Designing for Inconsistency – The Dependency-based PERICLES Approach

Jean-Yves Vion-Dury[1], Nikolaos Lagos[1], Efstratios Kontopoulos[2], Marina Riga[2], Panagiotis Mitzias[2], Georgios Meditskos[2], Simon Waddington[3], Pip Laurenson[4], Ioannis Kompatsiaris[2]

[1] Xerox Research Centre Europe (XRCE), 38240 Meylan, France
{Jean-Yves.Vion-Dury, Nikolaos.Lagos}@xrce.xerox.com

[2] Information Technologies Institute, CERTH, GR-57001 Thessaloniki, Greece
{skontopo, mriga, pmitzias, gmeditsk, ikom}@iti.gr

[3] King's College London, UK
simon.waddington@kcl.ac.uk

[4] Tate, London, UK
Pip.Laurenson@tate.org.uk

Abstract. The rise of the Semantic Web has provided cultural heritage researchers and practitioners with several tools for ensuring semantic-rich representations and interoperability of cultural heritage collections. Although indeed offering a lot of advantages, these tools, which come mostly in the form of ontologies and related vocabularies, do not provide a conceptual model for capturing contextual and environmental dependencies contributing to long-term digital preservation. This paper presents one of the key outcomes of the PERICLES FP7 project, the Linked Resource Model, for modelling dependencies as a set of evolving linked resources. The proposed model is evaluated via a domain-specific representation involving digital video art.

Keywords: digital preservation, LRM, ontology, digital video art, dependency.

1 Introduction

With the advent of the *Semantic Web*, Cultural Heritage (CH) researchers and practitioners have been gradually adopting the respective approaches and technologies in order to ensure the *semantic interoperability* between physical artefacts and intangible attributes. Work in this direction is mostly revolving around deploying relevant established ontologies and vocabularies. *CIDOC CRM* [3], acknowledged as an ISO standard (21127:2006), is arguably the most popular ontology for representing concepts and information in CH and museum documentation. Other similar resources include *Europeana*[1], a multilingual digital library for facilitating user access to an integrated

[1] www.europeana.eu/portal/

content for European cultural and scientific heritage, and the *Getty vocabularies*[2] that provide structured terminology for works of art, architecture, material, culture, as well as artists, architects and geolocations.

Although the utility of the above vocabularies is indisputable in capturing descriptive metadata, they do not provide the conceptual model for capturing contextual and environmental dependencies that contribute significantly in *long-term digital preservation*, which refers to the process of adopting a series of managed activities necessary to ensure continued access to and re-use of digital materials for as long as needed [2]. This is the aim of *PERICLES*[3], a four-year integrated project focusing on representing and evaluating the risks for long-term digital preservation of digital resources. The project treats the environments in which digital objects are created, managed and used, holistically as *digital ecosystems*, rather than focusing on individual digital objects. The first aspect of this approach involves developing a domain-independent ontology, the *Linked Resource Model* (*LRM*), to model dependencies as a set of evolving linked resources. The second aspect is *preservation by design* that relies on capturing relevant context and environment of a digital object at source. The approach is *model-driven* and considers digital objects as generated and existing within an evolving continuum.

PERICLES addresses preservation challenges in two domains: (a) digital artworks (e.g. digital video artworks and software-based artworks); (b) experimental scientific and associated space operations data. To this end, the project will deliver two preservation prototypes, as well as a portfolio of models, services, tools and research that supports the development of practice related to the notion of preservation ecosystems and life-cycle management. Whilst the project's focus lies in the preservation of both cultural and scientific heritage, this paper is concerned solely with the former domain and presents early results in applying the PERICLES design principles to a specific challenge encountered by those working on the conservation of digital video artworks. In the rest of the paper, Section 2 describes related work for representing dependencies; Section 3 offers a description of the overall use case, followed by concrete descriptions of the LRM and its domain-specific representations in Sections 4 and 5, and the paper is concluded with final remarks and directions for future work.

2 Related Work: Existing Dependency Models

The *PREMIS Data Dictionary* [7] defines three types of relationships between objects: *structural*, *derivation* and *dependency*. From the PERICLES perspective, derivation and dependency relationships are the most relevant. A *derivation relationship* results from the replication or transformation of an object. A *dependency relationship* exists when one object requires another to support its function, delivery, or coherence. Examples include a font, style sheet, DTD or schema that are not part of the file itself. Objects can also be related to events through user-defined dictionaries of terms, and events can in turn be linked to agents that performed those events.

[2] www.getty.edu/research/tools/vocabularies/

[3] www.pericles-project.eu/

The *Open Provenance Model* (*OPM*) [9] introduces the concept of a provenance graph that aims to capture the causal dependencies between entities. The most relevant concept from our perspective is *process* that represents actions performed on or caused by artefacts, and resulting in new artefacts.

In a preservation context, [10] defines notions of *module*, *dependency* and *profile* to model use by a community of users. A *module* is defined as a software/hardware component or knowledge base that is to be preserved, and a *profile* is the set of modules that are assumed to be known. A *dependency relation* is then defined by the statement that module A depends on module B if A cannot function without B. For example, a `readme.txt` file depends on the availability of a text editor.

The authors of [11] also define the more specific notion of task-based dependency, expressed as Datalog rules and facts. For instance, `Compile(HelloWorld.java)` denotes the task of compiling 'HelloWorld.java'. Since the compilability of the latter depends on the availability of a compiler, this dependency can be expressed using a rule of the form: `Compile(X):- Compilable(X,Y)`, where the binary predicate `Compilable(X,Y)` denotes the appropriateness of `Y` for compiling `X`. This more formal approach enables various tasks to be performed, such as risk and gap analysis for specific tasks, possibly considering contextual information, such as user profiles.

In [6] the authors elaborate a more sophisticated approach (as compared to [11]). The original notion of *task* formerly associated with dependencies is now abstracted toward the notion of *intelligibility*, which allows for typing dependencies. The LRM approach goes one step further toward genericity, by allowing any kind of dependency specialization, the intelligibility being replaced by the notion of *intention*, which can be described informally or formally through additional properties. Moreover, LRM offers a much richer topology for dependency graphs through managing dependencies as instances instead of properties. By combining genericity and semantic refinement, we expect a tighter management of consistency criteria, whatever semantics might be potentially involved.

3 Case Study: Video Art and Preservation Challenges

Video as a medium is now a mainstream element of contemporary art practice. 'Video' is a broad term used to refer to a wide array of rapidly changing technologies, including the formats in which an artwork might be made, archived and displayed and the equipment used in its presentation. One of the key issues for artists' video relates to how changes that will impact the appearance of the video are tracked and managed over time. Video artworks typically require the following three main components to be displayed: (a) the media (i.e. the video file), (b) the equipment needed for display in the gallery, and, (c) the installation instructions. In recent years video has moved from tape to file-based delivery and storage.

The lack of consensus, tools and services around the preservation of high-value video within the broader preservation community has created a significant problem for previously well-managed collections. There is usually no retention schedule for works; rather it is assumed that they will be preserved in perpetuity. The planning horizon for heritage institutions is often expressed as 100 or 500 years and sometimes as forever.

As already mentioned in the introduction, within the PERICLES project the primary aim is to focus on modelling the risks related to the long-term digital preservation of digital objects in general, and, amongst them, the media elements of video art objects. One of the scenarios in which we are exploring the benefits of such modelling is ensuring the consistent playback of video files and it is this use case that is presented later in the paper (Section 5). In order to identify and understand the technical variables involved in the consistent playback of digital video files, Tate commissioned a report by Dave Rice [8]. Thus, we are not aiming to model the domain at the artwork level, but rather the specifics of dependencies between digital things within a system which forms part of the artwork. It is therefore a partial model related to the dependencies of some of the components of the artwork.

In this case, modelling enables the conservator to better understand the digital dependencies within the system and also helps identify areas where automation might be achievable. Modelling also facilitates communication with computer scientists and software developers who might provide tools to support activities related to long term digital preservation and the corresponding dependencies. The process of establishing any such model is the identification of the key entities, the relevant types of dependency and change and, finally, the study of scenarios regarding how this information might be used in the assessment of material that comes into the repository.

4 LRM: An Ontology Focusing on Change Management

The *Linked Resource Model* (*LRM*) is an upper level ontology designed to provide a principled way for modelling evolving ecosystems, focusing on aspects related to the changes taking place. This means that, in addition to existing preservation models that aim to ensure that records remain accessible and usable over time (e.g. see [1]), the LRM also aims to model how changes to the ecosystem, and their impact, can be captured. It is important to note here that we assume that a policy governs at all times the dynamic aspects related to changes (e.g. conditions required for a change to happen and/or impact of changes). As a consequence, the properties of the LRM are dependent on the policy being applied; therefore, most of the defined concepts are related to what the policy expects. At its core the LRM defines the ecosystem by means of participating entities and dependencies among them. A set of other properties and specialised entity types are also provided but they are all conditioned on what is allowed/required by the policy. The notion of policy is not further defined here, as it is out of the scope of this work. The main concepts of the static LRM are illustrated in Fig. 1 (the prefix `pk` refers to the LRM namespace) and discussed further below.

`Resource`. Represents any physical, digital, conceptual, or other kind of entity and in general comprises all things in the universe of discourse of the LRM Model[4]. A resource can be *Abstract* (c.f. `AbstractResource` in Fig. 1), representing the abstract

[4] This definition is close to CIDOC CRM's Entity – we are exploring possible mappings.

part of a resource, for instance the idea or concept of an artwork, or *Concrete* (c.f. `ConcreteResource` in Fig. 1), representing the part of an entity that has a physical extension and can therefore be accessed at a specific location (a corresponding attribute called `location` is used to specify spatial information; for instance for a `Digital-Resource`, which represents objects with a digital extension, this information can be the URL required to retrieve and download the corresponding bit stream). The above two concepts can be used together to describe a resource; for example, both the very idea of an artwork, as referred by papers talking about the artist's intention behind the created object, and the corresponding video stream that one can load and play in order to manifest and perceive the artwork. To achieve that, the abstract and concrete resources can be related through a specific `realizedAs` predicate, which in the above example could be used to express that the video file is a concrete realization of the abstract art piece. It should be noted that an abstract resource could be connected to one or several concrete resources; in that case, the concrete resources could be aggregated (a class `AggregatedResource` is defined in the LRM, although not shown in Fig. 1), and `realizedAs` could be used to connect the abstract resource to the aggregated one.

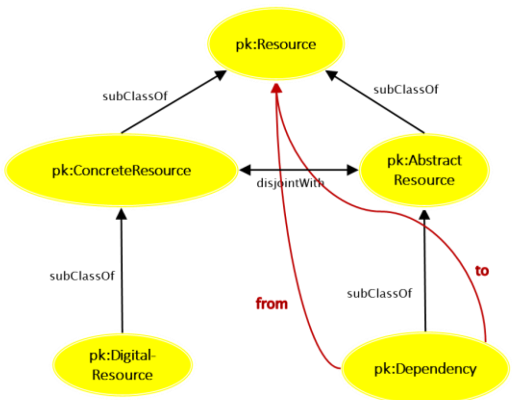

Fig. 1. Main concepts of the static LRM.

`Dependency`. The core concept of the static LRM is that of a *dependency*. An LRM `Dependency` describes the context under which change in one or more entities has an impact on other entities of the ecosystem. The description of a dependency minimally includes the intent or purpose related to the corresponding usage of the involved entities. From a functional perspective, we expect that dedicated policies/rules will further refine the context (e.g. conditions, time constraints, impact) under which change is to be interpreted for a given type of dependency. For instance, consider a document containing a set of diagrams that has been created using MS Visio 2000, and that a corresponding policy defines that MS Visio drawings should be periodically backed up as JPEG objects by the workgroup who created the set of diagrams in the first place[5]. According to the policy, the workgroup who created the set of JPEG objects should be able to access but not edit the corresponding objects. The classes and properties related to the `Dependency` class can be used to describe each such conversion in terms of its temporal information and the entities it involves along with their roles in the relationship (i.e. person making the conversion and object being converted). In addition, the LRM `Dependency` is strictly connected to the intent underlying a specific change. In the case described here the intent may be described as "*The workgroup who created the set of diagrams wants to be able to access* (*but not edit*) *the diagrams created using MS Visio 2000. Therefore, the workgroup has decided to convert these diagrams to JPEG format*" and it implies the following.

- There is an explicit dependency between the MS Visio and JPEG objects. More specifically, the JPEG objects are depending on the MS Visio ones. This means that if an MS Visio object 'MS1' is converted to a JPEG object, 'JPEG1', and 'MS1' is edited, then 'JPEG1' should either be updated accordingly or another JPEG object 'JPEG2' should be generated and 'JPEG1' optionally deleted (the use case is not explicit enough here to decide which of the two actions should be performed). This dependency would be particularly useful in a scenario where MS Visio keeps on being used for some time in parallel to the JPEG entities, which are in turn used for back up purposes.
- The dependency between 'MS1' and 'JPEG1' is unidirectional. Actually, JPEG objects are not allowed to be edited and, if they are, no change to the corresponding MS Visio objects should apply.
- The dependency applies to the specific workgroup, which means that if a person from another workgroup modifies one of the MS Visio objects, no specific conversion action has to be taken (the action should be defined by the corresponding policy).

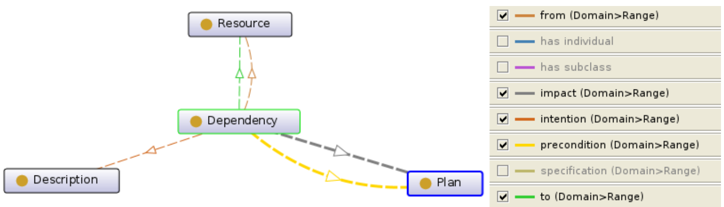

Fig. 2. A view of the `Dependency` concept in LRM.

To enable recording the intent of a dependency, we can relate in the LRM the `Dependency` entity with an entity that describes the intent via a property that we name "`intention`", as illustrated in Fig. 2. In Fig. 2, properties `from` and `to` indicate the directionality of the dependency.

The LRM model provides also concepts that allow recording when a change is triggered and what is the impact of this change on other entities. Let us take once more the above example: we need to be able to express the fact that transformation to

[5] This example is adapted from a use case described in [1], pp. 52-53.

JPEG objects is possible only if the corresponding MS Visio objects exist and if the human that triggers the conversion has the required permissions to do that (i.e. belongs to the specific workgroup). The impact of the conversion could be to generate a new JPEG object or update an existing one. The action to be taken (i.e. generate or update) in that case, would be decided based on the policy governing the specific operation. Assuming that only the most recent JPEG object must be archived, then the old one must be deleted and replaced by the new one (conversely deciding to keep the old JPEG object may imply having to archive the old version of the corresponding old MS Visio object as well).

`Plan`. The condition(s) and impact(s) of a change operation are connected to the `Dependency` concept in LRM via `precondition` and `impact` properties as illustrated in Fig. 2. These connect a `Dependency` to a `Plan`, which represents a set of actions or steps to be executed by someone/something (either human or software). The Plan can be used, thus, as a means of giving operational semantics to dependencies. Plans can describe how preconditions and impacts are checked and implemented (this could be for example defined via a formal rule-based language, such as SWRL). The temporally coordinated execution of plans can be modelled via activities. A corresponding `Activity` class is defined in LRM, which has a temporal extension (i.e. has a start and/or end time, or a duration). Finally, a resource that performs an activity, i.e. is the "bearer" of change in the ecosystem, either human or man-made (e.g. software), is represented by a class called `Agent`[6].

5 Domain-specific Extension: The DVA Ontology

In this work we propose a domain-specific representation for modelling dependencies between different digital entities that impact our ability of preserving digital video art. For the representation of digital entities we re-use several constructs from CIDOC-CRM and CRMdig [4], ensuring, thus, *semantic interoperability* with other ontologies already aligned with these models. On the other hand, the mechanisms for representing dependencies are based on the aforementioned LRM. As already mentioned, we mostly focus on representing and evaluating the risks for long-term digital preservation. The adopted representation approach is presented under the scope of a specific challenge, i.e. sustaining consistent video playback.

As seen in Fig. 3, the digital video, i.e. one of the concrete resources of a digital video artwork, contains: (a) stream(s) for video, audio (optional) and subtitles (optional), (b) a codec, and (c) a container (or wrapper). Relations between these entities and a digital video are represented via properties `has-stream`, `has-container` and `has-codec`, respectively, with further categorization of the latter to `has-video-codec` and `has-audio-codec`.

[6] Classes `Activity` and `Agent` relate to provenance information. We explored potential mappings between LRM and PROV (www.w3.org/TR/prov-o), a widely used resource for representing provenance information, but some PROV constraints (i.e. Activities cannot be Entities) are structurally incompatible with the LRM.

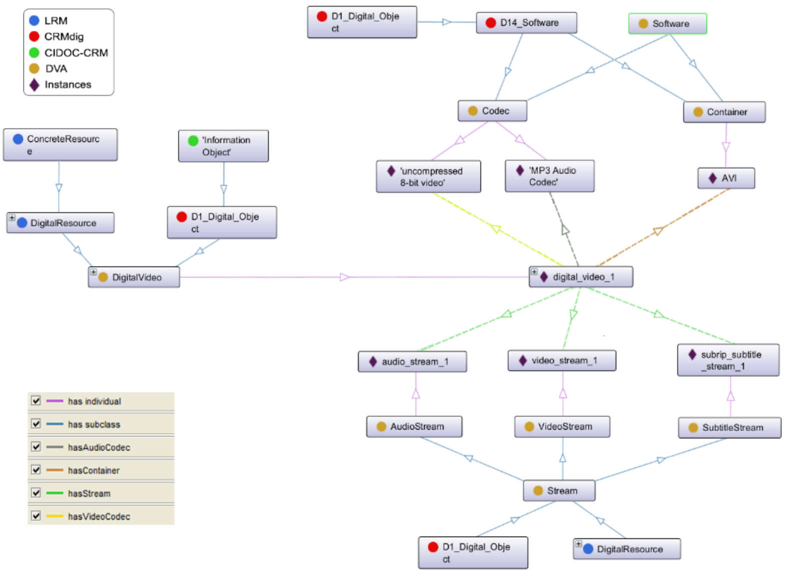

Fig. 3. Representation of a digital video resource, along with its descriptive details.

Additionally, key LRM notions, such as `AggregatedResource` and `Dependency`, have been integrated and fully-adopted in the DVA ontology. As mentioned in Section 4, an abstract resource may be realised as one or more concrete resources; in the latter case, the concrete resources are aggregated into one `AggregatedResource` instance, while the abstract with the aggregated resources are connected via property `realizedAs`. Fig. 4 illustrates such an example, as represented in the DVA ontology.

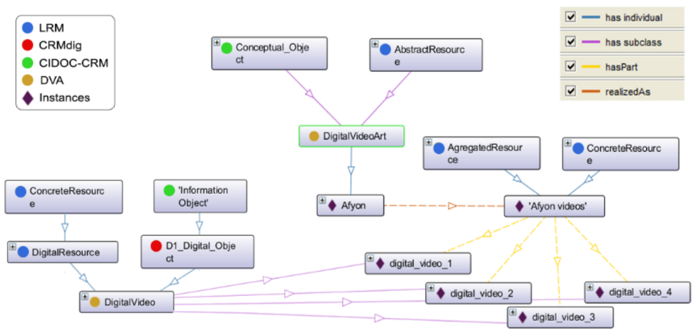

Fig. 4. Example of a digital video artwork with aggregated resource.

The concept of `Dependency` is adopted from LRM to represent relations between digital video artworks and complementary entities (i.e. media players, wrappers and relevant software). A specific challenge concerning the preservation of digital video artworks is to sustain the consistent playback of their video files. In this context, Fig. 5 displays the dependency of the playback activity to the digital video file.

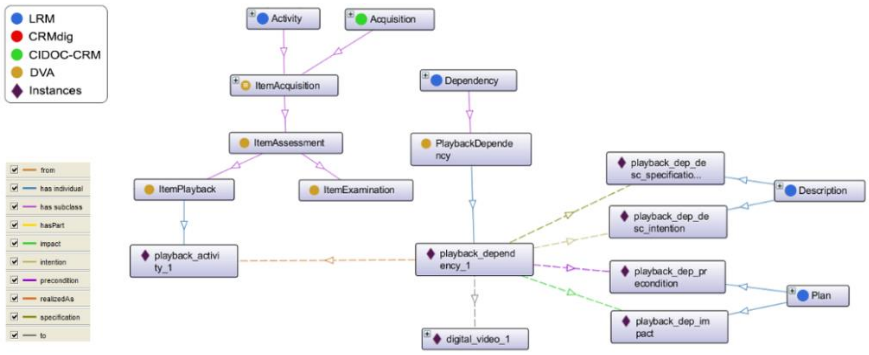

Fig. 5. A view of the playback dependency concept in the DVA ontology.

Furthermore, since a digital video file is associated to a container, playing the container correctly depends on the usage of an appropriate media player. `PlayerDependency` involves the compatibility of media players with certain video containers. Specifically, a video container (e.g. 'AVI') depends on the media players supporting its playback. This classification offers the possibility to spot proper media players for a certain playback activity.

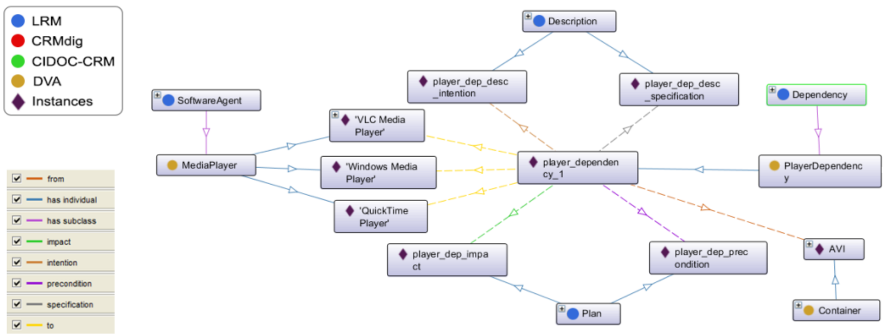

Fig. 6. A view of the player dependency concept in the DVA ontology.

The above representations allow handling various cases, like e.g. finding compatible media players for a digital video or detecting inconsistencies in the video's aspect ratio, colour matric etc. We have implemented this validation layer on top of the DVA ontology as *SPARQL rules* (*SPARQL Inferencing Notation - SPIN* [5]). In SPIN, SPARQL queries can be stored as RDF triples alongside the RDF domain model, enabling the linkage of RDF resources with the associated SPARQL queries, as well as sharing and re-using SPARQL queries. SPIN supports the definition of SPARQL inference rules that can be used to derive new RDF statements from existing ones through iterative rule application, serving as a ready-to-use framework. The presentation of the implemented SPIN constructs is outside the scope of this paper.

6 Conclusions and Future Work

This work presents early results in using a domain-independent ontology, the *Linked Resource Model* (*LRM*), to model and manage change in a cultural heritage setting. Viewing dependencies as complex constructs instead of simple links between resources, allows defining the semantics governing a change in terms of the intention underlying this change, the pre-conditions that should be satisfied to trigger it, and the corresponding resulting impact(s) on the ecosystem itself. We have illustrated via our case study that the LRM can be combined with CIDOC-CRM (and its CRMdig extension for modelling digital resources) and we are planning of exploring tighter integration in the near future. Of interest to us is also extending the LRM ontology in order to represent causality and temporal aspects.

7 Acknowledgements

This work was supported by the European Commission Seventh Framework Programme under Grant Agreement Number FP7-601138 PERICLES.

References

1. Australian Government Recordkeeping Metadata Standard Version 2.0 Implementation Guidelines (2010). Available at: www.naa.gov.au/records-management/publications/agrk-metadata-standard.aspx. Accessed: Apr-15.
2. Digital Preservation Coalition: *Digital Preservation Handbook*. York, UK, 2008. Available online: www.dpconline.org/advice/preservationhandbook. Accessed: Apr-15.
3. Doerr, M.: The CIDOC CRM, an Ontological Approach to Schema Heterogeneity. *Semantic interoperability and integration* 4391, 2005.
4. Doerr, M., Theodoridou, M.: CRMdig: A generic digital provenance model for scientific observation. *TaPP'11, 3rd USENIX Workshop on the Theory and Practice of Provenance*, Heraklion, Crete, Greece, June 20-21, 2011.
5. Knublauch, H., Hendler, J. A., Idehen, K.: SPIN - overview and motivation. World Wide Web Consortium, W3C Member Submission, Feb. 2011.
6. Marketakis, Y., Tzitzikas, Y.: Dependency Management for Digital Preservation Using Semantic Web Technologies. *Int. Journal on Digital Libraries* 10(4): 159-77, 2009.
7. PREMIS Data Dictionary for Preservation Metadata (Official Web Site), The Library of congress, USA. www.loc.gov/standards/premis/. Accessed: Apr-15.
8. Rice, D.: Sustaining Consistent Video Presentation (March 2015). *Tate Research Articles*. Available at: www.tate.org.uk/research/publications/sustaining-consistent-video-presentation. Accessed: Apr-15.
9. The Open Provenance Model Core Specification (v1.1). http://eprints.soton.ac.uk/271449/1/opm.pdf. Accessed: Apr-15.
10. Tzitzikas, Y.: Dependency Management for the Preservation of Digital Information. *Database and Expert Systems Applications*, pp. 582-92. Springer Berlin Heidelberg, 2007.
11. Tzitzikas, Y., Marketakis, Y., Antoniou, G.: Task-Based Dependency Management for the Preservation of Digital Objects Using Rules. *Artificial Intelligence: Theories, Models and Applications*, pp. 265–74. Springer Berlin Heidelberg, 2010.