DEAL WITH CONFLICT, CAPTURE THE RELATIONSHIP: THE CASE OF DIGITAL OBJECT PROPERTIES

Angela Dappert
The British Library
Boston Spa, Wetherby, West Yorkshire LS23 7BQ, UK

ABSTRACT

Properties of digital objects play a central role in digital preservation. All key preservation services are linked via a common understanding of the properties which describe the digital objects in a repository's care. Unfortunately, different services deal with properties on sometimes different levels of description. While, for example, a preservation characterization service may extract the *fontSize* of a string, the preservation planning service may require the preservation of the text's *formatting*. Additionally, a value for the same property may be obtained in various ways, sometimes resulting in different observed values. Furthermore, properties are not always equally applicable across different file formats.

This report investigates where in these three situations relationships between properties need to be defined to overcome possible misalignments.

The analysis was based on observations gained during a case study of the nature of the properties that are captured in different institutions' preservation requirements and those of use in Planets preservation services.

1. INTRODUCTION

Planets [7] is a four-year project co-funded by the European Union to address core digital preservation challenges. In the Planets project, we have been developing a tool set of digital preservation services. Properties of digital objects play a central role in how these digital preservation services co-operate. All key preservation services are linked via a common understanding of the properties which can be used to capture the description of a digital object in a repository's care [5]. Unfortunately, we observe that different services tend to express the properties at different levels. There is, for example, a gap between the properties extracted by typical tools and the properties that stakeholders use to express their preservation requirements. It also has been observed that values for properties may be obtained in different ways; this may result in different observed values. Additionally, inherent differences between file formats make the comparison of some properties difficult.

In this paper, we analyse preservation plans and preservation services to determine what sorts of properties are expressed. We categorize how their values can be obtained. Each category determines property values in a particular functional or relational way. We illustrate the categories with real-life examples.

This work impacts practitioners, researchers and tool developers. The analysis shows where we can push the boundaries of automation to compute properties. It supports the argument that incomplete, approximate and heuristic values need to be accommodated. It illustrates why there is a need for an expression language for properties to define derived properties. It also illustrates why there is a need for robust aggregate comparisons of digital object property values. Finally, it argues that there is a need to capture the semantics of similar properties.

1.1. Preservation Services that Use Digital Object Properties

Preservation services that use digital properties (see Figure 1) include

- Preservation characterization services, such as the XCL services [16] or JHOVE [1], use file format knowledge to extract property values from digital objects in order to describe them. They may, for example, determine the dimensions of an image file.
- Testbed services, such as the Planets Testbed service [2], derive statistics on the performance of preservation action services, such as those performed by a file format migration tool. They determine to what degree those services preserve properties for representative corpora of digital objects. They, for example, measure the degree to which a service preserves *imageWidth* by evaluating it on many object migrations.
- Preservation monitoring services of the future will determine when a preservation risk for a digital object has arisen and trigger preservation planning.
- Preservation planning services, such as Plato [3], determine which preservation action workflow best preserves the significant characteristics[1] [6] of a sample object set and issue a recommendation of action.
- Preservation action services, such as ImageMagick [10], execute migrations and other preservation actions on specific preservation objects and environments.

[1] In this paper "property" refers to an abstract trait of a digital object, while "characteristic" refers to a property / value pair of a concrete digital object.

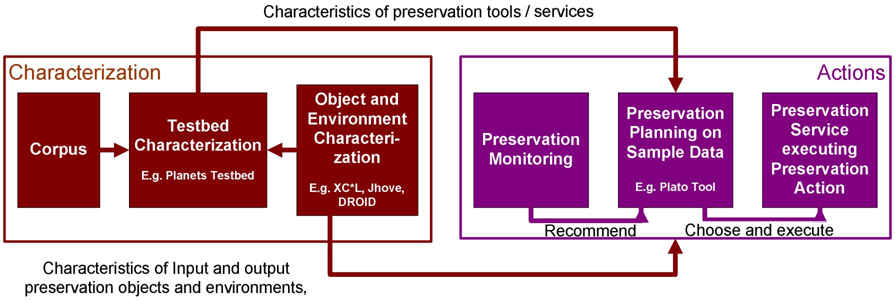
Characteristics of preservation tools / services
Characterization
Corpus
Testbed Characterization
E.g. Planets Testbed
Object and Environment Characterization
E.g. XC*L, Jhove, DROID
Actions
Preservation Monitoring
Preservation Planning on Sample Data
E.g. Plato Tool
Preservation Service executing Preservation Action
Recommend
Choose and execute
Characteristics of Input and output preservation objects and environments,

Figure 1: Digital Preservation Services

1.2. Approaches to Describing Digital Object Properties

In order for these services to work together, they need a common definition of properties. This is necessary in order to refer to properties unambiguously and to ensure interoperability and exchange across not only services, but also systems and institutions. In the preservation community, the definition of digital object properties is currently supported through the following approaches.

- Registries, such as Pronom [17], record properties that are applicable to a given file format, together with data constraints or a controlled vocabulary.
- Preservation metadata dictionaries, such as PREMIS [13], define common preservation metadata elements to describe properties of digital objects or their environments, together with data constraints or a controlled vocabulary, in a file format independent way.
- The InSPECT project [11] identified properties that apply to content types, such as images or emails, rather than to file formats.
- Controlled vocabulary registries, such as the Authorities and Vocabularies service of the Library of Congress [12], capture these properties' permissible values.
- Since related properties are often not immediately comparable, it is useful to develop a properties ontology which captures not only properties of digital objects but also describes them and the relationships between properties explicitly. The Planets Property Ontology is an example. A subset of it, the XCL ontology, is described in [14]. The issues discussed in this report illustrate why such a rich description of digital object properties is needed.

2. POSSIBLE PROPERTY CLASHES ACROSS SERVICES

Different preservation services deal with properties on different levels of description. While, for example, a preservation characterization service may extract the *fontSize* of a string, the preservation planning service may require the preservation of the text's *formatting* in general. These properties may be related in interesting ways and are not comparable through simple equalities.

As a first generation proposal, Heydegger [8] outlines a framework of how property differences between preservation planning and preservation characterization services might be reconciled. This problem deserves generalized development resulting in both theoretical and practical solutions.

2.1. Preservation Services Interactions

Clashes between preservation services may show up in the following situations.

2.1.1. Preservation Planning and Preservation Actions

Stakeholders specify significant characteristics [6] of their preservation objects that need to be preserved (or obtained) through a preservation action. Preservation planning and preservation action services need to determine reliably whether these significant characteristics have been preserved. They request the values for the properties mentioned in the significant characteristics from the preservation characterization service. The characterization service is supposed to deliver the values for these properties in the required way. The preservation planning service additionally requests characteristics that describe the preservation action tools' performance from the testbed service in order to select tools that suit the sample data. These also need to align with the properties expressed in the significant characteristics.

2.1.2. Preservation Monitoring

Policy documents can specify which characteristics of digital objects and their environments manifest a preservation risk. In order to determine whether an object is at risk the monitoring service requests the object's characteristics from the characterization service. The properties used by the two services need to align.

2.1.3. Testbed Experimentation

During a testbed experiment, a preservation action service is tested on a set of digital objects, called a corpus. During the test, derivative objects are created whose property values are compared to the property values of the original objects. The results of this comparison describe the behaviour of a preservation action service based on the degree to which the service preserves the properties' values. There are two possible clashes. Firstly, this result is only meaningful if the testbed tests for a set of properties that are relevant to the users, whose requirements are captured by preservation planning services. Therefore the properties used in preservation planning and those tested in the testbed should align. Secondly, the testbed needs to obtain values of the measured property from preservation characterization services and their properties need to align

Additionally, the testbed needs to aggregate test results that describe tool characteristics (rather than object characteristics) in a way that is most meaningful to their users and write them to a registry ready for use. Preservation planning services weigh those service characteristics to determine the optimal service for the users' specific preservation needs. The properties used by both need to align.

2.1.4. Corpus Design

A corpus is a set of digital objects with known characteristics for use in experiments. In order to compile benchmark corpora on which one can run testbed experiments in a representative way, one has to have an understanding of the applicable and relevant properties. Testbed results are meaningful to preservation planning services only if they are derived on a corpus of digital objects that reflects real life applications and contains instances of all properties that are relevant to users. It is, therefore, important that a corpus covers all properties that might be expressed by users in significant characteristics.

2.1.5. Preservation Action Tool Enhancement

Developers of a migration tool would like to ensure that a digital object after migration with this tool has the same properties as the digital object before migration. To achieve this they specify which property of the source format is to be transformed into which property of the target format. They then migrate sample files and test whether their assessment of property relationships was accurate and whether the migration tool maintained the properties faithfully. The properties of the source and target file format need to align.

They may also ask human subjects to assess the degree of conformance of the target to the source object. The properties that the human subjects apply are not necessarily the properties which where defined by the tool developers. In this case corrections of the property relationships and of the tool are necessary.

2.2. Stakeholders of Digital Object Properties for Preservation Purposes

The stakeholders interested in digital object properties are

- Creators of file formats who need to know how to design file formats so that properties of file formats can be reliably and consistently implemented across supporting applications, can be easily extracted, and validated, and can be migrated to different file format representations without damaging the content.
- Creators and curators of files who need to know which file formats have reliably determinable characteristics.
- Users of files who need to know how well validated a file is after undergoing a preservation action.
- Preservation policy officers and preservation plan developers who need to know which significant characteristics should be specified in their policy documents and validated reliably.
- Migration tool developers who need to know which characteristics to use in order to measure the authenticity delivered by their migration tool.
- Characterization tool developers who need to know how to extract characteristics or infer them from others.
- Testbed, corpora, preservation action and planning services developers, who need to know which properties can be obtained and which are required by users.

3. POSSIBLE PROPERTY CLASHES ACROSS VALUE ORIGINS

During research within the Planets project we observed that the values of digital object properties can be obtained in several ways. This section suggests an initial categorization of their value origin. It shows

- how the value for the same property can be obtained in different ways, possibly resulting in clashing, observed values.
- how different properties can be related to derive one property's value from others. This can help to mitigate the property clashes described in the previous section.

3.1. Value Origins

3.1.1. Extractable, File-Based Value Origins

Category description:
The value origin is a function of the simple digital object: f(object).

The original source of values may be a file, byte-stream or bit-stream. Values are extracted using a tool which implements an algorithm. For effective, scalable preservation, the tool would support *automatic* extraction of properties.

Examples:

- *imageWidth*
- *colourSpace* in PNG and other formats
- *linkURLs* in HTML
- *numberOfAudioChannels*

Derivability:
Algorithms for value extraction are based on file format specifications. This category is implemented for basic file-format-based properties in preservation characterization services, such as the XCL services [16] or JHOVE [1].

3.1.2. Extractable, Complex Value Origins

Category description:

The value origin is a function of a complex digital object and/or the object's environment:

f (object$_1$, …, object$_n$, environment).

These are property values that cannot be taken from the file alone, but rather need to be extracted from

- a representation – that is, the set of files that makes up one complete rendition or execution of a digital object (such as an HTML file with its embedded JPG files).
- a representation including auxiliary files (such as style sheets, non-embedded fonts, java scripts in HTML files, and schema definitions).
- the whole rendering stack (i.e. the preservation object's processing and presentation software and hardware environment).

These properties are not captured in a file format specification alone but are based on the whole environment as depicted in Figure 2.

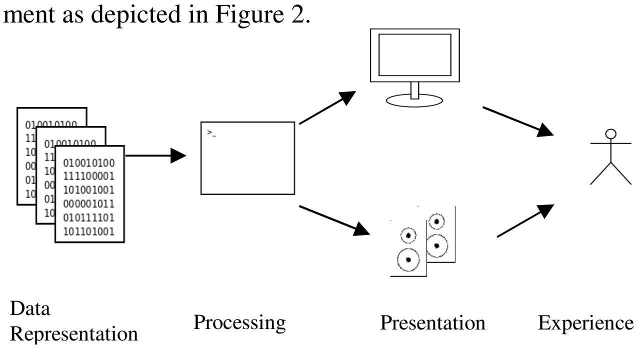

Figure 2. Digital Objects and Their Rendering Stack. (Adapted with permission from Jan Schnasse)

Examples:

- A Microsoft Word document contains a link to a JPG file. One needs to look at both files to infer characteristics about the image's appearance in the document.
- The *colour* of a hyperlink in an HTML file is determined by the accompanying stylesheet. Both files need to be considered to characterize the *colour* of the hyperlinks.
- The presentation of an HTML file depends on browser settings or the choice of browser. Characteristics will vary depending on configuration.
- The actual layout of a Microsoft Word document on paper depends on the printer driver.
- *imageWidth* can be obtained from the rendering software, e.g. Adobe Photoshop.
- *fileSize*, since it depends on the operating system, is derived by asking the file system, rather than counting the actual bytes.

Derivability:

This is a generalization of characterizing one file at a time without regard to its environment. Once we include multiple files and environments into our scope, we expand the set of automatically extractable properties. This category could be implemented now. Some very useful information can be extracted easily; but some with, sometimes, considerable effort.

3.1.3. Non-Extractable, Complex Value Origins

Category description:

The value origin is a function that approximates the property's value

f" (complex object, environment) ≈ f (complex object, environment).

These are properties that are too complex to capture reliably in an algorithmic way, but they can be approximated by related metrics.

Examples:

- The stakeholders' observation of *imageQuality* does not always align with existing image quality metrics. But it is possible to define an acceptable metric which can be measured and compared [9].
- Different parameter configurations of *frequencies*, *amplitudes* and *modulations* can produce comparable sound to the human ear. Even if the representations are not identical, they can have an identical effect for the user. In this case, the property *perceivedSound* is an approximate metric which maps the measurable sound properties onto it.
- Pixel-wise different images may have the same effect on the human eye or rendering devices, since some differences cannot be perceived or rendered.

Multiple metrics can be created to define which combinations are perceived as the same *imageQuality*, *sound* or *colour*, respectively.

Derivability:

By definition, these characteristics cannot be inferred from extractable characteristics unless an algorithmically supported metric is developed. This category can be implemented now, but with, sometimes, considerable effort for development of the algorithmically supported metrics.

3.1.4. Implicit Semantics Value Origins

Category description:

The value origin is a heuristic that results in a value, as well as a confidence measure. The value and confidence measure are repeatable and always give the same results.

(f ' (complex object, environment, heuristic),
conf (complex object, environment, heuristic))

These are properties that require interpretation of semantics that is not captured in the preservation object and its environment. This can, for example, be achieved by employing knowledge-based heuristics.

Examples:

- Some CAD drawings of pipes only specify where pipes are, but not how they are connected. The connections may be clear to the user, but difficult to extract from the object and its environment.
- Older PDF formats do not have structural components such as titles, abstracts, footers. Even in newer PDF formats, functions supporting structural components

are currently rarely used in practice during the document creation process. They can, therefore, not be reliably automatically identified.

Derivability:
Implicit semantics require knowledge-based reasoning to infer property values. The property values in this category can be determined reliably and repeatably, but with considerable effort.

3.1.5. Inferable Value Origins

Category description:
The value origin is a composite function of other value origins: f(g_1 (object), …, g_n (object)).

These are properties that are not explicitly captured in the file format, but can be inferred from other properties. Values may be inherited in an object or property hierarchy, derived through a function from other values, or logically inferred.

This can also be used to relate properties that have synonymous names, by explicitly stating their equivalence.

Examples:

- *aspectRatio* of an image may be calculated as *imageWidth* / *imageHeight*
- *colourFidelity* can be measured from either of two different functions: *averageColour* or *histogramShape*
- *wordCount* can be measured in several ways: e.g., count hyphenated words as one or as multiple words
- *resolutionInPPI* can be mapped via its data type to *resolutionInLinesPerMillimeter*
- *imageWidth* of an image, used as property in one file format, may be inferred from the property *width*, used in another file format, by stating its equivalence with *width.*
- *bitDepth*, is described as one non-negative number in PNG and as three non-negative numbers (one per colour channel) in TIFF. Even though the property is the same in both cases, they have different data types for their values. This can in many cases be expressed through a functional relationship with which one can be derived from the other.

Derivability:
Algorithms for the value inference need to be defined. Even though this category can be implemented now, it has not widely been done. The property values in this category can be determined reliably and repeatably.
The specification of how the involved properties are related can be used to resolve clashes in levels of granularity between preservation services as discussed in Section 2.

3.1.6. Non-Predictable Value Origins

Category description:
The property value is always the same, but the observed value can be different at different times, for example due to interpretation.
f (complex object, environment, interpretation)
These are characteristics that possibly have different values when evaluated by different mechanisms (e.g. different people or the same person at different times).

Examples:

- *colourVibrance* can be judged differently by different observers.

Derivability:
The property values in this category can, by definition, not be reliably inferred.
For testbed purposes, the statistical average of these properties may well be determinable (See for example the Mean Observer Score metric [15].) But for the individual digital object, these techniques can not be applied.

3.1.7. Time Varying Value Origins

Category description:
The property value is different at different times, depending on environmental changes. The observed value, therefore, can be different at different times.
f (complex object, environment, time)
These are properties whose characteristics cannot be reliably reproduced because of time varying behaviour / value change over time.

Examples:

- A time varying sequence of images in an HTML table cell, such as flashing advertisements, will result in different extracted images at different times.

Derivability:
The property values in this category can, by definition, not necessarily be repeatably inferred.

3.1.8. Indeterminable Value Origins

Category description:
The value can not be observed because the digital object is corrupted or the required knowledge is incomplete. In this situation, property values are not measurable at the time because you lack information.

Examples:

- An old Cyrillic font that is used in a document is not available on our machine configuration. An interesting discussion of this can be found in [4].

Derivability:
The property values in this category can, by definition, not be determined.

3.2. Property Categories that Are Independent of Digital Objects

There are additional property types that are independent of digital objects, but they still affect preservation services.

3.2.1. Representation Independent Properties

There are preservation properties that are independent of the file, representation or rendering stack.

There may, for example, be a requirement

> "If a preservation action is chosen, it must be either a *migration* or a *data refresh*. Other preservation action types are not supported."

This requirement guides the preservation plan by specifying the property *preservationActionType*, but does not refer to properties which could be extracted from digital objects.

3.2.2. User Experience Properties

Different users experience (see Figure 2) the same performance[1] of a digital object differently. E.g. somebody who participated in a competition will perceive images documenting the event different from somebody who was not involved or who does not understand the rules underlying the competition. Properties that describe the stakeholder's experience rather than the system's performance – those that relate to the psychological effect of object characteristics on a stakeholder - were not investigated within the Planets project.

This category is different from the *Non-Predictable Value Origins* category discussed in Section 3.1.6, since it considers emotional impact rather than how the value is obtained.

3.3. A Property Can Have Several Origins for a Value

If there are multiple ways of obtaining its value, a property can belong to several of the categories described in this section. E.g. *imageWidth* can be extracted from a file (category *Extractable, File-Based Value Origin*), calculated from other properties, such as *resolution* and *pixelCount* (category *Inferable Value Origin*), obtained from the rendering software (category *Extractable, Complex Value Origin*), or measured by hand from a printed sheet (category *Non-Predictable Value Origin*). *authorName* can be extracted from XML mark-up, HTML headers, MS Windows file properties, etc. (category *Extractable, File-Based Value Origin*) or entered by hand (category *Non-Predictable Value Origin*). *lineLength* can be extracted from a vector graphic (category *Extractable, File-Based Value Origin*) or calculated through heuristic algorithms based on a raster representation of the line (category *Implicit Semantics Value Origin*).

Whenever there are multiple origins for the value of a property there is a risk that there is a clash of the observed values and that they, therefore, represent a related rather than an identical property.

One important task of a property ontology is to capture those origins and their relationships.

3.4. Manually vs. Automatically Extracted Properties

Values for properties can be obtained automatically or manually. Much research has gone into automatically extractable properties. For large volumes of objects, manual declaration of property values by means of free format texts is unworkable. Unfortunately, it is evident that a large set of properties that users require can be extracted automatically only with great difficulty or not reliably. There is a justified desire, where possible, to capture relationships such that most characteristics can be automatically inferred from automatically extractable characteristics. However, as the *imageWidth* and *authorName* examples illustrate, whether or not a property is obtained automatically is an orthogonal issue to our discussion.

3.5. Resolving Property Clash

Property ontologies have to deal with the semantics of similar properties so that they can be compared or derived from each other. This can be used to overcome the clashes between different preservation services that were observed in Section 2. From the preceding analysis, we observe that properties that are related to each other functionally (e.g. through a value origin definition in the *Inferrable Value Origins* category), can be related to each other through this definition within or across preservation services.

In all situations of clash, properties that are derived through non-repeatable value origins (e.g. through a value origin definition in *Non-Predictable* and *Time-Varying Value Origins* categories), cannot reliably be compared to other properties through simple equality metrics. They may be assessed with complex comparison metrics.

Properties that are non-determinable, e.g. in the *Indeterminable Value Origins* category, cannot be compared to others.

4. POSSIBLE PROPERTY CLASHES ACROSS FILE FORMATS

A key task of many preservation services is to compare properties of a digital object before and after a preservation action, such as a migration, in order to assess the quality of the preservation action. This may be hard to do due to incompatible file formats. This section discusses the reasons for this.

4.1. Properties for Different File Format Paradigms

4.1.1. Various Primary Components and Content Structures

Some related properties are hard to compare across file formats because those formats are represented in fundamentally different paradigms. Each file format has pri-

[1] rendering or execution

mary components[1]. Properties apply to those components and are used to characterize a digital object of this file format. For example, a substring component of a text document can be described by the *fontType*, *fontColour*, and *fontSize* properties. When file format paradigms use different types of primary components, properties may not be easy to compare.

For example, both a Word document and a PDF document may represent the same text, but their underlying paradigms are quite different. PDF documents' primary components are representation elements, such as elements of the page layout. Their properties describe a fixed-layout 2D[2] document with an underlying page orientation. Word documents' primary components are content elements, such as text strings, columns, or titles. Their properties describe them mostly independent of the page layout; for example, Microsoft Word has no notion of the page co-ordinate points where a paragraph starts. This results in a phenomenon where seemingly identical properties can actually refer to quite different properties. For example, the property *pageNumber* in Microsoft Word is determined by the author of the document. It may start with page numbering of a title page, or start after an introduction to the document. The PDF document displays page numbers starting with the first physical page. Even though it may display a different logical page number, it has no "awareness" of it.

Likewise, both vector graphics and raster graphics capture images. But while vector graphics describe the properties of content elements of the image (such as the *width*, *length* and *colour* of a line, or the *diameter* and *position* of circle), a raster image would represent the same content by recording properties of its representation elements, the pixels of the image. Raster image formats have no notion of properties of lines and angles; vector graphics formats have no notion of pixel properties.

Even though both the Open Document Format for Office Applications (ODF) and Office Open XML (OOXML) have content elements as primary components, their properties are not necessarily directly comparable because they use different models of how the text is structured. ODF uses a hierarchical content element decomposition into chapter, section, paragraph, marked up text, etc.. Properties apply to those structures. OOXML, however, applies its properties to runs of consistent mark-up which can span structural elements, for example, mark text as *bold* across paragraphs. In this case, one needs to not only capture the relationship between the properties, but also the relationship of the clashing structural elements.

Furthermore properties may cross content types, such as *image* or *text*. Font properties, for example, may cross text and image paradigms. Properties of fonts that are encoded as images cannot be easily compared to those of fonts that are encoded as characters.

4.1.2. Properties Describing Absolute and Relative Page Layout

In addition to differing primary components, file formats fundamentally differ by whether they have absolute vs. relative page layout. Of the example formats in this section, the image and PDF formats describe the absolute position of their content or representation elements, while Word and ODF documents describe the relative position of their content elements. Any properties describing positions on a page or positions of components relative to each other are hard to capture in their non-native representations.

4.1.3. Crossing File Format Paradigms

Which properties are easily extractable depends on the paradigm and primary components used. If one works within the paradigm of raster images, then pixel properties are easily extractable. From this perspective vector graphic elements are not easily extractable, and can, at best, be heuristically approximated. If one works within the paradigm of vector images, then graphic elements are the primary components with measurable properties. From this perspective, raster image pixel properties are not measurable.

Due to the inherent conceptual distance, shifting from one file format paradigm to another results in inaccuracies which make a reliable comparison based on properties hard. For example, one can convert a vector graphic into a raster image in order to compare it with another raster image to infer their similarities or differences. But the conversion algorithm does not necessarily produce a raster image that has pixel-wise equivalence to another raster image of the same content. This means that comparison metrics need to be developed that can anticipate the resulting inaccuracies while still capturing actual content differences.

4.2. Different Scope of Functionality of File Formats

Different file formats support different functionality. For example, OOXML has editing sessions, for which it records a modification and editing history. This functionality is not supported by some other file formats. It is therefore hard to compare properties relating to this differing functionality across file formats.

5. DISCUSSION AND CONCLUSION

This report investigates where in the preservation process interesting relationships between digital object properties occur that are not straight-forward to resolve. A property ontology is a way of modelling them explicitly in order to overcome possible misalignments.

The report suggests a categorization of how properties are obtained and discusses which of them can be used to resolve property clashes.

[1] The description language XCDL [16] calls them “norm elements”.

[2] In the common 2D versions

This work impacts practitioners, researchers and tool developers. The analysis shows where we can push the boundaries of automation to compute properties. It supports the argument that incomplete, approximate and heuristic values need to be accommodated. It illustrates why there is a need for an expression language for properties to define derived properties. It also illustrates why there is a need for robust aggregate comparisons of digital object property values. And it, finally, argues that there is a need to capture the semantics of similar properties.

From it we can develop a research roadmap into digital object properties for digital preservation tasks.

6. ACKNOWLEDGEMENTS

Work presented in this paper was carried out as part of the Planets project (IST-033789, http://www.planets-project.eu/) under the IST Programme of the European Sixth Framework Programme. The author is solely responsible for the content of this paper.

7. REFERENCES

[1] Abrams, S., Morrissey, S., Cramer, T. "“What? So What”: The Next-Generation JHOVE2 Architecture for Format-Aware Characterization." *The International Journal of Digital Curation*, Issue 3, Volume 4 | 2009. http://www.ijdc.net/index.php/ijdc/article/viewFile/139/174

[2] Aitken, B. "The Planets Testbed: Science for Digital Preservation" *The Code4Lib Journal*, ISSN 1940-5758, Issue 3, June 2008. http://journal.code4lib.org/articles/83

[3] Becker, C., Kulovits, H., Guttenbrunner, M., Strodl, D., Rauber, A., Hofman, H. "Systematic planning for digital preservation: evaluating potential strategies and building preservation plans" *International Journal on Digital Libraries (IJDL)*, December 2009. http://www.ifs.tuwien.ac.at/~becker/pubs/becker-ijdl2009.pdf

[4] Brown, G., Woods, K. "Born Broken: Fonts and Information Loss in Legacy Digital Documents" *Proceedings of the 6th International Conference on Preservation of Digital Objects. iPres 2009*, 2009 http://www.cs.indiana.edu/~kamwoods/BrownWoodsiPRES09_Final.pdf

[5] Dappert, A., Farquhar, A. "Implementing Metadata that Guides Digital Preservation Services" *Proceedings of the 6th International Conference on Preservation of Digital Objects. iPres 2009,* 2009. http://www.planets-project.eu/docs/papers/Dappert_MetadataAndPreservationServices_iPres2009.pdf

[6] Dappert, A., Farquhar, A. "Significance is in the Eye of the Stakeholder" *European Conference on Digital Libraries (ECDL)*, September/October 2009, In: M. Agosti et al. (Eds.): ECDL 2009, LNCS 5714, pp. 297-308, 2009, © Springer-Verlag Berlin Heidelberg 2009 http://planets-project.eu/docs/papers/Dappert_Significant_Characteristics_ECDL2009.pdf

[7] Farquhar, A., and Hockx-Yu, H. "Planets: Integrated services for digital preservation" *Int. Journal of Digital Curation* 2, 2 (November 2007), 88–99 http://www.ijdc.net/index.php/ijdc/article/viewFile/45/31

[8] Heydegger, V., Becker, C. "Specification of basic metric and evaluation framework" Planets Project external deliverable PP5/D1. http://planetarium.hki.uni-koeln.de/planets_cms/sites/default/files/Planets_PP5-D1_SpecBasicMetric_Ext.pdf

[9] Heydegger, V. "Just One Bit in a Million: On the Effects of Data Corruption in Files" *European Conference on Digital Libraries (ECDL)*, September/October 2009, In: M. Agosti et al. (Eds.): ECDL 2009, LNCS 5714, pp. 315-326, 2009, © Springer-Verlag Berlin Heidelberg 2009

[10] Image Magick http://www.imagemagick.org/script/index.php

[11] Knight, G., Pennock, M. “Data Without Meaning: Establishing the Significant Properties of Digital Research” *The International Journal of Digital Curation*, Issue 1, Volume 4 | 2009 http://www.ijdc.net/index.php/ijdc/article/viewFile/110/87

[12] Library of Congress. "Authorities and Vocabularies" (nd). http://id.loc.gov/authorities/about.html

[13] PREMIS Editorial Committee "PREMIS Data Dictionary for Preservation Metadata, Version 2." March 2008. http://www.loc.gov/standards/premis/v2/premis-2-0.pdf

[14] Puhl, J. et alii "eXtensible Characterisation Language Suite" Chapter 4. Planets Report PC/2-D12; PC/2-D13; PC/4-D7. http://planetarium.hki.uni-koeln.de/planets_cms/sites/default/files/PC2D12D13PC4D7-01.pdf

[15] Reckwerdt, B. "Quantitative Picture Quality Assessment Tools." http://www.videoclarity.com/WPUnderstandingJNDDMOSPSNR.html

[16] Thaller, M. *The eXtensible Characterisation Languages – XCL.* Verlag Dr. Kovač, Hamburg, 2009.

[17] The National Archives: PRONOM http://www.nationalarchives.gov.uk/pronom/