A Metadata Model for Electronic Images

Regina M. Mathis
Nova Southeastern University
mregina@nova.edu

Lucinda Caughey
Nova Southeastern University
caughey@nova.edu

Abstract

Finding images over the internet can be trying at best. Many industries such as advertising, marketing, image consultants, print media, and medical imaging deal with electronic images on a daily basis. Images are often stored by some non-descriptive ordering system and come in a variety of formats. Using image metadata, information such as the physical, technical and content related characteristics can be captured. Prior research has primarily addressed specific types of images and addressed the storage of metadata in a database format. This paper presents a Resource Description Framework Schema for the generic categorization of electronic image characteristics and traits. This schema is useful for image management and can be widely utilized in document management systems which traditionally use film and optical technology to store images. An image editor is prototyped with the capability of annotating the metadata into the image or exporting image metadata into other business oriented systems.

1. Introduction.

Many industries such as medical imaging, advertising, print media and insurance use electronic images on a daily basis. Using the World Wide Web of today, searching for an image on a specific subject matter is equivalent to finding a needle in a haystack. A typical search will produce hundreds of thousands of web resources. The reason is that the current Web is primary composed of Web pages with information displayed in natural language text and images, for humans to view and understand. Machines are only used to display the information either on screen or in printed format. Current search engines use Meta tags, or keywords, to categorize web resources. The search engine has no understanding of the content of the web page.

One method of improving on the efficiency of search engines is by using XML to structure a web resource. This allows one to search for information with string and structure matching tools, however this method is limited in that the user must know the exact specifications of the document and is therefore not sufficient for searching on a large scale.

The next generation World Wide Web will be the Semantic Web. This is an emerging technology where the Web, and Web resources, through the use of semantic markup, will become more machine-readable. The idea behind the Semantic Web is to add markup to the Web resource so that the meaning of the content can be captured and then encoded in a form that a machine can understand. This will give machines the ability to locate, organize and integrate available information. Unfortunately, the current representation of Web resources impedes this progression. Machines have difficulty understanding current day Web resources to the extent required to perform these functions. To remedy this problem, the representation of Web resources must change to a more detailed and structured format.

Electronic images are a type of web resource. These are images that come in many formats and are used in a variety of industries. By applying semantic markup to electronic images, powerful semantic search engines will be able to pinpoint exact images, relating to specific domains, quickly and efficiently. Metadata enables semantic search engines to understand the content and to infer relationships between the content and specific domains. Metadata can also relate specific images to application domains. This paper focuses on the use of metadata to describe electronic images and presents a Resource Description Framework Schema for electronic images. The goal is to formulate a paradigm for electronic images that is generic in its categorization of image characteristics so it can be applied across a wide range of application domains. An image editor is prototyped. This editor uses the image schema presented to capture metadata and then either encode it into the image or export it to an external database. Medical imaging, advertising, insurance, marketing, image consultants and print media are just a few of the application domains that will benefit from the use of metadata to describe electronic images.

This paper is organized as follows: Section 1 is an introduction, section 2 discusses prior research on image metadata models, section 3 formulates the Image Data Model, section 4 formalizes the vocabulary, section 5 discusses the prototype of the image editing tool and section 6 concludes our discussion. Appendix A includes the complete Image Schema.

2. Prior research.

Previous works varied in approach and have been limited in both image type and application domains. Lafon and Bos [3] introduce a system to describe photos in jpeg format using a jigsaw server and a data entry program written in java. The system includes an RDF schema which primarily addresses the descriptions in terms of personal photographic images. The semantic content can be applied directly to jpeg images however other image types cannot be annotated.

Schreiber, Dubbeldam, Wielemaker, and Wielinga [8] built an annotation tool around background knowledge contained in ontologies to index and search jpeg photographs. The authors based their study on a sampling of 100 images of apes from various CD collections. The annotations are built upon two categories of definitions. The first is based on the structure of a photo annotation which is independent of the subject matter and serves as a description template. The second is based on a domain specific ontology for the animal domain. The authors built an architecture which is similar to that of [3] but differ in the nature of the ontologies, description of the subject matter and they explicitly linked their architecture to a particular domain ontology.

Baptista and Kemp [1] present a metamodel for geolibraries which places emphasis on ontologies which represent domain dependent knowledge currently existing in the research community. Their model is based on the four dimensions in the georeferenced domain, space, time, theme and datatype. The authors presented a Spatio Temporal Library (STepLib) which is used in the context of digital earth projects. The authors used multilevel hierarchies of spatiotemporal data. The model presented is not designed exclusively for images but can address images by allowing them to be searched by content.

The work of Li and Wang [5] address automatic linguistic indexing of images using a statistical modeling approach. The authors use images to train a dictionary of statistical models each representing a model. The study focuses on two dimensional multi-resolution hidden Markov models. The findings are positive and the approach is useful in the domains of computer vision and content based image retrieval.

Soo, Lee, Li, Chen, and Chen [9] present a framework that is based on a sharable domain ontology coupled with a thesaurus. The framework facilitates image retrieval by using case-based learning techniques with a natural language phrase parser which converts a natural language query into a Resource Description Framework (RDF) representation. These authors build their framework on the premise that the images have semantic metadata associated with it.

Leung and Zheng [4] developed a data model for representing image content in a relational database. There data model is based on "facts" which describe the subject matter of the image. The authors use these facts to build a relational database structure which includes four tables to be used for indexing and searching.

The framework presented is both flexible and scalable. It incorporates some of the aspects of work done by [3], [4] and [9] however it adds increased flexibility by building a data model that is not constrained to the use of relation databases or image modifications. It is scalable in that the image schema is applicable to a wide range of image types and application domains.

3. Developing the image data model.

To describe what information to include in the vocabulary, it is necessary to identify suitable domain elements and properties of interest. The major elements and their properties must be described along with the relationships between these elements. We wish to describe electronic images of all universally acceptable image types and the schema is designed exclusively for images. Several questions need to be considered. Who or what is shown in the image? Where was it taken? Originating information must also be included such as who created the image, who owns it, when was it created, its format, is it part of a series of images and how it was created. In some cases such as with medical imaging, a history of changes in location may be necessary [7].

Answering these questions enables the elements to be described by a set of properties. The elements can be grouped into four categories. The image profile will include aspects such as title, subject, location, description and any identifiers such as name or social security number or an arbitrary identifier that may or may not be unique. There can be multiple occurrences of subject, subject resources and identifiers. Logistical information such as the owner of the image and the artist, photographer, technician or the person who created the image are also included in the image profile. The second category is related information. This category is used to describe a relationship with other resources. This can occur when the image is part of a series of images on a particular subject matter or perhaps as part of a patient's medical test results. The third category is history. History includes information such as changes in the location of the image, reason for the change in location and the date that this change occurred. Fourth, technical information includes the type of image, format, size, resolution and origin. For a specific image, there are one Profile, Technical, and History sections but each section will have many entries. There may be multiple Related sections for a single image. Table 1 summarizes the Domain Elements and their properties.

Table 1. Domain elements and their properties.

Elements	Properties	Description
Profile	Subject	The subject matter depicted

		in the image. Can be a resource or literal. Every image must contain at least one subject entry.
	Title	Title of the Image
	Description	Description of the image
	Identifier	Subject's name or social security number or another type of identifier. May or may not be unique.
	Location	Location depicted in the image
	Creator	The person or entity that created the image
	Owner	The owner of the image
	Creation Date	Image creation date.
Technical	Type	This field exists so that other types of multimedia can use this schema such as video. The default is image.
	Format	GIF, JPG, TIF, PNG, PCX
	Image Size	The size of the Image
	Resolution	The resolution of the Image
	Origin	35mm Film, Optical Disk, Digital Camera, Imaging Center, Diagnostic Test
History	Location Change	New location can be a resource or literal
	Location change Reason	Reason for the change in location
	Change Date	Date that the change occurred.
Related	Related Resource	Related resources such as patient diagnosis, tests or other images in the series
	Reason	Reason for the relationship

The image used as a test case during the development of the image RDFS is located at http://www.islandspots.com/images/Aruba/renisland.jpg. A loose representation of the data model with top level predicates is shown in Figure 1.

After careful analysis, to ensure optimal flexibility, we will refine two of the top level elements. The technical element needs a minor change and the history element needs some additional changes.

In the technical element, two predicates need additional descriptors. Size and resolution need a type to

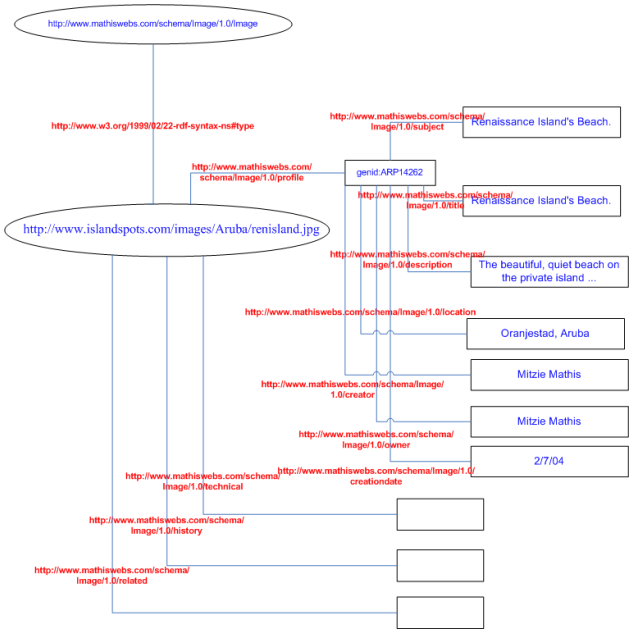

Figure 1. RDF graph with top level predicates.

specify the units associated with the values given for size and resolution. Image size is typically specified in inches, pixels, picas, points, centimeters or columns. Resolution is specified in either pixels/inch or pixels/cm. To address this, the rdf:value is used to specify the value and a unit predicate is added to specify the unit of measurement.

The history element needs further clarification. Many images may move, for example, from an imaging center where the image was taken, to the radiology department for review and analysis to a doctor's office. Typically, images will move in a particular sequence. Using the RDF container rdf:Seq indicates that the ordering of the elements inside the container is significant.

Keeping track of previous locations of the image is an important property in application domains such as medical imaging or advertising. There may also be a need to explain the reasons for the change in location. Each change in location for the image has its own Universal Resource Indicator (URI) which represents the new location. To accommodate this, a new location class, called LocationChange, has been created. This new class captures location changes and is separate from the container. The change is then related to the image through the container [7]. The LocationChange class supports the type of location change, reason for the location change and the date that the location change occurred.

The related element needs further discussion. This element is used when the image is related to some other image as in a series of images relating to a patient's medical history, or a series of images relating to a place. An image may be related to some other type of resource such as a web site with content depicted in the image. A relationship between the main element and these other

resources can be established using rdf:resource. This is a part of the img:Image block and takes on the form

```
  <img:related
rdf:resource="http://www.islandspots.com/im
ages/Aruba/ren1.jpg" />
  <img:related
rdf:resource="http://www.islandspots.com/is
lands/Aruba.html" />
```

A description of the resource and reason for the relationship can be described outside the img:Image block. If the resource is another image, then it is described as img:Image however if the resource is not an image, i.e. a web site, it is described using img:Resource.

3.1 Refining the property elements.

Dublin Core is a widely used metadata element set whose goal is to create a metadata model to define business data. Dublin Core is a simple set of vocabulary elements and its namespace is located at http://purl.org/dc/elements/1.1/ and the metadata element set is found at http://dublincore.org/2003/03/24/dces#. The commonly known prefix for the Dublin Core namespace is dc. Some of the Dublin Core elements such as title, subject, creator, description, date, type, format, identifier and rights appear to map to Image elements but need closer review. Because the elements are broad in scope, the Dublin Core group created dcterms, which are qualifiers. The namespace for the qualified Dublin Core schema is located at http://purl.org/dc/terms/.

Table 2 shows the Dublin Core properties that may match some of the Image properties. The match column suggests suitable replacements. Some of the Dublin Core properties are similar but after reviewing the schema for those elements, it became clear the semantic intent is different from the semantic intent of the property in the Image schema.

Table 2. Comparison of Image Properties and Dublin Core Properties.

Image Property	DC Property	DC Description/Comments	Match
Title	Title	A name given to the resource	*
Subject	Subject	The topic of the content of the resource	*
Description	Description	An account of the content of the resource	*
Identifier	Identifier	An unambiguous reference to the resource within a given context	
Owner	Publisher	An entity responsible for making the resource available	
Owner	Rights	Information about rights held in and over the resource	
Creator	creator	An entity primarily responsible for making the content of the resource	*
CreationDate	dcterms:created	Date of creation of the resource	*
Type	Type	The nature or genre of the content of the resource	
Format, ImageSize, Size Unit, Resolution, Resolution Unit	Format	The physical or digital manifestation of the resource	
Format, ImageSize, Size Unit, Resolution, Resolution Unit	dcterms:formatScheme	A set of format encoding schemes	
Origin	Source	A reference to a resource from which the present resource is derived	
Related	Relation	A reference to a related resource	
Change Date	dcterms:modified	Date on which the resource was changed	

Using the Dublin Core elements from Table 2 to replace the elements in the Image vocabulary simplifies the Image vocabulary. Example 1 shows the Image vocabulary with these replacements.

```
<?xml version="1.0"?>
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#"

xmlns:img="http://www.mathiswebs.com/schema
/Image/1.0/"

xmlns:dc="http://purl.org/dc/elements/1.1/"

xmlns:dcterms="http://purl.org/dc/terms/">

  <img:Image
rdf:about="http://www.islandspots.com/image
s/Aruba/renisland.jpg" >

<!--  Image Profile-->
    <img:profile rdf:parseType="Resource">
             <dc:subject>       Renaissance
Island's Beach.</dc:subject>
             <dc:title>         Renaissance
Island's Beach.</dc:title>
             <dc:description>
                    The  beautiful,  quiet
beach on the private island Renaissance
Island.   Guests of the Renaissance Aruba
Resort  and  Casino  can  use  the  island
freely.
             </dc:description>
             <img:location>Oranjestad,
Aruba
             </img:location>
             <dc:creator>Mitzie
Mathis</dc:creator>
             <img:owner>Mathis
Webs</img:owner>
             <dcterms:created>February   7,
2004</dcterms:created>
    </img:profile>

<!--  Image Technical Details -->
    <img:technical
rdf:parseType="Resource">
                <img:type> image</img:type>
             <img:format>
JPEG</img:format>
             <img:size
rdf:parseType="Resource">

       <rdf:value>640X480</rdf:value>

       <img:sizeUnit>pixel</img:sizeUnit>
             </img:size>
                <img:resolution
rdf:parseType="Resource">

       <rdf:value>72</rdf:value>

       <img:resolutionUnit>pixel</img:resol
utionUnit>
             </img:resolution>
    </img:technical>

<!--  Image History -->
    <img:history >

       <rdf:Seq>
                 <rdf:_1
rdf:resource="http://www.mathiswebs.com/cli
ent/islandspots" />
       </rdf:Seq>
              </img:history>

<!--  Related Resources of Image -->
    <img:related
rdf:resource="http://www.islandspots.com/im
ages/Aruba/ren1.jpg" />
    <img:related
rdf:resource="http://www.islandspots.com/is
lands/Aruba.html" />

    </img:Image>

<!-- Location Change definitions -->
       <img:LocationChange
rdf:about="http://www.mathiswebs.com/client
/islandspots">

       <img:changeType>Move</img:changeType
>
             <img:changeReason>Moved     to
Production</img:changeReason>

       <img:changeDate>04/16/04</img:change
Date>
       </img:LocationChange>

<!-- Related Resource Definitions -->
    <img:Image
rdf:about="http://www.islandspots.com/image
s/Aruba/ren1.jpg">
       <dc:title>One     of     the     pools
</dc:title>
       <img:reason>Part  of  the  Renaissance
Aruba    Resort    and    Casino    property
</img:reason>
    </img:Image>
    <img:Resource
rdf:about="http://www.islandspots.com/islan
ds/Aruba.html">
       <img:reason>Description     of     the
Renaissance Aruba Resort and Casino hotel
</img:reason>
    </img:Resource>
</rdf:RDF>
```

Example 1. Image vocabulary with Dublin Core elements.

4. Formalizing the vocabulary.

Formalizing the vocabulary with RDFS guarantees that the image descriptions “are semantically and syntactically consistent across implementations” [7, p. 118]. An RDFS class is used to represent types or categories [10] or more plainly, anything that can replace an rdf:Description block with an associated rdf:type [7].

Image and Resource can be explicitly defined as

```
  <rdf:type
rdf:resource="http://www.mathiswebs.com/sch
ema/Image/1.0/Image> and
   <rdf:type
rdf:resource="http://www.mathiswebs.com/sch
ema/Image/1.0/Resource>
```

in the rdf:Description blocks, identifying Image and Resource as classes. Resource in this context refers to a Web resource. Since electronic images are Web resources, Image is a subclass of Resource. LocationChange has already been established as a class so these three elements make up the classes for the image schema. Resource is defined as a subclass of the RDF Resource type. Image is defined as a subclass of Resource and LocationChange is a subclass of the Image Class.

In an effort to augment use of the image schema across many application domains, we need to establish the relationships between classes. The domain or domains chosen for the rdfs:domain value must be carefully identified. The rdfs:domain property indicates that the property applies to a specific class [10] . The rdfs:range values specified will indicate that the values of a particular property are instances of the specified class [10] . If the range of values is specified as literal, then the acceptable values for the property are literals. The inner relationships among properties can be defined using rdfs:subPropertyOf. The domain, range and sub-property values indicate the relationships of the properties to the classes. Table 3 Summarizes the Domain, Range and Sub-property values for the properties.

Table 3. Domain, range and sub-property values for properties in the image schema.

Property	Domain	Range	SubPropertyOf
Profile	Image	Image	
Identifier	Image	Literal	Profile
Location	Image	Resource	Profile
Owner	Image	Resource	Profile
Technical	Image	Image	
Type	Image	Literal	Technical
Format	Image	Literal	Technical
ImageSize	Image	Literal	Technical
SizeUnit	Image	Literal	ImageSize
Resolution	Image	Literal	Technical
ResolutionUnit	Image	Literal	Resolution
Origin	Image	Resource	Technical
History	Image	LocationChange	
ChangeReason	Image	LocationChange	History
Related	Image	Resource	
RelatedResource	Image	Resource	Related
Reason	Image	Literal	Related

It is interesting to note that history has LocationChange as its domain. This indicates that the history property applies to the LocationChange class. The related property is different in that the range extends beyond image and can be another resource, for example, a web site, database, or another image.

The properties identifier, location and owner are sub-properties of profile and are similarly composed. The remaining identifiers are described in a similar fashion. The complete schema is stored at http://www.mathiswebs.com/schema/Image/1.0/ and is shown in Appendix A.

5. Image editor prototype.

The design of the Image Editor is influenced by the image schema having the major categories of elements being captured by using the Markup Menu. This image editing tool can do some basic image modifications however the primary purpose is to create semantic metadata. Semantic metadata can either be saved as part of the image, depending on image type, or exported for external use.

The semantic metadata can be saved as part of the comment block for JPEG, GIF, and PNG type images. The image editing tool is written in Java and currently Java Advanced Imaging does not support writing GIF comment blocks because of patent issues. As a result, saving the image metadata as part of GIF files will not be supported in the initial releases of this image editor. Information regarding the GIF format is included for completeness. The specifications for including the semantic metadata for each of the three types listed are shown in Table 4.

Table 4. Image types that can allow Semantic Metadata to be embedded directly into the image [6].

GIF	Comments are included in Comment Extension blocks and any number may appear anywhere after the Global Color Table. Each comment may contain up to 255 7-bit ASCII characters, including all the ASCII control codes. This prototype will include comments before all image data in the GIF file.
JPEG	Metadata can be stored in comment blocks however these comments will only be read by JPEG File Interchange Format (JFIF) transporters. SPIFF is an international standard,

	coined the "official JPEG format" but is not widely used. SPIFF readers will be unable to read the comment block however will not interrupt or modify the data stream. SPIFF in effect will blindly pass the comment block.
PNG	Data may be stored as a private chunk however this data will only be readable to semantically aware PNG readers.

The File menu allows the exporting of the metadata in two forms, RDF code and ASCII delimited format. This provides the flexibility to store the data in a database format or as part of a website. The RDF code uses the RDF/XML serialization and can be incorporated into well formed XML documents.

The Markup Menu includes Namespace, Image Profile, Technical Info, History, Related Resources and Location Changes. The Namespace option allows the user to enter the specific URL of the image and the namespaces to be used with corresponding prefixes. Three namespaces are defaulted and the user can enter additional namespaces if desired. A screen shot of the namespace option is depicted in Figure 2. The user can enter additional namespace details if desired.

Image Profile allows the input of the specific information regarding the image profile. Aspects such as title, subject, location, description and any identifiers such as name or social security number or an arbitrary identifier that may or may not be unique are entered. There can be multiple occurrences of subject, subject resources and identifiers. To accommodate these multiple occurrences, an ADD button occurs after each of the fields where multiple entries can be displayed. The Technical Info section is similarly modeled and includes the technical aspects of the image as described. History, Related Resources and Location Changes can all be found under the Markup Menu and are similar in appearance and functionality. A representative screenshot showing the Technical Details section is shown in Figure 3.

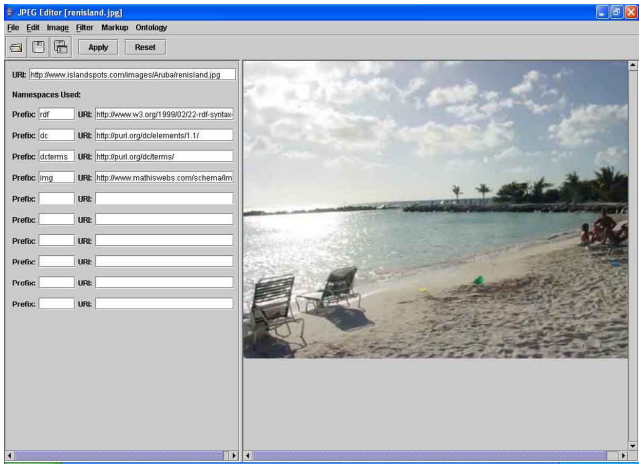

Figure 2. Capturing the namespace information.

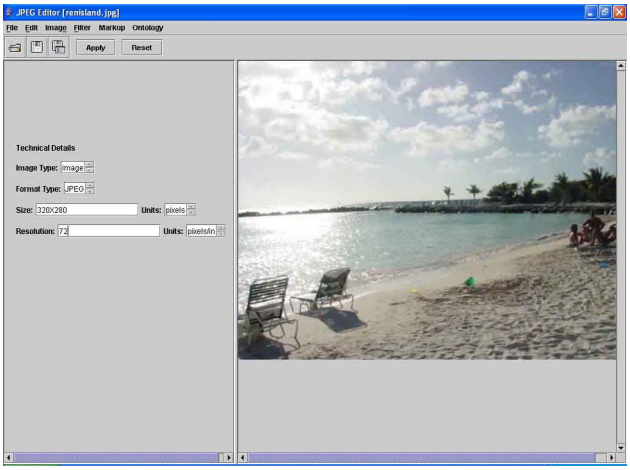

Figure 3. The technical details section of image editor

6. Conclusion

The World Wide Web has so much information that seeking and finding specific images has become a chore at best. A typical search for an image relating to a particular subject will present hundreds of thousands of web resources, some of which are not images. It is extremely difficult to find the image needed.

Providing images with semantic metadata will enable powerful and robust semantic search engines to find the desired image. An RDF schema has been presented for electronic images. This schema provides both flexibility and scalability in creating image metadata for universally acceptable image types and can be widely utilized in a variety of application domains. An image editing tool has been prototyped. The prototype is based on the image schema presented however it allows additional fields to be edited and entered. The image schema provides greater flexibility in the ability to describe universally acceptable image types. Sectors such as advertising, marketing, image consultants, periodicals, and medical imaging will benefit from the efficient storage, management and retrieval of images using application specific business systems.

7. References

[1] Baptista, C. S., & Kemp, Z. (2000). An Integrated Metamodel for Knowledge Representation in Geolibraries. IEEE Advances in Digital Libraries 2000 (ADL 2000), May 2000, 151-159.

[2] Brickley, D. (n.d.). RDF: Understanding the Striped RDF/XML Syntax (). Retrieved May 1, 04 from http://www.w3.org/2001/10/stripes/

[3] Lafon, Y., & Bos, B. (2000 September 28). Describing and retrieving photos using RDF and HTTP (). Retrieved April 28, 04 from http://www.w3.org/TR/2000/NOTE-photo-rdf-20000928/

[4] Leung, C. H. C., & Zheng, Z. J. (1995). Image Data Modelling for Efficient Content Indexing. Proceedings of the International Workshop on Multi-Media Database Management Systems (IW-MMDBMS '95), , 143-150.

[5] Li, J., & Wang, J. Z. (2003). Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(9), 1075-1088.

[6] Murray, J. D., & vanRyper, W. (1996). Encyclopedia of Graphics File Formats, Second Edition. Retrieved June 1, 2004, from O'Reilly & Associates Web Site: http://netghost.narod.ru/gff/graphics/formats.htm

[7] Powers, S. (2003). Practical RDF. Sebastopol, CA: O'Reilly & Associates.

[8] Schreiber, A. Th., Dubbeldam, B., Wielemaker, J., & Wielinga, B. (2001). Ontology-Based Photo Annotation. IEEE Intelligent Systems, May 2001, 66-74.

[9] Soo, V., Lee, C., Li, C., Chen, S., & Chen, C. (2003). Automated Semantic Annotation and Retrieval Based on Sharable Ontology and Case-based Learning Techniques. Proceedings of the 2003 Joint Conference on Digital Libraries (JCDL '03), , 61-72.

[10] W3C (2004 February 10). RDF Primer (). Retrieved March 7, 04 from http://www.w3.org/TR/2004/REC-rdf-primer-20040210

8. Appendix A

```
  <?xml version="1.0"?>
  <rdf:RDF xml:lang="en"
       xmlns:rdf="http://www.w3.org/1999/0
2/22-rdf-syntax-ns#"
       xmlns:rdfs="http://www.w3.org/2000/
01/rdf-schema#"
       xml:base="http://www.mathiswebs.com
/schema/Image/1.0">

  <!-- Resource Definition -->
  <rdfs:Class rdf:ID="Resource">
       <rdfs:subClassOf
rdf:resource="http://www.w3.org/2000/01/rd
f-schema#Resource" />
       <rdfs:label        xml:lang="en">Web
Resource</rdfs:label>
       <rdfs:comment      xml:lang="en">Web
resource</rdfs:comment>
  </rdfs:Class>

  <!-- Image Definition -->
  <rdfs:Class rdf:ID="Image">
       <rdfs:subClassOf
rdf:resource="#Resource" />
       <rdfs:label
xml:lang="en">Electronic
Image</rdfs:label>
       <rdfs:comment
xml:lang="en">Resource    for    Electronic
Images </rdfs:comment>
  </rdfs:Class>

  <!-- Location Change Definition -->
  <rdfs:Class rdf:ID="LocationChange">
       <rdfs:subClassOf
rdf:resource="#Image" />
       <rdfs:label      xml:lang="en">Image
Location Change</rdfs:label>
       <rdfs:comment    xml:lang="en">Image
Location Change</rdfs:comment>
  </rdfs:Class>

  <!-- Profile Property Definition -->
  <rdf:Property rdf:ID="profile">
   <rdfs:label         xml:lang="en">Image
Profile</rdfs:label>
   <rdfs:comment xml:lang="en">
      Profile   information   for   image.
Includes information such as subject,
title, description, identifier, location,
creator, owner and creation date.
   </rdfs:comment>
   <rdfs:range rdf:resource="#Image"/>
   <rdfs:domain rdf:resource="#Image"/>
  </rdf:Property>

  <!-- Identifier Property Definition -->
  <rdf:Property rdf:ID="identifier">
   <rdfs:subPropertyOf
rdf:resource="#profile" />
   <rdfs:label         xml:lang="en">Image
Identifier</rdfs:label>
   <rdfs:comment xml:lang="en">
      Identifier  for  image.    Maybe  be
unique  or  not.    Could  be  name,  social
security number, or picture number.
   </rdfs:comment>
   <rdfs:range rdf:resource="#Literal"/>
   <rdfs:domain rdf:resource="#Image"/>
  </rdf:Property>

  <!-- Location Property Definition -->
  <rdf:Property rdf:ID="location">
   <rdfs:subPropertyOf
rdf:resource="#profile" />
   <rdfs:label         xml:lang="en">Image
Location</rdfs:label>
```

```
    <rdfs:comment xml:lang="en">
       Location depicted in Image.  Can be
a resource or a literal.
    </rdfs:comment>
    <rdfs:range rdf:resource="#Resource"/>
    <rdfs:domain rdf:resource="#Image"/>
   </rdf:Property>

   <!-- Owner Property Definition -->
   <rdf:Property rdf:ID="owner">
    <rdfs:subPropertyOf
rdf:resource="#profile" />
    <rdfs:label    xml:lang="en">Owner    of
Image</rdfs:label>
    <rdfs:comment xml:lang="en">
       Owner of Image.
    </rdfs:comment>
    <rdfs:range rdf:resource="#Resource"/>
    <rdfs:domain rdf:resource="#Image"/>
   </rdf:Property>

   <!-- Technical Property Definition -->
   <rdf:Property rdf:ID="technical">
    <rdfs:label         xml:lang="en">Image
Technical Information</rdfs:label>
    <rdfs:comment xml:lang="en">
       Technical  information  for  image.
Includes information such as type, format,
size, resolution, and origin.
    </rdfs:comment>
    <rdfs:range rdf:resource="#Image"/>
    <rdfs:domain rdf:resource="#Image"/>
   </rdf:Property>

   <!-- Type Property Definition -->
   <rdf:Property rdf:ID="type">
    <rdfs:subPropertyOf
rdf:resource="#technical" />
    <rdfs:label     xml:lang="en">Technical
Type</rdfs:label>
    <rdfs:comment xml:lang="en">
       Default  is  image.    Reserved  for
future  expansion  of  other  multimedia
types.
    </rdfs:comment>
    <rdfs:range rdf:resource="#Literal"/>
    <rdfs:domain rdf:resource="#Image"/>
   </rdf:Property>

   <!-- Format Property Definition -->
   <rdf:Property rdf:ID="format">
    <rdfs:subPropertyOf
rdf:resource="#technical" />
    <rdfs:label         xml:lang="en">Image
format</rdfs:label>
    <rdfs:comment xml:lang="en">
       GIF, JPG, TIF, PNG, PCX
    </rdfs:comment>
    <rdfs:range rdf:resource="#Literal"/>
    <rdfs:domain rdf:resource="#Image"/>
   </rdf:Property>

   <!-- ImageSize Property Definition -->
   <rdf:Property rdf:ID="imageSize">
    <rdfs:subPropertyOf
rdf:resource="#technical" />
    <rdfs:label         xml:lang="en">Image
Size</rdfs:label>
    <rdfs:comment xml:lang="en">
     Size  of  the  image.    Specify  number
values. ex. 640X320, 2X4
    </rdfs:comment>
    <rdfs:range rdf:resource="#Literal"/>
    <rdfs:domain rdf:resource="#Image"/>
   </rdf:Property>

   <!-- SizeUnit Property Definition -->
   <rdf:Property rdf:ID="sizeUnit">
    <rdfs:subPropertyOf
rdf:resource="#imageSize" />
    <rdfs:label  xml:lang="en">Image  Size
Units</rdfs:label>
    <rdfs:comment xml:lang="en">
       Specify  the  units  that  the  image
size is recorded in. Ex. px, in, cm.
    </rdfs:comment>
    <rdfs:range rdf:resource="#Literal"/>
    <rdfs:domain rdf:resource="#Image"/>
   </rdf:Property>

   <!-- Resolution Property Definition -->
   <rdf:Property rdf:ID="resolution">
    <rdfs:subPropertyOf
rdf:resource="#technical" />
    <rdfs:label         xml:lang="en">Image
Resolution</rdfs:label>
    <rdfs:comment xml:lang="en">
       Specify the resolution of the image
in numeric values, ex. 72, 300.
    </rdfs:comment>
    <rdfs:range rdf:resource="#Literal"/>
    <rdfs:domain rdf:resource="#Image"/>
   </rdf:Property>

   <!--    Resolution    Units    Property
Definition -->
   <rdf:Property rdf:ID="resolutionUnit">
    <rdfs:subPropertyOf
rdf:resource="#resolution" />
    <rdfs:label    xml:lang="en">Resolution
Units</rdfs:label>
    <rdfs:comment xml:lang="en">
       Specify   the   units   that   the
resolution  is  recorded  in.  Ex.  px/in,
px/cm.
    </rdfs:comment>
    <rdfs:range rdf:resource="#Literal"/>
    <rdfs:domain rdf:resource="#Image"/>
   </rdf:Property>

   <!-- Origin Property Definition -->
   <rdf:Property rdf:ID="origin">
    <rdfs:subPropertyOf
rdf:resource="#technical" />
    <rdfs:label
xml:lang="en">Origin</rdfs:label>
```

```
    <rdfs:comment xml:lang="en">
       Specify the origin of the image
such as imaging center, digital camera,
photo camera, etc.  Can be resource or
literal value.
    </rdfs:comment>
    <rdfs:range rdf:resource="#Resource"/>
    <rdfs:domain rdf:resource="#Image"/>
   </rdf:Property>

   <!-- History Property Definition -->
   <rdf:Property rdf:about="history">
    <rdfs:label         xml:lang="en">Image
Location History</rdfs:label>
    <rdfs:comment xml:lang="en">History of
Image Location Changes</rdfs:comment>
    <rdfs:range rdf:resource="#Image"/>
    <rdfs:domain
rdf:resource="#LocationChange"/>
   </rdf:Property>

   <!--  ChangeType Property Definition --
>
   <rdf:Property rdf:ID="changeType">
    <rdfs:subPropertyOf
rdf:resource="#LocationChange" />
    <rdfs:label        xml:lang="en">Change
Type</rdfs:label>
    <rdfs:comment xml:lang="en">
       Type of change. Ex. Move, Add, New
    </rdfs:comment>
    <rdfs:range rdf:resource="#Literal"/>
    <rdfs:domain rdf:resource="#Image"/>
   </rdf:Property>

   <!-- Change Reason Property Definition
-->
   <rdf:Property rdf:ID="changeReason">
    <rdfs:subPropertyOf
rdf:resource="#LocationChange" />
    <rdfs:label        xml:lang="en">Change
Reason</rdfs:label>
    <rdfs:comment xml:lang="en">
       List reasons for image location
changes.
    </rdfs:comment>
    <rdfs:range rdf:resource="#Literal"/>
    <rdfs:domain rdf:resource="#Image"/>
   </rdf:Property>

   <!-- Change Date Property Definition --
>
   <rdf:Property rdf:ID="changeDate">
    <rdfs:subPropertyOf
rdf:resource="#LocationChange" />
    <rdfs:label    xml:lang="en">Date    of
Location Change</rdfs:label>
    <rdfs:comment xml:lang="en">
       List Date of location change.
    </rdfs:comment>
    <rdfs:range rdf:resource="#Literal"/>
    <rdfs:domain rdf:resource="#Image"/>
   </rdf:Property>

   <!-- Related Property Definition -->
   <rdf:Property rdf:ID="related">
    <rdfs:label       xml:lang="en">Related
Resources</rdfs:label>
    <rdfs:comment xml:lang="en">
       Resources that are related to the
Image.
    </rdfs:comment>
    <rdfs:range rdf:resource="#Resource"/>
    <rdfs:domain rdf:resource="#Image"/>
   </rdf:Property>

   <!-- Related Reason Property Definition
-->
   <rdf:Property rdf:ID="reason">
    <rdfs:subPropertyOf
rdf:resource="#reason" />
    <rdfs:label       xml:lang="en">Related
reason</rdfs:label>
    <rdfs:comment xml:lang="en">
       Reason for relationship
    </rdfs:comment>
    <rdfs:range rdf:resource="#Literal"/>
    <rdfs:domain rdf:resource="#Image"/>
   </rdf:Property>

   </rdf:RDF>
```