DATA SCIENCE JOURNAL

Li, Y and Hu, C 2016 Process Materials Scientific Data for Intelligent Service Using a Dataspace Model. *Data Science Journal,* 15: 7, pp. 1–17, DOI: http://dx.doi.org/10.5334/dsj-2016-007

RESEARCH PAPER

Process Materials Scientific Data for Intelligent Service Using a Dataspace Model

Yang Li[1,2] and Changjun Hu[1,2]

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing (USTB), Beijing, 100083, China

[2] Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing, 100083, China

Corresponding author: Yang Li (mailbox.liyang@gmail.com)

Nowadays, materials scientific data come from lab experiments, simulations, individual archives, enterprise and internet in all scales and formats. The data flood has outpaced our capability to process, manage, analyze, and provide intelligent services. Extracting valuable information from the huge data ocean is necessary for improving the quality of domain services. The most acute information management challenges today stem from organizations relying on amounts of diverse, interrelated data sources, but having no way to manage the dataspaces in an integrated, user-demand driven and services convenient way. Thus, we proposed the model of Virtual DataSpace (VDS) in materials science field to organize multi-source and heterogeneous data resources and offer services on the data in place without losing context information. First, the concept and theoretical analysis are described for the model. Then the methods for construction of the model is proposed based on users' interests. Furthermore, the dynamic evolution algorithm of VDS is analyzed using the user feedback mechanism. Finally, we showed its efficiency for intelligent, real-time, on-demand services in the field of materials engineering.

Keywords: Intelligent services; Materials Scientific data; Semantic mapping; Big Data; evolutionary algorithm

1 Introduction

As many industries and research labs handle increasing amount of data in materials science, big data (Toffler 1980) is being considered as an important issue for materials engineering services. Extracting meaningful and valuable information from large-scale datasets is essential for providing new applications as well as improving the quality of existing services. Data management, reuse and collaboration in various sources pose new challenges to the field of materials science (Howe et al. 2008; Lynch 2008).

In recent years, along with the continuous accumulation of scientific data and the constantly changing of practical requirements, the "big data" management issues should be addressed in the aspect of domain scientific data. Materials scientific data comes from lab experiments, simulations, individual archives, enterprise and internet in all scales and formats. This data flood has outpaced our capability to process, analyze, store and understand these datasets. Materials science data possess the typical characteristics of big data.

1. Volume: Massive materials scientific data have been accumulated in the past several decades. This provides large amounts of available data, but makes it more difficult to rapidly obtain valuable information.
2. Variety: Materials scientific data are distributed in different sources and heterogeneous data are manifested in different formats, such as sheets, semi-structured XML, non-structured images, etc.
3. Velocity: Materials scientific data are constantly changing, especially the experimental data. Many of the data sources are very dynamic, and the number of data sources is also expanding.

4. Veracity: Materials scientific data sources are of widely differing qualities, with significant differences in accuracy and timeliness. Highly precise data are required in most of the engineering services.

The problem we mainly focused is how to manage materials scientific data in an intelligent way for better services based on users' requirements. The issues addressed for meeting the user demands in data services include: extract valuable data from variety resources and remove the redundant data; represent data information by mining the complex relationships in datasets; compose data queries using specific domain terminology; adapt changing requirements of the users in data organization. Thus it is difficult to meet these demands for data services using traditional technology. A new service model is required in this circumstance.

The main objective of this paper is to present the modelling and construction method for Virtual DataSpaces (VDS) for managing big data and providing data services in the field of materials engineering. The concept of VDS is proposed (Hu et al. 2016), which is different from the traditional data management technology. VDS is not only developed from dataspace (Franklin, Halevy & Maier 2005) but also represents an innovation in the deepening of it. As a new mode of data organization, VDS not only illustrates a new modelling method to organise and process the big data but also presents a new evolution method to manage the continually changing data, services, and user demands. VDS faces the complex data management and dynamic service demand in the field of materials engineering. The process and algorithm of VDS construction is based on the user feedback. Finally, the effectiveness of the model is verified by the case description of a materials science domain application and the comparisons of different models are also shown. VDS has its own characteristics, such as the modelling ideology of "data first", the subject related, the domain knowledge model, the emphasis of "services", the support of data-associated mapping and dynamic evolution, the real-time and on-demand service, etc. Therefore, the new mode "VDS" is suitable for effectively managing big data and dynamically providing intelligent services in materials engineering.

In addition to the introduction section, there are six sections in this paper. The rest of this paper is organised as follows. In Section 2, the previous work about big data management and intelligent services is introduced, and the related research about dataspace is investigated. In Section 3, the modelling theory and construction method of VDS is described. In Section 4, the evolution process of VDS is elaborated and analysed. In Section 5, the analysis and comparison between VDS and other models is discussed, and the application case of an intelligent service in the field of materials engineering is proposed. Finally, the conclusion and further work is presented in Section 6.

2 Related Work

2.1 Big Data

Both theoretical and practical studies on big data management have got increasingly attention given the background of new challenges. Several important characteristics about big data have been enumerated and the platform requirement challenges for big data architecture have been described (Wang et al. 2011). An overview of the research issues and achievements in the field of big data analysis has been provided and the multidimensional data analysis problems about big data have been discussed (Cuzzocrea, Song & Davis 2011). A new algorithm with the help of a semantic graph for the more efficiently and intelligently processing big data has been proposed (Qu 2012). However, because the semantics of big data are only described with RDFs, it has limited semantic representation ability. Active data (Simonet, Fedak & Ripeanu 2012) as a programming model has been proposed to alleviate the complexity of the data lifecycle and automatically improve the expressiveness of data management applications. However, this model does not consider the semantic association characteristics and does not form a refined ideology about dynamic evolution.

The formalisation of a time window selection strategy along with a literature review is presented (Ballings, Poel 2012). This study analysed the improvement in churn-model performance by extending the customer event history from one to sixteen years and uses logistic regression, classification trees and bagging in combination with classification trees. Using a time window could substantially decrease data-related burdens, such as data storage, preparation and analysis. This is particularly valuable when decreasing computational complexity is paramount. A new method to create necessary summary information by reducing the dimension of a coalition transaction data is proposed (Lee, Lee & Sohn 2013) to develop a behaviour-scoring model and contribute big data analysis for a coalition loyalty program. For addressing the problem of big data analysis in the new business intelligence era (Kwon, Sim 2013), evaluated the causality between the data set

characteristics as independent variables and the performance metrics as dependent variables using a multiple regression method. These methods and ideas about big data processing provide a guide for our work.

2.2 Intelligent Service

An intelligent service refers to a service that automatically recognises the explicit or implicit user demands and then meets those demands proactively, efficiently, safely and accurately. The prerequisites for an intelligent service software environment include a standard information infrastructure, a data accumulation that can be efficiently used, the opening and sharing about data services, and quality assurance about data legitimacy. Related research about intelligent service environments has developed to a certain extent in recent years. Intelligent service environments might become increasingly important with the development of big data in the future.

A novel Artificial Neural Network (ANN)-based service selection (ANNSS) algorithm (Cai et al.2009) was proposed to overcome the shortcomings of blindness and randomness in traditional service selection algorithms. The novel algorithm chooses exactly the most appropriate service in a ubiquitous web services environment. An intelligent service-integrated platform (Yeh, Chen & Chen 2011) has also been proposed. It employs the software agent as the framework to construct an integrated information system mechanism for preventing monetary loss due to the information gap. It also employs radio frequency identification (RFID) technology to realise the smart shelf as the trigger point for the retrieval of commodity messages. This framework might help enhance the performance of sales outlets and improve customer service while addressing the time effect issue with a popular commodity.

A solution for the converged context management framework has been presented (Baladron 2012). This framework takes advantage of the features of intelligence and convergence in next-generation networks. At the same time, it allows the seamless integration, monitoring, and control for heterogeneous sensors and devices under a single context-aware service layer. This layer is centred on a context intelligence module that combines clustering algorithms and semantics to learn from the users' usage histories, and takes advantage of this information to infer missing or high-level context data. Finally, it provides personalised services to end users using a context-aware method. The mobile agent based on a distributed geographic information service system has been created using the intelligent service chain construction technology (Liu, Zhang & Li 2012). A web-based intelligent self-diagnosis medical system (Pyung-Jin, Heon & Ungmo 2009) has been presented to go beyond finding the name a disease by suggesting synthetic preventive health care methods based on analysing lifestyle, food and nutrition. These technical methods and service architectures for intelligent services have a reference value for our research.

2.3 Dataspace

Researchers are trying to seek a new technology to address the new challenges of big data management and the intelligent service environment. Since the proposal of the concept of "dataspace" (Franklin, Halevy & Maier 2005) as a new mode of data service, researchers have designed wide variety of dataspace models and have proposed several prototype systems that are consistent with their respective needs.

From the perspective of data, the essence of a dataspace is a collection of big data. Dataspace technology could be used in a wide range of areas, such as Personal Information Management (PIM), organising and processing scientific or engineering data, social network, and so on. The model research is the basis of dataspace construction. With the different domain features, there might be different data models to describe and organise the complex data for different service requirements. The primary existing dataspace models are as follows:

(1) iDM (iMeMex Data Model)
The iDM is the dedicated data model of the iMeMex system (Dittrich, Salles 2006). It organises and expresses all of the personal data resources in the form of a resource view and a resource view class. The iDM is the first dataspace model that is able to describe heterogeneous personal data resources in a unified form. However, this model uses a new query language, iQL, which based on XPath and a SQL-like query language. For ordinary users, it is difficult to get started quickly.

(2) UDM (Unified Data Model)
UDM is a data model that is suitable for desktop search systems (Pradhan 2007). UDM adopts the database/information retrieval (DB/IR) integration approach that is able to dive into data

items to retrieve the desktop dataspace. However, its new query language TALZBRA is very complex, and this model does not support shortcut queries.

(3) Probabilistic Semantic Data Model (P-DM)
P-DM is a dataspace model completely based on probability (Saema, Dong & Halevy 2009). P-DM uses the probabilistic mediate schema (Saema, Dong & Halevy 2008) and probabilistic semantic mapping to achieve the semantic integration of heterogeneous data sources. This model addresses the problem of uncertainty (Dong, Halevy & Yu 2009; Singh, Jain 2011) in different levels of dataspace and supports the top-k query response that could improve the quality of queries. However, its schema matching probability is not very accurate, and the model is difficult to extend.

(4) Domain Model
The Domain Model is a dataspace model that is similar to the ontology method (Dong, Halevy 2005; Dong, Halevy & Madhavan 2005). It supports a simple semantic query operation, but the mapping between the domain model and the data sources needs to be constructed manually.

(5) CoreSpace Model & TaskSpace Model
The CoreSpace Model and the TaskSpace Model are the core parts of the personal dataspace management system, OrientSpace (Li, Meng 2008; Li, Meng 2009; Li, Meng & Kou 2009), which automatically constructs a personal dataspace. This system considers the behaviour characteristics of the subject, and highlights the effect of subject characteristics on the dataspace. However, an in-depth discussion about the inconsistency and the optimisation of the weight calculation is needed.

(6) Triple Model
The Triple Model is a flexible data model (Zhong, Liu & Qian 2008) that is similar to the RDF and expresses heterogeneous data in the form of triples. This model represents the hierarchy of file resources through a graph model, thus it can express the heterogeneous data simply and flexibly in dataspace. However, the Triple Model does not support path expression queries, does not consider uncertainty, and queries by Subject Predicate Object (SPO) are difficult.

In addition to the above six models, there are some other dataspace models, such as the resource space model (RSM) (Zhuge 2004), the PAD model (Dong et. al 2009), etc. They provide the related theoretical basis for dataspace research.

Researchers have developed several prototype systems based on the model research about dataspace. The prototype of iMeMex supports a keyword query, a structured query and a path query, but does not support a semantic query (Blunschi 2007). Semex (Dong, Halevy 2005) is a prototype system that supports a neighbouring keyword query based on the domain model and global relational view, but it lacks a full-text index about the text content and unduly depends on the model. OrientSpace (Li, Meng 2008; Li, Meng & Zhang 2008) is a personal dataspace prototype system that supports the "pay-as-you-go" method. It supports incremental optimisation of the data service by constantly changing based on user behaviour analysis. These prototype systems are all personal dataspaces, and are not very similar to the scientific dataspace in the field of engineering application.

PAYGO (Madhavan et al. 2007) supports automatic pattern matching and similarity clustering analyses that support automatic evolution based on user feedback, but it lacks a detailed description. UDI (Saema, Dong & Halevy 2008) supports the mode merge and automatic evolution, but its assumption is too simple and restricts the scope of application. Roomba (Jeffery, Franklin & Halevy 2008) is the first dataspace system that genuinely emphasises evolution. It introduces a user feedback mechanism, stores the data using generic triples, and adopts the method of instance-based string similarity matching. Quarry (Howe 2008) materialised data to generic triples and realised data integration using the global mode, but it occupies a large amount of storage space.

CopyCat (Ives 2009) embodies a more interactive data integration approach based on the SCP (Smart Copy and Paste) model. This prototype supports system optimisation by automatically learning based on user feedback. Similar to CopyCat, Octopus (Cafarelia, Halevy & Khoussainova 2009) is a dataspace prototype system that includes a variety of functions, such as searching, information extraction, data cleaning,

data integration, etc. It supports keyword search and provides the "best-effort" operation based on query relevance ranking. These two prototype systems both do not distinguish the different stages of the lifecycle, but rather try to seamlessly integrate all of the stages of dataspace.

The Self-Organising Maps (SOM) (Guerrero 2012) method was used to screen the level of satisfaction of dialysis patients in the NephroCare network, which belongs to Fresenius Medical Care (FME), a global provider of dialysis services. SOM is a neural network model for clustering and projecting high-dimensional data into a low-dimensional space, and it could support the identification of potential improvements for specific patient groups by analysing data provided by a questionnaire. This method could preserve the topological relationships of original high-dimensional dataspace; therefore, its idea is a good inspiration for dataspace construction.

The above dataspace prototypes have more or less satisfied the specific demands of big data processing, such as semantic integration, pay-as-you-go, schema mapping, etc. However, most of them are limited by the coarse-grained architecture research and model construction, and rarely consider the issue of demand-oriented dynamic evolution in the modelling process of the dataspace. Meanwhile, these prototypes do not provide the data service features in the field of materials. Based on this previous work, we propose the concept and method of Virtual DataSpace (VDS) (Liu 2012). This technology could convert physical data into virtualisation processing, achieve the dynamic evolution of data and realise the individualised on-demand service based on data associate modelling. It is necessary to deeply research the modelling process and the evolutionary algorithm of VDS for big data processing and intelligent services in the field of materials engineering.

3 Virtual Dataspace Model

3.1 VDS Definition

VDS is a new scientific data service mode that faces the domain demands of an engineering expert system, especially the materials application demands. It can be defined as follows.

VDS denotes the sets of data, services and their relationships with the subject of user requirements by supporting virtualisation processing and dynamic evolution. VDS is composed of four-tuples, VDS = (S_{UR}, D_S, DR_S, S_S). S_{UR} is the subject of user requirements, which mainly refers to the demands of user; D_S is the data resource set; DR_S is the data relationship set among the data resources; S_S is the service resource set.

According to the definition of VDS, VDS means the collection of data resources and services that relate to the specific subject. It uses virtualisation technology and the dynamic (Saleheen, Lai 2013) evolution method to expand the data resources and service resources in accordance with the actual needs of subjects. From the basic structure on the relationships of subjects, data sets, and services in VDS (**Figure 1**), it is noted that the "subject" is also the user who uses the "virtual data" for their specific needs.

Considering a specific data item, VDS also could be described as an m-dimensional vector. VDS = (V_1, V_2, . . . , V_m), where V_i is an n-dimensional vector, $V_i = (P_{i1}, P_{i2}, \ldots, P_{in})$ and P_{ij} is an instance of a triple (subject, data set, service). Thus, VDS is an m*n-dimensional space as in Eq. (1).

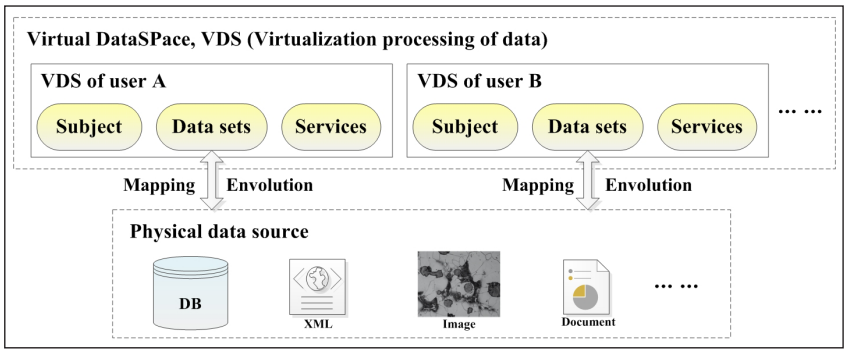

Figure 1: The basic structure of the relationships between subject, data sets, and services in VDS.

$$\begin{pmatrix} P_{11} & P_{12}\ldots P_{1j}\ldots & P_{1n} \\ P_{21} & P_{22}\ldots P_{2j}\ldots & P_{2n} \\ \ldots & \ldots\ldots\ldots & \ldots \\ P_{i1} & P_{i2}\ldots P_{ij}\ldots & P_{in} \\ \ldots & \ldots\ldots\ldots & \ldots \\ P_{m1} & P_{m2}\ldots P_{mi}\ldots & P_{mn} \end{pmatrix} \quad (1)$$

Describe the relationship of data, services and subjects in VDS as an m*n-dimensional matrix as in Eq. (2).

$$\mathrm{Re}la = \sum_{i=1}^{m}\sum_{j=1}^{n} R_{ij} \quad (2)$$

R_{ij} is a Boolean value. Consider the data, services and subjects as the basic element items, represented as E_k. When a correlation exists between two element items, for example, E_i is related to E_j, thus R_{ij} is assigned to 1; otherwise it is 0. The total number of relationships is m*n.

Around a particular subject such as "the automobile manufacturers need high hardness material for car board", according to the associated construction of this subject, the subject-specific VDS is obtained as in Eq. (3).

$$VDS_{sub} = \sum_{i=1}^{n}\sum_{j=1}^{m} P_{ij}R_{ij} = \begin{pmatrix} P_{1i} & \ldots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{m1} & \ldots & P_{mn} \end{pmatrix} * \begin{pmatrix} R_{1i} & \ldots & R_{1n} \\ \vdots & \ddots & \vdots \\ R_{m1} & \ldots & R_{mn} \end{pmatrix} \quad (3)$$

Assume that the number of R_{ij}, which is equal to 1, is N, then obtain the subject-related VDS as an N-dimensional vector, $VDS_{sub} = (P_1, P_2, \ldots, P_N)$, thus, VDS_{sub} is an m*n-dimensional space as in Eq. (4).

$$VDS_{sub} = \begin{pmatrix} sub_1 & data_1 & servise_1 \\ sub_2 & data_2 & servise_2 \\ \ldots & \ldots & \ldots \\ sub_N & data_N & servise_N \end{pmatrix} \quad (4)$$

Because VDS_{sub} is around the same subject, $sub_1 = sub_2 = \ldots = sub_N$. Accordingly, one could obtain the data and services that are of interest to the particular users, i.e., obtain a particular VDS based on the requirement of the subject. Therefore, VDS_{sub} also could be considered as a subset of VDS.

Based on the above theoretical analysis, the working process of VDS could be divided into three main stages (**Figure 2**). (1) In the initialisation phase, users have not accessed the data, thus there are no associated relationships between data and subjects. (2) In the use phase, gradually establish the associations between users and the data of interest along with the accessing and manipulating of data by the subjects. (3) In the evolution phase, join the users with their data of interest to form a dedicated space that also could be automatically optimised according to changes in the users' interests and requirements.

3.2 VDS Model Construction

The construction of VDS model mainly includes three steps, build semantic data view, establish user requirement model and match schema automatically.

The construction of the global semantic data view (Qin, Atluri 2009) is the primary condition of realising the automatic mapping between the local data sources and the global model. The global semantic view could be defined as, $GSV = (C_s, R_s, I_s, A_s, Ru_s)$, where C_s represents the domain core concept sets. R_s denotes the semantic relationship sets of domain concepts, i.e., $R_s = C_1{*}C_2{*}\ldots{*}C_n$. I_s are the instance sets. A_s denotes the axiom sets about the domain concepts and relationships. Ru_s represents the Horn rule sets, which support the discovery of domain implicit knowledge by rule-based reasoning.

To further increase the expression strength of an associated relationship between core concepts, we also include a certain number of constraint axioms into the global semantic view. The constraint axioms can

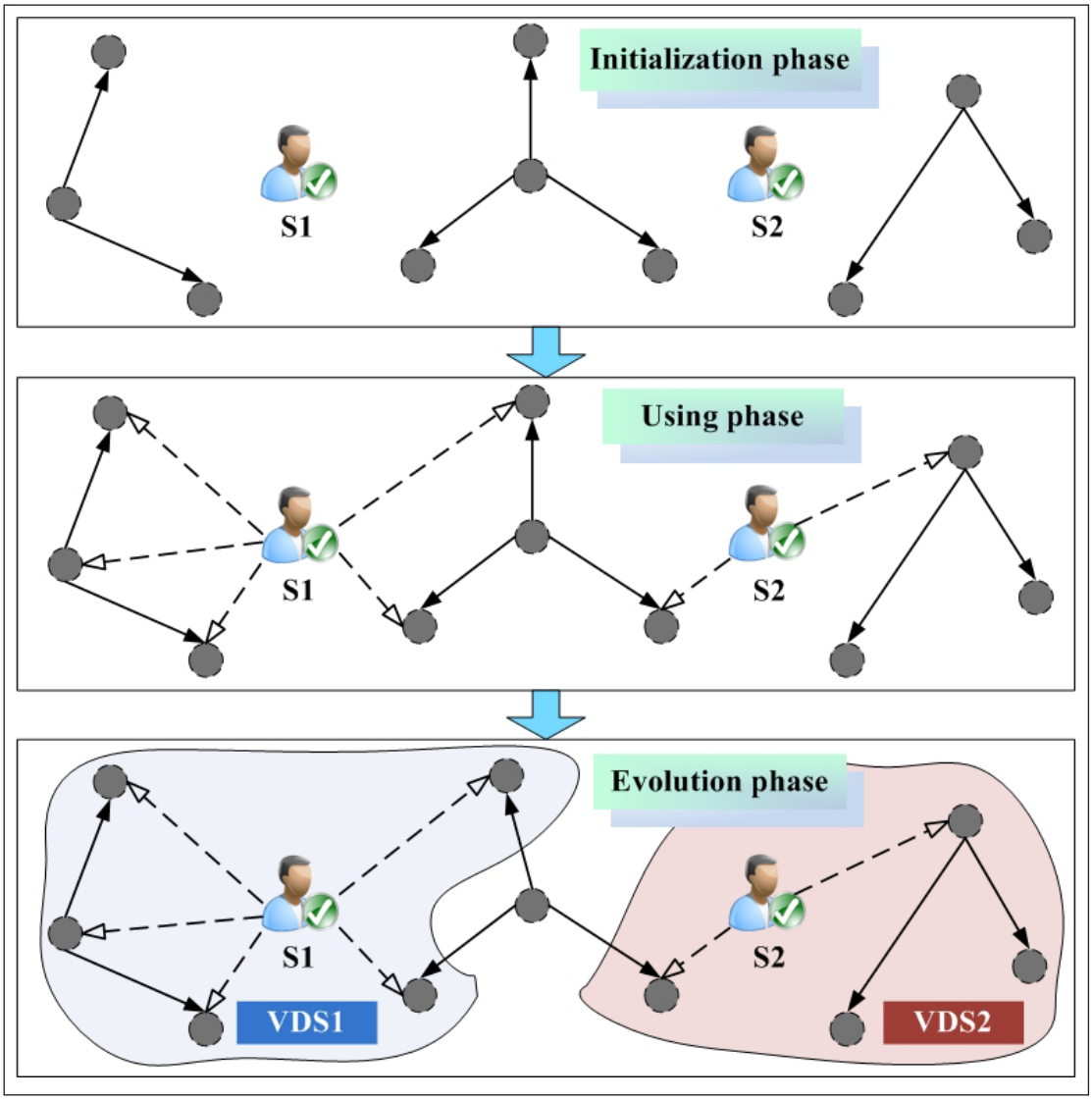

Figure 2: The working principle of VDS in different stages.

be classified as two types: the value constraint, which constrains the range of attribute, and the cardinality constraint, which constrains the quantity of value.

In addition, OWL DL (Sirin 2007) is also a good ontology description language that has a stronger semantic expression and reasoning function. Therefore, its constraint axioms also could be used to describe the global semantic view.

Considering the "subject" as the core concept of VDS, it is important for both the "data" and the "service". Thus, subject mainly refers to the user requirement. It is necessary to analyse the relationship between subject and the other elements in VDS, we find complex relationships among subjects and users. The user maps to different subjects according to his/her demands, whereas a specific subject could be applied to different users. Data items and services are organized by subjects in VDS. This management is evolved according to different requirements and using behaviour.

User Requirement Model (**Figure 3**) is defined as a set of four-tuples, URM = (C^l, R^l, W^l, T^l), where C^l denotes the core concept sets that the subject is interested in, $C^l = C_{key} + C_{temp}$, where C_{key} defines the keynote requirement and C_{temp} denotes the temporary requirement. R^l denotes the associated relationship of core concepts. W^l denotes the weight value of concept, i.e., the interesting degree of the subject for a core concept. T^l denotes the update time of W^l.

The schema mappings are built automatically among data items, data sources and global semantic view. It is constructed by a semantic query, which reinforced the relationships between semantic concept items. Ontology similarity is used in this method to achieve automatic schema mapping. This measurement could be defined as,

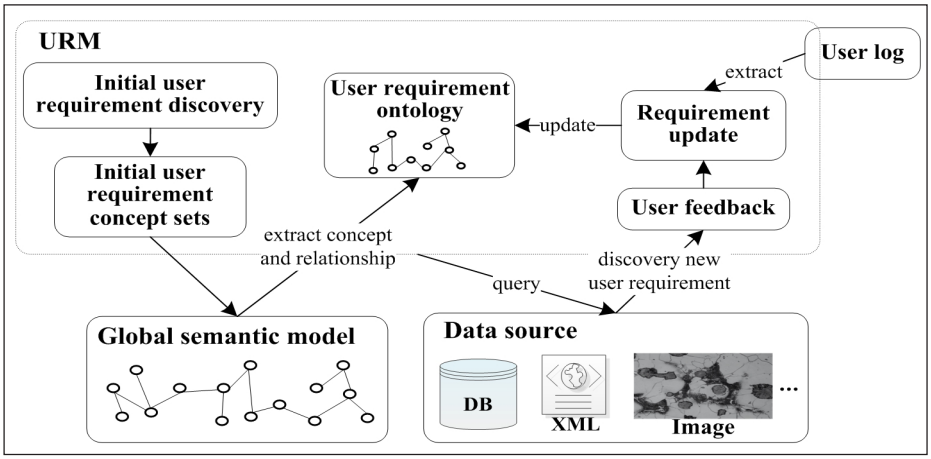

Figure 3: The user requirement model of VDS.

$$Sim(C_1,C_2)=\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}Sim(c_{1i},c_{2j})}{n*m} \tag{5}$$

where C_1 and C_2 are the semantic concept items. C_{1i} denotes the sub-concept in C_1 and C_{2j} denotes the sub-concept in C_2. If the attributes or the instances of concepts c_1 and c_2 are the same, the concepts c_1 and c_2 are the same. If the sub-concept (i.e., subclass) or parent concept (i.e., super class) of concepts c_1 and c_2 are similar, then the concepts c_1 and c_2 are similar. If all of the sub-concepts or brother concepts of c_1 are similar with the concept c_2, then the concepts c_1 and c_2 are similar.

Therefore this model only contains the data and services that are required by the subject. It adopts the manner of pay-as-you-go, thus the overhead of construction is quite small. This model is gradually improved along with the use of data by the subjects.

4 Model Evolution Based On Services

The changing demands in data services require the model evolution in VDS. The evolved model could help better adoption of VDS along with the service time and changing requirements. On one hand, the changes in services and user requirements from time to time lead to continuous changes in data items of VDS. On the other hand, the changes of semantic relationships and information in applications are changing due to the heterogeneous data sources.

The evolution of user requirements means the changes in interest areas of the subject that are embodied in the user feedback. The user feedback is mainly described as two parts. One is the feedback of the requirement model (i.e., the interest model), which supports the amendment of core concepts by subjects. Each subject could correct the interest model according to their own demands, and would not affect the use of VDS by other subjects. The other is the feedback of pattern matching results. It supports feedback correction for the results of pattern matching by subjects in the process of automatic matching and mapping between the local data source mode and the global mode.

In VDS, users feed their expectations and intentions to the system through the feedback instance. The user feedback instance could be described as follows.

The User Feedback Instance is a set of four-tuples, UFI = (AttV, R^G, Exp, Prov), where AttV is a set of key-value pairs that is composed of the attributes and the corresponding values, i.e., AttV = <att_i, v_i>, i ∈ [1,n]. R^G denotes a set of associated relationships in the global model. Exp is a Boolean value, which supports determining whether the attribute key-value pairs "AttV" could meet the user expectations. Prov denotes the provenance of feedback that could be specified by the user and could be automatically obtained from the candidate matching.

For example, $Prov_1$ = <'huserSpecified', M_D>, where 'huserSpecified' denotes that AttV is specified by the subject. M_D denotes that the attribute key-value pairs "AttV" is automatically obtained from the data source in the process of pattern matching. M_D is a set of candidate mappings, which supports retrieving the AttV from heterogeneous data sources, i.e., M_D = <m_1, m_2, . . . , m_k>.

Algorithm. RefineMappings (Map *Mi,* UFI *instan*)

```
Inputs Map: A set of candidate mappings
       UFI: A set of user feedback instances
Outputs Map: A set of refined mappings
Begin
1   If (Mi ≠ null){
2      S_Map = Mi;
3      O_Map;
4      Foreach O_L ∈ S_Map {
5         If (O_L ≠ null)
6            {Add <O_L, O_G> To O_Map}
7      }
8       AnnotateMappings(O_Map, UFI);
9       C_Map = CombineMappings(O_Map);
10      Return C_Map;
11  }
End
```

Table 1: The evolution algorithm of schema matching using the feedback instances.

Another example is UFI_2 = ({<Type, 'Nano Materials'>}, SPoSM, false, {m_2}). This feedback instance means that the type of 'Nano Materials' could not meet the needs of "Service Performance of Structural Materials (SPoSM)" according to the candidate matching "m_2". Actually, this is a 'nano-material' rather than a 'structural material'.

Table 1 illustrates the evolution algorithm of schema matching using the feedback instances. First, select a set of candidate mappings as input. Second, in the input mapping, automatically obtain the information about classes, attributes and relationships from the local mode and the global semantic view. Next, annotate the matching and mapping between the local data source mode and the global semantic view using the user feedback instances. Then, identify cases where matching or mapping does not meet the user requirement. Finally, construct the new mapping by correcting and merging the current mappings that meet the user requirement and return the refined mappings.

Overall, dynamic evolution is an important feature of VDS, which is different from the traditional approach of data management. Based on the research of the dynamic evolution mechanism, users could quickly acquire more efficient data management and the individualised intelligent services.

5 Experimental Results

5.1 Comparison with other models

As a new data logical organisation and data management method, VDS has its own characteristics. **Table 2** illustrates the characteristics comparison between VDS and traditional data management methods.

Through comparison with the traditional modes, VDS has the following significant features: (1) complex association with multi-source and heterogeneous data; (2) virtualisation processing; (3) pay-as-you-go, incremental improvements to the data model; and (4) on-demand service mode, dynamic evolution with the changes of user requirements. In short, VDS is a data service mode with the feature of "data first", "schema-later", and "best-effort".

Considering the big data management needs to quickly obtain the valuable information from the complex data, an intelligent service mainly involves getting the required knowledge with a high value density from the related data with a low value density. The technical characteristics of VDS could well satisfy the needs of big data processing and intelligent service. Therefore, VDS is a more optimised technical method, and is suitable for solving the various issues of intellectualised big data services.

	Traditional database	Semantic data integration	VDS
Data object	Data in relational database	All data	All data
Data mode	Relational model (Schema-First)	Ontology model (Schema-First)	Multiple models (Schema-Later)
Data type	Structured data (tables)	Structured data, Semi-structured data, Non-structured data	Structured data, Semi-structured data, Non-structured data
Data source	Single source, isomorphism	Multi-source, heterogeneous	Multi-source, heterogeneous
Data association	Simple association, structural stability	Complex association, structure is relatively stable	Complex association, dynamic evolution
Semantic	Without semantic	Pre-established semantic information	Gradually improved semantic information
Quality of data access	Accurate and complete results	Accurate and complete results	Currently optimal results (Best-effort)
Construction and services	First building, after use (Pay-before-you-go)	First building, after use (Pay-before-you-go)	Construction and optimisation with use (Pay-as-you-go)

Table 2: Characteristics comparison between VDS and traditional data management methods.

Model comparison	iDM (ETH Zürich)	UDM (University of Washington)	P-DM (Stanford University)	T-DM (Carleton University)	CSM (Renmin University of China)	VDM (USTB)
Data source	Centralised, heterogeneous data	Centralised, heterogeneous data	Distributed, heterogeneous data	Distributed, heterogeneous data	Centralised, heterogeneous data	Distributed, heterogeneous data
Model structure	Based on the graph	Based on the sort tree	Based on the probability mode	Based on the RDF	Based on the graph	Based on the ontology
Integration approach	Only database	Database, information extraction	Data integration	RDF	Association rules	Semantic integration
Applicable field	Personal Information Management (PIM)	Personal Information Management (PIM)	Not involved in the specific application areas	Not involved in the specific application areas	Personal Information Management (PIM)	The field of materials engineering
Uncertainty	Does not support	Does not support	Support	Does not support	Does not support	Support
Subject feature	Does not consider	Does not consider	Does not consider	Does not consider	Individual users as the core	Users in material field as the core
Applicability	Good query performance, and support the semantic	Query interface	Support the top-k sorting query results	The processing capability of query language is strong	Support the multi-faceted semantic queries	Diversified query strategy and strong semantic support

Table 3: The model comparison of dataspaces.

Although the current research about dataspace models has been studied, most of the applications are concentrated in the field of Personal Information Management (PIM). There are some generic models, but most did not give the application examples in specific areas. For the VDS proposed in this paper, we developed a "Materials Scientific Data Sharing Service Platform" based on the construction of a Materials Virtual DataSpace (MatVDS) to implement intelligent service applications in the field of materials engineering. Those with a focus on the field of materials engineering are very scarce. **Table 3** illustrates the comparison

Prototype system	Data type	Location of data source	Integration model	Type of schema integration	Processing of schema integration	Endpoints of schema matching	Processing procedure of mapping
MatVDS	Structured (stru), semi-structured (semi), unstructured (unst)	Local, distributed	Proprietary model	Union, merge	Automatic, manual	Source mode (sour) and integrated mode (inte)	Automatic
OrientSpace	stru, semi, unst	Local	Proprietary model	—	Automatic	—	Automatic
SEMEX	stru, semi, unst	Local, distributed	Proprietary model	Merge	Manual	sour & inte	Automatic
iMeMex	stru, semi, unst	Distributed	Proprietary model	Union	Automatic	sour & sour	Semi-automatic
PayGo	stru	Distributed	Proprietary model	Union	Automatic	sour & sour	Automatic
UDI	stru	Local	Proprietary model	Merge	Automatic	sour & sour, sour & inte	Automatic
Roomba	—	—	Universal model	Union	Automatic	sour & sour	Automatic
Quarry	semi	Local	Universal model	Union	Automatic	—	—
Cimple	stru	Distributed	Proprietary model	Merge	Manual	sour & sour, sour & inte	Semi-automatic
CopyCat	stru, semi	Distributed	Proprietary model	Union	Semi-automatic	sour & sour	Semi-automatic
Octopus	stru, semi	Distributed	Proprietary model	Merge	Semi-automatic	sour & inte	Semi-automatic

Table 4: The comparison of MatVDS and other dataspace systems in the initialisation phase.

of dataspace models from various aspects such as the data source, the model internal principle, the integration method, the field of application, and so on. This comparison shows differences in the construction method and function of these models. Unlike the needs of personal data management, VDM is applied to the engineering field; therefore, it has a complex construction and implementation. VDM fully considers the user demands and emphasises the central position of the subject. Furthermore, VDM uses the ontology as the realisation technology of the model; thus, it has a unique advantage in resolving the problem of semantic heterogeneity.

MatVDS has realised the automatic construction of a virtual dataspace for the application of materials engineering. **Table 4** and **Table 5** illustrate the comprehensive comparison of MatVDS and other dataspace systems in the process of automatic construction. The comparison shows that MatVDS focuses on complex and distributed data sources in the field of materials. The integration approach of MatVDS is more flexible. MatVDS supports the combination of automatic and manual, i.e., pre-completed part of the integration work before use; and gradually improves it using the method of "pay-as-you-go" in the use process. MatVDS provides a diversified query strategy, and the query service system is a better fit for the actual needs of the engineering field. The automatic construction of a dataspace system is difficult and complex. The process of pursuing automation inevitably results in the loss of accuracy to some degree. These losses are acceptable in MatVDS and future research will focus on how to improve the accuracy of automatic construction.

Evolution is the most important problem to be solved in dataspace research. It is also the main basis for measuring the practicability of dataspace systems. **Table 6** illustrates the comparison of MatVDS and other dataspace systems from the aspect of evolution. Compared with other dataspace systems, MatVDS has two types of feedback mechanisms, and makes the user requirement the centre of focus; thus, VDS realised dynamic evolution in the process of schema matching and mapping.

5.2 Case study in Materials Scientific Data Services

A "Materials Scientific Data Sharing Service Platform" is constructed, as Materials Virtual DataSpace (MatVDS) (**Figure 4**) to implement intelligent service applications in the field of materials engineering. This platform integrated the massive, distributed and heterogeneous data resources under twelve categories: material basis, non-ferrous & alloy materials, ferrous materials, composite materials, organic polymer materials, inorganic non-metallic materials, information materials, energy materials, biomedical materials, natural materials & products, building materials, and road traffic materials.

Prototype system	User interest and behaviour	Load	Query type	Query result
MatVDS	Consider the user interests and behaviour habits	Pre-integrated, and dynamic evolution	Keyword query, structured query, visualisation query	Union
OrientSpace	Consider the behaviour characteristics and habits	Dynamic evolution	Keyword query, structured query	Union
SEMEX	Does not consider	Run-time integration	Keyword query, structured query	Merge
iMeMex	Does not consider	Run-time integration	Keyword query, structured query	Union
PayGo	Does not consider	Run-time integration	Keyword query	Union
UDI	Does not consider	Run-time integration	Structured query	Merge
Roomba	Does not consider	Pre-integrated	Keyword query, structured query	Merge
Quarry	Does not consider	Run-time integration	Structured query	Union
Cimple	Does not consider	Run-time integration	Keyword query, structured query	Merge
CopyCat	Does not consider	Pre-integrated	Visualisation query	Union
Octopus	Does not consider	Pre-integrated	Keyword query	Merge

Table 5: The comparison of MatVDS and other dataspace systems in the use phase.

Prototype system	Use output	Evolution method	Evolution processing
MatVDS	Sort results, browsing, sources	A variety of user feedback instances, explicit and implicit	Matching and mapping, the user requirement model
OrientSpace	Sort results, browsing, sources	User behaviour, implicit	Core dataspace, task space, associated information
SEMEX	Results, browsing	—	—
iMeMex	Results, browsing, sources	—	—
PayGo	Sort results	User feedback, explicit	Sorting query results
UDI	Sort results	—	—
Roomba	Results	User feedback, explicit	Matching
Quarry	Results, browsing	User feedback, explicit	Matching
Cimple	Sort results, browsing	User feedback, explicit	Matching
CopyCat	Results, sources	User feedback, explicit	Integration mode, mapping
Octopus	Results	User feedback, explicit	Integration mode, mapping

Table 6: The evolutionary comparison of MatVDS and other dataspace systems.

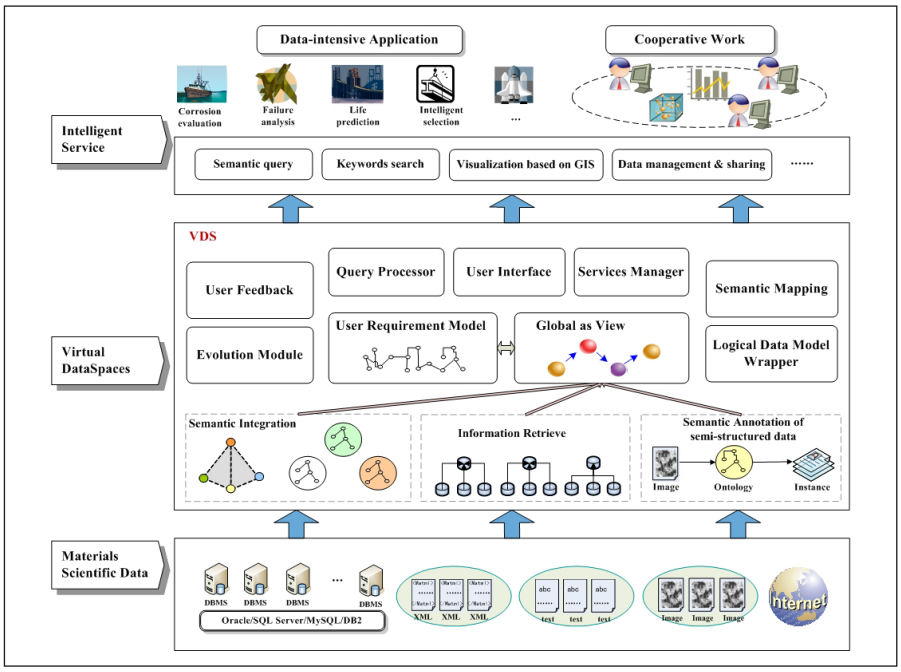

Figure 4: The system architecture of MatVDS.

MatVDS has the characteristics of "Data as a Service, DaaS", virtualisation processing, high-quality services, scalability, high reliability, and so on. It collected in total nearly 500,000 data resource items and continues to grow rapidly. The global semantic view of MatVDS has been built based on the domain core concepts and hierarchical structure in the field of materials engineering. This includes materials data from various perspectives, such as the materials categories, the basic features, the organisational structure, the chemical composition, the processing technology, the application performance, and so on.

Relationships of the users, data services and subjects are described by MatVDS. A specific dataspace is built for every user according to their requirements, which includes user preferences, data of interest and semantic relations of big materials scientific data.

The diversified and personalised intelligent services are achieved by the VDS model based on its evolution in the field of materials engineering (see **Figure 5**). Intelligent retrieval is the core and basis of these intellectualised data services. The flexible services are modularization in MatVDS based on the intelligent retrieval.

Because of the semantic extraction, relevant knowledge and data sets are provided to users that may interested in. For example, the **Figure 6** shows the results of the semantic search for query "corrosion-resistance". One result shows "GB/T 18982-2003", which belongs to the ferrous materials category, and stored as a "pdf" file. The result of "X2CrTi12" belongs to the ferrous materials category, and stored in both semi-structured data XML files and non-structured data chart files. While the result of "EVA" belongs to the energy materials category, which stored in databases as table files. "Furan resin" belongs to the organic polymer material category and stored as image files. Therefore, it proves that we could quickly and accurately acquire the relevant data services based on the domain knowledge.

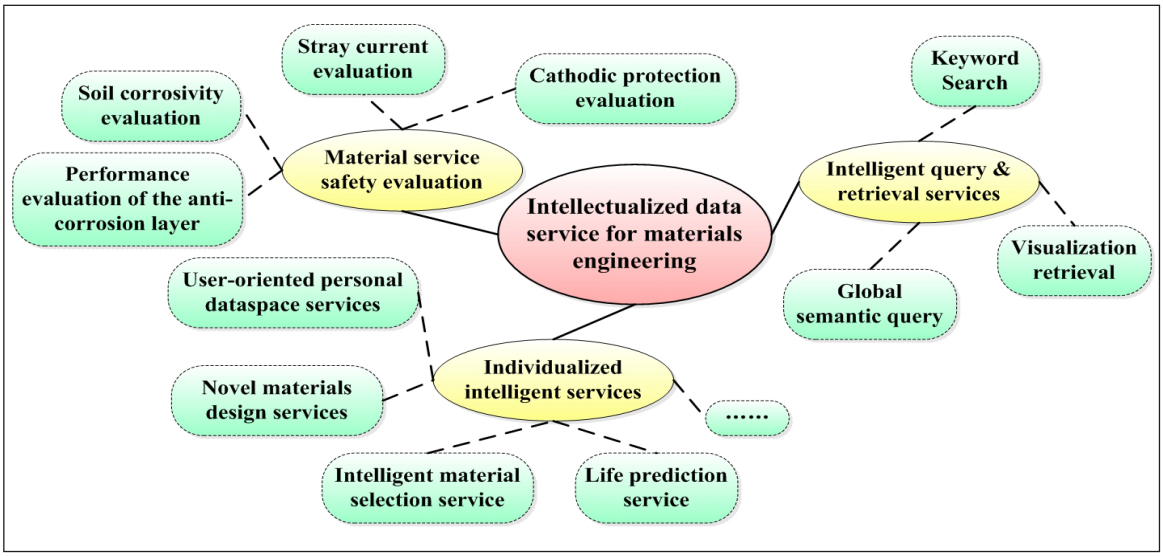

Figure 5: The collection of intelligent services in MatVDS.

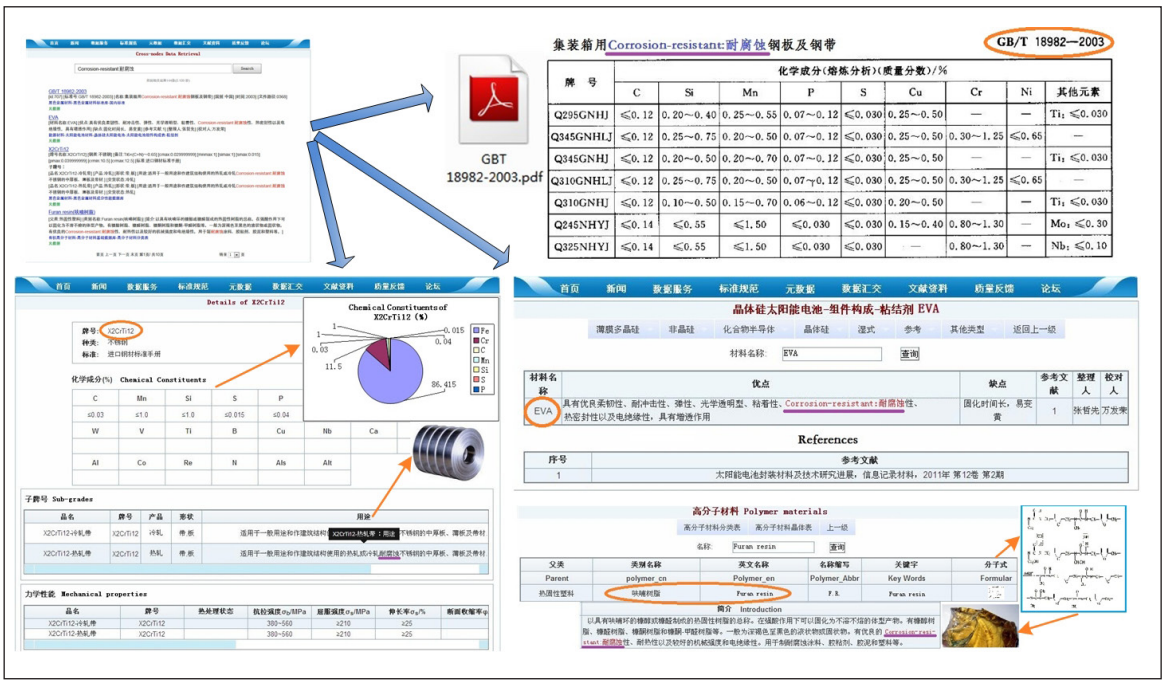

Figure 6: The demand-oriented data service instance in MatVDS.

6 Conclusions and Future Work

This paper proposed the model of VDS for materials scientific big data management, sharing and services in the field of materials engineering. The conceptual model and theories are introduced for the dataspace model.

And the automatic construction is also described to implement the VDS model. The VDS could also be evolved automatically to adopt the changing requirements of the users and service demands. Comparisons with other dataspace models are shown in the experimental results. And MatVDS is implemented for supporting intelligent services in materials engineering field. The results shows the efficiency of the model in applying intelligent services for the specific domain.

The future work of VDS includes improving the optimized method for automatic schema matching and mapping, developing better evolutionary algorithms for VDS evolution, and extending the scope of intelligent services in the field of materials engineering.

Acknowledgements

This work was supported by the R&D Infrastructure and Facility Development Program of China (Grant No. 2005DKA32800), the 2012 Ladder Plan Project of Beijing Key Laboratory of Knowledge Engineering for Materials Science (Grant No. Z121101002812005), the Key Science–Technology Plan of the National Twelfth Five-Year-Plan Project of China (Grant No. 2011BAK08B04), the Fundamental Research Funds for the Central Universities (No. FRFTP-12-079A) and the National Key Basic Research and Development Program (973 Program) of China (Grant No. 2013CB329601).

Competing Interests

The authors declare that they have no competing interests.

References

Baladron, C, Aguiar, J M, Carro, B, et al. 2012 Framework for intelligent service adaptation to user's context in next generation networks. *IEEE Communications Magazine, 50*(3): 18–25. DOI: http://dx.doi.org/10.1109/MCOM.2012.6163578

Ballings, M and **Poel, D V** 2012 Customer event history for churn prediction: How long is long enough? *Expert Systems with Applications, 39*(18): 13517–13522. DOI: http://dx.doi.org/10.1016/j.eswa.2012.07.006

Blunschi, L, Dittrich, J P, Girard, O R, et al. 2007 A dataspace odyssey: The iMeMex personal dataspace management system. *CIDR*: 114–119.

Cafarelia, M J, Halevy, A and **Khoussainova, N** 2009 Data integration for the relational web. *VLDB Endowment, 2*(1): 1090–1101. DOI: http://dx.doi.org/10.14778/1687627.1687750

Cai, H, Hu, X, Lv, Q and **Cao, Q** 2009 A novel intelligent service selection algorithm and application for ubiquitous web services environment. *Expert Systems with Applications, 36*(2): 2200–2212. DOI: http://dx.doi.org/10.1016/j.eswa.2007.12.071

Cuzzocrea, A, Song, I Y and **Davis, K C** 2011 Analytics over large-scale multidimensional data: the big data revolution. In: *ACM 14th international workshop on Data Warehousing and OLAP (DOLAP'11)*, pp. 101–104. DOI: http://dx.doi.org/10.1145/2064676.2064695

Dittrich, J P and **Salles, M A V** 2006 iDM: A unified and versatile data model for personal dataspace management. In: *32nd international conference on Very Large Data Bases (VLDB)*. Seoul, Korea, pp. 367–378.

Dong, X and **Halevy, A Y** 2005 A platform for personal information management and integration. In: *PhD Workshop of VLDB*. Trondheim, Norway, pp. 26–30.

Dong, X, Halevy, A Y and **Madhavan, J** 2005 Reference reconciliation in complex information spaces. In: *SIGMOD/PODS*. Baltimore, USA, pp. 85–96. DOI: http://dx.doi.org/10.1145/1066157.1066168

Dong, X, Halevy, A Y and **Yu, C** 2009 Data integration with uncertainty. *The VLDB Journal, 18*(2): 469–500. DOI: http://dx.doi.org/10.1007/s00778-008-0119-9

Dong, Y L, Shen, D R, Kou, Y and **Nie, T Z** 2009 Data Organization Model and Relationship Discovering Model in Dataspace. *Journal of Computer Research and Development, 46*(z2): 191–199.

Franklin, M, Halevy, A and **Maier, D** 2005 From databases to dataspaces: a new abstraction for information management. *ACM Sigmod Record, 34*: 27–33. DOI: http://dx.doi.org/10.1145/1107499.1107502

Guerrero, J D M, Marcelli, D, Soria-Olivas, E, et al. 2012 Self-Organising Maps: A new way to screen the level of satisfaction of dialysis patients. *Expert Systems with Applications, 39*(10): 8793–8798. DOI: http://dx.doi.org/10.1016/j.eswa.2012.02.001

Howe, B, Maier, D, Rayner, N and **Rucker, J** 2008 Quarrying dataspaces: Schemaless profiling of unfamiliar information sources. In: *24th International Conference on Data Engineering Workshop (ICDEW)*, Cancún, México, pp. 270–277. DOI: http://dx.doi.org/10.1109/icdew.2008.4498331

Howe, D, Costanzo, M, Fey, P, et al. 2008 Big data: The future of biocuration. *Nature, 455*: 47–50. DOI: http://dx.doi.org/10.1038/455047a

Hu, C J, Li, Y, Cheng, X and **Liu, Z Y** 2016 A Virtual Dataspaces Model for large-scale materials scientific data access. *Future Generation Comp. Syst., 54*: 456–468. DOI: http://dx.doi.org/10.1016/j.future.2015.05.004

Ives, Z, Knoblock, C, Minton, S, et al. 2009 Interactive data integration through smart copy \& paste. In: *CIDR*. Asilomar, USA, pp. 299–310.

Jeffery, S R, Franklin, M J and **Halevy, A Y** 2008 Pay-as-you-go user feedback for dataspace systems. In: *SIGMOD/PODS*. Vancouver, Canada, pp. 847—860. DOI: http://dx.doi.org/10.1145/1376616.1376701

Kwon, O and **Sim, J M** 2013 Effects of data set features on the performances of classification algorithms. *Expert Systems with Applications, 40*(5): 1847–1857. DOI: http://dx.doi.org/10.1016/j.eswa.2012.09.017

Lee, M Y, Lee, A S and **Sohn, S Y** 2013 Behavior scoring model for coalition loyalty programs by using summary variables of transaction data. *Expert Systems with Applications, 40*(5): 1564–1570. DOI: http://dx.doi.org/10.1016/j.eswa.2012.08.073

Li, Y K and **Meng, X F** 2008 Research on personal dataspace management. In: *ACM SIGMOD/PODS Conference (IDAR)*. Vancouver, Canada, pp. 7–12. DOI: http://dx.doi.org/10.1145/1410308.1410311

Li, Y K and **Meng, X F** 2009 Exploring Personal CoreSpace for DataSpace Management. In: *Semantics, Knowledge and Grid (SKG)*. Zhuhai, China, pp. 168–175. DOI: http://dx.doi.org/10.1109/skg.2009.46

Li, Y K, Meng, X F and **Kou, Y B** 2009 An Efficient Method for Constructing Personal DataSpace. In: *Web Information Systems and Applications Conference (WISA)*. Xuzhou, China, pp. 3–8. DOI: http://dx.doi.org/10.1109/wisa.2009.39

Li, Y K, Meng, X F and **Zhang, X Y** 2008 Research on Dataspace. *Journal of Software, 19*(8): 2018–2031. DOI: http://dx.doi.org/10.3724/SP.J.1001.2008.02018

Liu, X, Zhang, X and **Li, W** 2012 Geographic information intelligent service chain construction and application in Taiwan Strait. In: *8th International Conference on Information Science and Digital Content Technology (ICIDT)*, pp. 251–254.

Liu, Z Y, Hu, C J, Li, Y and **Hu, J Y** 2012 DSDC: a domain scientific data cloud based on virtual dataspaces. In: *26th IEEE International Parallel and Distributed Processing Symposium Workshops \& PhD Forum (IPDPSW)*, pp. 2176–2182. DOI: http://dx.doi.org/10.1109/ipdpsw.2012.269

Lynch, C 2008 Big data: How do your data grow? *Nature, 455*: 28–29. DOI: http://dx.doi.org/10.1038/455028a

Madhavan, J, Jeffery, S R, Cohen, S, et al. 2007 Web-scale data integration: You can only afford to pay as you go. In: *Third Biennial Coneference on Innovative DataSystems Research (CIDR)*. Asilomar, USA, pp. 342–350.

Pradhan, S 2007 Towards a novel desktop search technique. *Database and Expert Systems Applications, 4653*: 192—201. DOI: http://dx.doi.org/10.1007/978-3-540-74469-6_20

Qin, L and **Atluri, V** 2009 Evaluating the validity of data instances against ontology evolution over the semantic web. *Information and Software Technology, 51*(1): 83–97. DOI: http://dx.doi.org/10.1016/j.infsof.2008.01.004

Qu, Z 2012 Semantic Processing on Big Data. *Intelligent and Soft Computing, 129*: 43–48. DOI: http://dx.doi.org/10.1007/978-3-642-25986-9_7

Saema, A D, Dong, X and **Halevy, A Y** 2008 Bootstrapping pay-as-you-go data integration systems. In: *SIGMOD/PODS.* Vancouver, Canada, pp. 861–874.

Saema, A D, Dong, X and **Halevy, A Y** 2009 Data modeling in dataspace support platforms. *Conceptual Modeling: Foundations and Applications, 5600*: 122–138.

Saleheen, S and **Lai, W** 2013 User centric dynamic web information visualization. *Science China Information Sciences, 56*(5): 1–14. DOI: http://dx.doi.org/10.1007/s11432-013-4871-0

Simonet, A, Fedak, G and **Ripeanu, M** 2012 Active Data: A Programming Model for Managing Big Data Life Cycle. In: Grid'5000 Collaboration, pp. 1—26.

Singh, M and **Jain, S K** 2011 A Survey on Dataspace. *Advances in Network Security and Applications, 196*(3): 608–621. DOI: http://dx.doi.org/10.1007/978-3-642-22540-6_59

Sirin, E, Parsia, B, Grau, B C, et al. 2007 Pellet: A practical OWL-DL reasoned. *Web Semantics: Science, Services and Agents on the World Wide Web, 5*(2): 51—53. DOI: http://dx.doi.org/10.1016/j.websem.2007.03.004

Toffler, A 1980 The third wave: The classic study of tomorrow. USA: Bantam.

Wang, S, Wang, H, Qin, X and **Zhou, X** 2011 Architecting big data: challenges, studies and forecasts. *Chinese Journal of Computers*, *34*(10): 1741–1752. DOI: http://dx.doi.org/10.3724/SP.J.1016.2011.01741

Yeh, K C, Chen, R S and **Chen, C C** 2011 Intelligent service-integrated platform based on the RFID technology and software agent system. *Expert Systems with Applications*, *38*(4): 3058–3068. DOI: http://dx.doi.org/10.1016/j.eswa.2010.08.096

Zhong, M, Liu, M C and **Qian, C** 2008 Modeling heterogeneous data in dataspace. In: *Information Reuse and Integration (IRI)*. Las Vegas, USA, pp. 404–409. DOI: http://dx.doi.org/10.1109/iri.2008.4583065

Zhuge, H 2004 Resource space model, its design method and applications. *Journal of Systems and Software*, *72*(1): 71–81. DOI: http://dx.doi.org/10.1016/S0164-1212(03)00058-X

How to cite this article: Li, Y and Hu, C 2016 Process Materials Scientific Data for Intelligent Service Using a Dataspace Model. *Data Science Journal,* 15: 7, pp. 1–17, DOI: http://dx.doi.org/10.5334/dsj-2016-007

Submitted: 28 April 2016 **Accepted:** 02 June 2016 **Published:** 08 July 2016

Copyright: © 2016 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See http://creativecommons.org/licenses/by/4.0/.

]u[*Data Science Journal* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS