

Towards Realization of Domain-Specific Scientific Data Cloud for Material Data Sharing

Jingjun Ge

School of Computer and Communication Engineering, University of Science and Technology Beijing, China

School of Information Engineering, Nanchang Hangkong University, Nanchang, China

Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing

Email: gejingjun@126.com

Changjun Hu, Xin Liu, Wei Lin, and Haolei Zuo

School of Computer and Communication Engineering, University of Science and Technology Beijing, China

Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing

Email: {changjunhu, weiliu, xinliu, haoleizuo}@ies.ustb.edu.cn

Abstract—Scientific data sharing has been an important area of research for several years. However, due to the sharp increase in scientific data, the existing scientific data sharing systems are becoming complicated and cannot meet the demands of current scientific domain communication. In this paper, the realization of material scientific data cloud is introduced to manage large-scale scientific data resources. In fact, we provide a framework for improved integration and data sharing capability improvements through interconnecting agent system and the unified environment for data mapping and integration. At present, we have been realized the prototype system of material scientific data cloud and demonstrate its effectiveness and practicality in material scientific data sharing project implementation.

Index Terms—scientific data sharing, interconnecting agent system, data integration, data resource graph

I. INTRODUCTION

In recent years, greatly accelerating process of global information has resulted in science data with unprecedented speed growth and communication between multiple disciplines more closely. Faced with these massive, distributed scientific data, the scientific community has been seeking for how to manage and share large-scale scientific data [1]. Cloud computing and Big Data technology brings new opportunity for scientific data sharing. For now, Considerable research has focused on software toolkits for data integration and management. There are several prominent projects focused on open-source cloud platform related to scientific data management: OSDC (Open Scientific Data Cloud) [2], Pervasive DataCloud2 [3], and OpenNebula [4], et al.

OSDC [2] started in 2010 is a hosted platform managed by a single entity (the Open Cloud Consortium), which is a persistent, distributed storage and

computing resource designed to manage, analyze, share and archive scientific data. OSDC supports elastic, on demand virtual machines, similar to Amazon's EC2 [5] service, storage for medium to large datasets provided by the Hadoop [6] or Sector [7]. Pervasive DataCloud2 [4] is a secure and reliable on-demand services platform powered by Amazon Web Services for software developers who need to rapidly create on-demand data integration. The platform can leverage Pervasive data services, including data adapters, distributes data infrastructure software and ETL tools that integrate, analyze, secure, manage and harvest data from disparate sources. OpenNebula is an open source cloud computing toolkit for managing heterogeneous distributed data center infrastructures, including features for integration, management, scalability, security and accounting. OpenNebula emphasizes standardization, interoperability and portability, providing users and administrators with a choice of several cloud interfaces (EC2 Query [8], OGF OCCI [9] and vCloud) and hypervisors (Xen [10], KVM [11] and VMware [12,13,14]), and a flexible architecture that can accommodate multiple hardware and software combinations in a data center [15,16,17].

There are currently great challenges for addressing the problem of large-scale scientific data sharing in order to meet the demands of scientific domain communication.

(1) The interconnection of scientific data resource centers. Currently, the traditional connections are used so that scientific datasets can be replicated over multiple geographically distributed data resource centers. The direction of future effort is more considering the high-efficiency and energy-saving method for interconnecting data resource centers, with these connections, medium to large scientific datasets can easily be ingested and shared.

(2) Large scale and complex domains scientific data integration. With the scale of scientific datasets increasing from MB to GB and PB, and the scientific data involving more and more domain-specific data formats. Data integration is becoming extremely complicated.

Corresponding author.

Email addresses: changjunhu@ies.ustb.edu.cn (Changjun Hu).

Currently, there is lack of such a platform to support large-scale scientific domain data integration.

(3) Mapping and Index data from diversity, dispersion data sources need to capture both text values and structural information of domain scientific data, to build a separate index for each attribute to support structured queries, and to structure an inverted list to support keyword search.

Unique compared to the exist data frameworks system incorporating the above, Domain-Specific Scientific Data Cloud (DSDC) presented in this paper is an efficient data services platform for integrating ,managing, analyzing, archiving and sharing scientific data. Through the Internet to interconnect various data resource centers, DSDC is designed to build up a unified data sharing environment and a unified service resources graph that supports large-scale scientific data integration, massive storage, high-performance distributed data processing, and visualization data queries.

The main contributions of this paper include:

(1) A high-efficiency and energy-saving method for interconnecting data resource centers.

(2) On-the-fly integration and automatically build mapping and associations from scientific data resources. When having more data sources or losing some data sources, we need not do from scratch.

(3) Create a unified environment for visualizing data services on an as-needed basis, customizing scientist's data spaces based on data resource graph.

This paper is organized as follows. In Section 2 we describe the DSDC architecture. Section 3 discusses the Interconnecting Agent System for data resource centers. Section 4 gives the unified environment for data mapping and integration. In Section 5, we present the application example of material scientific data sharing project, and conclude in Section 6.

II. DSDC MODEL ARCHITECTURE

The core architecture of DSDC (shown in Figure 1) consists of four tiers including Unified Management, Data Services, Virtual Data Space, Resource Interconnect and Data Resource Centers.

The Unified Management is to provide unified configuration, deployment and monitoring for distributed data resources, computing resources and storage resources, and provides a visual interface management tool for administrator.

The Data Services based on the service resources provides scientists with data resource services, data retrieval services, data customization services, data computing services and data storage services, et al.

The Virtual Data Space, including Bridge Model for Resource Mapping, On-the-fly Data Integration Model that builds a data resource graph for data resources sharing.

The Resource Interconnection is the basis for achieving DSDC. Based upon Interconnecting Agent System and multi-network integration technology, achieve the concatenation of multiple data resource centers. The data resource centers include unified access the physical

machines service relied on visualization of machines and transparent interconnection service for the distributed meta-data storages.

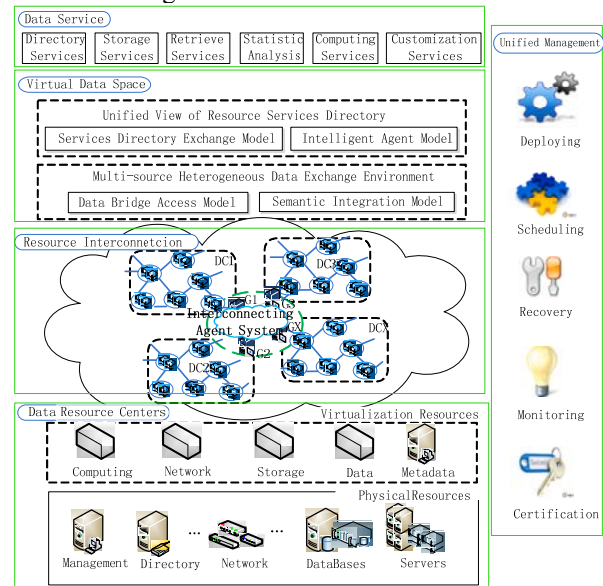


Figure 1. The Model Architecture of DSDC

III. INTERCONNECTING AGENT SYSTEM

In this section, the definition of the Interconnecting Agent System (IAS) is presented as follows:

Definition 1: The Interconnecting Agent system *IAS* = (*Main Server*, *Meta-Data Server*, *Concatenation Server*, *Backup Server*) is a 4-Tuple, where:

(1) *Main Server*, being used as Connector between Interconnecting Agent Systems, is a set of primary servers. Meanwhile, within the Interconnecting Agent System, each server of *Main Server* is responsible for meta-data service resource management, task monitoring, and load balancing, etc;

(2) *Meta-Data Server* is considered the collection of meta-data and structural information. Each data resource center has its own meta-data server. Each meta-data server is responsible for capturing resource structural information within each data resource center.

(3) *Concatenation Server* is a collection of servers which responsible for unified meta-data information conveying between data resource centers. *Concatenation Server*, using Mediator / Wrapper Architecture, in fact, makes the data resource information of every data resource center possible to spread over all the data resource centers.

(4) *Backup Server* is a collection of storage servers responsible for backup data resource for data resource center.

Interconnecting Agent System (shown in Figure 2) connected by a series of multiple subordinate servers, is considered as server network system.

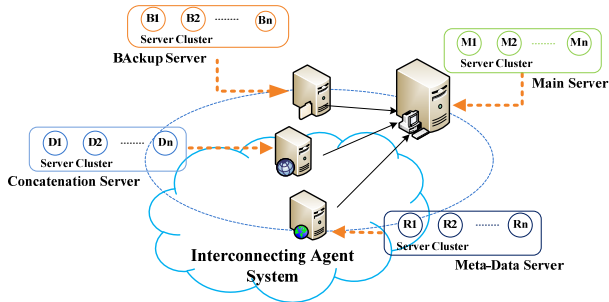


Figure 2 the Interconnecting Agent System

Interconnecting Agent System has a modular architecture, designed to perform interconnection over multiple structured and semi-structured data resources in a flexible and a reconfigurable way. Various modular component servers perform their own duties. For example, Concatenation Servers dedicated provide the unified meta-data information conveying, Meta-Data Servers are responsible for capturing data resource meta-data information within each data resource center, and some others are responsible for storage data resource backup etc. Interconnecting Agent System consists of both interconnection occurred between the data resource centers and IAS, and interconnection occurred within IAS. With the two connections, scientific data resource centers just like a modular component can be easily plug into DSDC. It provides data resources interconnection management that enables large-scale scientific datasets can easily be ingested and be shared.

IV. MAPPING AND INTEGRATION ENVIRONMENT

In this section, we describe two models of DSDC designed to perform data mapping and integration over multiple structured and semi-structured domain-special scientific data resources in a flexible and reconfigurable way. The Bridge Model for Resource Mapping which uses mediated schema, provides a method to map data to the data resource graph. On-the-fly integration model supports pay-as-you-go data integration for those data from multiple geographically distributed data resource centers. The combination of two models achieves the unified environment for data mapping and integration. The complete definition of these two models will be given in the following.

A. Bridge Model for Resource Mapping

Definition 2: The Bridge Model for Resource Mapping, $Bridge=(C, M, \text{ and } S)$, is a 3-Tuple, where:

- (1) C is the data of the requester; the specific performance is in the form of meta-data;
- (2) M is the data intermediate services, and is responsible for the interaction between C and S ;
- (3) $S \in S_C$, S represents a data source, S_C is the set of all data sources;

The relationship between the three elements of the model can be expressed as $C \xrightarrow{M} S$. Through the description of the relationship between C, M , and S , we can see that, C and S are not directly connected, the data access between C and S entirely depends on M , M uses different data

access API to get data from S and passes data to C , C modifies data and returns data to M , and M passes data to S .

Definition 3: The Data Intermediate Services

$M=(S_{Type}, Meta, Graph, Rule)$ is expressed as a 4-Tuple, where:

- (1) S_{Type} indicates the type of data source, and the current data source types includes DBMS, File System, Excel, XML, etc.
- (2) $Meta$ is the meta-data data requests;
- (3) $Graph$ is a data resource graph, which is the collection of data objects with multiple roots, can record all operations of data objects.
- (4) $Rule$ are generated rules for the data resource graph, according for the different data sources, it is used different rules of access model.

M is a sub-elements of Bridge model defined above. M can access data sources, collect data according to application requirements, generate data resource graph based on different rules, assemble the data resource graph, and send the changed data objects of the data resource graph to the data sources. M has many data wrappers, and each wrapper is responsible for obtaining data corresponds to particular data sources. The input of M corresponds to various forms of data sources and technologies (XML, JMS, JCA, JDBC, etc.). However, the output of M is always in the same form of data resource graph.

Definition 4: $Graph=(Root, Object, Record)$, is a 3-Tuple, where:

- (1) $Root$ is corresponds to each of target data source described by the meta-data, and each data resource graph can contain many Roots.
- (2) $Object$, is an essential component of data resource graph. And it mapped to the data elements within the data resource centers,
- (3) $Record$ is the record of modify data resource graph, and the modify record is initially empty.

$Graph$ composed by the $Root$ data object, all data object associated with the root, and modifies record. We use $Graph$ to represent relationship between data resource centers, as well as within data resource centers. $Graph$ is generated by M , C can traverses, read and modify data object of $Graph$. Data resource graph can be serialized to XML Schema, and $Graph$ provides many functions for operator, e.g. $CreateDataGraph()$, $GetRootObject()$ and $DeleteDataGraph()$, etc.

Definition 5: $Object=(Name, Type, \sum \langle ai, vi \rangle, ISA)$ is a 4-Tuple, where:

- (1) $Name$ is the name of the data object, and is used as its unique identifier;
- (2) $Type$ is the description of all data object types;
- (3) $\sum \langle ai, vi \rangle$ is attribute lists of data objects, where: $\langle ai, vi \rangle$ indicate respectively the attributes and the corresponding attribute values;
- (4) ISA is the relationship between data objects, and ISA is defined as partial order upon $Object$, with the nature of the reflexive and transitive closure;

$Object$ is the basic constitute elements of $Graph$ defined above, each Data Object uses a series of

properties to store their contents, and the connection between Data objects is depended on the relationship of ISA. Data objects provide many operating functions for data object, such as *CreateDataObject()*, *GetDataObject()* and *DeleteDataObject()*, etc. Meanwhile, *Object* also offers access operations for properties, such as *getter* operator and *setter* operator.

The Bridge Model for Resource Mapping, which is based on the algebra theoretical foundation, provides an approach to automatically build data resource graph to make data resource mapping and index more intuitive. Three mapping levels occur in resource mapping process: Data Level, Semantics Level, and Query Level. We construct data resource graph on an as-needed basis, and use Query in the XPath [18] language to traverse the data resource graph, and capture the result of Query answering to build a separate mapping and index for each attribute to support structured queries, and to structure an inverted list to support keyword search. The example of material evaluation as follow (shown in Figure 3), indicates that how to query and traverse data resource graph by specific operating functions on the data resource graph to obtain both text values and structural information of material evaluation.

```
//First get the root element of the data resource graph
```

```
DataObject root = dataGraph.getRootObject();
```

```
// Obtain the Object of material evaluation graph
```

```
DataObject Mat= root.getDataObject("Components");
```

```
// Traverse the material component
```

```
DataObject Mat= (DataObject) MatComponent.get();
```

```
// If you choose Material Components of 'Matrix', directly  
capture 'Matrix' material component from material evaluation  
graph
```

```
String Mat_C = MatComponent.CaputeProperties("Matrix");
```

```
//Query Density properties of 'Matrix' Material Components
```

```
String Mat_Q = MatComponent.getString("Density");
```

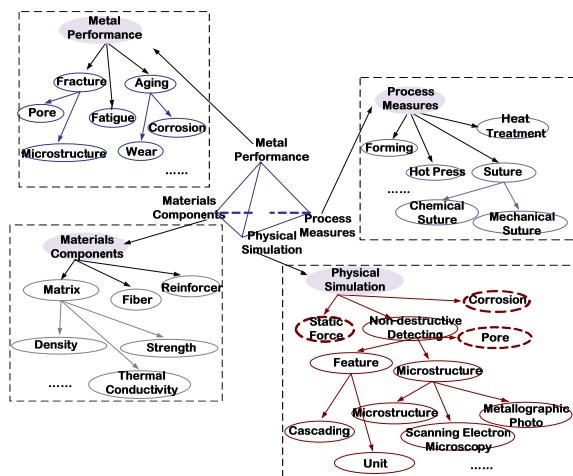


Figure 3. the data resource graph of material evaluation information

B. On-the-fly Integration Model for Large-scale Scientific Data

Definition 6: On-the-fly Integration Model for large-scale scientific data defined as the follows:

On-the-fly Integration=(*Graph*,*P*,*Axiom*,*Rule*), where:

- (1) *Graph* is the collection of data resource graphs;
- (2) *P* is the collection of correlation attributes between semantic mapping graph;
- (3) *Axiom* is the axioms of semantic model;
- (4) *Rule* is a set of Horn-style rules, and the rule is constituted by *Graph* and *P*;

On-the-fly Integration can reclassify data resource graphs according to the axioms and rules, find the implicit knowledge in data resource graphs, and integrate data resource graphs on demand.

Definition 7: Correlation Attributes $P=(ISA, P_L, P_D)$, is a 3-Tuple, where:

- (1) *ISA* is the relationship between data objects, for example, the relationship of between the parent class and subclass;
- (2) *P_L* is the set of association properties between data resource graphs, that is the attribute set of object types;
- (3) *P_D* is the set of association properties between data resource graph and specific data types, that is the attribute set of data types;

On the basis of the data resource graph generated by the Bridge Model for Resource Mapping (Shown in Figure 4), On-the-fly Integration Model, According to various domain knowledge and rule, integrates scientific data resources on an as-needed basis. On-the-fly Integration Model do not need full integration, even when we have more data sources or lose some data sources, we need not integrate data from scratch. On-the-fly Integration Model supports queries that combine keywords and structure on the data resource graphs. The result of query answering can be exponential in the size of the mapping attribute correspondences, so that When encoding resource mappings using a Bayes Net. On-the-fly Integration Model provides an API for customizing scientist's data spaces and index for all the data resources by the data resource graph.

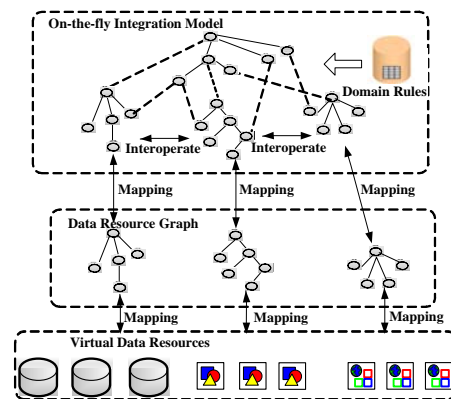


Figure 4. scientific data resources integration on the data resource graph

V. MATERIAL SCIENTIFIC SHARING APPLICATION

The platform of the Material Scientific Data Cloud is a prototype system of Domain-Specific Scientific Data Cloud. According to the characteristic of material domain scientific data, the platform improves sharing capabilities of material scientific data, and provides advanced applications for various material research. Currently, the platform has gathered a large number of material scientific data resources in North China, has built multiple data resource centers, including metallic materials, inorganic non-metallic materials, organic polymer materials, composite materials.

The Material Scientific Data Cloud platform includes three layers: virtual infrastructure, virtual data, and data service (as shown in Figure 5). The platform is a distributed architecture based on HADOOP [6] and the existing application server clusters.

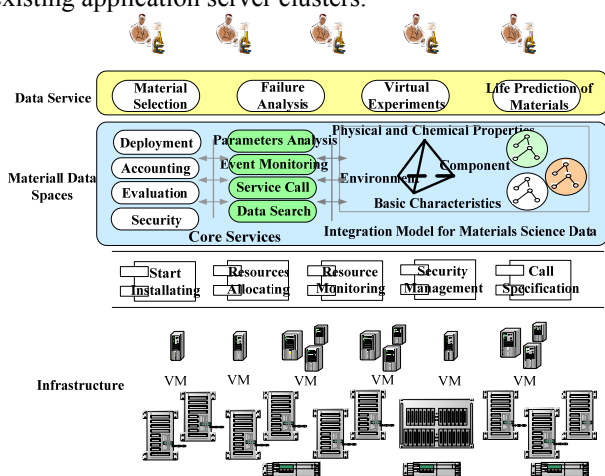


Figure 5 The Architecture of Materials Scientific Data Cloud

Material Scientific Data Cloud based upon the unified environment for integrating and sharing large-scale scientific data, take advantage of a hosted model (Interconnecting Agent System, Bridge Model for Resource Mapping, and On-the-fly Integration Model), without adopting federated or virtual organization, concatenates material scientific data resources, mapping between the data resource centers, and supports data integration and universal search on the data resource graph, as well as new data visualization techniques and interfaces using a graph-based GUI to improve the integration and sharing management capabilities of material scientific data resources in the future.

VI. CONCLUSIONS AND FUTURE WORK

This article shows how Domain-Specific Scientific Data Cloud and its models (Interconnecting Agent System, Bridge Model for Resource Mapping, and On-the-fly Integration Model) help large-scale scientific data resources to aggregate data resources, to build a virtual data resource graph, to generate a unified environment for data mapping and integration, and to provide visualization management interfaces. The Interconnecting Agent System guarantees data resources interconnection high flexibility, and saves cost. The Bridge Model for Resource Mapping enables data resources map to data

resource graph more intuitive. On-the-fly Integration Model integrates scientific data resources on an as-needed basis, and supports queries that combine keywords and structure on the data resource graphs. Currently, we have developed the prototype system of Material Scientific Data Cloud based on DSDC. Now we are applying to the scientific data sharing project construction, and will gradually be promoted.

However, DSDC and its technologies need to be progressive in practice, be amended and improved. In the future work, we need to do more in-depth research on construction, semantic data integration, storage, authentication, and security management. We will need tools for managing both the underlying resources and the resulting distributed data services. Further extensions of IAS are planned. Furthermore, we need to extend the optimize infrastructure usability regarding to the green-IT paradigm, accelerate DSDC from early prototypes to production systems.

ACKNOWLEDGMENT

This work is supported by the National Basic Research Program of China under Grant No. 2013CB329605; Key Project of the National Twelfth-Five Year Research Program of China under Grant No. 2011BAK08B04, the 2012 Ladder Plan Project of Beijing Key Laboratory of Knowledge Engineering for Materials Science, No. Z121101002812005.

REFERENCES

- [1] Myers J et al., "A Collaborative Informatics Infrastructure for Multi-scale Science," In Proceedings of the Challenges of Large Applications in Distributed Environments (CLADE) Workshop, June 2004.
- [2] Robert L. Grossman, Yunhong Gu, Joe Mambretti, Michal Sabala, Alex Szalay, and Kevin White, "An overview of the Open Science Data Cloud," In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, Chicago, Illinois, pp. 21-25, June 2010.
- [3] Rapidly Develop Data Services On-demand With Pervasive Datacloud2(2010). Retrieved Jan 5, 2012 from: http://cloud.pervasive.com/Portals/1/Documents/DataCloud2_ServiceDescription_14.pdf.
- [4] OpenNebula Project(2008). Retrieved June 7, 2012 from <http://www.opennebula.org/>.
- [5] Amazon. Amazon Elastic Compute Cloud(2008). Retrieved June 7, 2012 from <http://aws.amazon.com/ec2>.
- [6] D. Borthaku. The Hadoop distributed file system: Architecture and design(2007). Retrieved Jan 5, 2012 from lucene.apache.org/hadoop.
- [7] Y. Gu and R. L. Grossman, "Sector and sphere: Towards simplified storage and processing of large scale distributed data," *Philosophical Transactions of the Royal Society A*, also arxiv:0809.1181, 2009.
- [8] EC2 Instance Metadata Query Tool. Retrieved Jan 5, 2012 from <http://aws.amazon.com/code/1825>.
- [9] Installation, configuration & example requests for OGF-OCCE v0.2. Retrieved June 7, 2012 from <http://dev.opennebula.org/projects/ogf-occe>.
- [10] P. Barham, B. Dragovic, K. Faser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization," In Proceedings of the Nineteenth

ACM Symposium on Operating Systems Principles (SOSP), October 2003.

- [11] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, "KVM: The Linux Virtual Machine Monitor," In *Proceedings of the Linux Symposium*, pp. 225–230, 2007.
- [12] L. Grit, D. Irwin, A. Yumerefendi, and J. Chase, "Virtual Machine Hosting for Networked Clusters: Building the Foundations for "Autonomic" Orchestration," In *Proceedings of the First International Workshop on Virtualization Technology in Distributed Computing (VTDC)*, November 2006.
- [13] VMware virtualization technology (2007). Retrieved June 7, 2012 from <http://www.vmware.com>.
- [14] Virtual Data System (2007). Retrieved June 7, 2012 from <http://vds.uchicago.edu>.
- [15] L. Mao, Y. Yang, H. Xu, "Design and Optimization of Cloud-Oriented Workflow System," *Journal of Software*, vol. 8, no. 1, pp.251-258, 2013.
- [16] G. Wei, "Complex Learning System for Behavior Factor based Data Analysis," *Journal of Software*, vol. 8, no. 4, p.1003-1010, 2013.
- [17] Q. Chen, Y. Ou, H. Sun, "Design and Implement of Customer Communication Behavior Analysis System," *Journal of Software*, vol. 6, no. 8, pp.1484-1491, 2011.
- [18] World Wide Web Consortium. XML path language (XPath) Version 1.0. W3C Recommendation(1999). Retrieved June 7, 2012 from <http://www.w3.org/TR/xpath.html>

Jingjun Ge is currently a Ph.D. candidate in the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. He received his M.S degree in C.S in 2005 from the Graduate School of Changsha University of Science and Technology. He joined Material Science Data Sharing Laboratory (a member of researcher) and Domain Knowledge Management Laboratory, USTB, China. His current research interests focus on semantic data integration and domain-special scientific data management.

Changjun Hu received the Ph.D. degrees in C.S, in 2001 from the Graduate School of the Peking University. During his career, she has held various research and academic positions. Since 2004, he has been with University of Science and Technology Beijing, China, where he is currently the Head of the School of Computer and Communication Engineering and the Director of the High Performance Computing and Data Engineering Laboratory. In the past decade, he has been working on a number of industry-sponsored projects in wireless communications and mobile Internet. His research interests include high performance computing and data engineering.

Prof Hu is an IEEE Fellow and ACM.

Xin Liu is currently a Ph.D. candidate in the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His current research interests are in the area of machine learning and data mining, data mining inspired by computational immunology, data translation and reliability issue in knowledge discovery.

Wei Lin is currently a M.S. candidate in the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. She joined Material Science Data Sharing Laboratory, USTB, China. Her current research interests focus on text learning, web mining and imbalance learning.

Haolei Zuo is currently a M.S. candidate in the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. He joined Material Science Data Sharing Laboratory and Domain Knowledge Management Laboratory, USTB, China. His current research interests are in the area of machine learning and data mining, specialized in causal discovery, data mining for software engineering, data mining inspired by computational immunology, data translation and reliability issue in knowledge discovery.