F2R: Publishing File Systems as Linked Data

Shaopeng He
Scientific Data Center
Computer Network Information Center, CAS
University of Chinese Academy of Sciences
Beijing, China

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

Abstract—**As a major role for data storage and knowledge management, file systems have been used both in enterprise contexts and personal information sphere for a long time. However, file systems organize file data using plain file hierarchies and have very little support for semantic annotation, linkage and semantic categorization. It is hard to integrate the file data with the Web of Data. In this paper, we present F2R, a lightweight system for exposing file systems as Linked Data and automatically linking files to DBpedia or external sources. We propose four kinds of file metadata to enrich the information of files and adopt semantic web tools to automatically link to other sources, which enables new possibilities of the Web-based data integration and semantic actions.**

Keywords-Linked Data; file systems; RDF; file metadata; semantic web

I. INTRODUCTION

File systems can be considered as one of the primary means for knowledge organization and storage. Large volume of data such as PDF files, text files and images, are stored as independent files in file systems. Even though many full-fledged storage systems exist, such as DBMS (Database Management System), XML stores and so on, file systems have been widely used for enterprises and individuals, and have stored large data for a long time with the advantage of simplicity and usability.

Traditional researches about file systems mostly focus on file sharing, file transfer and metadata extraction. Some protocols, such as NFS (Network File System), WebDAV (Web-based Distributed Authoring and Versioning), are invented for sharing files between different computers. Other protocols, such as FTP and GridFTP, provide high-speed file transfer. Metadata extraction has been extensively studied in the field of digital libraries and computer science, such as DROID [1] (Digital Record Object Identification), NLNZ-Metadata Extractor [2], Aperture [3] and so on.

File systems provide very few restrictions to users to manage files. Users can specify paths and names as their ideas. Therefore it is difficult to achieve unified management of file data in the heterogeneous environment. Especially, in enterprise contexts, the unified management of files from different departments has become a thorny issue. What's more, the lack of support for semantic annotation leads to less available information. Fortunately, Linked Data provides a solution to integrate file systems with the web of data. Therefore, publishing file systems as Linked Data is of high interest for information integration.

Linked Data [4] provides a light-weight incremental scalable and extensible mechanism for the web of data integration and linkage. As the implementation technology of Semantic Web [5], Linked Data can convert large volume of data, such as unstructured data and structured data that using different specifications, into structured data with the unified standard by RDF [6] (Resource Description Framework) model. It makes file data understandable for computers.

More and more researchers are encouraged to publish their data as Linked Data [7]. In this paper, we propose a light-weight solution to expose file systems as Linked Data by F2R. The rest of paper is organized as follows: Section 2 discusses related work. Section 3 describes the key solutions to realize F2R in detail. Section 4 outlines details about the F2R architecture and implementation. Section 5 outlines some experiments to prove the availability of F2R. Section 6 concludes the paper.

II. RELATED WORK

With the advent of big data, effective storage of large-scale file data becomes a hot topic, which promotes many distributed file systems, such as GFS (Google File System) [8], HDFS (Hadoop Distributed File System) [9], MooseFS [10].

However, due to the limited organization structure of files, weak support for metadata and the lack of stable and unique identifiers for files, file systems have only rarely been considered in the field of Web-based data integration and semantic actions. As Linked Data are getting more mature, some researchers begin to integrate file systems with the web of data by Linked Data.

Some Linked Data tools exist currently to publish data, such as Triplify [11], D2RQ [12], which are mainly concentrated in the relational database. For file systems, B. Schandl and N. Popitsch propose a solution named TripFS [13], which only exposes local file systems as Linked Data. Zhihong Shen and Yufang Hou present a system VDB-FilePub [14]. As a component of VDB System, VDB-FilePub extends to publish distributed file systems as Linked Data. However, for external links detection, TripFS only offers a mechanism of manually specifying appropriate data sources for different files, and VDB-FilePub only provides an interface for third-party implementations to detect links.

In this paper, we propose F2R, a light-weight solution for file systems to expose their data as Linked Data and automatically detect rich external links from DBpedia [15]. We

*Corresponding author: Jianhui Li, E-mail: lijh@cnic.cn

978-1-4673-5253-6/13/$31.00 ©2013 IEEE

also present richer metadata by extracting useful metadata from external sources.

On the basis of previous researchers, we propose some improvements. First, in terms of metadata extraction, we divide file metadata into four categories, such as physical metadata, built-in content metadata, user-defined content metadata and external sources metadata. We extract metadata from external sources to enrich the metadata of files and provide users with a wealth of information. Second, we adopt semantic web tool SPARQL [16] to detect external links, and develop a number of extraction rules, which are used to detect semantically related resources.

III. Representing Files As Linked Data

Some principles must be followed in order to publish file systems as Linked Data [17]:

- Use URIs as names for things.
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful RDF information.
- Include RDF statements that link to other URIs so that they can discover related things.

Based on the above principles, the most important part of publishing file systems as Linked Data is the URIs. We can provide a wealth of useful information, and external links to semantically related resources by the URIs. Therefore we divide the process into four primary tasks:

- Adopt an appropriate manner to represent files.
- Extract metadata from files to provide users with more information.
- Use the link discovery strategy to automatically find more external data sources.
- Provide a service for users to access to Linked Data.

In the following we outline the detail of each task.

A. Representing Files with URIs

In file systems, files and directories are identified by absolute or relative paths. They are not globally unique and stable. Therefore they have to be converted when they are integrated with the web of data. URI scheme [18] is a means to directly reuse these paths to form URIs. We present the following format to represent file:

<base url>/ <file source> /<file uuid>

Here “file source” is the name of the source file system. If the file system is local, then the file source can be omitted. “File uuid” represents the unique file.

An example for a PDF file “Linked Data - The Story So Far.pdf” is shown as bellow:

http://www.csdb.cn/mysource/09b205be-bf80-4ab9-8ddc-802be95220bb

For each URI, it corresponds to an RDF file, which is represented as “Fig. 1”:

```
<rdf:Description    rdf:about="09b205be-bf80-4ab9-8ddc-
802be95220bb">
   <nfo:filename rdf:datatype=” xsd:string”>
          Linked Data - The Story So Far.pdf
   </nfo:fileName>
   <f2r:fileType>pdf</f2r:fileType>
   <nie:byteSize rdf:datatype=”xsd:integer”>
                 2351032
   </nie:byteSize >
   <nfo:fileLastModified rdf:datatype=” xsd:dateTime”>
          2012-06-20 11:25:05
   </nfo:fileLastModified >
   </nfo:belongsToContainer rdf:resource=”20ffe851-dee3-
4234-9891-0b5d9eedf362” />
</rdf:Description>
```

Figure 1. A Linked Data representation of a PDF file

All RDF/XML files will eventually be stored in a Linked Data database.

```
<rdf:Description    rdf:about="a5486a44-a8d7-4b39-9714-
bccb03595d92">
    <f2r:type>file</f2r:type>
   <nfo:filename rdf:datatype=” xsd:string”>
          Bar-headed_Goose.jpeg
   </nfo:filename >
   <f2r:filetype>jpeg</f2r:filetype>
   <nie:byteSize rdf:datatype=”xsd:integer”>
          114421
   </nie:byteSize >
   <nfo:fileLastModified rdf:datatype=” xsd:dateTime”>
          2012-04-18 10:39:11
   </nfo:fileLastModified>
   </nfo:belongsToContainer          rdf:resource=”5c6f44bf-
78d0-4606-a931-751e724702d0” />
</rdf:Description>
```

Figure 2. Physical Metadata description of a JPEG file

B. *Extracting File Metadata*

On the basis of the VDB-FilePub, file metadata is divided into four sections: physical metadata, built-in content metadata, user-defined content metadata and external sources metadata. We will discuss these four categories with a picture file.

The physical metadata describes the file name, parent path, size, file format, and the last modified time information and so on. File systems provide these descriptions to extract physical metadata via the file system interface. "Fig. 2" shows the physical metadata description for "Bar-headed_Goose.jpeg".

For a directory, we adopt "nfo:belongsToContainer" to describe a file's parent directory, and "nie:hasPart" to describe its sub files and sub folders. For linking feature of a file, we do some special processing. If it is a hard link of a file, we extract the content as the alias with "f2r:alias". If it is a symbolic link, we will ignore the file and not take it as an entity.

Built-in content metadata includes type-related attributes of files, which are often stored in file systems as a part of original files. For example, the EXIF info of a JPEG picture is stored as file metadata information, including aperture, shutter speed, focal length, and date. As is shown in "Fig. 3", it contains some EXIF attributes of the "Bar-headed_Goose.jpeg".

```
<rdf:Description    rdf:about="a5486a44-a8d7-4b39-9714-bccb03595d92">
    <nexif:imageWidth>220</ nexif:imageWidth>
    <nexif:imageLength >172</ nexif:imageLength>
    <nexif:bitsPerSample>8</ nexif:bitsPerSample>
</rdf:Description>
```

Figure 3. Built-in Content Metadata description of a JPEG file

User-defined metadata is manually added via F2R's user-defined interface. In most cases, physical metadata and build-in content metadata are not enough to express useful information for files. We offer an interface for users to describe information. "Fig. 4" shows the user-defined metadata information.

```
<rdf:Description    rdf:about="a5486a44-a8d7-4b39-9714-bccb03595d92">
    <dbpedia:familia> Anatidae </dbpedia:familia >
    <dbpedia:binomial >
            Anser indicus
    </depedia: binomial >
    <f2r:distribution>
            Eurasia , North Africa, Indian subcontinent .etc
    </f2r: distribution >
</rdf:Description>
```

Figure 4. User-defined Content Metadata description of a JPEG file

In addition to the three above-mentioned three kinds of metadata, we propose another metadata to supplement the description of the file information, such as external sources metadata. We extract metadata from the data source and directly present main information to users as a supplement. To achieve this goal, the first step is to find related data sources, and then extract the metadata. In section *C*, we will discuss the detail about detecting the relevant data source. When we get the relevant data source, according to the ontology of the data source description, we can use the SPARQL query language to extract the metadata information. "Fig. 5" shows some metadata extracted from DBpedia.

As is shown in "Fig. 5", for the metadata of external data sources, we have developed a number of rules to extract some key metadata. For example, we extract metadata from DBpedia via some rules, such as "foaf:isPrimaryTopicOf" that used to extract the URI of related resources, and "dbpedia:hasPhotoCollection" that used to detect some set of related photos, and "dbpedia: abstract" for some detail description. What's more, we can specify different rules for different entities to realize personal customization by providing an interface. And some existing vocabulary, such as NEPOMUK File Ontology (NFO) and Dublin Core, can be used to specify the extraction rule.

```
<rdf:Description    rdf:about="a5486a44-a8d7-4b39-9714-bccb03595d92">
    <dbpedia:abstract rdf:datatype="xsd:string">
    Synonyms Eulabeia indica The Bar-headed Goose (Anser indicus) is a goose which breeds in Central Asia in colonies of thousands near mountain lakes and winters in South Asia, as far south as peninsular India. It lays three to eight eggs at a time in a ground nest.
    </dbpedia:abstract>
    <foaf:isPrimaryTopicOf
            rdf:resource="http://en.wikipedia.org/wiki/Bar-headed_Goose "  />
    <dbpedia: hasPhotoCollection
                    rdf:resource="http://www4.wiwiss.fu-berlin.de/flickrwrappr/photos/ Bar-headed_Goose"  />
</rdf:Description>
```

Figure 5. External Sources Metadata of a JPEG file

C. *Detecting Related Links to External Sources*

To link files to external sources, some tools are proposed, which apply various heuristics to detect semantically related resources (e.g. shared identifiers or object similarity [19]). These heuristic tools rely on the data file metadata information, such as file name, file format, file size, etc.

We propose a solution to automatically detect related links from DBpedia via file metadata and SPARQL. DBpedia is the

Linked Data version of Wikipedia, which associates with many other data sets, such as Geonames [20], MusicBrainz [21], the DBLP bibliography [22], WordNet [23], etc. The English version of the DBpedia data set currently describes 3.77 million "things" with 400 million "facts". We define some dependency rules and adopt the file name metadata as dependency. For DBpedia, we choose some vocabulary, such as "owl:sameAs", "dbpedia:wikiPage ExternalLink", to detect links. While we use some main metadata vocabulary ("foaf:depiction", "dbpedia:abstract", "dbpedia:hasPhoto Collection") to extract metadata. "Fig. 6" shows the related link to the external data source.

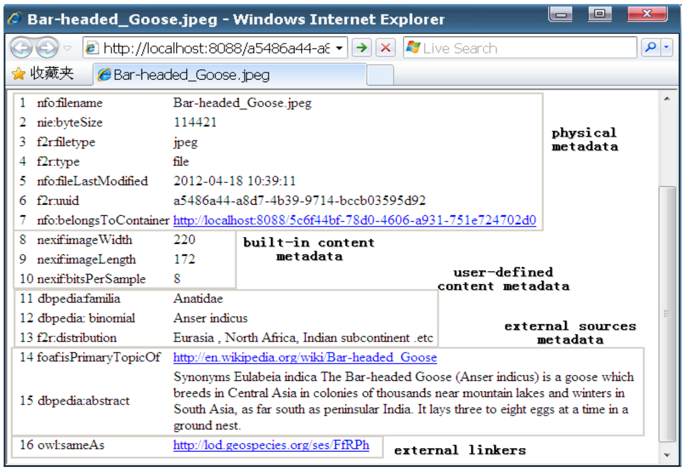
```
<rdf:Description rdf:about="a5486a44-a8d7-4b39-9714-bccb03595d92">
        <owl:sameAs rdf:resource=" http://es.dbpedia.org/resource/Anser_indicus" />
        <owl:sameAs rdf:resource=" http://lod.geospecies.org/ses/FfRPh" />
        <owl:sameAs rdf:resource=" http://rdf.freebase.com/ns/m.01c7_s" />
        <dbpedia: wikiPageExternalLink rdf:resource=" http://juliesmagiclightshow.com/-bar-eaded_goose .php" />
</rdf:Description>
```

Figure 6. External Links to DBpedia, GeoSpecies and Wikipedia

D. Serving Linked Data

Once file systems are represented as RDF resources and link to other related resources on the web, it is necessary to provide web services for users to access the data via URI. In F2R, we offer two kinds of services. One serves for common users. The other one is for professional users.

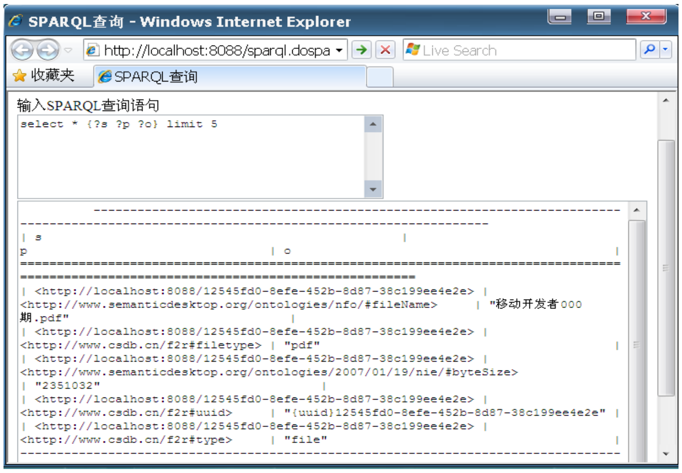
As is shown in "Fig. 7", users can access the file data through browsers. By clicking the link, users can link directly to the related data source.


Bar-headed_Goose.jpeg - Windows Internet Explorer
http://localhost:8088/a5486a44-a8
Live Search
收藏夹
Bar-headed_Goose.jpeg
physical metadata
built-in content metadata
user-defined content metadata
external sources metadata
external linkers

1	nfo:filename	Bar-headed_Goose.jpeg
2	nie:byteSize	114421
3	f2r:filetype	jpeg
4	f2r:type	file
5	nfo:fileLastModified	2012-04-18 10:39:11
6	f2r:uuid	a5486a44-a8d7-4b39-9714-bccb03595d92
7	nfo:belongsToContainer	http://localhost:8088/5c6f44bf-78d0-4606-a931-751e724702d0
8	nexif:imageWidth	220
9	nexif:imageLength	172
10	nexif:bitsPerSample	8
11	dbpedia:familia	Anatidae
12	dbpedia: binomial	Anser indicus
13	f2r:distribution	Eurasia , North Africa, Indian subcontinent .etc
14	foaf:isPrimaryTopicOf	http://en.wikipedia.org/wiki/Bar-headed_Goose
15	dbpedia:abstract	Synonyms Eulabeia indica The Bar-headed Goose (Anser indicus) is a goose which breeds in Central Asia in colonies of thousands near mountain lakes and winters in South Asia, as far south as peninsular India. It lays three to eight eggs at a time in a ground nest.
16	owl:sameAs	http://lod.geospecies.org/ses/FfRPh

Figure 7. Access to metadata of a JPEG file in F2R

For professional users, they can submit their SPARQL queries through the webpage shown in "Fig.8" to search more information.


SPARQL查询 - Windows Internet Explorer
http://localhost:8088/sparql.dospa
Live Search
收藏夹
SPARQL查询
输入SPARQL查询语句

```
select * {?s ?p ?o} limit 5
```

```
          ------------------------------------------------------------------------
----------------------------------------------------------------
| s                                                      |
p                                 | o                                          |
================================================================================
======================================================
| <http://localhost:8088/12545fd0-8efe-452b-8d87-38c199ee4e2e> |
<http://www.semanticdesktop.org/ontologies/nfo/#fileName>      | "移动开发者000
期.pdf"                                     |
| <http://localhost:8088/12545fd0-8efe-452b-8d87-38c199ee4e2e> |
<http://www.csdb.cn/f2r#filetype> | "pdf"                                      |
| <http://localhost:8088/12545fd0-8efe-452b-8d87-38c199ee4e2e> |
<http://www.semanticdesktop.org/ontologies/2007/01/19/nie/#byteSize>
| "2351032"                                 |
| <http://localhost:8088/12545fd0-8efe-452b-8d87-38c199ee4e2e> |
<http://www.csdb.cn/f2r#uuid>     | "{uuid}12545fd0-8efe-452b-8d87-38c199ee4e2e" |
| <http://localhost:8088/12545fd0-8efe-452b-8d87-38c199ee4e2e> |
<http://www.csdb.cn/f2r#type>     | "file"                                     |
--------------------------------------------------------------------------------
```

Figure 8. SPARQL Service in F2R

IV. IMPLEMENTATION

F2R adopts a modular design, and tasks are realized in different modules. As is shown in "Fig. 9", from an architecture point, F2R is designed for two parts: the F2R engine and F2R server. F2R engine is mainly responsible for the release of file data, and F2R server provides services to users, including Linked Data interface and SPARQL service. In the view of function, F2R consists of five modules: file systems connecter, metadata extractor, linker, storage module, and server module. Specific implementations for each module are represented in the following.

File systems connector mainly interacts with the file system. In addition to fetching files from a local file system, connector also implements the interaction with files from FTP and WebDAV. Many files exist in other systems, therefore we provide an interface to integrate with third-party connectors.

Metadata extractor is divided into four parts. For physical metadata, F2R utilizes file systems to extract attributes of files directly. F2R adopts a java framework Aperture to extract built-in content metadata, which supports the common form of documents. Currently registered file types are as many as 51,537, so F2R offers interfaces to allow different extractors to integrate into F2R. For user-defined metadata, users can add attributes through the web interface. While external sources metadata can be extracted in the linker module.

Linker is aim to discover semantically related links. Following certain rules, F2R adopts SPARQL semantic tools and depends on file name to detect external links from DBpedia. These link discovery rules include "owl:sameAs", "rdfs:seeAlse" vocabulary etc. Meanwhile, F2R extracts some metadata by using some other rules, such as "dbpedia: abstract"," foaf:isPrimaryTopicOf".

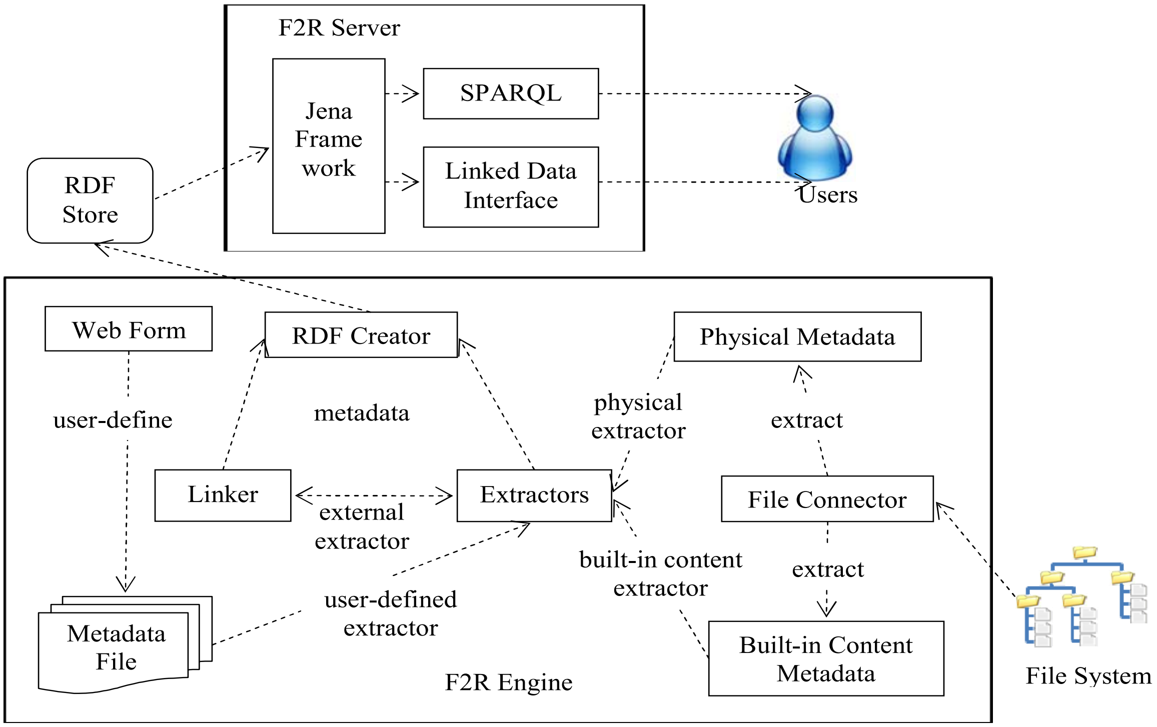


Figure 9. The Architecture of F2R

The storage module that publishes files as Linked Data mainly stores data via RDF. F2R utilizes dom4j to format metadata information of files, and stores them in Jena TDB [24], which has been used to manage RDF data efficiently.

The server module provides SPARQL service and Linked Data interface. In the implementation of this module, F2R integrates Jena Semantic Framework [25] into the module. And we implement the web server via Java NIO and socket to provide web interface.

V. Experiment

After the implementation of F2R, we do some experiments to verify the availability of the system. We apply F2R to publish scientific data as Linked Data.

In the Scientific Database Project [26], more than 500 special data sets were created by 51 institutions. These datasets contain various types of data, such as relational data and file data. A large number of these data, such as images, videos, spatial data, are stored in different file systems.

We choose two datasets from the Scientific Database Project as the experimental data. They are the plant dataset and the animal dataset, which are provided by different organizations of CAS (Chinese Academy of Sciences). More than 100 thousands entities exist in the two datasets. And various types of file data are stored in file systems, such as image files, video files, Text Files, Zip files and other files.

In the two datasets, we apply F2R to publish these files as Linked Data. "Table. Ⅰ" shows the average number of different metadata that are extracted by F2R for an entity.

TABLE I. The Statistical Result Of Linked Data

Dataset	physical metadata	built-in content metadata	external sources metadata	external links
plant dataset	10	7	5	4
animal dataset	10	6	4	3

From "Table. Ⅰ", we can get the average number of different metadata for an entity. The average of physical metadata for an entity is maximum, and the one of external links metadata is minimum. From the result, we find that it is available for F2R to publish file as Linked Data. And it surely extracts more the metadata information by external sources, and detects some useful links from Dbpedia by using semantic tools. In the future, we will apply F2R to serve VDB [27] (Visual Database) and Voovle [28] (Scientific Data Search Engine) to provide more Linked Data service for end users.

VI. Conclusions And Future Work

In this paper, we have presented and discussed F2R, a service that exposes file systems as Linked Data according to

Linked Data principles. On the basis of TripFS and VDB-FilePub, we divide metadata into four categories to enrich metadata information. Meanwhile we propose a solution to automatically detect related links via SPARQL, and we specify some rules to detect more useful metadata from DBpedia. Then we implement F2R with Java and Jena semantic framework. Users can access to Linked Data through browser or SPARQL interface.

As a service that publishes file systems as Linked Data, F2R still needs to be improved. In the future, we will apply F2R in more scientific data set, in order to integrate larger volume of scientific file data to the Linked Data Cloud. And we will try to improve the metadata extraction by some semantic crawl tools, which can crawl many useful properties from the web. We plan to apply some algorithms into the linker module, and integrate them with SPARQL.

ACKNOWLEDGMENT

The work is partially supported by Natural Science Foundation of China under Grant No. 91224006, the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant No. XDA06010202, “Twelfth Five-Year” Plan for Science & Technology Support under Grant No. 2012BAK17B01-1, The Special Project of Informatization of Chinese Academy of Sciences in “the Twelfth Five -Year Plan” under Grant No. XXH12504.

REFERENCES

[1] DROID. Available: http://sourceforge.net/projects/droid/

[2] Metadata Extraction Tool. Available: http://meta-extractor.source forge .net/

[3] Aperture Framework. Available: http://aperture.sourceforge.net

[4] Linked Data, http://linkeddata.org/.

[5] World-Wide Web Consortium: Semantic Web, http://www.w3.org/ standards /semanticweb.

[6] World-Wide Web Consortium: Resource Description Framework, http://www.w3.org/RDF.

[7] C. Bizer, et al., "Linked Data - The Story So Far," presented at the International Journal on Semantic Web and Information Systems(IJSWIS), 2009.

[8] Ghemawat, S., Gobioff, H., and Leung, S. 2003. The Google file system. In Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles (Bolton Landing, NY, USA, October 19 - 22, 2003). SOSP '03. ACM Press, New York, NY, 29-43.

[9] D. Borthakur. The hadoop distributed file system: Architecture and design. Hadoop Project Website, 2007.

[10] MooseFS, http://www.moosefs.org/

[11] Soren Auer, Sebastian Dietzold, Jens Lehmann,Sebastian Hellmann, and David Aumueller. Triplify:Light-weight Linked Data Publication from Relational Databases. In WWW ’09:Proceedings of the 18th international conference on World wide web, pages 621–630, New York, NY, USA, 2009. ACM.

[12] Christian Bizer, Andy Seaborne. D2RQ – Treating Non-RDF Databases as Virtual RDF Graphs. In ISWC’04:Proceedings of the 3rd International Semantic Web Conference.

[13] 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.

[14] Zhihong Shen, Yufang Hou, Jianhui Li. Publishing Distributed Files as Linked Data. FSKD’11.

[15] Soren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. ISWC'07/ASWC'07 Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference, Pages 722-735. 2007

[16] Prud’hommeaux E, Seaborne A. SPARQL query language for RDF. W3C, 2008 01 15, 2008. http://www.w3.org/TR/rdf-sparql-query/

[17] How to Publish Linked Data on theWeb.http://wifo5-03.informatik.uni-mannheim.de/bizer/ pub/LinkedDataTutorial/#intro

[18] T. Berners-Lee, L. Masinter, and M. McCahill. Uniform Resource Locators (URL) (RFC 1738). Network Working Group, 1994.

[19] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Discovering and Maintaining Links on the Web of Data. In Proceedings of the 8th International Semantic Web Conference (ISWC 2009), 2009.

[20] Geonames, http://www.geonames.org/

[21] Aaron Swartz ,MusicBrainz: A Semantic Web Service. Intelligent Systems, IEEE’02. page 76-77.2002

[22] DBLP Bibliography,http://www.informatik.uni-trier.de/~ley/db/

[23] WordNet, http://wordnet.princeton.edu/

[24] Jena TDB，http://jena.apache.org/documentation/tdb/

[25] Brian McBride, Jena: A Semantic Web Toolkit, IEEE Internet Computing, v.6 n.6, p.55-59, November 2002

[26] Data Application Environment for Science Research Project. Available: http://www.csdb.cn

[27] VisualDB - A Management and Publishing Tool for Scientific Data, Available: http://vdb.csdb.cn

[28] Scientific Data Search Engine. Available: http://voovle.csdb.cn