King's Research Portal

DOI:

[10.1007/s10796-016-9709-z](https://doi.org/10.1007/s10796-016-9709-z)

Document Version

Peer reviewed version

[Link to publication record in King's Research Portal](#)

Citation for published version (APA):

Lagos, N., Riga, M., Mitziias, P., Vion-dury, J., Kontopoulos, E., Waddington, S., Laurenson, P., Meditskos, G., & Kompatsiaris, I. (2016). Dependency modelling for inconsistency management in Digital Preservation – The PERICLES approach. *INFORMATION SYSTEMS FRONTIERS*. <https://doi.org/10.1007/s10796-016-9709-z>

Citing this paper

Please note that where the full-text provided on King's Research Portal is the Author Accepted Manuscript or Post-Print version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version for pagination, volume/issue, and date of publication details. And where the final published version is provided on the Research Portal, if citing you are again advised to check the publisher's website for any subsequent corrections.

General rights

Copyright and moral rights for the publications made accessible in the Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Research Portal

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Dependency Modelling for Inconsistency Management in Digital Preservation – The PERICLES Approach

Nikolaos Lagos¹, Marina Riga², Panagiotis Mitziias², Jean-Yves Vion-Dury¹, Efstratios Kontopoulos², Simon Waddington³, Georgios Meditskos², Pip Laurenson⁴, Ioannis Kompatsiaris²

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, pmitziias, 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 semantically 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 adoption of the proposed model and the consistency of its representation is evaluated via a specific instantiation involving the domain of digital video art.

Keywords: digital preservation, LRM, ontology, digital video art, dependency, inconsistency detection, SPIN rules.

Acknowledgements

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

1 Introduction

With the advent of the *Semantic Web*, Cultural Heritage (CH) researchers and practitioners have been gradually adopting Semantic Web 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* (Doerr 2005), 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 content for European cultural and scientific heritage, and its corresponding Data Model and terminologies, 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 to *long-term digital preservation*. Long-term digital preservation can be understood as 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 (Digital Preservation Coalition 2008). *PERICLES*³ is a four-year integrated project focusing on representing and evaluating the risks for long-term digital preservation of digital resources that also aims to address the need for a conceptual model for capturing these key contextual and environmental dependencies. 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 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*. The second aspect involves developing a domain-independent ontology, the *Linked Resource Model (LRM)*, to model dependencies as a set of evolving linked resources.

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

---

¹ www.europeana.eu/portal/

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

³ www.pericles-project.eu/

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 our model-driven continuum approach to preservation. Related work on representing dependencies is presented in Section 3. Section 4 offers a description of the overall use case, followed by concrete descriptions of the LRM in Section 5. Furthermore, specific examples given in Section 6 lead to a better understanding of digital dependencies that exist within the digital ecosystem, and facilitate the automated detection of inconsistencies. The paper is concluded with final remarks and directions for future work.

2 Model-driven Continuum Approach to Preservation

Traditionally, lifecycle models have been used as a point of reference for many approaches to digital preservation. They describe the preservation process as a linear sequence of distinct phases and activities, such as creation, productive use, modification and disposal, and might typically be used in higher-level organisational planning and for detecting gaps in procedures (Lagos et al. 2015). They typically envisage a clear distinction between active life and end-of-active life, where digital materials are submitted to an archive or repository at the end of their active life. The materials are then maintained as far as possible in a reusable form, aiming to preserve both the content and state. Relevant approaches include the Open Archival Information System (OAIS) (CCDS 2012), the DCC lifecycle model (Higgins 2008), and the UK Data Archive research data lifecycle (UK Data Archive 2015). Overviews of lifecycle models for research data are provided by Ball (2012) and the CEOS Working Group on Data Life Cycle Models and Concepts (WGISS 2012).

Continuum approaches on the other hand combine two main aspects. Firstly, there is no distinction made between active life and end-of-active life; that is, preservation is fully integrated into the active life of the digital objects. A second aspect is that preservation is non-custodial, that is we do not aim to remove entities from their environment, both physical and organisational and place them in the custody of a third party (Lagos et al. 2015). Such approaches have been described in Lagos et al. (2015) and McKemmish (2001).

When a continuum approach is followed, it is necessary to consider risks during active life that can occur due to changes in the environment and to determine and perform mitigating actions. We are particularly concerned with complex digital ecosystems that can comprise data, software, processes, user communities, policies and technical services; indeed, any entity that can potentially have a significant impact on the reusability of digital materials. In previous work, experiments are performed on representations of the digital ecosystem itself such as a sandbox. However, when considering complex interdependent ecosystems, in which change can propagate across multiple entities, this approach becomes impractical. We therefore adopt a model driven approach, where the models provide an abstract representation of essential features of the ecosystem, which can then be analysed and manipulated independently of the ecosystem itself. Our approach to model-driven preservation is illus-

trated in Fig. 1. The basic idea is that models are extracted from the digital ecosystem, analysed and then used to determine preservation actions on the ecosystem. In such an approach, the use of appropriate models to represent the dependencies among sets of evolving linked resources is crucial.

Fig. 1. Model-driven approach to preservation of a digital ecosystem.

3 Related Work: Existing Dependency Models

The *PREMIS Data Dictionary* (PREMIS 2015) 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.

The *Open Provenance Model (OPM)* (Moreau 2011) 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, Tzitzikas (2007) defines the 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.

Tzitzikas et al. (2010) 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 addition to the above work, Marketakis & Tzitzikas (2009) introduce the notion of *intelligibility* (which seems to be related to the notion of *task* defined in Tzitzikas et al. (2010)), 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.

4 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, since the move to a file based rather than a tape based environment, 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 specifications. Ontologically a video installation is conceived of as an installed event, bringing together the different elements to realise the artwork in a specific location at a specific time (Laurenson 2008).

In recent years video has moved from tape to file-based delivery and storage and the digital preservation community has been rapidly developing workflows, tools and services for the preservation of high-value video. Within a fine art context 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 6), as an example to help illustrate some of the ideas explored in this paper. 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 (Rice 2015). 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.

5 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 metadata models that aim to ensure that records remain accessible and usable over time (e.g. see (National Archives of Australia 2011)), 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. 2 (the prefix `lrm` 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 model⁴. A resource can be *Abstract* (c.f. `AbstractResource` in Fig. 2), representing the abstract part of a resource, for instance the idea or concept of an artwork, or *Concrete* (c.f. `ConcreteResource` in Fig. 2), representing the part of an entity that has a physical

---

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

extension. Any `ConcreteResource` can 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, the location 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 media component of 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 an element of the 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. 2), and `realizedAs` could be used to connect the abstract resource to the aggregated one.

Fig. 2. 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 place⁵. 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).

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. 3. In Fig. 3 the properties `from` and `to` indicate the directionality of the dependency.

Fig. 3. A view of the `Dependency` concept in LRM.

⁵ This example is adapted from National Archives of Australia (2011), pp. 52-53.

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 consider once more the above example: we need to be able to express the fact that transformation to 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).

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. 3. Each of the above properties (defined as *FunctionalProperty*) connect a *Dependency* to a unique *Plan*, which represents a set of actions or steps to be executed by someone/something (either human or software). The corresponding OWL fragment is shown in Table 1. 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). 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.

Table 1. OWL fragment representing the *Dependency* class.

lrm:Dependency	rdfs:subClassOf	lrm:Resource.
lrm:from	rdf:type	owl:ObjectProperty ;
	rdfs:domain	lrm:Dependency ;
	rdfs:range	lrm:Resource .
lrm:to	rdf:type	owl:ObjectProperty ;
	rdfs:domain	lrm:Dependency ;
	rdfs:range	lrm:Resource .

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 ontology for representing provenance information, but some PROV constraints are structurally incompatible with the LRM, i.e. an *Activity* cannot be an *Entity*.

```

lrm:precondition    rdf:type          owl:ObjectProperty,
owl:FunctionalProperty ;

                    rdfs:domain       lrm:Dependency ;
                    rdfs:range        lrm:Plan .

lrm:impact          rdf:type          owl:ObjectProperty,
owl:FunctionalProperty ;

                    rdfs:domain       lrm:Dependency ;
                    rdfs:range        lrm:Plan .

```

6 Domain-specific Extension: The DVA Ontology

Within PERICLES we have developed a domain-specific model to represent resources related to digital preservation of digital video art (DVA). For the representation of digital entities, we reuse and extend several constructs from CIDOC-CRM (Doerr 2005), CRMdig (Doerr & Theodoridou 2011) and LRM, ensuring, thus, *semantic interoperability* with other ontologies already aligned with these popular models. The mechanism for representing dependencies is based on the relevant LRM notions. The adopted representation approach is presented under the scope of a specific challenge, i.e. sustaining consistent video playback. The following subsections give a detailed presentation of the developed DVA domain ontology along with the relevant extensions, instantiations and mechanisms for consistency checking.

6.1 Ontology Representation

As already described in Vion-Dury et al. (2015), a digital video, i.e. one of the concrete resources (`lrm:ConcreteResource`) of a digital video artwork, incorporates: (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 `dva:hasStream`, `dva:hasContainer` and `dva:hasCodec` respectively, with a further refinement of the latter into `dva:hasVideoCodec` and `dva:hasAudioCodec`.

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

that a specific digital video artwork named “*Afyon*”⁷, where the concept of the artwork (abstract resource) is realised as a set (aggregated resource) of digital videos (concrete resources).

Fig. 4. Example of digital video artwork *Afyon* as an `lrm:AggregatedResource`.

The notion of `lrm:Dependency` is adopted to represent relations between digital video artworks and associated 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, consistent playback requires the selection of an appropriate media player that supports the container. An instance of `dva:PlayerDependency` may indicate the compatibility of media players with certain video containers. Specifically, a video container (e.g. AVI) de-

⁷ <http://www.mustafahulusi.com/afyon.html>

depends on the media players supporting its playback Fig. 6. 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.

6.2 Formalization as an Ontology Design Pattern

Within the context of PERICLES and the DVA ontology, an *Ontology Design Pattern (ODP)* was introduced in Mitziias et al. (2015) for representing digital video resources⁸ (see Fig. 7). This work was motivated by the problem of consistent presentation of digital video files in the context of digital preservation. The aim of this pattern is to model digital video files, their components and other associated entities, such as codecs and containers. The proposed design pattern facilitates the creation of relevant domain ontologies that will be deployed in the fields of media archiving and digital preservation of videos and video artworks.

6.3 Example Evaluation of Inconsistencies

The described ontology-based representations allow automatic reasoning and handling of various inconsistent cases, within the context specified by the domain of interest. In this section we present an example implementation of such a validation layer, which is based on the DVA ontology presented in Section 6.1 and uses the SPARQL Inferencing Notation (SPIN) (Knublauch et al. 2011). SPIN is a SPARQL-based rule and con-straint language for performing queries on RDF graphs. 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 their consequent sharing and reuse. SPIN can also be used to derive new RDF statements from existing ones through iterative rule application. The example described here illustrates

⁸ Also available at <http://ontologydesignpatterns.org/wiki/Submissions:DigitalVideo>

LRM's in-ferencing capacities, using topological information from a corresponding dependency network.

Fig. 7. Digital Video ODP schematic view.

In the DVA ontology, inconsistent or ‘problematic’ instances are spotted and respectively classified as error (`dva:ErrorItem`) or warning (`dva:WarningItem`) entities, incorporating at the same time corresponding descriptive message fields (i.e. properties `dva:hasErrorText` and `dva:hasWarningText`) that specify the nature of the problem. An error item describes an inconsistency, whose impact may completely affect or prevent the operation/functionality of a resource. A warning item also indicates an inconsistency but it does not prevent the operation/functionality of a resource; instead, it affects the conceptual/visual output of an action (e.g. playback activity of a digital video). The following subsections present two representative evaluation scenarios taken from Rice (2015) that have been implemented through relevant SPIN rules.

Detect Inconsistency in Container’s Metadata (no Aspect Ratio Information).

This scenario aims to detect whether a container’s metadata carry the aspect ratio information (i.e. 4:3, 16:9, 21:9) of a digital video, which is necessary for the consistent playback of video files. It is possible that some types of containers do not include information on the aspect ratio value of the digital video, even though this information may already be known by the (human) creators or owners of the files.

As an example, consider two digital video files wrapped by different container types. In the DVA ontology, this information is represented with the following triples:

<code>?digital_video_1</code>	<code>a</code>	<code>dva:DigitalVideo</code>
<code>?digital_video_1</code>	<code>dva:hasContainer</code>	<code>?avi</code>
<code>?avi</code>	<code>a</code>	<code>dva:Container</code>
<code>?avi</code>	<code>rdfs:label</code>	<code>'AVI'</code>
<code>?digital_video_2</code>	<code>a</code>	<code>dva:DigitalVideo</code>
<code>?digital_video_2</code>	<code>dva:hasContainer</code>	<code>?matroska</code>
<code>?matroska</code>	<code>a</code>	<code>dva:Container</code>
<code>?matroska</code>	<code>rdfs:label</code>	<code>'MATROSKA'</code>

where `dva:DigitalVideo` is a subclass of `lrm:DigitalResource` and `lrm:ConcreteResource`, and `dva:Container` is a subclass of `lrm:SoftwareAgent`

Due to limitations of the AVI container, the aspect ratio of `digital_video_1` is not stored in the file’s metadata; this information can be represented in the ontology with the following triples:

?avi	dva:includesAspectRatio	false
?matroska	dva:includesAspectRatio	true

When a playback activity is performed (and captured through corresponding ontology instantiations), the ontology should infer any inconsistency related to unspecified aspect ratio of a digital video in its container's metadata. Thus, in the above example, `digital_video_1` that has an AVI container will be classified as `dva:WarningItem`. An explanatory text will be attached to the item, via the use of the property `dva:hasWarningText`. The SPIN rule that was implemented to detect missing aspect ratio values in the digital video's container metadata is given below.

```

CONSTRUCT
{
  ?digital_video    a                               dva:WarningItem.
  ?digital_video    dva:hasWarningText
                    "No aspect ratio information in container".
}
WHERE
{
  ?digital_video    dva:hasContainer                ?container.
  ?digital_video    a                               dva:DigitalVideo.
  ?container        a                               dva:Container.
  ?container        dva:includesAspectRatio        false.
}

```

It is important to note that the faulty entity (here `digital_video_1`) is classified as a `dva:WarningItem` and not as a `dva:ErrorItem`, since the digital video will be played back but, possibly, not with the proper size/resolution; the media player cannot track the actual aspect ratio of the digital video and it will apply a default value instead.

Detect Inconsistency in Playback Activity (Incompatible Player).

In this case, SPIN rules check if the available media players of a given system (installation) are qualified to play the available digital video files properly. They demonstrate, in practice, how compatible media players could be detected for certain video files, based on the supported containers defined for each player. The examples given below relate to the capacities of different software players when this paper was authored.

As sample digital video files, again consider the aforementioned instantiations of `digital_video_1` and `digital_video_2` and their related containers. Based on the ontology representation, these containers may be connected with compatible media players through instantiations of class `dva:PlayerDependency`, as seen in Fig. 6 and in the triples below:

?player_dependency_1	lrm:from	?avi

```

                                lrm:to      ?windows_media_layer
                                lrm:to      ?quicktime_player
                                lrm:to      ?vlc_player

?player_dependency_2  lrm:from  ?matroska
                     lrm:to    ?vlc_player

?windows_media_player  a          dva:MediaPlayer
?windows_media_player  rdfs:label 'Windows Media Player'

?quicktime_player      a          dva:MediaPlayer
?quicktime_player      rdfs:label 'QuickTime Player'

?vlc_player            a          dva:MediaPlayer
?vlc_player            rdfs:label 'VLC Media Player'

```

where `dva:MediaPlayer` is a subclass of `lrm:SoftwareAgent`.

By interpreting the above representation manually, we may conclude that `digital_video_1` could be efficiently reproduced with any of those three media players, while `digital_video_2` could be reproduced only with VLC. In order for the ontology to automatically infer a media player incompatibility for a digital video, the corresponding instantiation of a `dva:PlaybackActivity` should be considered; the notions involved in a playback activity instance can be seen in the triples below:

```

?playback_activity_2 a          dva:PlaybackActivity
?playback_activity_2 dva:playsResource  ?digital_video_2
?playback_activity_2 lrm:uses          ?windows_media_player

```

If the used media player is not defined as compatible with the video's container for a specific playback activity, then this specific instance of activity should be classified as `dva:ErrorItem`. By evaluating the above ontology instantiations, we expect that `playback_activity_2` will be classified as an error item, because it uses Windows Media Player, which is not compatible with the container (MATROSKA) of `digital_video_2`. The SPIN rule that checks the compatibility of media players for a specific playback activity can be seen below.

```

CONSTRUCT
{
  ?activity          a          dva:ErrorItem .
  ?activity          dva:hasErrorText
                    "Incompatible player for playback activity".
}
WHERE
{
  ?digital_video     dva:hasContainer  ?container.
  ?digital_video     a              dva:DigitalVideo.
}

```

```

?dependency      lrm:from          ?container.
?dependency      a                  dva:PlayerDependency.

?activity        dva:playsResource  ?digital_video.
?activity        lrm:used           ?player.

MINUS
{
  ?dependency      lrm:to          ?player.
} .
}

```

7 Conclusions and Future Work

This work presents results related to the use of 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 have demonstrated how our model can be used to detect inconsistencies when combined with SPIN, a well-known notation for representing SPARQL rules and constraints on Semantic Web models. In the near future, we are planning of exploring the representation of causality and temporal aspects, as well as related inference frameworks.

8 Conflict of Interest

The authors declare that they have no conflict of interest.

9 References

1. National Archives of Australia. (2011). Australian Government Recordkeeping Metadata Standard Version 2.0 Implementation Guidelines. http://www.naa.gov.au/Images/agrkms-implementation-guidelines_tcm16-50156.pdf. Accessed February 2016.
2. Ball, A. (2012). Review of Data Management Lifecycle Models (version 1.0). REDm-MED Project Document redm1rep120110ab10. Bath, UK: University of Bath. <http://opus.bath.ac.uk/28587/1/redm1rep120110ab10.pdf>. Accessed 28 May 2015.
3. CCSDS - Consultative Committee for Space Data Systems. (2012). Reference Model for an Open Archival Information System (OAIS), Recommended Practice, CCSDS 650.0-M-2 (Magenta Book) Issue 2.

4. Digital Preservation Coalition. (2008). *Digital Preservation Handbook*. York, UK. <http://www.dpconline.org/advice/preservationhandbook>. Accessed April 2015.
5. Doerr, M. (2005). The CIDOC CRM, an Ontological Approach to Schema Heterogeneity. Semantic interoperability and integration. Dagstuhl Seminar Proceedings, 4391.
6. Doerr, M., Theodoridou, M. (2011). 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.
7. Higgins, S. (2008). The DCC Curation Lifecycle Model. *International Journal of Digital Curation*, 3(1), 134–40. <http://ijdc.net/index.php/ijdc/article/view/69>. Accessed 28 May 2015.
8. Knublauch, H., Hendler, J. A., Idehen, K. (2011). SPIN - overview and motivation. World Wide Web Consortium, W3C Member Submission. <https://www.w3.org/Submission/spin-overview/>. Accessed 13 February 2016.
9. Lagos, N., Waddington, S., Vion-Dury, J.-Y. (2015). On the Preservation of Evolving Digital Content - The Continuum Approach and Relevant Metadata Models. In E. Garoufallou, R. J. Hartley, and P. Gaitanou (Ed.), *Metadata and Semantics Research* 544, (pp. 15–26). Cham: Springer International Publishing.
10. Laurenson, P. (2008). Authenticity, Change and Loss in the Conservation of Time-Based Media. In J. Schachter and S. Brockmann (Ed.), *(Im)Permanence : Cultures in/out of Time*, (pp. 284). Pittsburgh, PA. University Park, Pa.: Center for the Arts in Society Distributed by the Penn State University Press. Also printed as a Tate Paper via Tate Online <http://www.tate.org.uk/research/publications/tate-papers/06/authenticity-change-and-loss-conservation-of-time-based-media-installations>.
11. Marketakis, Y., Tzitzikas, Y. (2009). Dependency Management for Digital Preservation Using Semantic Web Technologies. *International Journal on Digital Libraries*, 10(4), 159-177.
12. McKemmish, S. (2001). Placing records continuum theory and practice. *Archival Science*, 1(4), 333-359.
13. Mitzi P., Riga, M., Waddington, S., Kontopoulos, E., Meditskos, G., Laurenson, P. and Kompatsiaris, I. (2015). An Ontology Design Pattern for Digital Video. In Proceedings of the 6th Workshop on Ontology and Semantic Web Patterns (WOP 2015) co-located with the 14th International Semantic Web Conference (ISWC 2015). Vol. 1461. Bethlehem, Pennsylvania, USA.
14. Moreau, Luc, Clifford, Ben, Freire, Juliana, Futrelle, Joe, Gil, Yolanda, Groth, Paul, Kwasnikowska, Natalia, Miles, Simon, Missier, Paolo, Myers, Jim, Plale, Beth, Simmhan, Yogesh, Stephan, Eric and Van den Bussche. (2011). The Open Provenance Model core specification (v1.1). *Future Generation Computer Systems*, 27(6), 743-756. <http://eprints.soton.ac.uk/271449/1/opm.pdf>. Accessed 13 February 2016.
15. Rice, D. (2015). Sustaining Consistent Video Presentation. *Tate Research Articles*. <http://www.tate.org.uk/research/publications/sustaining-consistent-video-presentation>. Accessed 13 February 2016.
16. The Library of Congress. (2015). PREMIS - Data Dictionary for Preservation Metadata (Official Web Site). <http://www.loc.gov/standards/premis/>. Accessed April 2015.
17. Tzitzikas, Y. (2007). Dependency Management for the Preservation of Digital Information. *Database and Expert Systems Applications*, 4653 (pp. 582-592). Springer Berlin Heidelberg.
18. Tzitzikas, Y., Marketakis, Y., Antoniou, G. (2010). Task-Based Dependency Management for the Preservation of Digital Objects Using Rules. *Artificial Intelligence: Theories, Models and Applications*, 6040 (pp. 265–274). Springer Berlin Heidelberg.

19. UK Data Archive. Research Data Lifecycle. <http://www.data-archive.ac.uk/create-manage/life-cycle>. Accessed 28 May 2015.
20. Vion-Dury, J.-Y., Lagos, N., Kontopoulos, E., Riga, M., Mitziias, P., Meditskos, G., Waddington, S., Laurenson, P. and Kompatsiaris, I. (2015). Designing for Inconsistency - The Dependency-based PERICLES Approach. In T. Morzy, P. Valduriez, L. Bellatreche (Ed.), *New Trends in Databases and Information Systems*, 539, (pp. 458-467). Springer Berlin Heidelberg.
21. WGISS - Committee on Earth Observation Satellites Working Group on Information systems and Services. (2012). Data Life Cycle Models and Concepts CEOS 1.2. http://ceos.org/document_management/Working_Groups/WGISS/Interest_Groups/Data_Stewardship/White_Papers/WGISS_DSIG_Data-Lifecycle-Models-And-Concepts-v13-1_Apr2012.docx. Accessed 28 May 2015.