VisualDB: Managing and Publishing Scientific Data on the Web

Zhihong SHEN, Jianhui LI, Chengzan LI, Xing HE, Xianming SU
Scientific Data Center
Computer Network Information Center, CAS
Beijing, China
E-mail: bluejoe@cnic.cn

***Abstract*—Scientific data have become an important part of human knowledge and are playing an important role in knowledge discovery and subsequent scientific research. Some systems and tools are developed to help data owners manage and publish their data online. However, a generic integration framework for relational databases and files is still absent. This paper presents VisualDB as a Web-based management and publishing system for scientific data, which connects relational records and files distributed in multiple distributed repositories, serving them as a whole data source through VdbServer. After reviewing related work, the Linked Entities Model (LEM) is proposed as a general logical model for representing relational records and files. Then problems and corresponding solutions of managing and publishing scientific data are systematically discussed. After introducing the architecture of VisualDB, the paper is concluded with introduction of application and future work of VisualDB.**

Keywords-scientific data; relational database management system; relational record; file sharing; data management; data publishing;Linked Data; Linked Entities Model; VisualDB

I. INTRODUCTION AND MOTIVATION

Scientific data, as the product of scientific research, have become an important part of human knowledge and are playing an important role in knowledge discovery and subsequent scientific research. In the Scientific Database Project [1] supported by CAS (Chinese Academy of Sciences) from 2005 to 2010, more than 500 special databases were established by 51 institutions, and the total volume of data is nearly 160 TB. A large quantity of these data, such as captured images, digital videos, spatial data and other data generated by huge scientific instruments, are organized directly by file systems [2]. Other scientific records such as information of plants, animals, soil, materials, are stored in a DBMS (Database Management System), mostly in a RDBMS (Relational Database Management System) such as Oracle, SQL Server, MySQL, PostgreSQL, and Microsoft Access.

Once scientific data are stored in some format, data owners will need to manage and update them periodically, and share them with other researchers online. However, functions offered by a DBMS or a file system are some limited. The biggest problem is that the view presented by these systems is not the one data owners want to show users. In a RDBMS, what users can see is a number of tables and columns with ugly names and strange types. It is difficult to know what the records mean, and what relationships exist between them. Similarly, in a file system, what users can see is a set of files organized by directories, and it is also difficult to know the meanings of these files. Furthermore, the search capabilities provided by DBMS and file systems are not only limited but also difficult to use. To find a record, users have to learn the SQL language, which is closely related with the physical structure of tables. To find a file, what users can input as search filters are only some native properties such as *filename*, *creation time*, and *size*. Finally, some inherent weaknesses exist. For example, file systems only provide a single hierarchical schema with a tree of directories. Some RDBMS such as Microsoft Access was not born for web-based collaboration.

It is therefore necessary to design an abstract layer upon DBMS and file systems where scientific data are stored. In this paper, VisualDB [3] is presented as a system to manage and publish scientific data on the Web, in which a VdbServer is offered to expose all data as a whole data source. This paper is organized as follows. In section 2 we review related work on data management and publishing systems, especially in science research environment. In section 3, Linked Entities Model (LEM) is proposed as a general logical model for representing relational records and files. The main tasks of and VisualDB's solutions to scientific data management and publishing are systematically discussed in section 4 and section 5. In section 6, we present the VisualDB architecture and implementation. The conclusions are given in section 7.

II. RELATED WORK

General DBMS products provide appropriate database management software such as SQL Server Management Studio for SQL Server, SQL Enterprise Manager for Oracle, and so on. Some third-party tools such as phpMyAdmin[4], DBExplorer[5] are developed for operating DBMS on the Web. However, these tools are highly dependent on the DBMS behind and the interfaces are often complicated, thus keeping users with little expertise on database out. Some independent tools appear to shorten the gap. FileMaker Pro[6] is a powerful, easy-to-use database software that helps users create custom databases for their own unique needs, and securely publish users' databases on the Web in a few clicks. Caspio[7] provides a platform empowering users to quickly create

online databases, web applications, and web forms, all without writing a single line of code.

Also, some tools for file management and publishing appear. FileVista[8] is a web file manager for storing, managing and sharing files online through web browser and users are allowed to upload, download and organize any types of file with an intuitive user interface. Other tools such as PHPfileNavigator, filerun, and idcfilemanager provide similar functions and interfaces. Another example is SRB/iRODS[9], a Data Grid Management System (DGMS) developed by DICE(Data Intensive Computing Environments), which presents users with a single file hierarchy for data distributed across multiple storage systems.

In the science research environment, some management system and tools arise. Laboratory information management system (LIMS) offers a set of key features such as workflow and data tracking support, flexible architecture, and smart data exchange interfaces to support a modern laboratory's operations. Products on such system are Darwin LIMS, Watson LIMS, Nautilus LIMS and Sample Manager, etc. The NuGenesis Scientific Data Management System (SDMS) [10] provides an automated electronic repository that stores and manages all types of scientific data to a centralized database, offering excellent integration with a multitude of research applications.

VisualDB is developed by Scientific Data Center, Computer Network Information Center, CAS (Chinese Academy of Sciences), and now VisualDB 2.0 is available. Compared with other systems mentioned above, one distinguishing feature is that VisualDB aims at integration of scientific data distributed in multiple distributed repositories. The differences between kinds of DBMS and file servers are completely hidden and scientific data stored as either relational records or files are unified and linked in a uniform format. Furthermore, VisualDB publishes scientific data not only in HTML pages but also with Linked Data interfaces, which makes system interoperation and data integration easier. On the same time, strict access controls are enabled in management and publishing processes.

III. LINKED ENTITIES MODEL FOR SCIENTIFIC DATA

As mentioned above, a large part of scientific data exist as files, and at the same time, the other are mostly stored in relational databases. A general logical model supporting the representation for both types is therefore required. VisualDB proposed such a model called LEM, which describes how entities are linked together and organized in a database.

A. Entity, Entity Type and Entity Set

Each record and file (with its metadata) is represented as an entity in LEM. An EntityType represents type of an entity, containing a list of fields, and each field has an Abstract Field Type (AFT). An EntitySet means a set contains a number of entities in the same entity type, which is always built from a query on the entity type filtered by some conditions.

VisualDB uses AFTs to hide the differences of types of physical columns brought by different RDBMS. String, Integer, Double, Date, and others such as File/Files (and its sub-types Image/Images, Movie/Movies, etc.), GPSLocation, ChemicalStructuralFormula, which are common for scientific data, are all AFT.

An example of the entity type is *Plant*. It contains fields such as *familyName*, *genusName*, *soilPH*, *distribution*, in which the field *soilPH* has an Integer value, while the others have String values.

B. Links between Records and Files

As entities are linked, One-To-Many, Many-To-Many and Many-To-One relationships are allowed to be descried between two entity types. When a relationship is defined, two bridge fields will be derived automatically for the entity types in both sides. Value of a BridgeField is either a reference or a collection. A reference means a pointer to another entity, and a collection consists of a set of references.

Links between entities includes those between relational records and between records and files. Take a record about a ginkgo tree as an example, it may have a field named *Photos*, which contains a series of photos about the ginkgo tree, and all photos are file entities in JPEG format.

As an example of LEM, Fig. 1 shows a cloud diagram of a model in a material database, in which 22 entities (shown as circles) and 20 relationships (shown as lines) are descried.

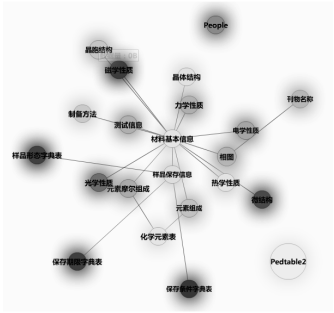

Figure 1. A cloud diagram of LEM for the material database
Source: http://resstat.csdb.cn/quality/qualitydetail-cn.csdb.matsci_cn.csdb.matsci.InorgNonmetMat.html

C. Links between Entities and External Services

The value of scientific data is enlarged when they are linked with other computation models. For example, two maps are submitted to a web service called *MapAlignment*, or an EntitySet contains a series of PH indicators are submitted to a trend analysis system. In this way, LEM leads users who are consuming scientific data on the Web, to relevant services to discover more knowledge easily.

A feature similar to Windows Shell OpenWithList is introduced into LEM. Some Entity-level commands (such as show, download, insert, update and delete) and EntitySet-level commands (such as list, query, download,

import and export) can be configured in an OpenWithList of an EntityType.

In VisualDB, a corresponding handler should be designated for each command. Once VisualDB receives a command, it will package all required data in certain format and pass them to the corresponding handler, which generally refers a local or remote web page or web services module.

D. Mapping Records and Files to Entities

Unlike LEM, the physical data model (PDM) describes how a DBMS or a file system organizes and stores data. A PDM always contains multiple repositories, and each Repository corresponds to a database connection (so-called SqlRepository) or a root path in a file system (so-called FileRepository). In VisualDB, a SqlRepository may refer to a database in any type of DBMS product (Oracle, or MySQL, etc.), and a FileRepository may be a directory on local machine or a remote server, connected through FTP or SAMBA protocol.

Each record in a relational database is mapped into one entity. As the first step, the table which contains the record will be mapped into an entity type, with all columns being mapped into fields. Readable names are recommended to be taken for entity types and fields because the original names of tables and columns are often short and obscure. Next, a DBMS-independent access interface should be provided to load records from different databases. Once a record is loaded, conversion from columns to fields following defined mapping rules will be performed, in which each field will be bound to one or multiple columns in a table. Finally, linked records will be loaded and converted into entities, and a reference or a collection containing these entities will be generated as a bridge field of the parent entity.

Similarly, in a file system, each file with its metadata is mapped into one entity. We divide metadata of files into three kinds: physical metadata (such as: name, path, file length, last modified time), built-in content metadata (metadata which can be extracted from the file itself) and user-defined content metadata. The mixed metadata format forms the structure of an entity type's attributes. First, physical metadata and built-in content metadata will be automatically extracted through multiple extractors. Second, an interface is provided for users to enter user-defined content metadata manually. Once all metadata are ready, a conversion from the merged metadata to fields of a file entity is performed. Furthermore, unlike relational records, content of the file, often in binary format, is also exposed.

E. Classification and Navigation

Classification helps users easier to understand the subject, format or other extra information of an entity, depending on the purpose of each classification system. LEM allows administrators to define multiple classification systems (always defined as a tree), such as the subject classification and the nation classification, and then specify which among them is available for each EntityType. When a data editor input a new entity, he/she can also specify the classification that the entity belongs to. This process, sometimes is also called "tagging", enriches semantic meanings on original scientific data.

Navigation helps data consumers understand how data are organized in a data set. LEM allows administrators to define multiple sets of navigation systems. When browsing within a data set, users can select a node in a navigation tree to see which entities it contains. Each node in the navigation tree is bound to a defined entity set or a dynamic entity set built on the fly following by defined generation rules.

IV. Managing Scientific Data

Managing scientific data includes managing records and links, managing files and metadata, and providing fine-grained access control in the management process.

A. Managing Records and Links

Management of records involves insertion, deletion and modification on records. VisualDB offers a range of Web-based input controls, including DateTimePicker, WYSWYG HTML Editor, select box, file uploader, and so on. A more obvious example is an input control for chemical structure formulas.

In addition, since many data editors are accustomed to Microsoft Excel to input data, an Excel-like interface is also provided in VisualDB.

When entering values of fields for a given record, users can also input the value of its bridge fields. In this way, links between the records are generated automatically. For example, when inputting basic information of the carbohydrate in a material database, it is allowed to enter the information of its elements, which may be defined as a bridge field named *elements*. In the input area, users can add new entities such as carbon, hydrogen and oxygen (which may have another entity type named *ChemicalElement*), or simply choose them from a candidate list. Once users complete the input form, the carbohydrate and its elements are automatically linked.

Since importing and exporting data for an EntitySet are common work for data administrators. CSV, Excel or XML format is available in VisualDB. Fig. 2 shows a screenshot of the management interface for relational records in VisualDB.

Figure 2. Management interface for relational records in VisualDB

B. Managing Files and Metadata

A Web-based file manager is somewhat similar to the Windows Explorer, it enables users to upload, delete, move, and copy a file. For files in common formats (such as TXT format), VisualDB also provides an online editor, allowing users to update the document without downloading it first of all.

Since all operations are performed remotely, a fast and easy-to-use upload interface for batch files is provided. Fig. 3 shows a screenshot of the management interface for files in VisualDB.

Figure 3. Management interface for files in VisualDB

Unlike traditional file management system, VisualDB will show an "Add/Update Metadata" option in the context menu when users select a file. Once selected, a pop-up window will be displayed, in which users can input user-defined content metadata for the file.

C. Access Control

Based on RBAC (Role Based Access Control), a fine-grained access control framework is provided in VisualDB, in which the operated object is not limited to an entity, but to a certain field of an entity, which is particularly important in the process of scientific data sharing. Fields of an entity may be divided into various facets, and users in certain roles can only view some limited facets (such as basic information of a nano-material), but are restricted to access other information (such as the information projects in which the nano-material is applied).

Who has the right to access the information of an entity? This depends on the data owner's wishes. A data owner can be the editor of one entity, or someone specified by system administrators. A typical use case of VisualDB is that a number of biological data owners, who are researchers in various groups of different laboratories in distributed sub-centers, are organized to manage and share their own data in a virtual project team.

V. PUBLISHING SCIENTIFIC DATA

Publishing scientific data covers publishing the data within a Web-based browsing system, and publishing machine-understandable interfaces for programs. In both processes, VisualDB provides plenty of options for customization.

A. HTML Representations for Scientific Data

HTML representations for scientific data, includes pages listing entities in an EntitySet and pages displaying detail information of an entity. The big challenge is how to display different types of data in a right way. For each field, VisualDB provides a number of view controls to choose, such as: text labels, HTML fragment viewers, picture viewers, video players, map views and so on.

When one entity is published, all links to other entities will also be published. For example, as one of the fields of a ginkgo tree, a bridge field *photos* may be displayed within all photos and their metadata.

In addition to providing description information for an entity, for a file, however, the content of the file also need to be offered. Content of the file is often packaged into a binary stream. By parsing HTTP request header, VisualDB enables clients to retrieve the entire file or a part of a file in compressed or uncompressed format.

For a given entity type, URLs of listing one entity set, showing one entity's information in detail and downloading the content of an entity is shown as below:

<base url>/page/list(<entity set name>)

<base url>/page/view(<entity type name>/<entity id>)

<base url>/page/download(<identity type name>/<entity id>)

For a record, the entity id corresponds to the primary key value. However, for a file, a unique value should be generated as the entity id, such as:

http://www.plant.csdb.cn/page/view(photo/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. Linked Data Interfaces for Scientific Data

HTML Representations is friendly to human, but understandable for machines such as a data acquisition program. Linked Data [11] is a lightweight solution to publish machine-understandable information on the Web, and more and more people are encouraged to publish their datasets as Linked Data. Frameworks such as D2RQ Platform [12], Triplify [13], TripFS [14] are developed to expose relational databases and file systems as Linked Data.

VisualDB provides Linked Data interfaces based on LEM. For each entity, it is assigned a URI:

<base url>/wod/resource/<entity type name>/<entity id>

An example of such URI is shown as below:

http://www.plant.csdb.cn/wod/resource/plant/1

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

http://www.plant.csdb.cn/wod/data/plant/1

http://www.plant.csdb.cn/wod/html/plant/1

The later URL will simply redirect to:

http://www.plant.csdb.cn/page/view(plant/1)

Once VisualDB receives an HTTP request with Accept: application/rdf+xml in the header, it will find the corresponding entity, wraps its attributes into a RDF model and output response in RDF/XML. All links to other entities will also be represented as RDF Links. Fig. 4 shows a RDF representation for the description of red soil

in a soil database, within links to multiple physical and chemical fields.

Figure 4. RDF representation for the description of red soil
Source:
http://www.soil.csdb.cn/wod/data/cn.csdb.soil.soiltype/soilType/20422

C. Customization

When using a publishing system, users may have requirements on customization, such as how many entities should be listed in per page, in which style a series of images shall be displayed.

VisualDB offers a range of options for customization. For a published page, banners, headers, footers, navigation menus and background images within can be changed as users wish. A number of styles and frameworks are also provided to choose. Furthermore, a number of widgets such as a RSS reader, a pie chart, and a picture slide, are provided for drag and drop page design. Finally, the most attentive, each function component provides configurable options.

VisualDB uses a number of tags to encapsulate all visible /invisible components, which make it easier to enable customizing a Web-based component with various parameters. As an example, codes of a parameterized DataGrid component are shown as below:

```
<sdc:ctrl id="table" type="DataGrid">
    <sdc:param name="query" value="$query"/>
    <sdc:param name="bgcolor" value="#cccccc"/>
    <sdc:param name="sizePerPage" value="30"/>
    <sdc:param name="columns" value="$columnIDs"/>
    <sdc:param name="showTitle" value="false"/>
</sdc:ctrl>
```

When publishing HTML pages, the Web Publisher will merge parameters and templates within components to generate customized pages.

VI. Implementation

VisualDB has been being developed continuously since 2001. Many methods discussed above have been realized in the last release version VisualDB 2.0. However, some new features are planned to be introduced in the coming version VisualDB2011. The software architecture of VisualDB2011 is shown as Fig. 5.

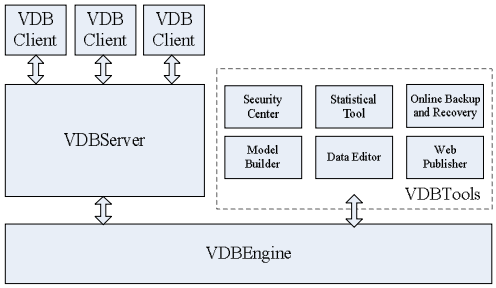

Figure 5. Architecture of VisualDB

VdbEngine enables global settings configuration, stores and loads LEM modules, maintains a security framework, and connects to all user defined repositories in different locations through different protocols. Within a typelib is provided to support field types of scientific data be managed in a pluggable way. A cache is also embedded to ensure high performance of data access.

Based on VdbEngine, VdbServer exposes relational records and files distributed in multiple repositories as a whole data source. When connecting to a VdbServer, a granted VDB client can remotely update and access data in VisualDB. Hessian binary web service protocol [15] is chosen for its high performance.

VdbTools includes Model Builder, Data Editor, Web Publisher, Statistical Tool, Security Center and Online Backup and Recovery Tool.

The Model Builder provides an integrated environment for database modelers, in which users design LEM and define mapping rules from PDM to LEM. Data Editor provides data management functions, in which a SQLExplorer and a FileExplorer are provided to help user manage relational records and files. Web Publisher helps data administrators publish web pages with customization.

In Security Center, system administrators manage users, roles, and assign privileges associated with entities for them. Statistical Tool does some statistics on scientific data, outputs tables and charts about the number of records, files and entities, total size of all entities in different types. Online Backup and Recovery Tool, connecting to BOOSS (Blue Ocean Online Storage Service) [16] in Scientific Data Center, enables online data backup and recovery between VisualDB and BOOSS.

VII. Conclusion

VisualDB is originally developed to manage and publish relational databases and files distributed in different research institutions in the Scientific Database Project. Up to now, we have deployed VisualDB in 47 research institutes to help them build management and sharing system for chemical data, spatial data, material data, ecological data, and so on.

Furthermore, we developed a semantic search engine voolve [17], which connects to all registered VdbServers to fetch all models and data, and provides an integrated search interface to end users.

Another completed work is resource statistics on scientific data. 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. Fig. 5 shows a statistical table about the sizes of different scientific datasets, with each line displaying the name, total size, SOF(size of all files), SOR(size of all records), NOR(number of records) and NOF(number of files) of each data set.

科学数据库节点名称	总数据量	文件型数据量	关系型数据量	记录数	文件数
化学主题数据库	5.69GB	2.57GB	3.12GB	2646179	506514
化合物参考数据库	24.94GB	5.53MB	24.94GB	128169405	21
天文科学数据主题库	16.64TB	16.15TB	501.96GB	337391325	12654871
材料科学主题数据库	6.52GB	4.92GB	1.6GB	140368	11509
光学技术专业数据库	1.51GB	1.15GB	369.58MB	92537	7490
南海海洋科学数据库	768.06GB	768.06GB	428.19KB	4329	1145837
武汉植物园引种保育数据库	18.45GB	18.4GB	49.23MB	208182	39909
西双版纳热带植物园植物引种与保育数据库	34.34GB	34.31GB	32.72MB	167199	20448
中国动物主题数据库	7.05GB	6.97GB	82.89MB	240842	23399
聚实数据库	22.21GB	22.21GB	375.85KB	3671	16655
中国微生物与病毒主题数据库	1.41TB	5.53MB	1.41TB	409183851	21
重要病毒性病原检测专业数据库	9.44MB	5.53MB	3.91MB	1170	21
疾病相关基因整合信息数据库	411.76MB	411.18MB	599.11KB	114	8918
转录组与蛋白质组整合数据库	-	-	-	-	-
系统生物学中多组学综合数据库	276.65GB	266.75GB	9.9GB	99668000	134615

Figure 6. A statistical table about the sizes of different scientific datasets
Source: http://resstat.csdb.cn/quality/qualitylist.html

In the future, we plan to improve VisualDB, and linking more VdbServers, through which a semantic web of scientific data can be constructed with each piece of data accessible and linked. To meet other requirements of data sharing, we also plan to develop an online mash-up platform to help domain researcher to build virtual subject database on demand.

Another plan is DBSpace, a hosting edition of VisualDB in BOOSS environment. In DBSpace, spaces for database and files will be allocated on demand. This plan will encourage more students, individual scientists and research teams to create their own data sets, modeling online, managing and publishing their data on the Web.

ACKNOWLEDGMENT

The work is supported by 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] A Management and Publishing System for Scientific Data. Available: http://vdb.csdb.cn/

[4] phpMyAdmin . Available: http://www.phpmyadmin.net/home_page/index.php

[5] DBExplorer. Available: http://code.google.com/p/jdbexplorer/

[6] Database Application By FileMaker | FileMaker Pro 11. Available: http://www.filemaker.com/products/filemaker-pro/

[7] Caspio - Online Databases Made Easy. Available: http://www.caspio.com/

[8] Web File Manager – FileVista. Available:http://www.gleamtech.com/products/filevista/web-file-manager

[9] The DICE Storage Resource Broker . Available: http://www.sdsc.edu/srb/index.php/Main_Page

[10] Waters NuGenesis Informatics: SDMS, electronic lab notebooks. Available:http://www.nugenesis.com/

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

[12] The D2RQ Platform v0.7 - Treating Non-RDF Relational Databases as Virtual RDF Graphs. Available: http://www4.wiwiss.fu-berlin.de/bizer/d2rq/spec/

[13] Triplify, Available: http://triplify.org/

[14] 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.

[15] Hessian Binary Web Service Protocol . Available: http://hessian.caucho.com/

[16] Blue Ocean Online Storage Service. Available: http://onlinestore.csdb.cn

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