A Study of File System Objects' Metadata

Shah Khusro, Syed Rahman Mashwani, Azhar Rauf, Saeed Mahfooz, Shaukat Ali

Department of Computer Science, University of Peshawar, Pakistan
{khusro, syed.rahman, azhar.rauf, saeedmahfooz, shoonikhan}@upesh.edu.pk

Abstract: At this digital age we have more information than all previous generations combined, the major fraction of which reside in the File Systems of our desktops. This huge information is growing day by day and is hard to manage. The roles of metadata cannot be neglected; it assists to deal with the problems that arise from information overload. This paper is a study of File System objects' metadata. Different researcher categorized metadata in the context of their research; we start our discussion with the explanation of almost all types of file's metadata. Then we briefly discuss the existing metadata schemas. We think that the sun of tomorrow does not seem to be shining on semantic less metadata schemas; so, we extend our discussion on the need of semantic (machine friendly) metadata schemas, followed by a discussion on the Semantic Web vocabularies that could be exploited for the purpose.

[Shah Khusro, Syed Rahman Mashwani, Azhar Rauf, Saeed Mahfooz, Shaukat Ali. **A Study of File System Objects' Metadata.** *Life Sci J* 2013;10(11s):343-348] (ISSN:1097-8135). http://www.lifesciencesite.com. 63

Keywords: File System; Semantic File System; Semantic Web; Information Retrieval

1. Introduction

People confront huge information from different sources in this digital age which is becoming difficult to manage. People use to organize their files in different ways according to their expertise. Some people invest time in adding tags and titles to the files. But according to the study [1], people don't label the files and folders with consistent naming conventions. So, the end result is that at most of the times they are unable to locate the files that are organized by their selves. If they are unable to locate their own organized information then how will others do?

Metadata is structured information that describes a digital object [2]. It plays an important role in organizing and retrieving digital objects semantically to overcome the shortcomings of hierarchal File Systems. File metadata can be exploited to improve navigation and searching in File System. There exists a magnitude of metadata schemas in different domains. However, Metadata need to be expressed in a format that is understandable by a large number of applications to overcome data silos and metadata interoperability problems [3]. We restrict our study to the File System objects' metadata; other domains are out of scope of this work.

We organize the rest of the paper as: section 2 describe metadata classification, section 3 describes some popular metadata schemas and problems with the current metadata schemas, section 4 describes Semantic Web vocabularies that could be used in the context of File System and finally section 5 concludes the paper.

2. Metadata Classification

Different researchers have categorized metadata differently in the context of their research. In the following we try to discuss all these categorizations of metadata. Additionally, this section also provides answers to the questions like: what are sources (origin) of metadata? How it is created, maintained (updated) and used? Where it resides? etc.

Function/Purpose Based Classification of Metadata:

Metadata can be categorized according to its proposed functionality such as descriptive, structural, and administrative (rights management and preservation) metadata [4]. Descriptive metadata is the information that is used to search and locate resource such as title, subject, abstract, author, and keywords etc. Structural metadata is the information that indicates how the components of resource are organized or put in order such as the information that tells how to order pages to form chapters. Administrative metadata includes information that assists in management of a resource such as file creation time, file type and access control etc. Administrative metadata is further divided into right management and preservation metadata sub categories. Right management metadata includes information about intellectual property rights while preservation includes information that are required to preserve and for archiving of a resource.

Internal and External Metadata:

Metadata can be stored internally as part of data files or it can be stored separately. Below we discuss both mechanisms in detail separately to understand where metadata could be stored, along with advantages and disadvantages of each.

Internal: Metadata that is embedded in a digital object that it describes is called internal metadata. Internal metadata travels along with file its self. ID3 tags, EXIF data, XMP tags (XMP use RDF as a model for internal metadata) are examples of internal metadata. Oftenly, schemas of internal metadata are fixed but there are also some metadata standards like XMP from adobe that also provide support for the inclusion of custom metadata. Unlike external metadata, as internal metadata is embedded in File System object it is about, so it does not need any special action to maintain the association of data and metadata.

One main problem with existing file metadata is that mostly these are application dependant and proprietary. In order to make use of it applications would need to know the semantic s of such schemas. And application would also need to know how to read/write a particular file format for the purpose of consuming its internal metadata. Unlike external metadata another problem with internal metadata is access restriction policies which are imposed on the whole file including internal metadata. It is not possible to define separate policies for metadata that allow applications to access internal metadata [5].

Extended Attributes and file folks can also be used for storing internal metadata in the File System objects. The metadata that can easily be re-generated from file data are oftenly placed in extended attributes/forks by applications. So that if in case metadata is lost, could be reproduced.

Internal metadata could be edited via various tools and utilities. But if you are not an owner of the digital object then you should know exactly what you are edit because modifying some of the fields of internal metadata like copyrights, owner etc. is against the law1 [6].

External: Metadata that is stored outside the File System object that it describes is called external metadata. It is easier to manage and consume such metadata as it resides in a repository separately from data. Applications can easily read/write external metadata with the need to know about the file format, if they understand the semantics of metadata schema. It enables the development of general metadata model that may be used by all file formats independently of their file formats. Unlike internal metadata, external metadata is independent of restrictions defined on file data. Restrictions on external metadata and its file data can be defined separately so if in case restrictions are defined on File System objects, applications can still access the file meta for indexing or any other purpose [5].

External metadata do not travel along with file its self, it is lost if file is copied to external media, sent via email etc. There is need some heuristics that extract file related metadata from repository and compress it so that it could be send via email copied to external media etc. Bernhard Schandl [3,7] does the same with his siles in the prototypic implementation that wraps a set of siles into a zip file (including its full RDF descriptions), so that it can be sent via email or copied to an USB stick etc. On a target system it can again be imported into the sile repository. Some heuristics are desired to make external metadata portable without the involvement of user.

There is also need of some mechanisms to keep the associations between metadata and the data stable and updated. The connection between external metadata and files may break, if files are moved and renamed etc or if the metadata repository its self is moved from its location. In former case, to make external metadata dependent applications operate/respond properly, it must be ensured that metadata repository must kept be updated probably at real time as File System objects moved, deleted, updated or renamed. In later case, applications may not operate correctly if the repository its self is renamed or moved. For instance, file versioning system SVN stores metadata in a hidden (.svn) subfolder. Applications use this subfolder to compare the versions of the files with SVN server. If this folder is renamed, moved or deleted then applications like Tortoise SVN would not show the SVN Update option, if right mouse button is clicked in the SVN folder using file manager. To maintain the integrity of the associations between files and their external metadata records, Niko Popitsch uses Gorm in Y2 [5] and Bernhard Schandl uses DSNotify in TripFs [3].

Intrinsic, extrinsic, implicit and explicit metadata:

Metadata can be classified according to production of metadata or the way metadata is created (automatically or by human intervention) [8]. We also discuss which metadata could be used directly (in a form it is available) or need any further processing.

Metadata that is extracted directly in automated way from file data is called intrinsic metadata. And unlike intrinsic, extrinsic metadata needs the involvement of human to be assigned to a file. Intrinsic metadata can be re-generated from file data if lost, while extrinsic metadata cannot be re-generated. Examples of intrinsic metadata include thumbnail, width, height etc of an image, number of pages, words, lines etc in a document, length of an audio or video file etc. While tags assigned to a file by a user manually is an example of extrinsic metadata.

[1] Digital Millennium Copyright Act (DMCA)

The already extracted metadata from a file is called explicit. Explicit metadata can directly be accessed and usable without any further processing while on other hand implicit metadata is not pre-extracted from file and thus not directly be usable [5]. For example, in EXIF headers the image width, height, camera model, date and time picture taken etc are explicit metadata while any other information that is derived from date and time, geo information etc would be implicit metadata.

Low Level and High Level Metadata:

Metadata can be classified according to its level of semantic abstraction such as low level and high level metadata [8]. The Low level metadata, machines correctly interpret the intended meanings of metadata while human may not correctly interpret the metadata according to its intended meanings. Examples of low level metadata include mime type, image width/height, file sizes, hierarchical parent/child relationships, GPS coordinates embedded in image files, file extension, file ownership or access rights etc.

In high level both machines and humans need to understand the intended meanings of metadata. High level metadata carry more value for the end users as compared to low level technical metadata [9]. GPS coordinates may carry no or less value for humans than the name of places located at these coordinates. Rating (five star rating) of an audio mp3 file and other tags assigned to a file etc are examples of high level metadata.

Generalized and Specialized (Domain or file type depended) Metadata:

Metadata can also be classified according dependency [8] or the scope of metadata. Some metadata are widespread and general purpose and may apply to a wide variety of file formats. Such as the scope of creation date etc is very diverse and can be used with almost any data type. While on other hand some metadata are specialized and can only be used in a specific domain. Such as, the number of slides in a file may be only useful for presentation related documents such as Microsoft Power Point etc and the position of tumor is possible to be valuable only for medical applications.

Property, Content and Context Based Metadata:

Every file has property, content and context based metadata. Property based metadata describes the contents of the file but are not actually extracted from the content. Property based metadata is further divided into regular attributes and extended attributes. Regular attributes are the attributes that are common in all types of files and are strictly defined by File System; while extended attributes are file type specific and vary accordingly.

Metadata that is extracted from the contents of the file is content based metadata. It may include the internal organization of the file or statistical information about the contents of a file. For instance, words frequency in a document, music tempo of an audio and subtitles or images in a video etc. Context metadata is the contextual information of a file. It includes information about the existence and usage of the file, its relations with other objects etc [10,11].

Source (File System, application, hardware and User) based classification of Metadata:

Metadata can be classified on the basis of its source from where it is supplied. This includes File System, application, hardware and User. File System based metadata is created and strictly maintained by File System about File System objects such as, time stamp and access control etc of an object.

Application based metadata about a resource is associated and may be maintained by an application. This includes author's name, initial and time stamp etc of a document. Document created date that is associated by application may reflect the exact time as compared to created time that is associated by File System. Because the value of created, modified and accessed time maintained by File System does change if the file travel to another volume, sent and copied to another desktop computer over email or USB drive, copied from web etc. But the value of created time that is associated by application (Microsoft Word) or hardware device (digital camera) does not affect if moved to somewhere else.

Some metadata is also generated by hardware devices such as by digital camera etc. Information that is associated by image capturing device may include camera maker, camera model, flash mode, accelerometer, digital compass and GPS etc. Some hardware generated metadata is also fruitful to discover the context of a file. Metadata is also generated by user such as, adding tags and other properties to a file by a user. Assigning tags makes the discovery of file easy via search or any other application. Tags or other properties of a file (i.e. image file) can be added or changed via details pane located at the bottom of file manager (in Windows 7), or via properties dialog box or while saving a file [12]. These assigned tags make the files searchable via search application for instance, current Windows 7 search application etc.

Physical metadata, built-in content metadata and user-defined content metadata

Zhihong Shen et al. [13] divide files metadata into physical metadata, built-in content metadata and user-defined content metadata. Physical metadata describes the physical properties of the file, such as: file name, path, type, creation time, modification time etc. Built-in content metadata are the embedded attributes of the file, which is normally stored as part of the file. This may includes the ID3 and EXIF info of an mp3 and JPEG respectively. And finally, the authors define the user-defined content metadata as the metadata that is defined by the user for the file. Unlike built-in content metadata, the User-defined content metadata is stored externally to the original file.

3. Existing (traditional) Metadata Schemas

Metadata schemas (schemes) are sets of metadata elements describing particular type of information resource [4]. There exists a magnitude of file metadata schemas to describe files. In the following we briefly discuss some popular metadata schemas.

EXIF:

EXIF is de facto standard specially developed for the description of digital still images. It allows information such as the title, copyrights, GPS, temporal, manufacturer, width and height, thumbnail etc about still image to embed in the file itself. However, EXIF is also used for the description of audio files. It is supported by almost all cameras, smart phones, scanners and other images related manufacturers. Software applications such as Picasa View and some popular search applications exploit EXIF data for the purpose of managing and retrieving of digital still image files.

ID3:

ID3 [14] is de facto standard specifically designed for the description of MP3 files. It allows information such as the title, year, genre, artist, album etc about MP3 audio file to embed in the file itself. Incorporating descriptive information into audio files is called tagging. Windows Media Player, Winamp, MediaMonkey, iTunes and many other software applications exploit ID3 tags for managing MP3 files.

MPEG7:

MPEG7 (formally named "Multimedia Content Description Interface") is an ISO/IEC standard developed by MPEG for the description of multimedia content. Unlike MPEG1, MPEG2 and MPEG4, It does not deal with the encoding of the multimedia files but it represents information about the content to allow fast and efficient searching. So, MPEG7 could be used to improve the functionalities of previous MPEG standards. It uses XML to store metadata [15].

However, detail discussion on the existing metadata schemas is out of scope of this paper, the intent is to highlight their problems in the context of our paper. The major problems with the existing file metadata schemas are that they are mostly proprietary, application dependant and are not machine friendly. In order to make use of metadata, applications would need to know the semantic s of such schemas. And For the purpose of consuming internal metadata application would also need to know how to read/write a particular file format. Imposing access restriction policies on the file makes its (internal) metadata out of reach of the applications. New relations can't be inferred on the basis of existing relations. They do not facilitate to retrieve information semantically via complex queries; for instance, get me pictures of my fifth wedding anniversary that were taken at xyz restaurant of Paris towards Eifel Tower in the evening with my wife's iPhone. There exist more than one metadata schemas for single purpose; interoperability among different metadata schemas is desired. Metadata need to be expressed in a format that is understandable by a large number of applications to overcome data silos and metadata interoperability problems [7]. Linking metadata elements internally (within the same File System) and externally (other File Systems and web of data) also need to be considered.

4. Semantic Web Vocabularies /Semantic Metadata

Semantic metadata is built using ontologies which makes it machine friendly; means that machine can read, understand and process it [16]. RDF based description provides more semantics and enables semantic applications to reason over the metadata and infer new relations on the basis of existing relations. In the following we discuss some popular Semantic Web vocabularies that could be applicable in the context of File System (but are not limited to):

Friend of a Friend (FOAF):

FOAF is RDF/OWL based vocabulary that provides a variety of terms describing people, relations between them and the things they create or do. FOAF also describes organization, group, project and document but it's the main focus of is describing people. FOAF is a widely used vocabulary, many web and desktop based applications are aware of it.

There also exist a number of tools to create FOAF file very easily. For instance, FOAF-a-Matic2 is a JavaScript application with which one can create FOAF description of himself easily by entering natural language text information in a web form. After

[2] http://www.ldodds.com/foaf/foaf-a-matic Cited March, 2013

creating FOAF you can make tools to easily discover your FOAF by putting “<link rel="meta" type="application/rdf+xml" title="FOAF" href="foaf.rdf" />” markup in you the head of html homepage [17]. FOAF in our context could be used to relate/make connections between File System resources and persons, groups or organizations.

*Dublin Core Metadata Terms (DC-**terms)**:*

DC-terms are a set of vocabulary terms which efficiently describe physical and digital objects in diverse domains. DC-terms is most widely used semantic vocabulary describing all types of resources [5] [2]. It could be used to describe File System resources in broader and generic way.

NEPOMUK File Ontology (NFO):

NFO is specifically designed to describe the contents of File System. It provides terms to describe files, folders and their properties. It also provides vocabulary to describe remote files, compressed files and files attached to other objects.

NEPOMUK Annotation Ontology (NAO):

NAO as the name shows is designed to annotate desktop resources. NAO provides vocabulary that enables users to provide labels, descriptions, tags and ratings to desktop resources. Annotations can be textual or non-textual. A textual annotation is human-readable annotation that relates a resource to a literal. Non-textual annotation means a semantic annotation pointing to a resource. We can use the vocabulary provided by NAO to annotate File System resources as well as to make relations between resources.

NEPOMUK Contact Ontology (NCO) and VCard Ontology:

NCO intends to provide vocabulary for describing contact information. Contact information includes phone number, IM account, postal and email addresses etc. VCard ontology is the RDF/OWL mapping to vCard specification (RFC6350) and also used for the same purpose. Contact information is the most important element of every Personal Information Management system. These vocabularies could be used to relate File System resources to agents (individuals, groups, organizations).

NEPOMUK Calendar Ontology (NCAL):

NCAL provides vocabulary to describe calendaring entries i.e. events, tasks to do etc. In File System, files could be shuffled semantically in the hierarchy according to the users entered calendaring data. For instance, to make the meeting related files prominent in the File System hierarchy at meeting time.

NEPOMUK ID3 Ontology (NID3), Music Ontology and NEPOMUK EXIF Ontology (NEXIF):

Both the NID3 and Music Ontology intend to provide vocabulary for describing audio/music files. NID3 is the RDF mapping to existing ID3 metadata standard, so it enables to express ID3 information in RDF. Similarly like NID3, NEXIF is the RDF mapping to existing EXIF standard. NEXIF provides vocabulary to describe images files. All these vocabularies could be exploited in File System to describe audio and image files.

5. Conclusions

Metadata has a key role to deal with the tsunami of information. It helps in the organizing and retrieving of the information in the File Systems of our desktops. In this article, we carried out a detailed discussion on the metadata of files. Different researchers categorized metadata in the context of their research work; we have studied and discussed about all types of file metadata in detail. The article also discussed the current popular metadata schemas and the drawbacks of semantic less metadata schemas. Then we extended our discussion to the advantages of metadata schemas based on Semantic Web technologies. And at the end, we highlighted some Semantic Web vocabularies that could be used for the purpose.

Corresponding Author:
Syed Rahman Mashwani
Department of Computer Science, University of Peshawar, Pakistan
E-mail: syed.rahman@upesh.edu.pk

References:

1. Whittaker S, Bergman O, Clough P (2010) Easy on that trigger dad: a study of long term family photo retrieval. Personal and Ubiquitous Computing 14: 31-43.
2. Koutsomitropoulos DA, Solomou GD, Papatheodorou TS (2009) Metadata and semantics in digital object collections: A case-study on CIDOC-CRM and Dublin Core and a prototype implementation. Journal of Digital Information 10.
3. Schandl B, Haslhofer B (2010) Files are Siles: Extending File Systems with Semantic Annotations. International Journal on Semantic Web and Information Systems (IJSWIS) 6: 1-21.
4. NISO. Understanding metadata; 2004. NISO Press. http://www.niso.org/standards/resources/UnderstandingMetadata.pdf
5. Popitsch N (2011) Building Blocks for Semantic Data Organization on the Desktop [Doctoral Dissertation]: University of Vienna.

6. Riecks D. The Top 12 Myths about Embedded Photo Metadata. http://www.controlledvocabulary.com/blog/top-metadata-myths.html
7. Schandl B, Haslhofer B (2009) The Sile Model: A Semantic File System Infrastructure for the Desktop. 6th European Semantic Web Conference (ESWC2009). Heraklion, Greece: Springer.
8. Westermann U, Klas W (2003) An analysis of XML database solutions for the management of MPEG-7 media descriptions. ACM Computing Surveys (CSUR) 35: 331-373.
9. Haslhofer B, Klas W (2010) A survey of techniques for achieving metadata interoperability. ACM Computing Surveys (CSUR) 42: 7.
10. NGO HB, Bac C, Silber-Chaussumier F, Le TQ. Towards ontology-based semantic file systems; 2007. IEEE. pp. 8-13.
11. Xu Z, Karlsson M, Tang C, Karamanolis C (2009) Semantic file system. United States: Hewlett-Packard Development Company, L.P.
12. Microsoft. Add tags or other properties to a file. http://windows.microsoft.com/en-au/windows-vista/add-tags-or-other-properties-to-a-file
13. Shen Z, Hou Y, Li J. Publishing distributed files as Linked Data; 2011. IEEE. pp. 1694-1698.
14. Nilsson M, ID3-Contributors. id3v2; . http://id3.org/
15. Mallorca Pd. MPEG-7; 2004. http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm
16. Al-Khalifa HS, Davis HC. The evolution of metadata from standards to semantics in E-learning applications; 2006. ACM. pp. 69-72.
17. Brickley D, Miller L (2010) FOAF vocabulary specification 0.98. Namespace Document 9.

11/6/2013