Modeling Organizational Preservation Goals to Guide Digital Preservation

Angela Dappert, Adam Farquhar

The British Library
Boston Spa, Wetherby, West Yorkshire LS23 7BQ, UK
angela.dappert@bl.uk, adam.farquhar@bl.uk

Abstract

Digital preservation activities can only succeed if they go beyond the technical properties of digital objects. They must consider the strategy, policy, goals, and constraints of the institution that undertakes them and take into account the cultural and institutional framework in which data, documents and records are preserved. Furthermore, because organizations differ in many ways, a one-size-fits-all approach cannot be appropriate.

Fortunately, organizations involved in digital preservation have created documents describing their policies, strategies, workflows, plans, and goals to provide guidance. They also have skilled staff who are aware of sometimes unwritten considerations.

Within Planets [Farquhar 2007], a four-year project co-funded by the European Union to address core digital preservation challenges, we have analyzed preservation guiding documents and interviewed staff from libraries, archives, and data centers that are actively engaged in digital preservation. This paper introduces a conceptual model for expressing the core concepts and requirements that appear in preservation guiding documents. It defines a specific vocabulary that institutions can reuse for expressing their own policies and strategies. In addition to providing a conceptual framework, the model and vocabulary support automated preservation planning tools through an XML representation.

Introduction

This paper introduces a conceptual model and vocabulary for preservation guiding documents. Preservation guiding documents include documents, in a broad sense, which specify requirements that make the institution's values or constraints explicit and influence the preservation planning process. They may be policy, strategy, or business documents, applicable legislation, guidelines, rules, or even a choice of temporary runtime parameters. They may be oral representations as well as written representations in databases, source code, web sites, etc..

The model and vocabulary can be shared and exchanged by software applications. They offer a starting point for creating individualized models for an institution. Below, we show how they can be used to describe requirements for individual institutions, possibly, but not necessarily, in a machine-interpretable form. Furthermore, we show how these requirements can then be used in the context of comprehensive preservation planning.

To perform the analysis, the team used a combination of top-down and bottom-up methods. We examined the literature [e.g. ERPA 2003, Solinet 2008, ALA 2007, JISC 2006, PADI 2008, Cornell 2008, CRL 2008] to create a top-down model from first principles. To complement this, we analyzed actual preservation guiding documents of archives, national libraries, and data centers for their content [e.g. Australia 2002, Hampshire, Georgia 2005, UKDA 2008, Florida 2007], and interviewed decision makers [Dappert 2008] to determine factors that influence their preservation choices. We extracted relevant concepts and vocabulary from the material to populate our model and compiled a list of example requirements. A more detailed description of this work can be found in [Dappert 2008]. Aspects of this model were based on or developed together with ideas in the TNA conceptual model which underlies PRONOM [Sharpe 2006], the PLANETS conceptual model [Sharpe 2008], and the OAIS model [CCSDS 2002].

Context

The context of our conceptual model is the process of preservation planning for a digital collection [Strodl 2006]. The goals of this process are to

- identify which parts of the collection present the greatest risks.
- identify candidate preservation actions that could be taken to mitigate the risks.
- evaluate the candidate preservation actions to determine their potential costs and benefits. The cost includes the cost of executing the action, the cost of needed infrastructure for sustaining the results of the action, and the cost of essential characteristics lost in the action (e.g. loss of authenticity) etc.. The benefits come from mitigating the risks and increase in proportion to value of the object and the severity of the risk. The costs and benefits are not necessarily monetary.
- provide justified recommendations for which actions to execute on which collections.

All of these activities should be based on institutional requirements which extend beyond considering file formats and characteristics of individual digital objects to take into account the goals and limitations of the institution, features of its user community, and the environment in which its users access digital content.

The Core Conceptual Model

The core conceptual model implicitly describes the institution and consists of the components in Figure 1. In summary, any Preservation Object has one or more Environments. Every Environment in which the Preservation Object is embedded consists of one or more Environment Components, such as hardware and software components, the legal system, and other internal and external factors. Environment Components are described through their Characteristics, which are Property / Value pairs.

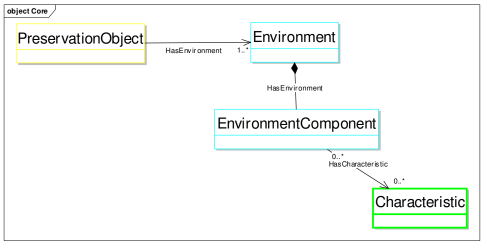

Figure 1 Institutional Data Model

We realized early that requirements express constraints on many levels of granularity. We, therefore, defined **Preservation Objects** as follows:

> A Preservation Object is any object that is directly or indirectly at risk and needs to be digitally preserved.

and introduced the following **Preservation Object Types** as illustrated in Figure 2:

> Collection, Deliverable Unit, Expression, Component, Manifestation, Bytestream.

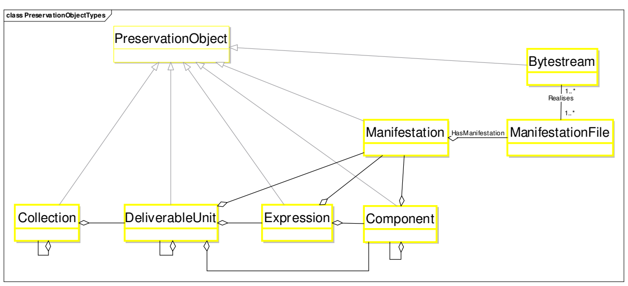

Figure 2 Preservation Object Types

Each Preservation Object Type is related to an other with the "containedIn" relationship (except that a Bytestream is contained in its Manifestation via its Manifestation File).

A **Bytestream** is the primary, physical Preservation Object. If it is at risk of decay or obsolescence it becomes the object of preservation. We create and execute preservation plans to preserve it. A Bytestream is, however, embedded in a larger context.

A **Manifestation** is the collection of all Manifestation Files that are needed to create one rendition of a logical data object. A Bytestream is realised by its **Manifestation File**. Manifestation and Manifestation File are logical descriptions of physical Bytestreams.

Collection, **Deliverable Unit**, **Expression**, and **Component** are logical objects.

In the simplest case, a Bytestream, Manifestation File, and Manifestation have a one-to-one correspondence. For example, a book that is represented as a single PDF file in the PDF format.

In other cases, however, several Bytestreams may be contained in one Manifestation File and several Manifestation Files may be contained in one Manifestation. For example, several image Bytestreams might be contained in a single Manifestation File.

Example:

- A digital file (Bytestream) is part of its Manifestation (e.g. a MPEG-4 video Bytestream is part of an HTML Manifestation of an article).
- This Manifestation represents an Expression of this article, the specific intellectual or artistic form that the article takes as it is realized, which contains a video stream. There may be other Expressions, such as a static still image Expression that holds an image in place of the video stream.
- All Expressions of this article make up the Deliverable Unit. The Deliverable Unit is the abstract concept representing the distinct intellectual creation, which is the article. There might be several Expressions with several Manifestations of the same article (e.g. an HTML, a PDF, an XML, a publisher specific format).
- The article is part of another Deliverable Unit, the issue (hence the recursive link in the diagram).
- And the issue is part of the Deliverable Unit journal, which is the logical object describing all issues of the same title.
- The journal belongs to a Collection. The Collection might be static for the institution, such as the Science Collection, or it might be determined dynamically, such as the Collection of all articles that contain TIFF3.0 files. Collections may contain digital and non-digital objects.
- Collections may be recursively contained in larger Collections.
- Finally, all Collections are part of the whole institution, which is modelled as the top-level Collection.
- Deliverable Units or Expressions consist of logical Components for which Values for Characteristics can be measured or assigned, such as a "table" Component or a "title" Component of a journal article.

Since higher-level objects (such as the Manifestation that includes the affected Bytestream, and the Collection in which this Manifestation is held) are indirectly affected by its preservation need, they also need to be considered during preservation planning. Thus, they are indirectly Preservation Objects. Conversely, an institution can not consider the preservation of each individual data object in isolation. Institutions need to take a global look at all their Collections and resources in order to prioritise their Preservation Actions and co-ordinate preservation activity. In order to facilitate this, the model goes well beyond planning for the individual data object.

Every Preservation Object has one or more **Environments** which may fulfil different roles. For example, a Bytestream or a Manifestation may have creation, ingest, preservation, and access Environments; a Collection may have an internal, a physical delivery, and an online delivery Environment.

Environments for Preservation Objects at a higher level must accommodate the requirements of Preservation Objects at a lower level. As long as a Bytestream is part of its Manifestation, it will live in the Manifestation's Environment. When it is taken out of the Manifestation's Environment, for example to be used in a migration, then the Bytestream's individual Environment requirements will influence the Environment of its new Manifestation.

It is worth noting that it may not be possible to derive the best Environment from a Bytestream's file format. If, for example, a Word file contains only text without formatting, headers and tables, etc., then a .txt output might be considered perfectly adequate, even though this would in general not be considered an ideal migration format for a Word file. Institutions may wish to specify whether an Environment is necessary, recommended, or acceptable.

Every Environment consists of a number of Environment Components. These include the commonly considered software and hardware environments. They also include factors such as the community, legal or budgetary restrictions. **Environment Components** are defined as follows:

> A factor which constrains a Preservation Object and that is necessary to interpret it.

There is a close relationship between an Environment and an extended notion of Representation Information as it is defined in OAIS [CCSDS 2002]. Other examples of extended notions of Representation Information are discussed in [Brown 2008].

The top-level **Environment Component Types** (see Figure 4) include software, hardware, community and Content/Self. The name 'Content/Self' refers to the intellectual content of the Preservation Object. In the case of Preservation Objects which are individual items, the word 'content' or 'intellectual content' provides a good name, but in the case of Preservation Objects which are collective items the word 'self' better reflects the intention. The Content/Self has associated three factors:

- its semantic and syntactic interpretation,
- the format in which it is encoded, and
- its physical realisation.

The Content/Self is actually an Environment Component; several may be associated with a single Preservation Object. They can then be treated like other Environment Components with their associated Characteristics and Values and be used in the preservation planning process in a uniform way. We decompose the OAIS "Digital object" into two aspects: the intellectual content Content/Self and its physical Realisation.

See the full report [Dappert 2008] for additional Environment Component Types that have been extracted from preservation guiding documents. Policy Factors, in particular, are discussed in depth.

Characteristics describe the state of Environment Components as Property / Value pairs. Values may be stored directly as object values, referenced indirectly through registries or in inventories, or extracted dynamically through characterisation processes. The vocabulary for Properties can be found in the full report [Dappert 2008].

The Full Conceptual Model

The full conceptual model which describes the institution embedded in the preservation planning domain consists of the components in Figure 3.

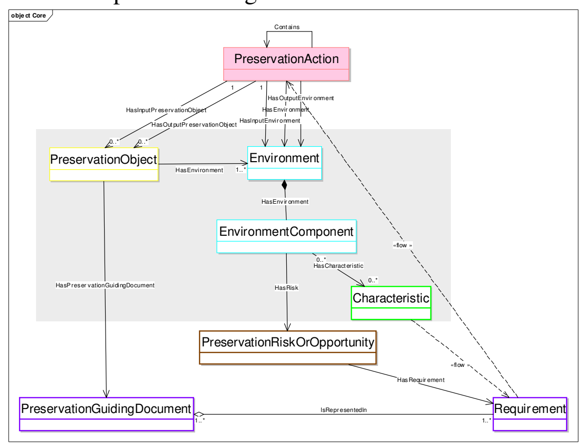

Figure 3 Full Data Model. The Shaded Area Indicates the Core Model.

Degradation of Preservation Objects is caused by two things:

- Preservation Risks
- Executing imperfect, lossy Preservation Actions

Acceptable levels of degradation are defined in an institution's Requirements, which specify permissible or desirable Characteristics of Environment Components. They make the institution's values explicit, influence the preservation process, and are captured in Preservation Guiding Documents.

Changes to an Environment Component, such as obsolescence of hardware or software components, decay of data carriers, or changes to the legal framework may introduce **Preservation Risks**.

An individual institution's Preservation Risks are specified in **Risk Specifying Requirements**. Whenever Characteristics of a Preservation Object's Environment Component violate the Values which are specified in the Requirement then the Preservation Object is considered at risk.

Once a Risk Specifying Requirement is

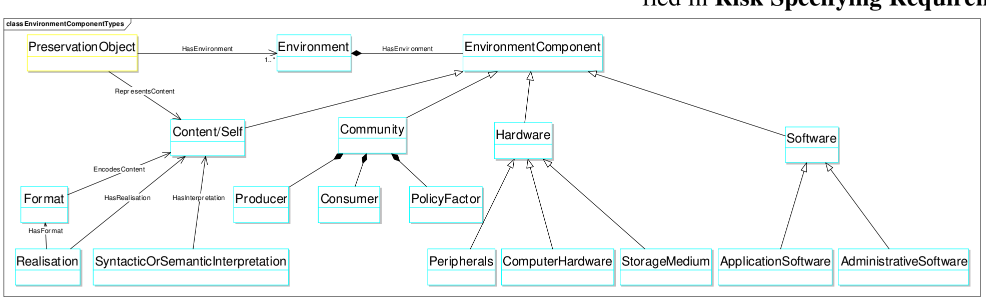

Figure 4 Environment Component Types

violated, a preservation monitor-ring process should notice this and trigger the preservation planning process. It, in turn, determines the optimal Preservation Action to mitigate this risk.

Preservation Object Selecting Requirements are a sub-type of Risk Specifying Requirements which specifies which subset of Preservation Objects is at risk.

A composite **Preservation Action** may consist of elementary Preservation Actions and may include conditional branches and other control-flow constructs.

When a Preservation Action is applied to a Preservation Object and its Environment, it produces a new Preservation Object and/or a new Environment in which the Preservation Risk has been mitigated. Every Preservation Action, therefore, has not only an Input Preservation Object and (at least one) Input Environment, but also an Output Preservation Object and Output Environment. For example, if a Microsoft Word Bytestream is migrated to a PDF Bytestream this results in a new Preservation Object, which might have slightly different Characteristics, but also a new Environment in which it can be used – in this case the platform needs to at least contain a PDF viewer. This approach works for migration, emulation, hardware and other solutions.

For any given Preservation Object and its Environment, there are multiple possible Preservation Actions which might mitigate the Preservation Risk. Which of these Preservation Actions is the most suitable for the Preservation Object can be derived from the information in the **Requirements**.

In order to determine whether an abstract Requirement is applicable and satisfied, one needs to evaluate the concrete Values of the Characteristics of Environment Components which describe the actual Preservation Objects or the concrete Values of a candidate Preservation Action at a given time.

Machine-interpretable Requirements can be expressed in OCL (the Object Constraint Language). They refer solely to concepts and vocabulary contained in the model. Requirements may define the context, pre- and post-conditions, have associated Importance Factors, which specify the importance of the requirement for the institution, as well as a specification of the Operators to be applied to determine whether the requirement is satisfied, and a Tolerance which specifies to what degree deviation from the Requirement can be tolerated.

Requirement Types

During our literature and document analysis, we extracted Requirements that we categorized into the Requirement Types depicted in Figure 5. Besides Risk Specifying Requirements, which were already discussed earlier, there are further Requirement Types.

Preservation Guiding Requirements specify which kinds of Preservation Actions are desirable for the Preservation Object. For example: The size of the Preservation Action's output Preservation Object should not exceed a maximal size as set by the institution. They are dependent on

- which input Characteristics of the Preservation Objects need to be met to consider the Preservation Action.
- which output Characteristics of the Preservation Objects are permissible or desirable (either in absolute terms or in relationship to Characteristics of the input Preservation Object, which might be a derivative or the original submitted to the institution).
- which Characteristics of the Preservation Action itself are desirable.

Action Defining Requirements (sub-type of Preservation Guiding Requirement) define which kinds of Preservation Actions are desirable independent of the Characteristics of the Preservation Object, but dependent only on the Characteristics of the Preservation Action itself. For example PDF may, for a given institution, not be an acceptable preservation output format of a Preservation Action (independent of any input Characteristics of Preservation Objects).

Significant Properties (sub-type of Preservation Guiding Requirement) are often limited to Characteristics of Bytestreams or Components for which it is possible to evaluate Values automatically. Our definition is close to the more expansive one expressed by Andrew Wilson, National Archives of Australia: "the Characteristics of digital objects that must be preserved over time in order to ensure the continued accessibility, usability, and meaning of the objects, and their capacity to be accepted as evidence of what they purport to record." We, too, consider Significant Properties at any level of Preservation Object Type. We, however, treat them as Requirements rather than Characteristics. While Preservation Guiding Requirements in general can combine constraints on multiple Characteristics on several levels of Preservation Object Types, Significant Properties refer to one Characteristic at a time.

Preservation Process Guiding Requirements (sub-type of Preservation Requirement) describe the preservation process itself independent of the Characteristics of the Preservation Object or the Preservation Actions. For example: A preservation planning process should be executed for every data object at least every 5 years,

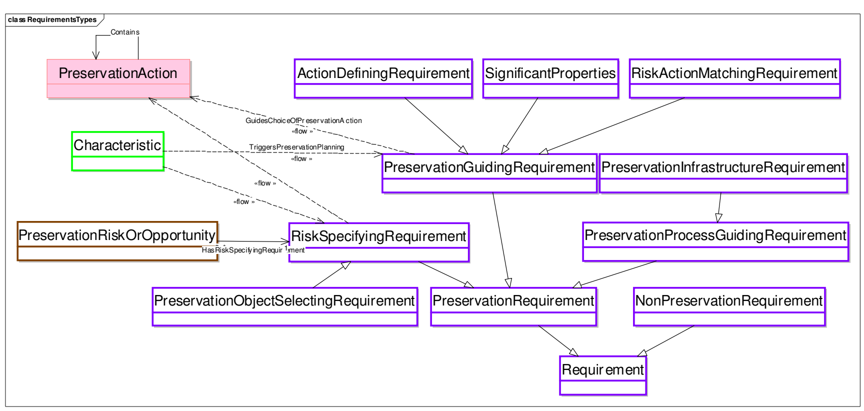

Figure 5 Requirements Types

independent of the Preservation Risks that are established for this data object. These requirements do not influence the preservation planning process.

Preservation Infrastructure Requirements (sub-type of Preservation Process Guiding Requirement) are particularly prominent in preservation guiding documents. They specify required infrastructure Characteristics with respect to security, networking, connectivity, storage, etc.. For example: Mirror versions of on-site systems must be provided.

Non-Preservation Requirements (sub-type of Requirement) specify the set of requirements found which specify processes relevant to preservation, but not part of preservation itself.

Risk / Action Matching Requirements (sub-type of Preservation Guiding Requirement) specify that a candidate Preservation Action has to be an appropriate match to a given Preservation Risk. They are rarely stated explicitly in preservation guiding documents.

Preservation Risk Types are (see Figure 6)

- NewVersion: A new version of the Environment Component is available. This creates a risk of future obsolescence, or a risk of having to support too many versions of this Environment Component.
- NotSupportedOrObsoleteSupport: The Environment Component is no longer sufficiently supported. This creates a risk that support will cease altogether, rendering the Environment Component non-functional.
- DeteriorationOrLoss: The Environment Component is deteriorating or has been lost. Reconstruction or replacement become necessary.
- Proprietary: The Environment Component is proprietary. There is a risk that it cannot be replaced since the specifications for it are unknown.
- UnmanagedGrowth: The institution's Environment is becoming too diverse to manage. A normalization Preservation Action is needed to simplify or unify the Environment.

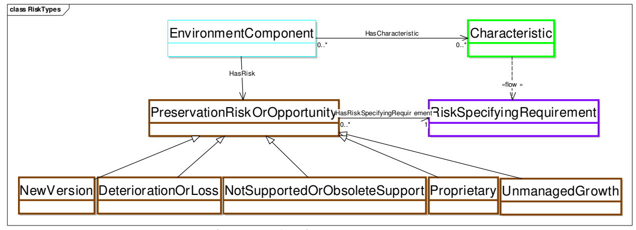

Figure 6 Risk Types

Corresponding to every Preservation Risk Type and the type of the affected Environment Component and Preservation Object, there are appropriate Preservation Actions. For example, the risk of data carrier failure can be mitigated by a carrier refresh. The risk of file format obsolescence can be mitigated by migrating objects to an alternative format.

Preservation Action Types are replacement, repair and reconstruction (See Figure 7).

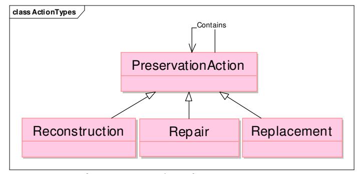

Figure 7 Action Types

The diagram (Figure 8) and table (Figure 9) illustrate the correspondence between Preservation Risk Type, Environment Component Type, Preservation Object Type and Preservation Action Type.

Most of them are self-explanatory. Some deserve some comment:

- Modification of Content/Self might represent an action such as the reconstruction of a deteriorated file, or a file that is modified in order to satisfy new legal requirements.
- One possible Preservation Action is to not do anything (wait and see).

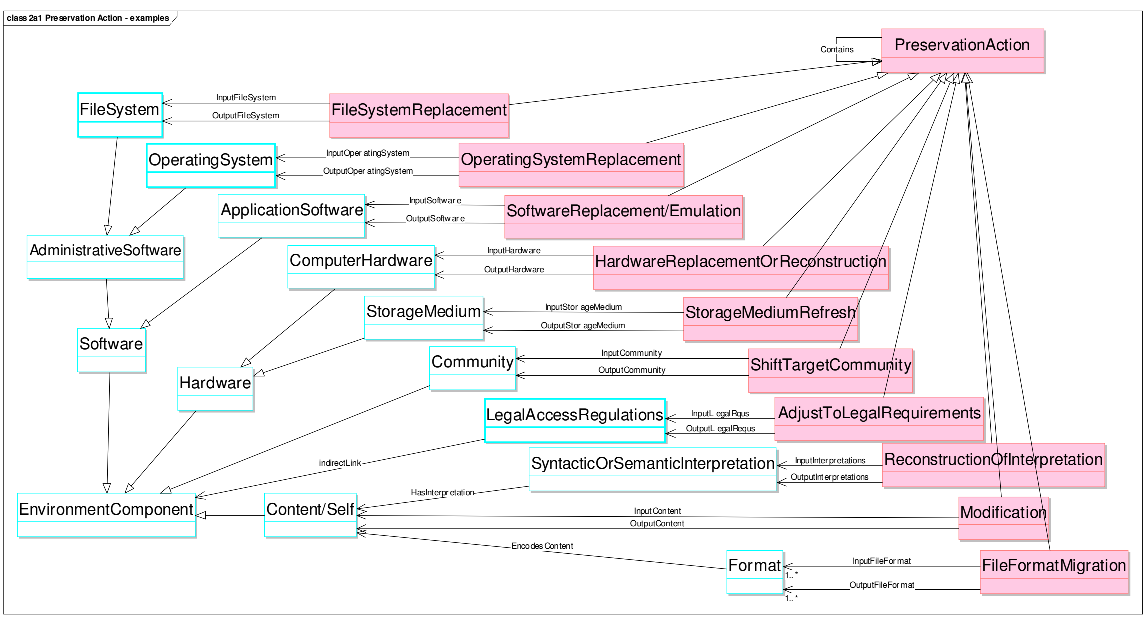

Figure 8 Risk-Action Matching Requirement

Example Risks	Preservation Object Type	Environment Component Type	Preservation Risk Type	Preservation Action Type
Data carriers deteriorate and cannot be read	Bytestream	Data Carrier	Deterioration	Replacement
The data object becomes corrupted on the carrier and the original byte stream cannot be retrieved.	Bytestream	Realization	Deterioration	Reconstruction
Essential hardware components are no longer supported or available	Collection	Hardware	Not supported	Replacement
Software components are proprietary and the dependence is unacceptable to the institution.	Collection	Software	Proprietary	Replacement
The community requires new patterns of access, such as access on a mobile phone, rather than a workstation	Collection	Hardware and Software	Obsolete	Replacement
File formats become obsolete.	Bytestream	Format	Obsolete	Replacement
The legislative framework changes and the data or access to it has to be adapted to the new regulations	Collection	Legislation	New Version	Replacement

Figure 9 Risk-Action Matching Requirement

- Migration does not always imply that a different file format is chosen. For example, a collection might contain PDF files which do not include all of the fonts needed. One might migrate them from PDF (without embedded fonts) to PDF (with all fonts embedded).
- The needs of the target community might be a deciding factor for the choice of Preservation Actions, and, conversely, the choice of Preservation Actions will shape and change the community, just as it changes other Environment Components. Shifting the target community might be a somewhat unintuitive Preservation Action, which is parallel to all other forms of Environment replacement. An example might be turning a research data centre into a history-of-science repository, as the material contained in the collection ceases to live up to contemporary standards of scientific use.
- Community has producers and consumers which may be technical (e.g. repository or IT staff, publishing staff) or content oriented (authors or readers). They may consider a digital object obsolete under different circumstances.

Use to Model Institutional Requirements

The diagram in Figure 10 gives an overview of how the model described in this report can be used to create an institutional preservation guiding document. It introduces the General Model that consists of the concepts and vocabulary that are described in this paper, and the Instantiated Model that an institution might create to reflect its individual state and requirements.

The numbering in the text refers to components in the diagram. Numbering including the letter "a" describes components in the general model. Numbering including the letter "b" describes components in an instantiated model.

(1a) The conceptual model, as discussed in this paper, defines the basic concepts that are needed in the domain of organizational preservation guiding documents and the relationships between them. They comprise Preservation Objects, Environments, Environment Components, Characteristics, Preservation Actions, Risks and Requirements.

(2a) The specific vocabulary defines
- subtypes of the basic concepts,
- properties for all types of Environment Components,
- allowable values for these properties.

It is a representative (i.e. not exhaustive) specific vocabulary.

(3a) The requirements base describes sets of organizational requirements which may be contained in preservation guiding documents. They are expressed solely in terms of the concepts and attributes of our conceptual model and of the specific vocabulary. They may be parameterized so that they can be instantiated to a specific institution's conditions. We plan to represent requirements in OCL.

(4a) The elements in the conceptual model, the specific vocabulary, and the requirements base can be translated into several implementation specific machine-interpretable representations, for example based on an XML schema.

(1b) The institution chooses which of these concepts are supported in its setting and are needed by its preservation planning service. Since the conceptual model is very concise, in most cases all of the concepts would be expected to be used.

(2b) The institution chooses which specific vocabulary applies to it. The institution also assigns values to the

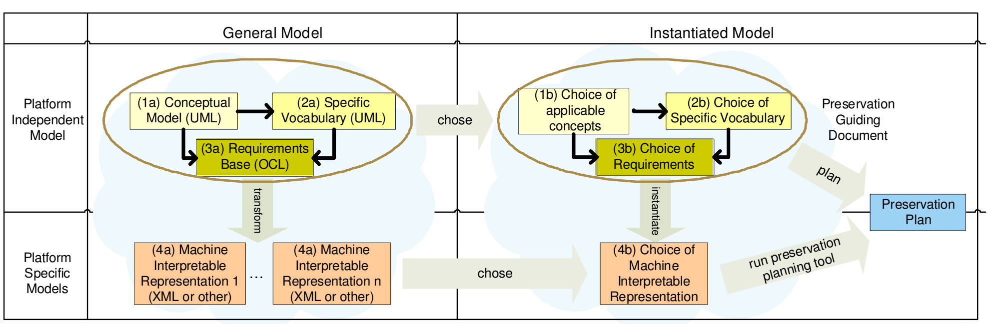

Figure 10 Modelling institutional requrements

Characteristics of its Environment Components if these values will not be measured automatically, or otherwise specifies the method of obtaining measurements or derivations. It will, for example, need registries of tools, formats, and legislative requirements, and need inventories of its collections, software licenses and staff members.

(3b) The institution chooses which Requirements in the Requirements base apply and instantiates them, so that they are now un-parameterized. It specifies Importance Factors, Operators, and Tolerances.

The outputs of steps (1b), (2b) and (3b) form the core part of a preservation guiding document.

(4b) From the choices of steps (1b), (2b), (3b), and the choice of machine-interpretable language results an instantiated machine-interpretable description of the institutional Requirements. This serves as a basis for automated preservation planning. Many requirements in preservation guiding documents, especially on higher institutional levels, may not be machine-interpretable, but it can still be useful to represent the machine-interpretable subset for automatic evaluation.

The planning tool now matches the Requirements in the machine-interpretable version of the preservation guiding document (4b) against the state of the institution to see which Preservation Actions can best satisfy the Requirements under the given state.

Use to Perform Comprehensive Preservation Planning

This model is well-suited for describing any Preservation Object Type and a wide range of preservation processes (e.g., monitoring, planning, characterisation).

First, for example, characterisation tools are defined to work on the Component and Bytestream level. But there are also tools that characterise on a higher level, such as collection profiling tools which analyse Characteristics of a Collection at a given time and produce profiles describing the Collection. They could in principle share the conceptual model and associated processes.

Second, preservation planning needs to compare the Characteristics of a Preservation Object before and after the execution of a candidate Preservation Action in order to evaluate the action against an institution's Requirements. The result is an evaluation score for how suitable each candidate Preservation Action is with respect to the Institution's Requirements. The utility analysis of Plato [Becker 2008] is an example of this.

Preservation Requirements express constraints on all levels of Preservation Objects in the Preservation Object hierarchy (e.g. budgetary constraints on the Collection level; preserving interactivity at the Expression level) and might even mix Characteristics from several levels (e.g. specifying constraints on Collections which contain Bytestreams with a certain Characteristic).

Since each possible Preservation Action may impact multiple levels in the Preservation Object hierarchy, the evaluation of a Preservation Action must be determined on all levels. That is, for every candidate Action, we can evaluate how well it satisfies the Requirements associated with a specific Bytestream, as well as how well it satisfies the Requirements for the whole of its Manifestation, Deliverable Unit, or even Collection.

If for example, a concrete Preservation Action exceeds the Institution's budget, then it need not be considered for a given Bytestream. Equally, if it violates a Collection principle, even though it would be very suitable for preserving a specific Manifestation, it need not be considered. This sort of higher-level constraint is very useful to rule out unsuitable candidate Preservation Actions at a lower level.

Conversely, it is necessary to not just evaluate a concrete Preservation Action's utility in isolation on a lower level, but rather place it in a higher level context. When combining the evaluations from lower levels, with constraints on the higher level, then the evaluation of a Preservation Action might shift in the more global perspective. Planning algorithms need to take this into account.

For example,

- Preservation Action A is considered more suitable than Preservation Action B in the evaluation for a digital file. But if we look now onto a higher level then it might not be possible to combine Preservation Action A with the suggested Preservation Actions for the other files in the Manifestation, which is an inherent Preservation Process Guiding Requirement on Manifestation level. This might, for example, be the case if the Actions' outputs require incompatible environments.
- For a .png file we decide that it is best migrated to a .gif file. When we look at the enclosing Deliverable Unit "web page" we see that the references to the image are broken and that the best Action would now add the Preservation Action "rename the links". When we look at the next higher Deliverable Unit "website" we see that they use java script for their links. The renamed links would not work. The best option is now to use a redirect list for the web server to the image on the server side instead of adding the Preservation Action "rename the links".

It is necessary for the Environment at a higher level to accommodate the Environments required at a lower level. For example, the Manifestation Environment needs to accommodate the Environment for all files in the Manifestation.

Conclusion

This paper introduced a conceptual model and vocabulary for preservation guiding documents. We showed how the model and vocabulary can be used to model requirements for individual institutions, possibly in a machine-interpretable form, and how these requirements can then be used to perform *comprehensive* preservation planning that

- accommodates a full range of preservation planning processes such as monitoring, characterization, comparison of characteristics, and evaluation of candidate preservation actions.
- allows processes to be associated with a full range of entities from institutions, and collections, down to byte-streams and atomic logical components of digital

objects. It is, for example, necessary to refer to characteristics at a lower level to represent requirements at a higher level. For example, in order to specify "collections which contain files that exceed 1 GB", you need to be able to specify the file property "file size" as well as collection properties.

- considers technical as well as organizational properties. Some institutions mandate a particular "technical preservation strategy" (migration, for example) at the preservation policy level, regardless of the lower level technical requirements. This demonstrates the need to integrate institutional and data object considerations in the conceptual model.
- accommodates all types of preservation actions, from software actions (e.g. migration, emulation, file repair), hardware related actions (e.g. data carrier replacement or hardware replacement / reconstruction / repair), to organisational actions (e.g. adapt processes to new legislation, adapt to new requirements of the designated community).

The conceptual model presents a simple but expressive representation of the preservation planning domain. The model and vocabulary can be shared and exchanged by software applications. They offer a convenient starting point for creating individualized models for an institution; this holds true even if the institution does not require a machine-interpretable specification. The model views preservation planning as a process that identifies and mitigates risks to current and future access to digital objects.

This paper represents the current state of our work. We expect to modify and improve it over the coming year in response to feedback and experience applying it in the Planets project.

Acknowledgements

The authors would like to thank colleagues on the Planets project, in particular Bart Ballaux, Michaela Mayr, Rob Sharpe, Sara van Bussel for contributions to this work.

Work presented in this paper is partially supported by the European Community under the Information Society Technologies (IST) Programme of the 6th FP for RTD - Project IST-033789. The authors are solely responsible for the content of this paper.

References

Farquhar, A., and Hockx-Yu, H. *Planets: Integrated services for digital preservation.* Int. Journal of Digital Curation 2, 2 (November 2007), 88–99.

ERPA, Electronic Resource Preservation And Access Network Sept. 2003. *ERPA Guidance: Digital Preservation Policy Tool*. http://www.erpanet.org/ guidance/docs/ERPANETPolicyTool.pdf

Solinet. *Contents of a Digital Preservation Policy*. http://www.solinet.net/Preservation/Resources%20and%20Publications/Contents%20of%20a%20Digital%20Preservation%20Policy.aspx

ALA, American Library Association 2007. *Definitions of Digital Preservation.* ALA Annual Conference, Washington, D.C., June 24, 2007 http://www.ala.org/ala/alcts/newslinks/digipres/PARSdigdef0408.pdf

JISC Digital Preservation and Records Management Programme Nov. 2006. *Digital Preservation briefing paper*. http://www.jisc.ac.uk/publications/publications/pub_digipreservationbp.aspx,

PADI *Preservation Access to Digital Information*. http://www.nla.gov.au/padi/

Cornell University Library 2008. *Digital Preservation Management: Implementing Short-term Strategies for Long-term Problems. Tutorial*. http://www.library. cornell.edu/iris/tutorial/dpm/terminology/strategies.html

CRL, The Center for Research Libraries and OCLC Online Computer Library Center, Inc., Feb 2008. *Trustworthy Repositories Audit & Certification: Criteria and Checklist.* http://www.crl.edu/PDF/trac.pdf

National Archives of Australia, Dec 2002. *An Approach to the Preservation of Digital Records*. http://www.naa.gov.au/Images/An-approach-Green-Paper_tcm2-888.pdf

Hampshire Record Office. *Digital Preservation Policy*. http://www.hants.gov.uk/record-office/policies/ digital.html

Digital Archives of Georgia 2005. *Digital Preservation Policy.* http://sos.georgia.gov/archives/who_are_we/rims/digital_History/policies/policy%20-%20Digital%20Preservation%20Policy.pdf

UKDA, UK Data Archive Mar 2008. *Preservation Policy*

Florida Digital Archive Aug 2007. *Policy Guide*. http://www.fcla.edu/digitalArchive/pdfs/DigitalArchivePolicyGuide.pdf

Dappert, A., Ballaux, B., Mayr, M., van Bussel, S., 2008. *Report on policy and strategy models for libraries, archives and data centres*. PLANETS report PP2-D2. http://www.planets-project.eu/

Sharpe, R . 2006. *SDB Data Model, V1.R2.M0*. Private correspondence 10-Nov-2006

Sharpe, R. 2008. *PLANETS core conceptual model*, Version 1.4. Private correspondence

CCSDS, January 2002. *Reference Model for an Open Archival Information System (OAIS). CCSDS 650.0-B-1, Blue Book* (the full ISO standard). http://public.ccsds.org/publications/archive/650x0b1.pdf

Strodl, S. et alii, 2006. *The DELOS testbed for choosing a digital preservation strategy*. In Proceedings of the 9th International Conference on Asian Digital Libraries (ICADL'06) (Kyoto, Japan, November 27-30 2006), Springer, pp. 323–332.

Brown, A., 2008. *White Paper: Representation Information Registries*. PLANETS report PC/3-D7 (Restricted to programme participants)

Becker, C., Kulovits, H., Rauber, A., Hofman, H. 2008. *Plato: A Service Oriented Decision Support System for Preservation Planning*. JCDL'08, Pittsburgh, Pennsylvania, USA.