformation, because a PreservationRule must be expressed as structured data in order to be executable. Additional details about PreservationRules are provided in section 5.4, Preservation Rules, below.

5.1. Preservation Management Documents

An abstract class, PreservationManagementDocument has several specializations, shown in Figure 7, that should be common in Preservation Environments:

- Preservation Agreement: a document representing an agreement between an Initial Source and a Preservation Director to preserve certain Preservation Targets, often specifying terms and conditions applicable to their preservation;
- Preservation Service Contract: an agreement between a Preservation Service Provider and a Preservation Director for the provision of one or more Preservation Services;
- NoticeOfIntentToSubmit: a document that a Submitter uses to inform a Preservation Service Provider of a proposal to transfer PreservationTargets for preservation;
- SubmissionDocument: a document that contains information about a Submission and usually accompanies it;
- PreservationActionReport: a document summarizing or detailing the performance of one or more Preservation Actions;
- ProblemReport: a document detailing any problem found in an object or during execution of a Preservation Action; and
- ProblemResolutionReport: a document describing actions taken in response to a ProblemReport, including whether the response resolved the problem.

- PreservationAssessmentReport: a document describing the results of a PreservationAssessment.

As suggested by the class whose label is "….," there can be other types of PreservationManagementDocuments, including ones designed for specific Preservation Environments.

5.2. Preservation Management Data

PreservationManagementData, depicted in Figure 8, is an abstract class with five specializations:

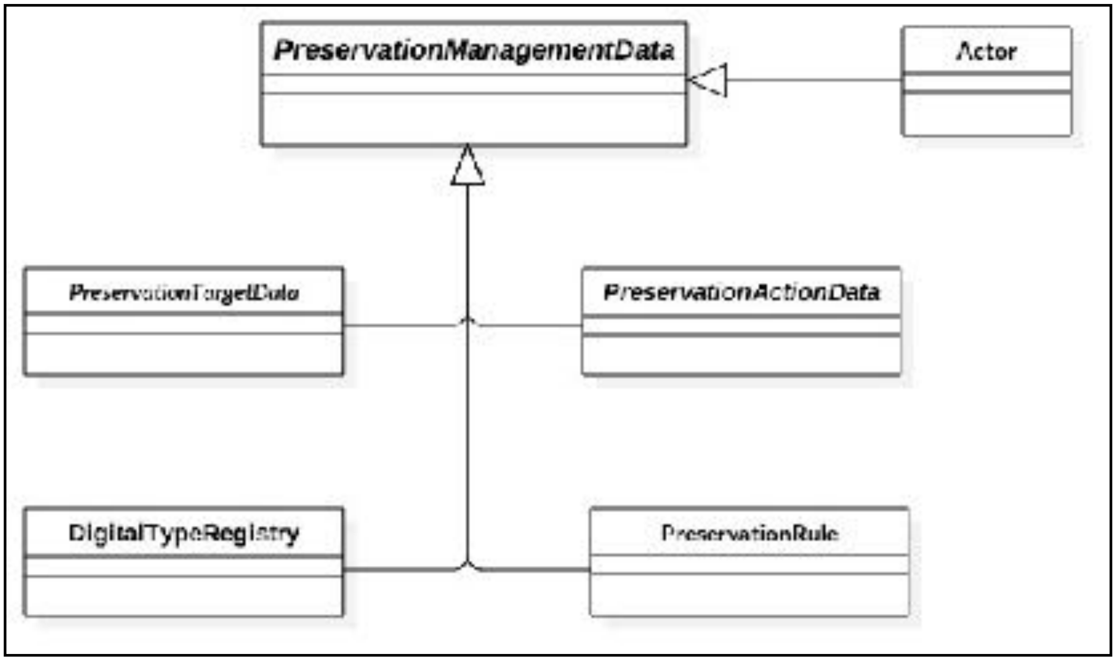


Figure 8, Preservation Management Data

- PreservationTargetData comprise data that describe PreservationTargets or any of their specializations, elements or their subclasses.
- DigitalTypeRegistry is a database describing types of DigitalComponents.
- PreservationActionData describe PreservationActions, the Parties and PreservationTargets involved and the results or outcomes of the action, including any problems encountered and their resolution. PreservationActionData should be created every time a PreservationAction is executed, regardless of whether it completes successfully.
- PreservationRule is a business rule that governs execution of one or more Preservation Actions and impacts one or more PreservationTarget objects.
- Actor describes individuals who may or do participate in carrying out PreservationActions.

These five classes are interrelated. Whenever a PreservationAction produces a change of state in any PreservationTarget, the results should be recorded as PreservationTargetData. Similarly, if a PreservationAction creates a new BinaryEncoding or DigitalComponent or removes a PreservationTarget from the Preservation Environment, the result should be captured. A PreservationRule concerns a PreservationAction, a PreservationTarget, often both. A

DigitalTypeRegistry describes technical characteristics of types of DigitalComponents used to instantiate PreservationTargets. By definition, an Actor participates in PreservationAction.

PreservationManagementData specializations inherit Features from the generalization. All specializations of PreservationManagementData should indicate the source and date of data entry. The source of the data should be identified by both the type of source and the specific source. Types of sources include Actor, PreservationActionData, PreservationManagementDocument, PreservationManagementData, HeuristicInformation and PreservationRule. A data entry could have more than one type of source.

> *For example, data might be generated automatically in the execution of a PreservationAction and an Actor might input additional data during its execution.*

Requirements related to PreservationManagementData include the possibility of imposing controls to ensure data quality.

5.2.1. PreservationTargetData

PreservationTargetData is structured data about PreservationTargets. It has two immediate specializations, TargetDescription and TargetState, as shown in Figure 9, Preservation Target Data. TargetDescription encompasses data about PreservationTargets, their specializations and elements. Six of its seven specializations, described in detail in chapter 4, are necessary to define what a PreservationTarget is and how it is composed and expressed: PreservationTarget, BinaryEncoding, ComponentDescription, ManifestationDescription, PermanentFeature, and PermanentFeatureExpression. The seventh specialization of TargetDescription is HeuristicInformation, described in section 4.6 above. PreservationTargetData describe basic characteristics of objects, once created, should be invariant.

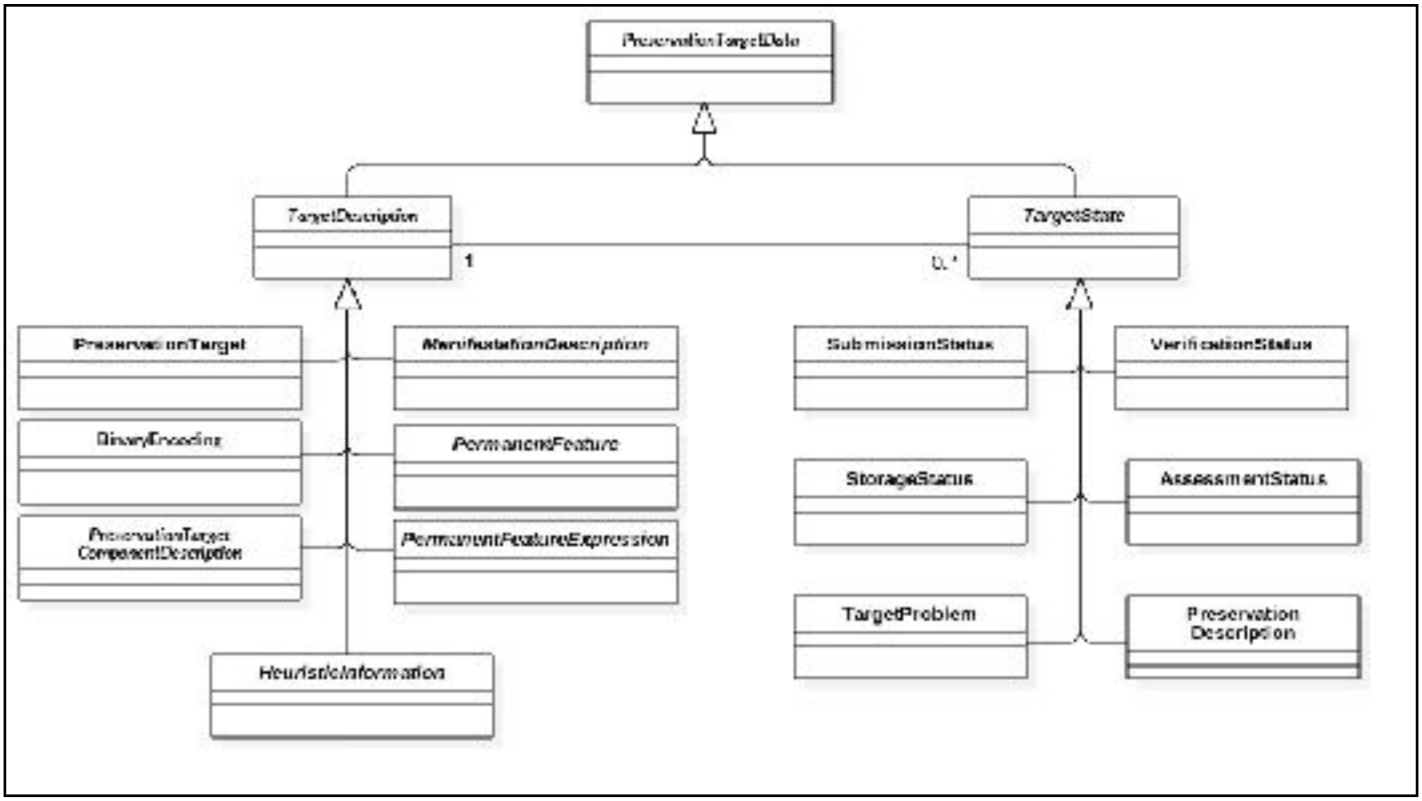


Figure 9, Preservation Target Data

In contrast, TargetState describes the state of a PreservationTarget at a given time. While TargetDescription consists of data about PreservationTargets that are basically static, TargetState comprises data that are subject to change; that is, the set of values of the attributes of the PreservationTarget, its elements and its associations with other objects that may change over time. State changes include creation, deletion and modification of the value of one or more Features of an object. An instance of TargetState should be recorded when a PreservationTarget is created or introduced in a Preservation Environment and subsequently whenever there is a change of state of a PreservationTarget, BinaryEncoding, DigitalComponent, ManifestationDescription or ComponentDescription; when existing state data are verified; when a PreservationAssessment is performed; or when a problem related to the PreservationTarget occurs. TargetState, thus, captures the impact of a PreservationAction on any PreservationTarget. TargetState data should be cumulative, constituting a complete audit trail on the preservation of a PreservationTarget.

TargetState specializations include:

- SubmissionStatus about the authorization, Submitter, transmission, contents or conditions of a Submission;
- StorageStatus describing the storage of DigitalComponent;
- Assessment Status, describing the results of Preservation Assessment,
- VerificationStatus, describing the results of Verification,
- TargetProblem, characterizing any problem related to a PreservationTarget, as well as any attempt to resolve the problem and its outcome, and
- PreservationDescription, a generic category for other data that describe the preservation of a PreservationTarget.

There are no subclasses of TargetState corresponding to PreservationChange or Access. The result of a PreservationChange would be the production of a new BinaryEncoding. Data about it should be stored in an instance of BinaryEncoding and related ComponentDescriptions and Manifestations. The impacts of a PreservationChange would be determined by Preservation Assessment and, therefore, recorded in AssessmentStatus. Whether a new BinaryEncoding successfully preserves the PreservationTarget would be documented in VerificationStatus. The data describing the handling of any problems that occur in PreservationActions should be stored as PreservationActionData, but the impact of ProblemHandling on any object should be stored in the classes that would be used if no problem occurred. When Access results in a change of state, data describing the results should be stored in appropriate subclasses of TargetState. Access that involves only output of information or production of Manifestations for external clients would not change the state of any PreservationTarget.

Before any description of a PreservationTarget is created, there can be no data about its state; however, a PreservationTarget may be described before it is submitted to a Preservation Environment, indicating, for example, that its Submission is authorized. TargetState data might not be created until the PreservationTarget is actually submitted. Thus, the multiplicity of TargetState in relation to TargetDescription may be zero. Every instance of TargetState must relate to one and only one TargetDescription because state data describes the state of an object. Over time, many instances of state data related to a given object may accumulate; however, there may be no instance of TargetState related to an ObjectDescription because an object may be described, at least partially, before it is submitted to or created in a Preservation Environment.

Additional subclasses of TargetState may be created to serve the needs of different situations.

5.2.2. DigitalTypeRegistry

A DigitalTypeRegistry is a database describing types of DigitalComponents. A DigitalTypeRegistry contains many DigitalTypeRegistryEntries, one describing each type of DigitalComponent within the scope of the registry. A registry entry should describe the technical characteristics of the type of DigitalComponent that are important for preservation and instantiation. As shown in Figure 10, Digital Type Registry, a Preservation Environment may include specializations of DigitalTypeRegistry. Figure 10 includes three specializations corresponding to the different specializations of DigitalComponent. A DataFormatRegistry describes the encoding of data files and would most often describe ContentComponents. A SoftwareRegistry describes types of SoftwareComponents, and an InstantiationObjectsRegistry describes types of objects that may function as InstantiationComponents.

PaaST Requirements provide for the inclusion of existing registries by importing them. For purposes of illustration, Figure 10 includes three examples of imported registries. They are the MIMETypeRegistry, an instance of DigitalTypeRegistry;[25] PRONOMRegistry, an instance of DataFormatRegistry, which is a registry of data formats maintained by the National Archives of the U.K;[26] and the National Software Reference Library Reference Data Set (NSRL RDS), an instance of SoftwareRegistry, which is maintained by the U.S. National Institute of Standards and Technology.[27]

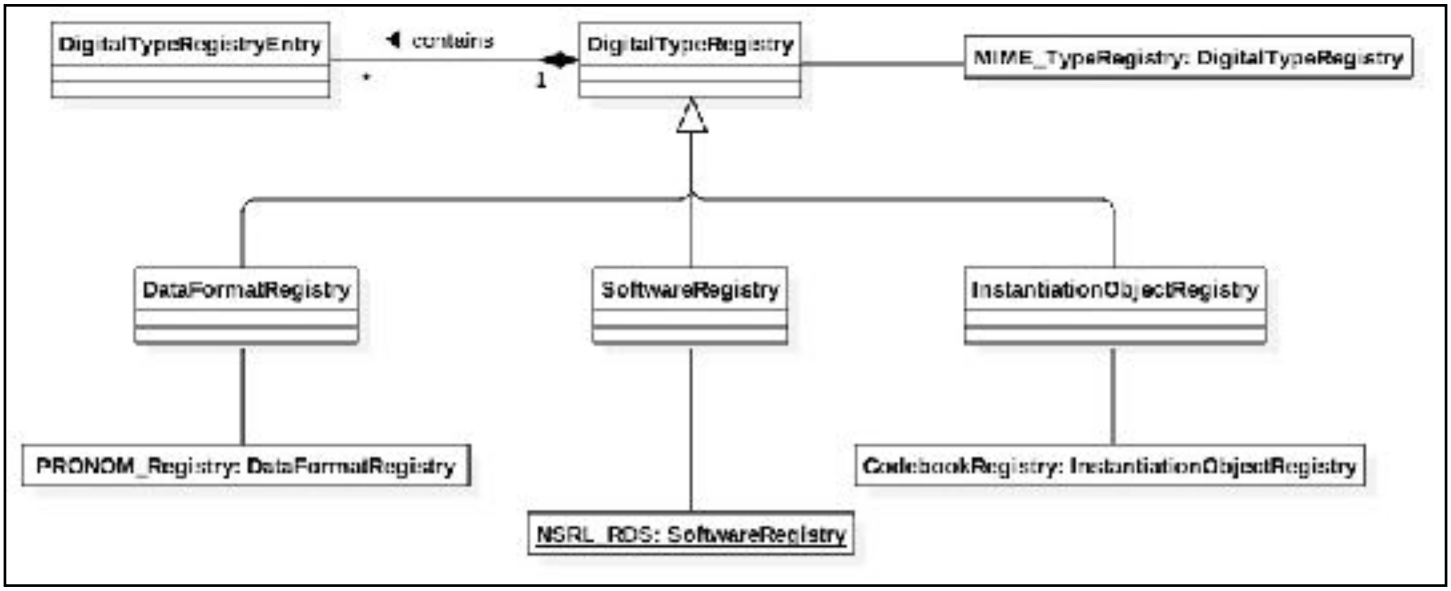

DigitalTypeRegistryEntry
contains
*
1
DigitalTypeRegistry
MIME_TypeRegistry: DigitalTypeRegistry
DataFormatRegistry
SoftwareRegistry
InstantiationObjectRegistry
PRONOM_Registry: DataFormatRegistry
CodebookRegistry: InstantiationObjectRegistry
NSRL_RDS: SoftwareRegistry

Figure 10, Digital Type Registry

[25] Internet Assigned Names Authority. Media Types. 2017-07-06. https://www.iana.org/assignments/media-types/media-types.xhtml.

[26] The National Archives (U.K.). The Technical Registry, PRONOM. https://www.nationalarchives.gov.uk/PRONOM/Default.aspx.

[27] National Institute of Standards and Technology (U.S.). National Software Reference Library Reference Data Set. https://www.nist.gov/sites/default/files/data-formats-of-the-nsrl-reference-data-set-16.pdf.

The organization of DigitalTypeRegistries into specializations is customizable. A Preservation Director might decide to create registries that are tailored to the DigitalComponents being preserved in a given case, as illustrated by the inclusion in Figure 10 of a CodebookRegistry as a hypothetical instance of InstantiationObjectRegistry.

Every DigitalTypeRegistryEntry should include an attribute that indicates whether the type of DigitalComponent described by the entry is obsolete. PaaST requirements provide the capability for propagating this status to all the instances of DigitalComponents of the type described and to all BinaryEncodings that contain such components, indicating the need to implement PreservationChange to overcome the obsolescence.

5.2.3. PreservationActionData

PreservationActionData is an abstract class whose specializations correspond to the categories of Preservation Action Services in the requirements. As shown in Figure 11, Preservation Action Data, the specializations are SubmissionProcessingData, PreservationStorageData, PreservationChangeData, PreservationAssessmentData, VerificationData and AccessData. There is also a subclass, ProblemHistoryData, that identifies any problems encountered in Preservation Action and the responses to problems. ProblemHistoryData is included because the record of any of the other subclasses would be incomplete if details about problems and problem handling were not included. While there are no subclasses for Information Management or Preservation Management Capabilities, data about execution of these Capabilities are in fact captured in both PreservationActionData and TargetState data.

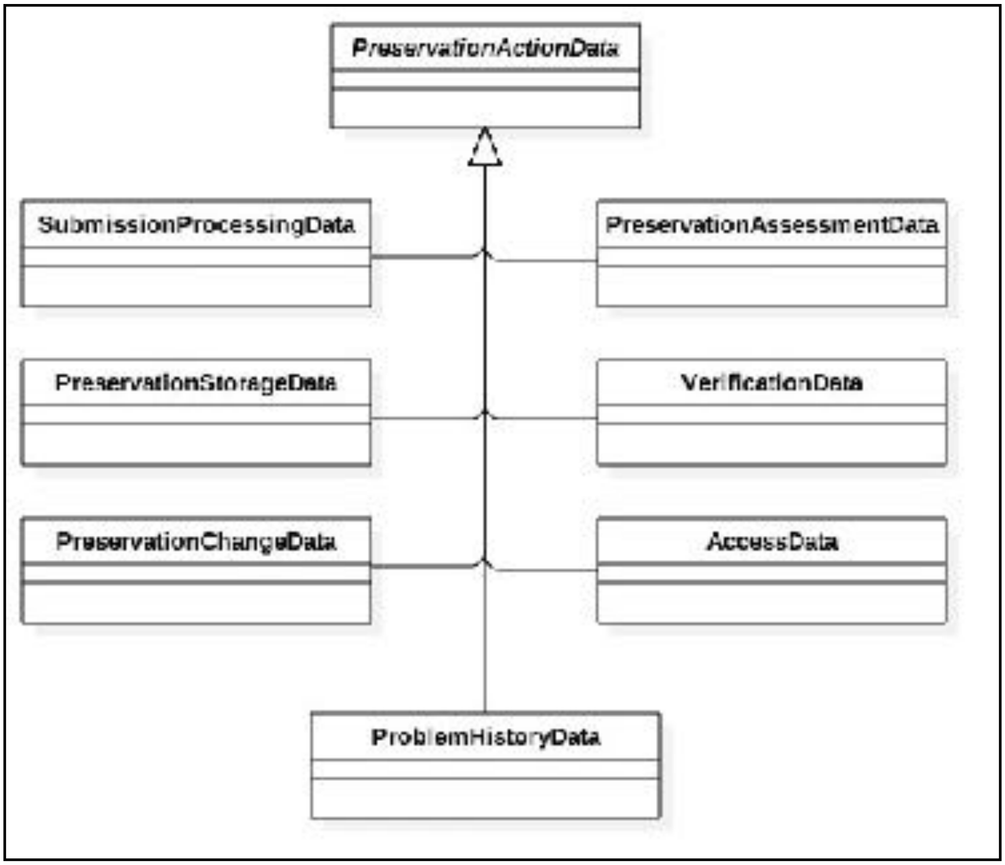


Figure 11, Preservation Action Data

Additional subclasses of PreservationActionData may be created to serve the needs of different situations.

5.2.4. PreservationRule

The third specialization of PreservationManagementData is PreservationRule. This class is discussed in detail in section 5.4, PreservationRules, below.

5.2.5. Actor

An Actor is a role played by an entity that uses any PaaST capabilities. Actor is described above in section 3.2. While user management is a generic function that is out of scope of the PaaST requirements, an adequate record of preservation must include data about the Actors involved in PreservationActions, including authorizations and restrictions on their participation, as well as their activity.

Data stored in an instance of Actor should include data uniquely identifying the Actor, data specifying the capabilities the Actor is authorized to use or prohibited from using, specifications of the modes in which an Actor may participate in an action, details on rights, conditions and limitations on information the Actor may input, output or modify

5.3. Management Sets

The PaaST requirements enable the definition and implementation of ManagementSets defined to satisfy particular management objectives. A ManagementSet is not necessarily a PreservationCollection. Rather it is a means of grouping objects for purposes such as execution of an action on all members of the set or application of a PreservationRule to all members. The members of a ManagementSet do not have to be instances of the same class or hierarchy of classes. ManagementSets that should be useful in most Preservation Environments include SubmissionSets, PreservationActionSets and PreservationNetworks. Optionally, a Preservation Director or Preservation Service Provider could create other ManagementSets to achieve particular management purposes.

For example, within a Preservation Environment, all email, regardless of provenance, might be identified as a set to provide a basis for a consistent approach to their preservation, such as preserving all email messages as separate documents in a format that conforms to the RFC 2822 standard for Internet email.[28]

Similarly, a ManagementSet might be defined as comprising all PreservationTargets related to a given topic in order to facilitate discovery and access. Such sets would often be unrelated to policies or actions taken for preservation itself because they would include technologically heterogenous materials.

5.3.1. Preservation Action Sets

A PreservationActionSet is a set of PreservationTargets established to facilitate executing a PreservationAction or Actions collectively on all members of the set. The criteria used to define PreservationActionSet depend on the particular management objective to be served. A PreservationActionSet may be defined solely on the basis of one or more common Features of PreservationTargets, without specifying the Preservation Action.

For example, a PreservationActionSet defined on the basis of documentary form could serve several purposes. It would enable the articulation of a template identifying PermanentFeatures common to all

[28] Internet Engineering Task Force. Internet Message Format. 2001. https://www.ietf.org/rfc/rfc2822.txt

instances of that form. It would also allow the formulation of PreservationRules applicable to all members of the set. It could also be used in executing a Verification process on all PreservationTargets having that form.

The definition of a PreservationActionSet might specify a PreservationAction as well as the objects to be processed in the action.

A PreservationActionSet could comprise PreservationTargets, Binary Encodings, or DigitalComponents or some combination of them.

For example, a PreservationActionSet comprising all DigitalComponents in a given format would facilitate mass migration if the format became obsolete, but migrating a DigitalComponent to a new format creates a new BinaryEncoding; therefore, the BinaryEncodings in which the DigitalComponents belong should be included in the set. Furthermore, every BinaryEncoding must be linked to an IntellectualEntity; therefore, the PreservationTargets should also be included.

Conversely, if a PreservationActionSet were defined on the basis of document form, the primary members of the set would be the PreservationTargets in that form, but an IntellectualEntity is represented by a BinaryEncoding; therefore, the BinaryEncodings would also be members as would the DigitalComponents contained in them.

A PreservationActionSet may have a reflexive relation to itself; that is, a PreservationActionSet could be a member of another PreservationActionSet or could contain one or more other such sets.

For example, the set of all XML documents could use the same SoftwareComponent to instantiate each of the documents. It could include as a subset all those XML documents that conform to a given schema.

PreservationActionSet may be defined orthogonally, so that objects can belong simultaneously to several sets that are independent of one another.

Examples of two orthogonal PreservationActionSets that could have members in common:

PreservationActionSets defined according to subject matter;

PreservationActionSets defined on the basis of technical characteristics, such as format.

A PreservationActionSet needs to be retained as a set only as long as the management objective that motivated creating the set is in force. Thus, it does not constitute a PreservationCollection. PreservationActionSets could be defined to meet the policies and needs of different situations.

5.3.1.1. Submission Sets

A SubmissionSet is a specialization of a PreservationActionSet where the PreservationAction is Submission. A SubmissionSet is a set of related IntellectualEntities, Binary Encodings, and DigitalComponents transferred to a Preservation Environment together. The IntellectualEntities in a SubmissionSet may include both PreservationTargets and HeuristicInformation. SubmissionSets are defined to support the orderly transfer and ingest of Preservation Targets into a Preservation Environment. In many cases, a SubmissionSet will be equivalent to a PreservationCollection; however, that is not necessarily the case. While it is advisable to retain PreservationManagementInformation about every Submission, a SubmissionSet that is not a PreservationCollection does not need to be maintained as a set after ingest is completed.

Conditions imposed on a Submission could specify the timing and means of transfer, the identity of authorized Submitters, what information should accompany the SubmissionSet, criteria for acceptance, and procedures for handling any problem discovered in Submission Processing. Such conditions could be set out in one or more PreservationRules.

5.3.2. Preservation Networks

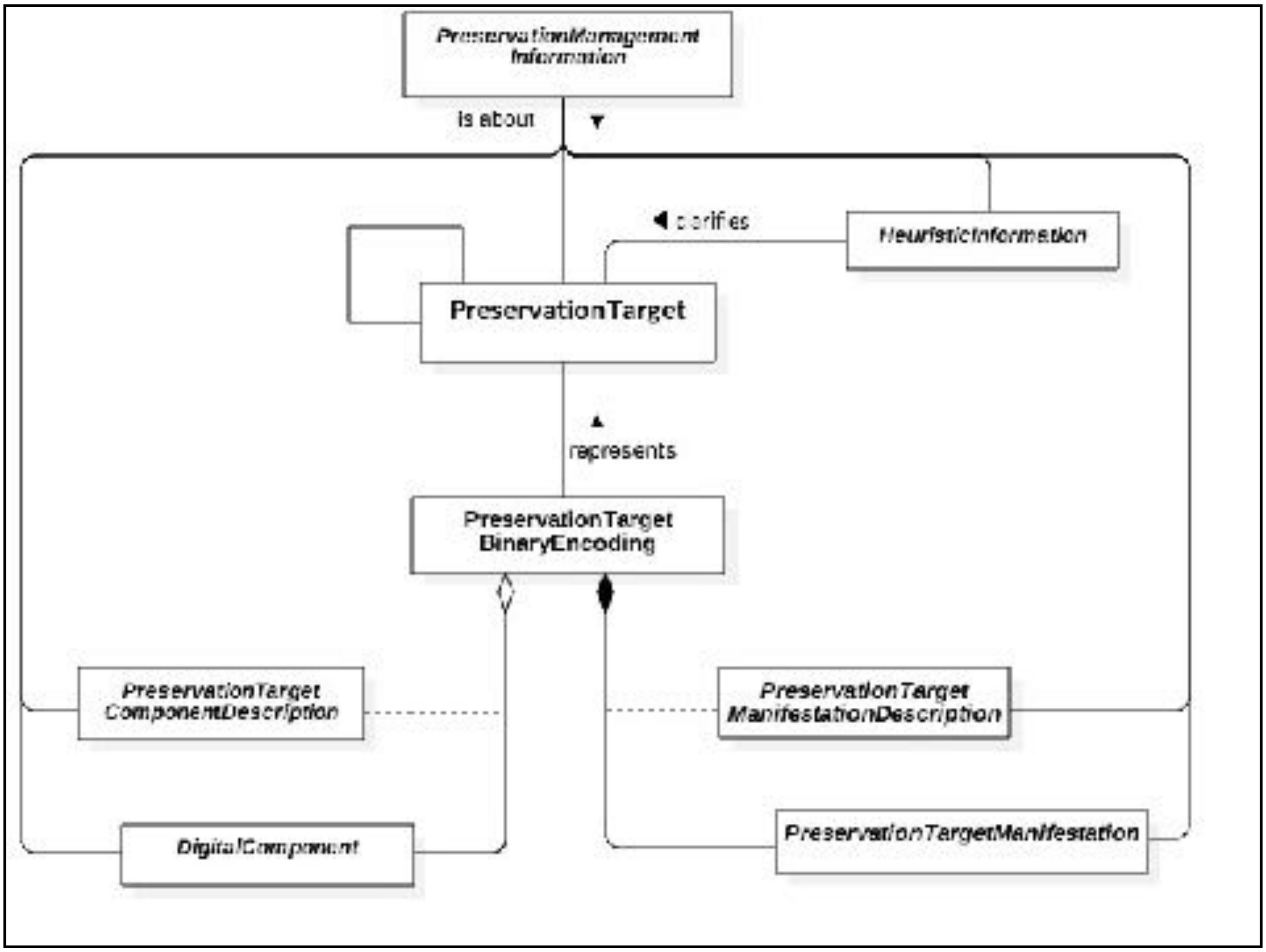

PreservationManagement Information
is about
clarifies
HeuristicInformation
PreservationTarget
represents
PreservationTarget BinaryEncoding
PreservationTarget ComponentDescription
PreservationTarget ManifestationDescription
DigitalComponent
PreservationTargetManifestation

Figure 12, Preservation Network

Any PreservationTarget has at least one BinaryEncoding and at least two DigitalComponents needed to instantiate it. There may be many more, including additional Binary Encodings and DigitalComponents, and possibly several HeuristicInformation objects that clarify it. Every DigitalComponent has an associated ComponentDescription. Over time, there will be a growing quantity of PreservationManagementInformation about all of these objects. All of these objects are significant, and many are necessary, for successful preservation of the PreservationTarget. The set of all such objects constitute a PreservationNetwork with the PreservationTarget as its focus.[29] A PreservationNetwork could also include any other PreservationTarget associated with the focal PreservationTarget. Associated PreservationTargets would include any PreservationCollection in which the focal PreservationTarget belongs or, if it is itself a PreservationCollection, the PreservationTargets that belong in it. There might also be ad hoc relationships among PreservationTargets.

[29] The concept of ‘PreservationNetwork’ expands on the OAIS concept of ‘Representation Network,’ but a Representation Network is, by definition, more limited: “The set of Representation Information that fully describes the meaning of a DigitalComponent.” OAIS p. 1-15.

For example, there could be related documents in two different archival fonds, or a database may contain data imported from another database that had a very different structure and purpose or was created by a different Producer.

As indicated by its red color in Figure 12, Preservation Network, a PreservationTarget is the focus of a PreservationNetwork. In many cases, the most important part of the network will comprise other PreservationTargets related to the focal one, indicated by a generic reflexive association in the figure. This association is left unspecified in order to able the Preservation Director to define networks appropriate to different situations and objectives. One obvious possibility is that the reflexive relationship enables allows associating a PreservationItem with a PreservationCollection and with other Items in the Collection. This association also makes it possible to link PreservationTargets that have a common origin, but not necessarily the same provenance.

For example, a PreservationNetwork could be defined to link records created by a person in different and largely independent capacities, such as records created by a person in the role of owner of a private business and records created by that person pursuing an avocation such as poet or amateur sportsman.

PreservationNetworks could also be defined to associate PreservationTargets related to the same topic or event, regardless of how or where they originated.

The defining characteristic of a PreservationNetwork, therefore, is the need or desire to preserve relationships among the objects that are nodes in the network, especially in cases where these relationships go beyond part/whole or member/collection associations.

Requirements numbered 9.6 provide for identifying the members of a PreservationNetwork. Doing so provides an empirical basis for a management overview of the preservation of the focal PreservationTarget and related objects. A PreservationNetwork also opens up opportunities for Actors to learn of items of interest that they might not otherwise discover.

5.4. Preservation Rules

Generally, the PaaST requirements are articulated in terms that support a variety of PreservationTargets, policies, practices and objectives. In any Preservation Environment, the applicable requirements need to be supplemented by rules that are appropriate for the PreservationTargets to be preserved; support the specific objectives of preservation; and respond to the conditions of that environment.

There are many ways that rules can be articulated and implemented in a Preservation Environment. When a rule applies to all instances of a class or subclass, it could be established as a constraint on the class or on one or more of its Features. Constraints may also be applied to actions. Constraints that apply categorically can be preconditions, postconditions, invariants or consistency constraints. Preconditions and postconditions apply before or after an action is taken, while invariant and consistency constraints must be true at all times.

An example of a precondition is that, in order to be accepted for preservation in a Preservation Environment, a PreservationTarget should be authorized for submission.

An example of a postcondition is that, when a new BinaryEncoding is created in a PreservationChange, the new encoding must support proper instantiation of the PreservationTarget.

An example of an invariant constraint is the requirement that every BinaryEncoding include at least one ContentComponent and at least one Software Component

An example a consistency constraint is the generic rule that every BinaryEncoding should express every PermanentFeature of the PreservationTarget it represents.

Some rules, however, are too complex to be implemented by any of these options. This can be the case with rules that apply to several different classes, rules that apply differently to different instances of a class, and rules that depend on the existence or details of one or more instances or Features. A broad category of complex rules comprises those that cannot be reliably or fully implemented by automated means, but require human participation.

For example, some PermanentFeatures may only be expressed in Renderings and require a human to examine the Renderings.

Such cases could be addressed by PreservationRules. A PreservationRule is an executable business rule that specifies when and under what conditions one or more requirements are implemented, as well as how to determine if the rule was followed successfully. A PreservationRule constrains implementation of a capability articulated in one or more PaaST requirements and governs one or more objects and/or actions.[30]

The PaaST Domain Model includes a rich conceptualization of PreservationRule that should support a wide range of situations. As shown in Figure 13, PreservationRule, a PreservationRule has four essential components. The RuleDomain specifies the context in which the rule is applied. Success Criterion defines the basis for evaluating success in implementing a rule. RuleTask is an action that carries out the rule, and RuleEvaluator is an action that applies a Success Criterion to determine if the result(s) of the RuleTask are acceptable. All of these classes are associated with PreservationRule by composition. They are parts of the definition of a rule, and they would become meaningless if the rule were rescinded.

A RuleDomain identifies the classes of objects and actions involved in application of a PreservationRule, other objects that function as controls in application of the rule, and the Actor(s) authorized to participate in rule execution. An object that is governed by a PreservationRule is a RuledObject. A RuledObject is a role played with respect to the PreservationRule by an instance of one of the classes defined previously, including PreservationTarget, PermanentFeature, BinaryEncoding, Manifestation, HeuristicInformation and their specializations and subparts. Analogously, a RuledAction is any PreservationAction governed by a PreservationRule. Conceivably, a PreservationRule might apply only to a RuledAction and not a RuleObject; however, an action that did not impact any object would be literally inconsequential. Hence, any PreservationRule that governs a RuledAction necessarily applies to at least one RuledObject. But a PreservationRule might apply only to a RuledObject and not to any action.

For example, a rule that every PreservationTarget must have PermanentFeatures that uniquely distinguish it from all other PreservationTargets should be in force at all times, even when no PreservationAction is being performed.

Hence the multiplicity of these elements in the RuleDomain is that there can be zero or more RuleActions and there must be at least one RuledObject. There can be more than one RuledObject. Thus, a PreservationAction that involves a PreservationCollection may impact PreservationItems in that Collection or peer or higher level Collections.

[30] Other business rules may be implemented in a Preservation Environment, such as rules governing access to information. If they do not impact at least one PreservationTarget, such rules do not qualify as PreservationRules and are not discussed in this document.

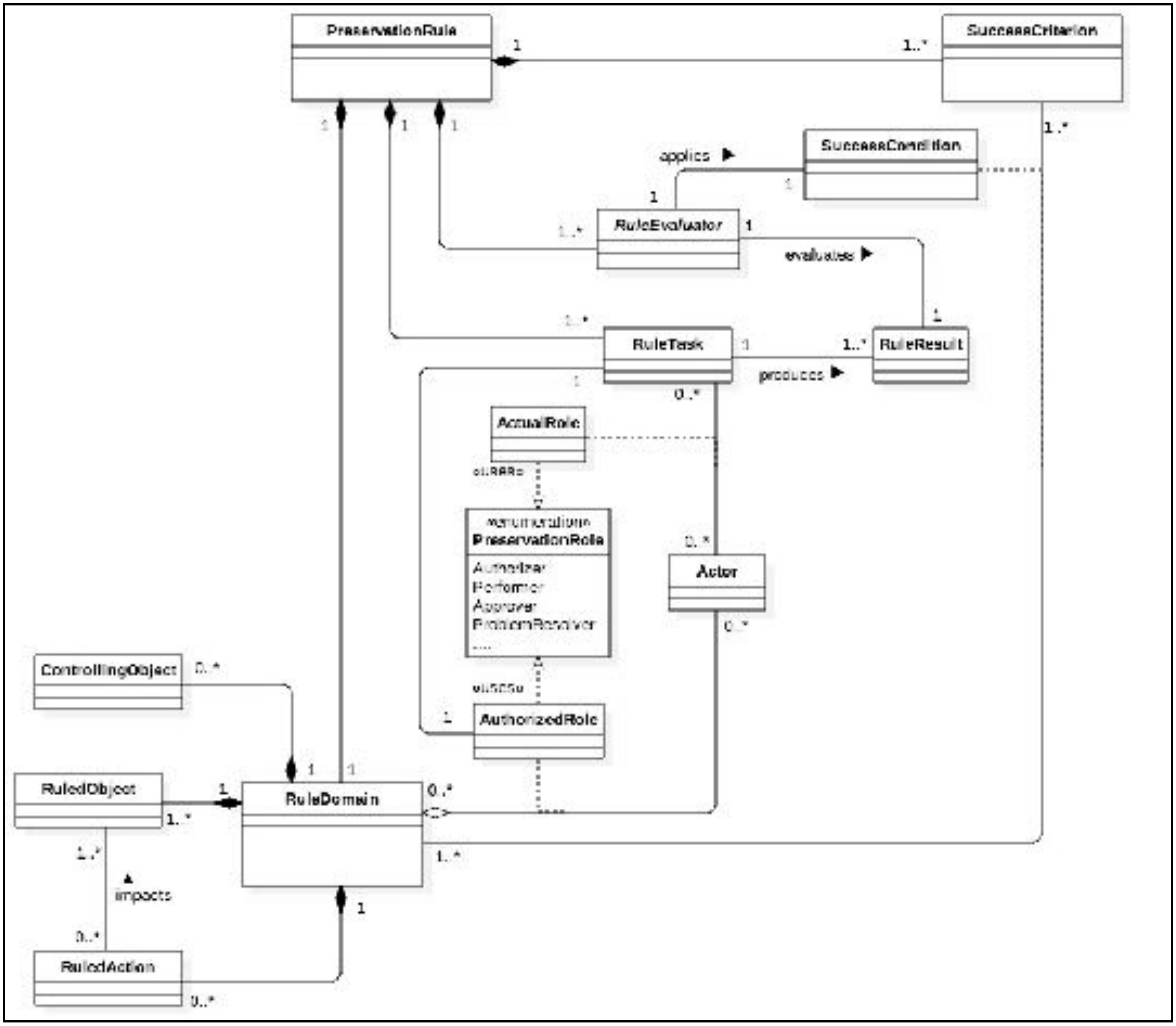


Figure 13, Preservation Rule

For example, in some email applications, each mail folder in an email account corresponds to a single file. A PreservationRule might dictate that each email message be stored as a separate object. Unbundling the folder files to separate individual messages would impact the BinaryEncodings of the folder, the individual messages, and also the Collection that comprises all folders in the account.

Conversely, implementing a PreservationRule on a PreservationItem may also impact a PreservationCollection in which the Item is a member.

For example, if Items that belong in a PreservationCollection are submitted in several installments over time, each Submission changes the membership of the Collection.

Every instance of a class may be a RuledObject under a given rule.

For example, a PreservationRule governing Submission may specify that an instance of a class of PreservationTargets should be submitted to the Preservation Environment when it is 10 years old.

> *Therefore, the RuledObjects that should be submitted in a given year are all PreservationTargets created 10 years previously.*

An object may be explicitly declared as a RuledObject in a RuleDomain. Additionally, an object can be qualified as a RuledObject as a result of its association with another RuledObject.

> *For example, a basic PreservationRule is that a PermanentFeature must be preserved in every Manifestation of a PreservationTarget. If a PreservationTarget is declared to be a RuledObject in a Rule Domain, every BinaryEncoding of the PreservationTarget and every DigitalComponent in the BinaryEncoding that express a PermanentFeature become a RuledObject by association with the PreservationTarget.*

The third element in RuleDomain is ControllingObject. Like RuledObject, a ControllingObject is a role of an instance of another class. In this role either the existence or one or more Features of the object function as controls in the application of the PreservationRule.

> *For example, if a Preservation Environment includes a significant set of PreservationTargets that are all members of a specialization of IntellectualEntity and the Preservation Director has defined PermanentFeatures that should be found in every instance of the specialization that is designated as a PreservationTarget, then each PermanentFeature of the specialization is a ControllingObject for every PreservationTarget that belongs to that class.*

An object may take on the role of ControllingObject in a RuleDomain because of an existing association with a RuledObject in that domain. A RuleDomain may contain one or more ControllingObjects.

> *For example, when a PreservationChange is made, the PermanentFeatures of the PreservationTarget must be expressed in the BinaryEncoding produced by the change process. Thus, each of the PermanentFeatures of the PreservationTarget is a ControllingObject in the application of this rule.*

A PreservationRule may not have any ControllingObject.

> *For example, a general PreservationRule is that a BinaryEncoding must represent one and only one IntellectualEntity. Operationally, that means that every BinaryEncoding must be associated with one and only one IntellectualEntity. Both the BinaryEncoding and the IntellectualEntity are RuledObjects. There is no other object referenced in applying the rule. Thus, there is no ControllingObject.*

While instances of PreservationActions and of the classes that can have the role of RuledObject or ControllingObject might be involved in many different PreservationRules, the roles of RuledAction, RuledObject and ControllingObject are defined with respect to a particular RuledObject or ControllingObject. Hence their association with RuleDomain is one of composition.

A RuleDomain may, but does not necessarily, include Actor. A rule whose execution is fully automated and does not involve any person or external system, by definition, does not involve any Actor. When a person or external system is allowed or required to be involved in the execution of a PreservationRule, the Actor’s involvement is specified through the association class, Authorized Role. Instances of this class should indicate if the Actor’s participation is required or optional as well as specifying the nature of the participation. Possible modes of participation are listed in the enumeration, PreservationRole, in Figure 13. Predefined roles in the enumeration include Authorizer, which identifies an Actor as one whose approval is needed for the rule to be applied in specified circumstances, Performer, identifying an Actor who should be actively involved in execution of the rule, Approver, an Actor designated to determine whether execution of the rule was satisfactory, and

ProblemResolver, an Actor to whom problems should be addressed. The ellipsis, "…," at the bottom of this list indicates that a Preservation Director may define other ParticipationRoles as needed. The AuthorizedRole should also identify the RuleTask for which the PreservationRole is authorized.

Some PreservationRules may relate primarily to Actors or modes of Actors' participation in PreservationActions, such as rules authorizing Submitters to transmit specified PreservationTargets. Even a rule that primarily concerns an Actor will also involve the PreservationActions and PreservationTargets to which the Actor is or is not authorized.

> *For example, a rule authorizing a Submitter necessarily entails a Submission action and must specify the PreservationTargets the Submitter is authorized to submit. Similarly, a rule authorizing an Actor should indicate what functionalities or objects the client is authorized to access in the Preservation Environment.*

There is not necessarily any Actor actively involved in the execution of a PreservationRule, but there may be more than one. Accordingly, a RuleDomain contains zero or more instances of Actor.

> *An example of a PreservationRule not involving an Actor is one whose execution is automatic. One such rule is that every BinaryEncoding must represent one and only one IntellectualEntity. A PaaST implementation might include a trigger that automatically checks whether every new BinaryEncoding satisfies this criterion.*
>
> *An example of a PreservationRule involving more than one Actor would be a rule governing Submission that both specified the authorized Submitter and required approval by the Preservation Director for acceptance.*

One or more Actors may be involved in the execution of a RuleTask. Given that an Actor may be involved in many tasks, the involvement of an Actor in a specific RuleTask is determined by an instance of the association class, ActualRole. ActualRole uses the same enumeration, PreservationRole, as AuthorizedRole. Carrying out any RuleTask that involves an Actor should include determining that the ActualRole is an AuthorizedRole of that Actor in that task.

Logically, every PreservationRule must have at least one action that implements it. Such an action is called a RuleTask. Specifically, a RuleTask is an action that produces one or more RuleResults, where a RuleResult is some output or outcome that is significant for determining if the rule was applied successfully. If we think of a rule as a statement of the form, "If X, then Y," there must be either at least one RuleTask that has the result, Y, or the RuleResults of several tasks must add up to Y. Additionally, a RuleTask may have one or more RuleResults that are related to or indicative of Y, but not equivalent to it. A PreservationRule may have many RuleTasks.

Every RuleTask should produce one or more RuleResults. RuleResults should be defined to be testable. Testing of a RuleResult is done by a RuleEvaluator, which is a method specifically designed to determine if a particular result is as it should be. Thus, there should be a one-to-one association between RuleResult and RuleEvaluator. However, this does not mean that there can only be one tool or procedure for evaluating a RuleResult. An instance of RuleEvaluator is defined with respect to a RuleResult and, as with all classes in the domain model, is independent of the way it is implemented. More precisely, a RuleEvaluator compares a RuleResult to a SuccessCondition applicable to that result.

A SuccessCondition is determined by applying a SuccessCriterion of the PreservationRule to the RuleDomain and, more specifically to the instance of a RuledObject or RuledAction involved in an actual implementation of the rule. A SuccessCriterion is a standard, principle or reference point used to determine whether an execution of a PreservationRule satisfies the rule. When a SuccessCriterion is applied to a particular case, data in the RuleDomain determine the value of the corresponding SuccessCondition. For a RuledObject, that value is either (1) the value that a Feature of a

RuledObject should have in order to satisfy the applicable SuccessCriterion; (2) a condition related to a RuledObject that should be true after completion of a RuleTask; or (3) a combination of the two. For a RuledAction, the value to be tested by the RuleEvaluator is either a postcondition that should be true when the operation completes successfully, or the value of a parameter returned by the operation.

> *A common example of a SuccessCondition that is a value of a Feature of a RuledObject is the benchmark value of a PermanentFeature.*
>
> *An example of a SuccessCondition that is a condition related to a RuledObject is, in Submission processing, the authorization of a Submitter to submit a particular PreservationTarget.*
>
> *Examples of SuccessConditions that are combinations are cases where the value of a Feature of a RuledObject must be consistent with Features of ControllingObjects. For example, if a RuledObject is a PreservationItem that is a member of a PreservationCollection, the value of any Feature of the PreservationItem that relates to a criterion for membership in the PreservationCollection must be consistent with the criterion. Thus, if the PreservationCollection has a limited date range, the date of an Item in the Collection must fall within that range.*

Several PreservationRules can have SuccessCriteria that are conceptually identical, but an instance of SuccessCriterion can belong to one and only one PreservationRule by virtue of its association with the RuleDomain of that rule. An instance of SuccessCriterion must be tailored to the RuledObjects and/or RuledActions in the domain and must support application of any ControllingObjects in that domain.

> *For example, a SuccessCriterion related to PreservationTargets is that every BinaryEncoding express every PermanentFeature of the related target. This criterion would be applicable in rules governing both Submission and PreservationChange. The SuccessCondition, however, would be different in the two cases. In Submission Processing, frequently the benchmarkValue of a PermanentFeature will need to be determined by examining the BinaryEncoding. The SuccessConditions in that case would be the discovery of the benchmarkValue in one or more elements of the BinaryEncoding, the determination that the benchmarkValue is identical in all elements where the PermanentFeature is expressed, and its recording in the TargetDescription. In PreservationChange, the SuccessConditions would be confirmation that the PermanentFeatureExpression(s) in the new BinaryEncoding matched the benchmarkValue.*

To illustrate the articulation of a PreservationRule, consider a situation where a BinaryEncoding includes a SoftwareComponent that has become obsolete and that, in the Preservation Environment, there is a PreservationRule that stipulates that in such situations a new BinaryEncoding should be created using a replacement SoftwareComponent that is not obsolete. A precondition for application of this rule is that there is an appropriate replacement SoftwareComponent. Supposing that this precondition is satisfied, consider the simple case where the BinaryEncoding that includes the obsolete software consists only of one ContentComponent and one SoftwareComponent. Applying the rule in this case involves several steps:

1. Bring the replacement SoftwareComponent into the PreservationEnvironment;
2. Register the replacement SoftwareComponent as appropriate for instantiating ContentComponents that have the same format as the ContentComponent in the current case;
3. Save the replacement SoftwareComponent in PreservationStorage;
4. Create a new BinaryEncoding that includes the replacement SoftwareComponent; and

5. Verify that the new BinaryEncoding expresses every PermanentFeature of the PreservationTarget it represents.

Each of these steps could consist of a single RuleTask, but a single step could entail several RuleTasks. The process of articulating this PreservationRule, and the need for the classes shown in Figure 13, is well illustrated merely by analyzing step 1 above.

Step 1 involves two RuleTasks. The RuledAction in both steps is the Submission of the replacement SoftwareComponent. The first RuleTask would be for a human Actor, in the PreservationRole of Authorizer, to authorize the introduction of the replacement SoftwareComponent. This would be accomplished by the Authorizer creating an appropriate data entry in PreservationManagementData. The RuleResult of this task would be the data entry in a SoftwareRegistry. There are two SuccessCriteria for this RuleResult. The first is that the data entry exists. Applying that criterion to the RuleDomain, the SuccessCondition is that the data identifies the approved replacement SoftwareComponent. The second SuccessCriterion is that the data entry was authorized. The corresponding SuccessCondition is that the entry must indicate that the authorization was done by an Actor identified as an Authorizer of this RuleTask in the RuleDomain. The two SuccessCriteria are identified as having a prior/next association. So an appropriate RuleEvaluator evaluates each of the RuleResults in turn according to the applicable Success Conditions. The second RuleTask in step 1 is for a human Actor, in the PreservationRole of Performer, to introduce a file containing the replacement SoftwareComponent into the Preservation Environment. The RuleResult of this is the existence of the file in the Preservation Environment. The applicable SuccessCriterion is verification that the file is present in the environment and the related SuccessCondition is that the file contains the authorized SoftwareComponent.

Full articulation of the PreservationRule requires comparable analysis of the remaining steps, breaking each of them down into RuleTasks; identifying the RuleResult(s) of each task; defining one or more SuccessCriteria to be applied for each RuleResult; applying each SuccessCriterion to the RuleDomain to determine an appropriate SuccessCondition; specifying a RuleEvaluator for each result; and identifying the AuthorizedRoles of any Actors in each RuleTask. For some steps, this analysis will be relatively simple. For example, step 2 would involve only one RuleTask analogous to the first RuleTask in step 1, and step 3 would have only one RuleTask, committing the file containing the SoftwareComponent to PreservationStorage and one SuccessCriterion, the presence of the file in storage. However, analysis of step 4 and especially step 5 would be quite complex. Notably, articulation of step 5 would include an element of the Domain Model not present in the prior steps: ControllingObject. The PreservationTarget would be a ControllingObject whose role would be to identify each of the PermanentFeatures that would have to be expressed by the new BinaryEncoding. In turn, each of these PermanentFeatures would be a ControllingObject used to determine if the PermanentFeatureExpression in the BinaryEncoding was appropriate.

Failure to satisfy a PreservationRule constitutes one or more PreservationProblems. A PreservationProblem is a condition that fails to satisfy one or more SuccessCriteria in the implementation of a PreservationRule. Accordingly, a SuccessCriterion must have one or more ProblemHandlingInstructions. A ProblemHandlingInstruction is a specialization of PreservationRule that specifies what is to be done when a problem is identified. When another PreservationRule fails to execute successfully, the RuleTask in the failed rule becomes the RuledAction in the ProblemHandlingInstruction. A ProblemHandlingInstruction has one or more SuccessCriteria that are those SuccessCriteria of the failed rule that were not satisfied. Any RuledObject in the RuleDomain of the failed rule to which an unsatisfied SuccessCriterion applies become a RuledObject in the RuleDomain of the ProblemHandlingInstruction.

6. What Can Be Done

The process of preservation starts with a Preservation Commitment, that is a decision to preserve one or more IntellectualEntities, which are thus qualified as PreservationTargets. A Preservation Commitment is an agreement, obligation or intention to preserve one or more PreservationTargets. There are three basic types of Preservation Commitment:

- Preservation Agreement: a Preservation Commitment voluntarily established between an Initial Source and a Preservation Director to preserve certain Preservation Targets, specifying terms and conditions applicable to their preservation;
- Preservation Obligation: a Preservation Commitment imposed by a Preservation Director on an Initial Source over whom the Preservation Director has authority; and
- Preservation Intention: a Preservation Commitment unilaterally articulated by an Initial Source to preserve one or more PreservationTargets. A Preservation Intention may or may not identify the Preservation Director responsible for deciding how preservation is carried out or the Preservation Service Provider responsible for carrying out preservation.

If any of the activities required to implement the Preservation Commitment is to be carried out by a Preservation Service Provider, depending on the type of commitment, either the Initial Source or the Preservation Director would negotiate a Preservation Service Contract with the provider. There could be several Preservation Service Contracts if different providers are to provide the necessary services The set of Preservation Service Contracts should detail what is to be preserved; what HeuristicInformation should be used in support of preservation; what Actors should be involved in which roles; what PreservationManagementInformation should be generated or collected; the PreservationRules that apply; and which providers are responsible for which services.

Preservation is accomplished in a Preservation Environment by implementing actions defined in the PaaST requirements. The requirements are grouped into sets of related Capabilities under three main headings: Preservation Action Services, Information Management Capabilities and Preservation Management Capabilities. Preservation Action Services encompass those actions which accomplish the preservation of InformationEntities. Information Management Capabilities enable creation, updating, maintenance, quality control and use of the data and information needed to carry out, manage and evaluate preservation. Preservation Management Capabilities enable assessment of whether PreservationActions have been carried out properly, whether the objects used in digital preservation are what they should be and, crucially, wether PreservationTargets have been preserved successfully. Preservation Action Services rely on both Information and Preservation Management Capabilities.

The following sections describe the major Services. Each section ends with a table identifying the high level requirements related to each Service. The numbers of the high level requirements match those in the complete requirements provided in chapter 10.

6.1. Preservation Action Capabilities

Preservation Action Services execute the actions that accomplish digital preservation, including bringing PreservationTargets into and under the control of a Preservation Environment, ensuring DigitalComponents are stored in a way that guarantees their integrity and accessibility, implementing changes in Preservation Targets or in the technology needed to operate on and provide access to them, and providing to authorized parties copies of PreservationTargets, access to information about them, and access to Manifestations.

6.1.1. Submission Processing Service

The Submission Processing Service encompasses the transfer and ingest of PreservationTargets into a Preservation Environment. By and large, the Submission Processing Service aims at determining

- that the PreservationTarget(s) contained in a Submission are ones which should be preserved under provisions of an applicable Preservation Service Contract;
- that all objects that should be included in a Submission are present and nothing extraneous is included;
- that adequate and accurate information needed to process the Submission, to preserve the PreservationTarget(s), and to identify and maintain relationships among objects is available; and
- that the DigitalComponents in the Submission are intact and have appropriate properties. If these conditions are satisfied, the PreservationTargets are accepted into the Preservation Environment and relevant data is captured as PreservationManagementInformation.

The depth and extent of examination and analysis in processing Submissions are specified in PreservationRules. The PaaST requirements for Submission processing provide for extensive, in depth review; however, Preservation Service Contracts may specify only a subset of these requirements as appropriate given the nature of the PreservationTarget, the objectives of preservation, the extent and reliability of existing information about the contents and the specific responsibilities of a particular Preservation Service Provider.

Submission Processing includes three main subcomponents. The first involves capturing information about a Submission. This subprocess produces a database that describes what is or should be included in a Submission; how the objects in the Submission are related; the terms and conditions governing the actual transfer, and the PreservationRules that apply to the submission.

The second part of Submission Processing is review of a NoticeOfIntentToSubmit. The NoticeOfIntentToSubmit is a PreservationManagementDocument that a Submitter would send to the PreservationServiceProvider prior to transmission of the Submission. A NoticeOfIntentToSubmit enables the Preservation Service Provider to create a data model of the Submission, as described under capturing information about a Submission, prior to receipt. This model comprises critical benchmarks for determining if a received Submission is what it should be. The NoticeOfIntentToSubmit is optional and in some cases it may not even be possible.

> *For example, digital PreservationTargets may be discovered after the fact in a body of analog materials accessioned by an archives.*

Where it is possible, however, requiring a NoticeOfIntentToSubmit would facilitate actual processing of a Submission as well as reducing the possibility that Submissions would have to be revised or even rejected.

The final stage of Submission Processing is receipt and review of a Submission. In the review, PreservationManagementInformation that accompanies the Submission would either be compared with the information derived from a NoticeOfIntentToSubmit or analyzed to create the data necessary for acceptance of the Submission. Once the data model of the Submission is finalized, it is used to compare with the objects included in the Submission. There are three possible outcomes of the review. The Submission may be accepted or rejected or, alternatively, the Submitter may be asked to address problems discovered in Submission Processing and, possibly, to submit a revised

Submission in whole or in part. If there are no problems with the Submission, or alternatively if all problems are resolved satisfactorily, the PreservationManagementData about the Submission is finalized and the DigitalComponents in the Submission are sent to PreservationStorage.

The Submission Processing requirements address the situation where Preservation Targets are transferred to a Preservation Environment for the first time. When Preservation Targets had been submitted previously to another Preservation Environment, some Submission Processing requirements might be waived if the process of transferring PreservationTargets and related PreservationManagementInformation to another Preservation Environment was sufficiently reliable. Alternatively, submission to the new Preservation Environment might involve invoking requirements that had not been addressed previously, or it might involve applying most or all of the Submission requirements, but only on a sample of the transferred objects.

Table 1, Submission Processing

1.1	Process PreservationManagementInformation in a Submission.
1.1.1	Process a PreservationManagementDocument That Describes a Submission.
1.1.2	Capture PreservationManagementData About a Submission.
1.1.3	Identify a PreservationRule Applicable to a Submission.
1.1.4	Store PreservationManagementData About a Submission.
1.2	Process a NoticeOfIntentToSubmit
1.2.1	Receive a NoticeOfIntentToSubmit
1.2.2	Review a NoticeOfIntentToSubmit
1.2.3	Approve a NoticeOfIntentToSubmit
1.2.4	Disapprove a NoticeOfIntentToSubmit.
1.2.5	Require Modification of a NoticeOfIntentToSubmit.
1.2.6	Process a Modified NoticeOfIntentToSubmit.
1.3	Process a Submission.
1.3.1	Receive a Submission
1.3.2	Identify a PreservationRule Applicable to a Submission
1.3.3	Review SubmissionInformation included with a Submission
1.3.4.	Process PreservationManagementData Included With a Submission.
1.3.5	Inspect an object in a Submission
1.3.6	Address a problem with a received Submission
1.3.7	Accept a received Submission
1.3.8	Reject a Submission

1.3.9	Send an accepted DigitalComponent to Preservation Storage.

6.1.2. Preservation Storage Service

Preservation Storage includes controls and methods implemented to ensure that the objectives of preservation are respected when DigitalComponents are sent to storage, maintained in storage and retrieved. As mentioned in the Introduction, the requirements do not address generic capabilities that can be expected to be available in any computing environment; therefore the requirements for Preservation Storage do not cover the entire scope of storage capabilities, but focus on ensuring that the DigitalComponents that instantiate PreservationTargets are properly controlled to ensure their preservation. The Preservation Storage requirements are neutral with respect to the physical media, hardware and software used. Rather they aim at managing the storage of DigitalComponents so that PreservationTargets can be instantiated with all PermanentFeatures unchanged.

Activities under Preservation Storage start with storing a DigitalComponent. There are three contexts in which a DigitalComponent is sent to storage. It is stored temporarily when received in a Submission, prior to acceptance of the Submission. The same conditions that apply to permanent storage must apply to temporary storage to ensure the DigitalComponent is not lost or corrupted while the Submission is being processed. The second context is when a Submission is accepted and the DigitalComponent is sent to permanent storage. The third is the permanent storage of a DigitalComponent created as part of a PreservationChange.

The second major activity under Preservation Storage ensures the integrity of a DigitalComponent in storage. At the basic level this entails ensuring that the bit string that constitutes a DigitalComponent remains unchanged. At a higher level it means ensuring the DigitalComponent continues to conform to its ComponentDescription and other relevant PreservationManagementData. Additional capabilities related to DigitalComponents include retrieval and retention management. Finally, Preservation Storage includes capabilities for managing both physical and logical packages. This capability forms a basis for management options to store groups of related objects together; for example, all DigitalComponents in a PreservationCollection or a PreservationNetwork.

Table 2, Preservation Storage

2.1	Store a DigitalComponent
2.2	Ensure the integrity of a DigitalComponent in storage
2.3	Retrieve a DigitalComponent from storage
2.4	Manage the Retention of DigitalComponent sin Storage
2.5	Manage a storage package

6.1.3. Preservation Change Service

A Preservation Change is a PreservationAction that produces a new BinaryEncoding of a PreservationTarget. A Preservation Change may be necessitated by the obsolescence of one or more DigitalComponents in a BinaryEncoding. Preservation Change might also be effected in accordance with a policy that requires standardizing formats for certain classes of objects. A

Preservation Change process might also be executed to respond to a request by an external Access Client for a copy of a DigitalComponent in a special format; however, this process might be limited to delivering the requested copy without creating a new Binary Encoding that is maintained in the Preservation Environment.

A Preservation Change may involve modifying or replacing any or all DigitalComponents in a BinaryEncoding, including ContentComponents, SoftwareComponents, and InstantiationComponents. In many instances, modifying or replacing one type of DigitalComponent will require changing other types.

> *For example, if a ContentComponent is normalized to a standard format to eliminate dependency on proprietary or obsolete software, it would be necessary to include software appropriate for the standard format in the new a BinaryEncoding.*

Preservation Change may be implemented for a single PreservationTarget or categorically for a set or class of PreservationTargets. The decisions on the scope of a Preservation Change is a matter of policy and, thus, can vary in different situations.

> *A common example of a categorical Preservation Change is when a SoftwareComponent is replaced. Often the same software is used in instantiating or manifesting all Content Objects in the native format the software operates on. Thus, when a SoftwareComponent is to be replaced, it might be preferable to create new a Binary Encoding for all Preservation Targets that have a Binary Encoding including the software that is to be replaced.*

> *A one-at-a-time approach to Preservation Change might be followed if the policy were to acquire the capability to migrate to a different format, but to apply it only in response to a request for Manifestation.*

As with Preservation Storage, Preservation Change requirements do not specify any particular change or even method for changing. Rather they define capabilities to ensure that any changes that affect either the DigitalComponents that instantiate PreservationTargets or the process of instantiation adhere to the objectives and basic requirements for preservation.

The Preservation Change Service includes Capabilities for both creating new BinaryEncodings and for capturing PreservationManagementData about them. These data are essential for PreservationManagement.

Table 3, Preservation Change

3.1	Create New BinaryEncoding
3.2	Create PreservationManagementData About a BinaryEncoding
3.2.1	Create PreservationActionData about the creation of a new BinaryEncoding.
3.2.2	The service shall provide the capability to create PreservationTargetData about a new BinaryEncoding.
3.2.3	Create data describing a PermanentFeatureExpression of a new BinaryEncoding.
3.2.4	Assign a new BinaryEncoding to a RuleDomain.

6.1.4. Access Service

Access is a capability that provides Actors with access to PreservationTargets, PreservationManagementInformation, and Capabilities. Thus, "access" refers not just to the ability to search for and receive access to PreservationTargets and related information, but more broadly to access to participation in actions in the Preservation Environment.

The Access Service includes three sequential sets of functions and the management of PreservationManagementData about Access. The first set of functions covers receiving requests for access to information, PreservationAction Capabilities, Information Management Capabilities or Preservation Management Capabilities. The second set determines if the requested Access is authorized and feasible. Authorization includes both conditions on access imposed on particular information resources and access rights and restrictions of individual Actors. The third set comprises responding to requests.

Capabilities included in the Access Service do not entail PreservationActions because accessing a PreservationTarget or information about it does not in itself result in any change to a PreservationTarget.

As with other Capabilities, Access capabilities do not extend to the full range of services generally included under the rubric of access to or dissemination of information, but focus on those that are a direct consequence of preservation. Thus Access Service requirements provide only minimal capability for discovery, specifically searching or browsing PreservationManagementInformation, and only to the extent authorized under applicable controls. The technological infrastructure of a Preservation Environment should facilitate interoperability between PaaST capabilities and generic services for search, discover and dissemination.

Table 4, Access

4.1	Receive a Request for Access
4.1.1	Receive Request for Access to Information
4.1.2	Receive Request to Access PreservationAction Capabilities
4.1.3	Receive Request to Access Information Management Capabilities
4.1.4	Receive Request to Access Preservation Management Capabilities
4.2	Determine Appropriate Response
4.2.1	Determine if the Requested Access Is Authorized
4.2.2	Determine if it is Possible to Provide the Requested Access
4.3	Respond to an Authorized and Feasible Request.
4.3.1	Respond to a Request for Access to information.
4.3.2	Respond to Request for PreservationAction Capabilities
4.3.3	Respond to a Request for Access to Information Management capabilities.
4.3.4	Respond to request for Access to Preservation Management Capabilities
4.4	Manage PreservationActionData about Access

6.2. Information Management Capabilities

Information Management Capabilities support creating and managing PreservationManagementInformation about objects, actions, and Actors. These Capabilities include organizing, managing and reporting information and data.

Information Management Capabilities support organizing information into classes and sets. A class is a group of objects all of which have at least one Feature in common. A class may have subclasses. Every subclass inherits all the Features of the superclass. A set is a group of objects demarcated by the associations among them. Members of a set can belong to multiple classes that do not have any common Feature defined at the class level. A set may have subsets, but the subsets do not necessarily inherit any Feature of the superset other than the attribute or attributes that determine membership in the set. Capabilities for managing a class or set of objects encompass defining the group and implementing it in the data management system. They do not address individual members of a group. Class and Set Management facilitate implementing different preservation policies and fine tuning a Preservation Environment to the types of IntellectualEntities being preserved and their technical characteristics.

6.2.1. Class Management

6.2.1.1. Class Definition

Class Management Capabilities support definition of classes of objects and their Features, and establish a conceptual framework for managing PreservationTargets. A class is any group of objects that have at least one Feature in common. The PaaST domain model defines classes needed in all Preservation Environments. Class Management Capabilities enable Actors to define additional classes, so that implementations of the requirements can be responsive to different needs.

Classes could be defined to group, and separate, records with different provenance.

For example, a government archives that preserves records from many different agencies may want to define the records of each agency as a separate class. Conversely, in an archives that only preserves records from a single Producer, such a subdivision would not offer any advantages.

Similarly, an organization that only preserves data that is publicly available would not need to classify Preservation Targets according to access conditions.

Classes should also be identified to facilitate using specific preservation techniques.

For example, in a situation where format migration is used to overcome obsolescence, classes should be declared to include all files in each format, regardless of provenance or access conditions. Conversely, in a situation where operating system emulation is the standard technique for overcoming obsolescence, a preservation class would include all the software objects that run under a given emulator and subclasses could be defined for the formats that run under each class of software objects.

Another basis for defining classes would be to facilitate preservation of Permanent Features.

For example, if documentary form is a PermanentFeature, then each document genre would define a class regardless of the provenance or digital format of the documents in that class.

Class Management deals with classes and subclasses, not with individual objects that are members of a class. Defined classes can be independent of the grouping of objects into SubmissionSets, PreservationCollections, and PreservationNetworks.

> *E.g., an aggregate of records is a PreservationCollection, but several record aggregates may contain documents of the same classes, such as letters and reports. Conversely, records (a type of PreservationItem) could belong to many different classes of information types; e.g., text, photograph, audio/visual, etc.*

Multiple, independent classification schemes may be implemented within a Preservation Environment.

> *E.g., textual documents may be classified according to document types, such as letter, report, directive. DigitalComponents might be classified by basic encoding method, such as character encoded or raster image. A document of a given type might be instantiated in either or both encodings. Objects might also be classified according to access restrictions.*

Table 5, Class Management

5.1	Define a Class
5.1.1	Implement a Class of Objects Defined in the PaaST domain model.
5.2	Provide the Capability for an Actor to Define a Class of Objects
5.3	Define associations between classes
5.4	Implement a Data Profile for an Attributable Class

6.2.2. Set Management

The PaaST domain model defines two top level sets: PreservationCollections and ManagementSets. A PreservationCollection is a group of objects that should be preserved together as a set. A ManagementSet is a group of objects that is established to facilitate accomplishing a specific management objective. The PaaST domain model defines three specializations of ManagementSet: SubmissionSet, PreservationActionSet, and PreservationNetwork. A SubmissionSet is a set of objects that are transferred to a Preservation Environment together. A PreservationActionSet is a set of objects that are subject to specified PreservationActions. A PreservationNetwork is a set of objects that is focused on a specific PreservationTarget and that are either closely related to it or contribute to understanding or using it.

Set Management includes general capabilities for defining a set and its membership, specifying relationships among members that derive from their membership, and instantiating a set. In addition, Set Management specifically supports defining PreservationCollections and ManagementSets.

Table 6, Set Management

6.1	Define a Set
6.2	Define the Membership of a Set
6.3	Specify an Association Between Members of a Set
6.4	Instantiate a Set
6.5	Define a Preservation Agregate

6.6	Define a ManagementSet
6.6.2	Define a PreservationActionSet
6.6.3	Define a SubmissionSet
6.6.4	Define a PreservationNetwork

6.2.3. PreservationManagementDocument Capabilities

A PreservationManagementDocument is a human readable object whose contents are about preservation or relevant to preservation management. PreservationManagementDocument Capabilities include management of each of the specializations of PreservationManagementDocument that are defined in the PaaST domain model. In addition, PreservationManagementDocument Capabilities enable Actors to define other specializations of PreservationManagementDocument, customized to different situations. These requirements also address generating, sending and receiving PreservationManagementDocuments, and extracting PreservationManagementData from PreservationManagementDocuments.

Tavle 7, Preservation Management Documents

7.1	Manage a PreservationManagementDocument Defined in the PaaST Data Model
7.2	Define a PreservationManagementDocument Specialization.
7.3	Generate a PreservationManagementDocument
7.4	Send a PreservationManagementDocument.
7.5	Receive a PreservationManagementDocument
7.6	Generate PreservationManagementData From a PreservationManagementDocument

6.2.4. PreservationManagementData Capabilities

PreservationManagementData is structured data about PreservationTargets, PreservationActions, PreservationRules and Actors. PreservationManagementData Capabilities include requirements for managing data entry for all subclasses of PreservationManagementData. Additional requirements specifically address the management of each of the subclasses of PreservationManagementData defined in the PaaST domain model; namely, PreservationActionData, PreservationTargetData, PreservationRules and Actors.

Table 8, Preservation Management Data

8.1	Manage Data Entry
8.2	Manage PreservationTargetData
8.2.1	Manage TargetDescriptionData

8.2.2	Manage TargetStateData
8.3	Manage DigitalTypeRegistries
8.4	Manage PreservationActionData
8.5	Manage PreservationRule
8.6	Manage Actor

6.2.5. Reporting Capabilities

These Capabilities support producing, sending, and managing reports about objects, actions and problems. Reporting requirements include generating reports in the form of predefined PreservationManagementDocuments, generating reports ad hoc, and sending and receiving reports.

Table 9, Reporting

9.1	Define the scope of a report.
9.2	Generate a report in the form of a defined PreservationManagementDocument.
9.3	Generate a report ad hoc.
9.4	Send a report
9.5	Receive a report

6.3.Preservation Management Capabilities

Preservation Management Capabilities include enforcing PreservationRules, assessing the status of preservation efforts and verifying the success of the preservation of PreservationTargets.

6.3.1. Implementing PreservationRules

The first subset of Preservation Management Capabilities supports Implementing preservation PreservationRules. The requirements address enforcing the rules, which are articulated using PreservationManagementData Capabilities, and collecting data about rule application.

Table 10 Preservation Rules

10.1	Define a PreservationRule
10.2	Enforce Rules
10.3	Capture PreservationManagementData about rule application

6.3.2. Assessment Capabilities

A Preservation Assessment involves inspecting PreservationManagementInformation or DigitalComponents, comparing related PreservationManagementData, or comparing PreservationManagementData with the DigitalComponents or realizations of the Manifestations they describe.

Inspection of PreservationManagementInformation encompasses examining not only the data it contains, but also their associations and associated objects.

> *For example, inspection of a PreservationTarget would include examining its PermanentFeatures, BinaryEncodings, ComponentDescriptions, PermanentFeatureExpressions and Manifestations.*

Inspection may be accomplished using software tools, where appropriate, or by observation by a human Actor. The results are captured as PreservationManagementData and may be published in a PreservationActionReport. The Inspection requirements provide for correlating data from multiple assessments of the same scope. This could be desirable in a variety of cases.

> *For example, if a new BinaryEncoding of a PreservationTarget it produced, it might be wise to inspect the PreservationNetwork of the PreservationTarget and compare it with prior inspections of the same network to see if all associations remain intact and implementable.*

The purpose of comparison in both cases is to ensure consistency or, alternatively, discover and address any discrepancies.

Comparing related data involves either comparing the values of the same data elements about a single object where the values were obtained by using different methods, from different sources, or at different times; comparing similar data about different objects; comparing PreservationActionData with data about objects or Actors involved in the actions; or comparing a PreservationRule with data that was used or generated in applying the rule.

Comparing PreservationManagementData to DigitalComponents and Manifestations tests whether the data accurately and adequately describes a DigitalComponent or an Manifestation.

Comparison may show either consistency or some discrepancy. A discrepancy is a situation where data values that should be identical are not; or data values that should be consistent are not; or data do not correspond to the things they describe. Any data discrepancy is flagged as a preservation problem. Preservation Handling capabilities are invoked to address problems.

Table 11, Assessment

11.1	Inspection
11.1.2	Capture PreservationManagementData from an inspection
11.1.3	Inspect objects
11.1.3.2	Inspect TargetDescriptions
11.1.3.3	Inspect TargetState
11.1.3.4	Inspect Manifestations
11.1.3.5	Inspect DigitalComponents
11.1.3.6	Inspect groups of related objects

11.2	Evaluate PreservationManagementData
11.2.1	Compare Related PreservationManagementData
11.2.2	Address Discrepancies in Compared PreservationManagementData
11.2.3	Analyze data
11.2.4	Address Discrepancies between PreservationManagementData and related objects
11.3	Report on Preservation Assessment

6.3.3. Verification

Verification Capabilities enable determining whether PreservationTargets have been preserved successfully and also whether HeuristicInformation does clarify the PreservationTarget or targets with which it is associated.

The first process in verifying preservation entails verifying that PreservationTargets can be instantiated. If a PreservationTarget cannot be instantiated, its preservation is, to say the least, incomplete. Minimal requirements for instantiating a PreservationTarget are that it has at least one BinaryEncoding that is operative; i.e., not obsolete; that a current BinaryEncoding is adequate to instantiate the PreservationTarget; that the DigitalComponents needed to instantiate the BinaryEncoding exist and are not corrupted; that a RuntimeVersion of the PreservationTarget can be instantiated; and that a Rendering of a MachineReadable PreservationTarget can be instantiated. Verifying whether a PreservationTarget is preserved includes determining whether its PermanentFeatures are correctly expressed in both DigitalComponents and Manifestations.

Verification requirements also address the preservation of PreservationCollections. Preservation of the collection entails preservation of its members. At the lowest level, a PreservationCollection comprises PreservationItems as its members; therefore, verifying the preservation of a PreservationCollections requires as a preliminary step verifying the preservation of the PreservationItems it includes. In the case of hierarchical Collections, Verification of PreservationCollections proceeds upward through members that are themselves Collections.

HeuristicInformation is information about a PreservationTarget that contributes to understanding and properly using it.   Requirements for verifying HeuristicInformation include determining that the PreservationTarget(s) with which it is associated exist and that the association is appropriate. This verification could be done by automated or human methods.

> *For example, a software tool could be used to determine if the structure and semantics of a textual document conform to the XML Schema associated with it. In contrast, determining whether a scholarly publication about the context in which the document was created, such as a history of an organization which kept the document as a record, actually clarifies the document will often require expert judgment.*

The generic requirements for verifying that a PreservationTarget can be instantiated also apply to HeuristicInformation. As in general, any failure to verify HeuristicInformation should be flagged as a problem and referred to Problem Handling.

Table 12, Verification

12.1	Capture Data about Verification

12.2	Handle a Problem Discovered in Verification
12.3	Verify PreservationTarget Preservation
12.4	Verify Preservation of a PreservationCollection
12.5	Verify HeuristicInformation

6.3.4. Problem Handling

Problem Handling Capabilities include recognizing problems, recording PreservationManagementData about problems and addressing problems. The requirements specify recognizing problems related to PreservationActions, PreservationTargets, associations between objects, and PreservationManagementInformation. Addressing problems includes characterizing the severity of a problem, applying the appropriate ProblemHandlingInstruction, generating a ProblemReport, notifying the designated ProblemResolver and processing responses.

Table 13, Problem Handling

13.1	Recognize Problems
13.2	Record PreservationManagementData about Problems
13.3	Address Problems

7. Archival Preservation

PaaST requirements are articulated to enable preservation of practically any type of digital information for as wide a variety of objectives and purposes as possible. Potential purposes of preservation can be divided into two basic categories, prospective and retrospective. The prospective purpose aims at maximizing the potential for reuse of PreservationTargets, including repurposing them for uses that may be unrelated to those for which they were created.

> *An example of a prospective purpose is the preservation of data about natural phenomena for use in scientific research. Reuse of preserved data could be for purposes not imagined in the projects in which the data was originally created or captured. For example, data generated in clinical trials might be used to test a new method of statistical analysis.*

Maximizing the potential for reuse may entail changing the PreservationTargets; such as converting data from one system of measurement to another, calculating parameters other than those used in the original research, or reorganizing data according to a different model or taxonomy.[31]

Retrospective preservation, in contrast, aims at optimizing the potential for recognizing, in the primitive sense of knowing again, the message a PreservationTarget was originally intended to convey. Archival preservation, the preservation of PreservationTargets as Records, is preservation in context. The meaning of a Record as such can only be recognized in the light of information about the context in which the PreservationTarget was a Record. A Record is defined in archival science as "A document made or received in the course of a practical activity as an instrument or a by-product of such activity, and set aside for action or reference."[32] This definition has multifold implications for preservation. While 'document' is often thought of as synonymous with textual information, this constraint is not valid in archival science or its sister discipline, diplomatics. For digital preservation, PaaST adopts a definition of 'Document' from diplomatics. A Document is a persistent and indivisible unit of information that has a fixed form and stable content.[33] A Record is a document in a context of use. It may not be possible to discern the message a Record conveyed in a particular context without knowledge of that context because Records include Documents that were created prior to, and in distinctly different contexts than those in which they are Records. Thus, the essential contextual information may not be contained in the Document itself.

> *Using the example cited in the description of prospective preservation, looking at a set of clinical trial data one could never infer solely from the data that they were used in mathematical research on methods of statistical analysis.*

At a minimum, archival preservation requires information about:

1. Provenance: information about the practical activity in which a Document was used, who used it, and how it was instrumental or produced in that activity;
2. Archival Bond: the set of relationships among Records that arise as a consequence of their use in the same activity.

   Knowledge of other Records used in the same activity and how they are related can be essential to understanding the significance of a Record.

[31] Any such changes should be subject to rigorous controls to ensure that the data remains valid.

[32] IP Dictionary

[33] Duranti, Luciana. Diplomatics: New Uses for an Old Science. Lanham, MD: Scarecrow Press, 1998.

3. Authenticity: the ability of a Record to communicate the message it was intended to provide in the context in which it was a Record.

   An assessment of authenticity enables us to determine whether the possibility of recognizing the meaning of a Record has been impacted by the passage of time. Ideally, it should not be.

The first two facets of archival context determine what a Record is. The third determines that the Record has not changed in any significant way.

Archival preservation can be accomplished using the PaaST requirements by adding appropriate specializations to existing superclasses to the PaaST domain model. As indicated in Figure 14, Archival PreservationTargets, the specializations needed for archival preservation have the same relationships as their generalizations. The classes needed for archival preservation are labelled with the namespace, “Archival Preservation.”

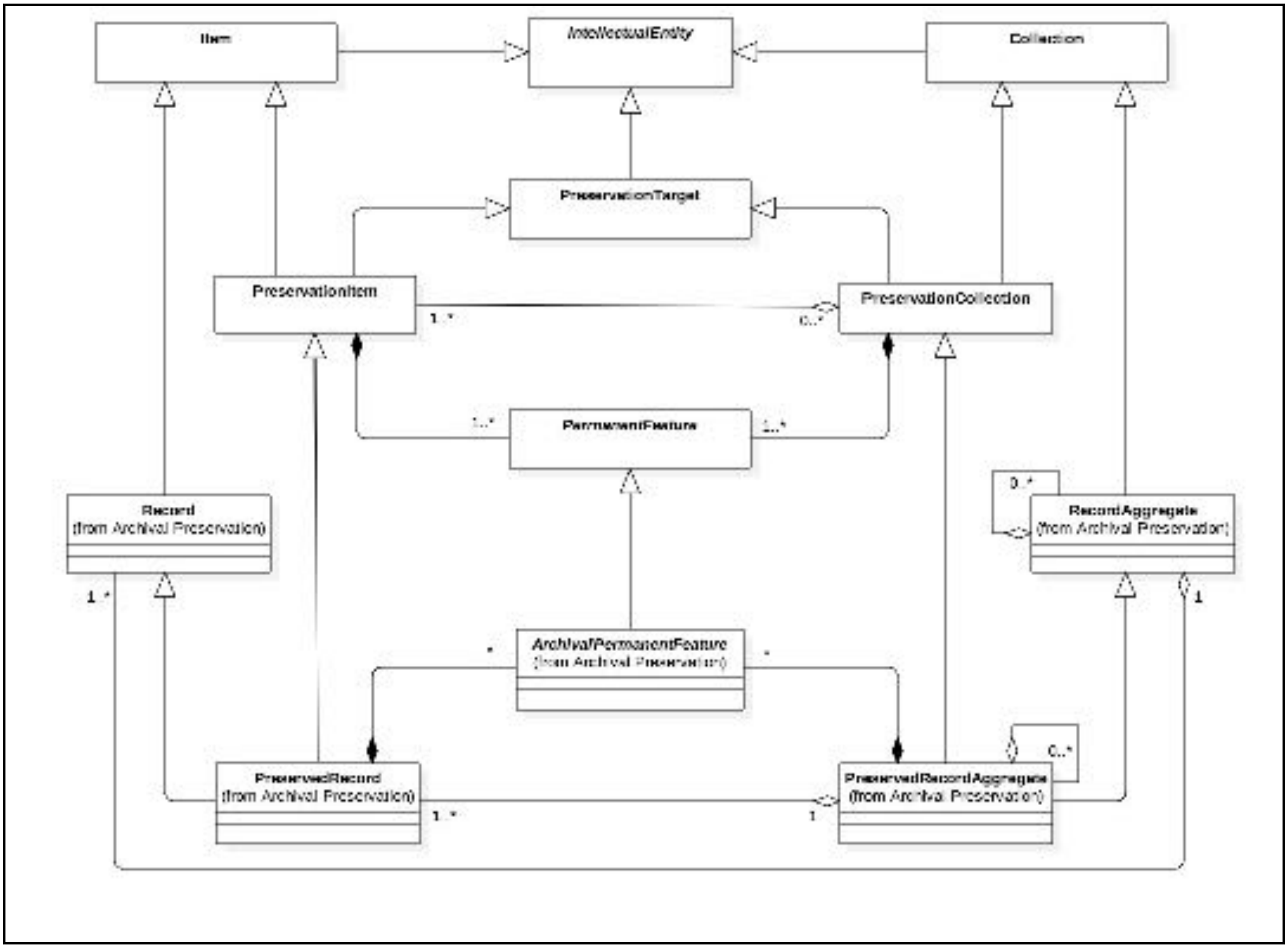


Figure 14, Archival PreservationTargets

The key specializations are the two classes, Record and RecordAggregate. A Record is a type of Item and a RecordAggregate is a type of Collection. Similarly, a PreservedRecord is a type of PreservationItem and a PreservedRecordAggregate is a type of PreservationCollection. As specializations of PreservationTarget, both PreservedRecord and PreservedRecordAggregate have

PermanentFeatures. Both need PermanentFeature for identification, manifestation and integrity, as described in 4.4, Preservation Targets, above. A PreservedRecord may have PermanentFeatures that derive from the type of document that it is, as well as ArchivalPermanentFeatures that relate specifically to its status as a Record. The class, ArchivalPermanentFeature, is a specialization of PermanentFeature. A PreservedRecordAggregate must also have ArchivalPermanentFeatures and it may have other PermanentFeatures in common with other Collections that have the same type of structure.

A RecordAggregate contains Records. It may contain lower level RecordAggregates. The one significant difference between the general case and archival preservation is in the multiplicity of these aggregation associations. Because a Record is, by definition, a document in a specific context of use, a Record must belong to a RecordAggregate, while an Item does not necessarily belong to a Collection. Moreover, because the context determines the meaning of a Record, a Record can only belong to one RecordAggregate.[34] A single Document, or copies of the same Document, can be different Records in different contexts; i.e., when used by different entities or for different purposes.

The multiplicity on the Record end of the aggregation association is also different from the general case. While in the general case, a Collection is a group of several Items, a RecordAggregate may contain only one Record. This necessity derives from another basic concept of archival science, that of the archival fonds. An archival fonds is the highest level RecordAggregate and is defined as “The entire body of records of an organization, family, or individual that have been created and accumulated as the result of an organic process reflecting the functions of the creator.”[35]

A Record may a member of multiple Collections that are not, and are basically independent of, RecordAggregates. Records of different records creators might be grouped into Collections based on type of document, such as correspondence or report, or on the type of activities in which they were used, such as finance or student enrollment. Digital Records could be grouped according to their data formats. Such Collections would constitute ManagementSets, as described above in section 5.3, rather than RecordAggregates.

Archival preservation uses and generates PreservationManagementInformation. The special information needed for archival preservation consists of subtypes of the general classes of PreservationManagementInformation, as shown in Figure 15, Archival Preservation Information. Archival preservation uses RecordsManagementDocument and RecordsManagementData, specializations of PreservationManagementDocument and PreservationManagementData, respectively. Given that records management can vary significantly across different records keepers, a Preservation Director can make use of the capabilities for defining subclasses of PreservationManagementDocument, provided by requirement numbered 5.2.1.13, and for defining subclasses of PreservationManagementData, provided by requirement numbered 5.2.1.14, to accommodate different records management practices. Furthermore, RecordsManagementDocument and RecordsManagementData are attributable classes, enabling customization down to the level of document and data instances.

[34] This model does not exclude the possibility of preserving multiple provenance in a PaaST implementation. Multiple provenance could be preserved by identifying a Document as a Record of different RecordsCreators, possibly in different AreaOfActivity, and different RecordAggregates.

[35] Pearce-Moses, Richard. A Glossary of Archival and Records Terminology. Chicago: Society of American Archivists, 2005. http://www.archivists.org/glossary/index.asp

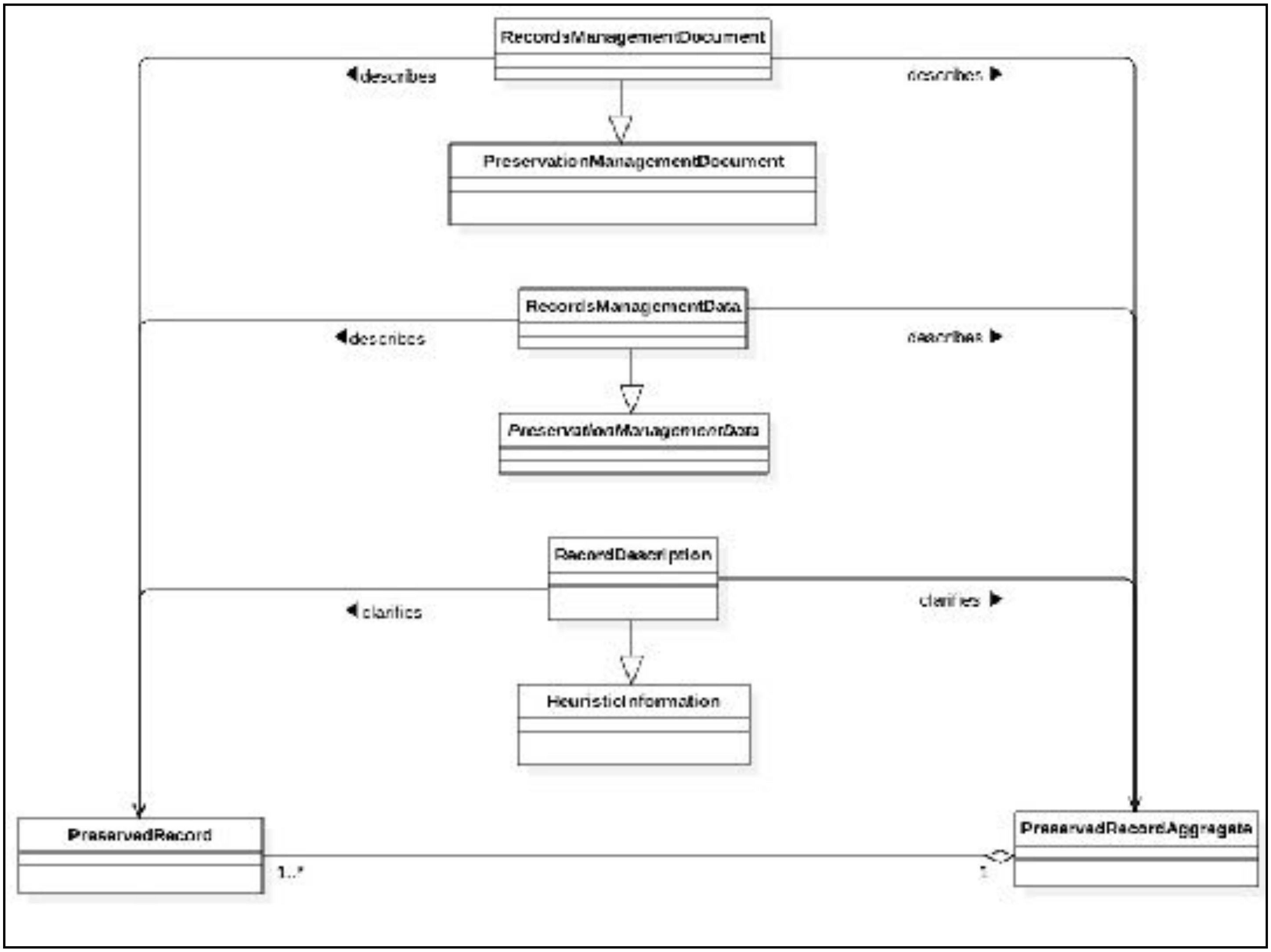


Figure 15, Archival Preservation Information

Figure 15 also includes a specialization of HeuristicInformation, RecordDescription. This class is especially important in archival preservation because descriptions may be essential to understanding the context of records.

7.1. Archival Attributable Classes

Record and RecordAggregate are attributable classes, adopted from the Object Management Group's specification, Records Management Services (RMS),[36] as described in section 9.5.3 below. Special attributes may be assigned to instances of these classes and their subclasses to meet the needs of different situations.

Having classes that can be customized in different situations is especially valuable in archival preservation because often, though not always, Records are subject to a records management regime prior to being submitted to a Preservation Environment. Information and data generated and kept in a records management system can be valuable in preserving Records; however, that

[36] Object Management Group. Records Management Services (RMS). Version 1.0 November 2011. http://www.omg.org/spec/RMS/1.0.

information can vary significantly in different legal, cultural, organizational and personal settings, as well as with the use of different records management software. Even when the same records management approach is implemented by several records creators, the implementations can have significant variations. Such variations can be accommodated by means of attributable classes.

Archival preservation in PaaST builds further on the RMS specification. The two have different orientations. The RMS specification addresses the management of current Records, while PaaST addresses closed Records. A closed Record not only has a fixed form and stable content, but its archival bond is unalterable. The activity is which it was a Record is finished and, therefore, no additional relationships can be established in the context of that activity.[37] Nonetheless, an RMS implementation can be the source of Records submitted to a Preservation Environment.

Figure 16, Managed Record Preservation, shows how elements from the RMS model can be integrated with PaaST model elements, as well as illustrating how PaaST can be adapted to different situations. The model elements in Figure 16 are illustrative only. They constitute neither a complete mapping of the RMS model to PaaST nor a definitive illustration of adapting classes to different situations. Classes in Figure 16 with a tan background are imported from the RMS specification.

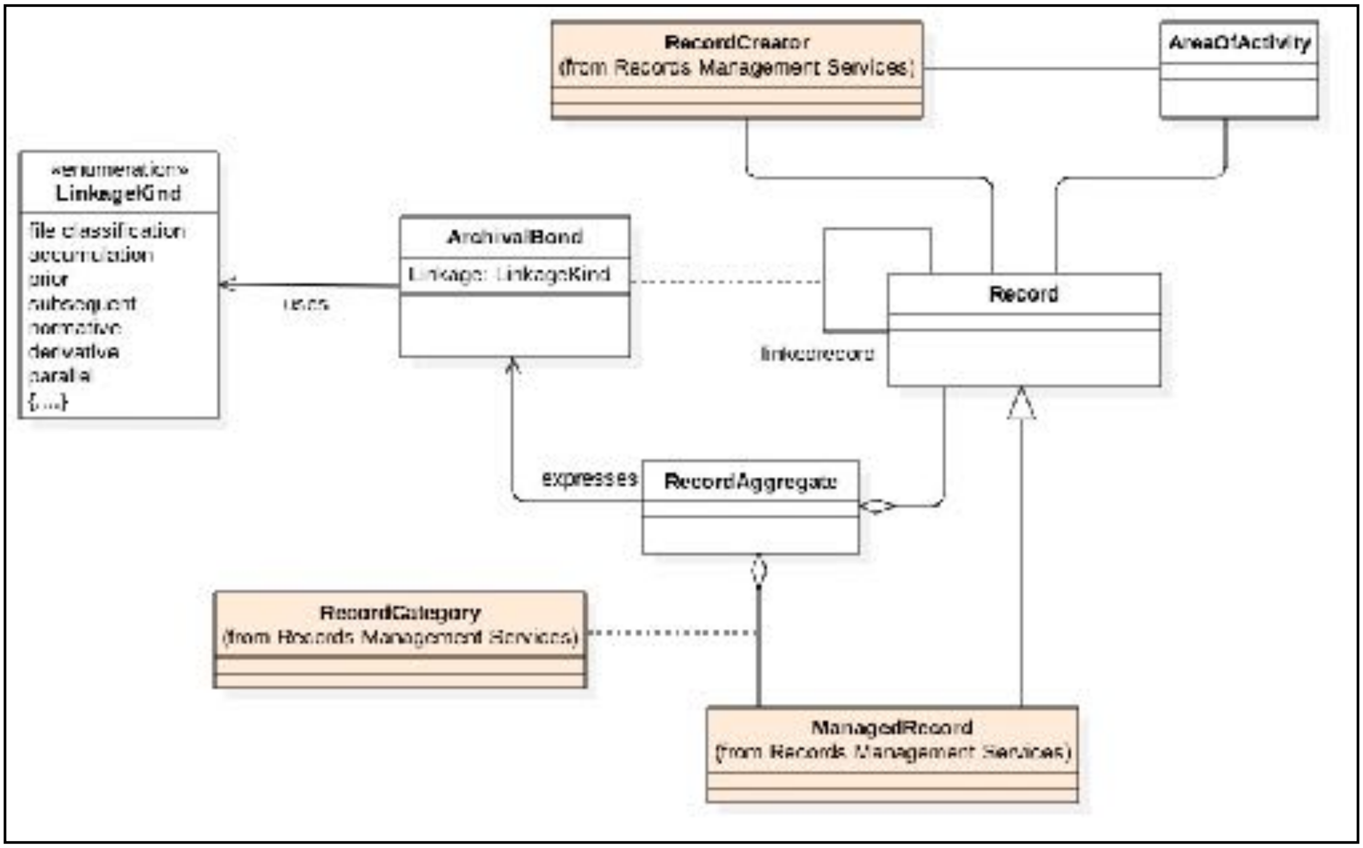


Figure 16, Managed Record Preservation

Figure 16 introduces the RMS classes, ManagedRecord, RecordCategory, and RecordCreator. Put simply, a ManagedRecord is a subtype of Record that has been subjected to a records management regime. Through a RecordCategory, records management assigns a ManagedRecord to a RecordAggregate, an association that must be preserved with the Record. RecordCategory can be

[37] A document that is a record could be used in other activities; however, because a record is by definition a document in context, the same document in a different context is a different record.

attributed with the specific classifications used in a particular recordkeeping system. A RecordCreator is the party that produced or acquired and used a Record in the course of a practical activity. Identification of the RecordCreator is essential in defining a Document as a Record.

Figure 16 also introduces two new PaaST classes, AreaOfActivity and ArchivalLink. AreaOfActivity identifies the activity in which a RecordsCreator used a Record. It is the second essential element of provenance. ArchivalLink is an association between two Records that is part of the network that constitutes the archival bond. This association between Records arose as a consequence of their both being used in the same activity. Commonly, ArchivalLinks are expressed in a persistent manner by grouping Records together in a RecordAggregate. Hence Figure 16 includes an association from RecordAggregate to ArchivalLink showing the RecordAggregate as expressing the ArchivalLink. The membership of a ManagedRecord in an RecordAggregate is indicated by its RecordCategory.

ArchivalLink also indicates the flexibility of PaaST. As Figure 16 shows, ArchivalLink is also an association class that specifies a relationship between Records. As revealed by the enumeration, LinkageKind, ArchivalLinks are not necessarily limited to those expressed by RecordAggregate or RecordCategory. Some records are never subjected to records management, but simply accumulated over time by the same RecordCreator. Thus, one kind of ArchivalLink is “accumulation.” Further, even if a RecordCreator never organizes Records into aggregates, there is a prior/subsequent relationship between an outgoing piece of correspondence that replies to one received by the RecordCreator. Preserving prior/subsequent links can be very important; for example, in preserving email. Other types of relationships can be defined as kinds of ArchivalLinks as needed.

7.2. ArchivalPermanentFeatures

As stated above, the class, ArchivalPermanentFeature, is a specialization of PermanentFeature. ArchivalPermanentFeatures must be defined for every PreservedRecord and PreservedRecordAggregate in order to achieve archival preservation.

The minimal ArchivalPermanentFeatures for every PreservedRecord are the identities of the RecordCreator and the RecordAggregate. Implementing additional ArchivalPermanentFeatures enhances the quality of archival preservation. The following ArchivalPermanentFeatures are derived from the results of earlier InterPARES projects about data that support the authenticity of Records:

ArchivalPermanentFeatures that support the identification of a Record sufficiently to distinguish it from all other Records are:

- names of the persons concurring in its creation,
- date(s) and time(s) of issuing, creation and transmission,
- the matter or action in which it participated,
- the expression of its archival bond,
- documentary form,
- BinaryEncoding,
- the indication of any attachment(s),
- digital signature, and
- name of the person responsible for the business matter.

ArchivalPermanentFeatures that support the integrity of a PreservedRecord are:

- name(s) of handling persons over time,
- name of person responsible for keeping the record,
- indication of annotations,
- indication of technical changes,
- indication of presence or removal of digital signature, and
- dates/times of records management transactions.

If any of these ArchivalPermanentFeatures are not explicit either in the Record or in the records management system used to keep it, they irrelevant to preservation. As with PreservationTargets in general, the PermanentFeatureExpression of each ArchivalPermanentFeature must be specified, identifying where and how it is expressed and how the expression can be verified. In the case of PreservedRecords this can be complicated because some ArchivalPermanentFeatures may not be contained or adequately expressed in the Records, but in related IntellectualEntities, such as in a records management database or in the documentation of the filing system. In such cases, the related entities should be designated as PreservationTargets, identified as parts of a PreservationNetwork that includes the PreservedRecords, and preserved with them.

The minimal ArchivalPermanentFeatures for every PreservedRecordCollection are association with the RecordCreator, the date span of the aggregate, the total number and identifiers of the Records or lower level RecordAggregates that are members of it, and the arrangement, ordering or relationships of the members of the RecordAggregate.

8. Evaluating Success

There are many different facets that could be taken into consideration in evaluating the success of a preservation program, such as whether Submissions are carried out efficiently and effectively, whether Access services provide clients with requested information and enforce restrictions appropriately. Such considerations can be addressed through normal management methods. This chapter addresses the core of preservation: whether PreservationTargets are preserved successfully. In brief, successful preservation means that a PreservationTarget can be instantiated appropriately, that the Manifestation has all the PermanentFeatures of the PreservationTarget, that each PermanentFeature corresponds to the benchmarkValue for that feature and, as appropriate, the information conveyed by a PreservationTarget is clarified by HeuristicInformation associated with it. While the criteria for success are few, there are many factors that could be the cause of failure to satisfy one or more of the four criteria. The following table lists questions can be used to determine whether preservation has been successful. The numbers in the second column of the table identify the highest level requirements that provide the capability for addressing the question. In many cases, these requirements are broken out to a greater level of granularity in the tables of requirements in chapter 10.

Note that questions about PreservationCollections assume that successful preservation of the individual PreservationItems that are members of a Collection is assessed in the first group of questions, about PreservationTargets. Success in preserving hierarchical Collections can be evaluated by starting with the lowest level Collection, whose members are Items, and proceeding iteratively upwards applying the questions about PreservationCollections to successive levels.

Similarly, questions relating to evaluating the success of archival preservation address only those aspects of Records and RecordAggregates that are different than generic PreservationItems and PreservationCollections. Thus, for example, after question 34 is answered by identifying the ArchivalPermanentFeatures of a PreservedRecordAggregate, the successful preservation of the RecordAggregate needs to be further explored by addressing questions 22 to 33.

Table 14. Questions Relevant To Evaluating Success

1	12.3	Is a PreservationTarget preserved?
2	12.3.1	Does a PreservationTarget have at least one operative, i.e., not obsolete, BinaryEncoding?
3	12.3.1.1	Is data about an operative BinaryEncoding appropriate?
4	12.3.1.1.1	Is data bout the ComponentDescriptions in an operative BinaryEncoding appropriate?
5	12.3.1.1.1.1	Does each operative BinaryEncoding include a set of ComponentDescriptions suitable for instantiating the PreservationTarget?
6	12.3.1.1.2	Does an operative BinaryEncoding include appropriate Manifestations?
7	12.3.1.1.2.1	Does an operative BinaryEncoding include at least one RuntimeVersion of the PreservationTarget?
8	12.3.1.1.2.2	Does an operative BinaryEncoding of a human-readable PreservationTarget includes at least one Rendering of the PreservationTarget?
9	12.3.1.1.3	Does an operative BinaryEncoding express every PermanentFeature of the PreservationTarget?

10	12.3.1.1.3.1	Does an operative BinaryEncoding include at least one PermanentFeatureExpression for every PermanentFeature of the PreservationTarget?
11	12.3.1.1.3.2	Is each PermanentFeatureExpression appropriate for the element in which it is expressed?
12	12.3.1.1.3.3	Is each PermanentFeatureExpression verifiable?
13	12.3.1.2	Can the DigitalComponents described by a BinaryEncoding be used to instantiate the PreservationTarget?
14	12.3.1.2.1	Does each DigitalComponent identified in the operative BinaryEncoding exist in the PreservationEnvironment?
15	12.3.1.2.2	Does each DigitalComponent match its ComponentDescription in the BinaryEncoding?
16	12.3.2	Can a PreservationTarget be correctly manifested?
17	12.3.2.1	Can a RuntimeVersion of a PreservationTarget be correctly loaded in a computer?
18	12.3.2.2	Can a Rendering of a human-readable PreservationTarget be correctly output from a RuntimeVersion.
19	12.3.2.3	Does a Manifestation preserve the PermanentFeatures of a PreservationTarget?
20	12.3.2	Can each specified Rendering of a PreservationTarget be generated from the operative BinaryEncoding?
21	12.3.2.1	Does each specified Rendering of a PreservationTarget correctly display the PermanentFeatures associated with it?
22	12.4	Is a PreservationCollection preserved?
23	12.4.1	Does a PreservationCollection have at least one operative BinaryEncoding?
24	12.4.2	Does an operative BinaryEncoding of a PreservationCollection identify its members?
25	12.4.2.1	Does an operative BinaryEncoding of a PreservationCollection satisfy criteria for membership?
26	12.4.2.2	Does an operative BinaryEncoding of a PreservationCollection describe any associations of its members?
27	12.4.3	Does the BinaryEncoding express every PermanentFeature of the PreservationCollection?
28	12.4.3.4	Is the expression of every PermanentFeature verifiable?
29	12.4.4	Does the operative BinaryEncoding include all members that should be in the PreservationCollection?
30	12.4.5	Can the PreservationCollection be instantiated correctly?
31	12.4.5.1	Does a Manifestation of the PreservationCollection include the correct members?
32	12.4.5.2	Does a Manifestation of the PreservationCollection correctly implement associations among members?
33	12.4.5.3	Does a Manifestation of the PreservationCollection correctly express the PermanentFeatures of the Collection?
34	12.5	Does HeuristicInformation contribute to the understanding ad appropriate use of a PreservationTarget?

35	12.6	Is a Record preserved successfully?
36	12.6.1	Is a PreservationTarget preserved as a Record?
37	12.6.1.1	Is a PreservedRecord associated with a PreservedRecordAggregate?
38	12.6.1.2	Does a PreservedRecord have ArchivalPermanentFeatures?
39	12.6.2	Is a RecordAggregate preserved successfully?
40	12.6.2.1	Does a PreservedRecordAggregate have ArchivalPermanentFeatures.

While these questions are stated in a singular form appropriate to evaluate the preservation of a single PreservationTarget, they could readily be rephrased to apply to a broader scope; such as a set of PreservationTargets or all the PreservationTargets that are our should be preserved in a Preservation Environment.These questions can be answered using data generated by applying the PaaST requirements for Inspection, Evaluation of PreservationManagementData and Verification.

In effect, each of the thirty-nine questions defines a facet of the criteria for successful preservation. Hence, the likelihood of success could be bolstered by reformulating the questions into SuccessCriteria for PreservationRules. As with PaaST in general, the questions and SuccessCriteria can be fine tuned to correspond to the objectives, needs and empirical realities of different situations.

8.1. Success in Archival Preservation

Success in archival preservation can be evaluated using the criteria and questions set out above. The questions should be refined specifically to address the actual ArchivalPermanentFeatures of PreservedRecords and PreservedRecordCollections. In addition, methods of tools designed to assess the authenticity of records could be used, with the data and documents produced in such assessments added as specializations of RecordsManagementData and RecordsManagementDocument, as appropriate.

9. Implementation and Customization

PaaST requirements are intended to support as broad a range of situations as possible, including differences in what is being preserved, the objectives of preservation, the policies that govern preservation, who is involved in what capacity and even, to some extent, how preservation is accomplished. This broad range entails a need for flexibility and customization in implementing the requirements. At the most general level, customization can be described as the determination of the Preservation Environment. Determining the Preservation Environment involves both defining the overall environment and deciding on the division of responsibilities that distinguishes different local Preservation Environments within the overall environment.

Options for customization of Preservation Environments include designation of PermanentFeatures, selection of the requirements that are to be implemented within a global or local Preservation Environment, articulation of PreservationRules, definition of ManagementSets, and refinement of the PaaST domain model.

9.1. Customizing Requirements

Paast requirements can be customized in two ways: addition and selection. While there are 1,350

requirements set out in chapter 10, the set is intentionally incomplete in order to enable the approach embodied in the requirements to be implemented in the widest possible variety of contexts. The requirements in chapter 10 cover the broad range of functionality and data needed to preserve digital information and to manage preservation. They need to be supplemented by breaking down existing requirements into more precise stipulations appropriate to different situations.

Most PaaST requirements are formulated with the phrase, "The service shall provide the capability to …." The intent is to ensure that the requirements provide a sufficient and trustworthy basis for digital preservation while allowing the Preservation Director to decide which requirements are needed in given situation. Such decisions enable the assignment of preservation requirements to multiple Preservation Service Providers, such as executing some capabilities in-house and assigning others to external contractors. PermanentFeatures

Probably the most important option for customizing an implementation of PaaST is the designation of PermanentFeatures because this action determines fundamentally what it means to preserve the IntellectualEntities selected as PreservationTargets and, more than anything else, whether preservation is successful. Detailed description of PermanentFeatures is found in section 4.2, PreservationTargets.

9.2.PreservationRules

A requirement is a rule. The PreservationRules requirements, numbered 10, enable a Preservation Director to add further rules in the form of specifications related to the objects acted on in executing PaaST requirements, the criteria and conditions for successful execution, actions to take when execution is not successful, the Actors who may or should be involved in carrying out a requirement, and the PreservationRoles in which they are authorized to act.

9.3. Management Sets

Another method of customization is the definition of ManagementSets, as described in section 5.3. ManagementSet, and its specializations, PreservationActionSet, SubmissionSet, and PreservationNetwork, can be defined to address different circumstances and objectives. Once defined, a ManagementSet can be the object of a PreservationRule. These capabilities are enabled by the requirements specified under the heading, ManagementSets, numbered 6.

9.4. PreservationManagementDocuments

The PaaST domain model defines several types of PreservationManagementDocuments that are needed in a Preservation Environment, but permit a Preservation Director to specify data elements that should be included as well as the precise form of each document. These capabilities are enable by the requirements specified under the heading, PreservationManagementDocuments, numbered 7. In addition, requirements under Reporting enable a Preservation Director to define additional types of documents.

9.5. Refinement of the PaaST domain model

There are several means of extending and adapting the domain model to suit different situations. They inclusive of capabilities for defining additional classes as specializations of the basic classes in the domain model, redefinition of model elements, and specification of attribute profiles. The first two options make use of capabilities articulated in the UML standard. The last option adopts a pattern from the OMG Records Management Services specification. Refinement of the model is supported by Class Management requirements, numbered 5.

9.5.1. Specialization

UML supports the definition of more general/more specific relationships between two classes. The relationship itself is called “generalization,” and this term is also used to qualify the more general class in the relationship. The more specific class is called a specialization of the more general one. A specialization inherits all the Features of its generalization, and it may have additional Features that are not defined at the more general level.

Figure 17, Example of Specialization, illustrates specialization of the general class, IntellectualEntity, with a specialization, Book. IntellectualEntity is shown in Figure 17 with three attributes: content, form and genre.[38] Book, as a specialization of IntellectualEntity, inherits all three attributes, but specializations can have additional Features that are not present in the superclass. Figure 17 shows Book with two additional attributes, Title and author, that are appropriate to this type of IntellectualEntity.

The PaaST domain model includes several classes that are basic to digital preservation and that can be specialized in different contexts. These classes are described in chapter 4, What’s Involved: PaaST Classes. Specializations can be added using the capabilities defined under the heading of Class Management in the PaaST requirements. One area where every Preservation Director should consider adding specializations is to the class, IntellectualEntity.

> *For example, a data center that collects observational data in a given scientific domain might not have need for many of the general specializations of IntellectualEntity identified in section 4.1, Types of IntellectualEntity; however, it might find it advantageous to define several, more precise specializations reflecting the different types of data sets and heuristic information common in the domain.*

[38] The attributes shown in Figure 13 are for purposes of illustration only and are not intended as complete specifications of the attributes of the two classes.

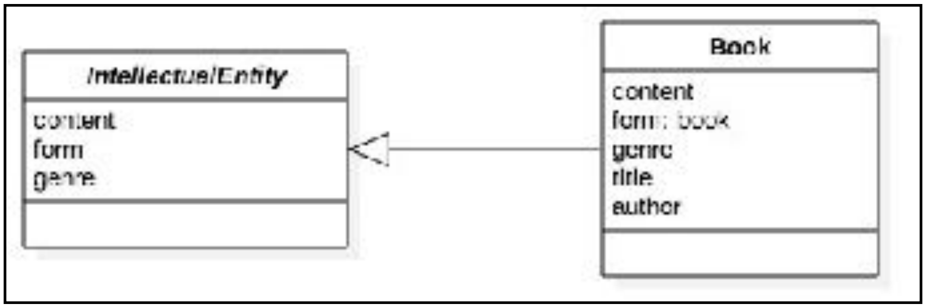


Figure 17, Example Of Specialization

9.5.2. Redefinition

A redefinable element in UML is a model element that is defined in a specific context, but may be redefined in a different context, where context is determined within the applicable domain model. For example, if the element is a Feature, the context is the class that owns the Feature.

Figure 17 includes a simple example of a redefined element in the attribute, form. At the general level of IntellectualEntity, form could be any pattern or structure that is capable of communicating information. For the specialization, Book, however, the form must be that of a book. Thus Book redefines form to restrict the attribute.

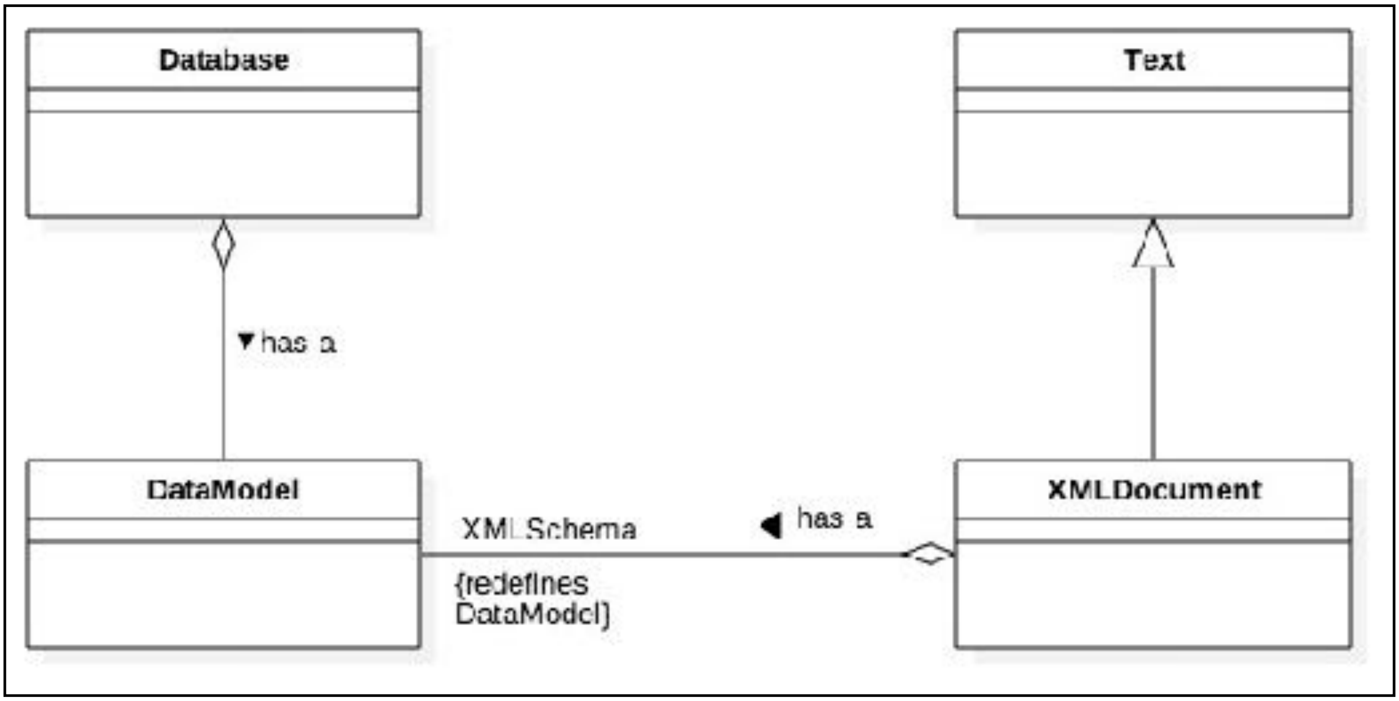


Figure 18, Example Of Redefinition

The context of redefinition could also be determined by associations between or among classes. This is illustrated in Figure 18, Example of Redefinition. The two classes at the top of the figure, Database and Text, are specializations of IntellectualEntity. Database has a part, DataModel, which defines the structure of data elements in the Database. Text has a specialization, XMLDocument. An XMLDocument is a type of semi-structured data which also has a DataModel, in this case an XMLSchema. XMLSchema redefines DataModel in the context of XMLDocuments.

9.5.3. Attributable Objects

PaaST provides another method for customization by adopting the pattern of the Attribute Profile Package from the OMG Records Management Services (RSM) specification.[39] An Attribute Profile enables redefinition of attributes of individual instances of a class, rather than redefining them for an entire class. A class whose instances can be modified in this way is called an AttributableClass.

The RMS Attribute Profile diagram is included below as Appendix 2. The RMS Attribute Profile can be used in PaaST, with the only major changes being a different enumeration of AttributableClasses and replacement of the place holder, DocumentType with IntellectualEntityType. In PaaST, AttributableClasses include IntellectualEntity, Item, Collection, BinaryEncoding, ComponentDescription, PreservationTarget, PreservationItem, PreservationCollection, PermanentFeature and PermanentFeatureExpression. Specializations of these AttributableClasses are also AttributableClasses.

Attributable objects should be especially useful in digital preservation when there are PreservationTargets, such as works of digital art, that have unique or highly uncommon Features.

[39] Object Management Group. Records Management Services (RMS). Version 1.0 November 2011. Section 8.2, Attributable Profile Package, pp. 15-23. http://www.omg.org/spec/RMS/1.0,

10. PaaST Requirements

10.1. Preservation Action Services

10.1.1. Submission Processing

	Process Submission	
Identifier	**Requirement**	**Calls**
1.	The service shall provide the capability to process a Submission.	
	Receive a Submission	
1.1	The service shall provide the capability to receive a Submission.	
1.1.1	The service shall provide the capability to store a received Submission.	
1.1.2	The service shall provide the capability to receive a PreservationManagementDocument that contains SubmissionInformation.	
1.1.2.1	The service shall provide the capability to store a received PreservationManagementDocument that contains SubmissionInformation.	
1.1.3	The service shall provide the capability to receive SubmissionData.	
1.1.3.1	The service shall provide the capability to store received SubmissionData.	
1.1.4	The service shall provide the capability to receive a DigitalComponent as part of a Submission.	
1.1.4.1	The service shall provide the capability to store a DigitalComponent received in a Submission.	
	Process a PreservationManagementDocument that contains SubmissionInformationP	
1.2	The service shall provide the capability to process a PreservationManagementDocument that contains SubmissionInformation.	
1.2.1	The service shall provide the capability to extract SubmissionData from a PreservationManagementDocument that contains SubmissionInformation.	
1.2.2	The service shall provide the capability to associate a PreservationManagementDocument that contains SubmissionInformation with related objects.	
1.2.2.1	The service shall provide the capability to associate a PreservationManagementDocument that contains SubmissionInformation with a Submission	
1.2.2.1.1	The service shall provide the capability to associate a PreservationManagementDocument that contains SubmissionInformation with an IntellectualEntity contained in a Submission.	
1.2.2.1.1.1	The service shall provide the capability to associate a PreservationManagementDocument that contains SubmissionInformation with a PreservationTarget contained in a Submission.	
1.2.2.1.1.2	The service shall provide the capability to associate a PreservationManagementDocument that contains SubmissionInformation with HeuristicInformation contained in a Submission.	
1.2.2.1.2	The service shall provide the capability to associate a PreservationManagementDocument that contains SubmissionInformation with a BinaryEncoding contained in a Submission.	

	Process Submission	
Identifier	Requirement	Calls
1.2.2.1.3	The service shall provide the capability to associate a PreservationManagementDocument that contains SubmissionInformation with a DigitalComponent contained in a Submission.	
	Evaluate SubmissionData.	
1.3	The service shall provide the capability to evaluate SubmissionData.	
1.3.1	The service shall provide the capability to identify the basis for authorizing a Submission.	
1.3.1.1	The service shall provide the capability to determine if a Submission is authorized by a Preservation Agreement.	
1.3.1.2	The service shall provide the capability to determine if a Submission is authorized by a Preservation Service Contract.	
1.3.1.3	The service shall provide the capability to determine if a Submission is authorized by another PreservationManagementDocument.	
1.3.2	The service shall provide the capability to evaluate SubmissionData about the transfer of a Submission.	
1.3.2.1	The service shall provide the capability to determine if the schedule for transfer of a Submission is acceptable.	
1.3.2.2	The service shall provide the capability to determine if the means of transfer of a Submission is acceptable.	
1.3.3.3	The service shall provide the capability to determine if the Submitter is authorized to transmit the Submission.	
1.3.3	The service shall provide the capability to determine if the contents of a Submission are appropriate.	
1.3.3.1	The service shall provide the capability to determine if an IntellectualEntity identified in SubmissionData is authorized for Submission.	
1.3.3.1.1	The service shall provide the capability to determine if a PreservationTarget identified in SubmissionData is authorized for Submission.	
1.3.3.1.2	The service shall provide the capability to determine if HeuristicInformation identified in SubmissionData is authorized for Submission.	
1.3.3.1.3	The service shall provide the capability to determine if the set of IntellectualEntities identified in SubmissionData constitutes an authorized SubmissionSet.	
1.3.3.1.3.1	The service shall provide the capability to determine if the set of IntellectualEntities identified in SubmissionData includes all the IntellectualEntities that should be included in an authorized SubmissionSet.	
1.3.3.1.3.2	The service shall provide the capability to determine if the set of IntellectualEntities identified in SubmissionData includes any IntellectualEntity that should not be included in an authorized SubmissionSet.	
1.3.4	The service shall provide the capability to assess the quality of SubmissionData	11.1.3 and 11.2.
	Evaluate DigitalComponents described in SubmissionData	

	Process Submission	
Identifier	**Requirement**	**Calls**
1.4	The service shall provide the capability to determine if the set of DigitalComponents identified in SubmissionData as included in a SubmissionSet is appropriate.	
1.4.1	The service shall provide the capability to determine if each DigitalComponent identified in SubmissionData as included in a SubmissionSet is associated with an IntellectualEntity in the SubmissionSet.	
1.4.1.1	The service shall provide the capability to determine if each DigitalComponent identified in SubmissionData as included in a SubmissionSet is associated with a ComponentDescription.	
1.4.1.2	The service shall provide the capability to determine if each DigitalComponent identified in SubmissionData as included in a SubmissionSet is qualified as a specialization of DigitalComponent.	
1.4.1.2.1	The service shall provide the capability to determine if the qualification of a DigitalComponent identified in SubmissionData as a specialization of DigitalComponent is appropriate.	
1.4.1.2.1.1	The service shall provide the capability to determine if a DigitalComponent qualified as a ContentComponent in SubmissionData is appropriately qualified as such.	
1.4.1.2.1.2	The service shall provide the capability to determine if a DigitalComponent qualified as a SoftwareComponent in SubmissionData is appropriately qualified as such.	
1.4.1.2.1.2.1	The service shall provide the capability to determine if a SoftwareComponent is appropriate for processing a ContentComponent with which it is associated.	
1.4.1.2.1.3	The service shall provide the capability to determine if a DigitalComponent qualified as an InstantiationComponent in SubmissionData is appropriately qualified as such.	
1.4.1.2.1.3.1	The service shall provide the capability to determine if an InstantiationComponent is appropriate for application to a ContentComponent with which it is associated.	
1.4.1.2.1.3.2	The service shall provide the capability to determine if an InstantiationComponent can be processed by a SoftwareComponent with which it is associated.	
1.4.2	The service shall provide the capability to determine if each DigitalComponent identified in SubmissionData as included in a SubmissionSet is associated with a BinaryEncoding of an IntellectualEntity in the SubmissionSet.	
1.4.2.1	The service shall provide the capability to determine if the set of DigitalComponents in a BinaryEncoding is appropriate for instantiating the IntellectualEntity that the BinaryEncoding represents.	
1.4.2.2	The service shall provide the capability to determine if a BinaryEncoding includes at least one Manifestation of the IntellectualEntity that the BinaryEncoding represents.	
1.4.2.2.1	The service shall provide the capability to determine if the set of Manifestations in a BinaryEncoding is appropriate for instantiating the IntellectualEntity that the BinaryEncoding represents.	
1.4.2.2.2	The service shall provide the capability to determine if the set of Manifestations in a BinaryEncoding includes a RuntimeVersion.	

	Process Submission	
Identifier	**Requirement**	**Calls**
1.4.2.2.2.1	The service shall provide the capability to determine if a RuntimeVersion in a BinaryEncoding is appropriate for loading the IntellectualEntity that the BinaryEncoding represents on a computer.	
1.4.2.2.3	The service shall provide the capability to determine if the set of Manifestations in a BinaryEncoding of a HumanReadableIntellectualEntity includes at least one Rendering.	
1.4.2.2.3.1	The service shall provide the capability to determine if the set of Renderings in a BinaryEncoding of a HumanReadableIntellectualEntity is appropriate for instantiating that IntellectualEntity.	
	The service shall provide the capability to Compare related SubmissionData.	
1.5	The service shall provide the capability to compare related SubmissionData.	11.2.1 and 11.2.2
	Compare Data about a DigitalComponent with the DigitalComponent.	
1.6	The service shall provide the capability to compare SubmissionData about a DigitalComponent with the DigitalComponent itself.	11.1.3.5 and 12.3.1.2
	Determine the acceptability of a proposed Submission.	
1.7	The service shall provide the capability to determine the acceptability of a proposed Submission.	
1.7.1	The service shall provide the capability to approve a proposed Submission.	
1.7.2	The service shall provide the capability to disapprove a proposed Submission.	
1.7.2.1	The service shall provide the capability to reject a proposed Submission in its entirety	
1.7.2.2	The service shall provide the capability to reject a part of a Submission proposed	
1.7.3	The service shall provide the capability to require modification of a proposed Submission.	
	Determine the acceptability of a received Submission.	
1.8	The service shall provide the capability to determine the acceptability of a received Submission.	
1.8.1	The service shall provide the capability to accept a received Submission.	
1.8.2	The service shall provide the capability to reject a received Submission.	
1.8.2.1	The service shall provide the capability to reject a received Submission in its entirety	
	The service shall provide the capability to reject a part of a received Submission	
	The service shall provide the capability to require modification of a received Submission.	
	Capture SubmissionStatusData.	
1.9	The service shall provide the capability to capture SubmissionStatusData.	
1.9.1	The service shall provide the capability to capture SubmissionStatusData about a proposed Submission.	

	Process Submission	
Identifier	**Requirement**	**Calls**
1.9.2	The service shall provide the capability to capture SubmissionStatusData about a received Submission.	
	Review SubmissionInformation	
1.3.3	The service shall provide the capability to review SubmissionInformation.	
1.3.3.1	The service shall provide the capability to review the assignment of an object in SubmissionInformation.	
1.3.3.1.1	The service shall provide the capability to review the assignment of an object to a class in SubmissionInformation.	
1.3.3.1.2	The service shall provide the capability to review the designation of an object as an IntellectualEntity in SubmissionInformation.	
1.3.3.1.2.1	The service shall provide the capability to review the designation of an IntellectualEntity as a PreservationTarget in SubmissionInformation.	
1.3.3.1.3	The service shall provide the capability to review the designation of an object as a BinaryEncoding in SubmissionInformation.	
1.3.3.2	The service shall provide the capability to review the designation of a PermanentFeature in SubmissionInformation.	
1.3.3.3	The service shall provide the capability to review data about the composition of an object in SubmissionInformation.	
1.3.3.4	The service shall provide the capability to review the characterization of an object in SubmissionInformation.	
1.3.3.5	The service shall provide the capability to review data about a PermanentFeature in SubmissionInformation.	
1.3.3.6	The service shall provide the capability to review data about HeuristicInformation in SubmissionInformation.	
	Process PreservationManagementData Included With a Submission.	
1.3.4	The service shall provide the capability to process PreservationManagementData included with a Submission.	
1.3.4.1	The service shall provide the capability to extract PreservationManagementData from a PreservationManagementDocument included in SubmissionInformation.	
1.3.4.2.2	The service shall provide the capability to designate an IntellectualEntity as a PreservationTarget based on SubmissionInformation.	
1.3.4.2.3	The service shall provide the capability to designate an PreservationTarget as a PreservationItem based on SubmissionInformation.	
1.3.4.2.4	The service shall provide the capability to designate an PreservationTarget as a PreservationAggregate based on SubmissionInformation.	
1.3.4.2.5	The service shall provide the capability to designate an IntellectualEntity as HeuristicInformation based on SubmissionInformation.	
1.3.4.2.6	The service shall provide the capability to define an association between objects based on SubmissionInformation.	

	Process Submission	
Identifier	**Requirement**	**Calls**
1.3.4.2	The service shall provide the capability to process PreservationManagementData obtained from SubmissionInformation.	
1.3.4.2.1	The service shall provide the capability to create a data entry for an object based on SubmissionInformation.	
1.3.4.2.7	The service shall provide the capability to compare PreservationManagementData obtained from a Submission with related existing PreservationManagementInformation.	
1.3.4.2.7.1	The service shall provide the capability to identify any conflict between PreservationManagementData obtained from a Submission and existing PreservationManagementInformation as a problem.	
1.3.4.2.7.2	The service shall provide the capability to invoke the appropriate ProblemHandlingInstruction to address a conflict between PreservationManagementData obtained from a Submission and existing PreservationManagementInformation.	
1.3.4.2.7.3	The service shall provide the capability to process a ProblemResolutionResponse responding to a problem with a PreservationManagementData obtained from a Submission.	
	Inspect an object in a Submission	
1.3.5	The service shall provide the capability to inspect an object in a Submission.	
1.3.5.1	The service shall provide the capability to inspect a DigitalComponent in a Submission.	
1.3.5.2	The service shall provide the capability to inspect a ManifestationDescription in a Submission.	
1.3.5.4	The service shall provide the capability to verify a Feature of a IntellectualEntity included in a Submission.	
1.3.5.4.1	The service shall provide the capability to confirm the existence of a Feature of a IntellectualEntity.	
1.3.5.4.2	The service shall provide the capability to determine the value of a Feature of a IntellectualEntity.	
1.3.5.4.3	The service shall provide the capability to capture data about the expression of a Feature of a IntellectualEntity in a BinaryEncoding.	
1.3.5.4.4	The service shall provide the capability to verify the expression of a Feature of a IntellectualEntity in a DigitalComponent.	
	Address a problem with a received Submission	
1.3.6	The service shall provide the capability to address a problem with a received Submission.	
1.3.6.1	The service shall provide the capability to identify a problem with a received Submission.	
1.3.6.2	The service shall provide the capability to invoke the appropriate ProblemHandlingInstruction for a problem with a received Submission.	
1.3.6.3	The service shall provide the capability to process a ProblemResponseReport related to a problem with a received Submission.	
	Accept a received Submission	
1.3.7	The service shall provide the capability to accept a received Submission.	

	Process Submission	
Identifier	**Requirement**	**Calls**
1.3.7.1	The service shall provide the capability to accept an IntellectualEntity in a Submission.	
1.3.7.2	The service shall provide the capability to accept a BinaryEncoding in a Submission.	
1.3.7.3	The service shall provide the capability to accept a DigitalComponent in a Submission.	
1.3.7.4	The service shall provide the capability to accept PreservationManagementInformation received in a Submission.	
1.3.7.5	The service shall provide the capability to accept a received Submission provisionally.	
1.3.7.5.1	The service shall provide the capability to make acceptance of a Submission conditional on problem resolution.	
	Reject a Submission	
1.3.8	The service shall provide the capability to reject a Submission.	
1.3.8.1	The service shall provide the capability to reject a Submission in its entirety	
1.3.8.2	The service shall provide the capability to reject a part of a Submission	
1.3.8.2.1	The service shall provide the capability to reject an IntellectualEntity in a Submission	
1.3.8.2.2	The service shall provide the capability to reject a BinaryEncoding in a Submission	
1.3.8.2.3	The service shall provide the capability to reject a DigitalComponent in a Submission	
1.3.8.2.4	The service shall provide the capability to reject PreservationManagementInformation in a Submission	
	Send an accepted DigitalComponent to Preservation Storage.	
1.3.9	The service shall provide the capability to send an accepted DigitalComponent to Preservation Storage.	

10.1.2. Preservation Storage

	Manage Preservation Storage
Identifier	Requirement
2.	The service shall provide the capability to manage Preservation Storage
	Store a DigitalComponent
2.1	The service shall provide the capability to store a DigitalComponent,
2.1.1	The service shall provide the capability to store a DigitalComponent contained in a received Submission in a temporary storage location.
2.1.2	The service shall provide the capability to store a DigitalComponent contained in a Submission in a permanent storage location once it is accepted in Submission Processing.
2.1.3	The service shall provide the capability to store a DigitalComponent created in a PreservationChange in a permanent storage location.
	Ensure the integrity of a DigitalComponent in storage
2.2	The service shall provide the capability to ensure the integrity of a DigitalComponent in storage.
2.2.1	The service shall provide the capability to ensure that the bit string that constitutes any DigitalComponent remains unchanged while the object is in storage.
2.2.1.1	The service shall provide the capability to ensure that the bit string that constitutes any DigitalComponent remains unchanged when there is a change in a storage condition.
2.2.1.1.1	The service shall provide the capability to ensure that the bit string that constitutes any DigitalComponent remains unchanged when its storage location is changed.
2.2.1.1.2	The service shall provide the capability to ensure that the bit string that constitutes any DigitalComponent remains unchanged when there is a change in the physical medium on which it is stored.
2.2.1.1.3	The service shall provide the capability to ensure that the bit string that constitutes any DigitalComponent remains unchanged when there is a change in the device on which it is stored.
2.2.1.1.4	The service shall provide the capability to ensure that the bit string that constitutes any DigitalComponent remains unchanged when it is copied from one storage unit to another.
2.2.1.2	The service shall provide the capability to ensure the integrity of a DigitalComponent when there is a change in a storage management system.
2.2.2	The service shall provide the capability to verify the integrity of a stored DigitalComponent. (Calls verification)
2.2.2.1	The service shall provide the capability to ensure the persistence of a specified relationship between a stored DigitalComponent and another object.
2.2.2.2	The service shall provide the capability to verify that a stored DigitalComponent is consistent with PreservationManagementData about it.
2.2.2.1.1.1	The service shall provide the capability to ensure the persistence of a specified relationship between a stored DigitalComponent and another DigitalComponent in a BinaryEncoding.

	Manage Preservation Storage
Identifier	Requirement
2.2.2.1.1.2	The service shall provide the capability to ensure the persistence of a relationship between a stored DigitalComponent and a Manifestation in a BinaryEncoding specified in a ManifestationDescription.
2.2.2.1.1	The service shall provide the capability to ensure the persistence of a specified relationship between a stored DigitalComponent and a BinaryEncoding.
	Retrieve a DigitalComponent from storage
2.3	The service shall provide the capability to retrieve a DigitalComponent from storage.
2.3.1	The service shall provide the capability to retrieve from storage a DigitalComponent that is part of a BinaryEncoding.
2.3.2	The service shall provide the capability to retrieve from storage a DigitalComponent that is part of a ManagementSet.
2.3.3	The service shall provide the capability to retrieve from storage a DigitalComponent that is a member of a set of DigitalComponents.
2.3.4	The service shall provide the capability to retrieve from storage a DigitalComponent that is a member of a class of DigitalComponents.
	Manage the Retention of DigitalComponent sin Storage
2.4	The service shall provide the capability to delete a DigitalComponent from storage.
2.4.1	The service shall provide the capability to delete a DigitalComponent from storage in accordance with a PreservationRule.
2.4.2	The service shall provide the capability to delete a DigitalComponent from storage in response to a command from an authorized Actor.
2.4.3	The service shall provide the capability to verify the deletion of a DigitalComponent from storage.
2.4.4	The service shall provide the capability to prevent the deletion of a DigitalComponent from storage.
2.4.4.1	The service shall provide the capability to prevent the deletion of a DigitalComponent from storage when it is part of a BinaryEncoding that has not also been marked for deletion.
2.4.4.2	The service shall provide the capability to prevent the deletion of a DigitalComponent from storage in accordance with a PreservationRule.
	Manage a storage package
2.5	The service shall provide the capability to manage a storage package.
2.5.1	The service shall provide the capability to create a storage package
2.5.1.1	The service shall provide the capability to create a storage package as a physical unit.
2.5.1.2	The service shall provide the capability to create a storage package as a logical unit.
2.5.1.3	The service shall provide the capability to specify the contents of a storage package.

	Manage Preservation Storage
Identifier	**Requirement**
2.5.1.3.1	The service shall provide the capability to create a storage package that comprises one or more BinaryEncodings of an IntellectualEntity and PreservationManagementInformation about that entity.
2.5.1.3.2	The service shall provide the capability to create a storage package that comprises one or more BinaryEncodings of a PreservationTarget, related HeuristicInformation and PreservationManagementInformation about those objects.
2.5.1.3.3	The service shall provide the capability to create a storage package that comprises a MangementSet and PreservationManagementInformation about the set.
2.5.1.3.3.1	The service shall provide the capability to create a storage package that comprises a SubmissionSet.
2.5.1.3.3.2	The service shall provide the capability to create a storage package that comprises a PreservationActionSet.
2.5.1.3.3.3	The service shall provide the capability to create a storage package that comprises a PreservationNetwork.
2.5.1.4	The service shall provide the capability to nest a storage package in another storage package.
2.5.2	The service shall provide the capability to store a storage package
2.5.3	The service shall provide the capability to retrieve a storage package
2.5.4	The service shall provide the capability to modify a storage package
2.5.5	The service shall provide the capability to delete a storage package
2.5.5.1	The service shall provide the capability to retain an object in a storage package when the package is deleted.
2.5.6	The service shall provide the capability to identify an item in a storage package.
2.5.7	The service shall provide the capability to identify a storage package that contain a given item .

10.1.3. Preservation Change

	Execute Preservation Change
Identifier	**Requirement**
3.	The service shall provide the capability to execute a Preservation Change.
	Create New BinaryEncoding
3.1	The service shall provide the capability to create a new BinaryEncoding of a PreservationTarget.
3.1.1	The service shall provide the capability to create a new BinaryEncoding of a PreservationTarget by replacing a DigitalComponent that is part of an existing Encoding of the PreservationTarget.
3.1.1.1	The service shall provide the capability to create a new BinaryEncoding of a PreservationTarget using different methods of replacing a DigitalComponent.
3.1.1.1.1	The service shall provide the capability to create a new BinaryEncoding of a PreservationTarget by migration of a DigitalComponent that is part of the Encoding to a different format.
3.1.1.1.2	The service shall provide the capability to create a new BinaryEncoding of a PreservationTarget by normalization of a DigitalComponent that is part of the Encoding to a standard format.
3.1.1.1.3	The service shall provide the capability to create a new BinaryEncoding of a PreservationTarget by changing the set of DigitalComponents that constitute an encoding of that target.
3.1.1.1.3.1	The service shall provide the capability to create a new BinaryEncoding of a PreservationTarget by combining two or more DigitalComponents.
3.1.1.1.3.2	The service shall provide the capability to create a new BinaryEncoding of a PreservationTarget by replacing a SoftwareComponent with different software.
3.1.2	The service shall provide the capability to indicate the approval status of a method of creating a new BinaryEncoding.
3.1.2.1	The service shall provide the capability to prohibit the implementation of a method of creating a new BinaryEncoding if the method has not been approved.
3.1.2.2	The service shall provide the capability to provide the capability for a user authorized by the Preservation Director to approve the creation of new BinaryEncoding of a PreservationTarget on a case by case basis.
3.1.2.3	The service shall provide the capability to prohibit the implementation of a method of creating a new BinaryEncoding if the method would not express a PermanentFeature of a PreservationTarget properly.
3.1.2.3.1	The service shall provide the capability to prohibit the implementation of a method of creating a new BinaryEncoding if the method would not express a PermanentFeature of a PreservationItem properly.
3.1.2.3.2	The service shall provide the capability to prohibit the implementation of a method of creating a new BinaryEncoding of a PreservationTarget if the method would not express a PermanentFeature of a PreservationCollection.
3.1.2.3.3	The service shall provide the capability to prohibit the implementation of a method of creating a new BinaryEncoding of a PreservationTarget if the method would prevent instantiation of a PreservationTarget.
3.1.2.5	The service shall provide the capability to approve the implementation of a method of creating a new BinaryEncoding.
3.1.2.5.1	The service shall provide the capability to approve the implementation of a method of creating a new BinaryEncoding globally.

	Execute Preservation Change
Identifier	**Requirement**
3.1.2.5.2	The service shall provide the capability to approve the implementation of a method of creating a new BinaryEncoding conditionally.
3.1.2.6	The service shall provide the capability to provide the capability for an Actor authorized by the Preservation Director to approve the creation of new BinaryEncoding of a PreservationTarget on a case by case basis.
3.1.3	The service shall provide the capability to verify that a BinaryEncoding preserves a PreservationTarget.
3.1.3.1	The service shall provide the capability to verify that a BinaryEncoding of a PreservationTarget expresses every PermanentFeature of the PreservationTarget.
3.1.3.2	The service shall provide the capability to verify that a BinaryEncoding of a PreservationTarget can be instantiated.
3.1.3.2.1	The service shall provide the capability to verify that a BinaryEncoding supports instantiation of a PreservationTarget in a RuntimeVersion.
3.1.3.2.1	The service shall provide the capability to verify that a BinaryEncoding supports instantiation of a HumanReadablePreservationTarget in a Rendering.
3.1.3.2.2	The service shall provide the capability to verify that a Manifestation of a BinaryEncoding of a PreservationTarget preserves all ManifestPermanentFeatures of the PreservationTarget.
3.1.3.2	The service shall provide the capability to generate a problem report identifying any PermanentFeature that is not preserved when a BinaryEncoding of a PreservationTarget is instantiated.
3.1.4	The service shall provide the capability to assess whether a BinaryEncoding preserves a PreservationTarget.
3.1.5	The service shall provide the capability to determine if creation of a new BinaryEncoding of a PreservationTarget by the replacement of a DigitalComponent would impact the preservation of any other PreservationTarget.
3.1.5.1	The service shall provide the capability to identify all BinaryEncodings that contain a DigitalComponent that has been targeted for replacement.
3.1.5.1.1	The service shall provide the capability to assess whether an IntellectualEntity could be instantiated using a new BinaryEncoding created by replacing one or more DigitalComponents in an existing BinaryEncoding.
3.1.5.1.1.1	The service shall provide the capability to assess the impact of creating a new BinaryEncoding by replacing a DigitalComponent that is targeted for replacement on the Instantiation of a PreservationTarget.
3.1.5.1.1.2	The service shall provide the capability to assess the impact of creating a new BinaryEncoding by replacing a DigitalComponent that is targeted for replacement on the expression of a PermanentFeature of a PreservationTarget.
3.1.5.1.1.3	The service shall provide the capability to assess the impact of creating a new BinaryEncoding by replacing a DigitalComponent that is targeted for replacement on the the relationship of a PreservationTarget to another IntellectualEntity.
3.1.5.1.1.3.1	The service shall provide the capability to assess the impact of creating a new BinaryEncoding by replacing a DigitalComponent that is targeted for replacement on the the relationship of a PreservationTarget to another PreservationTarget.

	Execute Preservation Change
Identifier	**Requirement**
3.1.5.1.1.3.2	The service shall provide the capability to assess the impact of creating a new BinaryEncoding by replacing a DigitalComponent that is targeted for replacement on the the relationship of a PreservationTarget to HeuristicInformation.
	Create PreservationManagementData About a BinaryEncoding
3.2	The service shall provide the capability to create PreservationManagementData about a BinaryEncoding created in the Preservation Environment.
3.2.1	The service shall provide the capability to create PreservationActionData about the creation of a new BinaryEncoding.
3.2.1.1	The service shall provide the capability to create SubmissionProcessingData about a new BinaryEncoding when its creation involved Submission of an object.
3.2.1.2	The service shall provide the capability to create PreservationStorageData about a new BinaryEncoding.
3.2.1.3	The service shall provide the capability to create PreservationChangeData about a new BinaryEncoding.
3.2.1.4	The service shall provide the capability to create PreservationAssessmentData about a new BinaryEncoding.
3.2.1.5	The service shall provide the capability to create VerificationData about a new BinaryEncoding.
3.2.1.6	The service shall provide the capability to create AccessData about a new BinaryEncoding.
3.2.1.7	The service shall provide the capability to create ProblemHistoryData about a new BinaryEncoding.
3.2.2	The service shall provide the capability to create PreservationTargetData about a BinaryEncoding created in the Preservation Environment.
3.2.2.1	The service shall provide the capability to create TargetStateData about a BinaryEncoding created in the Preservation Environment.
3.2.2.1.1	The service shall provide the capability to create SubmissionStatus data about an object submitted in connection with the creation of a new BinaryEncoding in the Preservation Environment.
3.2.2.1.2	The service shall provide the capability to create StorageStatus data about a DigitalComponent that is part of a new BinaryEncoding.
3.2.2.1.3	The service shall provide the capability to create AssessmentStatus data about an Assessment of a new BinaryEncoding.
3.2.2.1.4	The service shall provide the capability to create PreservationDescription data about a new BinaryEncoding.
3.2.2.2	The service shall provide the capability to create TargetDescriptionData about a BinaryEncoding created in the Preservation Environment.
3.2.2.2.1	The service shall provide the capability to create a data entry describing a new BinaryEncoding.
3.2.2.2.1.1	The service shall provide the capability to associate a new BinaryEncoding with an IntellectualEntity.
3.2.2.2.1.2	The service shall provide the capability to associate a new BinaryEncoding with HeuristicInformation.
3.2.2.2.1.3	The service shall provide the capability to identify an earlier BinaryEncoding as the source from which a BinaryEncoding created in the Preservation Environment was derived.

	Execute Preservation Change
Identifier	Requirement
3.2.2.2.1.4	The service shall provide the capability to associate a new BinaryEncoding with a PermanentFeature of a PreservationTarget that the BinaryEncoding represents.
3.2.2.2.2	The service shall provide the capability to create a ComponentDescription entry describing a DigitalComponent that is part of a new BinaryEncoding.
3.2.2.2.3	The service shall provide the capability to create a ManifestationDescription describing a Manifestation that is part of a new BinaryEncoding.
3.2.2.2.3.1	The service shall provide the capability to create a ManifestationDescription describing a RuntimeVersion that is part of a new BinaryEncoding.
3.2.2.2.3.2	The service shall provide the capability to create a ManifestationDescription describing a Rendering that is part of a new BinaryEncoding.
3.2.2.2.2.1	The service shall provide the capability to associate a DigitalComponent that is created as part of a new BinaryEncoding with an earlier DigitalComponent from which it was derived.
3.2.3	The service shall provide the capability to create a data entry describing a PermanentFeatureExpression of a BinaryEncoding created in the Preservation Environment to a RuleDomain.
3.2.4	The service shall provide the capability to assign a BinaryEncoding created in the Preservation Environment to a RuleDomain.

10.1.4. Access

	Provide Access Services
Identifier	**Requirement**
4.	The service shall provide the capability to provide access services.
	Receive a Request for Access
4.1	The service shall provide the capability to receive a request for access.
	Receive Request for Access to Information
4.1.1	The service shall provide the capability to receive a request for access to information.
4.1.1.1	The service shall provide the capability to receive a request for access to an IntellectualEntity.
4.1.1.1.1	The service shall provide the capability to receive a request for access to a PreservationTarget.
4.1.1.1.2	The service shall provide the capability to receive a request for access to HeuristicInformation.
4.1.1.2	The service shall provide the capability to receive a request for access to PreservationManagementInformation.
4.1.1.3	The service shall provide the capability to receive a request to search for information.
4.1.1.4	The service shall provide the capability to receive a request to browse information.
4.1.1.5	The service shall provide the capability to receive a request for a copy of information.
4.1.1.5.1	The service shall provide the capability to receive a request for a copy of an IntellectualEntity.
4.1.1.5.2	The service shall provide the capability to receive a request for a copy of a DigitalComponent.
4.1.1.5.3	The service shall provide the capability to receive a request to receive a copy via instantiation online.
4.1.1.5.4	The service shall provide the capability to receive a request to receive a copy of via a file transfer mechanism.
	Receive Request to Access PreservationAction Capabilities
4.1.2	The service shall provide the capability to receive a request for access to PreservationAction capabilities.
4.1.2.1	The service shall provide the capability to receive a request for Submission Processing.
4.1.2.2	The service shall provide the capability to receive a request for Preservation Storage.
4.1.2.3	The service shall provide the capability to receive a request for Preservation Change.
4.1.2.3.1	The service shall provide the capability to receive a request for access to create a BinaryEncoding.
4.1.2.3.2	The service shall provide the capability to receive a request to create a new BinaryEncoding by migration from an existing BinaryEncoding.
4.1.2.3.3	The service shall provide the capability to receive a request to create a new BinaryEncoding by importing a file.
	Receive Request to Access Information Management Capabilities
4.1.3	The service shall provide the capability to receive a request to access Information Management capabilities.
4.1.3.1	The service shall provide the capability to receive a request to input a class definition.

	Provide Access Services
Identifier	**Requirement**
4.1.3.2	The service shall provide the capability to receive a request to input a set definition.
4.1.3.3	The service shall provide the capability to receive a request for access to input capabilities.
4.1.3.4	The service shall provide the capability to receive a request to input an IntellectualEntity.
4.1.3.4.1	The service shall provide the capability to receive a request to input a PreservationTarget.
4.1.3.4.2	The service shall provide the capability to receive a request to input HeuristicInformation.
4.1.3.5	The service shall provide the capability to receive a request to input a BinaryEncoding.
4.1.3.6	The service shall provide the capability to receive a request to input a PreservationManagementDocument.
4.1.3.7	The service shall provide the capability to receive a request to input PreservationManagementData.
4.1.3.8	The service shall provide the capability to receive a request to input a PreservationRule.
4.1.3.9	The service shall provide the capability to receive a request to Reporting capabilities.
	Receive Request to Access Preservation Management Capabilities
4.1.4	The service shall provide the capability to receive a request to access Preservation Management capabilities.
4.1.4.1	The service shall provide the capability to receive a request to inspect.
4.1.4.2	The service shall provide the capability to receive a request to evaluate preservation status.
4.1.4.3	The service shall provide the capability to receive a request to verify preservation.
	Determine Appropriate Response
4.2	The service shall provide the capability to determine the appropriate response to a request for access.
	Determine if the Requested Access Is Authorized
4.2.1	The service shall provide the capability to determine if the requested access is authorized.
4.2.1.1	The service shall provide the capability to determine if the Actor is authorized to received the requested access.
4.2.1.2	The service shall provide the capability to determine if requested information is authorized for release
4.2.1.3	The service shall provide the capability to deny an unauthorized request.
4.2.1.3.1	The service shall provide the capability to inform the Actor of the reason for denying an unauthorized request.
4.2.1.3.1.1	The service shall provide the capability to terminate processing of an access request when it is determines that the requested access is not authorized.
4.2.2.1	The service shall provide the capability to inform the Access Client whether it is feasible to provide the requested access.
	Determine if it is Possible to Provide the Requested Access
4.2.2	The service shall provide the capability to determine if it is possible to provide the requested access.
	Respond to an Authorized and Feasible Request.

	Provide Access Services
Identifier	**Requirement**
4.3	The service shall provide the capability to respond to an authorized and feasible request.
	Respond to a Request for Access to information.
4.3.1	The service shall provide the capability to respond to a request for access to information.
4.3.1.1	The service shall provide the capability to search for information.
4.3.1.2	The service shall provide the capability to browse information.
4.3.1.3	The service shall provide the capability to deliver a copy of information.
4.3.1.3.1	The service shall provide the capability to deliver a copy of an IntellectualEntity.
4.3.1.3.2	The service shall provide the capability to deliver a copy of a DigitalComponent.
4.3.1.3.3	The service shall provide the capability to provide the capability deliver a copy via instantiation online.
4.3.1.3.4	The service shall provide the capability to deliver a copy of via a file transfer mechanism.
	Respond to Request for PreservationAction Capabilities
4.3.2	The service shall provide the capability to respond to a request for PreservationAction capabilities.
4.3.2.1	The service shall provide the capability to respond to a request for Submission Processing.
4.3.2.2	The service shall provide the capability to respond to a request for Preservation Storage.
4.3.2.3	The service shall provide the capability to respond to a request for access to Preservation Change capabilities.
4.3.2.3.1	The service shall provide the capability to create a new BinaryEncoding.
	Respond to a Request for Access to Information Management capabilities.
4.3.3	The service shall provide the capability to respond to a request for access to Information Management capabilities.
4.3.3.1	The service shall provide the capability to input an IntellectualEntity.
4.3.3.2	The service shall provide the capability to input PreservationManagementInformation.
4.3.3.2.1	The service shall provide the capability to input a PreservationManagementDocument.
4.3.3.2.2	The service shall provide the capability to input PreservationManagementData.
4.3.3.2.3	The service shall provide the capability to input a PreservationRule.
4.3.3.3	The service shall provide the capability to input HeuristicInformation.
4.3.3.4	The service shall provide the capability to input a class definition.
4.3.3.5	The service shall provide the capability to input a set definition.
	Respond to request for Access to Preservation Management Capabilities
4.3.4	The service shall provide the capability to respond to a request for access to Preservation Management capabilities.
4.3.4.1	The service shall provide the capability to respond to a request for Inspection.

	Provide Access Services
Identifier	**Requirement**
4.3.4.2	The service shall provide the capability to respond to a request to evaluate preservation status.
4.3.4.3	The service shall provide the capability to respond to a request to verify preservation.
	Manage PreservationActionData about Access
4.4	The service shall provide the capability to Manage PreservationActionData about Access.
4.4.1	The service shall provide the capability to generate PreservationActionData about Access.
4.4.2	The service shall provide the capability to save PreservationActionData about Access.

10.2. Information Management Capabilities

10.2.1. Class Management

Class Management	
Identifier	
	Define a Class
5.1	The service shall provide the capability to define a class of objects.
	Implement a Class of Objects Defined in the Paast domain model.
5.1.1	The service shall provide the capability to implement a class of objects defined in the PaaST domain model.
5.1.1.1	The service shall provide the capability to implement the IntellectualEntity class as defined in the PaaST domain model.
5.1.1.2	The service shall provide the capability to implement the MachineReadableIntellectualEntity class as defined in the PaaST domain model.
5.1.1.3	The service shall provide the capability to implement the HumanReadableIntellectualEntity class as defined in the PaaST domain model.
5.1.1.4	The service shall provide the capability to implement the PreservationTarget class as defined in the PaaST domain model.
5.1.1.5	The service shall provide the capability to implement the PreservationItems class as defined in the PaaST domain model.
5.1.1.6	The service shall provide the capability to implement the PreservationCollection class as defined in the PaaST domain model.
5.1.1.7	The service shall provide the capability to implement the PermanentFeature class as defined in the PaaST domain model.
5.1.1.8	The service shall provide the capability to implement the PermanentFeatureExpression class as defined in the PaaST domain model.
5.1.1.9	The service shall provide the capability to implement the HeuristicInformation class as defined in the PaaST domain model.
5.1.1.10	The service shall provide the capability to implement the BinaryEncoding class as defined in the PaaST domain model.
5.1.1.11	The service shall provide the capability to implement the ComponentDescription class as defined in the PaaST domain model.
5.1.1.11.1	The service shall provide the capability to implement the ContentComponent class as defined in the PaaST domain model.
5.1.1.11.2	The service shall provide the capability to implement the SoftwareComponent class as defined in the PaaST domain model.
5.1.1.11.3	The service shall provide the capability to implement the InstantiationComponent class as defined in the PaaST domain model.
5.1.1.12	The service shall provide the capability to implement the PreservationTargetComponent class as defined in the PaaST domain model.

Class Management	
Identifier	
5.1.1.13	The service shall provide the capability to implement the Manifestation class as defined in the PaaST domain model.
5.1.1.13.1	The service shall provide the capability to implement the RuntimeVersion class as defined in the PaaST domain model.
5.1.1.13.2	The service shall provide the capability to implement the Rendering class as defined in the PaaST domain model.
5.1.1.14	The service shall provide the capability to implement the ManifestationDescription class as defined in the PaaST domain model.
5.1.1.14.1	The service shall provide the capability to implement the RuntimeVersionDescription class as defined in the PaaST domain model.
5.1.1.14.2	The service shall provide the capability to implement the RenderingDescription class as defined in the PaaST domain model.
5.1.1.15	The service shall provide the capability to implement the PreservationAction class as defined in the PaaST domain model.
5.1.1.16	The service shall provide the capability to implement the PreservationManagementInformation class as defined in the PaaST domain model.
5.1.1.16.1	The service shall provide the capability to implement the PreservationManagementDocument class as defined in the PaaST domain model.
5.1.1.16.2	The service shall provide the capability to implement the PreservationManagementData class as defined in the PaaST domain model.
5.1.1.16.3.1	The service shall provide the capability to implement the Actor class as defined in the PaaST domain model.
5.1.1.16.3.1.1	The service shall provide the capability to implement the AuthorizedRole class as defined in the PaaST domain model.
5.1.1.16.3.1.2	The service shall provide the capability to implement the ActualRole class as defined in the PaaST domain model.
5.1.1.16.3.1.3	The service shall provide the capability to implement the PreservationRole enumeration as defined in the PaaST domain model.
5.1.1.16.2.1.3.1	The service shall provide the capability to allow an authorized Actor to add a role to the PreservationRole enumeration.
5.1.1.16.4	The service shall provide the capability to implement the PreservationTargetData class as defined in the PaaST domain model.
5.1.1.16.4.1	The service shall provide the capability to implement the TargetDescription class as defined in the PaaST domain model.
5.1.1.16.4.2	The service shall provide the capability to implement the TargetState class as defined in the PaaST domain model.
5.1.1.16.4.3	The service shall provide the capability to implement the PreservationActionData class as defined in the PaaST domain model.

Class Management	
Identifier	
5.1.1.17	The service shall provide the capability to implement the PreservationRule class as defined in the PaaST domain model.
5.1.1.17.1	The service shall provide the capability to implement the RuleDomain class as defined in the PaaST domain model.
5.1.1.17.1.1	The service shall provide the capability to implement the RuledObject class as defined in the PaaST domain model.
5.1.1.17.1.2	The service shall provide the capability to implement the RuledAction class as defined in the PaaST domain model.
5.1.1.17.1.3	The service shall provide the capability to implement the ControllingObject class as defined in the PaaST domain model.
5.1.1.17.1.4	The service shall provide the capability to implement the RuleTask class as defined in the PaaST domain model.
5.1.1.17.1.5	The service shall provide the capability to implement the RuleResult class as defined in the PaaST domain model.
5.1.1.17.1.6	The service shall provide the capability to implement the RuleEvaluator class as defined in the PaaST domain model.
5.1.1.17.2	The service shall provide the capability to implement the SuccessCriterion class as defined in the PaaST domain model.
5.1.1.17.3	The service shall provide the capability to implement the SuccessCondition class as defined in the PaaST domain model.
	Provide the Capability for an Actor to Define a Class of Objects
5.2	The service shall provide the capability to enable an Actor to define a class of objects.
5.2.1	The service shall provide the capability to enable an Actor to define a specialization of a class of objects defined in the PaaST domain model.
5.2.1.1	The service shall provide the capability to define a subclass of IntellectualEntity.
5.2.1.1.1	The service shall provide the capability to define a subclass of MachineReadableIntellectualEntity.
5.2.1.1.2	The service shall provide the capability to define a subclass of HumanReadableIntellectualEntity.
5.2.1.2.3	The service shall provide the capability to define a subclass of Item.
5.2.1.2.4	The service shall provide the capability to define a subclass of Collection.
5.2.1.2	The service shall provide the capability to define a subclass of PreservationTarget.
5.2.1.2.1	The service shall provide the capability to define a subclass of PreservationItem.
5.2.1.2.2	The service shall provide the capability to define a subclass of PreservationCollection.
5.2.1.3	The service shall provide the capability to define a subclass of BinaryEncoding.
5.2.1.4	The service shall provide the capability to define a subclass of DigitalComponent.
5.2.1.4.1	The service shall provide the capability to define a subclass of ContentComponent.
5.2.1.4.2	The service shall provide the capability to define a subclass of SoftwareComponent.

Class Management	
Identifier	
5.2.1.4.3	The service shall provide the capability to define a subclass of InstantiationComponent.
5.2.1.5	The service shall provide the capability to define a subclass of ComponentDescription.
5.2.1.5.1	The service shall provide the capability to define a subclass of ContentComponentDescription
5.2.1.5.2	The service shall provide the capability to define a subclass of InstantiationComponentDescription
5.2.1.5.3	The service shall provide the capability to define a subclass of SoftwareComponentDescription
5.2.1.6	The service shall provide the capability to define a subclass of Manifestation.
5.2.1.6.1	The service shall provide the capability to define a subclass of RuntimeVersion.
5.2.1.6.2	The service shall provide the capability to define a subclass of Rendering.
5.2.1.7	The service shall provide the capability to define a subclass of PermanentFeature.
5.2.1.8	The service shall provide the capability to define a subclass of PermanentFeatureExpression.
5.2.1.9	The service shall provide the capability to define a subclass of HeuristicInformation.
5.2.1.10	The service shall provide the capability to define a subclass of PreservationRule.
5.2.1.11	The service shall provide the capability to define a subclass of SuccessCriterion
5.2.1.12	The service shall provide the capability to define a subclass of PreservationAction.
5.2.1.13	The service shall provide the capability to define a subclass of PreservationManagementDocument.
5.2.1.14	The service shall provide the capability to define a subclass of PreservationManagementData.
5.2.2	The service shall provide the capability to to define a class of objects that is not a specialization of any class defined in the PaaST domain model.
	Define associations between classes
5.3	The service shall provide the capability to enable an Actor to define an association between two user defined classes.
5.3.1	The service shall provide the capability to define a generalization association between two user defined classes.
5.3.2	The service shall provide the capability to define an aggregation association between two user defined classes.
5.3.3	The service shall provide the capability to define a composition association between two user defined classes.
	Implement a Data Profile for an Attributable Class
5.4	The service shall provide the capability to define a Data Profile for an Attributable Class
5.4.1	The service shall provide the capability to define a Data Profile for IntellectualEntity.
5.4.2	The service shall provide the capability to define a Data Profile for Item.
5.4.2.1	The service shall provide the capability to define a Data Profile for Record.
5.4.3	The service shall provide the capability to define a Data Profile for Collection.

	Class Management
Identifier	
5.4.3.1	The service shall provide the capability to define a Data Profile for RecordAggregate.
5.4.4	The service shall provide the capability to define a Data Profile for BinaryEncoding.
5.4.5	The service shall provide the capability to define a Data Profile for ComponentDescription.
5.4.6	The service shall provide the capability to define a Data Profile for PreservationTarget.
5.4.7	The service shall provide the capability to define a Data Profile for PreservationItem.
5.4.8	The service shall provide the capability to define a Data Profile for PreservationCollection.
5.4.9	The service shall provide the capability to define a Data Profile for PermanentFeature.
5.4.10	The service shall provide the capability to define a Data Profile for PermanentFeatureExpression.
5.4.11	The service shall provide the capability to define a Data Profile for ArchivalLink.
5.4.12	The service shall provide the capability to define a Data Profile for AreaOfActivity.
5.4.13	The service shall provide the capability to implement a Data Profile for an Attributable Class
	97

10.2.2. Set Management

	Set Management
Identifier	**Requirement**
6.	The service shall provide the capability to manage a set of objects
	Define a Set
6.1	The service shall provide the capability to define a set of objects.
6.1.1	The service shall provide the capability to define a set of objects whose members include instances of different classes.
6.1.2	The service shall provide the capability to define a Feature of a set of objects.
6.1.2.1	The service shall provide the capability to define an attribute of a set of objects.
6.1.2.1.1	The service shall provide the capability to define an attribute of a set of objects that is proper to the set as such.
6.1.2.1.2	The service shall provide the capability to define an attribute of a set of objects that is a Feature of all members of the set.
6.1.2.1.2.1	The service shall provide the capability to define a criterion for membership in a set of objects.
6.1.2.1.2.2	The service shall provide the capability to define an attribute of a set of objects that specifies the permitted values of a Feature that is common to all members of the set.
6.1.2.1.2.2.1	The service shall provide the capability to define an attribute of a set of objects that specifies the permitted values of a Feature that is common to all members of the set that are instances of a specified class.
6.1.2.1.3	The service shall provide the capability to define a count of the members of a set as an attribute of a set of objects.
6.1.2.1.4	The service shall provide the capability to define a count of the classes that have one or more instances that are members of a set of objects as an attribute of that set.
6.1.2.1.5	The service shall provide the capability to define a count of the members of a set that are instances of a specific class as an attribute of a set of objects.
6.1.2.1.6	The service shall provide the capability to specify a count of members of a set as a criterion for determining completeness of the set.
6.1.2.2	The service shall provide the capability to define a operation of a set of objects.
6.1.3	The service shall provide the capability to delete the definition of a set of objects.
6.1.3.1	The service shall provide the capability to prevent the deletion of the definition of a set of objects if there is any object identified as a member of the set.
	Define the Membership of a Set
6.2	The service shall provide the capability to assign an object as a member of a set of objects.
6.2.1	The service shall provide the capability to assign an object as a member of a set of objects ad hoc.
6.2.2	The service shall provide the capability to assign an object as a member of a set of objects automatically when it satisfies the criteria for membership in the set.

	Set Management
Identifier	**Requirement**
6.2.3	The service shall provide the capability to specify the existence in the Preservation Environment of an object designated as a member of a set as a criterion for determining whether the set is complete.
	Specify an Association Between Members of a Set
6.3	The service shall provide the capability to specify an association between members of a set of objects.
6.3.1	The service shall provide the capability to provide the capability for members of a set of objects that are instances of different classes to inherit an association specified between the classes.
6.3.2	The service shall provide the capability to establish ad hoc an association between members of a set of objects.
6.3.3	The service shall provide the capability to define an order among members of a set of objects.
6.3.3.1	The service shall provide the capability to define a hierarchical order among members of a set of objects.
	Instantiate a Set
6.4	The service shall provide the capability to instantiate a set of objects.
6.4.1	The service shall provide the capability to assign an object that satisfies the criteria for membership in a set of objects to that set when the set is instantiated.
6.4.2	The service shall provide the capability to establish an association between two members of a set when those objects are also members of classes that are associated upon instantiation of the set.
6.4.3	The service shall provide the capability to instantiate an order of a set of objects.
6.4.3.1	The service shall provide the capability to assign a member of a set of objects to a position in the order of the set when the order is instantiated.
6.4.2	The service shall provide the capability to delete a set of objects.
6.4.2	The service shall provide the capability to delete any data associating a class with a set of objects when the set is deleted.
6.4.2	The service shall provide the capability to delete any data associating an object with a set of objects when the set is deleted.
	Define a Preservation Agregate
6.5	The service shall provide the capability to define a PreservationCollection.
6.5.1	The service shall provide the capability to specify the provenance of a PreservationCollection.
6.5.2	The service shall provide the capability to specify the date range of a PreservationCollection as an attribute of the collection.
6.5.3	The service shall provide the capability to specify an order of a PreservationCollection as its original order.
6.5.4	The service shall provide the capability to limit membership in a PreservationCollection.
6.5.4.1	The service shall provide the capability to limit membership in a PreservationCollection to PreservationItems that have the same provenance as the collection.
6.5.4.2	The service shall provide the capability to limit membership in a PreservationCollection to PreservationItems that have a Feature whose value is within a set of discrete values.

	Set Management
Identifier	**Requirement**
6.5.4.3	The service shall provide the capability to limit membership in a PreservationCollection to PreservationItems that have a Feature within a range of values.
6.5.5	The service shall provide the capability to designate a PreservationItem as a member of a PreservationCollection.
6.5.6	The service shall provide the capability to designate a PreservationCollection as a member of another PreservationCollection.
6.5.6.1	The service shall provide the capability to specify an association between PreservationCollections that are members of the same PreservationCollection.
6.5.7	The service shall provide the capability to require that a date of a member of a PreservationCollection be within the date range of the PreservationCollection.
	Define a ManagementSet
6.6	The service shall provide the capability to define a ManagementSet.
6.6.1	The service shall provide the capability to define the basis for membership of an object in a ManagementSet.
6.6.1.1	The service shall provide the capability to identify an object as a member of a ManagementSet.
6.6.1.2	The service shall provide the capability to identify an object as a member of a ManagementSet ad hoc.
6.6.1.3	The service shall provide the capability to define a general criterion as the basis for membership of an object in a ManagementSet.
6.6.1.3.1	The service shall provide the capability to define provenance as the basis for membership of an object in a ManagementSet.
6.6.1.3.2	The service shall provide the capability to define a Feature as the basis for membership of an object in a ManagementSet.
6.6.1.3.3	The service shall provide the capability to define the value of a Feature as the basis for membership of an object in a ManagementSet.
6.6.1.3.4	The service shall provide the capability to define an association with another object that is a member of the set as the basis for membership of an object in a ManagementSet.
6.6.1.3.5	The service shall provide the capability to provide the capability for an Actor to define a criterion as the basis for membership of an object in a ManagementSet.
6.6.1.3.6	The service shall provide the capability to formulate a PreservationRule that identifies an object as a member of a ManagementSet.
6.6.1.4	The service shall provide the capability to define a ManagementSet as a subset of another ManagementSet.
	Define a PreservationActionSet
6.6.2	The service shall provide the capability to define a PreservationActionSet.
6.6.2.1	The service shall provide the capability to define the scope of a PreservationActionSet by specifying a PreservationAction.
6.6.2.2	The service shall provide the capability to formulate a PreservationRule that defines a PreservationActionSet.

	Set Management
Identifier	**Requirement**
6.6.2.2.1	The service shall provide the capability to formulate a PreservationRule defining a PreservationActionSet that specifies a PreservationAction as a ControlledAction in the RuleDomain.
6.6.2.2.1	The service shall provide the capability to formulate a PreservationRule defining a PreservationActionSet that identifies the objects that are members of the set as ControlledTargets in the RuleDomain.
	Define a SubmissionSet
6.6.3	The service shall provide the capability to define a SubmissionSet.
6.6.3.1	The service shall provide the capability to define the composition of SubmissionSet.
6.6.3.1.1	The service shall provide the capability to define the composition of a SubmissionSet by identifying the objects that should be included in it.
6.6.3.1.1.1	The service shall provide the capability to define the composition of a SubmissionSet by identifying the PreservationTargets that should be included in it.
6.6.3.1.1.2	The service shall provide the capability to define the composition of a SubmissionSet by identifying the DitigalComponents that should be included in it.
6.6.3.1.1.3	The service shall provide the capability to define the composition of a SubmissionSet by identifying the HeuristicInformation that should be included in it.
6.6.3.1.2	The service shall provide the capability to define the composition of SubmissionSet ac hoc.
6.6.3.1.3	The service shall provide the capability to define the composition of SubmissionSet in accordance with PreservationManagementData extracted from a PreservationManagementDocument.
6.6.3.1.4	The service shall provide the capability to define the composition of SubmissionSet using a general criterion.
6.6.3.1.5	The service shall provide the capability to define the relationship between a SubmissionSet and an IntellectualEntity.
6.6.3.1.5.1	The service shall provide the capability to define the relationship between a SubmissionSet and an IntellectualEntity contained in the SubmissionSet.
6.6.3.1.5.2	The service shall provide the capability to define the relationship between a SubmissionSet and an IntellectualEntity previously accepted into the Preservation Environment.
6.6.3.1.5.3	The service shall provide the capability to define the relationship between a SubmissionSet and a PreservationTarget.
6.6.3.1.5.3.1	The service shall provide the capability to define the relationship between a SubmissionSet and a PreservationItem.
6.6.3.1.5.3.2	The service shall provide the capability to define the relationship between a SubmissionSet and a PreservationCollection.
6.6.3.2	The service shall provide the capability to specify the SubmissionInformation that should accompany a Submission
6.6.3.3	The service shall provide the capability to identify a Submitter authorized to submit a SubmissionSet.
6.6.3.4	The service shall provide the capability to specify conditions for the transfer of a SubmissionSet.
6.6.3.4.1	The service shall provide the capability to require that a NoticeOfIntentToTransfer be submitted prior the transfer of a SubmissionSet.

	Set Management
Identifier	**Requirement**
6.6.3.4.1.1	The service shall provide the capability to require that a NoticeOfIntentToTransfer be approved prior the transfer of a SubmissionSet.
6.6.3.4.2	The service shall provide the capability to specify an acceptable vehicle for the transfer of a SubmissionSet.
6.6.3.4.3	The service shall provide the capability to specify packaging for the transfer of a SubmissionSet.
6.6.3.4.4	The service shall provide the capability to specify the timing of the transfer of a SubmissionSet.
6.6.3.5.5	The service shall provide the capability to provide the capability for an Actor to specify a condition for the transfer of a SubmissionSet.
6.6.3.5	The service shall provide the capability to specify conditions for the acceptance of a SubmissionSet.
6.6.3.5.1	The service shall provide the capability to require that the Submission satisfies all transfer conditions for acceptance of the SubmissionSet.
6.6.3.5.2	The service shall provide the capability to require that any problem identified in review of a NoticeOfIntentToTransfer be resolved satisfactorily for acceptance of the SubmissionSet.
6.6.3.5.3	The service shall provide the capability to require that the objects contained in a SubmissionSet match the definition of the set's membership as a condition for acceptance of the SubmissionSet.
6.6.3.5.3.1	The service shall provide the capability to require that no PreservationTargets that should be in a SubmissionSet are included as a condition for acceptance of the SubmissionSet.
6.6.3.5.3.1.1	The service shall provide the capability to require that all PreservationTargets that should not be in a SubmissionSet are included as a condition for acceptance of the SubmissionSet.
6.6.3.5.3.2	The service shall provide the capability to require that all HeuristicInformation that should be in a SubmissionSet is included as a condition for acceptance of the SubmissionSet.
6.6.3.5.3.2.1	The service shall provide the capability to require that no HeuristicInformation that should not be in a SubmissionSet is included as a condition for acceptance of the SubmissionSet.
6.6.3.5.3.3	The service shall provide the capability to require that SubmissionInformation accompanying a Submission be satisfactory for the SubmissionSet to be accepted.
6.6.3.5.3.4	The service shall provide the capability to provide the capability for an Actor to specify a condition for the acceptance of a SubmissionSet.
6.6.3.5.3.5	The service shall provide the capability to specify a condition for the acceptance of a SubmissionSet in a PreservationRule.
	Define a PreservationNetwork
6.6.4	The service shall provide the capability to define a PreservationNetwork.
6.6.4.1	The service shall provide the capability to identify a PreservationTarget as the focus of a PreservationNetwork
6.6.4.1.1	The service shall provide the capability to identify all BinaryEncodings of a PreservationTarget as members of a PreservationNetwork that has the PreservationTarget as its focus.
6.6.4.1.1.1	The service shall provide the capability to identify all DigitalComponents contained in the BinaryEncodings of a PreservationTarget as members of a PreservationNetwork that has the PreservationTarget as its focus.

	Set Management
Identifier	**Requirement**
6.6.4.1.1.2	The service shall provide the capability to identify all Manifestations of a PreservationTarget as members of a PreservationNetwork that has the PreservationTarget as its focus.
6.6.4.1.1.2.1	The service shall provide the capability to identify all RuntimeVersions of a PreservationTarget as members of a PreservationNetwork that has the PreservationTarget as its focus.
6.6.4.1.1.2.2	The service shall provide the capability to identify all Renderings of a PreservationTarget as members of a PreservationNetwork that has the PreservationTarget as its focus.
6.6.4.1.2	The service shall provide the capability to identify all ComponentDescriptions contained in the BinaryEncodings of a PreservationTarget as members of a PreservationNetwork that has the PreservationTarget as its focus.
6.6.4.1.3	The service shall provide the capability to identify all instances of HeuristicInformation that clarify a PreservationTarget as members of a PreservationNetwork that has the PreservationTarget as its focus.
6.6.4.2	The service shall provide the capability to identify all PreservationManagementInformation related to the members of a PreservationNetwork as part of that network.
6.6.4.3	The service shall provide the capability to identify a PreservationItem that is part of a PreservationCollection that is the focus of a PreservationNetwork as a member of the PreservationNetwork of the PreservationCollection.
6.6.4.4	The service shall provide the capability to identify a PreservationTarget associated with another PreservationTarget that is the focus of a PreservationNetwork as a member of the PreservationNetwork of the focus target.
6.6.4.5	The service shall provide the capability to identify a PreservationCollection in which a PreservationItem that is the focus of a PreservationNetwork belongs as a member of the PreservationNetwork of the focus item.

10.2.3. Document Management

	Manage PreservationManagementDocuments
Identifier	**Requirement**
7.	The service shall provide the capability to manage a PreservationManagementDocument
	Manage a PreservationManagementDocument Defined in the PaaST Data Model
7.1	The service shall provide the capability to manage a PreservationManagementDocument defined in the PaaST data model
7.1.1	The service shall provide the capability to manage a Preservation Agreement.
7.1.2	The service shall provide the capability to manage a Preservation Service Contract.
7.1.3	The service shall provide the capability to manage a NoticeOfIntentToSubmit.
7.1.4	The service shall provide the capability to manage a SubmissionDocument.
7.1.5	The service shall provide the capability to manage a PreservationActionReport.
7.1.6	The service shall provide the capability to manage a ProblemReport.
7.1.7	The service shall provide the capability to manage a ProblemResolutionReport.
7.1.8	The service shall provide the capability to manage a PreservationAssessmentReport.
	Define a PreservationManagementDocument Specialization.
7.2	The service shall provide the capability to define a PreservationManagementDocument specialization.
7.2.1	The service shall provide the capability to define the form of a PreservationManagementDocument specialization.
7.2.2	The service shall provide the capability to define the content of a PreservationManagementDocument specialization.
7.2.2.1	The service shall provide the capability to specify a selection of PreservationManagementData as contents of a PreservationManagementDocument specialization.
	Generate a PreservationManagementDocument
7.3	The service shall provide the capability to generate a PreservationManagementDocument
7.3.1	The service shall provide the capability to populate a PreservationManagementDocument with PreservationManagementData specified as its content.
	Send a PreservationManagementDocument.
7.4	The service shall provide the capability to send a PreservationManagementDocument.
7.4.1	The service shall provide the capability to send a PreservationManagementDocument to an Actor.
	Receive a PreservationManagementDocument
7.5	The service shall provide the capability to receive a PreservationManagementDocument.
7.5.1	The service shall provide the capability to store a PreservationManagementDocument.
	Generate PreservationManagementData From a PreservationManagementDocument

	Manage PreservationManagementDocuments
Identifier	**Requirement**
7.6	The service shall provide the capability to generate PreservationManagementData from a PreservationManagementDocument.
7.6.1	The service shall provide the capability to extract data from a PreservationManagementDocument.
7.6.1.1	The service shall provide the capability to extract semantically tagged data from a PreservationManagementDocument.
7.6.1.2	The service shall provide the capability to extract fielded data from a PreservationManagementDocument.
7.6.2	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to a category of PreservationManagementData.
7.6.2.1	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to PreservationTargetData.
7.6.2.1.1	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to TargetDescription.
7.6.2.1.1.1	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to IntellectualEntity.
7.6.2.1.1.2.1	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to PreservationTarget.
7.6.2.1.1.2.2	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to HeuristicInformation.
7.6.2.1.1.3.3	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to BinaryEncoding.
7.6.2.1.1.4	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to DigitalComponent.
7.6.2.1.1.4.1	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to ComponentDescription.
7.6.2.1.1.5	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to PermanentFeature.
7.6.2.1.1.5.1	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to PermanentFeatureExpression.
7.6.2.1.1.6	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to Instantiation.
7.6.2.1.1.6.1	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to RuntimeVersion.
7.6.2.1.1.6.2	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to Rendering.
7.6.2.2.1.7	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to DigitalTypeRegistry.
7.6.2.1.2	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to TargetState.

	Manage PreservationManagementDocuments
Identifier	Requirement
7.6.2.1.2.1	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to SubmissionStatus.
7.6.2.1.2.2	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to StorageStatus.
7.6.2.1.2.3	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to AssessmentStatus.
7.6.2.1.2.4	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to VerificationStatus.
7.6.2.1.2.5	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to PreservationDescription.
7.6.2.3	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to PreservationActionData.
7.6.2.3.1	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to SubmissionProcessingData.
7.6.2.3.2	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to PreservationStorageData.
7.6.2.3.3	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to PreservationChangeData.
7.6.2.3.4	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to AccessData.
7.6.2.3.5	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to AssessmentData.
7.6.2.3.6	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to VerificationData.
7.6.2.3.7	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to ProblemHistoryData.
7.6.2.4	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to PreservationRule.
7.6.2.4.1	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to RuleDomain.
7.6.2.4.2	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to SuccessCriterion.
7.6.2.4.3	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to PreservationCondition.
7.6.2.4.4	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to ProblemHandlingInstruction.
7.6.2.5	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to Actor.

	Manage PreservationManagementDocuments
Identifier	**Requirement**
7.6.2.5.1	The service shall provide the capability to determine when data extracted from a PreservationManagementDocument maps to Participation.
7.7	The service shall provide the capability to determine when categorized data extracted from a PreservationManagementDocument relates to an object in existing PreservationManagementData.
7.8	The service shall provide the capability to save data extracted from a PreservationManagementDocument in the appropriate category of PreservationManagementData when the category in which it belongs has been determined.
7.8.1	The service shall provide the capability to associate saved data extracted from a PreservationManagementDocument with the existing object when it has been determined that the data relate to an object in existing PreservationManagementData.
7.8.2	The service shall provide the capability to associate saved data extracted from a PreservationManagementDocument with a new object in the appropriate class when it has been determined that the data do no relate to an object in existing PreservationManagementData.
7.9	The service shall provide the capability to generate a report about data extracted from a PreservationManagementDocument.
7.9.1	The service shall provide the capability to describe in a report the categorization of data extracted from a PreservationManagementDocument.
7.9.1.1	The service shall provide the capability to describe in a report any problem in relating data extracted from a PreservationManagementDocument to a category of PreservationManagementData.
7.9.2	The service shall provide the capability to describe in a report the association of categorized data extracted from a PreservationManagementDocument with objects in existing PreservationManagementData.
7.9.2.1	The service shall provide the capability to describe in a report any new object created when categorized data extracted from a PreservationManagementDocument could not be associationwith objects in existing PreservationManagementData.
7.9.2.1	The service shall provide the capability to describe in a report any problem in determining whether data extracted from a PreservationManagementDocument relates to an existing object or not.

10.2.4. Data Management

	Manage PreservationManagementData
Identifier	**Requirement**
8.	The service shall provide the capability to manage PreservationManagementData.
	Manage Data Entry
8.1	The service shall provide the capability to manage data entry of PreservationManagementData.
8.1.1	The service shall provide the capability to create an instance of PreservationManagementData.
8.1.1.2	The service shall provide the capability to assign a persistent and unique identifier to an instance of PreservationManagementData.
8.1.2	The service shall provide the capability to update an instance of PreservationManagementData.
8.1.3	The service shall provide the capability to delete an instance of PreservationManagementData.
8.1.4	The service shall provide the capability to define a precondition that triggers a change in PreservationManagementData.
8.1.4.1	The service shall provide the capability to initiate a change in PreservationManagementData when a trigger condition is satisfied.
8.1.5	The service shall provide the capability to specify the date of PreservationManagementData was changed.
8.1.5.1	The service shall provide the capability to specify the date of PreservationManagementData was created.
8.1.5.2	The service shall provide the capability to specify the date when PreservationManagementData was updated.
8.1.5.3	The service shall provide the capability to specify the date when PreservationManagementData was deleted.
8.1.6	The service shall provide the capability to specify the source of PreservationManagementData.
8.1.6.1	The service shall provide the capability to specify the type of DataSource used to produce PreservationManagementData.
8.1.6.2	The service shall provide the capability to identify an individual DataSource used to produce PreservationManagementData.
8.1.7	The service shall provide the capability to specify the subclass of PreservationManagementData in which an instance of PreservationManagementData belongs.
8.1.8	The service shall provide the capability to specify the class of object that PreservationManagementData is about.
8.1.8.1	The service shall provide the capability to identify an individual object that PreservationManagementData is about.
8.1.8.1	The service shall provide the capability to assign an instance of PreservationManagementData to a set of objects.
8.1.9	The service shall provide the capability to manage an enumeration related to PreservationManagementData

	Manage PreservationManagementData
Identifier	**Requirement**
8.1.9.1	The service shall provide the capability to create an enumeration related to PreservationManagementData
8.1.9.1.1	The service shall provide the capability to create the enumeration, ClarificationType
8.1.9.1.2	The service shall provide the capability to create the enumeration, DataSourceType
8.1.9.1.3	The service shall provide the capability to create the enumeration, DataType
8.1.9.1.4	The service shall provide the capability to create the enumeration, RenderingType
8.1.9.1.5	The service shall provide the capability to create the enumeration, MediaType
8.1.9.1.6	The service shall provide the capability to create the enumeration, PreservationRole
8.1.9.1.7	The service shall provide the capability to create a user-defined enumeration,
8.1.9.2	The service shall provide the capability to add an entry to an enumeration
8.1.9.2.1	The service shall provide the capability to add an entry to the enumeration, ClarificationType
8.1.9.2.2	The service shall provide the capability to add an entry to the enumeration, DataSourceType
8.1.9.2.3	The service shall provide the capability to add an entry to the enumeration, DataType
8.1.9.2.4	The service shall provide the capability to add an entry to the enumeration, RenderingType
8.1.9.2.5	The service shall provide the capability to add an entry to the enumeration, MediaType
8.1.9.2.6	The service shall provide the capability to add an entry to the enumeration, PreservationRole
8.1.9.2.7	The service shall provide the capability to add an entry to a user-defined enumeration,
8.1.9.2	The service shall provide the capability to modify an entry in an enumeration
8.1.9.4	The service shall provide the capability to delete an entry from an enumeration
	Manage PreservationTargetData
8.2	The service shall provide the capability to manage PreservationTargetData.
	Manage TargetDescriptionData
8.2.1	The service shall provide the capability to manage a TargetDescription.
8.2.1.1	The service shall provide the capability to manage a PreservationTarget.
8.2.1.1.1	The service shall provide the capability to specify the date and time an PreservationTarget was created.
8.2.1.1.2	The service shall provide the capability to qualify a PreservationTarget as a PreservationItem.
8.2.1.1.3	The service shall provide the capability to manage a PreservationTarget as a PreservationCollection.
8.2.1.1.3.1	The service shall provide the capability to identify a PreservationItem as a member of a PreservationCollection.
8.2.1.1.4	The service shall provide the capability to identify the Producer of a PreservationTarget as part of the provenance of the PreservationTarget.
8.2.1.1.5	The service shall provide the capability to identify the date of creation of a PreservationTarget.

	Manage PreservationManagementData
Identifier	**Requirement**
8.2.1.1.6	The service shall provide the capability to associate a PreservationTarget with another PreservationTarget.
8.2.1.2	The service shall provide the capability to manage a BinaryEncoding.
8.2.1.2.1	The service shall provide the capability to specify the date and time a BinaryEncoding was created.
8.2.1.2.2.1	The service shall provide the capability to to qualify a BinaryEncoding as obsolete.
8.2.1.2.2.2	The service shall provide the capability to to specify the date and time a BinaryEncoding was qualified as obsolete.
8.2.1.2.3	The service shall provide the capability to specify a predecessor/successor association between two BinaryEncodings.
8.2.1.3	The service shall provide the capability to manage a PreservationTargetComponentDescription.
8.2.1.3.1	The service shall provide the capability to specify the date and time a PreservationTargetComponentDescription was created.
8.2.1.3.2	The service shall provide the capability to associate a PreservationTargetComponentDescription with a BinaryEncoding.
8.2.1.3.3	The service shall provide the capability to associate a PreservationTargetComponentDescription with a DigitalComponent.
8.2.1.3.4	The service shall provide the capability to indicate the type of DigitalComponent a PreservationTargetComponentDescription describes.
8.2.1.3.4.1	The service shall provide the capability to indicate if a PreservationTargetComponentDescription describes a ContentComponent.
8.2.1.3.4.2	The service shall provide the capability to indicate if a PreservationTargetComponentDescription describes a SoftwareComponent.
8.2.1.3.4.3	The service shall provide the capability to indicate if a PreservationTargetComponentDescription describes an InstantiationComponent.
8.2.1.3.5	The service shall provide the capability to ensure that the DigitalComponents identified as elements of a BinaryEncoding are appropriate for instantiating the related PreservationTarget.
8.2.1.3.5.1	The service shall provide the capability to ensure that a SoftwareComponent in a BinaryEncoding can process a ContentComponent associated with it.
8.2.1.3.5.2	The service shall provide the capability to ensure that an InstantiationComponent in a BinaryEncoding can be applied to a ContentComponent associated with it.
8.2.1.3.5.2.1	The service shall provide the capability to ensure that a SoftwareComponent in a BinaryEncoding can apply an associated InstantiationComponent to a ContentComponent associated with it.
8.2.1.3.5.3	The service shall provide the capability to ensure that a SoftwareComponent described in a PreservationTargetComponentDescription as enabling instantiation of the related PreservationTarget is capable of generating the Manifestation.
8.2.1.3.6	The service shall provide the capability to to qualify a PreservationTargetComponentDescription as obsolete.
8.2.1.3.6	The service shall provide the capability to to specify the date and time a PreservationTargetComponentDescription was qualified as obsolete.

	Manage PreservationManagementData
Identifier	**Requirement**
8.2.1.4	The service shall provide the capability to describe a Manifestation in a ManifestationDescription.
8.2.1.4.1	The service shall provide the capability to specify in a ManifestationDescription the date and time an Manifestation was created.
8.2.1.4.2	The service shall provide the capability to associate in a ManifestationDescription a Manifestation with a BinaryEncoding.
8.2.1.4.3	The service shall provide the capability to indicate in a ManifestationDescription the type of a Manifestation.
8.2.1.4.3.1	The service shall provide the capability to indicate in a ManifestationDescription if a Manifestation is a RuntimeVersion.
8.2.1.4.3.2	The service shall provide the capability to indicate in a ManifestationDescription if a Manifestation is a Rendering.
8.2.1.4.3.3	The service shall provide the capability to associate in a ManifestationDescription a Rendering with a RuntimeVersion capable of generating the Rendering.
8.2.1.5	The service shall provide the capability to manage a PermanentFeature.
8.2.1.5.1	The service shall provide the capability to specify the date and time a PermanentFeature. was created.
8.2.1.5.2	The service shall provide the capability to specify the scope of applicability of a PermanentFeature.
8.2.1.5.2.1	The service shall provide the capability to specify that a PermanentFeature belongs to every instance of a class of PreservationTargets.
8.2.1.5.2.2	The service shall provide the capability to specify that a PermanentFeature that belongs to a single PreservationTarget.
8.2.1.5.2.3	The service shall provide the capability to specify that a PermanentFeature belongs to every PreservationTarget that satisfies a specified constraint.
8.2.1.5.1	The service shall provide the capability to associate a PermanentFeature with a PreservationTarget.
8.2.1.5.2	The service shall provide the capability to indicate the type of a PermanentFeature.
8.2.1.5.2.1	The service shall provide the capability to indicate if a PermanentFeature is a UniquenessFeature.
8.2.1.5.2.2	The service shall provide the capability to indicate if a PermanentFeature is a ManifestPermanentFeature.
8.2.1.5.2.3	The service shall provide the capability to indicate if a PermanentFeature is an IntegrityFeature.
8.2.1.5.2.4	The service shall provide the capability to indicate if a PermanentFeature is another type.
8.2.1.5.3	The service shall provide the capability to indicate if a PermanentFeature is an attribute.
8.2.1.5.3.1	The service shall provide the capability to indicate, if a PermanentAttribute is preserved is satisfied by the existence of the attribute.
8.2.1.5.3.2	The service shall provide the capability to indicate, if the preservation of a PermanentAttribute requires that the attribute have a specific benchmark value.
8.2.1.5.3.2.1	The service shall provide the capability to capture the benchmark value of a PermanentAttribute.

	Manage PreservationManagementData
Identifier	**Requirement**
8.2.1.5.3.2.1.1	The service shall provide the capability to capture a single value as the benchmark value of a PermanentAttribute.
8.2.1.5.3.2.1.2	The service shall provide the capability to capture a range of values as the benchmark values of a PermanentAttribute.
8.2.1.5.3.2.1.3	The service shall provide the capability to capture a set of values as the benchmark values of a PermanentAttribute.
8.2.1.5.4	The service shall provide the capability to indicate if a PermanentFeature is an operation.
8.2.1.5.4.1	The service shall provide the capability to indicate if the preservation of a PermanentOperation is satisfied by the ability to perform the operation.
8.2.1.5.4.2	The service shall provide the capability to indicate if the preservation of a PermanentOperation requires that the operation return a specified result.
8.2.1.5.4.2.1	The service shall provide the capability to capture the required result of a PermanentOperation as the benchmark value of that operation.
8.2.1.6	The service shall provide the capability to manage a PermanentFeatureExpression.
8.2.1.6.1	The service shall provide the capability to specify the date and time a PermanentFeatureExpression was created.
8.2.1.6.2	The service shall provide the capability to associate a PermanentFeatureExpression with a PermanentFeature of a PreservationTarget.
8.2.1.6.2.1	The service shall provide the capability to ensure that a PermanentFeature is associated with a PermanentFeatureExpression in a BinaryEncoding.
8.2.1.6.3	The service shall provide the capability to identify the locus of a PermanentFeatureExpression.
8.2.1.6.3.1	The service shall provide the capability to identify if a PermanentFeatureExpression is a ComponentPermanentFeature.
8.2.1.6.3.1.1	The service shall provide the capability to associate a ComponentPermanentFeature with a PreservationTargetComponentDescription.
8.2.1.6.3.1.2	The service shall provide the capability to verify if a ComponentPermanentFeature is appropriate for the type of DigitalComponent described by a PreservationTargetComponentDescription.
8.2.1.6.3.1.2.1	The service shall provide the capability to verify if a ComponentPermanentFeature appropriately describes the expression of the PermanentFeature in a ContentComponent.
8.2.1.6.3.1.2.2	The service shall provide the capability to verify if a ComponentPermanentFeature appropriately describes the expression of the PermanentFeature in a SoftwareComponent.
8.2.1.6.3.1.2.3	The service shall provide the capability to verify if a ComponentPermanentFeature appropriately describes the expression of the PermanentFeature in a InstantiationComponent.
8.2.1.6.3.2	The service shall provide the capability to identify if a PermanentFeatureExpression is an ManifestPermanentFeature.
8.2.1.6.3.2.1	The service shall provide the capability to associate an ManifestPermanentFeature with an PreservationTargetInstantiation.
8.2.1.6.3.2.2	The service shall provide the capability to verify if an ManifestPermanentFeature is appropriate for the type of Manifestation with which it is associated.

	Manage PreservationManagementData
Identifier	**Requirement**
8.2.1.6.3.2.2.1	The service shall provide the capability to verify if an ManifestPermanentFeature appropriately describes the expression of the PermanentFeature in a RunTimeVersion.
8.2.1.6.3.2.2.2	The service shall provide the capability to verify if an ManifestPermanentFeature appropriately describes the expression of the PermanentFeature in a Rendering.
8.2.1.6.3.3	The service shall provide the capability to specify how a PermanentFeatureExpression may be verified.
8.2.1.6.3.3.1	The service shall provide the capability to specify method for verifying that a PermanentFeatureExpression properly expresses the PermanentFeature.
8.2.1.6.3.3.1.1	The service shall provide the capability to specify the role of an Actor in carrying out the method for verifying that a PermanentFeatureExpression properly expresses the PermanentFeature.
8.2.1.6.3.3.2	The service shall provide the capability to specify a result of an action that would constitute evidence that a PermanentFeatureExpression properly expresses the PermanentFeature.
8.2.1.7	The service shall provide the capability to manage HeuristicInformation.
8.2.1.7.1	The service shall provide the capability to specify the date and time HeuristicInformation was created.
8.2.1.7.2	The service shall provide the capability to define a Clarification Association between an instance of HeuristicInformation and a PreservationTarget.
8.2.1.7.2.1	The service shall provide the capability to specify the ClarificationType of a Clarification Association between an instance of HeuristicInformation and a PreservationTarget.
8.2.1.7.3	The service shall provide the capability to define a HeuristicAssociation between an instance of HeuristicInformation and an element of a PreservationTarget.
8.2.1.7.3.1	The service shall provide the capability to define a HeuristicAssociation between an instance of HeuristicInformation and a BinaryEncoding
8.2.1.7.3.2	The service shall provide the capability to define a HeuristicAssociation between an instance of HeuristicInformation and a Manifestation.
8.2.1.7.3.3	The service shall provide the capability to define a HeuristicAssociation between an instance of HeuristicInformation and a PreservationTargetComponentDescription.
	Manage TargetStateData
8.2.2	The service shall provide the capability to manage TargetState.
8.2.2.1	The service shall provide the capability to specify what an instance of TargetState describes.
8.2.2.1.1	The service shall provide the capability to specify the PreservationTarget that an instance of TargetState describes.
8.2.2.1.2	The service shall provide the capability to specify an element of a PreservationTarget that an instance of TargetState describes.
8.2.2.1.2.1	The service shall provide the capability to specify a BinaryEncoding of a PreservationTarget that an instance of TargetState describes.
8.2.2.1.2.2	The service shall provide the capability to specify a PreservationTargetComponentDescription of a PreservationTarget that an instance of TargetState describes.
8.2.2.1.2.3	The service shall provide the capability to specify a Manifestation of a PreservationTarget that an instance of TargetState describes.

	Manage PreservationManagementData
Identifier	**Requirement**
8.2.2.2	The service shall provide the capability to specify a relationship between TargetState and TargetDescription.
8.2.2.3	The service shall provide the capability to associate TargetState with PreservationActionData about a PreservationAction that impacted the state of the PreservationTarget.
8.2.2.2	The service shall provide the capability to manage a specialization of TargetState.
8.2.2.4.1	The service shall provide the capability to manage SubmissionStatus data.
8.2.2.4.1.1	The service shall provide the capability to manage data describing the authorization for submission of a PreservationTarget, Submitter, transmission, contents or conditions
8.2.2.4.1.2	The service shall provide the capability to manage data about the Submitter of a PreservationTarget.
8.2.2.4.1.2.1	The service shall provide the capability to manage data about an Actor authorized to submit a PreservationTarget.
8.2.2.4.1.2.2	The service shall provide the capability to manage data about an Actor who actually submits a PreservationTarget.
8.2.2.4.1.3	The service shall provide the capability to manage data describing the transmission a PreservationTarget.
8.2.2.4.1.3.1	The service shall provide the capability to manage data about the date and time of transmission a PreservationTarget.
8.2.2.4.1.3.1.1	The service shall provide the capability to manage data about the scheduled date and time of transmission a PreservationTarget.
8.2.2.4.1.3.1.2	The service shall provide the capability to manage data about the actual date and time of transmission a PreservationTarget.
8.2.2.4.1.3.2	The service shall provide the capability to manage data describing the receipt of a PreservationTarget.
8.2.2.4.1.3.3	The service shall provide the capability to manage data describing the results of processing a submitted PreservationTarget.
8.2.2.4.2	The service shall provide the capability to manage StorageStatus data.
8.2.2.4.2.1	The service shall provide the capability to manage data specifying the location of a DigitalComponent in PreservationStorage.
8.2.2.4.2.2	The service shall provide the capability to manage data concerning the integrity of a DigitalComponent in PreservationStorage.
8.2.2.4.3	The service shall provide the capability to manage AssessmentStatus data.
8.2.2.4.3.1	The service shall provide the capability to manage data about the results of an Inspection.
8.2.2.4.3.1.1	The service shall provide the capability to manage data about the results of inspecting a data object.
8.2.2.4.3.1.1.1	The service shall provide the capability to manage data about the results of inspecting a PreservationTarget.
8.2.2.4.3.1.1.2	The service shall provide the capability to manage data about the results of inspecting a BinaryEncoding.

	Manage PreservationManagementData
Identifier	**Requirement**
8.2.2.4.3.1.1.3	The service shall provide the capability to manage data about the results of inspecting a PreservationTargetComponentDescription.
8.2.2.4.3.1.1.4	The service shall provide the capability to manage data about the results of inspecting a Manifestation.
8.2.2.4.3.1.2	The service shall provide the capability to manage data about the results of inspecting a DigitalComponent.
8.2.2.4.3.1.3	The service shall provide the capability to manage data about the results of inspecting a Collection.
8.2.2.4.3.1.4	The service shall provide the capability to manage data about the results of instantiating a PreservationTarget.
8.2.2.4.3.2	The service shall provide the capability to manage data about the results of evaluating preservation
8.2.2.4.3.2.1	The service shall provide the capability to manage data about the results of comparing related PreservationManagementData
8.2.2.4.3.2.2	The service shall provide the capability to manage data about the results of comparing an instantiated PreservationTarget to PreservationManagementData about the target.
8.2.2.4.4	The service shall provide the capability to manage VerificationStatus data.
8.2.2.4.4.1	The service shall provide the capability to manage data about the results of verifying a Manifestation of a PreservationTarget.
8.2.2.4.4.2	The service shall provide the capability to manage data about the results of verifying the preservation of a PreservationTarget.
8.2.2.4.4.3	The service shall provide the capability to manage data about the results of verifying HeuristicInformation
8.2.2.4.5	The service shall provide the capability to manage PreservationDescription.
	Manage DigitalTypeRegistries
8.3	The service shall provide the capability to manage a DigitalTypeRegistry
8.3.1	The service shall provide the capability to maintain a DigitalTypeRegistry
8.3.1.1	The service shall provide the capability to create a DigitalTypeRegistry
8.3.1.2	The service shall provide the capability to update a DigitalTypeRegistry
8.3.1.3	The service shall provide the capability to delete a DigitalTypeRegistry
8.3.2	The service shall provide the capability to create a DigitalTypeRegistry specialization
8.3.2.1	The service shall provide the capability to create a DataFormatRegistry
8.3.2.2	The service shall provide the capability to create a SoftwareRegistry
8.3.2.3	The service shall provide the capability to create an InstantiationObjectRegistry
8.3.2.4	The service shall provide the capability to create a user defined DigitalTypeRegistry specialization
8.3.2.5	The service shall provide the capability to import a DigitalTypeRegistry specialization
8.3.3	The service shall provide the capability to manage a DigitalTypeRegistryEntry

	Manage PreservationManagementData
Identifier	**Requirement**
8.3.3.1	The service shall provide the capability to create a DigitalTypeRegistryEntry
8.3.3.2	The service shall provide the capability to update a DigitalTypeRegistryEntry
8.3.3.3	The service shall provide the capability to delete a DigitalTypeRegistryEntry
8.3.3.4	The service shall provide the capability to indicate whether the type of object described by a DigitalTypeRegistryEntry is obsolete or not.
8.3.3.4.1	The service shall provide the capability to propagate the obsolescence status of a DigitalTypeRegistryEntry to a DigitalComponent of the type described by the entry.
8.3.3.4.2	The service shall provide the capability to propagate the obsolescence status of a DigitalTypeRegistryEntry to a BinaryEncoding that contains a DigitalComponent of the type described by the entry.
8.3.3.5	The service shall provide the capability to associate a DigitalTypeRegistryEntry with a DigitalComponent of the type described by the entry.
8.3.3.5.1	The service shall provide the capability to associate a DigitalTypeRegistryEntry with a DigitalComponent that has a dependency relationship with the type of DigitalComponent described by the entry.
8.3.3.6	The service shall provide the capability to associate a DigitalTypeRegistryEntry with a ComponentDescription.
8.3.3.6.1	The service shall provide the capability to ensure that a ComponentDescription associated with a DigitalTypeRegistryEntry describes a DigitalComponent of the type covered by the DigitalTypeRegistryEntry.
	Manage PreservationActionData
8.4	The service shall provide the capability to manage PreservationActionData.
8.4.1	The service shall provide the capability to manage SubmissionProcessingData.
8.4.2	The service shall provide the capability to manage PreservationStorageData.
8.4.3	The service shall provide the capability to manage PreservationChangeData.
8.4.4	The service shall provide the capability to manage PreservationAssessmentData.
8.4.5	The service shall provide the capability to manage VerificationData.
8.4.6	The service shall provide the capability to manage AccessData.
8.4.7	The service shall provide the capability to manage ProblemHistoryData.
8.4.8	The service shall provide the capability to specify a relationship between an instance of PreservationActionData and an instance of PreservationTargetData.
8.4.8.1	The service shall provide the capability to relate PreservationActionData to TargetDescriptionData.
8.4.8.2	The service shall provide the capability to relate PreservationActionData to TargetStateData.
8.4.9	The service shall provide the capability to relate PreservationActionData to a PreservationRule.
8.4.9.1	The service shall provide the capability to relate PreservationActionData to a PreservationRule in which the PreservationAction is a RuleAction.

	Manage PreservationManagementData
Identifier	**Requirement**
8.4.9.2	The service shall provide the capability to relate PreservationActionData to a PreservationRule in which the PreservationAction is a RuledAction.
	Manage PreservationRule
8.5	The service shall provide the capability to manage a PreservationRule.
8.5.1	The service shall provide the capability to formulate an executable PreservationRule.
8.5.1.1	The service shall provide the capability to formulate an executable PreservationRule governing the state of an object.
8.5.1.2	The service shall provide the capability to formulate an executable PreservationRule that causes a RuleAction to be performed.
8.5.1.3	The service shall provide the capability to identify a PreservationRule as part of another PreservationRule.
8.5.1.4	The service shall provide the capability to specify a Success Criterion for a PreservationRule.
8.5.1.4.1	The service shall provide the capability to specify that a Success Criterion applies to an Action.
8.5.1.4.1.1	The service shall provide the capability to specify that a Success Criterion applies to a RuleAction
8.5.1.4.1.2	The service shall provide the capability to specify a Success Criterion that applies to a RuledAction.
8.5.1.4.2	The service shall provide the capability to specify that a Success Criterion applies to a PreservationTarget.
8.5.1.4.2.1	The service shall provide the capability to specify the application of a Success Criterion to a Feature of a PreservationTarget.
8.5.1.4.2.2	The service shall provide the capability to specify that a Success Criterion applies to a ControlledTarget.
8.5.1.4.2.3	The service shall provide the capability to specify that a Success Criterion relates to a ControllingObject.
8.5.1.4.3	The service shall provide the capability to specify a sequential relationship between Success Criteria.
8.5.1.4.4	The service shall provide the capability to specify a ProblemHandlingInstruction to be called if a Success Criterion is not satisfied.
8.5.1.5	The service shall provide the capability to specify the RuleDomain of a PreservationRule.
8.5.1.5.1	The service shall provide the capability to specify the Participation of an Actor in a RuleDomain.
8.5.1.5.1	The service shall provide the capability to specify an Authorizer in a RuleDomain.
8.5.1.5.1	The service shall provide the capability to specify Performer in a RuleDomain.
8.5.1.5.1	The service shall provide the capability to specify an Approver in a RuleDomain.
8.5.1.5.1	The service shall provide the capability to specify a ProblemResolver in a RuleDomain.
8.5.1.5.2	The service shall provide the capability to specify a RuledAction in a RuleDomain.
8.5.1.5.3	The service shall provide the capability to specify a ControlledTarget in a RuleDomain.
8.5.1.5.3.1	The service shall provide the capability to identify an individual PreservationTarget as a ControlledTarget in a RuleDomain.

	Manage PreservationManagementData
Identifier	**Requirement**
8.5.1.5.3.2	The service shall provide the capability to specify a class of PreservationTargets as ControlledTargets in a RuleDomain.
8.5.1.5.4	The service shall provide the capability to specify a ControllingObject in a RuleDomain.
8.5.1.5.4.1	The service shall provide the capability to identify an instance of PreservationManagementData as a ControllingObject in a RuleDomain.
8.5.1.5.4.2	The service shall provide the capability to identify a PreservationTarget as a ControllingObject in a RuleDomain.
	Manage Actor
8.6	The service shall provide the capability to manage an Actor
8.6.1	The service shall provide the capability to register an Actor
8.6.2	The service shall provide the capability to specify Access authorizations of an Actor
8.6.2.1	The service shall provide the capability to specify Access privileges of an Actor
8.6.2.2	The service shall provide the capability to specify Access restrictions of an Actor
8.6.2.3	The service shall provide the capability to limit the time period of an Actor’s Access authorization.
8.6.2.3.1	The service shall provide the capability to specify a start date for an Actor’s Access authorization.
8.6.2.3.2	The service shall provide the capability to specify an end date for an Actor’s Access authorization.
8.6.2.4	The service shall provide the capability to specify the domain in which an Access authorization of an Actor applies.
8.6.2.4.1	The service shall provide the capability to specify Access authorizations of an Actor with respect to a PreservationAction
8.6.2.4.1.1	The service shall provide the capability to specify the mode of Participation in which an Actor is authorized to participate in a PreservationAction
8.6.2.4.2	The service shall provide the capability to specify Access authorizations of an Actor with respect to a RuleDomain
8.6.2.4.3	The service shall provide the capability to specify Access authorizations of an Actor with respect to an IntellectualEntity
8.6.2.4.3.1	The service shall provide the capability to specify Access authorizations of an Actor with respect to a PreservationTarget.
8.6.2.4.3.2	The service shall provide the capability to specify Access authorizations of an Actor with respect to HeuristicInformation
8.6.2.4.3	The service shall provide the capability to specify Access authorizations of an Actor with respect to a ManagementSet.
8.6.3	The service shall provide the capability to capture PreservationManagementData about an Actor’s interactions in a Preservation Environment.
8.6.4	The service shall provide the capability to generate a report about an Actor
8.6.4.1	The service shall provide the capability to generate a report about an Actor’s authorizations

	Manage PreservationManagementData
Identifier	**Requirement**
8.6.4.2	The service shall provide the capability to generate a report about an Actor’s interactions in a Preservation Environment.

10.2.5. Reporting

	Reporting
Identifier	**Requirement**
9.	The service shall provide the capability to manage reports.
9.1	The service shall provide the capability to define the scope of a report.
9.1.1	The service shall provide the capability to define the scope of a report to cover a class of objects.
9.1.2	The service shall provide the capability to define the scope of a report to cover all members of a class of objects.
9.1.3	The service shall provide the capability to define the scope of a report to cover the members of a class of objects that satisfy a specified constraint.
9.1.4	The service shall provide the capability to define the scope of a report to cover a single instance.
9.1.4	The service shall provide the capability to define the scope of a report to cover a single instance and other objects associated with that instance.
9.1.5	The service shall provide the capability to define the scope of a report to cover a ManagementSet.
9.1.6	The service shall provide the capability to define the scope of a report to cover a PreservationNetwork.
9.	The service shall provide the capability to generate a report.
9.2	The service shall provide the capability to generate a report in the form of a defined PreservationManagementDocument.
9.2.1	The service shall provide the capability to generate a NoticeOfIntentToSubmit.
9.2.2	The service shall provide the capability to generate a SubmissionDocument.
9.2.3	The service shall provide the capability to generate a PreservationActionReport.
9.2.4	The service shall provide the capability to generate a ProblemReport.
9.2.5	The service shall provide the capability to generate a ProblemResolutionReport.
9.2.6	The service shall provide the capability to generate a report in the form of a user-defined PreservationManagementDocument.
9.3	The service shall provide the capability to generate a report ad hoc.
9.3.1	The service shall provide the capability to use a reporting tool to generate a report
9.4	The service shall provide the capability to send a report
9.4.1	The service shall provide the capability to send a report to an Actor designated to receive that report in the Actor’s registration.
9.4.2	The service shall provide the capability to send a report to an Actor designated ad hoc.
9.4.3	The service shall provide the capability to send a report to an Actor who requested the report.
9.5	The service shall provide the capability to receive a report
9.5.1	The service shall provide the capability to associate a received report with a pending action.
9.5.2	The service shall provide the capability to associate a received report with a PreservationProblem.

	Reporting
Identifier	Requirement
9.5.3	The service shall provide the capability to associate a received report with an IntellectualEntity.
9.5.4	The service shall provide the capability to associate a received report with a ManagementSet.
9.5.5	The service shall provide the capability to associate a received report with an Actor.
9.5.6	The service shall provide the capability to extract PreservationManagementData from a received report.
9.5.6.1	The service shall provide the capability to assign PreservationManagementData extracted from a received report to the proper subclass of PreservationManagementData.
9.5.6.2	The service shall provide the capability to associate PreservationManagementData extracted from a received report with an object to which it relates.

10.3. Preservation Management Capabilities

10.3.1. Implement Preservation Rules

	Implement PreservationRules
Identifier	**Requirement**
10.	The service shall provide the capability to implement a PreservationRule.
	Define a PreservationRule
10.1	The service shall provide the capability to define a PreservationRule.
	Define a RuleTask in a PreservationRule.
10.1.1	The service shall provide the capability to define a RuleTask in a PreservationRule.
10.1.1.1	The service shall provide the capability to identify a PreservationAction as a RuleTask in a PreservationRule.
10.1.1.2	The service shall provide the capability to define a RuleTask in a PreservationRule ad hoc.
10.1.1.3	The service shall provide the capability to associate an AuthorizedRole with a RuleTask in a PreservationRule.
10.1.1.4	The service shall provide the capability to identify a RuleResult that should be produced by a RuleTask.
	Define a RuleDomain.
10.1.2	The service shall provide the capability to define a RuleDomain.
	Identify a RuledObject in a RuleDomain.
10.1.2.1	The service shall provide the capability to identify a RuledObject in a RuleDomain.
10.1.2.1.1	The service shall provide the capability to identify a PreservationTarget as a RuledObject in a RuleDomain.
10.1.2.1.1.1	The service shall provide the capability to identify a PreservationItem as a RuledObject in a RuleDomain.
10.1.2.1.1.2	The service shall provide the capability to identify a PreservationCollection as a RuledObject in a RuleDomain.
10.1.2.1.2	The service shall provide the capability to identify an instance of HeuristicInformation as a RuledObject in a RuleDomain..
10.1.2.1.3	The service shall provide the capability to identify a BinaryEncoding as a RuledObject in a RuleDomain.
10.1.2.1.4	The service shall provide the capability to identify a ComponentDescription as a RuledObject in a RuleDomain.
10.1.2.1.5	The service shall provide the capability to identify a Manifestation as a RuledObject in a RuleDomain.
10.1.2.1.5.1	The service shall provide the capability to identify a RuntimeVersion as a RuledObject in a RuleDomain.

	Implement PreservationRules
Identifier	**Requirement**
10.1.2.1.5.2	The service shall provide the capability to identify a Rendering as a RuledObject in a RuleDomain.
10.1.2.1.5.3	The service shall provide the capability to identify a ManifestationDescription as a RuledObject in a RuleDomain.
10.1.2.1.6	The service shall provide the capability to identify a DigitalComponent as a RuledObject in a RuleDomain.
10.1.2.1.6.1	The service shall provide the capability to identify a ContentComponent as a RuledObject in a RuleDomain.
10.1.2.1.6.2	The service shall provide the capability to identify a SoftwareComponent as a RuledObject in a RuleDomain.
10.1.2.1.6.3	The service shall provide the capability to identify an InstantiationComponent as a RuledObject in a RuleDomain.
10.1.2.1.7	The service shall provide the capability to identify an PermanentFeature as a RuledObject in a RuleDomain.
10.1.2.1.8	The service shall provide the capability to identify an PermanentFeatureExpression as a RuledObject in a RuleDomain.
10.1.2.1.8.1	The service shall provide the capability to identify a ComponentPermanentFeature as a RuledObject in a RuleDomain.
10.1.2.1.8.2	The service shall provide the capability to identify a ManifestPermanentFeature as a RuledObject in a RuleDomain.
10.1.2.1.8.3	The service shall provide the capability to identify all instances of a class of Objects as RuledObjects in a rule domain RenderedPermanentFeatureExpression as a RuledObject in a RuleDomain.
	Identify a RuledAction in a RuleDomain.
10.1.2.2	The service shall provide the capability to identify a RuledAction in a RuleDomain.
10.1.2.2.1	The service shall provide the capability to identify a precondition that qualifies an action as a RuledAction in a RuleDomain.
10.1.2.2.2	The service shall provide the capability to identify a postcondition of a RuledAction that should be true after the PreservationRule is executed.
	Identify a ControllingObject in a RuleDomain.
10.1.2.3	The service shall provide the capability to identify a ControllingObject in a RuleDomain.
10.1.2.3.1	The service shall provide the capability to identify a Feature of a ControllingObject that should be used to specify a SuccessCondition when the PreservationRule is applied.
10.1.2.3.1.1	The service shall provide the capability to specify that the existence of a Feature of a ControllingObject should be used to specify a SuccessCondition when the PreservationRule is applied.
10.1.2.3.1.2	The service shall provide the capability to specify that the value of a Feature of a ControllingObject should be used to specify a SuccessCondition when the PreservationRule is applied.

	Implement PreservationRules
Identifier	**Requirement**
	Define a SuccessCriterion of a PreservationRule.
10.1.3	The service shall provide the capability to define a SuccessCriterion of a PreservationRule.
	Define a SuccessCriterion of a PreservationRule that applies to a RuledObject.
10.1.3.1	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that applies to a RuledObject.
10.1.3.1.1	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that applies to a Feature a RuledObject.
10.1.3.1.1.1	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that applies to an Attribute of a RuledObject.
10.1.3.1.1.1.1	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that applies to the existence of an Attribute of a RuledObject.
10.1.3.1.1.1.2	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that applies to the value of an Attribute of a RuledObject.
10.1.3.1.1.1.3	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that applies to the mode of expression of an Attribute of a RuledObject.
10.1.3.1.1.2	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that applies to an Operation of a RuledObject.
10.1.3.1.1.2.1	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that applies to the feasibility of executing an Operation of a RuledObject.
10.1.3.1.1.2.2	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that applies to a postcondition of an Operation of a RuledObject.
10.1.3.1.1.2.3	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that applies to a result value of an Operation of a RuledObject.
10.1.3.1.2	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that specifies the relationship between the states of a RuledObject before and after the execution of a RuledAction.
10.1.3.1.2.1	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that stipulates that the state of a RuledObject after the execution of a RuledAction must be identical to its state before the action.
10.1.3.1.2.2	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that allows a difference between the state of a RuledObject after the execution of a RuledAction and its state before the action.
10.1.3.1.2.3	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that requires a difference between the state of a RuledObject after the execution of a RuledAction and its state before the action.
10.1.3.1.2.4	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that specifies the relationship between the states of a RuledObject before and after the execution of a RuledAction in terms of a Feature of the RuledObject.

	Implement PreservationRules
Identifier	**Requirement**
10.1.3.1.2.4.1	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that specifies the relationship between the states of a RuledObject before and after the execution of a RuledAction in terms of an attribute of the RuledObject.
10.1.3.1.2.4.1.1	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that specifies the relationship between the states of a RuledObject before and after the execution of a RuledAction in terms of the existence of an attribute of the RuledObject.
10.1.3.1.2.4.1.2	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that specifies the relationship between the states of a RuledObject before and after the execution of a RuledAction in terms of the value of an attribute of the RuledObject.
10.1.3.1.2.4.1.3	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that specifies the relationship between the states of a RuledObject before and after the execution of a RuledAction in terms of the manner in which an attribute of the RuledObject is expressed.
10.1.3.1.2.4.2	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that specifies the relationship between the states of a RuledObject before and after the execution of a RuledAction in terms of an operation of the RuledObject.
10.1.3.1.2.4.2.1	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that specifies the relationship between the states of a RuledObject before and after the execution of a RuledAction in terms of the feasibility of performing an operation of the RuledObject.
10.1.3.1.2.4.2.2	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that specifies the relationship between the states of a RuledObject before and after the execution of a RuledAction in terms of a postcondition when an operation of the RuledObject is performed.
10.1.3.1.2.4.2.3	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that specifies the relationship between the states of a RuledObject before and after the execution of a RuledAction in terms of a result value when an operation of the RuledObject is performed.
	Define a SuccessCriterion of a PreservationRule that applies to a RuledAction.
10.1.3.2	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that applies to a RuledAction.
10.1.3.2.1	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that applies to the feasibility of executing a RuledAction.
10.1.3.2.2	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that applies to a postcondition of a RuledAction.
10.1.3.2.3	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that applies to a result value a RuledAction.
10.1.3.2.4	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that specifies a relationship between an outcome of a RuledAction and data about a comparable outcome of a prior execution of the RuledAction.
10.1.3.2.4.1	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that requires comparing successful execution to completion of a RuledAction with data about a prior successful execution of the RuledAction.

	Implement PreservationRules
Identifier	**Requirement**
10.1.3.2.4.2	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that requires comparing a postcondition of a RuledAction with data about a comparable postcondition of a prior execution of the RuledAction.
10.1.3.2.4.3	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that requires comparing a return value of a RuledAction with data about a comparable return value of a prior execution of the RuledAction.
10.1.3.2.4.4	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that specifies that an outcome of a RuledAction be equivalent to a comparable outcome of a prior execution of the RuledAction.
10.1.3.2.4.5	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that allows an outcome of a RuledAction to be different than a comparable outcome of a prior execution of the RuledAction.
10.1.3.2.4.6	The service shall provide the capability to define a SuccessCriterion of a PreservationRule that requires that an outcome of a RuledAction be different than a comparable outcome of a prior execution of the RuledAction.
	Define a RuleEvaluator.
10.1.4	The service shall provide the capability to define a RuleEvaluator.
10.1.4.1	The service shall provide the capability to associate a RuleEvaluator with a PreservationRule.
10.1.4.2	The service shall provide the capability to associate a RuleEvaluator with a SuccessCriterion of a PreservationRule.
	Enforce Rules
10.2	The service shall provide the capability to enforce a PreservationRule.
10.2.1	The service shall provide the capability to enforce a PreservationRule automatically.
10.2.1.1	The service shall provide the capability to enforce a PreservationRule automatically in its RuleDomain.
10.2.2	The service shall provide the capability to provide the capability for an Actor to invoke a PreservationRule.
10.2.2.1	The service shall provide the capability to constrain the PreservationRole of an Actor in applications of a PreservationRule to an AuthorizedRole specified in the RuleDomain.
10.2.2.2	The service shall provide the capability to determine if the ActualRole of an Actor in applications of a PreservationRule is an AuthorizedRole specified in the RuleDomain.
10.2.4	The service shall provide the capability to apply a PreservationRule to a RuledAction.
10.2.4.1	The service shall provide the capability to execute a RuleAction when a RuledAction is invoked.
10.2.4	The service shall provide the capability to apply a ControllingObject in the execution of a RuledAction.
10.2.4	The service shall provide the capability to apply a Feature of a ControllingObject specified in the RuleDomain in the execution of a RuledAction.

	Implement PreservationRules
Identifier	**Requirement**
10.2.6	The service shall provide the capability to apply a SuccessCriterion in the implementation of a PreservationRule.
10.2.6.3	The service shall provide the capability to derive a SuccessCondition by applying a SuccessCriterion to a RuleDomain in the implementation of a PreservationRule.
10.2.6.3.1	The service shall provide the capability to identify an object in the RuleDomain of a PreservationRule to which a SuccessCriterion is applicable.
10.2.6.3.1.1	The service shall provide the capability to identify a RuledObject in the RuleDomain of a PreservationRule to which a SuccessCriterion is applicable.
10.2.6.3.1.2	The service shall provide the capability to identify a RuledAction in the RuleDomain of a PreservationRule to which a SuccessCriterion is applicable.
10.2.6.3.1.3	The service shall provide the capability to identify a Feature of an object in the RuleDomain of a PreservationRule to which a SuccessCriterion is applicable.
10.2.6.3.1.3.1	The service shall provide the capability to identify an attribute of an object in the RuleDomain of a PreservationRule to which a SuccessCriterion is applicable.
10.2.6.3.1.3.2	The service shall provide the capability to identify an operation of an object in the RuleDomain of a PreservationRule to which a SuccessCriterion is applicable.
10.2.6.3.2	The service shall provide the capability to determine the value of a SuccessCondition by applying a SuccessCriterion to a RuleDomain in the implementation of a PreservationRule.
10.2.6.3.3	The service shall provide the capability to derive a SuccessCondition by applying a SuccessCriterion to a RuledAction in the RuleDomain in the implementation of a PreservationRule.
10.2.6.3.4	The service shall provide the capability to derive a SuccessCondition by applying a SuccessCriterion to a RuledObject in the RuleDomain in the implementation of a PreservationRule.
10.2.6.4	The service shall provide the capability to determine a SuccessCondition in an instance of applying a PreservationRule.
10.2.6.4.1	The service shall provide the capability to determine a SuccessCondition by applying a SuccessCriterion in an instance of applying a PreservationRule.
10.2.6.4.2	The service shall provide the capability to determine a SuccessCondition in applying a PreservationRule by identifying an element of the RuleDomain governed by a SuccessCriterion.
10.2.6.4.3	The service shall provide the capability to determine an SuccessCondition by applying a SuccessCriterion that targets an object to a RuledObject in an instance of application of a PreservationRule.
10.2.6.4.4	The service shall provide the capability to determine a SuccessCondition by applying a property of a ControllingObject that is specified in the RuleDomain and is relevant to a SuccessCriterion in an instance of application of a PreservationRule.
10.2.7	The service shall provide the capability to determine whether a SuccessCondition is satisfied when a PreservationRule is applied.
10.2.7.1	The service shall provide the capability to apply a RuleEvaluator to a SuccessCondition when a PreservationRule is applied.

	Implement PreservationRules
Identifier	**Requirement**
10.2.7.2	The service shall provide the capability to apply a RuleEvaluator to evaluate a RuleResult when a PreservationRule is applied.
10.2.7.3	The service shall provide the capability to call the appropriate ProblemHandlingInstruction if a RuleEvaluator returns a negative result.
10.2.7.4	The service shall provide the capability to process a ProblemResponseReport related to a problem found in applying a PreservationRule.
	Capture PreservationManagementData about rule application
10.3	The service shall provide the capability to generate PreservationManagementData when a PreservationRule is applied.
10.3.1	The service shall provide the capability to generate PreservationActionData about an action performed when a PreservationRule is applied.
10.3.1.1	The service shall provide the capability to generate PreservationActionData about a RuleTask performed when a PreservationRule is applied.
10.3.1.2	The service shall provide the capability to generate PreservationActionData about a RuledAction performed when a PreservationRule is applied.
10.3.1.3	The service shall provide the capability to generate PreservationActionData that characterizes the completion status of an action performed when a PreservationRule is applied.
10.3.1.4	The service shall provide the capability to generate PreservationActionData about the RuleResult produced when a PreservationRule is applied.
10.3.2	The service shall provide the capability to generate ProblemHistoryData about a problem that occurs in the application of a PreservationRule.
10.3.2.1	The service shall provide the capability to associate ProblemHistoryData generated in executing a ProblemHandlingInstruction called in an instance of applying a PreservationRule with PreservationHistoryData about the problem identified in the application of the rule.
10.3.2.2	The service shall provide the capability to associate ProblemHistoryData related to a problem found in applying a PreservationRule with PreservationActionData about the application of the rule.
10.3.3	The service shall provide the capability to generate PreservationTargetData about an object impacted when a PreservationRule is applied.
10.3.3.1	The service shall provide the capability to generate TargetDescriptionData about an object impacted when a PreservationRule is applied.
10.3.3.2	The service shall provide the capability to generate TargetStateData about an object impacted when a PreservationRule is applied.
10.3.4	The service shall provide the capability to store PreservationManagementData generated when a PreservationRule is applied.

10.3.2. Assessment

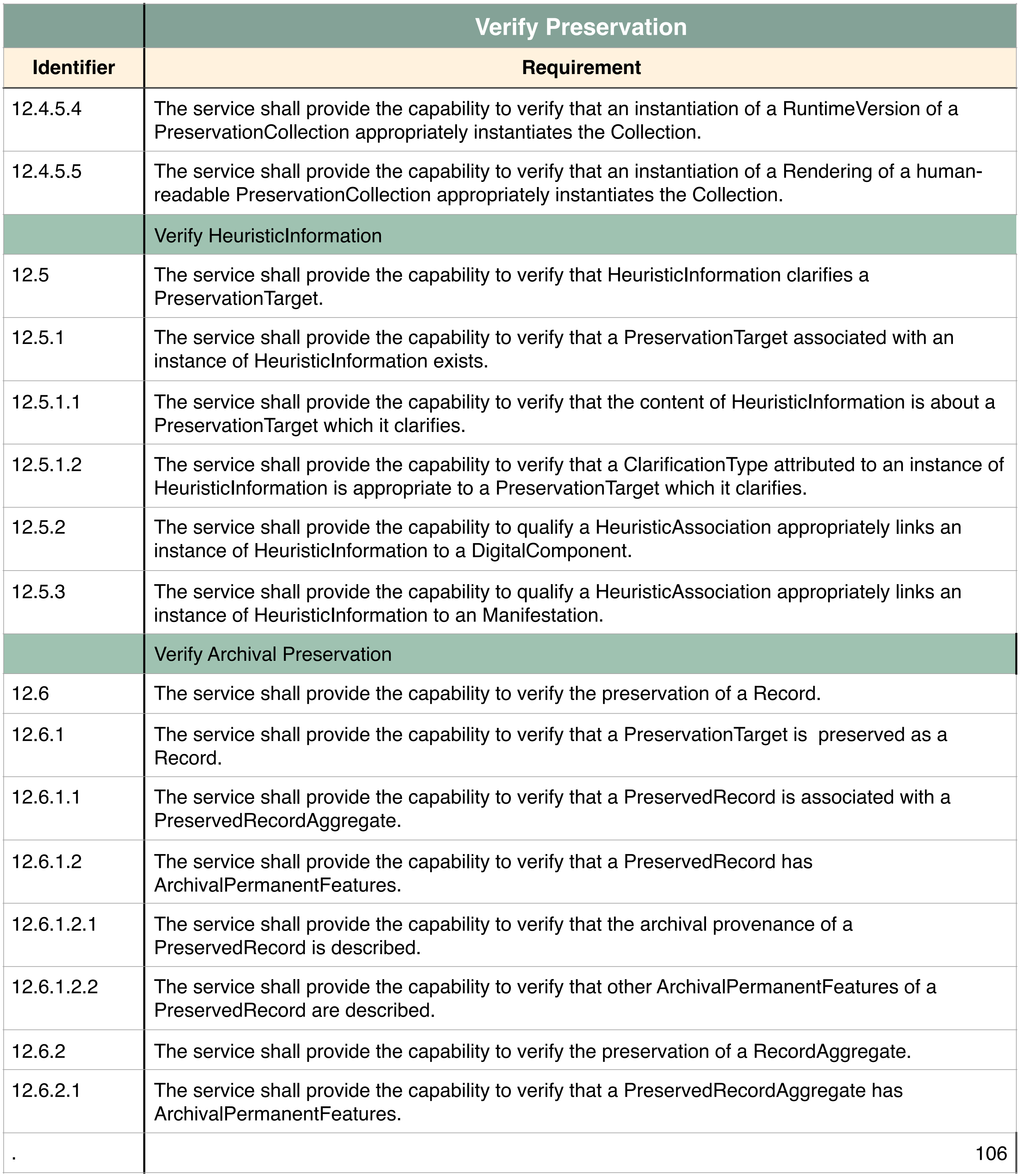
	Verify Preservation
Identifier	**Requirement**
12.	The service shall provide the capability to verify that preservation is successful.
	Capture data about Verification
12.1	The service shall provide the capability to capture data about a Verification action.
12.1.1	The service shall provide the capability to capture VerificationData about a Verification action.
12.1.2	The service shall provide the capability to capture VerificationStatus data about a verified object.
	Handle a problem discovered in Verification
12.2	The service shall provide the capability to handle a problem discovered in connection with a Verification.
12.2.1	The service shall provide the capability to qualify a failure to complete a Verification as a problem.
12.2.2	The service shall provide the capability to qualify a result of “false” in any a Verification as a problem.
12.2.3	The service shall provide the capability to invoke Problem Handling to address a problem related to Verification.
	Verify PreservationTarget preservation
12.3	The service shall provide the capability to verify that a PreservationTarget is preserved.
12.3.1	The service shall provide the capability to verify that an PreservationTarget has at least one BinaryEncoding that is operative; i.e., not obsolete.
12.3.1.1	The service shall provide the capability to verify that data about an operative BinaryEncoding is appropriate.
12.3.1.1.1	The service shall provide the capability to verify the ComponentDescriptions in an operative BinaryEncoding.
12.3.1.1.1.1	The service shall provide the capability to verify that the set of ComponentDescriptions in an operative BinaryEncoding identifies a set of DigitalComponents that are appropriate for instantiating the PreservationTarget represented.
12.3.1.1.1.1.1	The service shall provide the capability to verify that the set of ComponentDescriptions identifies at least one ContentComponent.
12.3.1.1.1.1.2	The service shall provide the capability to verify that the set of ContentComponents describe content that corresponds to the attributes of the PreservationTarget related to its content.
12.3.1.1.1.1.3	The service shall provide the capability to verify that the set of ComponentDescriptions identifies at least one SoftwareComponent.
12.3.1.1.1.1.1	The service shall provide the capability to verify that a SoftwareComponent described in a ComponentDescription is appropriate for processing the ContentComponents with which it is associated in an operative BinaryEncoding.
12.3.1.1.1.1.2	The service shall provide the capability to verify that the set of InstantiationComponents described in ComponentDescriptions in an operative BinaryEncoding is appropriate for application to the ContentComponents with which it is associated.

	Verify Preservation
Identifier	Requirement
12.3.1.1.1.1.2.1	The service shall provide the capability to verify that an InstantiationComponent described in a ComponentDescription is sufficient for instantiating the associated PreservationTarget.
12.3.1.1.1.1.2.2	The service shall provide the capability to verify that a SoftwareComponent described in a ComponentDescription is appropriate for applying all InstantiationComponents with which it is associated in an operative BinaryEncoding to the associated ContentComponents.
12.3.1.1.1.1.4	The service shall provide the capability to verify that the set of DigitalComponents in an operative BinaryEncoding is appropriate for generating a RuntimeVersion of the IntellectualEntity represented by the BinaryEncoding.
12.3.1.1.1.1.5	The service shall provide the capability to verify that the DigitalComponents in an operative BinaryEncoding are appropriate for generating a Rendering of a HumanReadableIntellectualEntity represented by the BinaryEncoding.
12.3.1.1.2	The service shall provide the capability to verify Instantiations in an operative BinaryEncoding.
12.3.1.1.2.1	The service shall provide the capability to verify if an operative BinaryEncoding includes at least one RuntimeVersion.
12.3.1.1.2.1.1	The service shall provide the capability to verify if a RuntimeVersion in an operative BinaryEncoding corresponds to related Features of the PreservationTarget that the BinaryEncoding represents.
12.3.1.1.2.1.2	The service shall provide the capability to verify if a RuntimeVersion includes appropriate DigitalComponents?
12.3.1.1.2.1.3	The service shall provide the capability to verify if a RuntimeVersion appropriately specifies how to load the DigitalComponents it includes?
12.3.1.1.2.2	The service shall provide the capability to verify if an operative BinaryEncoding of a human-readable PreservationTarget includes at least one Rendering of the PreservationTarget.
12.3.1.1.2.2.1	The service shall provide the capability to verify if a Rendering of a human-readable PreservationTarget corresponds to related Features of the PreservationTarget that the BinaryEncoding represents.
12.3.1.1.2.2.2	The service shall provide the capability to verify if a Rendering of a human-readable PreservationTarget is associated with an appropriateRuntimeVersion.
12.3.1.1.2.2.3	The service shall provide the capability to verify if a Rendering of a human-readable PreservationTarget appropriately specifies how to output the PreservationTarget from an associated RuntimeVersion.
12.3.1.1.3	The service shall provide the capability to Verify the PermanentFeatureExpressions in an operative BinaryEncoding.
12.3.1.1.3.1	The service shall provide the capability to verify that an operative BinaryEncoding includes at least one PreservationFeatureExpression of every PreservationFeature of the PreservationTarget it represents.
12.3.1.1.3.2	The service shall provide the capability to verify if each PermanentFeatureExpression is appropriate for the element in which it is expressed.
12.3.1.1.3.2.1	The service shall provide the capability to verify if each PermanentFeatureExpression is associated with an element of the operative BInaryEncoding in which it is expressed.

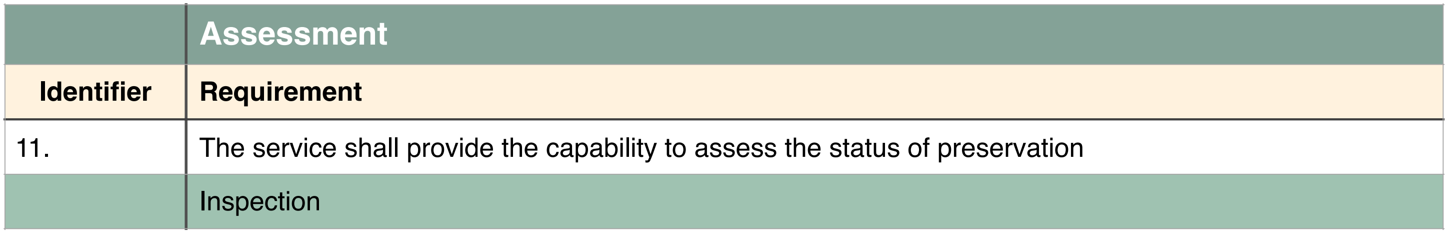
	Verify Preservation
Identifier	**Requirement**
12.3.1.1.x.3.1.1	The service shall provide the capability to verify that a ComponentPermanentFeature is associated with a ComponentDescription in an operative BinaryEncoding.
12.3.1.1.x.3.1.2	The service shall provide the capability to verify that an ManifestPermanentFeature is associated with a Manifestation in an operative BinaryEncoding.
12.3.1.1.3.2.2	The service shall provide the capability to verify that a ComponentPermanentFeature is appropriate for expressing the associated PermanentFeature in the DigitalComponent with which it is associated in an operative BinaryEncoding.
12.3.1.1.3.2.3	The service shall provide the capability to verify if each ManifestPermanentFeature can be inspected in the Manifestation in which it is expressed.
12.3.1.1.3.3	The service shall provide the capability to determine if every PermanentFeatureExpression in an operative BinaryEncoding can be verified.
12.3.1.1.3.3.1	The service shall provide the capability to determine if an appropriate method for verifying a PermanentFeatureExpression in an operative BinaryEncoding is specified.
12.3.1.1.3.3.2	The service shall provide the capability to verify if it can be determined if a PermanentFeatureExpression in an operative BinaryEncoding matches the benchmarkValue of the corresponding PermanentFeature of the PreservationTarget that the BinaryEncoding represents.
12.3.1.2	The service shall provide the capability to verify the DigitalComponents in an operative BinaryEncoding.
12.3.1.2.1	The service shall provide the capability to verify that a DigitalComponent described as being part of an operative BinaryEncoding is available in the PreservationEnvironment.
12.3.1.2.1.1	The service shall provide the capability to verify that a DigitalComponent described as being part of an operative BinaryEncoding is free from corruption.
12.3.1.2.1.2	The service shall provide the capability to verify that a DigitalComponent described as being part of an operative BinaryEncoding can be retrieved.
12.3.1.2.2	The service shall provide the capability to verify that a DigitalComponent corresponds to its ComponentDescription in an operative BinaryEncoding
12.3.1.2.2.1	The service shall provide the capability to verify that DigitalComponent matches its identification in a ComponentDescription.
12.3.1.2.2.2	The service shall provide the capability to verify a DigitalComponent matches the type of Component specified in its ComponentDescription in a operative BinaryEncoding.
12.3.1.2.2.2.1	The service shall provide the capability to verify that a DigitalComponent described by a ContentComponentDescription comprises content of the IntellectualEntity.
12.3.1.2.2.2.2	The service shall provide the capability to verify that a DigitalComponent described by a SoftwareComponentDescription is the software described.
12.3.1.2.2.2.3	The service shall provide the capability to verify that a DigitalComponent described by an InstantiationComponentDescription is the InstantiationComponent described.
12.3.1.2.2.2.4	The service shall provide the capability to verify other attributes of a DigitalComponent described in a ComponentDescription.
12.3.1.2.2.2.4.1	The service shall provide the capability to verify the file size of a DigitalComponent matches that specified in its ComponentDescription.

	Verify Preservation
Identifier	**Requirement**
12.3.1.2.2.2.4.2	The service shall provide the capability to verify the file name of a DigitalComponent matches that identified in its ComponentDescription.
12.3.1.2.2.2.4.3	The service shall provide the capability to verify the storage location of a DigitalComponent matches that identified in its ComponentDescription.
12.3.1.2.2.2.4.4	The service shall provide the capability to verify the data type of a DigitalComponent matches that of the DigitalTypeRegistryEntry associated with it.
12.3.2	The service shall provide the capability to verify the Manifestation of a PreservationTarget.
12.3.2.1	The service shall provide the capability to verify if a RuntimeVersion of a PreservationTarget can be correctly loaded in a computer.
12.3.2.2	The service shall provide the capability to verify if a Rendering of a human-readable PreservationTarget can be correctly output from a RuntimeVersion.
12.3.2.3	The service shall provide the capability to verify that a Manifestation preserves the PermanentFeatures of a PreservationTarget.
12.3.2.3.1	The service shall provide the capability to verify that a Manifestation of a PreservationTarget actually expresses a ManifestPermanentFeature.
12.3.2.3.2	The service shall provide the capability to verify that a ManifestPermanentFeature is correctly expressed in a Manifestation of a PreservationTarget.
12.3.2.3.2.1	The service shall provide the capability to verify that a ManifestPermanentFeature correctly expresses the benchmarkValue of the corresponding PermanentFeature of a PreservationTarget.
	Verify preservation of a PreservationCollection
12.4	The service shall provide the capability to verify that a PreservationCollection is preserved.
12.4.1	The service shall provide the capability to verify that a PreservationCollection has at least one operative BinaryEncoding.
12.4.2	The service shall provide the capability to verify that an operative BinaryEncoding defines the membership of a PreservationCollection.
12.4.2.1	The service shall provide the capability to verify if an operative BinaryEncoding reflects criteria for membership specified in a PreservationCollection.
12.4.2.1.1	The service shall provide the capability to determine if the criteria for membership in a PreservationCollection can be used to identify its members.
12.4.2.1.1.1	The service shall provide the capability to determine if the criteria for membership in a PreservationCollection can be used to identify any missing members.
12.4.2.2	The service shall provide the capability to verify if an operative BinaryEncoding reflects the relationships among members specified in a PreservationCollection.
12.4.3	The service shall provide the capability to verify expression of the PermanentFeatures of a PreservationCollection in an operative BinaryEncoding.
12.4.3.1	The service shall provide the capability to verify that an operative BinaryEncoding includes at least one appropriate PermanentFeatureExpression of each PermanentFeature of the associated PreservationCollection.

	Verify Preservation
Identifier	**Requirement**
12.4.3.1.2	The service shall provide the capability to verify that every PermanentFeatureExpression in an operative BinaryEncoding is appropriately associated with an element of the BinaryEncoding.
12.4.3.1.2.1	The service shall provide the capability to verify that a ComponentPermanentFeature is associated with a ComponentDescription in an operative BinaryEncoding.
12.4.3.1.2.1.1	The service shall provide the capability to verify that a ComponentPermanentFeature is appropriate for expressing the associated PermanentFeature in the Component with which it is associated in an operative BinaryEncoding.
12.4.3.1.2.1.2	The service shall provide the capability to verify if a ComponentPermanentFeature provides a basis for verifying if the benchmark value of the associated PermanentFeature is expressed in the Component with which it is associated in an operative BinaryEncoding.
12.4.3.3.2.2	The service shall provide the capability to verify that an ManifestPermanentFeature is associated with a Manifestation in an operative BinaryEncoding.
12.4.3.3.2.2.1	The service shall provide the capability to verify that an ManifestPermanentFeature is appropriate for expressing the associated PermanentFeature in the Manifestation with which it is associated in an operative BinaryEncoding.
12.4.3.3.2.2.2	The service shall provide the capability to verify if an ManifestPermanentFeature provides a basis for verifying if the benchmark value of the associated PermanentFeature is expressed in the Manifestation with which it is associated in an operative BinaryEncoding.
12.4.3.4	The service shall provide the capability to Determine if every PermanentFeatureExpression in an operative BinaryEncoding is verifiable.
12.4.4	The service shall provide the capability to Verify the membership of a PreservationCollections.
12.4.4.1	The service shall provide the capability to verify if every object identified as a member of a PreservationCollections in an operative BinaryEncoding satisfies the criteria for membership.
12.4.4.1.1	The service shall provide the capability to verify if every object correctly identified as a member of a PreservationCollections in an operative BinaryEncoding exists in the Preservation Environment.
12.4.4.2	The service shall provide the capability to determine if there are any missing members of a PreservationCollection.
12.4.4.3	The service shall provide the capability to verify that a member of a PreservationCollection identified in an operative BinaryEncoding is preserved.
12.4.5	The service shall provide the capability to verify that a PreservationCollection can be instantiated correctly.
12.4.5.1	The service shall provide the capability to verify that a Manifestation of a PreservationCollection includes all members of the Collection.
12.4.5.1.1	The service shall provide the capability to verify that a Manifestation of a PreservationCollection does not include as a member of the Collection any object that does not belong.
12.4.5.2	The service shall provide the capability to verify that a Manifestation of a PreservationCollection associates members in accordance with associations specified in the PreservationCollection.
12.4.5.3	The service shall provide the capability to verify that a Manifestation of the PreservationCollection correctly expresses the PermanentFeatures of the Collection



	Verify Preservation
Identifier	**Requirement**
12.4.5.4	The service shall provide the capability to verify that an instantiation of a RuntimeVersion of a PreservationCollection appropriately instantiates the Collection.
12.4.5.5	The service shall provide the capability to verify that an instantiation of a Rendering of a human-readable PreservationCollection appropriately instantiates the Collection.
	Verify HeuristicInformation
12.5	The service shall provide the capability to verify that HeuristicInformation clarifies a PreservationTarget.
12.5.1	The service shall provide the capability to verify that a PreservationTarget associated with an instance of HeuristicInformation exists.
12.5.1.1	The service shall provide the capability to verify that the content of HeuristicInformation is about a PreservationTarget which it clarifies.
12.5.1.2	The service shall provide the capability to verify that a ClarificationType attributed to an instance of HeuristicInformation is appropriate to a PreservationTarget which it clarifies.
12.5.2	The service shall provide the capability to qualify a HeuristicAssociation appropriately links an instance of HeuristicInformation to a DigitalComponent.
12.5.3	The service shall provide the capability to qualify a HeuristicAssociation appropriately links an instance of HeuristicInformation to an Manifestation.
	Verify Archival Preservation
12.6	The service shall provide the capability to verify the preservation of a Record.
12.6.1	The service shall provide the capability to verify that a PreservationTarget is preserved as a Record.
12.6.1.1	The service shall provide the capability to verify that a PreservedRecord is associated with a PreservedRecordAggregate.
12.6.1.2	The service shall provide the capability to verify that a PreservedRecord has ArchivalPermanentFeatures.
12.6.1.2.1	The service shall provide the capability to verify that the archival provenance of a PreservedRecord is described.
12.6.1.2.2	The service shall provide the capability to verify that other ArchivalPermanentFeatures of a PreservedRecord are described.
12.6.2	The service shall provide the capability to verify the preservation of a RecordAggregate.
12.6.2.1	The service shall provide the capability to verify that a PreservedRecordAggregate has ArchivalPermanentFeatures.
.	106



	Assessment
Identifier	**Requirement**
11.	The service shall provide the capability to assess the status of preservation
	Inspection

	Assessment
Identifier	**Requirement**
11.1	The service shall provide the capability to enable an inspection of preservation status.
11.1.1	The service shall provide the capability to ensure an inspection is conducted in accordance with an AuthorizedRole.
11.1.1.1	The service shall provide the capability to When an Actor has the AuthorizedRole of Authorizer for the conduct of an inspection, obtain the Actor’s approval to conduct the inspection.
11.1.1.2	The service shall provide the capability to When an Actor has the AuthorizedRole of Performer for the conduct of an inspection, allow the Actor to conduct the inspection.
11.1.1.2.1	The service shall provide the capability to When an Actor does not the AuthorizedRole of Performer for the conduct of an inspection, do not allow the Actor to conduct the inspection.
11.1.2	The service shall provide the capability to capture PreservationManagementData from an inspection of preservation status.
11.1.2.1	The service shall provide the capability to capture PreservationManagementData about an inspected object.
11.1.2.2	The service shall provide the capability to capture PreservationManagementData about the performance of the inspection.
11.1.2.3	The service shall provide the capability to capture PreservationManagementData regarding any problem discovered in an inspection.
11.1.2.3.1	The service shall provide the capability to capture ProblemHistoryData regarding any problem related to a PreservationAction discovered in an inspection.
11.1.2.3.2	The service shall provide the capability to capture TargetProblem data regarding any problem related to a PreservationTarget discovered in an inspection.
11.1.2.4	The service shall provide the capability to generate a PreservationActionReport concerning an inspection.
	Inspect objects
11.1.3	The service shall provide the capability to inspect an object.
11.1.3.1	The service shall provide the capability to inspect an object in order to determine what characteristics it has.
11.1.3.1.1	The service shall provide the capability to inspect an object in order to identify a Feature that it has.
11.1.3.1.1.1	The service shall provide the capability to inspect an object in order to identify an attribute that it has.
11.1.3.1.1.1.1	The service shall provide the capability to inspect an object in order to determine the value of an attribute that it has.
11.1.3.1.1.2	The service shall provide the capability to inspect an object in order to identify an operation that it has.
11.1.3.1.1.2.1	The service shall provide the capability to inspect an object in order to identify an attribute of an operation that it has.
11.1.3.1.1.2.2	The service shall provide the capability to inspect an object in order to identify a parameter of an operation that it has.

	Assessment
Identifier	**Requirement**
11.1.3.1.1.2.3	The service shall provide the capability to inspect an object in order to identify a constraint on an operation that it has.
11.1.3.1.1.2.4	The service shall provide the capability to inspect an object in order to identify an association of an operation that it has.
11.1.3.1.2	The service shall provide the capability to inspect a an object in order to identify an association it has.
11.1.3.1.2.1	The service shall provide the capability to inspect an object in order to identify an associated object.
11.1.3.1.3	The service shall provide the capability to inspect an object in order to determine its conformance with an applicable condition.
11.1.3.1.3.1	The service shall provide the capability to inspect an object in order to determine its conformance with an applicable rule.
11.1.3.1.3.2	The service shall provide the capability to inspect an object in order to determine its conformance with an applicable constraint.
11.1.3.1.3.3	The service shall provide the capability to inspect an object in order to determine its conformance with an applicable integrity constraint.
11.1.3.1.3.4	The service shall provide the capability to inspect an object in order to determine its conformance with an applicable consistency constraint.
11.1.3.1.3.5	The service shall provide the capability to inspect an object in order to determine its conformance with an applicable completeness constraint.
11.1.3.1.3.6	The service shall provide the capability to inspect an object in order to determine its conformance with an applicable content constraint.
11.1.3.1.3.7	The service shall provide the capability to inspect an object in order to determine its conformance with an applicable format constraint.
	Inspect TargetDescriptions
11.1.3.2	The service shall provide the capability to inspect a TargetDescription.
11.1.3.2.1	The service shall provide the capability to inspect a PreservationTarget.
11.1.3.2.2	The service shall provide the capability to inspect a PermanentFeature.
11.1.3.2.3	The service shall provide the capability to inspect a BinaryEncoding.
11.1.3.2.4	The service shall provide the capability to inspect a PreserationTargetComponentDescription.
11.1.3.2.5	The service shall provide the capability to inspect a Manifestation.
11.1.3.2.5.1	The service shall provide the capability to inspect a RuntimeVersion.
11.1.3.2.5.2	The service shall provide the capability to inspect a Rendering.
11.1.3.2.6	The service shall provide the capability to inspect an PermanentFeatureExpression.
11.1.3.2.6.1	The service shall provide the capability to inspect a ComponentPermanentFeature.
11.1.3.2.6.2	The service shall provide the capability to inspect an ManifestPermanentFeature.
11.1.3.2.7	The service shall provide the capability to inspect HeuristicInformation.

	Assessment
Identifier	**Requirement**
11.1.3.2.8	The service shall provide the capability to inspect a user-defined class of objects.
	Inspect TargetState
11.1.3.3	The service shall provide the capability to inspect TargetState.
11.1.3.3.1	The service shall provide the capability to inspect SubmissionStatus.
11.1.3.3.2	The service shall provide the capability to inspect StorageStatus.
11.1.3.3.3	The service shall provide the capability to inspect VerficationStatus.
11.1.3.3.4	The service shall provide the capability to inspect AssessmentStatus
11.1.3.3.5	The service shall provide the capability to inspect TargetProblem.
11.1.3.3.6	The service shall provide the capability to inspect PreservationDescription.
	Inspect Instantiations
11.1.3.4	The service shall provide the capability to inspect a specific instantiation of a Manifestation.
11.1.3.4.1	The service shall provide the capability to inspect a specific instantiation of a RunTimeVersion.
11.1.3.4.2	The service shall provide the capability to inspect a specific instantiation of a Rendering.
11.1.3.4.2.1	The service shall provide the capability to inspect a specific instantiation of a Rendering on an output device.
11.1.3.4.2.2	The service shall provide the capability to inspect a specific instantiation of a Rendering on a physical medium.
	Inspect DigitalComponents
11.1.3.5	The service shall provide the capability to inspect a DigitalComponent.
11.1.3.5.1	The service shall provide the capability to inspect a DigitalComponent in order to determine its name.
11.1.3.5.2	The service shall provide the capability to inspect a DigitalComponent in order to determine its size in bytes.
11.1.3.5.3	The service shall provide the capability to inspect a DigitalComponent in order to determine its format.
11.1.3.5.4	The service shall provide the capability to inspect a DigitalComponent in order to determine its storage location.
11.1.3.5.5	The service shall provide the capability to inspect a DigitalComponent in order to determine its integrity.
11.1.3.5.6	The service shall provide the capability to inspect a DigitalComponent in order to access a PermanentFeature expressed in the component.
11.1.3.5.7	The service shall provide the capability to inspect a stored DigitalComponent.
11.1.3.5.8	The service shall provide the capability to inspect a RuntimeVersion of a DigitalComponent.
	Inspect groups of related objects

	Assessment
Identifier	**Requirement**
11.1.3.6	The service shall provide the capability to inspect a group of related objects.
11.1.3.6.1.1	The service shall provide the capability to inspect a group of related objects in order to characterize the group as a whole.
11.1.3.6.1.2	The service shall provide the capability to inspect a group of related objects in order to identify its members.
11.1.3.6.1.2.1	The service shall provide the capability to inspect a group of related objects in order to identify subordinate group.
11.1.3.6.1.2.2	The service shall provide the capability to inspect a group of related objects in order to identify a member that is a single instance.
11.1.3.6.1.2.3	The service shall provide the capability to inspect a group of related objects in order to identify missing members.
11.1.3.6.1.2.4	The service shall provide the capability to inspect a group of related objects in order to identify objects that are incorrectly identified as members.
11.1.3.6.2	The service shall provide the capability to inspect a PreservationCollection.
11.1.3.6.3	The service shall provide the capability to inspect a ManagementSet.
11.1.3.6.3.1	The service shall provide the capability to inspect a PreservationActionSet.
11.1.3.6.3.2	The service shall provide the capability to inspect a SubmissionSet.
11.1.3.6.3.3	The service shall provide the capability to inspect a PreservationNetwork.
	Inspect PreservationActionData
11.1.3.7	The service shall provide the capability to inspect PreservationActionData.
11.1.3.7.1	The service shall provide the capability to inspect SubmissionProcessingData.
11.1.3.7.2	The service shall provide the capability to inspect PreservationChangeData.
11.1.3.7.3	The service shall provide the capability to inspect VerificationData.
11.1.3.7.4	The service shall provide the capability to inspect AssessmentData.
11.1.3.7.5	The service shall provide the capability to inspect AccessData.
11.1.3.7.6	The service shall provide the capability to inspect ProblemHistoryData.
	Evaluate PreservationManagementData
11.2	The service shall provide the capability to evaluate PreservationManagementData.
	Compare Related PreservationManagementData
11.2.1	The service shall provide the capability to compare related PreservationManagementData.
11.2.1.1	The service shall provide the capability to compare instances of the same data element related to the same object.
11.2.1.1.1	The service shall provide the capability to compare instances of the same data element related to the same object obtained at different times.

	Assessment
Identifier	**Requirement**
11.2.1.1.2	The service shall provide the capability to compare instances of the same data element related to the same object obtained by different methods.
11.2.1.1.3	The service shall provide the capability to compare instances of the same data element related to the same object obtained from different sources.
11.2.1.2	The service shall provide the capability to compare corresponding PreservationManagementData about related objects.
11.2.1.2.1	The service shall provide the capability to compare a PreservationTarget with a BinaryEncoding of that target.
11.2.1.2.2	The service shall provide the capability to compare a BinaryEncoding with a ComponentDescription contained in that encoding.
11.2.1.2.3	The service shall provide the capability to compare a BinaryEncoding with a Manifestation contained in that encoding.
11.2.1.2.3.1	The service shall provide the capability to compare a Manifestation with the associated ManifestationDescription.
11.2.1.2.4	The service shall provide the capability to compare BinaryEncodings of the same PreservationTarget.
11.2.1.2.5	The service shall provide the capability to compare a ComponentDescription with the DigitalComponent it describes.
11.2.1.2.6	The service shall provide the capability to compare ComponentDescriptions contained in the same BinaryEncoding.
11.2.1.2.6	The service shall provide the capability to compare ComponentDescriptions of the same DigitalComponents contained in different BinaryEncodings.
11.2.1.2.7	The service shall provide the capability to compare ComponentDescriptions that describe DigitalComponents linked to the same DigitalTypeRegistryEntry.
11.2.1.2.8	The service shall provide the capability to compare a PermanentFeature with an PermanentFeatureExpression of that Feature.
11.2.1.2.8.1	The service shall provide the capability to compare a PermanentFeature with the expression of that Feature in a DigitalComponent.
11.2.1.2.8.2	The service shall provide the capability to compare a PermanentFeature with the expression of that Feature in a Manifestation of the PreservationTarget.
11.2.1.2.8.3	The service shall provide the capability to compare the set of PermanentFeatures of a PreservationTarget with the set of PermanentFeatureExpressions in a BinaryEncoding of the PreservationTarget.
11.2.1.2.9	The service shall provide the capability to compare the PermanentFeatureExpressions of the same PermanentFeature in different loci.
11.2.1.2.9.1	The service shall provide the capability to compare the PermanentFeatureExpressions of the same PermanentFeature in different DigitalComponents.
11.2.1.2.9.2	The service shall provide the capability to compare a ComponentPermanentFeature with an ManifestPermanentFeature of the same PermanentFeature.

	Assessment
Identifier	**Requirement**
11.2.1.2.10	The service shall provide the capability to compare Manifestations of the same PreservationTarget.
11.2.1.2.10.1	The service shall provide the capability to compare a RuntimeVersion with a Rendering contained in the same BinaryEncoding.
11.2.1.2.10.2	The service shall provide the capability to compare RuntimeVersions of a PreservationTarget contained in the different BinaryEncodings of that target.
11.2.1.2.10.3	The service shall provide the capability to compare Renderings of a PreservationTarget contained in the different BinaryEncodings of that target.
11.2.1.2.10.4	The service shall provide the capability to compare Manifestations of different PreservationTargets that belong to the same class of IntellectualEntity.
11.2.1.2.11	The service shall provide the capability to compare a PreservationTarget with HeuristicInformation about that target.
11.2.1.2.12	The service shall provide the capability to compare data about groups of objects
11.2.1.2.12.1	The service shall provide the capability to compare data about a group of objects with data about the set of its members.
11.2.1.2.12.2	The service shall provide the capability to compare data about a group of objects with data about a single member of the group.
11.2.1.2.12.3	The service shall provide the capability to compare data about member of a group of objects with data about another member of the group.
11.2.1.2.12.4	The service shall provide the capability to compare data about similar groups of objects.
11.2.1.3	The service shall provide the capability to compare PreservationActionData with correlated data.
11.2.1.3.1	The service shall provide the capability to compare data about multiple instances of the same PreservationAction.
11.2.1.3.2	The service shall provide the capability to compare data about successive PreservationActions.
11.2.1.3.3	The service shall provide the capability to compare data about an object impacted by a PreservationAction before and after the action.
11.2.1.3.4	The service shall provide the capability to compare data about related objects impacted by a PreservationAction.
11.2.1.3.5	The service shall provide the capability to compare PreservationAction data with related PreservationState data.
11.2.1.5	The service shall provide the capability to compare related data about an Actor.
11.2.1.5.1	The service shall provide the capability to compare an AuthorizedRole of an Actor with the ActualRole of that Actor in a PreservationAction.
11.2.1.5.2	The service shall provide the capability to compare data about an Actor’s ActualRole in multiple PreservationActions.
11.2.1.5	The service shall provide the capability to compare a PreservationRule with correlated data.
11.2.1.5.1	The service shall provide the capability to compare a PreservationRule with correlated PreservationActionData.

	Assessment
Identifier	**Requirement**
11.2.1.5.1.1	The service shall provide the capability to compare a PreservationRule with PreservationActionData about a RuleAction governed by the rule.
11.2.1.5.1.2	The service shall provide the capability to compare a PreservationRule with PreservationActionData about a RuledAction governed by the rule.
11.2.1.5.2	The service shall provide the capability to compare a PreservationRule with correlated PreservationTargetData.
11.2.1.5.2	The service shall provide the capability to compare a PreservationRule with PreservationTargetData about a ControllingObject used in an application of the rule.
11.2.1.5.2	The service shall provide the capability to compare a PreservationRule with PreservationTargetData about a ControlledTarget used in an application of the rule.
11.2.1.5.3	The service shall provide the capability to compare a SuccessCondition with correlated data.
11.2.1.5.3.1	The service shall provide the capability to compare a SuccessCondition with PreservationActionData about a RuledAction in which the condition applies.
11.2.1.5.3.2	The service shall provide the capability to compare a SuccessCondition with PreservationTargetData about a RuledObject to which the condition applies.
	Address Discrepancies in Compared PreservationManagementData
11.2.2	The service shall provide the capability to manage a discrepancy in compared data as a problem.
11.2.2.1	The service shall provide the capability to qualify a discrepancy in data as a problem.
11.2.2.2	The service shall provide the capability to invoke Problem Handling capabilities to address a discrepancy in data.
	Analyze data
11.2.3	The service shall provide the capability to analyze PreservationManaagementData
11.2.3.1	The service shall provide the capability to analyze PreservationManaagementData produced by Inspection
11.2.3.2	The service shall provide the capability to analyze PreservationManaagementData produced by data comparison
11.2.3.3	The service shall provide the capability to analyze PreservationTargetData about a PreservationTarget to determine if it has been preserved successfully.
11.2.3.3.1	The service shall provide the capability to analyze PreservationTargetData about a PreservationTarget to determine if a preservationChange is needed.
11.2.3.3.2	The service shall provide the capability to analyze PreservationTargetData about a PreservationTarget to determine if other remedial action is needed.
11.2.3.4	The service shall provide the capability to analyze PreservationActionData about a PreservationRule to determine if it has been implemented successfully.
11.2.3.4.1	The service shall provide the capability to analyze PreservationActionData about a PreservationRule that has not been implemented successfully to identify appropriate corrective actions.
	Address Discrepancies between PreservationManagementData and related objects

	Assessment
Identifier	**Requirement**
11.2.4	The service shall provide the capability to manage a discrepancy between PreservationManagementData and a related Manifestation as a problem.
11.2.4.1	The service shall provide the capability to qualify a discrepancy between PreservationManagementData and a related Manifestation as a problem.
11.2.4.2	The service shall provide the capability to invoke Problem Handling capabilities to address a discrepancy between PreservationManagementData and a related Instantiation.
	Report on Preservation Assessment
11.3	The service shall provide the capability to generate a PreservationAssessmentReport.
11.3.1	The service shall provide the capability to incorporate PreservationManagementData in a PreservationAssessmentReport.
11.3.1.1	The service shall provide the capability to incorporate data generated in an Inspection in a PreservationAssessmentReport.
11.3.1.2	The service shall provide the capability to incorporate data generated in an Evaluation of PreservationManagementData in a PreservationAssessmentReport.
11.3.1.3	The service shall provide the capability to limit the scope of a PreservationAssessmentReport.
11.3.2.3.1	The service shall provide the capability to limit the scope of a PreservationAssessmentReport to a set of IntellectualEntities.
11.3.2.3.1.1	The service shall provide the capability to limit the scope of a PreservationAssessmentReport to a subclass of IntellectualEntities.
11.3.2.3.2	The service shall provide the capability to limit the scope of a PreservationAssessmentReport to a ManagementSet.
11.3.2.3.3	The service shall provide the capability to limit the scope of a PreservationAssessmentReport to a set of DigitalComponents.
11.3.2.3.4	The service shall provide the capability to limit the scope of a PreservationAssessmentReport to a class of PreservationManagementData.
11.3.2.3.4.1	The service shall provide the capability to limit the scope of a PreservationAssessmentReport to a subclass of PreservationManagementData.
11.3.2.3.5	The service shall provide the capability to limit the scope of a PreservationAssessmentReport to a set of problems identified in evaluation.
11.3.2.3.6	The service shall provide the capability to limit the scope of a PreservationAssessmentReport to a user-defined class.
11.3.3	The service shall provide the capability to incorporate PreservationManagementData captured in an inspection in a PreservationActionReport concerning that inspection.

10.3.3. Verify Preservation

	Verify Preservation
Identifier	**Requirement**
12.	The service shall provide the capability to verify that preservation is successful.
	Capture data about Verification
12.1	The service shall provide the capability to capture data about a Verification action.
12.1.1	The service shall provide the capability to capture VerificationData about a Verification action.
12.1.2	The service shall provide the capability to capture VerificationStatus data about a verified object.
	Handle a problem discovered in Verification
12.2	The service shall provide the capability to handle a problem discovered in connection with a Verification.
12.2.1	The service shall provide the capability to qualify a failure to complete a Verification as a problem.
12.2.2	The service shall provide the capability to qualify a result of “false” in any a Verification as a problem.
12.2.3	The service shall provide the capability to invoke Problem Handling to address a problem related to Verification.
	Verify PreservationTarget preservation
12.3	The service shall provide the capability to verify that a PreservationTarget is preserved.
12.3.1	The service shall provide the capability to verify that an PreservationTarget has at least one BinaryEncoding that is operative; i.e., not obsolete.
12.3.1.1	The service shall provide the capability to verify that data about an operative BinaryEncoding is appropriate.
12.3.1.1.1	The service shall provide the capability to verify the ComponentDescriptions in an operative BinaryEncoding.
12.3.1.1.1.1	The service shall provide the capability to verify that the set of ComponentDescriptions in an operative BinaryEncoding identifies a set of DigitalComponents that are appropriate for instantiating the PreservationTarget represented.
12.3.1.1.1.1.1	The service shall provide the capability to verify that the set of ComponentDescriptions identifies at least one ContentComponent.
12.3.1.1.1.1.2	The service shall provide the capability to verify that the set of ContentComponents describe content that corresponds to the attributes of the PreservationTarget related to its content.
12.3.1.1.1.1.3	The service shall provide the capability to verify that the set of ComponentDescriptions identifies at least one SoftwareComponent.
12.3.1.1.1.1.1	The service shall provide the capability to verify that a SoftwareComponent described in a ComponentDescription is appropriate for processing the ContentComponents with which it is associated in an operative BinaryEncoding.
12.3.1.1.1.1.2	The service shall provide the capability to verify that the set of InstantiationComponents described in ComponentDescriptions in an operative BinaryEncoding is appropriate for application to the ContentComponents with which it is associated.

	Verify Preservation
Identifier	**Requirement**
12.3.1.1.1.1.2.1	The service shall provide the capability to verify that an InstantiationComponent described in a ComponentDescription is sufficient for instantiating the associated PreservationTarget.
12.3.1.1.1.1.2.2	The service shall provide the capability to verify that a SoftwareComponent described in a ComponentDescription is appropriate for applying all InstantiationComponents with which it is associated in an operative BinaryEncoding to the associated ContentComponents.
12.3.1.1.1.1.4	The service shall provide the capability to verify that the set of DigitalComponents in an operative BinaryEncoding is appropriate for generating a RuntimeVersion of the IntellectualEntity represented by the BinaryEncoding.
12.3.1.1.1.1.5	The service shall provide the capability to verify that the DigitalComponents in an operative BinaryEncoding are appropriate for generating a Rendering of a HumanReadableIntellectualEntity represented by the BinaryEncoding.
12.3.1.1.2	The service shall provide the capability to verify Instantiations in an operative BinaryEncoding.
12.3.1.1.2.1	The service shall provide the capability to verify if an operative BinaryEncoding includes at least one RuntimeVersion.
12.3.1.1.2.1.1	The service shall provide the capability to verify if a RuntimeVersion in an operative BinaryEncoding corresponds to related Features of the PreservationTarget that the BinaryEncoding represents.
12.3.1.1.2.1.2	The service shall provide the capability to verify if a RuntimeVersion includes appropriate DigitalComponents?
12.3.1.1.2.1.3	The service shall provide the capability to verify if a RuntimeVersion appropriately specifies how to load the DigitalComponents it includes?
12.3.1.1.2.2	The service shall provide the capability to verify if an operative BinaryEncoding of a human-readable PreservationTarget includes at least one Rendering of the PreservationTarget.
12.3.1.1.2.2.1	The service shall provide the capability to verify if a Rendering of a human-readable PreservationTarget corresponds to related Features of the PreservationTarget that the BinaryEncoding represents.
12.3.1.1.2.2.2	The service shall provide the capability to verify if a Rendering of a human-readable PreservationTarget is associated with an appropriateRuntimeVersion.
12.3.1.1.2.2.3	The service shall provide the capability to verify if a Rendering of a human-readable PreservationTarget appropriately specifies how to output the PreservationTarget from an associated RuntimeVersion.
12.3.1.1.3	The service shall provide the capability to Verify the PermanentFeatureExpressions in an operative BinaryEncoding.
12.3.1.1.3.1	The service shall provide the capability to verify that an operative BinaryEncoding includes at least one PreservationFeatureExpression of every PreservationFeature of the PreservationTarget it represents.
12.3.1.1.3.2	The service shall provide the capability to verify if each PermanentFeatureExpression is appropriate for the element in which it is expressed.
12.3.1.1.3.2.1	The service shall provide the capability to verify if each PermanentFeatureExpression is associated with an element of the operative BInaryEncoding in which it is expressed.
12.3.1.1.x.3.1.1	The service shall provide the capability to verify that a ComponentPermanentFeature is associated with a ComponentDescription in an operative BinaryEncoding.

	Verify Preservation
Identifier	**Requirement**
12.3.1.1.x.3.1.2	The service shall provide the capability to verify that an ManifestPermanentFeature is associated with a Manifestation in an operative BinaryEncoding.
12.3.1.1.3.2.2	The service shall provide the capability to verify that a ComponentPermanentFeature is appropriate for expressing the associated PermanentFeature in the DigitalComponent with which it is associated in an operative BinaryEncoding.
12.3.1.1.3.2.3	The service shall provide the capability to verify if each ManifestPermanentFeature can be inspected in the Manifestation in which it is expressed.
12.3.1.1.3.3	The service shall provide the capability to determine if every PermanentFeatureExpression in an operative BinaryEncoding can be verified.
12.3.1.1.3.3.1	The service shall provide the capability to determine if an appropriate method for verifying a PermanentFeatureExpression in an operative BinaryEncoding is specified.
12.3.1.1.3.3.2	The service shall provide the capability to verify if it can be determined if a PermanentFeatureExpression in an operative BinaryEncoding matches the benchmarkValue of the corresponding PermanentFeature of the PreservationTarget that the BinaryEncoding represents.
12.3.1.2	The service shall provide the capability to verify the DigitalComponents in an operative BinaryEncoding.
12.3.1.2.1	The service shall provide the capability to verify that a DigitalComponent described as being part of an operative BinaryEncoding is available in the PreservationEnvironment.
12.3.1.2.1.1	The service shall provide the capability to verify that a DigitalComponent described as being part of an operative BinaryEncoding is free from corruption.
12.3.1.2.1.2	The service shall provide the capability to verify that a DigitalComponent described as being part of an operative BinaryEncoding can be retrieved.
12.3.1.2.2	The service shall provide the capability to verify that a DigitalComponent corresponds to its ComponentDescription in an operative BinaryEncoding
12.3.1.2.2.1	The service shall provide the capability to verify that DigitalComponent matches its identification in a ComponentDescription.
12.3.1.2.2.2	The service shall provide the capability to verify a DigitalComponent matches the type of Component specified in its ComponentDescription in a operative BinaryEncoding.
12.3.1.2.2.2.1	The service shall provide the capability to verify that a DigitalComponent described by a ContentComponentDescription comprises content of the IntellectualEntity.
12.3.1.2.2.2.2	The service shall provide the capability to verify that a DigitalComponent described by a SoftwareComponentDescription is the software described.
12.3.1.2.2.2.3	The service shall provide the capability to verify that a DigitalComponent described by an InstantiationComponentDescription is the InstantiationComponent described.
12.3.1.2.2.2.4	The service shall provide the capability to verify other attributes of a DigitalComponent described in a ComponentDescription.
12.3.1.2.2.2.4.1	The service shall provide the capability to verify the file size of a DigitalComponent matches that specified in its ComponentDescription.
12.3.1.2.2.2.4.2	The service shall provide the capability to verify the file name of a DigitalComponent matches that identified in its ComponentDescription.

	Verify Preservation
Identifier	**Requirement**
12.3.1.2.2.2.4.3	The service shall provide the capability to verify the storage location of a DigitalComponent matches that identified in its ComponentDescription.
12.3.1.2.2.2.4.4	The service shall provide the capability to verify the data type of a DigitalComponent matches that of the DigitalTypeRegistryEntry associated with it.
12.3.2	The service shall provide the capability to verify the Manifestation of a PreservationTarget.
12.3.2.1	The service shall provide the capability to verify if a RuntimeVersion of a PreservationTarget can be correctly loaded in a computer.
12.3.2.2	The service shall provide the capability to verify if a Rendering of a human-readable PreservationTarget can be correctly output from a RuntimeVersion.
12.3.2.3	The service shall provide the capability to verify that a Manifestation preserves the PermanentFeatures of a PreservationTarget.
12.3.2.3.1	The service shall provide the capability to verify that a Manifestation of a PreservationTarget actually expresses a ManifestPermanentFeature.
12.3.2.3.2	The service shall provide the capability to verify that a ManifestPermanentFeature is correctly expressed in a Manifestation of a PreservationTarget.
12.3.2.3.2.1	The service shall provide the capability to verify that a ManifestPermanentFeature correctly expresses the benchmarkValue of the corresponding PermanentFeature of a PreservationTarget.
	Verify preservation of a PreservationCollection
12.4	The service shall provide the capability to verify that a PreservationCollection is preserved.
12.4.1	The service shall provide the capability to verify that a PreservationCollection has at least one operative BinaryEncoding.
12.4.2	The service shall provide the capability to verify that an operative BinaryEncoding defines the membership of a PreservationCollection.
12.4.2.1	The service shall provide the capability to verify if an operative BinaryEncoding reflects criteria for membership specified in a PreservationCollection.
12.4.2.1.1	The service shall provide the capability to determine if the criteria for membership in a PreservationCollection can be used to identify its members.
12.4.2.1.1.1	The service shall provide the capability to determine if the criteria for membership in a PreservationCollection can be used to identify any missing members.
12.4.2.2	The service shall provide the capability to verify if an operative BinaryEncoding reflects the relationships among members specified in a PreservationCollection.
12.4.3	The service shall provide the capability to verify expression of the PermanentFeatures of a PreservationCollection in an operative BinaryEncoding.
12.4.3.1	The service shall provide the capability to verify that an operative BinaryEncoding includes at least one appropriate PermanentFeatureExpression of each PermanentFeature of the associated PreservationCollection.
12.4.3.1.2	The service shall provide the capability to verify that every PermanentFeatureExpression in an operative BinaryEncoding is appropriately associated with an element of the BinaryEncoding.

	Verify Preservation
Identifier	**Requirement**
12.4.3.1.2.1	The service shall provide the capability to verify that a ComponentPermanentFeature is associated with a ComponentDescription in an operative BinaryEncoding.
12.4.3.1.2.1.1	The service shall provide the capability to verify that a ComponentPermanentFeature is appropriate for expressing the associated PermanentFeature in the Component with which it is associated in an operative BinaryEncoding.
12.4.3.1.2.1.2	The service shall provide the capability to verify if a ComponentPermanentFeature provides a basis for verifying if the benchmark value of the associated PermanentFeature is expressed in the Component with which it is associated in an operative BinaryEncoding.
12.4.3.3.2.2	The service shall provide the capability to verify that an ManifestPermanentFeature is associated with a Manifestation in an operative BinaryEncoding.
12.4.3.3.2.2.1	The service shall provide the capability to verify that an ManifestPermanentFeature is appropriate for expressing the associated PermanentFeature in the Manifestation with which it is associated in an operative BinaryEncoding.
12.4.3.3.2.2.2	The service shall provide the capability to verify if an ManifestPermanentFeature provides a basis for verifying if the benchmark value of the associated PermanentFeature is expressed in the Manifestation with which it is associated in an operative BinaryEncoding.
12.4.3.4	The service shall provide the capability to Determine if every PermanentFeatureExpression in an operative BinaryEncoding is verifiable.
12.4.4	The service shall provide the capability to Verify the membership of a PreservationCollections.
12.4.4.1	The service shall provide the capability to verify if every object identified as a member of a PreservationCollections in an operative BinaryEncoding satisfies the criteria for membership.
12.4.4.1.1	The service shall provide the capability to verify if every object correctly identified as a member of a PreservationCollections in an operative BinaryEncoding exists in the Preservation Environment.
12.4.4.2	The service shall provide the capability to determine if there are any missing members of a PreservationCollection.
12.4.4.3	The service shall provide the capability to verify that a member of a PreservationCollection identified in an operative BinaryEncoding is preserved.
12.4.5	The service shall provide the capability to verify that a PreservationCollection can be instantiated correctly.
12.4.5.1	The service shall provide the capability to verify that a Manifestation of a PreservationCollection includes all members of the Collection.
12.4.5.1.1	The service shall provide the capability to verify that a Manifestation of a PreservationCollection does not include as a member of the Collection any object that does not belong.
12.4.5.2	The service shall provide the capability to verify that a Manifestation of a PreservationCollection associates members in accordance with associations specified in the PreservationCollection.
12.4.5.3	The service shall provide the capability to verify that a Manifestation of the PreservationCollection correctly expresses the PermanentFeatures of the Collection
12.4.5.4	The service shall provide the capability to verify that an instantiation of a RuntimeVersion of a PreservationCollection appropriately instantiates the Collection.

	Verify Preservation
Identifier	**Requirement**
12.4.5.5	The service shall provide the capability to verify that an instantiation of a Rendering of a human-readable PreservationCollection appropriately instantiates the Collection.
	Verify HeuristicInformation
12.5	The service shall provide the capability to verify that HeuristicInformation clarifies a PreservationTarget.
12.5.1	The service shall provide the capability to verify that a PreservationTarget associated with an instance of HeuristicInformation exists.
12.5.1.1	The service shall provide the capability to verify that the content of HeuristicInformation is about a PreservationTarget which it clarifies.
12.5.1.2	The service shall provide the capability to verify that a ClarificationType attributed to an instance of HeuristicInformation is appropriate to a PreservationTarget which it clarifies.
12.5.2	The service shall provide the capability to qualify a HeuristicAssociation appropriately links an instance of HeuristicInformation to a DigitalComponent.
12.5.3	The service shall provide the capability to qualify a HeuristicAssociation appropriately links an instance of HeuristicInformation to an Manifestation.
	Verify Archival Preservation
12.6	The service shall provide the capability to verify the preservation of a Record.
12.6.1	The service shall provide the capability to verify that a PreservationTarget is preserved as a Record.
12.6.1.1	The service shall provide the capability to verify that a PreservedRecord is associated with a PreservedRecordAggregate.
12.6.1.2	The service shall provide the capability to verify that a PreservedRecord has ArchivalPermanentFeatures.
12.6.1.2.1	The service shall provide the capability to verify that the archival provenance of a PreservedRecord is described.
12.6.1.2.2	The service shall provide the capability to verify that other ArchivalPermanentFeatures of a PreservedRecord are described.
12.6.2	The service shall provide the capability to verify the preservation of a RecordAggregate.
12.6.2.1	The service shall provide the capability to verify that a PreservedRecordAggregate has ArchivalPermanentFeatures.

10.3.4. Problem Handling

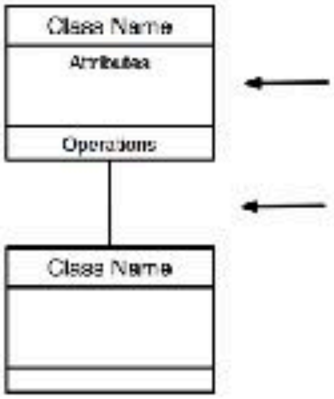
	Problem Handling
Identifier	**Requirement**
13.	The service shall provide the capability to address problems related to preservation.
	Recognize problems
13.1	The service shall provide the capability to recognize a problem related to preservation.
13.1.1	The service shall provide the capability to recognize a problem related to a PreservationAction.
13.1.1.1	The service shall provide the capability to recognize a problem in the execution of a PreservationAction.
13.1.1.1.1	The service shall provide the capability to determine when a Preservation Action fails to run to completion.
13.1.1.1.2	The service shall provide the capability to determine, when relevant, the point in the execution of a Preservation Action where the process aborted
13.1.1.2	The service shall provide the capability to recognize a problem in the outcome of a PreservationAction.
13.1.1.2.1	The service shall provide the capability to determine how the outcome of a PreservationAction deviates from what should have been the result.
13.1.1.3	The service shall provide the capability to specify other types of execution problems.
13.1.2	The service shall provide the capability to recognize a problem related to a PreservationTarget.
13.1.2.1	The service shall provide the capability to recognize a problem in the TargetDescription of a PreservationTarget.
13.1.2.2	The service shall provide the capability to recognize a problem in the TargetState of a PreservationTarget.
13.1.3	The service shall provide the capability to recognize a problem in the relationship between a PreservationTarget and another object.
13.1.3.1	The service shall provide the capability to recognize a problem in the relationship between a PreservationTarget and a BinaryEncoding.
13.1.3.2	The service shall provide the capability to recognize a problem in the relationship between a PreservationTarget and a HeuristicInformation object.
13.1.3.3	The service shall provide the capability to recognize a problem in the relationship between PreservationTargets.
13.1.4	The service shall provide the capability to recognize a problem in PreservationManagementInformation.
13.1.4.1	The service shall provide the capability to recognize a problem in a PreservationManagementDocument.
13.1.4.2	The service shall provide the capability to recognize a problem in PreservationManagementData.
13.1.5	The service shall provide the capability to recognize a problem in the application of a PreservationRule.
	Record PreservationManagementData about problems
13.2	The service shall provide the capability to capture PreservationManagementData about a problem.
13.2.1	The service shall provide the capability to allocate PreservationManagementData about a problem to the appropriate subclass of PreservationManagementData.
13.2.1.1	The service shall provide the capability to capture PreservationHistoryData about a problem related to a PreservationAction.

	Problem Handling
Identifier	**Requirement**
13.2.1.2	The service shall provide the capability to capture TargetProblem data about a problem related to a PreservationAction.
13.2.2	The service shall provide the capability to record data about the circumstances related to identification of a problem.
13.2.2.1	The service shall provide the capability to specify the date and time when a problem was identified
13.2.2.1.1	The service shall provide the capability to specify if there is a difference between the date and time when a problem was identified and the date and time when the problem occurred.
13.2.2.2	The service shall provide the capability to specify the basis for identifying a problem.
13.2.3	The service shall provide the capability to identify what a problem concerns.
13.2.3.1	The service shall provide the capability to identify a class of objects that a problem concerns.
13.2.3.1.1	The service shall provide the capability to identify an instance of a class that a problem concerns.
13.2.3.1.2	The service shall provide the capability to identify an association between instances that a problem concerns.
13.2.3.4	The service shall provide the capability to identify a ManagementSet that a problem concerns.
13.2.3.5	The service shall provide the capability to identify an Actor a problem concerns.
13.2.3.6	The service shall provide the capability to identify a PreservationAction a problem concerns.
13.2.3.7	The service shall provide the capability to identify PreservationManagementInformation a problem concerns.
13.2.4	The service shall provide the capability to specify the nature of a problem.
13.2.4.1	The service shall provide the capability to specify a problem with an element of PreservationManagementData.
13.2.4.1.1	The service shall provide the capability to specify if a required data element is missing.
13.2.4.1.2	The service shall provide the capability to specify if data does not satisfy a PreservationRule.
13.2.4.1.3	The service shall provide the capability to specify if there is a discrepancy between instances of the same data element.
13.2.4.1.3.1	The service shall provide the capability to specify if there is a discrepancy between instances of the same data element obtained from different sources.
13.2.4.1.3.2	The service shall provide the capability to specify if there is a discrepancy between instances of the same data element obtained at different times.
13.2.4.1.3.3	The service shall provide the capability to specify if there is a discrepancy between an element of PreservationManagementData and the object the data is about.
13.2.4.2	The service shall provide the capability to specify a problem with an object.
13.2.4.2.1	The service shall provide the capability to specify a problem concerning the existence an object.
13.2.4.2.2	The service shall provide the capability to specify a problem concerning a Feature an object.
13.2.4.2.3	The service shall provide the capability to specify a problem concerning an association of an object.

	Problem Handling
Identifier	**Requirement**
13.2.4.2.3	The service shall provide the capability to specify a problem concerning the Instantiation of an object.
13.2.4.4	The service shall provide the capability to specify a problem with a class of objects.
13.2.4.5	The service shall provide the capability to specify a problem with a ManagementSet.
13.2.4.6	The service shall provide the capability to specify a problem with a PreservationAction.
13.2.4.7	The service shall provide the capability to specify a problem with an Actor.
13.2.4.7.1	The service shall provide the capability to specify a problem concerning an AuthorizedRole of an Actor.
13.2.4.7.2	The service shall provide the capability to specify a problem concerning an ActualRole of an Actor.
	Address problems
13.3	The service shall provide the capability to address a problem
13.3.1	The service shall provide the capability to apply defined criteria to characterize the severity of a problem.
13.3.2	The service shall provide the capability to apply a ProblemHandlingInstruction
13.3.3	The service shall provide the capability to generate a ProblemReport describing a problem.
13.3.4	The service shall provide the capability to support efforts to resolve a problem.
13.3.4.1	The service shall provide the capability to refer a problem to an Actor designated as a ProblemResolver for the problem.
	Process Response to ProblemReport
13.3.4.2	The service shall provide the capability to The system shall have the capability to process a ProblemResolutionReport.
13.3.4.2.1	The service shall provide the capability to The system shall have the capability to receive a ProblemResolutionReport.
13.3.4.2.2	The service shall provide the capability to The system shall have the capability to associate a ProblemResolutionReport with ProblemHistoryData related to the problem.
13.3.4.2.3	The service shall provide the capability to The system shall have the capability to refer a ProblemResolutionReport to an Actor designated as an Approver for the problem.
13.3.4.2.4	The service shall provide the capability to The system shall have the capability to extract ProblemHistoryData from a ProblemResolutionReport.
13.3.4.2.4.1	The service shall provide the capability to The system shall have the capability to save ProblemHistoryData extracted from a ProblemResolutionReport.
13.3.4.2.5	The service shall provide the capability to The system shall have the capability to determine if a response to a problem appropriately addresses the problem.
13.3.4.3	The service shall provide the capability to implement a ProblemResolution.
13.3.4.3.1.1	The service shall provide the capability to correct problematic PreservationManagementData.
13.3.4.3.1.2	The service shall provide the capability to update a PreservationTarget to correct a problem.
13.3.4.3.1.3	The service shall provide the capability to create a new BinaryEncoding to correct a problem.

	Problem Handling
Identifier	Requirement
13.3.4.3.1.4	The service shall provide the capability to replace a DigitalComponent to correct a problem.
13.3.4.3.1.5	The service shall provide the capability to change a PreservationAction to correct a problem.
13.3.4.3.1.6	The service shall provide the capability to change a PreservationRule to correct a problem.
13.3.4.3.1.7	The service shall provide the capability to change the PreservationRole of an Actor to correct a problem.
13.3.4.3.1.8	The service shall provide the capability to implement other corrective action.
13.3.4.4	The service shall provide the capability to prevent implementation of a ProblemResolution disapproved by an authorized Approver.
13.3.4.5	The service shall provide the capability to create and maintain ProblemHistoryData about an effort to resolve a problem
13.3.4.5.1	The service shall provide the capability to record data indicating whether an attempt to resolve a problem satisfied a requirement for resolution.
13.3.4.5.1.1	The service shall provide the capability to record data identifying an applicable requirement for resolution that was not satisfied in an effort to resolve a problem.
13.3.4.5.2	The service shall provide the capability to record the status of an effort to resolve a problem.
13.3.4.5.2.1	The service shall provide the capability to initiate the status of an effort to resolve a problem as “open.”
13.3.4.5.2.2	The service shall provide the capability to change the status of an effort to resolve a problem to “resolved.”
13.3.4.5.2.3	The service shall provide the capability to change the status of an effort to resolve a problem to “partially resolved.”
13.3.4.5.2.4	The service shall provide the capability to change the status of an effort to resolve a problem to “failed.”

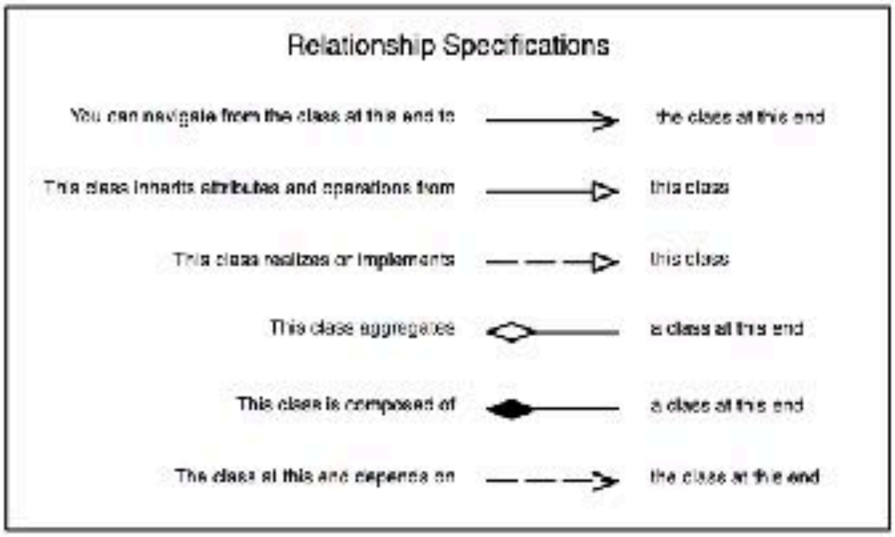
Appendix 1: Notation in UML Class Diagrams



A three part rectangular box symbolizes a Class. The name of the Class must appear in the top section. Attributes may appear in the middle section and operations may appear in the bottom section.

A line connecting two classes indicates they are related.

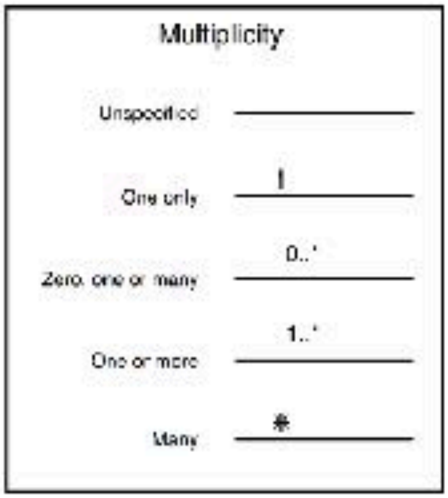
Either end of the line can be embellished to specify the relationship.

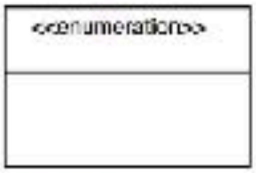

Relationship Specifications
You can navigate from the class at this end to
the class at this end
This class inherits attributes and operations from
this class
This class realizes or implements
this class
This class aggregates
a class at this end
This class is composed of
a class at this end
The class at this end depends on
the class at this end

Text, usually a verb or verb phrase, near an association line describes the relationship between the classes connected by the association.

Text in curly brackets { } specifies a constraint on the association; i.e., the association exists only if the constraint is true.

Text near the end of an association line identifies the role played in the association by an instance of the class at that end.





An enumeration is a list of literals that are the possible values that an attribute of the class that depends on the enumeration may have.

Its icon is a rectangle with two compartments. "<<enumeration>>" is placed in the top compartment and the name of the enumeration below it. The possible values in the enumeration are listed, one per line, in the lower compartment.

Appendix 2: The Records Management Services Attribute Profile

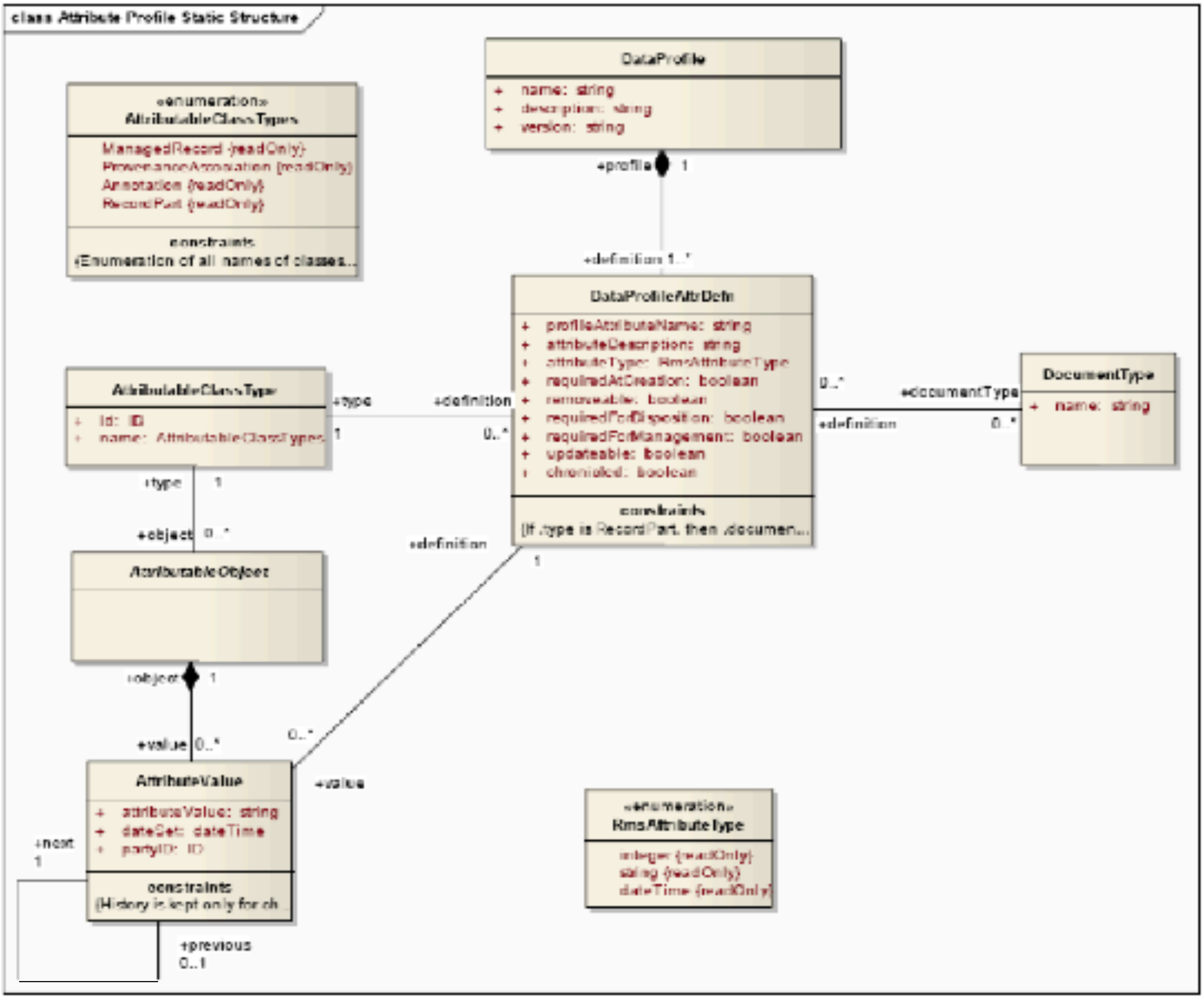


Figure 19 Attribute Profile Static Structure

(from OMG Records Management Services Specification)

Appendix 3: The PaaST Domain Model Overview

This appendix provides an overview of the PaaST domain model by means of three class diagrams for PreservationTargets, the stuff being preserved, PreservationActions, the activities that accomplish preservation, and PreservationManagementInformation, the information necessary to carry out PreservationActions and evaluate the success of preservation. The diagrams do not include every classifier included in the preceding chapters, only those that are basic to each package. Enumerations, for example, are not included in the diagrams. Furthermore, the three diagrams only include the basic associations between classes. A diagram may include one or more classes from other packages to show the relationships among packages. Where that is the case, the namespace from which the class originates is indicated below the name of the class.

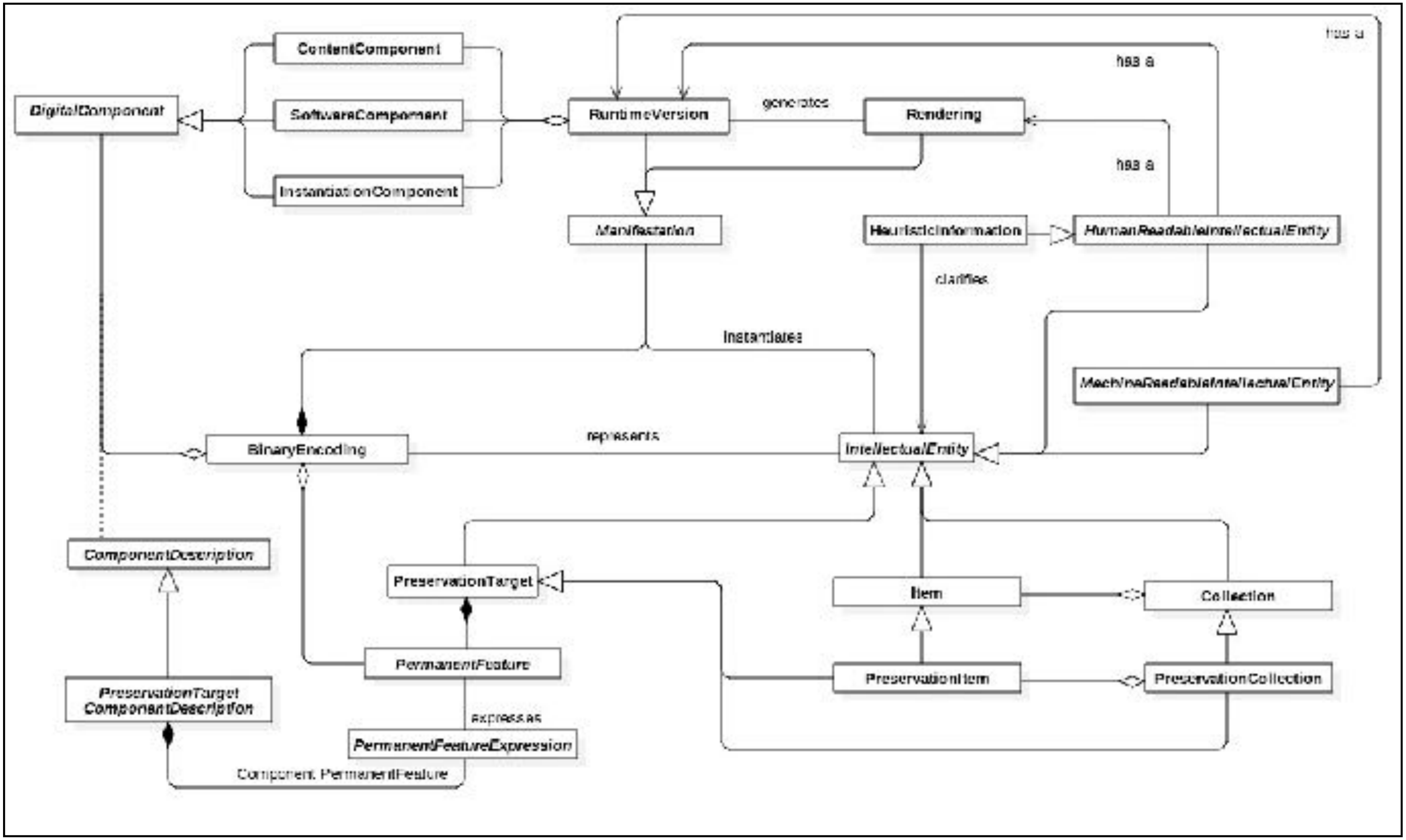


Figure 20, PreservationTargets

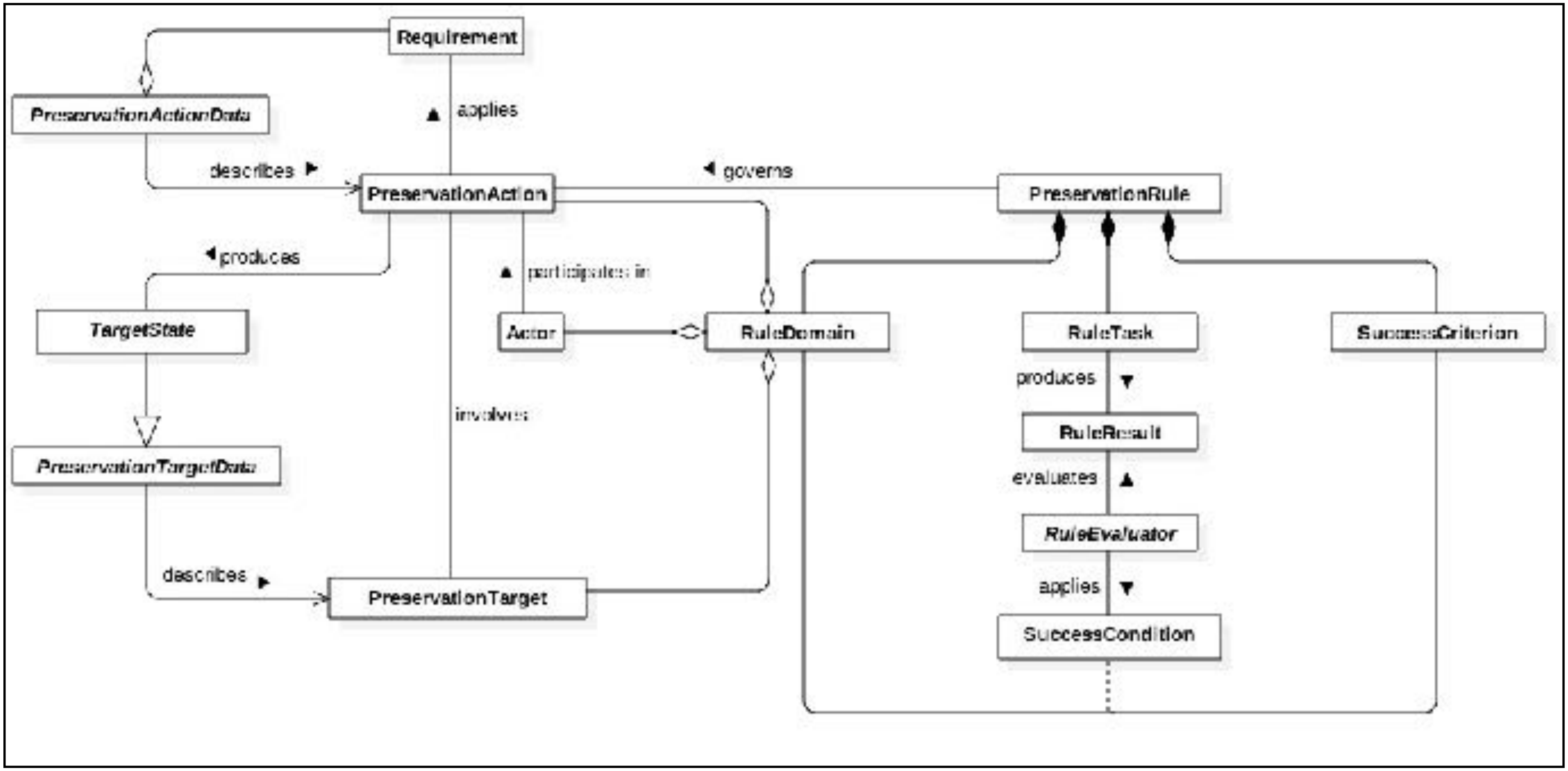


Figure 21. Preservation Actions

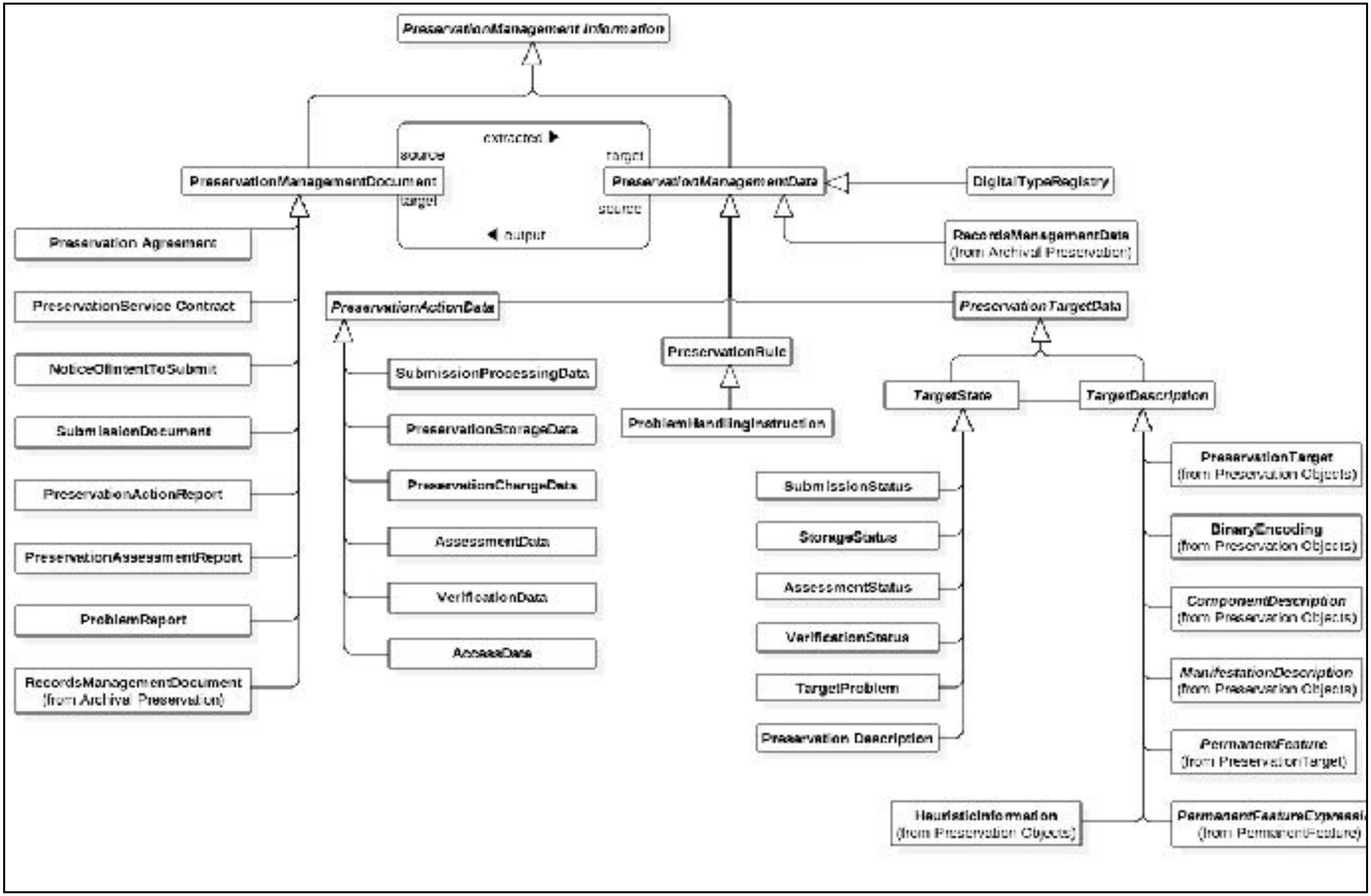


Figure 22, Preservation Management Information