Publishing Distributed Files as Linked Data

Zhihong Shen, Yufang Hou, Jianhui Li
Scientific Data Center
Computer Network Information Center, CAS
Beijing, China

Abstract—**File systems act as backbones of many information systems, however, it is still hard to search and integrate information about files which are distributed in different locations over different protocols. Traditional researches about file systems focus on file sharing and transfer, paying little attention to semantic annotation and interlinking of files in distributed environment. In this paper, we study how to publish and consume files as Linked Data. Some principles are discussed in detail, including choosing appropriate HTTP URIs, providing metadata and content of files, generating external links between files and other resources, connecting real repositories located in different places. Then we present VDB-FilePub, a component of VisualDB for publishing the files stored in different repositories as Linked Data. After introducing the architecture of VDB-FilePub, we give some scenarios of how other applications consume Linked Data exposed by VDB-FilePub in the Scientific Database Project.**

Keywords-Linked Data, file systems, content metadata, metadata extraction, file statistic, scientific data

I. Introduction and Motivation

Computer users are accustomed to use file systems to organize and store their data. Files such as excel sheets and JPEG images are widely used, even though full-fledged DBMS (Database Management System), GIS (Geographic Information System), XML/RDF stores and other information systems are provided. File systems keep playing key roles in our lives because of their simple interfaces to humans and programs.

In the Scientific Database Project [1] supported by CAS (Chinese Academy of Sciences) from 2005 to 2010, more than 500 special databases were created by 51 institutions, plus other data which archived by other information systems, the total volume of data is nearly 160 TB. A large portion of these data [2], such as captured images, digital videos, spatial data and other data generated by huge scientific instruments, are organized directly by file systems. So it brought a challenge for us to make these distributed files be accessible and searchable semantically through a unified interface.

Some problems occur when we consider file systems as knowledge organization and discovery system. Firstly, file systems only provide a single hierarchical schema with a tree of directories. Secondly, metadata about files in file systems are limited in few physical attributes such as filename, creator, size, and so on. Apparently the limited metadata about files are not enough for discovery. Furthermore, archived files are often located at different places and published using different protocols such as SAMBA, FTP, WebDAV and HTTP. If an application wants to consume these files, it has to consider all kinds of protocols, and user authentication systems. So it is necessary to find a cross-protocols solution to make the distributed files be accessed transparently and searched conveniently.

Linked Data is such a promising solution, more and more people are encouraged to publish their datasets as Linked Data [3]. In this paper, VDB-FilePub is presented as an approach to publish files as Linked Data. The remainder of the paper is organized as follows. In Section 2, we review related work. In Section 3, we discuss the key steps about how the VDB-FilePub works. In Section 4, we present details about the VDB-FilePub architecture and implementation. In Section 5, some application scenarios of consuming Linked Data interface of files over VDB-FilePub are given. In Section 6, we discuss the future work.

II. Related Work

Most researches about File Systems focus on file sharing and transfer. Protocols such as NFS(Network File System), CIFS(Common Internet File System), SAMBA(Server Message Block) and WebDAV(Web-based Distributed Authoring and Versioning) are invented for sharing files from different computers; others such as FTP, BitTorrent and GridFTP provide the solutions for high-speed and stable file transfer.

On the other hand, metadata extraction from files is the hot topic in digital library and computer science. DROID (Digital Record Object Identification) [4], a tool developed by The National Archives, performs automated batch identification of file formats. NLNZ-MetadataExtractor [5] developed by National Library of New Zealand can extract metadata from images, office documents, audio and video data, markup languages, internet files stored as ARC format, and so on. Aperture [6], a java framework for extracting and querying full-text content and metadata from various information systems, supports many file formats including plain text, HTML, XHTML, XML, RDF(Portable Document Format), RTF(Rich Text Format), Office, presentations, email, ical files, VCARD files, archives(zip, tar, gz, bz2) and so on.

As Semantic Web technologies are getting mature, a lot of RDFizers appear, they generate RDF representations for extracted metadata. ESW Wiki ConverterToRdf [7] and SIMILE RDFizer [8] list some RDFizers aiming at common file formats, such as JPEG, MARC/MODS, OAI-PMH, Email, iCanlendar, BibTex, and so on. Following the publishing Linked Data guidelines [9], some tools are developed to publish different types of information as Linked Data. OAI2LOD Server [10] exposes a Linked Data interface for any OAI-PMH compliant metadata repository. D2RQ Platform [11] and Triplify [12] provide frameworks to publish relational databases to Linked Data automatically. iM(interlinking Multimedia) [13] applies the Linked Data principles to fragments of multimedia items. Also some solutions [14] are proposed to publish real-time data streams as Linked Data on the fly.

Similar to our proposal, TripFS [15] provides a solution which exposes file systems as Linked Data. Compared with TripFS, our contribution in this paper can be summarized as follows. First, our proposal supports the description and publishing of user-defined content metadata of files, thus user can add semantic annotations for files. Second, we take into account file access out of the publisher server, across different protocols and user authentication systems. Third, we give some application scenarios to show how to consume the Linked Data exposed by VDB-FilePub.

III. Representation of Linked Files

Some key problems must be solved in order to publish files as Linked Data:

- According to principle No.1 and No.2 of Linked Data [16], accessible HTTP URIs for files should be provided;
- According to principle No.3 (provide useful information), metadata about files should be extracted or created and transformed into RDF model. Users could define special metadata for their files;
- According to principle No.4 (include links to other URIs), links among files, and links between files and external data sources should be established and published;
- The content of files should be provided in an appropriate way;
- Provide a unified interface for upper application to access files without concerning about how they are connected to the publisher server.

In the following we specify how these problems are solved.

A. URIs

We propose the URI for files as below:

<base url>/resource/file/<repository name>/<file id>

Here repository name is the local mapping name for remote file server directory, file path is the relative path referred to the remote file server directory.

An example of URI is shown as below:

http://www.plant.csdb.cn/resource/file/repo1/1f62836e1c020c57011c020c57700000

For each URI, the responding URLs for RDF/XML and HTML representations are proposed like:

http://www.plant.csdb.cn/data/file/repo1/1f62836e1c020c57011c020c57700000

http://www.plant.csdb.cn/page/file/repo1/1f62836e1c020c57011c020c57700000

The original file path such as *mypics/ginkgo.jpg* has to be encoded as *1f62836e1c020c57011c020c57700000* to keep every URI stable, and avoid problems brought by special characters in the path.

B. Metadata of Files

We divide metadata of files into three kinds: physical metadata, built-in content metadata and user-defined content metadata.

Physical metadata describe physical properties about files, which include file name, path, content length, creation date, modification date, content type, and file type (directory or file).

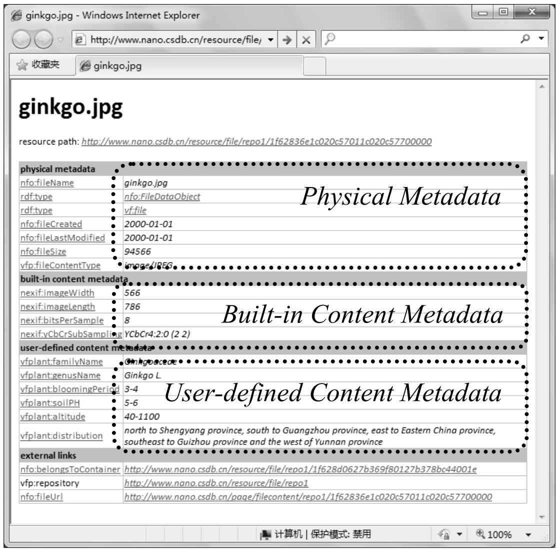
Built-in content metadata describe embedded attributes of files, which are often stored as a part of original files. For example, the EXIF info of a JPEG picture captured by camera is “built-in content metadata”, it includes attributes such as camera model, aperture, shutter speed, focal length, ISO metering, date, and so on.

But the embedded attributes are not enough for describing all information of files. Take the JPEG picture mentioned above as an example, except content metadata referred above, people need to know more about what does this JPEG picture talk about. Is it about a plant or an animal? So we need more metadata to describe it, we define this kind of metadata as "user-defined content metadata", which is often stored out of the original file. Fig. 1 shows an image with user-defined content metadata.

Family Name: *Ginkgoaceae*
Genus Name: *Ginkgo L.*
Blooming period: *3-4*
Soil PH: *5-6*
Altitude: *40-1100*
Distribution: *north to Shenyang province, south to Guangzhou province, east to Eastern China province, southeast to Guizhou province and the west of Yunnan province*

Figure 1. An image of a ginkgo, always with user-defined content metadata. The image is cited from http://www.plant.csdb.cn/sdb/teyou/zgty001a.jpg.

Fig. 2 shows the mixed metadata of a JPEG image in VDB-FilePub.


ginkgo.jpg - Windows Internet Explorer
http://www.nano.csdb.cn/resource/file/
收藏夹
ginkgo.jpg
Physical Metadata
Built-in Content Metadata
User-defined Content Metadata
计算机 | 保护模式: 禁用
100%

ginkgo.jpg

resource path: http://www.nano.csdb.cn/resource/file/repo1/1f62836e1c020c57011c020c57700000

physical metadata	
nfo:fileName	ginkgo.jpg
rdf:type	nfo:FileDataObject
rdf:type	vf:file
nfo:fileCreated	2000-01-01
nfo:fileLastModified	2000-01-01
nfo:fileSize	94566
vfp:fileContentType	image/JPEG
built-in content metadata	
nexif:imageWidth	566
nexif:imageLength	786
nexif:bitsPerSample	8
nexif:yCbCrSubSampling	YCbCr4:2:0 (2 2)
user-defined content metadata	
vfplant:familyName	Ginkgoaceae
vfplant:genusName	Ginkgo L
vfplant:bloomingPeriod	3-4
vfplant:soilPH	5-6
vfplant:altitude	40-1100
vfplant:distribution	north to Shengyang province, south to Guangzhou province, east to Eastern China province, southeast to Guizhou province and the west of Yunnan province
external links	
nfo:belongsToContainer	http://www.nano.csdb.cn/resource/file/repo1/1f628d0627b369f80127b378bc44001e
vfp:repository	http://www.nano.csdb.cn/resource/file/repo1
nfo:fileUrl	http://www.nano.csdb.cn/page/filecontent/repo1/1f62836e1c020c57011c020c57700000

Figure 2. Mixed metadata of an image in VDB-FilePub

VDB-FilePub enables users to define different metadata templates and assign one of them to each directory or file. So the form of user-defined content metadata is fully decided by users' intention. When defining a new metadata template, users should specify name, URI and value range for each property.

C. *Publishing External Links*

Links exist between files and directories inherently. VDB-FilePub uses nfo:belongsToContainer, tripfs:child to describe the parent-child relationship between a file and its parent directory. For each RDF resource, the nfo:fileUrl attribute is also provided to describe the download path for the source file.

More important information to be linked from a file is about "where is it from" and "how to use it". Provenance information helps people to know "where does a file come from" and "how does it come". In a general provenance model such as OPM (Open Provenance Model) [17] and Provenance Vocabulary [18], a file, often acts as an Artifact, is always linked to some Processes which use or generate the file and Actors who control the Processes. On the other hand, usage information plays important role by telling "how to use a file" including what parser or tool should be selected to read the file. These two kinds of links make files more available in scientific research processes.

Furthermore, when entities are extracted from the user-defined content metadata of a file, some new links can be generated automatically. By these links, consumers may navigate an image of a ginkgo to an introduction of ginkgo easily, and go further to a geographic introduction of the Tianmushan Mountain where ginkgo trees are distributed.

However, it is still a difficult task to generate links from files to provenance information, usage information and entities described in external data sources. Some studies focus on this issue and some tools and components are developed. Instead, VDB-FileFub provides an integration framework for such third-party software and a universal interface for storing and publishing external links of different types.

D. *Providing Content of Files*

Should content of files be represented as RDF statements? Many people have studied this subject a lot. As a part of W3C EARL(Evaluation And Report Language) , a specification for a vocabulary to represent content in the RDF [12] is proposed by the ERT WG(Evaluation and Repair Tools Working Group), in which several content envelope methods are discussed: cnt:ContentAsBase64, cnt:ContentAsText and cnt:ContentAsXML. An example using cnt:ContentAsBase64 to describe content is shown as below:

<cnt:ContentAsBase64
rdf:about="http://www.w3.org/Icons/w3c_home.png">
<cnt:bytes>77+9UE5HDQoaCgAAAA1JSERS{...}</cnt:bytes>
</cnt:ContentAsBase64>

In this paper, we do not provide Linked Data interface for file content, instead, a normal HTTP URL such as *http://www.plant.csdb.cn/asis/file/repo1/1f62836e1c020c57011c020c57700000* to download file is directly provided since HTTP/1.1 protocol [19] has already provided sufficient support to transfer file content.

E. *Serving FTP and WebDAV Files*

In this paper, a unified interface to access files stored at different file servers is provided. Up to now, FTP and WebDAV file systems are supported. When a user submit a request on a file by a URI, the server will parse the URI first, determine the location of this file, establish a connection to the file server and access the file using a specific connector. So we maintain a table for each repository. An example of such table is shown as Table 1.

TABLE I. A LIST OF FILE REPOSITORIES WHICH ARE DISTRIBUTED IN DIFFERENT SERVERS

NAME	KIND	HOST	PORT	ROOT_DIR	USER	PASS
repo1	LOCAL	-	-	/usr/bluejoe	-	-
repo2	FTP	192.168.0.13	21	/plants	ftpuser	****
repo3	WEBDAV	dav.csdb.cn	80	/photos	davuser	****

Considering the extension ability for other repositories, we provide another table to maintain the relationship between the repositories type and the connectors, which is shown as Table 2.

TABLE II. A MAP OF REPOSITORY KINDS TO CONNECTORS

REPOSITORY_KIND	CONNECTOR_CLASS
LOCAL	vfp.connectors.LocalFileConnector
FTP	vfp.connectors.FTPConnector
WEBDAV	vfp.connectors.WebDAVConnector

IV. IMPLEMENTATION

The VDB-FilePub has been implemented in pure Java. We store RDF data based on Jena Framework, and use XML files to store user-defined content metadata and their templates. The system's architecture is shown in Fig. 3.

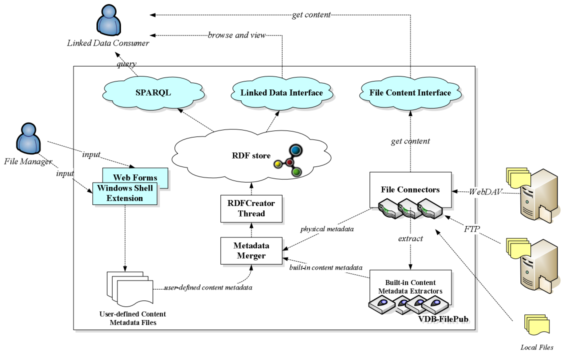


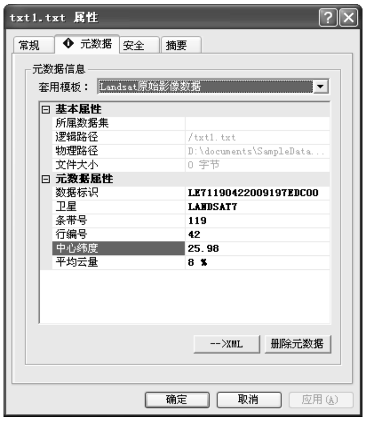
Figure 3. Architecture of VDB-FilePub

VDB-FilePub provides several file connectors for different types of file repositories. These connectors support high level applications to access local or remote directories and files in a same manner. VDB-FilePub also manages a multitude of built-in content metadata extractors. For typical document-like formats, such as word processor documents, spreadsheets and presentations, the extractors are simply derived from aperture extractors [20].

The core publishing work is done by a worker thread called RDFCreatorThread, which runs periodically (for example, once a week). At the specified interval, RDFCreatorThread drives connectors and picks all files in all repositories one by one, calls a MetadataMerger to combine the physical metadata, built-in content metadata and user-defined content metadata of a file together, encapsulates them as RDF data, and stores all data in the RDF store.

We use the VTL (Velocity [21] Template Language) to implement the HTML presentation for files. When the user requests for HTML presentation of a file, the server picks up the RDF resource from the RDF store, loads the VTL parser to parse the VTL template, generate the HTML page returned to the browser. The VTL grammar is very simple so it is easy for users to customize their HTML styles.

We have also implemented a shell extension program for Windows which is called Vfp4Win. Windows users could input user-defined content metadata for local files in Windows Explorer. All templates and user-defined content metadata are stored as XML files in certain directories according to different repositories. Fig. 4 shows a screenshot of the “property dialog box” of a file in Windows Explorer, in which users can input metadata attributes conveniently.



套用模板： Landsat原始影像数据

⊟ 基本属性	
所属数据集	
逻辑路径	/txt1.txt
物理路径	D:\documents\SampleData...
文件大小	0 字节
⊟ 元数据属性	
数据标识	LE71190422009197EDC00
卫星	LANDSAT7
条带号	119
行编号	42
中心纬度	25.98
平均云量	8 %

Figure 4. Vfp4Win, a Windows shell extension program which allows users input metadata of file in Windows Explorer.

V. CONSUMING LINKED FILES

As a component of VisualDB [22], a tool developed by SDC, CNIC, aiming at managing and publishing relational database and files, VDB-FilePub is originally designed to support the publishing of files, especially they are distributed in different research institutions.

In the Scientific Database Project, we have deployed VisualDB2.0 in 47 research institutions to help them publish their data in file systems and databases as Linked Data. Some upper level applications come after that by consuming all these online formatted data. Furthermore, we developed a semantic search engine voolve2.0 [23], which provides integrated searching about scientific files and databases to end users.

We also use Linked Data interface over VDB-FilePub to complete some statistics work about file datasets. Some attributes such as file size, content type and the number of files, could be used as statistical indicators, then we can get some figures about a file dataset. For example, we could know the size of a specific file dataset, and the proportion of image files in this dataset. Fig. 5 shows a pie chart about different file types of a scientific dataset in scientific resource statistic platform [2], which tells that most files are in hierarchical data format.

The statistics output is more meaningful when we add new dimensions to the statistical indicators. For example, we contrast sizes and numbers of files between different datasets. Also we find the change trends of one dataset by comparing the file number and size at different phrases. Fig. 6 shows a line chart about changes of one dataset size at different time.

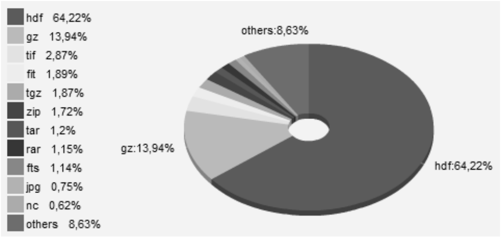


Figure 5. A statistical chart about the content type of a scientific dataset

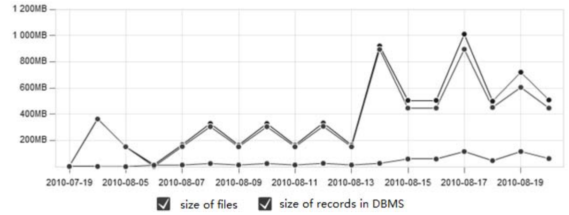


Figure 6. A statistical chart about changes of one dataset size at different time

VI. Future Work

In future work, we plan to improve the graphic interface for user-defined content metadata template management and metadata input; further we plan to provide or import more built-in content metadata extractors. New file connectors will be planned to be provided to make VDB-FilePub able to connect more repositories of new types (for example, connecting directories and files served by an Apache HTTP server in a local area network). Additionally, we will study how to control the access to Linked Data to protect the metadata and content of files.

Finally, the most important, we will study some link discovery algorithms and frameworks (such as SILK [24]) and introduce them into VDB-FilePub to generate some external links based on built-in content metadata and user-defined content metadata. Links between different types of resources are more complex and valuable. For example, linking from the image of ginkgo to scientific data about the climate, soil of its place, people and applications could discover more knowledge. In the future, we will provide a user interface for users to specify some link discovery rules. When the files are published, RDF links generated automatically by these rules will be published together.

Acknowledgment

The work is supported by the “Data Application Environment for Science Research” Project, CAS, 2005-2010(Project No: INFO-115-C01) and the “Specification and Service Platform, Basic Science Data Sharing Service System” Project (Project No: BSDN2009-17).

References

[1] Data Application Environment for Science Research Project. Available: http://www.csdb.cn
[2] Scientific Resource Statistic Platform. Available: http://resstat.csdb.cn
[3] C. Bizer, et al., "Linked Data - The Story So Far," presented at the International Journal on Semantic Web and Information Systems(IJSWIS), 2009.
[4] DROID. Available: http://sourceforge.net/projects/droid/
[5] Metadata Extraction Tool. Available: http://meta-extractor.sourceforge.net/
[6] Aperture Framework. Available: http://aperture.sourceforge.net
[7] ConverterToRdf. Available: http://esw.w3.org/ConverterToRdf
[8] RDFizers – SIMILE. Available: http://simile.mit.edu/wiki/RDFizers
[9] How to Publish Linked Data on the Web. Available: http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/
[10] B. Haslhofer and B. Schandl, "The OAI2LOD Server: Exposing OAI-PMH Metadata as Linked Data," presented at the International Workshop on Linked Data on the Web(LDOW2008), Beijing, China 2008.
[11] The D2RQ Platform v0.7 - Treating Non-RDF Relational Databases as Virtual RDF Graphs. Available: http://www4.wiwiss.fu-berlin.de/bizer/d2rq/spec/
[12] Triplify, http://triplify.org/
[13] M. Hausenblas and R. Troncy, "Interlinking Multimedia - How to Apply Linked Data Principles to Multimedia Fragments," presented at the International Workshop on Linked Data on the Web(LDOW2009), Madrid, Spain, 2009.
[14] D. F. Barbieri and E. D. Valle, "A Proposal for Publishing Data Streams as Linked Data," presented at the International Workshop on Linked Data on the Web(LDOW2010), Raleigh, North Carolina, 2010.
[15] B. Schandl and N. Popitsch, "Lifting File Systems into the Linked Data Cloud with TripFS," presented at the International Workshop on Linked Data on the Web(LDOW2010), Raleigh, North Carolina, USA., 2010.
[16] The D2RQ Platform v0.7 - Treating Non-RDF Relational Databases as Virtual RDF Graphs. Available: http://www4.wiwiss.fu-berlin.de/bizer/d2rq/spec/
[17] The Open Provenance Model (v1.00). Available: http://eprints.ecs.soton.ac.uk/14979/1/opm.pdf
[18] Olaf Hartig: Provenance Information in the Web of Data. In: International Workshop on Linked Data on the Web (LDOW2009). Madrid, Spain; 2009
[19] RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1 . Available: http://tools.ietf.org/html/rfc2616
[20] Extractors – aperture . Available: http://sourceforge.net/apps/trac/aperture/wiki/Extractors
[21] The Apache Velocity Project. Available: http://velocity.apache.org
[22] VisualDB - A Management and Publishing Tool for Scientific Data, Available: http://vdb.csdb.cn
[23] Scientific Data Search Engine. Available: http://voovle.csdb.cn
[24] J. Volz, et al., "Silk – A Link Discovery Framework for the Web of Data," presented at the International Workshop on Linked Data on the Web(LDOW 2009), Madrid, Spain, 2009.