Multimedia Semantic Annotation Propagation

Gilberto Zonta Pastorello Jr Jaudete Daltio Claudia Bauzer Medeiros
Institute of Computing
University of Campinas – UNICAMP
13084-971, Campinas, SP, Brazil
gilberto@ic.unicamp.br jaudete@gmail.com cmbm@ic.unicamp.br

Abstract

Scientific research is producing and consuming large volumes of multimedia data at an ever growing rate. Annotations to the data helps associating context and enhances content management, making it easier to interpret and share data. However, raw data often needs to go through complex processing steps before it can be consumed. During these transformation processes, original annotations from the production phase are often discarded or ignored, since their usefulness is usually limited to the first transformation step. New annotations must be associated with the final product, a time consuming task often carried out manually. Systematically associating new annotations to the result of each data transformation step is known as annotation propagation*. This paper introduces techniques for structuring annotations by applying references to ontologies and automatically transforming these annotations along with 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; and (ii) a set of transformations rules, based on ontological relations.*

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 and annotations have been used as the primary means of describing data sets. 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. Annotations, on the other hand, are more flexible, often representing personal remarks created by data producers or consumers, but are also more limited when considering management tasks.

Albeit helpful in improving data interpretation, metadata/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/annotations. Roughly speaking, this characterizes the scenario for *metadata evolution* or *annotation propagation* [4, 5]. This poses the following problems: (1) how to propagate relevant metadata/annotations that would otherwise be discarded during a transformation? and, (2) how to support automatic creation of metadata/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/annotations 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. Terms from ontologies help provide contextual information.

Our approach uses semantic annotations both from data and from operations that transform the data, i.e., we assume these annotations are available. We then 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.

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 shows an example that we use to illustrate our proposal. Section 3 defines the annotation propagation problem. Section 4 presents our solution to the problem using semantic annotations. Section 5 discusses related work. Section 6 presents conclusions and future work.

2 Motivating 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 JPEG images 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; and,
(5) convert the resulting map into a JPEG image.

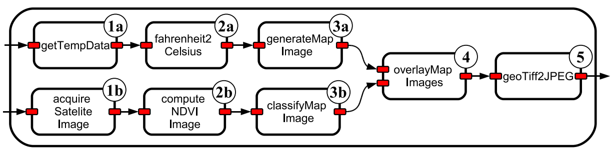

Figure 1. Data transformation process.

The resulting images illustrate 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.

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 images have no associated annotations.

[1]An image format where each pixel corresponds to a given geographical coordinate.

[2]Normalized Difference Vegetation Index (NDVI) indicates the levels of live green vegetation, usually over satellite images.

3 The Annotation Propagation Problem

3.1 Semantic Annotations

We combine characteristics of metadata and annotations into *semantic annotations*: using the structure from the first, filling its contents with references to ontologies, which provide the flexibility of the latter. Based on RDF structuring, 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. For space saving we omit the namespaces for terms in the text and figures.

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 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 M_i and its output interface O described by M_o.

[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.

3.2 Annotation Propagation

Let (T, I, O, D, D') denote an application of a data transformation operation T which has an input interface I and an output interface O, and is applied on a data set D, resulting in derived data D'. The definition for T was adapted from [7]. 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), $\mathsf{M_o}$ (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.

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 2 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 2 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 one 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 $(\mathsf{D'}, \mathsf{M'})$.

The top of the figure illustrates an ontology repository [8], a data space containing domain ontologies with the contextual and semantic information. The graphs in the boxes

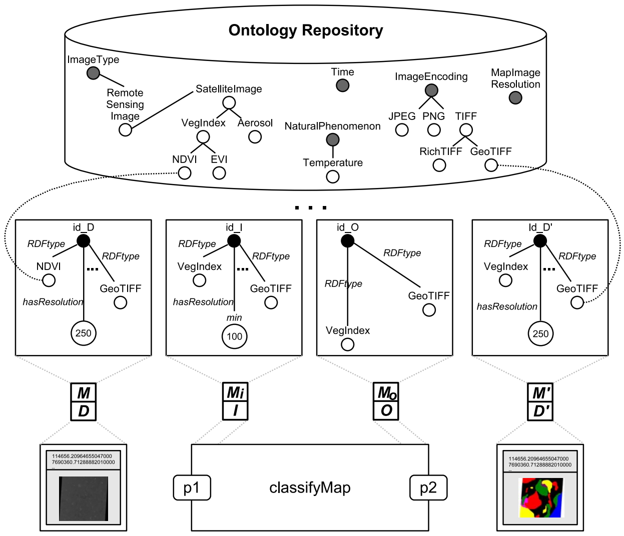

Figure 2. Data and interface annotations.

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. A consistent combination of these annotations (M, $\mathsf{M_i}$ and $\mathsf{M_o}$) should be used to generate the resuting propagated annotation (M').

4 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 annotations. 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.

4.1 Propagation Rules

Consider a transformation (T, I, O, D, D') with interface annotations M_i and M_o and let (D,M) and (D',M') be, respectively, its input and output data and annotations. The propagation problem can be simplified into the following issue: which mappings can be done between M and M_o, meaning what were the effects of applying T to D with respect to its annotations?

We divided our solution this problem in two parts: (i) a set of abstract propagation rules to select pairs of annotations units for comparison; 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. Our solution to the first part is a set of annotation propagation rules, presented next. Our solution to the second part is devised in Section 4.2. In this work, the annotation units are defined as RDF triples; however, other annotation units could also be used for the comparison, such as sub-paths, sub-trees or sub-graphs of the RDF graphs.

Table 1 shows a high level view of our annotation propagation algorithm. Its input includes the semantic annotations M and M_o, a set ($\mathfrak{R}$) of ontological relations ($\mathscr{R}_k$), such as the ones presented in Section 4.2. Its output is the resulting propagated semantic annotation (Δ).

```
 1.  Δ ← ∅
 2.  foreach 𝓡_k in ℜ do
 3.      𝓟_k ← ∅
 4.      Ω ← 𝓕_k(M)
 5.      Θ ← 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 1. Propagation Algorithm

Before the propagation rules can be applied, it is necessary to retrieve the annotation units from the RDF graph. To be general, lets define a deconstruction function $\mathscr{F}$ to accomplish this. The result is a set of comparable units, which is represented by Ω (for M) and Θ (for M_o), in lines 4 and 5, respectively. In this work this function simply singles out the RDF triples from the RDF graph, which are our basic comparison unit. Conversely, we also define a reconstruction function $\mathscr{F}^{-1}$ to recreate the RDF graph from the resulting unit(s) of the comparison of a pair of annotation units. In the algorithm, this is done in line 9, adding to the resulting semantic annotation Δ. The deconstruction and reconstruction steps are carried out for each relation, since each relation may be based on different comparison units. In this paper, all relations are based on annotation units as defined in Section 3.1.

The propagation rules simply enforce a systematic selection of a pair of annotation units and the application of a comparison under a given ontological relation. This is done in lines 6-8 of the algorithm. These steps are repeated for all the selected ontological relations (i.e., all $\mathscr{R}_k$ in $\mathfrak{R}$). The results for all possible pairs is then returned as the result for that relation (the variable $\mathscr{P}_k$ on the algorithm). If the annotation units do not match under the ontological relation (line 8), the result of $\mathscr{R}_k(a,b)$ is empty. Intuitively, the rules recursively consider single comparable annotations units, checking the compatibility between all pairs of units and generating new units for the resulting set.

Therefore, to use the our propagation mechanism one must specify: (i) a set ($\mathfrak{R}$) of ontological relations ($\mathscr{R}_k$), each with a template result ($\mathscr{P}_k$) specifying the outcome of the comparison of two annotation units under this relation; (ii) a set of functions $\mathscr{F}_k$ to translate M into Ω and M_o into Θ; (iii) a respective inverse $\mathscr{F}_k^{-1}$ to translate $\mathscr{P}_k$ into Δ.

4.2 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 among them will 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. Because of space limitation, only four ontological relations are listed here – see [11] for specifications of more ontological relations. The first three are on the classes category and the last one are on the instances category. It is possible to specify many other ontological relations to be used with our mechanism, specially when considering 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>`.

Class generalization. Relation based on the rdfs:subClassOf construct (defined as part of RDF Schema), which allows replacing general annotation units with more specific ones.

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

This relation should be read as: given two annotation units $a = \texttt{<sa,pa,oa>}$ and $b = \texttt{<sb,pb,ob>}$, generated by a deconstruction function for comparison under *subClassOf*, then the propagated annotation unit is the set composed by $\texttt{<sa,pb,ob>}$, 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, since generalization and specialization are both described by this ontological relation.

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

Class complement. Relation based on the owl:complementOf construct, which represents a class with all properties that are not part of the original class. When two or more units are returned, the result is the conjunction of them.

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

Instance equivalence. Relation based on the owl:sameAs, which states that two instances (represented by different URIs) are, in fact, the same.

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

5 Related Work

This paper is motivated by the increasing need in scientific applications to effectively manage multimedia data, which led to the proposal of a mechanism for annotation propagation. Related work involves therefore examples of multimedia data annotation and propagation proposals.

Metadata have been used in several contexts [1] including multimedia [9, 10]. Particularly, they have proven useful in establishing means of assigning descriptions to multimedia content, its execution and user interaction environments. The works of [3, 12] 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. 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 [13] 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 politically charged environments. It also points out the importance of metadata quality and completeness in the successful use of metadata. The work of [6] 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 its 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, often following standards of the Semantic Web [2, 14, 16]. 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[4] provide the possibility of semantic annotations on interfaces of operations, thereby meeting a requirement of our solution (i.e., describing the interfaces of operations). The Multimedia Annotation Interoperability Framework [15] also considers the problem of describing transformation operations focusing on multimedia content.

One use of the term "annotation propagation" 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). Our use of the term, however, regard data transformations and their impact on the annotations. 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 [4]. 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 the SQL query language that sup-

[4]e.g., SA-WSDL (`http://www.w3.org/2002/ws/sawsdl`).

ports a *propagate* clause to enable the application of propagation schemes. The solution of [4] is restricted to be used 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 [5] solves the annotation propagation problem taking advantage of schema mapping compositions. Their solution is restricted to relational algebra operations and is applicable to sequences of transformations modelled as workflows. Similar to [5], we also annotate content using ontologies, without restriction to database systems environments.

As far as we know, ours is the only solution that considers general data transformation operations. It is also the only one to take into account both previous data descriptions and operation interface description.

6 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. Our mechanism can also be extended by increasing the number of ontological relations. This allows tailoring annotation propagation behaviours, permitting new specific relationships to be considered.

As future work we intend to investigate less restrictive extensions to our propagation mechanism, also allowing, for instance, annotations that are not matched to be propagated. This can lead to richer annotations at the end of the propagation process. Dealing with multiple inputs/outputs, considering the annotation to an interface input and composition of operations are also aspects that we intend to explore further. Another possibility would be using RDF(S)/OWL inference rules or a reasoner instead of the propagation rules.

Acknowledgments. This work was supported by FAPESP (grant 2004/14052-3) and partially financed by CNPq. The authors would also like to thank the remarks from R. Lamparelli and R. Senra, and the insightful comments from the reviewers.

References

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

[2] R. Arndt, R. Troncy, S. Staab, L. Hardman, and M. Vacura. COMM: Designing a Well-Founded Multimedia Ontology for the Web. In *Proc. 6th Int. Semantic Web Conf. and 2nd Asian Semantic Web Conf.*, pages 30–43, 2007.

[3] 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.

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

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

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

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

[8] J. Daltio and C. B. Medeiros. Aondê: An Ontology Web Service for Interoperability across Biodiversity Applications. *Information Systems*, 33(7-8):724–753, 2008.

[9] 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.

[10] 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 Sep 2008).

[11] G. Z. Pastorello Jr, J. Daltio, and C. B. Medeiros. An Annotation Propagation Mechanism for Multimedia Content. Technical Report IC-08-017, Institute of Computing, University of Campinas, August 2008.

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

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

[14] 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.

[15] W3C Multimedia Semantics Incubator Group. Multimedia Annotation Interoperability Framework. `http://www.w3.org/2005/Incubator/mmsem/XGR-interoperability/` (as of September 2008).

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