Journal of Digital Imaging

A Grid-Based Model for Integration of Distributed Medical Databases

Yongxing Luo, Lijun Jiang , and Tian-ge Zhuang

Grid has emerged recently as an integration infrastructure for sharing and coordinated use of diverse resources in dynamic, distributed environment. In this paper, we present a prototype system for integration of heterogeneous medical databases based on Grid technology, which can provide a uniform access interface and efficient query mechanism to different medical databases. After presenting the architecture of the prototype system that employs corresponding Grid services and middleware technologies, we make an analysis on its basic functional components including OGSA-DAI, metadata model, transaction management, and query processing in detail, which cooperate with each other to enable uniform accessing and seamless integration of the underlying heterogeneous medical databases. Then, we test effectiveness and performance of the system through a query instance, analyze the experiment result, and make a discussion on some issues relating to practical medical applications. Although the prototype system has been carried out and tested in a simulated hospital information environment at present, the underlying principles are applicable to practical applications.

KEY WORDS: Grid, medical database, OGSA-DAI, metadata

INTRODUCTION

While many modern medical applications (such as telemedicine, epidemiology, etc.) involve accessing heterogeneous databases in a geographically distributed environment, there is an arising demand on integration of different data resources across different information systems or administrative domains to provide a uniform access interface for users. So, effective integration of heterogeneous medical databases has been studied as one of the most challenging fields. Much effort has been made in the research on integration of heterogeneous data resources, including the early multi-database systems (e.g., Multibase),[1] mediator systems (e.g., Garlic).[2] Projects like EnsEMBL and TAMBIS put emphasis on data integration in biomedical domain.[3,4]

Emerging as a new paradigm for distributed computing, Grid is defined as the "flexible, secure, coordinated resource sharing technology among dynamic collections of individuals, institutions, and resources".[5] As a potential solution for managing and collaborating distributed resources, Grid addresses the issue of heterogeneity by developing common interfaces for access and integration of diverse data sources. These interfaces can be used to represent an abstract view of data sources, which can permit homogeneous access to heterogeneous medical databases. So, Grid can provide medical applications with architecture for easy and transparent access to distributed heterogeneous resources across different organizations and administrative domains.

In this paper, we put emphasis on medical database integration by employing Grid technology. The rest of the paper is structured as follows. The Background section gives a brief introduction to the background and some related projects, especially OGSA-DAI. Then, in the Methods section, we present a prototype model for integration of heterogeneous medical data resources based on Grid technology with OGSA-DAI, and then, we

From the Biomedical Engineering Department, Shanghai Jiao Tong University, Shanghai 200240, People's Republic of China.

Correspondence to: Yongxing Luo, Biomedical Engineering Department, Shanghai Jiao Tong University, Shanghai 200240, People's Republic of China; e-mail: haiyangcook@sjtu.edu.cn

Online publication 3 January 2008

doi: 10.1007/s10278-007-9077-9

make an analysis on its functional components. In the Results section, we test effectiveness and performance of the system through a query instance in a simulated medical information environment. Then, we discuss some issues relating to practical applications in the Discussions section. Finally, some conclusions are drawn on the significance of the current achievements In the Conclusion section.

BACKGROUND

Related Works

Some Grid-related data access and integration solutions have been provided in some projects concerned. ISPIDER project is aimed at providing an environment for constructing and executing analyses over proteomic data and a library of proteomics-aware components that can act as building blocks for such analyses.[6] AutoMed wrappers are used to extract sources' metadata and accomplish schema mapping and integration.[7] Sybase Avaki for Data Grids is a complete packaged data Grid solution that provides a seamless,[8] transparent, and single unified view of data for compute Grid applications. OGSA-DQP is a service-based distributed query processor on the Grid,[9] which aims to exploit the service-oriented middleware provided by OGSA-DAI and OGSA reference implementation, Globus Toolkit (GT), by plugging into the port types defined by the constituent services of those frameworks. CaGrid is the underlying service-oriented infrastructure for caBIG which is to develop applications and architecture that connects together data,[10] tools, scientists, and organizations in an open federated environment. CaGrid defines two types of "Grid services" that can be registered as nodes on the Grid: Data Services and Analytical Services, and provides a standard infrastructure for bioinformaticians to advertise their services through common metadata defined in UML domain information model.

OGSA-DAI

Open Grid Service Architecture-Data Access and Integration (OGSA-DAI) is a project that develops middleware to assist with access and integration of data from separate sources via the Grid.[11] The project was funded by the UK e-Science core program and is working closely with the Global Grid Forum Data Access and Integration Services (GGF DAIS) work group.[12] The goal of OGSA-DAI is to provide a uniform service interface for data access and integration to databases exposed to the Grid, hiding differences such as database driver technology, data formatting techniques and delivery mechanisms. Therefore, OGSA-DAI services can provide the basic operations that can be used by higher level services to offer greater functionality, such as data federation and distributed queries. The OGSA-DAI architecture also encourages the design of efficient applications by supporting for grouping multiple requests on an OGSA-DAI service into a single message sent to a service. This reduces latency by increasing the granularity of interactions and reducing both the number of messages exchanged and the quantity of data transferred.

Grid Data Service (GDS) is the primary OGSA-DAI service that supports the interaction with a single database.[13] It manages an authenticated collection of requests to a resource and stores the status and results of requests. An instance of a GDS is created by the Grid Data Service Factory (GDSF). GDSF can be located by a DAI Service Group Registry (DAISGR), which is used to publish data resource metadata and capabilities. At present, data resources that can be exposed via OGSA-DAI include: relational databases such as MySQL, SQL Server, IBM DB2, Oracle; XML databases such as eXist, Xindice; files and directories in format such as OMIM, EMBL.

The extensibility and scope of OGSA-DAI made it an ideal tool when researchers began looking at better ways to manage the massive amounts of information created by some scientific research programs. Some projects have employed OGSA-DAI successfully as their core functional component of data integration and management, such as E-Diamond,[14] ConvertGrid,[15] InteliGrid,[16] and LEAD.[17]

METHODS

Overview of the Model

As Grid technology deals with sharing distributed heterogeneous resources without compromising local administration and OGSA-DAI provides functions to federate heterogeneous data resources, we designed here a new system model for integrating heterogeneous medical databases based on Grid technology with OGSA-DAI. The goal of the model is to provide a uniform query and browse mechanism for multiple distributed medical databases, shield their differences in data schema or location, and provide a new solution for the research on integration of medical databases.

The architecture of the model is shown as Figure 1. Data resources supported by OGSA-DAI include traditional relational databases (e.g., Oracle 9i, MySQL, and SQL Server) and file data, which are familiar data resources in modern hospitals. Then, we adopted OGSA-DAI as the core functional component of the model to assist in access and integration of separated databases via the Grid. Through OGSA-DAI middleware, we can enable users to access and process various medical databases through secure and transparent Grid system and implement the Grid interoperability characteristics, while hiding the underground heterogeneity and dynamics of those databases.

On the top of OGSA-DAI, some additional functional components should be developed to deal with query processing, metadata management, and transaction management issues in Grid environment, besides some basic services provided by OGSA-DAI. We will make an analysis on each of them in the next section. Finally, through some

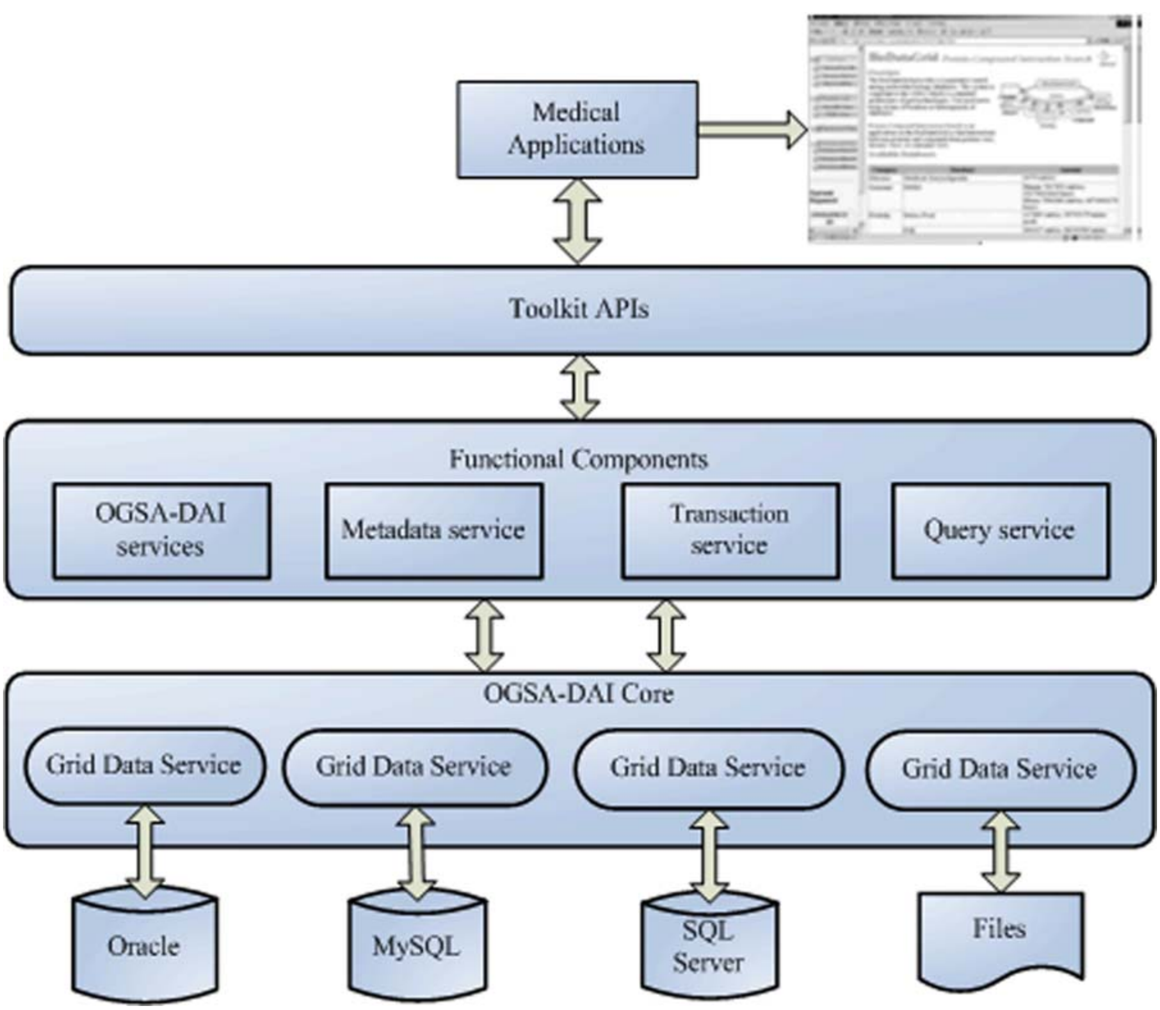

Fig 1. Architecture of the model.

client toolkit APIs, users can implement many medical applications based on Grid technology.

Analysis of Functional Components in the Model

OGSA-DAI

As we have mentioned above, OGSA-DAI is a middleware product which extends the Globus Toolkits with a means of exposing database interfaces to the Grid at server sites.[18] An interface is exposed as a set of GDS which is the point of contact for clients who wish to access, query, or update a data resource. As a specialization of Grid service, GDS is exposed to the Grid through its deployment in a Grid container running on a target host. An instance of GDS exposes zero or more data service resources and provides a number of operations, which allow information about the service to be retrieved. Applications can use the core GDS components directly to access individual data stores or can use a distributed query processor to coordinate access to multiple database services.

An OGSA-DAI client communicates with the OGSA-DAI wrapped data resources via XML documents. The document-based interaction model ensures platform independence on the client side. This design encourages coarse–grain interactions, which help to minimize round-trip message exchanges and unnecessary data movement. Provided with a Java client toolkit, which removes the need to construct or parse XML documents, clients can express requests by building objects representing OGSA-DAI activities. An activity is mapped to a java class that is responsible for processing the content of that activity. OGSA-DAI provides various activities to support common operations such as SQL and XML query, flat file access, data transformation, and asynchronous delivery to third parties. The activity framework is extensible; thus, users may define their own activities for specific operations.

Metadata Model

Metadata service is a key component of the system model, it is responsible for collecting and managing the metadata, whose purpose is to provide information about involved computing resources such as data repositories, machines, networks, programs, and data schema, etc. Therefore, metadata can represent a key element to effective resource discovery and utilization in the Grid environment.

OGSA-DAI services provide metadata about the DBMS that is being exposed to the Grid. For relational databases, the database schema may be extracted from the service, which may be helpful to higher level services such as distributed query processing. The metadata may be provided statically or dynamically. The static metadata model is extensible so that communities that employ OGSA-DAI to access databases within a Grid context can provide corresponding metadata for the databases they expose to the Grid.

AS metadata plays a critical role in Grid databases integration, a layered metadata model is put forward based on Grid technology with OGSA-DAI, which comprises information of data schema, data properties, and relations between data objects. As shown in Figure 2, the metadata model we designed includes three components:

Application Metadata. Based on corresponding OGSA-DAI services that can provide metadata about the DBMS, application metadata is defined to construct relationships between data object attributes that may be stored in different DBMSs across domains; thus, we can fill the semantic gap between heterogeneous data resources that is a major obstacle in the integration of distributed databases.

Here we adopted an object-connection model to construct application metadata. Firstly, we extract some intrinsic common properties of the same kind of data objects, such as name, sex, age of a patient that are almost invariable during his medical care, to construct object attributes to support identification. Note that an attribute may have different names in different databases, so we should establish effective mapping between object attributes we defined and corresponding attributes of each database. Through this mapping relation, we can ensure a proper extraction of all corresponding properties for constructing valid object attributes. In a distributed environment, a data object (e.g., a patient) may have various data information stored in different DBMSs across different domains, while they are closely associated with each other in semantics. So, the next step is the definition of object connection attribute that is used for establishing the relationship between data objects and connect them with each other in

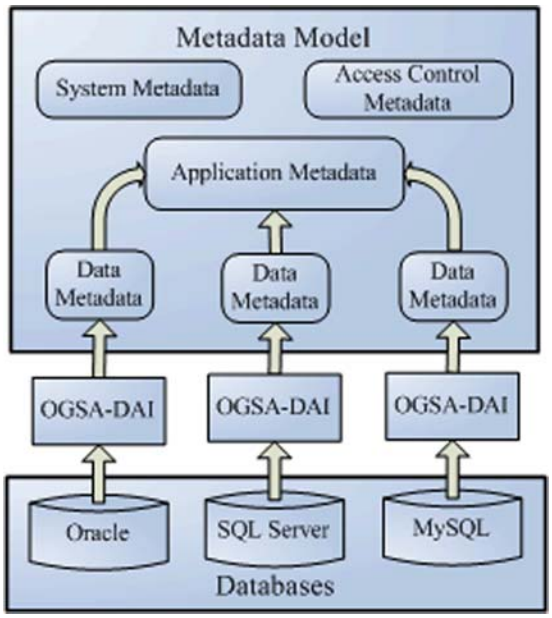

Fig 2. Architecture of the metadata model.

semantics; thus, we can implement seamless integration of distributed databases. For example, we can add an attribute named "location" to indicate hospitals or departments a patient went for medical care. By querying this connection attribute, all corresponding datasets, related in semantics, can be connected to provide integrated medical information about the patient. The connections can be built on tables or records, which will simplify query processing and improve the query efficiency.

As the de-facto standard for communication and a widely used format for storage of digital medical image, DICOM files indeed contain some acquisition-related metadata in the image header. However, this in-file metadata is often incomplete and not practical for data search and query. Therefore, we extract the in-file metadata and store it in a relational database with some additional metadata on attributes of these DICOM files such as file name, image type, and access path, while maintaining these DICOM files still stored in file systems. Thus, we can make use of the advantage of relational databases in managing structured data and the ability of file systems in managing unstructured data to provide an efficient way to access image data.

Access Control Metadata. It stores information about authentication and access control, including information of users logged in the system, such as IDs, password, user role, etc., to validate users' identification and restrict their operations on data. Users can be classified into anonymous users and registered users. An anonymous user receives default access level assigned by system administrator and enjoys restricted authorities on Grid nodes and data resources. For example, an anonymous user can only browse current status information of Grid node and query, browse, and download dataset he needs, but he cannot modify it. As for a registered user, besides those rights possessed by anonymous users, it also has the right assigned by the administrator to access and modify special data set stored in the system, share its data set and computing resources with other users, modify node information of its own Grid node, etc.

Users may enjoy the consistent authorization and authentication mechanism supported by Grid Security Infrastructure (GSI) which can provide "single sign-on" for users of Grid. That means once they log in, they are permitted to access multiple database resources, avoiding the complexity of identification authenticating by each database.

System Metadata. It stores information about Grid nodes, such as node name, IP address, location, storage capacity, network statement, etc. which is used to describe the status of the system, trace the origin of data, and optimize data access.

As security is a very crucial and complicated issue in distributed database management system, especially in Grid environment, it is important to establish the mechanism of identity authentication and access authorization so that the user's request can be executed with the appropriate privileges. The layered metadata model we designed can avoid direct access to data resources by users; any user who wants to access the system will first drive a domain access server (DAS) to query metadata catalog that contains corresponding metadata information. According to the logging information, different users can be mapped into different roles that are granted with different access permission levels based on the corresponding user metadata. After determining the appropriate access permissions, the request is either rejected or accepted.

Based on current system security strategy, we will implement an access control and authorization mechanism with fine granularity to ensure the access security in a more strict way to prevent any false or malicious manipulations on each item in databases.

Transaction Management

Transaction is a fundamental concept for database operations, which also is important for cooperation in a distributed environment. As a new paradigm for distributed computing, the Grid requires additional and more flexible mechanisms for controlling requests and outcomes than are typically offered by traditional distributed database transaction models. To adapt the dynamic Grid, transaction mechanism with different strictness and granularity should be implemented. As our prototype system mainly deals with accessing heterogeneous databases in Grid environment through a uniform interface, we designed a transaction mapping management component based on request that will be used mainly to transform global transaction submitted by the client into local transactions on various data resources.

As shown in Figure 3, when a global transaction is submitted to the transaction mapping manager, it will access a global metadata catalog which can implement address mapping through accessing corresponding metadata information stored in it to get the physical address of the domain access server (DAS). DAS can implement access control of data resources in a logical domain and also collaborate with other DASs to carry out data accessing across domains when the system is expanded to multiple domains. Then the DAS will access its domain metadata information catalog, which contains corresponding metadata we defined. With the help of metadata service that could eliminate the semantic heterogeneity between different databases, the global transaction is mapped into several local transactions on the underlying databases. Transactions with only one local transaction can be submitted immediately, as they will not pose any threat to the consistency of the data. If multiple local transactions are required by the global transaction, the Grid infrastructure appends a unique timestamp with every local transaction before submitting it to the corresponding database. Local transactions are then submitted to the database's queue of the local scheduler. The local transactions are executed strictly according to the timestamps allocated by the Grid. After every node finishes execution of its part of the transaction, the node where the transaction was originally submitted is informed to get the result of each transaction through corresponding DAS. Some cooperation work should be done to join the results into a global result returning to the client node.

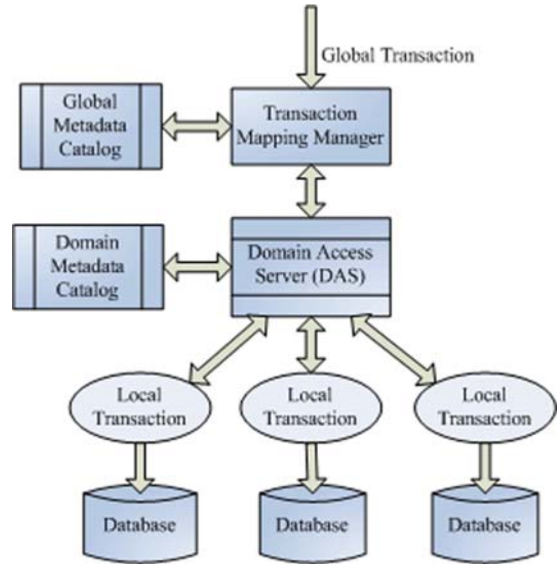

Fig 3. Transaction management model in the system.

Query Processing

Query for a single database could be accomplished by the basic activities of statement activities provided in OGSA-DAI, while distributed query processing need the coordinated work of the functional components mentioned above. Query processing component works like a coordinator during this process, the global query submitted to query component is decomposed into several sub-queries according to the integrated schema stored in metadata catalog. Using application metadata we defined above could eliminate the semantic heterogeneity between different databases by constructing integrated schema. Query service then executes the sub-queries by coordinating some corresponding data services.

As shown in Figure 4, users are not permitted to query the databases directly using SQL; instead, queries are sent as an XML document to the query service for necessary processing to meet requirements of OGSA environment. The query service can ascertain the access rights of an individual user by querying the user metadata, which contains information about users logged in the system. Having determined the appropriate access permissions, the query is either rejected or translated into appropriate SQL. The query service also ensures

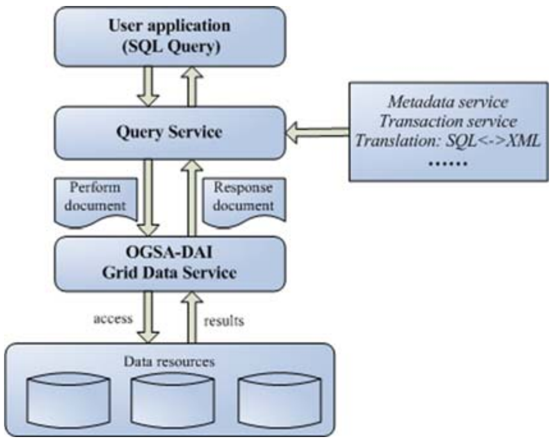

Fig 4. Query processing in the system.

that the access attempts are logged correctly by inserting the user details into the logs; this allows several users to be mapped to a single database account while maintaining traceability.

The SQL query is then placed inside a perform document in XML format and passed to the OGSA-DAI data service for accessing underlying heterogeneous databases. Finally, the result of the query is then returned by the OGSA-DAI data service as a result document. Adopting the format of XML document makes it relatively straightforward to transform the results into any format that is required using XSL transformations.

RESULTS

The experiments described in this section show the integration of heterogeneous databases in a seamless way based on Grid technology. The experiments have been conducted on the heterogeneous collection of machines with installation of different DBMS. All machines are connected with each other through a 100-m local area network (LAN); the basic information of the simulated environment are described in Table 1.

For the sake of privacy, all the medical data information we used is simulated including basic personal data information, basic health data information, and radiology data information. They are stored, respectively, in different database management systems located at three Grid nodes. Note that radiology data information includes only metadata extracted from DICOM files stored in the file system and corresponding information about accessing DICOM images.

To test effectiveness of the model system, we design a query instance to retrieve all the medical records of a patient given his ID. We submit the query plan through a client computer that is in the same local area network. Figure 5 shows the query result of the instance. To test performance of the model system, we implement the same query instance on these three DBMSs, respectively, and compare their response time with that of the joint query on the model system. As shown in Table 2, the performance of our model system is comparatively stable as the size of the table increases.

It can be seen from the results that our system integrates distributed, heterogeneous databases in a seamless way, so that medical data information can be highly shared and accessed very conveniently. Based on Grid technology, each of the functional components we designed enable a quick locating of target data by using semantic connection between distributed data objects; thus, the model system can hide the heterogeneities of those medical databases and provide an efficient way to query medical data including text and image with a comparatively stable performance.

DISCUSSIONS

As a prototype model, the system is currently implemented and tested in a simulated information

Table 1. Simulated Environment of Hospital Information System (100 m Local Area Network)

Machine			Databases		Grid Software	
Name	CPU (GHz)	Memory (MB)	Description	DBMS	Globus	OGSA-DAI
Bme-pc1	1.5	256	Basic personal information	SQL Server 2005	GT 4.0	OGSA-DAI WSRF 2.2
Bme-pc2	1.6	256	Basic health information	MySQL 4.1	GT 4.0	
Bme-pc3	1.2	512	Radiology information	Oracle 9i	GT 4.0	

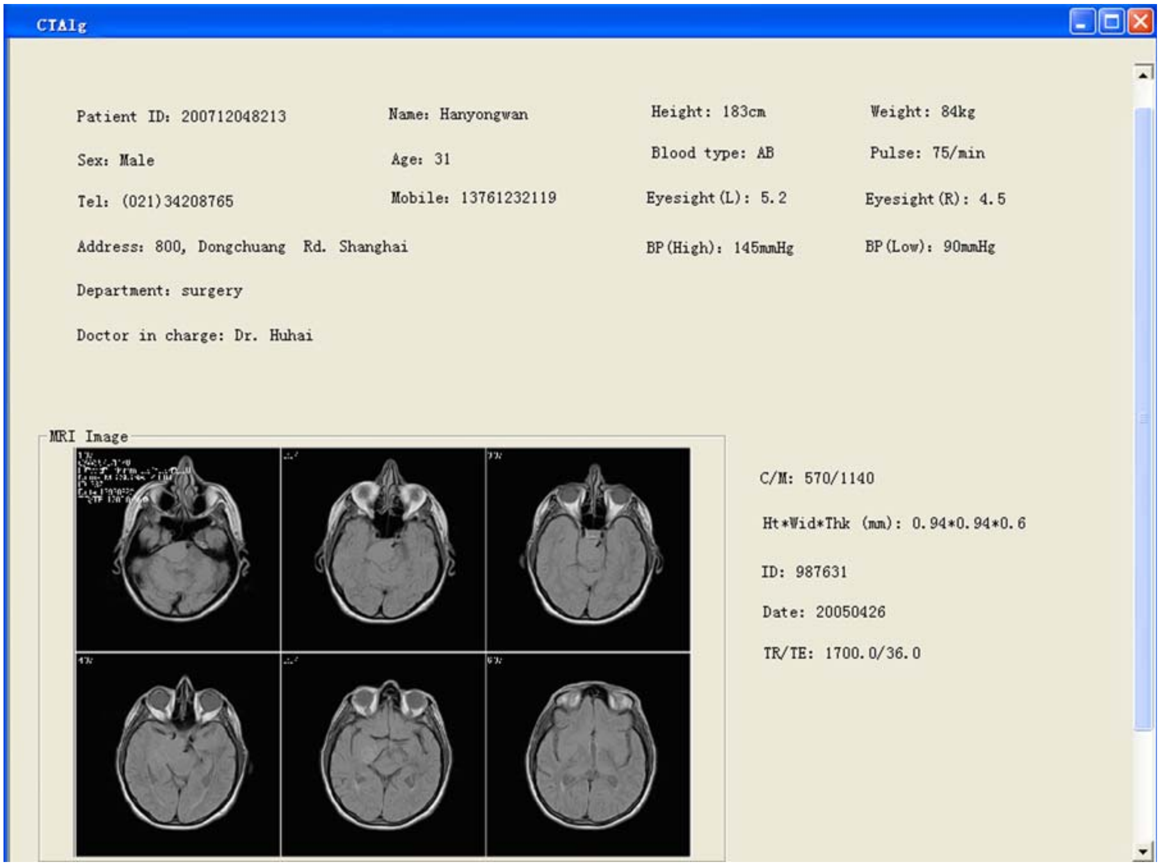

Fig 5. Result of the joint query instance (simulated data).

environment. Some issues as below should be taken into account for practical medical applications:

1) Efficiency: Grid applications always have special performance requirement which includes high-throughput, fault tolerance, etc. A high performance mechanism for query execution and data transfer is especially important to medical applications. It can be seen from the experiment result that there is a trend that performance of a joint query would deteriorate as the size of the table increases. Although there are some inherent reasons such as bandwidth, machine performance that we can hardly change at present, we still can improve the efficiency by developing new accessing mechanisms and optimizing data transfer.
2) Dynamic metadata modeling: In a practical medical database Grid environment, there will be more and more dynamic data resources,

Table 2. Comparison of Different Query's Response Time

Query / Records	Bme-pc1 (SQL Server)	Bme-pc2 (MySQL 4.1)	Bme-pc3 (Oracle 9i)	Joint query on system	Standard deviation
10,000	0.31s	0.27s	0.35s	0.67s	0.1829s
50,000	0.37s	0.33s	0.31s	0.83s	0.2479s
100,000	0.38s	0.41s	0.39s	0.89s	0.2486s
200,000	0.69s	0.63s	0.68s	1.13s	0.2331s

meaning metadata might be dynamic. Building effective connections between original data and dynamic metadata will be important to improve the effectiveness and efficiency of accessing dynamic distributed databases. On the other hand, the current metadata model that we designed support only limited data resources. In the future, we expect to expand our model to cover more data resource types, including relational databases, object-oriented databases, and unstructured data resources.

3) Security: data security is a fundamental requirement which mainly depends on access control and authentication mechanisms. OGSA-DAI provides a simple authentication layer that consists of a role-map file and an implementation class that does the role mapping between the Grid credentials used to access the system, which are based on the X509 certificates, and the credentials required to access the underlying DBMS. As each resource provider has to maintain a role-map file to authorize access to its resources, it puts a burden on the resource providers in managing the role-map files, especially when both the users and resource providers belong to multiple virtual organizations (VOs). Moreover, current authorization mechanisms cannot address all the issues that arise in dynamic Grid environments, which often encompass multiple organizations with different security policy, and some special security requirements relating to medical domain. So, new security mechanisms and access control methods should be developed to meet the special security requirements of medical database integration in Grid environment.

CONCLUSION

Effective integration of heterogeneous data sources has been studied as the most pressing challenge in medical application domain. In this paper, we present a prototype system for integration of heterogeneous medical data resources based on Grid technology with OGSA-DAI that provides a standard-based framework for accessing and integrating data resources in a distributed environment. It offers relative mature uniform service interfaces that support consistent access medical databases in a data-resource independent manner. After an overview of the architecture of the model, we analyze its functional components in detail, including OGSA-DAI, metadata, transaction, and query service components. Then, we test its effectiveness and performance through a query instance. The result shows that the system can provide an effective way to access underlying medical databases with a comparatively stable performance.

As the prototype system has been carried out and tested in a simulated information environment, there will be some practical issues to consider when we apply the underlying principles to practical medical applications, including security, efficiency, dynamic metadata modeling, etc. As a whole, the model that we designed provides a new solution to the seamless integration of heterogeneous medical databases in a dynamic and distributed environment.

ACKNOWLEDGEMENT

We would like to acknowledge contributions to this work by our colleagues, Dr. Chen Jun and Dr. Zhao Chenguang, for their support and by Ph.D. candidate Sun Xianjun for his help with the introduction of knowledge about Unix system.

REFERENCES

1. Landers T, Rosenberg RL: An overview of MULTIBASE, in Distributed systems, Vol. II: distributed data base systems, W.W. Chu. Artech House, Inc. 391–421, 1986

2. Carey MJ, Haas LM, Schwarz PM, et al: Towards heterogeneous multimedia information systems: The garlic approach. in Proceedings of the International Workshop on Research Issues in Data Engineering—Distributed Object Management—RIDE-DOM. 1995. Taipei, Taiwan: IEEE

3. Tohsato Y, Kosaka T, Date S, et al: Heterogeneous database federation using Grid technology for drug discovery process. Grid Computing in Life Science 3370:43–52, 2005

4. Stevens R, Baker P, Bechhofer S, et al: TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources. Bioinformatics 16(2):184–185, 2000

5. Foster I, Kesselman C, Tuecke S: The anatomy of the Grid: enabling scalable virtual organizations. Int J High Perform C 15(3):200–222, 2001

6. Zamboulis L, Hao F, Belhajjame K, et al: Data access and integration in the ISPIDER proteomics Grid. Data Integration in the Life Sciences, Proceedings 4075:3–18, 2006

7. Jasper E, Tong N, McBrien P, et al: View generation and optimisation in the automed data integration framework. Technical report, AutoMed Project. http://citeseer.ist.psu.edu/jasper03view.html, 2003

8. Grimshaw SA, Natrajan A, et al: From Legion to Avaki: The persistence of vision, in Grid computing: making the global infrastructure a reality, F. Berman, G. Fox, and T. Hey. 2003, pp 265–297 DOI 10.1002/0470867167.ch10

9. Alpdemir MN, Mukherjee A, Gounaris A, et al: OGSA-DQP: A service-based distributed query processor for the Grid. http://www.nesc.ac.uk/events/ahm2003/AHMCD/pdf/114.pdf

10. Saltz J, Oster S, Hastings S, et al: caGrid: Design and implementation of the core architecture of the cancer biomedical informatics Grid. Bioinformatics 22(15):1910–1916, 2006

11. Karasavvas K, Antonioletti M, Atkinson M, et al: Introduction to OGSA-DAI services. Scientific Applications of Grid Computing 3458:1–12, 2004

12. Antonioletti M, Krause A, Paton NW: An outline of the Global Grid forum data access and integration service specifications. Data Management in Grids 3836:71–84, 2005

13. Antonioletti M, Atkinson M, Baxter R, et al: The design and implementation of Grid database services in OGSA-DAI. Concurrency Computation Practice and Experience 17(2–4):357–376, 2005

14. Brady M, Gavaghan D, Simpson A, et al: eDiamond: a Grid-enabled federated database of annotated mammograms, in Grid computing—making the global infrastructure a reality, F. Berman, A. Hey, and G. Fox. John Wiley & Sons, Ltd. pp 923–943

15. Cole K, Ekin P, Mason L, et al: ConvertGrid: Grid enabling existing social science data infrastructures and services. Technical report, the ConvertGrid Project. http://www.ncess.ac.uk/research/pdp/papers/20060629_cole_ConvertGRID.pdf, 2005

16. Hannus M, Dolenc M, Klinc R, et al: Interoperability of Virtual Organizations on a Complex Semantic Grid. Technical report, the InteliGrid Project. http://www.inteliGrid.com/data/works/att/d51_2.content.03243.pdf

17. Droegemeier KK, Chandrasekar V, Clark R, et al: Linked environments for atmospheric discovery (LEAD): Architecture, technology roadmap and deployment strategy. in 85th AMS Annual Meeting, American Meteorological Society—Combined Preprints. 2005. San Diego, CA

18. Sotomayor B, Childers L: Globus Toolkit 4: programming Java services. San Fransisco: Morgan Kaufmann, 2006