Preserving Meaning, Not Just Objects: Semantics and Digital Preservation

David Dubin, Joe Futrelle, Joel Plutchak, and Janet Eke

Abstract

The ECHO DEPository project is a digital preservation research and development project funded by the National Digital Information Infrastructure and Preservation Program (NDIIPP) and administered by the Library of Congress. A key goal of this project is to investigate both practical solutions for supporting digital preservation activities today, and the more fundamental research questions underlying the development of the next generation of digital preservation systems. To support on-the-ground preservation efforts in existing technical and organizational environments, we have developed tools to help curators collect and manage Web-based digital resources, such as the Web Archives Workbench (Kaczmarek et al., 2008), and to enhance existing repositories' support for interoperability and emerging preservation standards, such as the Hub and Spoke Tool Suite (Habing et al., 2008). In the longer term, however, we recognize that successful digital preservation activities will require a more precise and complete account of the *meaning* of relationships within and among digital objects. This article describes project efforts to identify the core underlying semantic issues affecting long-term digital preservation, and to model how semantic inference may help next-generation archives head off long-term preservation risks.

Introduction: The Need for a Semantics of Preservation

The Preservation Semantics Problem

Like any information management activity, digital preservation efforts are guided by human understanding. Decisions about documenting a file

LIBRARY TRENDS, Vol. 57, No. 3, Winter 2009 ("The Library of Congress National Digital Information Infrastructure and Preservation Program," edited by Patricia Cruse and Beth Sandore), pp. 595–610

format, emulating an environment, or migrating from one system to another are made with an understanding of how levels of digital expression cascade and interrelate: voltage, bit, octet, pointer, integer, grapheme, pixel, polygon, color, pitch, text string, tree, image, tuple, file, and so on. The complexity of these relationships poses few serious problems for human beings—in fact, the problems lie precisely in the ease with which our minds interpret those relationships. Long-term preservation is distributed not only over time but also across the responsibilities of many different people. It is directed at collections much too large to allow thoughtful attention to individual resources. We must therefore build into our tools a more careful and precise encoding of the knowledge that guides our effortless mental deductions. The preservation hazards that result from current descriptive practice and our experiments with automated tools to ameliorate those risks are described in the sections that follow.

Our Goal

Our goal is to better understand the semantic problems arising in digital preservation, and how we might apply that understanding to the development of resources and tools. Specifically, we are experimenting with automated inferences about entities, their properties, the relationships within and between them, and how these facts are expressed in metadata descriptions. Enriching that metadata with new deduced assertions is one step in heading off digital preservation risks. We are working toward a deductive system for reasoning about anomalous or incomplete metadata. The aim is not to automatically deduce all missing information or to correct malformed records, but to call human attention to descriptions that are problematic or suspicious.

Our work begins with an analysis of the kinds of semantic problems posed by current descriptive practice and metadata schemas, informed by analyses of real-world data migration examples. We have applied the understanding gained in this analysis to the development of a draft metadata ontology (discussed in the section “Toward More Capable Archives and Repositories”), which moves us toward a more formal understanding of how descriptive information about archived digital resources is structured. This metadata ontology is key to a proof-of-concept experimental system composed of the Resource Description Framework (RDF) repository Tupelo and the BECHAMEL reasoning software.

The Problems: Understanding Semantic Preservation

Problems Posed by Descriptive Practice and Structures

In many preservation efforts metadata description may seem straightforward, but crucial information—including facts that seem obvious at first glance—is left unstated, and must be inferred by human readers. (An example is provided next.) As discussed previously, this situation may not be risky when people are available to reason about individual records, but a human-based manual approach does not scale over large collection sizes or over time. The sheer volume of digital information means we increasingly rely on automated machine processing of records. But software tools execute transactions using only knowledge that has been explicitly represented for them. Our aim therefore is to make those unstated facts available in a form that software can use.

This work begins with an investigation of the kinds of semantic problems posed by current information structures and implementations. These problems break down into three basic categories: (1) semantic problems relating to descriptive practice, (2) semantic problems relating to encoding standards, and (3) semantic problems relating to metadata schema design.

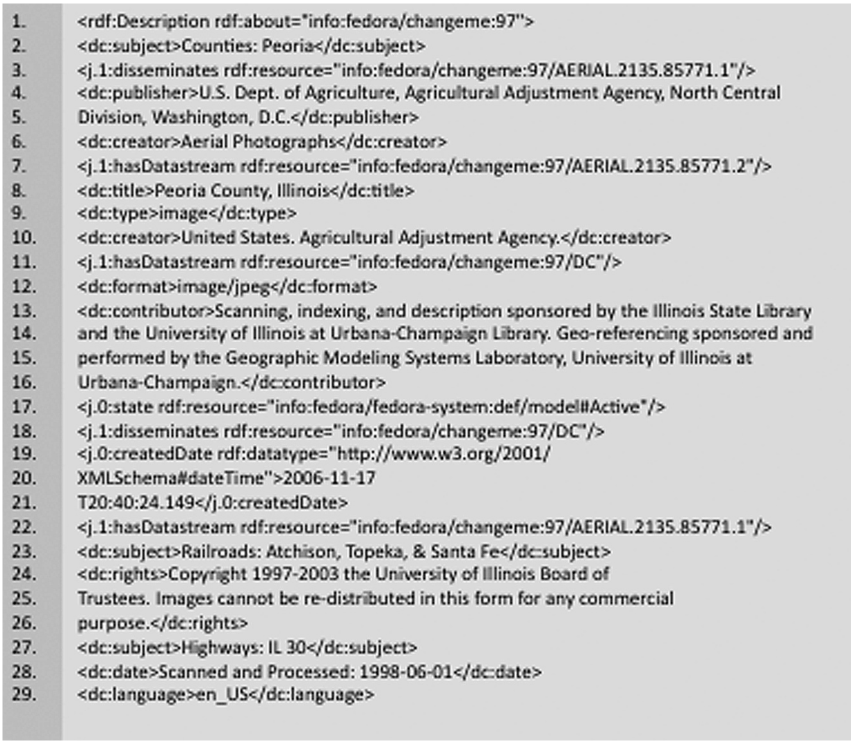
Semantic Problems Relating to Descriptive Practice

Some of the problems we face are a result of how resources are described using metadata, while other problems arise by way of how those descriptions are expressed, and what happens to them over time as they are migrated from one system to another. One semantic problem of particular interest to us is what Renear et al. (2002) describe as "ontological variation in reference." Essentially, metadata can fail to make critical distinctions in what, precisely, it is describing. The problem is illustrated in the metadata example in figure 1 below, which shows properties asserted at a number of different levels of abstraction.

We see in this example properties of the image itself (like its title and subject matter in lines 8 and 23) described alongside properties of the file that encodes the image (its MIME classification in lines 2 and 12), properties of the metadata description (its creation date in line 28), and properties of the repository software object that expresses the metadata description (e.g., that it disseminates resources, and has particular data streams associated with it; lines 1, 3, 7, 11, 18, and 22).

The main preservation risk proceeding from this mixing of levels is the inability to distinguish, without semantic information absent in the description, the level at which a particular property applies. For example, what is it exactly that has a MIME classification image/jpeg? Is it the Fedora record or is it one or both of the datastreams? That kind of ambiguity is easily resolved by a human reader without conscious effort, but preservation transactions (such as migration) are typically executed through software, which cannot. The preservation aim here is presumably to preserve access to the image. That aim may or may not depend on preserving the jpeg file expressing the image, and preserving the Fedora object that expresses the metadata is almost certainly not a requirement.

This example therefore illustrates the problem of mixed levels of description. We need to clarify and enrich metadata descriptions by linking their assertions explicitly to the appropriate entities, or else draw the at-



```
<rdf:Description rdf:about="info:fedora/changeme:97">
<dc:subject>Counties: Peoria</dc:subject>
<j.1:disseminates rdf:resource="info:fedora/changeme:97/AERIAL.2135.85771.1"/>
<dc:publisher>U.S. Dept. of Agriculture, Agricultural Adjustment Agency, North Central
Division, Washington, D.C.</dc:publisher>
<dc:creator>Aerial Photographs</dc:creator>
<j.1:hasDatastream rdf:resource="info:fedora/changeme:97/AERIAL.2135.85771.2"/>
<dc:title>Peoria County, Illinois</dc:title>
<dc:type>image</dc:type>
<dc:creator>United States. Agricultural Adjustment Agency.</dc:creator>
<j.1:hasDatastream rdf:resource="info:fedora/changeme:97/DC"/>
<dc:format>image/jpeg</dc:format>
<dc:contributor>Scanning, indexing, and description sponsored by the Illinois State Library
and the University of Illinois at Urbana-Champaign Library. Geo-referencing sponsored and
performed by the Geographic Modeling Systems Laboratory, University of Illinois at
Urbana-Champaign.</dc:contributor>
<j.0:state rdf:resource="info:fedora/fedora-system:def/model#Active"/>
<j.1:disseminates rdf:resource="info:fedora/changeme:97/DC"/>
<j.0:createdDate rdf:datatype="http://www.w3.org/2001/
XMLSchema#dateTime">2006-11-17
T20:40:24.149</j.0:createdDate>
<j.1:hasDatastream rdf:resource="info:fedora/changeme:97/AERIAL.2135.85771.1"/>
<dc:subject>Railroads: Atchison, Topeka, & Santa Fe</dc:subject>
<dc:rights>Copyright 1997-2003 the University of Illinois Board of
Trustees. Images cannot be re-distributed in this form for any commercial
purpose.</dc:rights>
<dc:subject>Highways: IL 30</dc:subject>
<dc:date>Scanned and Processed: 1998-06-01</dc:date>
<dc:language>en_US</dc:language>
```

Figure 1. Example of Multiple Levels of Abstraction in Metadata Description

tention of human analysts to records that cannot be disambiguated automatically.

Semantic Problems Relating to Encoding Standards

In addition to problems of descriptive practice, we face semantic problems stemming from limitations of the encoding technologies in which metadata descriptions are expressed. These problems generally fall into one of the following two categories:

- *Competing semantic relationships:* Preservation metadata formats typically overload a simple syntax with multiple competing semantic interpretations. Typical examples include XML applications where a small number of syntactic relationships (e.g., the parent/child relationship between elements) represent any number of semantic relationships (whole/part, property name/value, etc.) that are context dependent. Often a precise interpretation of these semantics can be found only in the execution of application software that consumes the file—and, presumably, in the mind of the programmer who wrote the application.

- *Unstructured data:* The information in resource descriptions may only be incompletely available for machine processing and verification. Crucial contextual data may exist only as natural language annotations or as unstructured information in the content of metadata fields.

The metadata example presented in figure 1 does not exhibit problems of syntactic overloading, because it conforms to a standard serialization of the RDF abstract model in which properties and relationships are explicitly identified. But the second problem is evident in how much information in this description is expressed in natural language text and annotations. (Note, for example, the *dc:date* element (line 28) in which the documented event (scanning and processing) is prepended to the date string.)

Problems with object models

Other potential semantic problems stemming from limitations of metadata encoding technologies concern the object models of repository systems themselves. Modeling decisions in repository design can create descriptive artifacts that leave their mark even after record migration. For example, a repository may mingle information about repository objects with the information that the repository objects are meant to preserve, creating problems when those records are further processed and contextual information is no longer available to help interpret the records and make further preservation decisions. A good illustration of this issue can be seen by revisiting the metadata description in figure 1, which was serialized from triples that were extracted from the RDF database backing a Fedora repository installation.

In figure 1, notice that in RDF terms this entire metadata description is "about" an object identified as *info:fedora/changeme:97* (line 1). This repository software object is the only resource identified by an *rdf:type* arc, and is therefore the only entity with an object class identification. Barring any explicit *type* identification in a resource description, Fedora objects seem to be the only kind of thing that the Fedora repository knows about. Expressed in that form, we cannot preserve any information except Fedora records, and those records assert no explicit preservation targets. A system like Fedora can preserve objects within the context of its own transactions, but the implicit knowledge directing such operations depends on the interpretation of programmers, with all the problems discussed so far.

On the other hand, it is not a design flaw of Fedora that its metadata record is centered internally on the Fedora digital object. Preservation ontology is properly a matter of descriptive practice, not software engineering. In fairness, our metadata example comes from a migration scenario in which RDF triples are extracted from Fedora's RDF store directly, rather than through a conventional export process. But this example serves to remind us that object modeling in a system such as Fedora plays the same role to the same ends as with other kinds of software: efficient source code management by and for the system developers. Object modeling decisions are not intended and cannot be expected to address the weaknesses of resource analysis and description. For long-term preservation, therefore, it is important to reduce ambiguous or implicit semantics in repository object models. That can mean either modifying those models or, as we have attempted, providing tools and techniques for migrating from repository object models to models that include better representations of preservation targets.

Semantic Problems with Published Metadata Schemas

Finally, in addition to semantic problems relating to descriptive practice to encoding standards, we see semantic problems stemming from limitations of published metadata schemas themselves. Published schemas formalize element sets on which the property/value ascriptions are based. Each of these metadata schemes not only expresses its unique view of the universe, but is itself grounded in basic ontological assumptions. A variety of ambiguities can still arise, as illustrated below, drawing again on our running example from figure 1.

We need to begin by understanding the logical parts of the metadata record and their relationships to one another. A metadata record describes some entity—an instance of a class like the class of books, images, or audio recordings. Metadata descriptions list properties of that entity, each of which has a value. For instance the "author" property of the book might take as its value the name of the author. Membership in a class requires that the instance respect defined class constraints (movies, for example, have running times, but books do not).

Consider the metadata statement *dc:type>image</dc:type>* (line 9) from our original example in figure 1. We easily recognize that the word "image" points us to an instance of a class, just as the name of an author points us to a particular person. A human reader would never conclude that a book was authored by a name or by a string expressing a name. Similarly, the word "image" licenses our inference that the property value for *dc:type* is a class of entities in the world rather than, for example, a quantity (such as 14 centimeters) or a quality (like monochrome). In this case we are cued to the existence of not just any entity, but to the very target of our preservation efforts—something much more important to us in the long run than the digital file or the bit sequence that only expresses this image contingently.

Computer software cannot make those kinds of meaningful distinctions without help. One kind of help would be a constraint on the range of allowable property values, but the Dublin Core element schema enforces no such constraint: *dc:type* can take any value that indicates the "nature or genre of the resource" (DCMI Namespace for the Dublin Core Metadata Element Set, Version 1.1, 2008).

A second kind of help would be a value string which, through its machine-readable structure or notation, indentifies a class. In an RDF expression this would be a Uniform Resource Identifier (URI) linked by an *rdf:type* property to some class declaration. The DCMI Type Vocabulary has this structure (http://dublincore.org/documents/dcmi-type-vocabulary), and interestingly, the scope note for *dc:type* in the DC Elements RDF schema recommends the use of that vocabulary (http://dublincore.org/documents/dces). Had the author of our metadata description used the URI *dcmitype:Image*, instead of the word "image," we would be one step closer to identifying the abstract image as an entity. The word "image," although it contains the same sequence of letters, is not linked in a standardized way to the declaration of a class. Assigning a DCMI type resource to the *dc:type* element simplifies the inference that an image exists, and that one or more of the metadata statements in that description are ascribing properties of an image—semantically, a significant step. But as our running example stands, the schema's flexibility invites ambiguity, and additional information is necessary to connect the literal value "image," with a formalized class such as *dcmitype:Image*.

Summary

We have seen in this section that descriptive practice, encoding standards, and published specifications may all complicate digital object preservation. Imprecise resource descriptions can make it impossible to determine the level at which a particular property applies. The flexibility offered by encoding standards brings risks as well as benefits. We've also seen how object modeling decisions and semantically underspecified metadata schemas can lead to incorrect or ambiguous usage. In the next section, we move from understanding the core semantic problems associated with descriptive practice and structures to looking at the resources and tools being developed by the ECHO DEPository project to identify semantic ambiguity in real-world metadata descriptions and highlight potential preservation risks.

Toward More Capable Archives and Repositories

Recap: The Need for Automated Inference Capability

Digital resource preservation efforts are distributed not only over time but typically across the responsibilities of people who may never consult with one another. Transactions like migration between systems are executed over large collections where close attention to individual records is too expensive, but where correct treatment of a resource often depends on knowledge that is incompletely or imprecisely represented in preservation metadata. Such ambiguities present few problems for human beings: our flexible minds make correct inferences without conscious effort. But the data to support those inferences are not expressed in a form that can guide the execution of our programs and utilities. We therefore need tools and methods that support the discovery and correction of preservation risks. The next section describes our experiments in developing these methods and tools.

Resources: BECHAMEL and Building a Metadata Ontology

BECHAMEL is a tool for expressing and testing semantic models of digital resources. (Dubin at al., 2003) It has been developed by researchers at the University of Illinois, the World Wide Web Consortium, and the University of Bergen. A BECHAMEL application can, for example, translate the bibliographic metadata for a journal article from one standard format into another by constructing a model of the author's affiliation with an institution. (Renear and Dubin, 2003) In our recent and current experiments the inputs to BECHAMEL are metadata descriptions retrieved from an RDF repository (Tupelo, 2008), together with schemas defined in the OWL Web Ontology Language (OWL, 2004). New facts deduced from those inputs are added back to the repository as annotations to the description. A technical overview of this approach is presented in the following sections.

Overcoming Semantic Problems in Metadata Encoding: A Resource and Description Vocabulary

Our aim is to enrich metadata with new assertions inferred from existing resource descriptions. Toward that aim we have identified classes, properties, and relationships for overcoming encoding problems, and we have expressed these in a schema. This vocabulary does not represent classes or properties for specific types of resources. Instead, it offers an ontology of metadata descriptions themselves. Simply stated, the vocabulary includes terms that can be used to describe records, metadata descriptions, and relationships between them and preservation targets. More specifically, the vocabulary is divided into the following sections.

- *W3C Standard Classes and Properties* These include classes and properties such as *rdfs:Resource*, *rdf:Statement*, and *owl:ObjectProperty*.
- *Alternate Reification Classes* Conventional use of the RDF reification vocabulary is based on an understanding that triples stand in a type/instance relationship with "tokens" appearing in RDF documents (RDF Semantics, 2004). But this interpretation, intended to support provenance documentation, presents puzzles for understanding how a serialized expression can stand in direct relationships with resources referred to by an abstract triple. (For those analysts who may be concerned with abusing the official account of RDF reification, the vocabulary includes separate classes for generalized statements, RDF statements, and abstract triples.)

- *Indication Relationships* This section includes a group of hierarchically organized relationships, based on recommendations in Piotr Kaminski's 2002 thesis. The relationships include indication, representation, denotation, identification, description, depiction, ascription, expression, and encoding.
- *Classes Based on the DCMI Abstract Model* In the Dublin Core Abstract Model, the term *metadata element* is used synonymously with the term *property*. But our classes, though based on that model, represent metadata elements as specialized names of properties, rather than as properties themselves. Classes in this section include metadata element, metadata element set, metadata statement, and metadata description.
- *Markup Structures* A third alternative to reifying RDF statements under the official W3 interpretation or through use of alternate classes is to reify the notation expressing the RDF. This section of the vocabulary includes classes for XML elements, XML documents, XML schemas, XML attributes, and URIs.

Summary

In summary, the Resource Description Vocabulary is an ontology of metadata descriptions themselves. Its aim is to provide a semantically sound framework for overcoming the encoding problems described in ”The Problems” section of this article. The next section walks us through a demonstration of how this ontology, as used by BECHAMEL, can help to highlight potential preservation risks.

Resolving Semantic Ambiguity: An Inference Example

In “The Problems” section of this paper we discussed problems of descriptive practice, encoding standards, and schema design. Now we present an illustration of how our inferencing software responds to those problems. In the example below, an ambiguous metadata statement from the record shown in figure 1 is identified and associated with the implied preservation target it describes.

Figure 2 shows one RDF statement extracted from figure 1, our running example

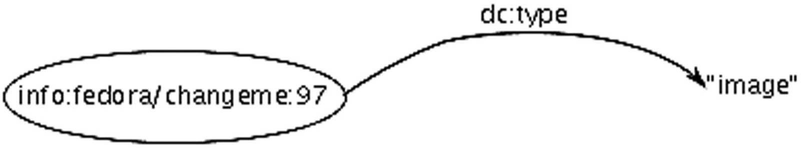


Figure 2. A fragment of the record shown in figure 1.

This RDF statement view shows the string value “image” assigned to the Dublin Core *type* property for the Fedora Object identified as *info:fedora/changeme:97*—as we discussed earlier when viewing the original markup record (fig. 1). The main issue is one of clearly identifying the *target* of our preservation efforts: an image in this case. Summarizing the concerns discussed earlier:

- The Fedora object is an amorphous resource, which seems to share properties of the image itself, the image content, and the bitstream encoding the image. The Fedora object cannot, therefore, be our preservation target.
- According to the formal schema definition, the Dublin Core Type property indicates the "the nature or genre of a resource," but need not identify the existence of any particular concrete object or abstract entity. As already seen, this vagueness in the formal schema opens the door to the use of values (such as the literal string "image") that are clear to human readers but which pose problems for machine processing.
- Although the word "image" invites a human reader to infer that our preservation target is an image, that information is not explicit enough to support automated processing. The inference depends not only on word meaning but also on the tacit background knowledge that the property value must in this case be a class (rather than, for example, a quality, quantity, or name).

To recap then, this image (fig. 2) illustrates the relationships between the Fedora object, the DC element "type," and the value "image" ambiguously expressed in the original record (fig. 1). In the next step, we begin to clarify these relationships.

Figure 3 below shows the first inference stage.

This RDF statement shows that the original *dc:type* arc has been identified as a metadata statement, and a new TAG URI has been generated to denote that statement. Although this stage of the processing began with conventional RDF reification, our assignment of *rdf:subject, rdf:predicate,* and *rdf:object* properties to our new *Metadata Statement* instance is a departure from orthodox RDF semantics. This first stage of inference processing has identified the metadata statement. In the next stage we take this a step further to identify the preservation target.

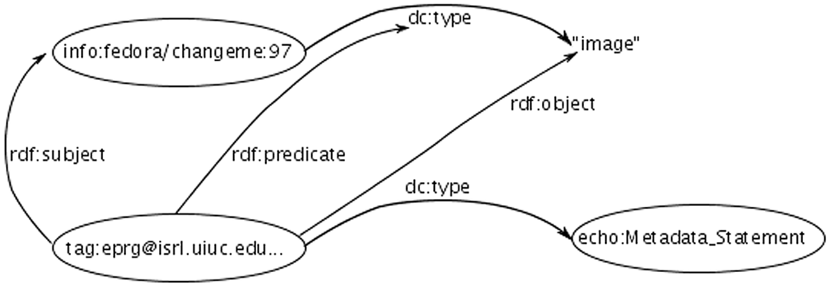


Figure 3. BECHAMEL has identified the fragment as a metadata statement

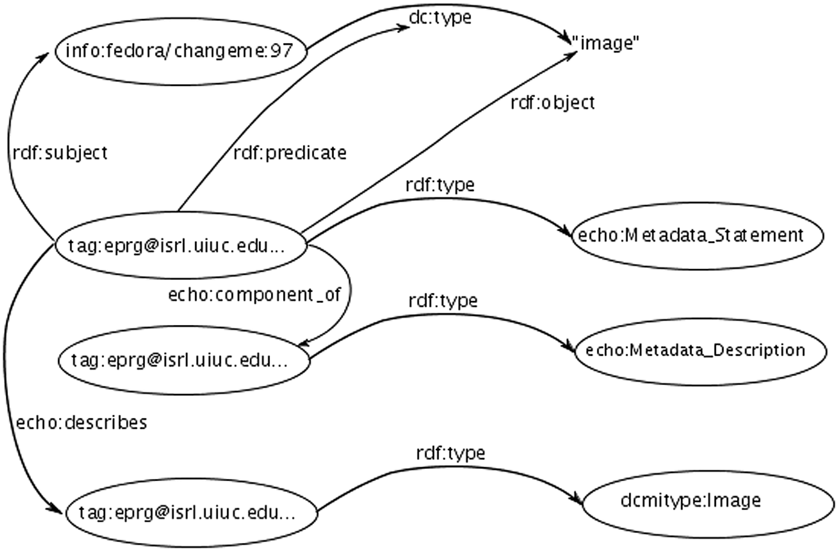


Figure 4. BECHAMEL has identified the metadata statement as a description of an image

Figure 4 shows the identification of the preservation target. The system infers that this metadata statement must be describing an abstract image that has both a class identity and an object identity distinct from the JPEG file, the bitstream encoding that file, the geography depicted in the image, and the Fedora object that serves as the locus for property attributions at all those levels. In addition, the metadata statement is identified as belonging to a metadata description. Identifying the preservation target should simplify the validation of later preservation transactions, making it easier to verify that essential properties persist across migrations and through translations from one format to another.

Automated Inference as a Preservation Service

The ontology and inferences that it supports allow us, even in cases where metadata records are terse and incomplete, to recover important distinctions, such as the distinction between a person and the metadata record describing that person. This knowledge is expressed in a portable syntax (RDF/OWL) with explicitly-defined semantics, so it can be maintained without having to modify the original record or transform it into another syntax (either of which could introduce further preservation risks). Indeed, BECHAMEL's ability to read and write from RDF databases (using Tupelo) means that it can read metadata records, apply rules and infer new assertions, and write those assertions back to the RDF database without altering the original records in any way. The "open world" of RDF/OWL means that automated inference can become a part of the preservation process without requiring that we redesign and reimplement institutional repositories to accommodate it. Instead, inference is a kind of service that can be used alongside those tools to head off preservation risks and fill gaps in representation.

The next section looks more closely at the architecture and proof-of-concept implementation of an archive that augments an institutional repository with inference capabilities and services.

System Architecture

We respond to the practice, standardization, and technical problems previously outlined in two ways:

- First, we design our systems for a world where metadata will vary greatly in their completeness, expressivity, and consistency. Preservation risks will arise, and we build tools with the aim of ameliorating those problems.
- Second, we propose an architecture for repositories that we hope will support more effective resource description and encoding: one that includes capabilities and services that will be needed in the next generation of digital content management systems.

Architecture: Overview

The proposed architecture augments typical institutional repository architectures with two new capabilities:

- The ability to manage not just bitstreams and associated metadata, but also associated semantics, expressed in standard RDF and OWL syntax.
- Automated services for detecting and/or correcting semantic ambiguity in metadata descriptions.

Architecture: The Tupelo Model

Tupelo is a middleware component providing semantic content management for distributed, heterogeneous applications. By *middleware*, we mean that Tupelo provides abstractions (known as contexts) that encapsulate different storage and retrieval technologies for data and metadata, including file systems, Web services, relational databases, and RDF stores. By way of these contexts, applications can exchange RDF statements and access raw octet streams associated with them. Tupelo can therefore serve the same role as a content management system (CMS) or institutional repository. But Tupelo differs from these systems in making only minimal assumptions about the structure of the information it manages, allowing applications to encode that structure as explicit RDF statements. RDF's open-world assumption and use of Uniform Resource Identifiers means that Tupelo can assemble descriptions from multiple, independent sources, even if those sources are not otherwise coordinated.

Tupelo has originally been designed to support science applications where data is produced, processed, and transformed by multiple people and software components. Such applications require preservation of workflow traces and the tracking of relationships between raw input and output results across distributed systems. These same challenges arise in digital preservation, where critical transformations may occur outside of the control of a repository system, or within metadata whose semantics are known at one stage of the process and unknown at another. Such transformations are distributed, heterogeneous processes, and tying a digital artifact to the process in which it participated requires portable, globally scoped identifiers that can be managed independently of the process itself. RDF usage enforces the global scope of identifiers by using URIs to identify nodes.

Connecting BECHAMEL to Tupelo

Our BECHAMEL client application retrieves an XML-serialized subgraph of the repository contents from Tupelo via Tupelo's HTTP-based client/server protocol, which is based on extending Nokia's proposed URIQA protocol (http://sw.nokia.com/uriqa/URIQA.html). The subgraph is submitted to BECHAMEL, together with supporting OWL Ontologies and standardized RDF vocabularies (e.g., Dublin Core). New RDF statements and annotations emerging from BECHAMEL's execution (see the inference example in figures 2–4) are then delivered back to the Tupelo server.

Observations on Implementation

Like the characteristics of the Tupelo architecture, we predict that inferential capabilities (such as those illustrated earlier in this section) will be basic services provided by and for future digital repositories. But the functional components of those repositories will be loosely coupled and distributed. Interpretive services are, furthermore, needed right away for systems based on current Content Management System technologies, and to aid in reforming descriptive practice as it stands today. For all these reasons, we have sought in our implementation to make the interpretation component a structurally distinct layer, communicating with the Tupelo middleware via general-purpose client/server protocols such as http. While we assume the resource descriptions and inferred knowledge will conform to the RDF abstract model, we have chosen to deliver them in conventional serialized forms, such as RDF/XML.

As with any similar project, a variety of engineering challenges require further experimentation and improvement. For example, at the BECHAMEL application layer, all RDF statements are expressed as if part of a single global graph, whether retrieved from Tupelo, parsed from an RDFS vocabulary, inferred by BECHAMEL itself, or drawn from any other source. But obviously, only a finite amount of input knowledge can be efficiently shared over the network between client and server. Our interpretive rules are themselves separate from the strategies for selecting, retrieving, and storing RDF statements, but pragmatically they cannot be totally independent of each other.

Lessons Learned and Next Steps

Our research contribution can be seen from one perspective as the technical groundwork for a future generation of improved automated digital preservation systems and methods. But one can also understand our findings as opportunities to apply human intelligence more effectively with existing tools and standards. It might never occur to a digital librarian that his or her preservation methods are being executed without clearly identifiable targets, or that a simple change (such as *dcmitype:Image* instead of "image") could dramatically reduce the work required to correct that problem. The exercise of encoding semantic knowledge with enough clarity and precision for a computer reveals complexities that our remarkable human minds would otherwise allow us to ignore. With the aid of that insight, much progress could be made in reforming the practices that prompt our development and research.

Conclusion

Institutional repositories and other current efforts for preserving digital artifacts face challenges resulting from underspecified metadata schemas, ambiguous usage, and metadata models that relate more to repository implementation than to issues of meaning. These entail very real risks to the integrity and usefulness of preserved digital artifacts as they are stored, managed, and retrieved. Descriptive practices that seem correct may introduce inconsistencies that are undetectable without manual inspection of each record—an unreasonable requirement for collections of even moderate size.

Improved metadata standards and repository metadata models are part of the solution to these problems, but we also see a role for automation in detecting and mitigating preservation risks. Our experimental archiving technologies, BECHAMEL and Tupelo, demonstrate that we can locate and correct ambiguous metadata expressions in the context of transactions such as import and export. As best practices evolve for digital preservation, we see reasoning capabilities like those demonstrated by BECHAMEL becoming an integral component of digital preservation systems, allowing curators to transform large collections with greater confidence that records will faithfully represent the information they are intended to preserve. Complementing interoperability models like ECHO DEPository's Hub and Spoke Tool Suite (Habing et al., 2008), we believe the techniques described in this paper point to a new generation of preservation tools, and reveal ways to use existing tools with more success.

References

Dubin, D., Sperberg-McQueen, C. M., Renear, A., & Huitfeldt, C. (2003). A logic programming environment for document semantics and inference. *Literary and Linguistic Computing, 18*(2), 225–233.

Habing, T., Ingram, W., Cordial, M., Manaster, R., & Eke, J. (2008). Developments in digital preservation at the University of Illinois: The Hub and Spoke architecture for supporting repository interoperability and emerging preservation standards. *Library Trends, 57*(3), this volume.

Kaczmarek, J., Hswe, P., Hauser, L., & Eke, J. (2008). The Web archives workbench: Taking an archival approach to the preservation of Web content. *Library Trends, 57*(3), this volume.

Kaminski. P. (2002). *Integrating information on the semantic web using partially ordered multi hypersets.* Unpublished master's thesis, University of Waterloo. Retrieved September 12, 2008, from http://www.ideanest.com/braque/Thesis-web.pdf

OWL Web Ontology Language. (2004). Retrieved October 2, 2008, from http://www.w3.org/TR/owl-features/

RDF Semantics. (2004). Retrieved October 2, 2008, from http://www.w3.org/TR/rdf-mt/

Renear, A., Dubin, D., Sperberg-McQueen, C. M., & Huitfeldt, C. (2002). Towards a semantics for XML markup. In E. Munson, R. Furuta, and J.I. Maletic (Eds.), *Proceedings of the 2002 ACM Symposium on Document Engineering* (pp. 119–126). New York: ACM.

Renear, A., & Dubin, D. (2003). Towards identity conditions for digital documents. In S. Suttontor (Ed.), *Proceedings of the 2003 Dublin Core Conference.* Seattle: University of Washington.

Tupelo. (2008). Retrieved October 2, 2008, from http://dlt.ncsa.uiuc.edu/wiki/index.php/Main_Page

URIQA. The URI Query Agent Model: A Semantic Web Enabler. (2003–2008). Retrieved October 2, 2008, from http://sw.nokia.com/uriqa/URIQA.html

David Dubin is a research associate professor at the Graduate School of Library and Information Science at the University of Illinois at Urbana-Champaign. David's research interests are in issues of expression and encoding in documents and digital information resources.

Janet Eke served as project coordinator of the NDIIPP-funded digital preservation projects based at the University of Illinois at Urbana-Champaign from fall 2004 through August 2008. She is now the research services coordinator at the Graduate School of Library and Information Science (GSLIS) at the University of Illinois at Urbana-Champaign where she helps to develop services, tools, and resources to support and promote the research efforts of the School. Previously at GSLIS she provided research services at a fee-based custom research unit and taught a master's course in online searching. Before joining GSLIS in 1998, she worked for many years in public libraries.

Joe Futrelle is a senior research coordinator at the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign. Joe is the technical lead programmer for the Tupelo Semantic Content Repository project, which is used to support the development of cyberinfrastructure components for e-Science, including the CyberIntegrator workflow system, the CyberCollaboratory portal environment, and the Digital Synthesis Environment for the publication and review of complex scientific data products. He has developed data management systems for a variety of science and science education projects, including work with the George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES), the Consortium of Strong Motion Observation Systems (COSMOS), and the National SMETE Digital Library project (NSDL). He has also researched digital library and digital preservation technologies for the National Archives and Records Administration and the National Cancer Institute.

Joel Plutchak graduated from the University of Wisconsin at Madison in 1981 with a degree in Computer Sciences. He has worked as a computer applications and support programmer at the University of Wisconsin's Space Science and Engineering Center (SSEC), as a programmer/analyst and system manager for Brown University's Department of Planetary Geology, and as a research programmer for the Department of Atmospheric Sciences at the University of Illinois at Urbana-Champaign. He is currently a research programmer at the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign, working in the Digital Library Technologies group.