INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

An Annotation Propagation Mechanism for Multimedia Content

Gilberto Zonta Pastorello Jr   *Jaudete Daltio*
Claudia Bauzer Medeiros

Technical Report - IC-08-017 - Relatório Técnico

August - 2008 - Agosto

An Annotation Propagation Mechanism for Multimedia Content*

Gilberto Zonta Pastorello Jr   Jaudete Daltio   Claudia Bauzer Medeiros

Institute of Computing – UNICAMP
PO Box 6176 – 13083-970
Campinas – SP – Brazil
gilberto@ic.unicamp.br
jaudete@lis.ic.unicamp.br
cmbm@ic.unicamp.br

Abstract

Scientific research is producing and consuming large volumes of multimedia data at an ever growing rate. Metadata – data about data – is the primary mechanism through which context is associated to content to enhance content management. It also makes it easier to interpret and share data and helps digital curation. However, raw data often needs to go through complex processing steps before it can be consumed. During these transformation processes, original metadata from the production phase is often discarded or ignored, since its usefulness is usually limited to the first transformation step. New metadata must be associated with the final product, a time consuming task often carried out manually. Systematically associating new metadata to the result of each data transformation step is know as *metadata evolution* or *annotation propagation*. This paper introduces techniques for semantically enhancing metadata and automatically transforming them along with the data transformation processes. This helps the construction of new annotated multimedia data sets, preserving contextual information. The solution is based on: (i) the notion of semantic annotations, which are metadata structures enriched with domain ontologies; (ii) a set of transformations rules, based on ontological relations; and (iii) workflows, which steer the sequence of transformations.

*This work was supported by FAPESP (grant number 04/14052-3), and partially financed by CNPq.

1 Introduction

Scientific applications are producing and consuming ever growing volumes of multimedia data, which may vary from sensor based data (e.g., aboard satellites or ground-based) to video and sound recordings. In this scenario, scientists constantly need to share and reuse their data sets, being hampered by the wide spectrum of data production devices, actors and contexts that are involved in a lifecycle that *produces*, *transforms* and *consumes* data.

Metadata – data about data – have been used for a long time as a means to help data management tasks. They are used to document and qualify data sets, thereby helping their interpretation and sharing. Metadata are, often, text fields associated to data (e.g., in a file header) to be directly applied in automated data management tasks, such as indexing, searching, or context integration [6, 9, 10, 12, 22, 23, 35, 37]. Metadata granularity and contents vary greatly, depending on factors such as application design, or the target domain. A wide variety of metadata standards have been proposed to help document data production and publication – e.g., Darwin Core [20] for biodiversity, ISO19115 [33] for geographic data, or MPEG-7 [48] for multimedia. Nevertheless, standards vary widely and metadata structures often impose severe restrictions on how to describe the data.

Annotations – descriptions of a data set – are also used with the same goal, but they are more flexible than metadata, often being personal remarks created by data producers or consumers [2, 28, 31]. Examples of annotations are keywords, a log file, or a voice recording. Annotations are harder to be used in an automated fashion, as they lack structure. However, the task of describing the data is facilitated, being personalised to a context.

Albeit helpful in improving data interpretation, metadata and annotations become less useful, or even useless, as soon as the data set goes through some sort of processing function. The resulting data set requires new metadata. Roughly speaking, this characterizes the scenario for *metadata evolution* or *annotation propagation* [8, 11]. This poses the following problems: (1) how to propagate relevant metadata and annotations that are discarded during a transformation? and, (2) how to support automatic creation of metadata and annotations for the transformed data, taking context into account? Our work contributes towards solving these two questions.

In more detail, the traditional life-cycle for data sets is *(a) production – (b) transformation – (c) consumption*, where stage (b) may involve several steps. Most data interpretation tasks occur in the last stage. Adding metadata to the process improves the interpretation, and the cycle becomes *(a) production – (a') annotation – (b) transformation – (b') (re-)annotation – (c) consumption*. The main interest in this paper is on how to (partially or totally) automate stage (b'). In particular, we are concerned with combining the notions of metadata, annotations and ontologies producing what we call *semantic annotations*, in which annotations are structured and are defined in terms of references to ontology concepts and/or relationships. Ontological terms help provide contextual information.

Our approach involves using semantic annotations both from data and from operations that transform the data. Taking advantage of that, we propose a mechanism through which annotations are generated and attached to data sets produced by a data transformation operation. These new annotations are derived from the annotations made on the input data and the operation's interface, using a set of propagation rules that are based on ontology terms and relationships.

This process is extended to a sequence of data transformation steps – and thus to propagation of data annotations. We adopt scientific workflows to specify and aid the execution of activities in scientific environments. We consider the operations that transform data to be workflow activities, and complex transformations are achieved by composing these activities into a workflow. Our propagation rules are applied in the sequence determined by the workflow: as data evolves and changes as defined by the workflow, so will semantic annotations. This maintains the context information (provided by annotations) associated with the content (the data sets).

The main contributions of this paper are therefore: (i) a general definition for the annotation propagation problem, applicable to several different environments of data transformation; and (ii) an extensible ontology-guided technique for solving the annotation propagation problem using semantic annotations. Though placed in the multimedia data management context, our solution can be extended to any environment where digital content is acquired, transformed and shared.

The remainder of the paper is organized as follows. Section 2 defines the annotation propagation problem, exemplifying it and classifying the problem in four categories. Section 3 presents our solution to the problem using semantic annotations. Section 4 considers operations that have more than one input and/or output, i.e., multiple input/output operations. Section 5 looks at composing several data transformation operations into a workflow. Section 6 shows the application of our mechanism in our running example. Section 7 discusses related work. Section 8 presents conclusions and future work.

2 The Annotation Propagation Problem

This section introduces the annotation propagation problem. Section 2.1 introduces a running example that illustrates our solution throughout the paper. Section 2.2 defines semantic annotations. Section 2.3 defines the problem of annotation propagation. Finally, Section 2.4 proposes a classification of the problem into four categories.

2.1 Example

Figure 1 illustrates a multimedia data transformation process in environmental modeling that combines satellite images with temperatures readings (from ground sensors) for a given region and period and generates an animation showing how temperature influences vegetation growth in the region. In a high level, the steps are the following:
(1a) acquire temperature data (data streams) and (1b) satellite images for a given region (in a format called GeoTIFF[1]);
(2a) convert the temperature readings and (2b) compute the greenness of vegetation (using the so-called NDVI method[2]) on the satellite image generating a NDVI image;
(3a) generate a temperature map interpolating readings (e.g., using Thiessen polygons) and (3b) classify the regions in the NDVI image according to the greenness range – both maps generated at step 3 are GeoTIFF images;
(4) combine the two images into one;
(5) convert the resulting map into a JPEG image; and,
(6) generate an animation from the iteration of the previous steps.

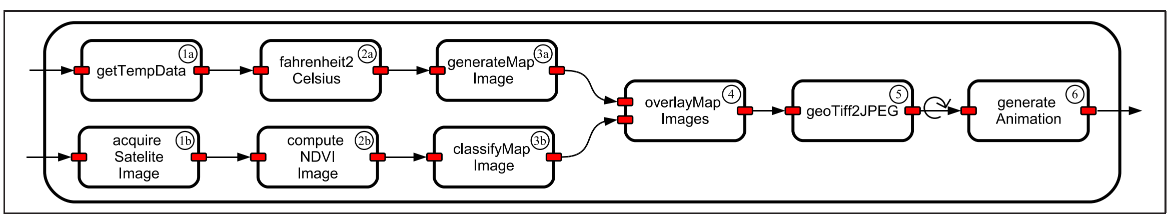

Figure 1: Example of a data transformation process.

The animation illustrates the correlation between temperature and vegetation conditions in the area and is sufficient for a high level view of this problem. However, it is unsuitable for any kind of scientific study on environmental conditions. First, each JPEG image should be annotated indicating that it is the result of combining satellite and sensor data. This may still not be enough – sensor type and calibration, satellite type and spectral band used must be informed. This contextual information is lost during the transformation process, unless all multimedia data are manually annotated. This is difficult for large processes and impossible if parts of the process are done by different people or organizations.

[1] An image format where each pixel corresponds to a given geographical coordinate.
[2] Normalized Difference Vegetation Index is an indicator for levels of live green vegetation, usually over satellite images.

The more complex the data and the transformations performed, the greater the need for contextual information. The annotations for the input data sets (satellite image and temperature readings) are provided by their source. However, at the end of the process, the output animation has no associated annotations.

2.2 Semantic Annotations

We consider two basic approaches to describe data sets: metadata and annotations. Metadata have a well defined structure, and annotations are notes added as comments, or explanations, without any defined structure or value range. The very flexibility of annotations hampers automated processing, whereas metadata are less flexible, but support functions like indexing or searching.

We combine both approaches into *semantic annotations*, in which metadata structures and part of their contents are defined by means of references to ontologies tailored to user needs. The latter can be extended as needed, providing contextual information. Automation is enabled by ontology related operations (e.g., alignment of terms). We define semantic annotations as follows.

> **Annotation Units**. An *annotation unit a* is a triple `<s,p,o>`, where `s` represents the subject being described, `p` represents a property that describes it, and, `o` represents a describing object or value.
>
> **Semantic Annotation**. A *semantic annotation* `M` is a set of one or more annotation units, with at least one unit having as its subject the entity being described.

A semantic annotation is materialized as an RDF graph, which is represented as a set of RDF triples (*subject – predicate – object*); subject and predicate are identified by an URI[3] while the object may be an URI or a literal. Note that an object on one annotation unit may be itself a subject on another unit. This is the basic structuring element for semantic annotations. Examples of semantic annotations are given in Section 2.3.

Our definition of semantic annotation falls under the category of formal and explicit (as opposed to informal and tacit) following the classification proposed by [45]. Similar to the definition in [5], annotation structures have a given format and the annotations are intended to be manipulated automatically.

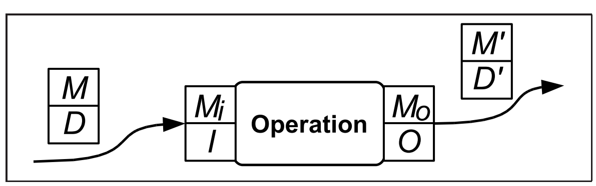

Figure 2: Data sets and interfaces and their respective annotations.

We assume that a data transformation operation is a black-box that can be invoked providing data as input and producing output data. Semantic annotations are basically used to describe two entities in our solution: the data sets used and the interfaces of transformation operations. Figure 2

[3]Or, to be more precise, a URIref, which is a URI that may have a fragment identifier (the symbol "#") at the end, for referencing parts of the URI.

shows a transformation operation. The input data set D is annotated with M and the output data set D' is annotated with M'. The operation has its input interface I described by semantic annotation $\mathsf{M_i}$ and its output interface O described by $\mathsf{M_o}$. Our solution assumes that operations must be described by an ontology that provides at least identification. Although the operations' input/output interfaces are semantically annotated, the operations themselves need not provide any description about their behaviour.

2.3 Annotation Propagation

Loosely speaking, given a data "production – transformation – consumption" lifecycle, the annotation propagation problem is how can one enable relevant annotations generated at the production stage to be used at the consumption stage? It is not enough to just repeat the annotation after the transformation, as some annotations might not be valid anymore and/or additional annotations might be needed.

More precisely, let $(\mathsf{T},\ \mathsf{I},\ \mathsf{O},\ \mathsf{D},\ \mathsf{D}')$ denote an application of a data transformation operation T which has an input interface I and an output interface O, andis applied on a data set D, resulting in derived data D'. Also, let $(\tau, \mathsf{M_i}, \mathsf{M_o}, \mathsf{M}, \mathsf{M}')$ denote an application of an annotation transformation τ that manipulates $\mathsf{M_i}$ (the annotation of I), output interface annotations $\mathsf{M_o}$ (the annotation of O) and M (the annotation of D) to achieve M' (the derived annotation of D'). The annotation propagation problem is defined as follows.

> **The Annotation Propagation Problem**. Consider a transformation T, with an input interface I and an output interface O, applied to a data set D, transforming it into another data set D'. Which transformation τ can generate the new annotations M', given the previous annotation M on the data, the annotation of the input interface $\mathsf{M_i}$ and the annotation of the output interface $\mathsf{M_o}$?

If we now extend this definition to consider semantic annotations, the problem becomes: how to combine the sets of annotation units from the semantic annotation of the data set, the input interface and the output interface, to generate a new set of annotation units that will constitute the new semantic annotation. The mechanism to do that should ensure the consistency of the new set as well as its completeness regarding the available annotations.

For the remainder of the text the term annotation refers to semantic annotation.

As the operations considered are black-box operations, the annotation propagation in each step must be carried out outside the scope of the operation, i.e., by an external application. Thus, data transformation and annotation propagation do not interfere with each other. Therefore, it is necessary to have a steering entity responsible for the invocation of the data transformation and the invocation of the annotation propagation mechanism. As will be seen, our approach applies workflow management systems (WFMS) as the steering entity and features from ontology services to implement the annotation propagation application.

Let us go back to our running example, and single out the classifyMapImage transformation operation that classifies an NDVI image generating a classified image. Figure 3 shows the association of annotations: M to the input data set (D), M' to the output data set (D'), $\mathsf{M_i}$ to the input interface of the operation (I) and $\mathsf{M_o}$ to the output interface of the operation (O). The bottom of Figure 3 portrays the transformation. The input data set (D) is an NDVI GeoTIFF image. The operation has one input interface (I), which takes one parameter (p1), and one output interface (O), which produces the parameter (p2). The output data set (D') is the classified GeoTIFF image. These entities (data sets and interfaces) are described with semantic annotations, e.g., the pair $(\mathsf{O}, \mathsf{M_o})$ denotes that the semantic annotation $\mathsf{M_o}$ is associated with output interface O, similarly to (D, M), $(\mathsf{I}, \mathsf{M_i})$ and $(\mathsf{D'}, \mathsf{M'})$.

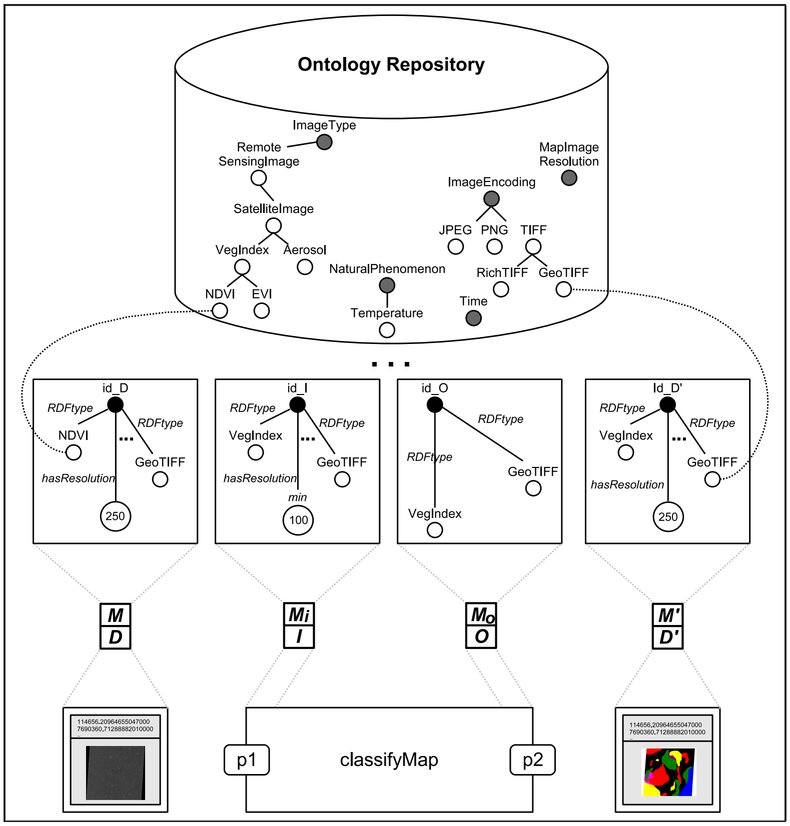

Figure 3: Semantic Annotation on Data and Interfaces.

The top of the figure illustrates an ontology repository [19], a data space containing domain ontologies with the contextual and semantic information. The graphs in the boxes in the middle of the figure show how annotations are structured. M (at the left) is the annotation for the data set D: it is a graph rooted at idD (the ID of the entity being described) and each edge defines the scope of one annotation *unit*. Units are stored as RDF triples. Thus, the annotation for D is {(idD, RDFType, NDVI), (idD, hasEncoding, GeoTIFF-bitmap), (idD, capturedBy, Terra-MODIS) (idD, usedBand, band4), (idD, usedBand, band5), (idD, hasOrbit/Point, 220/075F), (idD, hasDatum, WGS84), (idD, hasProjection, UTM_ELLIPSOID), (idD, capturedOn, 20010323), ...}, indicating that it is an *NDVI image*, encoded in a *bitmap GeoTIFF*, captured by the *Terra-MODIS* satellite sensor, used *spectral bands 4* and *5*, and so on. By the same token, $\mathsf{M_i}$ says that I takes a parameter that should be a *Vegetation Index (VI) image*, encoded in a *bitmap GeoTIFF*, and so on; $\mathsf{M_o}$ indicates that O has as its output a *VI image*, encoded in a *bitmap GeoTiff*, and so on.

2.4 Classification of Propagation Strategies

Before proceeding to the propagation mechanism, we introduce a classification of annotation propagation strategies. Figure 3 depicts the basic scenario (one transformation operation, one single data input and one single data output). M has elements with the values: "bitmap", "NDVI", "250", and "20010323", referencing, respectively, the ontology defined concepts *Encoding*, *Image Type*, *Resolution*, and *Timestamp*. Interface I expects a "Vegetation Index (VI) image", encoded as a "bitmap", with a resolution of at least "100" metres and can, as an optional feature, received a GeoTIFF image compressed with the LZW algorithm. Interface O generates a "Classified VI image" encoded as a "bitmap". The data set D is a "NDVI image" which is converted into D', a "Classified VI image". The question is: "which values should M' have?"

Given $(\mathsf{T},\ \mathsf{I},\ \mathsf{O},\ \mathsf{D},\ \mathsf{D}')$ and $(\tau, \mathsf{M_i}, \mathsf{M_o}, \mathsf{M}, \mathsf{M}')$, the code of T does not influence propagation, which should only consider relationships among $\mathsf{M_i}, \mathsf{M_o}, \mathsf{M}$ to produce M'. We define the relationship between the input (I) and output (O) of a transformation operation to be determined by the ontological relationships among their annotations ($\mathsf{M_i}$ and $\mathsf{M_o}$, respectively). Two factors can be used to classify the annotation propagation mechanism: (i) taking into account or not the relationship between the annotations on the input and the output of an operation; and (ii) taking into account or not annotations that cannot be matched against another (e.g., annotation units from M that cannot be compared to any other unit from $\mathsf{M_o}$). Hence, four types of transformations are possible.

Regarding item (i), if the relationship between the input and output annotations is ignored, annotation propagation is based on directly applying the propagation rules considering only M and $\mathsf{M_o}$. This is the first type of propagation: Closed $\mathsf{M} - \mathsf{M_o}$ ($\mathsf{CMM_o}$). If, instead, the propagation considers the relationship, first we determine which terms from $\mathsf{M_i}$ influence $\mathsf{M_o}$, and the rules are applied only to these terms. This is the second type of propagation: Closed $\mathsf{M} - \mathsf{M_o}$ through $\mathsf{M_i}$ ($\mathsf{CMM_oM_i}$).

Considering (ii), it is possible that parts of one annotation (e.g., M) cannot be compared to any other part on the other annotation (e.g., $\mathsf{M_o}$). Propagating any of these parts may produce richer annotations, at the cost of the possibility of introducing meaningless and/or erroneous annotations. Adding this degree of freedom to the previous two transformations, we have: the Open M–$\mathsf{M_o}$ propagation (called $\mathsf{OMM_o}$) and the Open M–$\mathsf{M_o}$ through $\mathsf{M_i}$ propagation (called $\mathsf{OMM_oM_i}$). Table 1 exemplifies the contents of each kind of propagation for the example using the $\mathsf{classifyMapImage}$ operation. The elements on the table cells are the values of an annotation unit.

	M	$\mathsf{M_i}$	$\mathsf{M_o}$	derived content of M' with transformation:			
				$\mathsf{CMM_o}$	$\mathsf{CMM_iM_o}$	$\mathsf{OMM_o}$	$\mathsf{OMM_iM_o}$
Encoding	*bitmap*	*bitmap*	*bitmap*	*bitmap*	*bitmap*	*bitmap*	*bitmap*
Image Type	*NDVI*	*V.I.*	*Class. V.I.*	*Class. V.I.*	*Class. V.I.*	*Class. V.I.*	*Class. V.I.*
Resolution (m)	*250*	*100+*	—	—	*250*	*250*	*250*
Timestamp	*20010323*	—	—	—	—	*20010323*	*20010323*
Compression	—	*LZW*	—	—	—	—	*LZW*

Table 1: Values of M' for each type of propagation.

For $\mathsf{CMM_o}$ the only comparable concepts (between M and $\mathsf{M_o}$) were *Encoding* and *ImageType* – thus M' in this case contains "bitmap" and "Classified VI image". Propagation type $\mathsf{CMM_iM_o}$ had the *Resolution* concept matched between $\mathsf{M_i}$ and $\mathsf{M_o}$, thus being included in the result. $\mathsf{OMM_o}$ propagated the M and $\mathsf{M_o}$ unmatched *Timestamp* concept, while for $\mathsf{OMM_iM_o}$ the *Timestamp* concept was first matched (between $\mathsf{M_i}$ and $\mathsf{M_o}$) and then propagated. Actually, differentiation of behaviour between $\mathsf{CMM_o}$ and $\mathsf{CMM_iM_o}$ occurs when the input annotation is not directly reflected into output annotation. Similarly, $\mathsf{OMM_iM_o}$ differs from $\mathsf{OMM_o}$ when one part of the input annotation does not match anything from the data annotations, as was the case in Table 1: the "LZW" was attached to M' in the $\mathsf{OMM_iM_o}$ propagation strategy. This last situation only occurs with optional fields on an input annotation. Using either of the two open propagation strategies may lead to error or inconsistencies – as is the annotation "LZW" for the $\mathsf{OMM_iM_o}$ strategy, since the output of the classifyMapImage is not annotated as compressed. This problem is not considered in this paper.

Section 3.3 presents our mechanism for propagation strategies $\mathsf{CMM_o}$ and $\mathsf{CMM_iM_o}$. Section 3.6 discusses the aspects involved in the propagation strategies $\mathsf{OMM_o}$ and $\mathsf{OMM_iM_o}$.

3 Semantic Annotation Propagation

This section presents our solution for the annotation propagation problem. In this solution, any data transformation operation performed on a data set must be accompanied by transformations on the associated metadata. No assumptions are made about the format or granularity of the data. The annotation propagation mechanism should work equally for any kind of multimedia data. The presented mechanism for annotation propagation considers a basic data transformation process: a single operation with one data input and one data output. Most definitions on data transformation are borrowed or adapted from [18]. Section 3.1 introduces concepts used in the solution. The solution consists in (1) breaking down the annotations into comparable units, (2) applying a set of propagation rules, and (3) generating a new annotation for the transformed data set. Section 3.2 describes phases (1) and (3). Section 3.3 presents the closed propagation rules for phase (2). Section 3.4 shows examples of the ontological relations that determine the behaviour of the propagation rules on phase (2). Section 3.5 shows a high level algorithm for the solution. Section 3.6 discusses the open propagation rules for phase (2).

3.1 Ontology Basics

From a computer science perspective, an ontology can be viewed as a data model that represents a set of concepts within a domain and the relationships between those concepts [30]. Knowledge in an ontology is formalized using four kinds of components:

- **Classes:** sets, or kinds of objects (concepts or categories of concepts in the domain), usually organized in taxonomies;
- **Instances:** the objects in the domain, represented as instances of a class;
- **Properties:** used to describe instances/classes. Properties may express attributes (features, characteristics, or parameters that objects can have and share) or express how objects can be related to one another;
- **Constraints:** abstractions that use properties to describe a class, usually expressed as axioms in description logic.

Many languages may be used to represent an ontology, such as RDF (*Resource Description Framework*) [44] and OWL (*Web Ontology Language*) [4]. The basic construct of the RDF model is a triple of the form *subject–predicate–object*, where *subject* refers to a resource, anything that can be denoted by a URI (Uniform Resource Identifier); *predicate* is a property of that resource; and *object* is the value of that property. The object can be a literal or another resource.

Our annotations are represented in RDF. Our propagation rules are thus based on ontological relationships expressed in RDF-Schema and OWL – see some relations in Section 3.4 and others in the appendix.

3.2 Semantic Annotation Deconstruction–Reconstruction

From a high level perspective, our solution for annotation propagation – τ – is processed in three steps:

1 – Deconstruction. Break $\mathsf{M}, \mathsf{M_i}$ and $\mathsf{M_o}$ annotations into comparable annotation units using some deconstruction function $\mathscr{F}$. Comparability is defined in terms of a set of ontological relations – e.g., to compare concepts under the OWL subClassOf relation, the semantic annotations must be broken into classes;

2 – Comparison/Transformation. Compare the annotation units under the set of relations used in the deconstruction phase, and apply propagation rules to these units according to the outcome of the comparison. The result is a set of transformed annotation units. For instance, if the previous phase produced class A from M and class B from $\mathsf{M_o}$, then this step will check whether A and B can be related via relation subClassOf. Our propagation rule for this relation will determine that the result is either A, B, or empty, depending respectively whether A is a subClassOf B, B is a subClassOf A, or neither – see Section 3.4;

3 – Reconstruction. Reconstruct the semantic annotation from the transformed units using a reconstruction function $\mathscr{F}^{-1}$ – e.g., in the example, an annotation unit will contain respectively A, B, or will be empty.

In our running example, the NDVI bitmap image used as input to operation classifyMapImage would have its annotations deconstructed into RDF tuples (annotation units) by $\mathscr{F}$. One such tuple would be (id_D, RDFType, GeoTIFF-bitmap). This would be propagated to the output classified image, and be reconstructed, along with any other propagated annotation unit, by $\mathscr{F}^{-1}$ into the output's annotation.

For some cases, the definition of $\mathscr{F}$ and $\mathscr{F}^{-1}$ is straightforward. For instance, when considering ontological relations using RDF representation, which is the case in this paper, the annotations are already expressed as triples and need not to be deconstructed. For more complex cases, $\mathscr{F}$ and $\mathscr{F}^{-1}$ may need more specific behaviours. For instance, if the annotation units we want to compare are pairs of paths on an RDF graph, $\mathscr{F}$ would start from an annotation seen as a directed graph, and obtain all root to leaf paths within the graph. These paths would then be compared under relations like common subpaths. Conversely, the $\mathscr{F}^{-1}$ (construct) function would recreate a (directed graph) annotation from the derived paths.

3.3 Closed Propagation Rules

This section presents the $\mathsf{CMM_o}$ and $\mathsf{CMM_iM_o}$ strategies of our solution.

Consider again transformation (T, I, O, D, D') with interface annotations $\mathsf{M_i}$ and $\mathsf{M_o}$. Let (D,M) and (D',M') be, respectively, its input and output data and annotations. First we point out that $\mathsf{M_i}$ and $\mathsf{M_o}$ are associated with a transformation operation which was designed and implemented by a given organization. Thus, $\mathsf{M_i}$ and $\mathsf{M_o}$ should be semantically and syntactically compatible. The propagation problem can be simplified into the following issues:

- which mappings can be done between M and $\mathsf{M_o}$ (for the $\mathsf{CMM_o}$ propagation level);
- which mappings can be done between $\mathsf{M_i}$ and $\mathsf{M_o}$, resulting in a new set of annotation units to

be considered; and how to compute mappings between M and this new set (for the $\mathsf{CMM_iM_o}$ propagation level).

We approach this problem by considering two issues: (i) a set of abstract propagation rules to be applied to the annotations; and (ii) a set of ontological relations, each specifying how to compare a pair of annotation units and which annotation should be derived from the comparison. The solution to the first subproblem is a set of generic and recursively defined annotation propagation rules, shown in Table 2. The solution to the second subproblem is devised in Section 3.4.

Mappings between annotations require their decomposition into sets of comparable units (by a deconstruction function $\mathscr{F}$). Then the rules are applied to the sets, generating a new set, which is translated back (by a reconstruction function $\mathscr{F}^{-1}$) to the annotation format. The outcome of this process is represented by $\mathscr{P}$ in rule (4) of Table 2.

Intuitively, the rules recursively consider single comparable annotations units, checking the compatibility between all pairs of units generating new units for the resulting set. In Table 2, Ω represents the deconstructed set generated from the annotations on the data (M), Θ represents the deconstructed set generated from the annotations on the output interface of the operation ($\mathsf{M_o}$) – considering that the $\mathsf{CMM_o}$ type of propagation is being used – and Δ is the set of resulting units to be translated back into the resulting new annotations (M'). If the $\mathsf{CMM_iM_o}$ type of propagation is being used, Θ represents the deconstructed set generated from the mapping between $\mathsf{M_i}$ and $\mathsf{M_o}$. The first three rules state that empty sets of units should not alter the resulting set.

To consider the application of first mapping M_i and M_o, an alignment among the corresponding ontology concepts is first executed, and the result is considered instead of M_o. Rule (4) is the essence of the propagation rules, and compares pairs of units as to how they should influence the resulting set. The last three rules are adopted to pick out the sets and recursively apply the first four rules.

Let $\mathscr{R}$ be a relation under which units of M, $\mathsf{M_i}$ and $\mathsf{M_o}$ are to be compared. Let $\mathscr{F}$ be a function that translates M into a set of comparable units Ω, $\mathsf{M_o}$ into Θ. The steps from Section 3.2 can be written as:

(1) Apply $\mathscr{F}$ to M generating Ω. Apply $\mathscr{F}$ to $\mathsf{M_o}$ ($\mathsf{CMM_o}$) or to a mapping between $\mathsf{M_i}$ and $\mathsf{M_o}$ ($\mathsf{CMM_iM_o}$), generating Θ;

(2) Apply rules in Table 2 to Ω and Θ, obtaining Δ, a set of annotation units;

(3) Apply $\mathscr{F}^{-1}$ to Δ, obtaining M'.

The contents of Δ depend on the kind of relation being considered. Each unit in Δ is determined by $\mathscr{P}$ – the expected result from the relation being considered.

Therefore, to use the propagation rules one must specify:
(i) a function $\mathscr{F}$ to translate M into Ω, $\mathsf{M_o}$ into Θ (for the $\mathsf{CMM_o}$ type of propagation), or the mapping between $\mathsf{M_i}$ and $\mathsf{M_o}$ into Θ (for the $\mathsf{CMM_iM_o}$ type of propagation); (ii) its inverse $\mathscr{F}^{-1}$ to translate Δ into M';
(iii) one or more relations $\mathscr{R}$ under which the units will be compared; and
(iv) a set of propagation results $\mathscr{P}$ which express the expected results from each ontological relation. As will be seen in Section 3.4, $\mathscr{P}$ is the result of pairwise comparison between all annotation units under an ontological relation $\mathscr{R}$.

Then, the rules are applied in sequence. $\mathfrak{R}$ is the set of all relations ($\mathscr{R}$) chosen to be used.

(1) $|\Omega| = 0 \wedge |\Theta| = 0 \leadsto \Delta = \emptyset$

(2) $|\Omega| = 0 \wedge |\Theta| = n \leadsto \Delta = \emptyset$, where: $n \geq 1$

(3) $|\Omega| = m \wedge |\Theta| = 0 \leadsto \Delta = \emptyset$, where: $m \geq 1$

(4) $|\Omega| = 1 \wedge |\Theta| = 1 \leadsto \Delta = \mathscr{P}$ (which is the resulting set for each relation on Section 3.4)

(5) $|\Omega| = 1 \wedge |\Theta| = n \leadsto \Delta$, where:

$$\Delta = \bigcup_{1 \leq j \leq n} \delta_j$$

and δ_j is the result of the application of rule (4) to a and γ_j, where $a \in \Omega \wedge \gamma_j \in \Theta$, for $1 \leq j \leq n$ and $n > 1$.

(6) $|\Omega| = m \wedge |\Theta| = 1 \leadsto \Delta$, where:

$$\Delta = \bigcup_{1 \leq i \leq m} \delta_i$$

and δ_i is the result of the application of rule (4) to γ_i and b, where $\gamma_i \in \Omega \wedge b \in \Theta$, for $1 \leq i \leq m$ and $m > 1$.

(7) $|\Omega| = m \wedge |\Theta| = n \leadsto \Delta$, where:

$$\Delta = \bigcup_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} \delta_{i,j}$$

and $\delta_{i,j}$ is the result of the application of rules (4) to γ_i and γ_j, where $\gamma_i \in \Omega \wedge \gamma_j \in \Theta$, for $1 \leq i \leq m$ and $1 \leq j \leq n$ and $m > 1$ and $n > 1$.

Table 2: General Closed Annotation Propagation Rules

3.4 Ontological Relations

The ontological relations used in this paper manipulate the basic elements from an OWL ontology, i.e., the classes, instances and properties. These elements are to be compared to determine which of them are to be used as the derived annotation.

The choice of which ontological relations to use in a propagation should be guided by which aspects are useful in a given transformation process. For instance, if class hierarchy relationships are useful, generalization and specialization ontological relations should be used – e.g., considering descendant full compatibility or restricting to only direct subclass compatibility.

We categorize the ontological relations according to which element from the annotation units are the focus of the comparison. Three categories are defined: classes, instances and properties. For all ontological relations presented, the annotations units being compared (a and b) are RDF triples $<$*Subject,Predicate,Object*$>$ and have the form $a =$ `<sa,pa,oa>` and $b =$ `<sb,pb,ob>`. The next ontological relations are all on the classes category. Appendix A contains other kinds of ontological relations that can be used.

Class generalization. Relation based on the rdfs:subClassOf construct, defined as part of RDF Schema. This construct establishes that if A is a sub-class of B, than all properties valid for B are also valid for A.

$$\mathscr{R}(a,b) = \begin{cases} \texttt{<sa,pa,oa>} & \text{if } \textit{sa subClassOf } \texttt{sb} \\ \texttt{<sa,pa,oa>} & \text{if } \textit{sa subClassOf } \texttt{ob} \end{cases}$$

This relation should be read as follows. Given two annotation units $a =$ `<sa,pa,oa>` and $b =$ `<sb,pb,ob>`, generated by a deconstruction function for comparison under *subClassOf*, then the propagated annotation unit is: `<sa,pa,oa>` if *sa* is a *subClassOf* `sb`. All other relations are to be read the same way.

Class specialization. This relation is also based on the rdfs:subClassOf construct, because generalization and specialization are both described by this ontological relation.

$$\mathscr{R}(a,b) = \begin{cases} \texttt{<sb,pa,oa>} & \text{if } \texttt{sb} \textit{ subClassOf } \texttt{sa} \\ \texttt{<ob,pa,oa>} & \text{if } \texttt{ob} \textit{ subClassOf } \texttt{sa} \end{cases}$$

Class equivalence. Relation based on OWL owl:equivalentClass. This construct stablishes that if A is equivalent to B, than all properties valid for B are also valid for A, and vice-versa.

$$\mathscr{R}(a,b) = \begin{cases} \texttt{<sa,pa,oa>} & \text{if } \texttt{sa} \textit{ equivalentClass } \texttt{sb} \\ \texttt{<sa,pa,oa>,<sb,pb,ob>} & \text{if } \texttt{sa} \textit{ equivalentClass } \texttt{ob} \\ \texttt{<sa,pa,oa>,<sb,pb,ob>} & \text{if } \texttt{oa} \textit{ equivalentClass } \texttt{sb} \\ \texttt{<sa,pa,oa>} & \text{if } \texttt{oa} \textit{ equivalentClass } \texttt{ob} \end{cases}$$

3.5 Algorithm

Table 3 shows the algorithm for the $\mathsf{CMM_o}$ type of propagation. For the $\mathsf{CMM_iM_o}$ type of propagation, Θ_i receives the result of the mapping between $\mathsf{M_i}$ and $\mathsf{M_o}$ in line (5) of the algorithm. The deconstruction and reconstruction steps (lines 4, 5 and 9) are carried out for each relation as specific functions ($\mathscr{F}_k^{-1}$) are needed for each one. If the annotation units do not match under the ontological relation (line 8), the result of $\mathscr{R}_k(a,b)$ is empty.

```
1.  Δ ← ∅
2.  foreach 𝓡_k in ℜ do
3.      𝓟_k ← ∅
4.      Ω_k ← 𝓕_k(M)
5.      Θ_k ← 𝓕_k(M_o)
6.      foreach a in Ω do
7.          foreach b in Θ do
8.              𝓟_k ← 𝓟_k ∪ 𝓡_k(a,b)
9.      Δ ← Δ ∪ 𝓕_k^{-1}(𝓟_k)
10. return Δ
```

Table 3: M–M_o Transformation Algorithm

3.6 Open Propagation Rules

Open propagation rules propagate annotation units if they do not match another unit under the chosen ontological relations. The differentiation between open and closed rules is done in the two of the general propagation rules.

The rules have a special version asserting a different behaviour in the case of two annotation units not being comparable. Section 3.3 presented the closed behaviour of the rules ($\mathsf{CMM_o}$ and $\mathsf{CMM_oM_i}$). The version that defines the open behaviour of the rules ($\mathsf{OMM_o}$ and $\mathsf{OMM_oM_i}$) are presented on Table 4. The rules that need adaptation are rules (2) and (3). The algorithm presented in Table 3 does not need adaptation, since the change is done on the function that tests the ontological relation ($\mathscr{R}_k(a,b)$).

(2a) $\vert\Omega\vert = 0 \land \vert\Theta\vert = n \rightsquigarrow \Delta = \Theta$, where: $n \geq 1$
(3a) $\vert\Omega\vert = m \land \vert\Theta\vert = 0 \rightsquigarrow \Delta = \Omega$, where: $m \geq 1$

Table 4: General Open Metadata Propagation Rules

The union step (line 8) of the algorithm presented in Table 3 is a bit more complicated when adding open rules. The annotations propagated from closed rules should precede the ones from open rules. So, if a given unit was already propagated by a closed rule it should not be considered by any other open rule.

Again, it is worth of notice that the open propagation strategies may generate richer annotations, but also erroneous and/or inconsistent ones. This issue is not addressed in this paper and is object for future work.

4 Multiple Input/Output Operations

If a transformation operation has more than one input or output, propagation becomes more complicated. We treat each case separately, for clarity sake. For multiple inputs, the idea consists in merging the annotations from each input data set into a single annotation (using ontology alignment), and doing the same for the annotations on the input interface. For multiple outputs, each output is treated as if it were a single output, generating new annotations for each output annotation available.

Multiple Input – Single Output. First, consider a transformation operation with multiple inputs and one output. An example is the overlayMapImages of our running example, which receives two GeoTIFF images as input and produces one GeoTIFF image. Figure 4 illustrates this case. Each input data set (and also, each input expected at the interface) has its associated annotation. To proceed with the propagation, it is necessary to combine the annotations on the input data (M_1, M_2, M_3, in the figure) into a single annotation (M). Likewise, the annotation on the inputs of the interface (M_{i1}, M_{i2}, M_{i3}) must be combined into another single annotation (M_i).

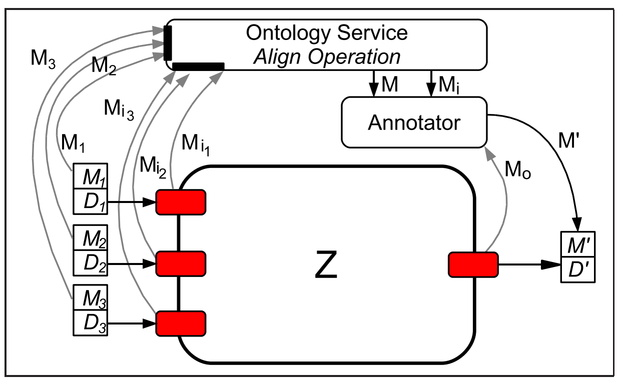

Figure 4: Dealing with multiple inputs.

This is accomplished by executing pairwise ontology alignment on the annotations, finally generating the two results over which the algorithm of Section 3.5 will be applied.

Single Input – Multiple Output. Here, there are basically two situations. First, the same output is replicated, to be fed into several different operations, which is actually not a multiple output case. Second, several different outputs are generated, representing different results from a transformation operation. In the first case, the propagation mechanism is executed as if for a single output and the result is replicated to each destination of the generated output.

In the second case, the propagation mechanism must be applied separately for each output of the operation's interface, generating a different annotation to each result, as shown in Figure 5, generating, respectively, M'_1, M'_2, and M'_3. This is equivalent to breaking the operation into several operations, each having the same input interface and input data, but only one of the outputs from the original operation.

Multiple Input – Multiple Output. This combines the strategies of the other cases. The inputs

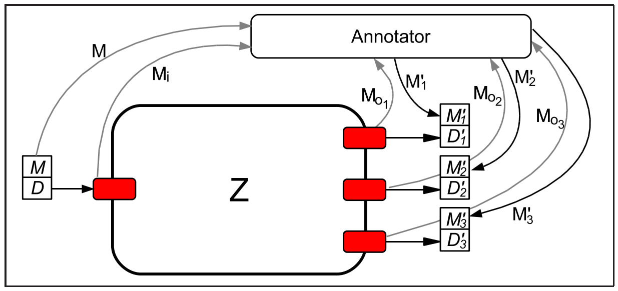

Figure 5: Multiple outputs and multiple propagations.

are combined to generate single annotations M and $\mathsf{M_i}$. These combined annotations are then used as if in a single input multiple output case.

5 Composition of Operations

Section 3 considered the problem of annotation propagation for one transformation with single input/output, extended in Section 4 to cover transformations with multiple inputs/outputs. The last issue is how to deal with a composite transformation process.

We model processes as workflows and transformations as workflow activities (e.g., the workflow of the running example). The execution of a process is carried out by a WFMS, which invokes operations to execute the specified activities. Each activity invocation is followed by an execution of the algorithm of Section 3.5.

The order in which the activities are executed is determined by the abstract workflow specification. The actual execution is done via workflow instantiations, which preserve the structure from the abstract version and assign data inputs to the workflow and actual operations to each activity. Workflow execution always unfold in an acyclic sequence of executed steps. Since annotation propagation is executed side by side with activity execution, the existence of cycles in the workflow does not hamper the propagation. Though cycles do not pose an issue, dealing with parallelism (splitting, synchronization, merging) requires attention.

Flow splitting (forking), synchronization, and merging are all controlled by the WFMS. However, a few situations might require special care with respect to the annotation propagation mechanism.

A fork can be treated as a single input/multiple output case, and thus solved by the approach described in Section 4. For a join, there are three possibilities: (i) it feeds an activity with a single input; (ii) it feeds an activity with multiple inputs (one of which is the result of the join); and, (iii) it feeds an activity with multiple undefined inputs. In the first case, the result of the join is forwarded directly. In case (ii), the join is treated as if it were a virtual activity with multiple inputs and a single output, as illustrated by Figure 6. With this, a multiple input strategy can be applied before proceeding with the execution. If the join ends in one activity with multiple inputs and the arriving data is already mapped to the multiple inputs in the workflow, the multiple input strategy can be applied directly. Finally, case (iii), if there is more than one arriving data sets for one of the inputs, the strategy for the activity with a single input is used for these data sets.

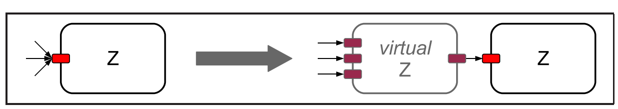

Figure 6: Creation of a virtual join activity.

In a workflow, specification of different sets of ontological relations can be associated with each activity. This allows to customise behaviour for each activity, according to the desired context, when applying the annotation propagation rules.

6 Revisiting The Example

Finally, we provide a sketch of an application of our solution, for the example in Figure 1. The process in the figure can be seen as an abstract workflow. Such workflows consist of a high level representation of a process, without being bound to any specific data or process instances. They are constructed based on input/output interface matching among the activities, not needing any knowledge about the actual data and annotation instances at execution time. The process in the figure has two inputs and produces one output. Each operation within the process receives a data set, transforms it and produces an output.

We assume that any instance of the transformation process is steered by a WFMS, which controls the data flow by invoking the activities in the appropriate order and by handling and forwarding the data. Figure 7 illustrates the execution of the workflow in Figure 1, highlighting workflow activity classifyMapImage, being executed. The bottom part of the figure shows that, at execution time, abstract transformation classifyMapImage is materialized (instantiated) into the executable operation A.

Operation A received the data set D and transformed it, generating D'. Annotation propagation is achieved as follows. The WFMS invokes the annotator service, passing as parameters the data annotations M, the input interface annotations M_i, and the output interface annotations M_o. The annotator executes a set of rules that examines an annotation and produces another annotation. In the figure, it receives annotations M, M_i and M_o and produces M', the derived annotation. The pair (D', M') can then be passed on to the next workflow steps. Ontology access and management take advantage of the Aondê Ontology Web Service [19]. It provides access, via Web services, to the most common functions available in ontology toolkits. Ontologies are stored in Ontology Repositories, built and managed by the service, organized into ontology and metadata data spaces. Aondê can be invoked by client applications, using SOAP-compliant messages, to find, rank and compare ontologies of interest, to correlate distinct ontologies to create views, and build new ontologies.

Derived annotations are thus automatically available at the end of the last step of a data transformation process.

As to the application of the propagation rules, consider again the example of Figure 3, describing the map classification operation, and the propagation strategies in Table 1. For the CMM_iM_o propagation, we have the following[4]:
$\Omega =$ *{(id_D,hasEncoding,GeoTIFF-bitmap),(id_D,RDFType,NDVI),*
(id_D,hasResolution,"250"), (id_D,hasTimestamp,"20010323")},
$\Theta =$ *{(id_O,hasEncoding,GeoTIFF-bitmap), (id_O,RDFType,ClassifiedVI)}*, and
$\Delta =$ *{(id_D',hasEncoding,GeoTIFF-bitmap), (id_D',RDFType,ClassifiedVI)}*.
The encoding was propagated through the *class equivalence* ontological relation, and the image type was propagated through the *class specialization* ontological relation. The other propagation strategies are similarly applied.

A large percentage of research on semantic enhancement and search based on ontologies considers that annotations are provided by one single ontology – i.e., users are concerned with the same kind of knowledge domains (see, for instance, the survey presented in [43] on 22 semantic search

[4]The strings with quoted values are literals, while those without them represent URI that were omitted for space saving.

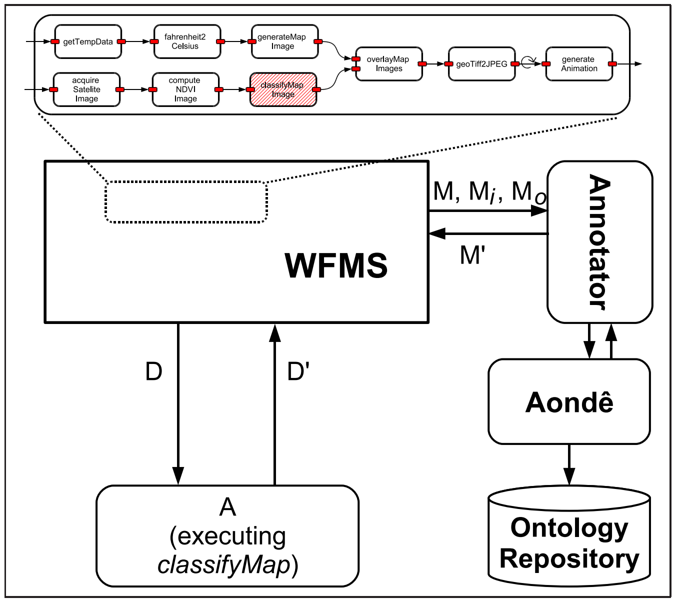

Figure 7: A sample execution.

approaches). However, multimedia data usage and interpretation is highly context dependent. Thus, a given multimedia data set may need distinct semantic annotations. Our solution supports this need at two different moments:

(1) *Flexible annotation structure*. Given semantic annotation units are RDF triples, users may easily at any time attach new units, enriching contextual information. Users may also delete or modify units, at the cost of information loss. An alternative would be to maintain information on versions of annotations (e.g., using a versioning mechanism such as proposed in [15]). This is however, matter of future work.

(2) *Possibility of manipulating ontologies*. Annotation propagation may require checking more than one ontology (e.g., when input data set D is annotated by one research group and interfaces I and O are annotated by researchers from another group, and those groups use distinct ontologies). This may require the construction of a new ontology (for instance, by aligning the two ontologies) or comparison of ontologies (e.g., via difference). This kind of possibility is supported in Figure 7 by invocation of Aondê, a Web service that can perform such operations. This kind of functionality is a must whenever one wants to take context into account in multimedia data manipulation.

7 Related Work

This paper is motivated by the increasing need, in scientific applications, to effectively manage multimedia data. Such data must be interpreted in a specific context including research goals, application domain, and spatio-temporal framework. Related work involves therefore examples of multimedia data manipulation in e-Science and annotation issues.

Our running example concerns the use of images and animations in agriculture. Image processing and content-based retrieval is also increasingly used in areas such as medicine [63], or biodiversity [16, 61]. Though most e-Science-related studies are restricted to image and/or video management, sound data are also used in biology (e.g., in studying bird communication [40]), in designing environments for acoustic comfort or in seafloor surveying with acoustic remote sensing. All these studies emphasise the need for accurate metadata, e.g., to assess data quality and to support scientific interpretation.

Metadata have been used in several contexts [2, 14, 55], including multimedia [21, 24, 29, 38, 41, 47, 48, 57]. Particularly, they have proven useful in establishing means of assigning descriptions to (multimedia) content and execution and user interaction environments. The works of [6, 53] argue that descriptions of (i) multimedia content and (ii) users' preferences related to multimedia content are essential to achieve what they call *universal multimedia access* (UMA). In UMA, the multimedia content should be available anywhere and anytime, possibly using content adaptation to achieve this. A similar vision is shared by [1] who discusses the issues arising from ambient intelligence; by [59] who describes how user input, sensor readings, available media and software are needed to achieve such an environment; and by [3] who analyses this scenario for distance learning. This view can be generalised to context description, using metadata to describe the whole environment along with the content manipulated.

However, there have been some difficulties in using metadata. The work of [56] analyses the trade-offs of using metadata in several scenarios, including appropriate uses where the environment is data-driven and/or requires analytical decision making, and less appropriate uses when considering more intuitive or or politically charged environments. It also points out the importance of metadata quality and completeness in the successful use of metadata. The work of [13] discusses lack of motivation to go through the tedious process of creating metadata, discussing the case for MPEG-7 and its commercial motivation as a reason for relative success. These works evidence the need for more automation on the generation of metadata, which is the main concern of our solution.

Other papers concern the adaptation of metadata to be used on the Web. Particularly, these initiatives aim at semantics-based or conceptual models, often following standards of the Semantic Web [25–27, 36, 49–51, 58, 62, 66]. The application of Semantic Web standards to describe multimedia content involves the production of semantic annotations and thus could profit from our annotation propagation solution. On another front, work with Semantic Web Services (e.g., [52, 65]) provide the possibility of semantic annotations on interfaces of operations, thereby meeting the second requirement of our solution (i.e., describing the interfaces of operations).

Work on annotation propagation has many aspects. It is important to notice that our use of the term "metadata evolution" does not directly correspond to ontology evolution, as used by some authors [60]. Nor does it regard updating metadata to reflect changes on the models of the real world, as in [46]. While these proposals consider the evolution of annotations with time, we deal with annotation transformation processes. Other uses of the term "annotation propagation" [32, 34, 39, 42] regard the automated update of annotations in a hierarchy (e.g., if a group receives an annotation, all entities from that group receive it as well; or if an annotation is corrected, all related annotations gets corrected as well). This kind of work is not concerned with data transformations and their impact on the annotations. Yet another approach is to use content-based retrieval to identify which multimedia items could receive the same annotations from previously annotated ones [17].

To the best of our knowledge, there are two approaches in which the notion of "annotation propagation" is the same as ours. The first solution is presented in [8]. Their solution is placed in a data warehouse environment, with the annotations stored in extra (data) columns. The paper presents rules for propagating these annotations fields to answer queries. Rule implementation is based on pSQL, an extension of SQL that supports a *propagate* clause to enable the application of propagation schemes. Query processing is wrapped in pre-processing and post-processing modules that deal with the extended part of the query. The paper discusses three propagation schemes. In the first, the annotation of the source field is copied to the query answer field of the corresponding row. The second scheme considers propagating the annotation regardless of how the query is posed, always propagating the same annotations for a given query, no matter how it is formulated. The third scheme allows the user to specify how the propagation should be performed. The solution of [8] is restricted to use within databases and data warehousing environments, being limited by the query language. It only considers fine-grained annotations, in an item by item basis. Our solution, on the other hand can be applied to any kind of transformation and allows any granularity on the annotations, provided that both the transformations and the data are described with semantic annotations.

The second proposal [11] solves the annotation propagation problem taking advantage of schema mapping compositions. A schema mapping links fields from one database schema s_1 to fields of another database schema s_2, establishing a correspondence between them (m_{12}). If there is a third schema s_3 and a mapping m_{23} between s_2 and s_3, a schema mapping composition consists in finding the composition of m_{12} and m_{23} into another mapping m_{13} between the schemas s_1 and s_3. Their solution is restricted to relational algebra operations and is applicable to sequences of transformations modelled as workflows. Using the same translation approach of [11], the solution in [7] could also be applied to solve the annotation propagation problem. Similar to [11], we also annotate content using ontologies, without restriction to database systems environments.

In addition, the work of [18] proposes a mechanism for tracing transformations within the context of data warehouses. They present aspects of dealing with general transformations, including dealing with multiple inputs/outputs and general sequences (or graphs) of data transformations. These issues were also dealt with by us in our mechanism, but in the context of annotation propagation.

As far as we know, ours is the only solution that considers general data transformation operations. It is also the only one that considers different levels of propagation based on the matching of previous descriptions and on the description of the effect of applying the operation. It is also the only solution concerned with propagating unmatched annotations (using open propagation rules – see Sections 2.4 and 3.6).

8 Concluding Remarks

This paper presented a new approach to annotation propagation on multimedia data sets. A solution to the annotation propagation problem offers several advantages, such as: (i) lessening annotation efforts, (ii) decreasing the loss of information along the transformation process, (iii) documenting data origins (for traceability), and (iv) providing quality information. This paper focused on the first two issues.

Our approach allows combining content-based metadata with contextual information provided by ontologies. Solutions to the annotation propagation problem have so far been restricted to database operations. Our definition showed how to extrapolate a solution to the problem that encompasses general transformation operations. Rather than having to consider the operations themselves, our propagation mechanism just needs to know the input/output interfaces of the operation. This is compatible with Semantic Web Services efforts and follows software engineering encapsulation principles.

The classification introduced in the paper, reflected in open and closed propagation rules, clarified the potential and risks of propagating annotations with no verifiable validity. It also shows how the mapping between input and output operation interfaces can result in more specific propagations.

The solution to the annotation propagation problem presented is general enough to be applied in service-based environments. It can also be applied in more specific or controlled environments, such as a database system or a digital library system. Our mechanism can be extended by increasing the number of ontological relations. This allows tailoring annotation propagation behaviours, permitting new specific real world relationships to be considered.

As future work, we intend to investigate annotation propagation considering particular outcomes of each data manipulation function, e.g., content-based annotation propagation generating annotations similar to the ones used in [54, 64, 67]. Parts of the output could be annotated individually with different annotations. This means that the annotation propagation mechanism could generate different annotations depending not only on the previous annotations and operation interfaces, but also on the resulting data set.

We also plan to investigate how to deal with the possibility of errors in open propagation rules. Using more specific ontology inference mechanisms could help evaluate the possibility of errors. Furthermore, execution logs with human validation could be used to automatically create extra annotations to evidence how annotations not directly involved were considered. This helps in decreasing the number of errors.

Acknowledgements

This work was supported by FAPESP (grant 2004/14052-3) and partially financed by CNPq.

A Other Ontological Relations

This appendix shows the treatment given to other ontological relations. **A1** and **A2** are relations among classes; **A3** and **A4** are relations among instances; **A5** through **A9** are relations among properties.

A1. Class complement. Relation based on the owl:complementOf construct, which represents a class with all properties that are not part of the original class.

$$\mathscr{R}(a,b) = \begin{cases} \texttt{<sa,pa,oa>,<sb,pb,ob>} & \text{if } \texttt{sa} \textit{ complementOf } \texttt{sb} \\ \texttt{<s,pa,oa>,<sb,pb,s>} & \text{if } \texttt{sa} \textit{ complementOf } \texttt{ob} \\ & \text{where } s = sa \cup ob. \\ \texttt{<sa,pa,s>,<s,pb,ob>} & \text{if } \texttt{oa} \textit{ complementOf } \texttt{sb} \\ & \text{where } s = sb \cup oa. \\ \texttt{<sa,pa,oa>,<sb,pb,ob>} & \text{if } \texttt{oa} \textit{ complementOf } \texttt{ob} \end{cases}$$

A2. Class disjointedness. Relation based on the owl:disjointWith, which states that the two classes do not share individuals or properties.

$$\mathscr{R}(a,b) = \emptyset$$

A3. Instance equivalence. Relation based on the owl:sameAs, which states that two instances (represented by different URIs) are, in fact, the same. Note that owl:sameAs encompasses owl:equivalentClass and owl:equivalentProperty as the distinction between class, instance and property is not necessary in OWL.

$$\mathscr{R}(a,b) = \begin{cases} \texttt{<sa,pa,oa>} & \text{if } \texttt{sa} \textit{ sameAs } \texttt{sb} \\ \texttt{<sa,pa,oa>,<sb,pb,ob>} & \text{if } \texttt{sa} \textit{ sameAs } \texttt{ob} \\ \texttt{<sa,pa,oa>,<sb,pb,ob>} & \text{if } \texttt{oa} \textit{ sameAs } \texttt{sb} \\ \texttt{<sa,pa,oa>} & \text{if } \texttt{oa} \textit{ sameAs } \texttt{ob} \end{cases}$$

A4. Instance type. Relation based on the rdf:type predicate, which associates a given type (represented by the object) to the subject.

$$\mathscr{R}(a,b) = \begin{cases} \texttt{<sa,pa,oa>} & \text{if } \texttt{sa} \textit{ RDFtype } \texttt{sb} \text{ and} \\ & \texttt{pa} \textit{ equivalentProperty } \texttt{pb} \\ & \text{and } \texttt{oa} \textit{ RDFtype } \texttt{ob} \end{cases}$$

A5. Property generalization. Relation based on rdfs:subPropertyOf, which defines that P is a subproperty of Q, than all that is valid for Q must also be valid for P.

$$\mathscr{R}(a,b) = \begin{cases} \texttt{<sb,pb,ob>} \\ \text{if } \texttt{pb} \textit{ subPropertyOf } \texttt{pa} \text{ and} \\ \texttt{sa} \textit{ equivalentClass } \texttt{sb} \text{ and } \texttt{oa} \textit{ equivalentClass } \texttt{ob} \\ \\ \texttt{<sb,pb,ob>} \\ \text{if } \texttt{pb} \textit{ subPropertyOf } \texttt{pa} \text{ and} \\ \texttt{sa} \textit{ sameAs } \texttt{sb} \text{ and } \texttt{oa} \textit{ sameAs } \texttt{ob} \\ \\ \texttt{<sb,pb,ob>} \\ \text{if } \texttt{pb} \textit{ subPropertyOf } \texttt{pa} \text{ and} \\ \texttt{sb} \textit{ subClassOf } \texttt{sa} \text{ and } \texttt{ob} \textit{ subClassOf } \texttt{oa} \end{cases}$$

A6. Property specialization. This relation is also based on the rdfs:subPropertyOf construct, with reversed representation.

$$\mathscr{R}(a,b) = \begin{cases} \texttt{<sa,pa,oa>} \\ \text{if } \texttt{pa} \textit{ subPropertyOf } \texttt{pb} \text{ and} \\ \texttt{sa} \textit{ equivalentClass } \texttt{sb} \text{ and } \texttt{oa} \textit{ equivalentClass } \texttt{ob} \\ \\ \texttt{<sa,pa,oa>} \\ \text{if } \texttt{pa} \textit{ subPropertyOf } \texttt{pb} \text{ and} \\ \texttt{sa} \textit{ sameAs } \texttt{sb} \text{ and } \texttt{oa} \textit{ sameAs } \texttt{ob} \\ \\ \texttt{<sa,pa,oa>} \\ \text{if } \texttt{pa} \textit{ subPropertyOf } \texttt{pb} \text{ and} \\ \texttt{sa} \textit{ subClassOf } \texttt{sb} \text{ and } \texttt{oa} \textit{ subClassOf } \texttt{ob} \end{cases}$$

A7. Property equivalence. Relation based on the owl:equivalentProperty, which defines that P is a equivalent to Q, than all that is valid for Q must also be valid for P, and vice-versa.

$$\mathscr{R}(a,b)=\begin{cases}\texttt{<sa,pa,oa>,<sa,pa,ob>}\\ \text{if } \texttt{pa} \textit{ equivalentProperty } \texttt{pb} \text{ and}\\ \texttt{sa} \textit{ equivalentClass } \texttt{sb}\\ \\ \texttt{<sa,pa,oa>,<sa,pa,ob>}\\ \text{if } \texttt{pa} \textit{ equivalentProperty } \texttt{pb} \text{ and}\\ \texttt{sa} \textit{ sameAs } \texttt{sb}\\ \\ \texttt{<sa,pa,oa>,<sb,pa,oa>}\\ \text{if } \texttt{pa} \textit{ equivalentProperty } \texttt{pb} \text{ and}\\ \texttt{oa} \textit{ equivalentClass } \texttt{ob}\\ \\ \texttt{<sa,pa,oa>,<sb,pa,oa>}\\ \text{if } \texttt{pa} \textit{ equivalentProperty } \texttt{pb} \text{ and}\\ \texttt{oa} \textit{ sameAs } \texttt{ob}\end{cases}$$

A8. Property domain. Relation based on the rdfs:domain, which lists the classes which can be the subject of the property.

$$\mathscr{R}(a,b)=\begin{cases}\texttt{<s,pa,oa>,<s,pb,ob>} & \text{if } \texttt{sa} \textit{ domain } \texttt{pb}\\ & \text{where } s=sa\cup sb\\ \texttt{<s,pa,oa>,<s,pb,ob>} & \text{if } \texttt{sb} \textit{ domain } \texttt{pa}\\ & \text{where } s=sa\cup sb\end{cases}$$

A9. Property range. Relation based on the rdfs:range, which lists the classes and/or data ranges which can be the object of the property.

$$\mathscr{R}(a,b)=\begin{cases}\texttt{<sa,pa,o>,<sb,pb,o>} & \text{if } \texttt{oa} \textit{ range } \texttt{pb}\\ & \text{where } o=oa\cup ob\\ \texttt{<sa,pa,o>,<sb,pb,o>} & \text{if } \texttt{ob} \textit{ range } \texttt{pa}\\ & \text{where } o=oa\cup ob\end{cases}$$

References

[1] E. Aarts. Ambient Intelligence: A Multimedia Perspective. *IEEE MultiMedia*, 11(1):12–19, 2004.

[2] M. Agosti and N. Ferro. A Formal Model of Annotations of Digital Content. *ACM Transactions on Information Systems*, 26(1):3, 2007.

[3] M. Almaoui, A. Kushki, and K. N. Plataniotis. Metadata-driven multimedia transcoding for distance learning. *Multimedia Systems*, 12(6):505–520, 2007.

[4] G. Antoniou and F. van Harmelen. Web Ontology Language: OWL. In S. Staab and R. Studer, editors, *Handbook on Ontologies in Information Systems*, pages 76–92, 2003.

[5] S. Bechhofer, L. Carr, C. Goble, S. Kampa, and T. Miles-Board. The Semantics of Semantic Annotation. In *Proc. 1st International Conference on Ontologies, Databases, and Applications of Semantics for Large-Scale Information Systems (ODBASE) – LNCS 2519*, pages 1152–1167, 2002.

[6] P. Beek, J. R. Smith, T. Ebrahimi, T. Suzuki, and J. Askelof. Metadata-driven Multimedia Access. *IEEE Signal Processing Magazine*, 20(2):40–52, 2003.

[7] P. A. Bernstein, T. J. Green, S. Melnik, and A. Nash. Implementing mapping composition. *The VLDB Journal*, 17(2):333–353, 2008.

[8] D. Bhagwat, L. Chiticariu, W. Tan, and G. Vijayvargiya. An annotation management system for relational databases. *The VLDB Journal*, 14(4):373–396, 2005.

[9] K. Böhm and T. C. Rakow. Metadata for Multimedia Documents. *SIGMOD Record*, 23(4):21–26, 1994.

[10] S. Boutemedjet and D. Ziou. A Graphical Model for Context-Aware Visual Content Recommendation. *IEEE Transactions on Multimedia*, 10(1):52–62, 2008.

[11] S. Bowers and B. Ludäscher. A Calculus for Propagating Semantic Annotations Through Scientific Workflow Queries. In *EDBT Workshops*, pages 712–723, 2006.

[12] F. P. Bretherton and P. T. Singley. Metadata: a User's View. In *Proc. 7th International Working Conference on Scientific and Statistical Database Management*, pages 166–174, 1994.

[13] D. C. A. Bulterman. Is It Time for a Moratorium on Metadata? *IEEE Multimedia*, 11(4):10–17, 2004.

[14] W. Cathro. Metadata: An Overview. In *Standards Australia Seminar: Matching Discovery and Recovery*, 1997. `http://www.nla.gov.au/nla/staffpaper/cathro3.html` (as of Apr 2008).

[15] W. Cellary and G. Jomier. Consistency of Versions in Object-Oriented Databases. In *Proc. 16th International Conference on Very Large Data Bases*, pages 432–441, 1990.

[16] Y. Chen, H. L. Bart Jr, and F. Teng. A Content-Based Image Retrieval System for Fish Taxonomy. In *Proc. 7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 237–244, 2005.

[17] J.-P. Chevallet, N. Maillot, and J.-H. Lim. Concept Propagation Based on Visual Similarity. In *Proc. 3rd Asia Information Retrieval Symposium*, 2006.

[18] Y. Cui and J. Widom. Lineage tracing for general data warehouse transformations. *The VLDB Journal*, 12(1):41–58, 2003.

[19] J. Daltio and C. B. Medeiros. Aondê: An Ontology Web Service for Interoperability across Biodiversity Applications. *Information Systems*, 2008. Accepted for publication, 46 pages, doi:10.1016/j.is.2008.02.001.

[20] DarwinCore (DwC) Biodiversity Information Standards (TDWG) working group. The Darwin Core Standard. `http://www.tdwg.org/activities/darwincore/` (as of Apr 2008).

[21] G. Davenport and M. Murtaugh. ConText: Towards the Evolving Documentary. In *Proc. 3rd ACM International Conference on Multimedia*, pages 381–389, 1995.

[22] N. Dimitrova. Context and Memory in Multimedia Content Analysis. *IEEE MultiMedia*, 11(1):7–11, 2004.

[23] E. Duval, W. Hodgins, S. Sutton, and S. L. Weibel. Metadata Principles and Practicalities. *D-Lib Magazine*, 8(4), 2002.

[24] K. Falkovych and F. Nack. Context aware guidance for multimedia authoring: harmonizing domain and discourse knowledge. *Multimedia Systems*, 11(3):226–235, 2006.

[25] M. Ferecatu, N. Boujemaa, and M. Crucianu. Semantic interactive image retrieval combining visual and conceptual content description. *Multimedia Systems*, 13(5-6):309–322, 2008.

[26] Alfio Ferrara, Luca A. Ludovico, Stefano Montanelli, Silvana Castano, and Goffredo Haus. A Semantic Web Ontology for Context-Based Classification and Retrieval of Music Resources. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2(3):177–198, 2006.

[27] R. Garcia and O. Celma. Semantic Integration and Retrieval of Multimedia Metadata. In *Proc. 5th International Workshop on Knowledge Markup and Semantic Annotation*, 2005.

[28] F. Geerts, A. Kementsietsidis, and D. Milano. MONDRIAN: Annotating and Querying Databases through Colors and Blocks. In *Proc. 22nd International Conference on Data Engineering*, 2006.

[29] R. Goularte, M. G. C. Pimentel, and E. S. Moreira. Context-aware support in structured documents for interactive-TV. *Multimedia Systems*, 11(4):367–382, 2006.

[30] T. R. Gruber. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies*, 43(5-6):907–928, 1995.

[31] K. Haase. Context for Semantic Metadata. In *Proc 12th ACM International Conference on Multimedia*, pages 204–211, 2004.

[32] P. Herrera, O. Celma, J. Massaguer, P. Cano, E. Gómez, F. Gouyon, and M. Koppenberger. Mucosa: A Music Content Semantic Annotator. In *Proc. 6th International Conference on Music Information Retrieval*, pages 77–83, 2005.

[33] International Organization for Standardization – TC 211. ISO19115:2003 – Geographic Information – Metadata. `http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26020` (as of Apr 2008).

[34] F. Kang and M. R. Naphade. A Generalized Multiple Instance Learning Algorithm for Iterative Distillation and Cross-Granular Propagation of Video Annotations. In *Proc. IEEE International Conference on Image Processing*, pages II–205–208, 2007.

[35] V. Kashyap and A. Sheth. Semantic Heterogeneity in Global Information Systems: The Role of Metadata, Context and Ontologies. In M. P. Papazoglou and G. Schlageter, editors, *Cooperative Information Systems*, pages 139–178. Academic Press, 1998.

[36] Latifur Khan, Dennis McLeod, and Eduard H. Hovy. Retrieval effectiveness of an ontology-based model for information selection. *VLDB Journal*, 13(1):71–85, 2004.

[37] M. L. Kherfi and D. Ziou. Image Collection Organization and Its Application to Indexing, Browsing, Summarization, and Semantic Retrieval. *IEEE Transactions on Multimedia*, 9(4):893–900, 2007.

[38] H. Kosch, L. Boszormenyi, M. Doller, M. Libsie, P. Schojer, and A. Kofler. The Life Cycle of Multimedia Metadata. *IEEE Multimedia*, 12(1):80–86, 2005.

[39] G. Langs, P. Peloschek, R. Donner, and H. Bischof. Annotation Propagation by MDL Based Correspondences. In *Proc. Computer Vision Winter Workshop 2006*, 2006.

[40] T. Lengagne, J. Lauga, and T. Aubin. Intra-syllabic acoustic signatures used by the king penguin in parent-chick recognition: an experimental approach. *Journal of Experimental Biology*, 204(4):663–672, February 2001.

[41] S. Little, M. Martinelli, O. Salvetti, U. Gudukbay, O. Ulusoy, G. Chalendar, and G. Grefenstette. Integration of Structural and Semantic Models for Multimedia Metadata Management. In *Proc. International Workshop on Content-Based Multimedia Indexing*, pages 40–45, 2007.

[42] P.-H. Luong and R. Dieng-Kuntz. A Rule-Based Approach for Semantic Annotation Evolution. *Computational Intelligence*, 23(3):320–338, 2007.

[43] C. Mangold. A survey and classification of semantic search approaches. *International Journal of Metadata, Semantics and Ontologies*, 2(1):23–34, 2007.

[44] F. Manola, E. Miller, and B. McBride. RDF Primer, February 2004. `http://www.w3.org/TR/rdf-primer/` (as of Apr 2008).

[45] C. C. Marshall. Towards an Ecology of Hypertext Annotation. In *Proc. 9th ACM Conference on Hypertext and Hypermedia*, pages 40–49, 1998.

[46] P. Missier, P. Alper, O. Corcho, I. Dunlop, and C. Goble. Requirements and Services for Metadata Management. *IEEE Internet Computing*, 11(5):17–25, 2007.

[47] R. P. Morris. System for Automatically Creating a Metadata Repository for Multimedia, 2005. Patent WIPO Pub. No. WO/2006/057738 — USPTO Pub. App. No. 20060112141 – `http://www.wipo.int/pctdb/en/wo.jsp?wo=2006057738` (as of Apr 2008).

[48] Moving Picture Experts Group (MPEG) – ISO/IEC-JTC1. ISO/IEC15938 (parts 1–10) Multimedia content description interface. `http://www.iso.org/iso/search.htm?qt=15938&sort=rel&type=simple&published=true` (as of Apr 2008).

[49] M. Naphade, J. R. Smith, J. Tesic, S. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale Concept Ontology for Multimedia. *IEEE Multimedia*, 13(3):86–91, 2006.

[50] F. Nack J. Ossenbruggen and L. Hardman. That Obscure Object of Desire: Multimedia Metadata on the Web, Part 2. *IEEE Multimedia*, 12(1):54–63, 2005.

[51] J. Ossenbruggen, F. Nack, and L. Hardman. That Obscure Object of Desire: Multimedia Metadata on the Web, Part 1. *IEEE Multimedia*, 11(4):38–48, 2004.

[52] OWL-S (The DAML Program). OWL-based Web Service Ontology. `http://www.daml.org/services/owl-s/` (as of Apr 2008).

[53] F. Pereira and I. Burnett. Universal Multimedia Experiences for Tomorrow. *IEEE Signal Processing Magazine*, 20(2):63–73, 2003.

[54] N. Rasiwasia, P. L. Moreno, and N. Vasconcelos. Bridging the Gap: Query by Semantic Example. *IEEE Transactions on Multimedia*, 9(5):923–938, 2007.

[55] A. Sen. Metadata management: past, present and future. *Decision Support Systems*, 37(1):151–173, 2004.

[56] G. Shankaranarayanan and A. Even. The Metadata Enigma. *Communications of the ACM*, 49(2):88–94, 2006.

[57] A. F. Smeaton. Content vs. Context for Multimedia Semantics: The Case of SenseCam Image Structuring. In *Proc. 1st International Conference on Semantics and Digital Media Technologies (LNCS4306)*, pages 1–10, 2006.

[58] G. Stamou, J. Ossenbruggen, J. Z. Pan, G. Schreiber, and J. R. Smith. Multimedia Annotations on the Semantic Web. *IEEE Multimedia*, 13(1):86–90, 2006.

[59] S. P. Stenton, R. Hull, P. M. Goddi, J. E. Reid, B. J. C. Clayton, T. J. Melamed, and S. Wee. Mediascapes: Context-Aware Multimedia Experiences. *IEEE Multimedia*, 14(3):98–105, 2007.

[60] L. Stojanovic, N. Stojanovic, and S. Handschuh. Evolution of the Metadata in the Ontology-based Knowledge Management Systems. In *Proc. 1st German Workshop on on Experience Management*, pages 65–77, 2002.

[61] R. S. Torres, C. B. Medeiros, M. A. Gonçalves, and E. A. Fox. A Digital Library Framework for Biodiversity Information Systems. *International Journal on Digital Libraries*, 6(1):3 – 17, February 2006.

[62] R. Troncy, W. Bailer, M. Hausenblas, P. Hofmair, and R. Schlatte. Enabling Multimedia Metadata Interoperability by Defining Formal Semantics of MPEG-7 Profiles. In *Proc. 1st International Conference on Semantics and Digital Media Technologies (LNCS4306)*, pages 41–55, 2006.

[63] S. K. Tzelepi, D. K. Koukopoulos, and G. Pangalos. A flexible content and context-based access control model for multimedia medical image database systems. In *Proc. Workshop on Multimedia and Security*, pages 52–55, 2001.

[64] Nuno Vasconcelos. From Pixels to Semantic Spaces: Advances in Content-Based Image Retrieval. *IEEE Computer*, 40(7):20–26, 2007.

[65] W3C. Semantic Annotations for Web Service Description Language (SAWSDL). `http://www.w3.org/2002/ws/sawsdl` (as of Apr 2008).

[66] H. Wang, S. Liu, and L.-T. Chia. Image retrieval with a multi-modality ontology. *Multimedia Systems*, 13(5-6):379–390, 2008.

[67] J. Yuan, J. Li, and B. Zhang. Exploiting spatial context constraints for automatic image region annotation. In *Proc. 15th International Conference on Multimedia*, pages 595–604, 2007.