

Research Article

Atmospheric Lidar Data Storage Model Based on Ontology

Hao Chen,¹ Dengxin Hua,² Yikun Zhang,¹ and Qing Yan²

¹School of Computer Science & Engineering, Xi'an University of Technology, Xi'an 710048, China

²School of Mechanical & Instrumental Engineering, Xi'an University of Technology, Xi'an 710048, China

Correspondence should be addressed to Hao Chen; haozi2638@163.com

Received 19 September 2016; Accepted 26 January 2017; Published 6 March 2017

Academic Editor: Alex P. James

Copyright © 2017 Hao Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ontology is an effective method to solve the problem of heterogeneous data in lidar measurements. Due to complexity and diversity of data structure, traditional method of ontology storage cannot be directly applied to lidar data. In this work, we proposed a novel ontology storage model based on the object-oriented data model, in which the mapping mechanism was established from ontology of lidar data to the object-oriented data model. A new storage model of lidar data is then obtained by a combination of the characteristics of lidar data and the syntax of OWL DL. Compared to the traditional method of ontology storage, we believe that the new storage model can better serve the sharing of lidar data.

1. Introduction

Air pollutions, such as Acid rain, haze, and light chemical smoke, have become increasingly prominent in China, which has been considered as a serious threat to public health and environmental safety. In order to better understand the pollutions, many tools have been developed to monitor the environment. Lidar is an active instrument for vertical structure of atmospheric profiling with high accuracy and high spatial and temporal resolution. Recently, a number of regional atmospheric monitoring networks have been built based on lidar [1–4]. Analysis of lidar data from every monitored site is the key to investigating the regional air quality and temporal and spatial distribution of pollutants and to forecasting the emergency incidents, such as dust storms, forest fires, and volcanic eruptions [5–7].

Lidar observations have been carried out by many independent institutes, which leads to inconsistency of standards in terms of the protocol of data acquisition and data storage. Current lidar data structures which vary degrees of heterogeneity in structure and semantics are hard to be shared without uniform data standard. The existing differences include the meaning of various concepts and the units of collected information. With the abilities of expressing and reasoning in semantic, ontology which has been widely used in data sharing and integration can describe the inner

relationships between various parameters' concepts or terms and reveal the hidden relationships by logical reasoning [8]. The ontology storage model of lidar data could be the prerequisite for exploring its huge quantity [9].

At present, there are four storage methods for ontology [10]: memory-based, plain text, specific management tools [11], and database [12, 13]. Methods of memory-based, plain text, and specific management have limitations in efficiency, analysis management, universality, and extensibility. The database storage model can further divide into three methods: relational database method, Native XML method, and object-oriented database method [14]. Among them, relational database method is the most popular and matured method in ontology storage. The limitation of this method is that database's pattern does not match ontology and has poor extensibility and semantic support. The Native XML method has better expansibility but lower efficiency. Object-oriented database method has higher semantic integrity and query efficiency but lacks the ability of storing the complex relationship between ontology's concepts [15].

The lidar data have complex structures [16, 17] and mass storage [18] because the lidar data contained many atmospheric parameters with strong correlations. The above-mentioned storage methods have not been applied successfully to atmospheric lidar data storage. Based on the study of the relationship between lidar data and ontology structure,

we proposed and developed a mapping mechanism for lidar data ontology and object-oriented storage model.

In our previous work, the representation method and mapping mechanism of OODM of atmospheric lidar data were proposed and show that the OODM for lidar data is effective [19]. In present work, the details of database state of lidar data's OODM, the mapping mechanism of lidar data ontology to OODM, the data storage mode, and structure based on the OODM and the detailed process of the experiment are given.

2. Related Technologies

Ontology consisting of three elements, individual (instance), class (concept), and attribute [20], is a description of the concepts and relationships. The concept is the main part of the ontology, and the instance is the embodiment of the concept. Both the concept and the instance are the entity elements of the ontology, while the attributes are the descriptive elements.

Object-oriented data model (OODM) composed of object, class, and inheritance concept has been widely used in data modeling due to its flexibility in expressing the objects. Class describes a set of common characteristics of objects, and it is considered as a collection of objects with the same characteristics, where each object has the same properties and operations. Class is also associated with each other by inheritance, overwrite, and relationship [21].

The structure of ontology is similar to object-oriented data model in many respects. Concepts and *attributes* of ontology can be described and expressed with the classes and its *attributes*. Instances can be saved as object. Axiom and function of ontology can be simulated by the methods of OODM. Object-oriented method considers everything as a set of various objects. The complex objects can be decomposed into a combination of simple objects. Similarly, any complex knowledge of the ontology can be decomposed into a combination of basic concepts.

3. Lidar Data Ontology

The definitions of the lidar data's ontology and the object-oriented model are given as follows.

Definition 1. A lidar data ontology \mathbf{O} consists of five elements $\{C, R, A, I, X\}$, where C is a set of lidar data's concepts; R is a set of relationships; A is a collection of attribute sets for each concept and relationship; I is a set of lidar data's instances and X is a set of axioms [22].

Each concept c_i , where $c_i \in C$, represents a set of the same kind of objects and can be described by the same set of attributes, denoted by $A^C(c_i)$. Each relationship $r_i(c_p, c_q)$, then $r_i \in R$, represents a binary association between concepts c_p and c_q , and the instances of such a relationship are pairs of (c_p, c_q) . The attributes of r_i are denoted by $A^R(r_i)$. Each axiom in X is either a constraint on the attribute values of concept and relationship or a constraint on the relationships between

concepts objects. Each constraint can be expressed as a prolog (rule).

Definition 2. An object-oriented data model (OODM) of lidar data is a finite set of class declarations. Each class describes a series of atmospheric parameter objects with common attributes and methods. The OODM is a tuple $S = \{Cs, As, Ds\}$, where Cs is a finite set of classes; As is a finite set of attributes; Ds is a finite set of class declarations. Ds contains exactly one declaration. The definition of declaration is defined as follows:

$$\text{Class } C \text{ is-a } C_1, \dots, C_k \text{ type-is } T,$$

where T denotes a type expression which was built based on the following syntax:

$$\begin{aligned} T &\rightarrow C| \\ &\text{Union } T_1, \dots, T_k \text{ End}| \\ &\text{Set of } T| \\ &\text{Record } A_1 : D_1, \dots, A_k : D_k \text{ End} \end{aligned}$$

The part of *is-a* is optional, which represents class inheritance, for example, specified inclusion within the sets of instances of the involved classes, while the part of *type-is* which can be used to assign objects of the instances of C by the type expression T restricts the structures of the classes. *Union*, *...*, *End* represents summary between an abstract class and a series of object classes.

Definition 3. The database instance J is a database state of lidar data's OODM. The instance J is defined by the notions of object identifier and value. Given finite atomic domains D_1, D_2, \dots, D_n , $n \geq 1$, then $D = D_1 \cup D_2 \cup D_3 \cup \dots \cup D_n$, $D_i \cap D_j = \emptyset$, $v_d \in D_i$, $i \neq j$. v_d is atomic value.

Given an infinite set ID , set V of value, and each $id \in ID$ then (id, v) is an object. Let $o = (id, v)$, then $identity(o) = id$, $value(o) = v$. For $O = \{o_1, o_2, \dots, o_k\}$, then $O = ID \times V$. $[[a_1 : v_1, a_2 : v_2, \dots, a_k : v_k]] \in V$, $a_i \in A$, $v_i \in V$, $1 \leq i \leq k$.

A database instance J is constituted by O^J, π^J, ρ^J , where O^J is a finite set of object identifiers, π^J is a mapping assigned to each class in C_s to a subset of O^J , denoted by $C \in C_s$, $\pi_i^J(C) \subseteq O^J$, and ρ^J is a mapping assigned a value in V to each object in O^J , denoted by $OID_k \in OID$, $\rho_i^J(OID_k) \subseteq V$.

4. The Mapping Mechanism of Lidar Data Ontology to OODM

As the definitions given in Section 3, ontology is constituted of structure and instance, while an object-oriented database is constituted of OODM and database instance. They share some commons in structure; for example, both of them express the concepts of the real world based on class concepts and relationships; both of them have also class and the correspondence properties.

The metamodel elements of ontology can be transformed into the metamodel elements of the OODM in structural

and instance. In order to fully support the object-oriented features, ontology concepts of lidar data have to be abstracted to core class, and ontology relationships would be abstracted to relationship class. The core class stores simple and fixed concepts with few correlation attributes. The relationship class stores relationships between different concepts and the relationship attributes of ontology [23].

Let $S = \{Cs, As, Ds\}$, which is the corresponding OODM of lidar data's ontology O_S , and φ is the mapping rule of structure transformation; thus $S = \varphi(O_S) = \{Cs, As, Ds\}$. The mapping rules are described as follows:

- (a) For each $c_i \in C$ in O_S , it can be mapped to the class C_o of the OODM, $C_o \in Cs$, $\varphi(c_i) \in Cs$.
- (b) For each $r_i \in R$ in O_S , it can be mapped to the class R_o of the OODM, $R_o \in Cs$, $\varphi(r_i) \in Cs$.
- (c) For each attribute set $A^C(c_i)$ of c_i in O_S , $c_i \in C$, it can be mapped to the attribute A_i of class C in the OODM, $\varphi(A^C(c_i)) = A_i \in As$.
- (d) For each attribute set $A^R(r_i)$ of r_i in O_S , $r_i \in R$, it can be mapped to the attribute A_i of class C in the OODM, $\varphi(A^R(r_i)) = A_i \in As$.
- (e) For each expression axiom of c_i in O_S , $c_i \in C$, it can be mapped to the class C 's declaration of the OODM: *Class C is-a C_1, \dots, C_k or Class C type-is Union T_1, \dots, T_k End.*
- (f) For each data type definition of c_i in O_S , $c_i \in C$, it can be mapped to the class C 's declaration of the OODM: *Class C type-is T End.*
- (g) For each data attribute axiom of c_i in O_S , $c_i \in C$, it can be mapped to the class C 's declaration of the OODM: *Class C type-is Record $A_1 : D_1, \dots, A_k : D_k$ End.*
- (h) For each data attribute axiom of c_i in O_S , $c_i \in C$, it can be mapped to the class C 's declaration of the OODM: *Class C is-a C_1, \dots, C_k type-is Record A: Set of T End.*

We supposed that the instance of the ontology O_S and the data instance of the corresponding OODM are derived as follows:

- (a) Each individual object identifier and class identifier in O_S can be mapped to the object and class's instance of the OODM.
- (b) Each individual data type identifier and object type identifier in O_S can be mapped to the data type attribute and object type attribute.
- (c) The individual axiom *Individual(o type(C))* of O_S can be mapped to the class declaration of the OODM's instance.
- (d) The individual axiom of O_S : *Individual(value(R_1, o_1), ..., value(R_n, o_n), value(U_1, v_1), ..., value(U_n, v_n)))* can be mapped to the class declaration of the OODM's instance $o : [A_1 : V_1, \dots, A_n : V_k]$.

The mapping algorithm from the lidar data ontology to the OODM is given by OWL DL syntax in Tables 1 and 2.

TABLE 1: Identifier mapping algorithm.

Syntax of ontology O_S	Mapping method
$c_i \in C$	Add $\varphi(c_i)$ to Cs
$r_i \in R$,	Add $\varphi(r_i)$ to Cs
$c_i \in C, A^C(c_i) \in A$	Add $\varphi(A^C(c_i))$ to As
$r_i \in R, A^R(r_i) \in A$	Add $\varphi(A^R(r_i))$ to As
$C \subseteq C_1 \cup C_2 \cdots \cup C_k$	Create(Class C type-is Union T_1, \dots, T_k) add to Ds
$C \subseteq C_1 \cap C_2 \cdots \cap C_k$	Create(Class C is-a C_1, \dots, C_k) add to Ds
$type(C)$	Create(Class C type-is T) add to Ds
$U \subseteq C, U \subseteq U_i$ or $\geq 1U \subseteq C_i$ or $T \subseteq \forall U.D_i$ or $T \subseteq \leq 1U$	Create(Class C type-is Record $A_1 : D_1, \dots, A_k : D_k$ End) add to Ds
$R \subseteq C, R \subseteq U_i$ or $\geq 1R \subseteq C_i$ or $T \subseteq \forall R.C_i$ or $R = (R_0)$	Create(Class C type-is Record $A : Set$ of T) add to Ds

TABLE 2: Instance mapping algorithm.

Syntax of ontology O_S	Mapping method
$o : C_1$	Create $o : type(C)$
$(o, o_i) : R_i$ or $(o, v_i) : U_i$, $R_i, U_i \in C, o_i \in O_S$	Create $o = \langle a_1 : v_1, a_2 : v_2, \dots, a_k : v_k \rangle, a_i \in A, v_i \in V$
$o_i \in O_S, o_1 = o_2 = \dots = o_k$, $o_i \neq o_j$	Create o add instance(S)

5. The Storage Mode Based on the OODM

Currently, the object-oriented database management system has implemented most of the features of the OODM; however, it lacks the mathematical theory and is relatively inefficient compared to the relational database system with a rigorous mathematical foundation, which was based on the relational algebra theory and set theory [24]. To further improve query efficiency of lidar data ontology, we proposed a storage model based on the relational database.

The data table is built based on the vocabulary of the OWL and Definition 3 given in Section 3. The table *Class* contains the information of ontology about the OODM, including types, class names, and class identifiers. The table *Property* contains the properties of class or of relationship class in ontology, including the identifiers, types, and characteristics; the table *PropertyOf* describes the affiliation between the property and the class; the table *SubSupClass* stores the structure information between two classes; the table *InstanceValue* describes the property values of the instance in the form of a triple; the table *InstanceOf* describes the affiliations of the class and the instance; the table *Restriction* describes the constraints of the properties.

The mapping rule from the OODM to a relational database is defined as follows: for the class of the OODM, the class identifier, name, and type are created and stored in table

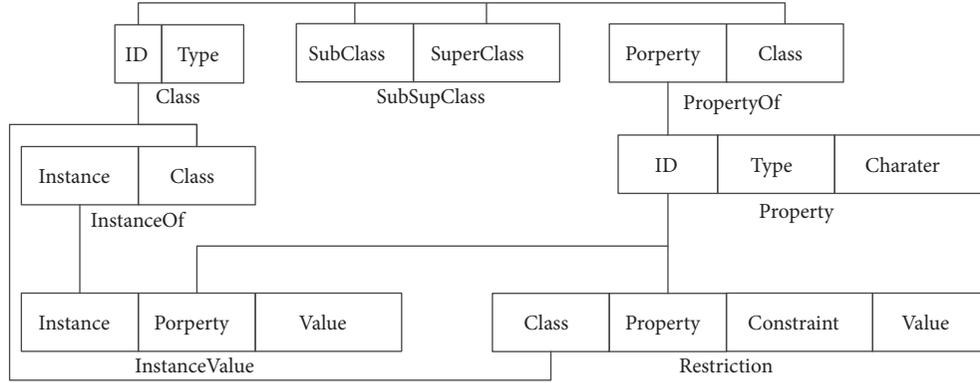


FIGURE 1: Object-oriented data storage structure.

Class; the variables of class are stored in the table *Property*, and property identifiers are then created. The properties and characters of the variables in class correspond to the *Type* and *Character* in Table *Property*. The table *PropertyOf* stores the identifiers of classes and of properties. The table *SubSupClass* stores the identifiers of the parent and child classes expressed the inheritance relationship of the two classes. The table *InstanceValue* stores the properties and the values of an object of the OODM. The identifier of *Property* corresponds to the identifier of the table *InstanceValue* and the property values are stored in the table *Value*. The table *InstanceOf* stores the identifiers of classes and of objects. The property values and the special constraints are stored in the table *Restriction* and correlated with each other in the identifiers of classes and of objects, as shown in Figure 1.

6. Application Example and Experimental

The atmospheric lidar data ontology storage model was examined using the experimental data collected in Laser Radar Center of Remote Sensing of Atmosphere of Xi'an University of Technology.

According to Definition 1, we can get a lidar data ontology $O_{lidar} = \{C_{lidar}, R_{lidar}, A_{lidar}, X_{lidar}\}$, where $A_{lidar} = A_{lidar}^C \cup A_{lidar}^R$. The parts of O_{lidar} are defined as follows:

$$C_{lidar} = \{Aerosol, AtmosTemperature, Wind, AtmosHumidity, Lidarsystem, LidarData, Measure, OtherFactor, GeoInfo, AtmosPressure\}$$

$$A_{lidar}^C(Aerosol) = \{lidarratio, extinctioncoefficient, scatteringcoefficient\}$$

$$A_{lidar}^C(AtmosTemperature) = \{averagetemperature, temperaturedeparture\}$$

$$A_{lidar}^C(Wind) = \{winddirection, windspeed\}$$

$$A_{lidar}^C(AtmosHumidity) = \{watervapourdensity, relativehumidity, absolutehumidity, dewpoint, frostpoint, mix-ingratio\}$$

$$A_{lidar}^C(Lidarsystem) = \{scatteringtype, wavelength, singlepulseenergy, method, signalnoiseratio, rangeresolution\}$$

$$A_{lidar}^C(LidarData) = \{azimuth, elevationangle, measuringtime, measuringheight, datavalue\}$$

$$A_{lidar}^C(Measurer) = \{name, age, email\}$$

$$A_{lidar}^C(OtherFactor) = \{altitude, latitude, longitude, gravityacceleration, pressure\}$$

$$A_{lidar}^C(GeoInfo) = \{altitude, latitude, longitude, gravityacceleration, pressure\}$$

$$A_{lidar}^C(AtmosPressure) = \{altitude, latitude, longitude, gravityacceleration, pressure\}$$

$$R_{lidar} = \{Measure(Lidarsystem, AtmosHumidity), Measure(Lidarsystem, Aerosol), Measure(Lidarsystem, Wind), Measure(Lidarsystem, AtmosTemperature), Control(Measurer, Lidarsystem), MeasureValueof(LidarData, Aerosol), MeasureValueof(LidarData, AtmosTemperature), MeasureValueof(LidarData, Wind), MeasureIn(Aerosol, GeoInfo), MeasureIn(Wind, GeoInfo), MeasureValueof(LidarData, AtmosHumidity), MeasureIn(AtmosTemperature, GeoInfo), MeasureIn(AtmosHumidity, GeoInfo), Measureresultof(LidarData, Lidarsystem), Measureresult(Lidarsystem, LidarData), SubConcept(OtherFactor, GeoInfo), SubConcept(OtherFactor, AtmosPressure)\}$$

$$A_{lidar}^R(Measure) = \{starttime, endtime, address\}$$

$$A_{lidar}^R(Control) = \{starttime, endtime, address\}$$

$$A_{lidar}^R(MeasureValueof) = \{accuracy\}$$

$$A_{lidar}^R(MeasureIn) = \{correlation\}$$

$$A_{lidar}^R(Measureresultof) = \{accuracy\}$$

$$X_{lidar} = \{infer(MeasureValueof(LidarData, Aerosol) \wedge Measureresultof(LidarData, Lidarsystem)) \rightarrow Measure(Lidarsystem, Aerosol), infer(MeasureValueof(LidarData, AtmosTemperature) \wedge Measureresultof(LidarData, Lidarsystem)) \rightarrow Measure(Lidarsystem, AtmosTemperature), infer(MeasureValueof(LidarData, Wind) \wedge Measureresultof(LidarData, Lidarsystem)) \rightarrow Measure(Lidarsystem, Wind), infer(MeasureValue-$$

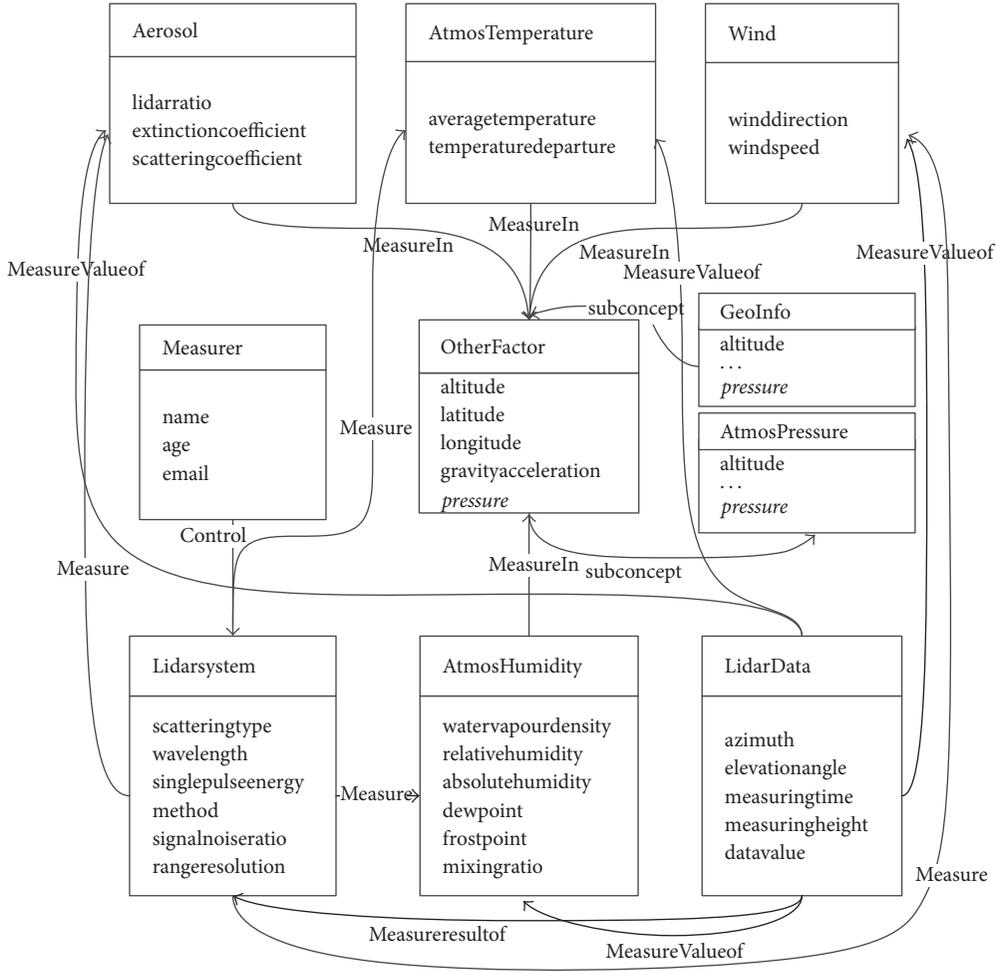


FIGURE 2: Ontology structure of O_{lidar} .

$of(LidarData, AtmosHumidity) \wedge Measurerresultof(LidarData, Lidarsystem)) \rightarrow Measure(Lidarsystem, AtmosHumidity), inverse(Measurerresultof(LidarData, Lidarsystem) \leftarrow Measureresult(Lidarsystem, LidarData))\}$

The structure diagram of O_{lidar} is shown in Figure 2.

The OODM's definition of atmospheric lidar data ontology O_{lidar} is based on Definition 2, shown in Box 1. The class *OtherFactor* is abstracted by the classes of *GeoInfo* and *AtmosPressure*. The method *GetData* (*time:DateTime*):*DataSet* returns the data of class *LidarSystem* or *Aerosol*, and the parameter is the observed times. The *measure* and *control* represent that a lidar system can measure amount of data and an engineer who would be able to control lidars, respectively.

Based on a combination of the storage model of OODM (Figure 2) and relation database, we constructed the data storage structure of lidar data ontology O_{lidar} . For the experiments, we compare the new storage model of the OODM with the extendable Native XML and with the vertical mode of the RMDB in different ontology's instance scale. Experimental setup CPU is Intel(R) Core(TM)2 Duo E7500

2.93 GHz, and the OS is windows server 2003 SP2, and the RMDB is Sql server 2005. The query language of Native XML storage model is XPath, and the query language of OODM and RMDB is sql language. We have investigated three types of query: querying all the instances of a class named *A*; searching all instances of the parent of a class named *B*; querying all the instances named *C*. Taking the limited memory into consideration, four different sizes of atmospheric lidar data ontology instances would be used, 1×10^3 , 2×10^3 , 1×10^4 , and 2×10^4 . Figures 3, 4, and 5 are the comparisons of query times for different data storage models, respectively.

Figures 3, 4, and 5 show that the XML costs the longest query time compared to the other two storage models. Figure 6 shows that increase in the number of instances results in a bigger size of data file and leads to a longer query time, which can be related to the speed of parsing the file and the read-write speed of the disk. Frequent reading and writing files will definitely cost much more time. In addition, the XML method must load the files into the memory for every operation, and thus a large amount of memory usage

```

Cs = {Aerosol, GeoInfo, AtmosTemperature, Wind, AtmosPressure,
AtmosHumidity, Lidarsystem, LidarData, Measurer}
As = {lidarratio, extinctioncoefficient, scatteringcoefficient,
altitude, latitude, longitude, . . .}
Ds = {
Class AtmosPressure is a OtherFactor type-is
Record
    altitude: Float
    latitude: Float
    longitude: Float
    gravityacceleration: Float
    pressure: Float
End
Class GeoInfo is a OtherFactor type-is
Record
    altitude: Float
    latitude: Float
    longitude: Float
    gravityacceleration: Float
    pressure: Float
End
Class OtherFactor type-is
Record
    Union GeoInfo, AtmosPressure
End
Class Aerosol type-is
Record
    lidarrotio: Float
    extinctioncoefficient: Float
    scatteringcoefficient: Float
    GetData(time:DateTime):DataSet
End
Class Lidarsystem type-is
Record
    scatteringtype: String
    wavelength: Float
    singlepulseenergy: Float
    method: String
    signalnoiseratio: Float
    rangeresolution: Float
    GetData(time:DateTime):DataSet
    measure: Set of LidarData[(1, n)]
End
Class LidarData type-is
Record
    azimuth: Float
    elevationangle: Float
    measuringtime: DateTime
    measuringheight: Float
    datavalue: Float
End
Class Measurer type-is
Record
    name: String
    age: Int
    email: String
    control: Set of LidarData[(1, n)]
End
Class Measure type-is
Record
    starttime: DateTime
    endtime: DateTime

```

```

        address: String
    End
    ...}
    
```

Box 1: Definition of O_{lidar} 's OODM.

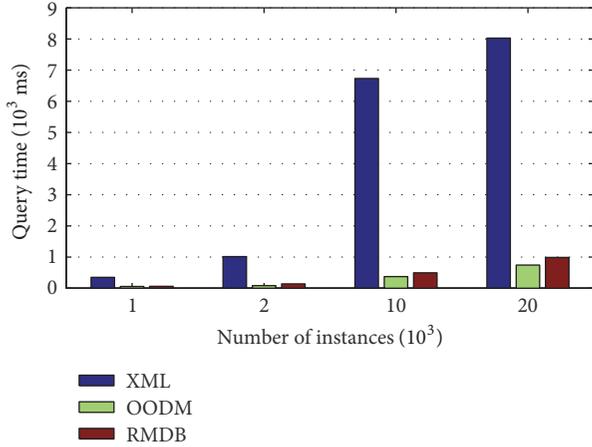


FIGURE 3: Comparison of type A's query times.

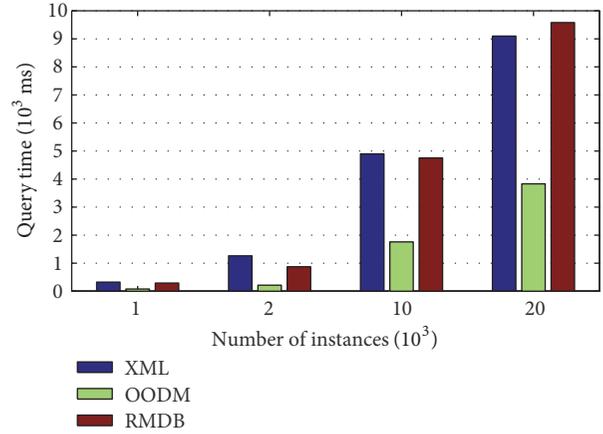


FIGURE 5: Comparison of type C's query times.

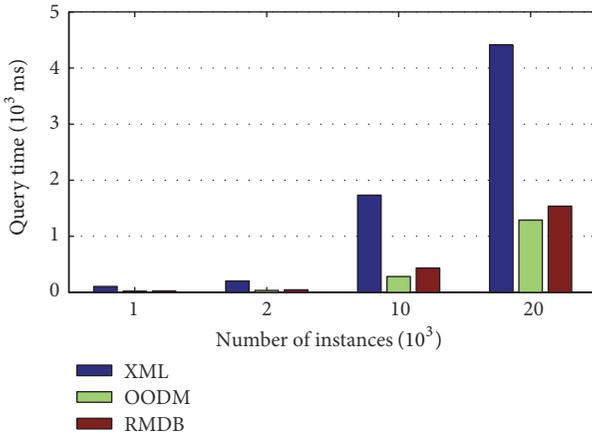


FIGURE 4: Comparison of type B's query times.

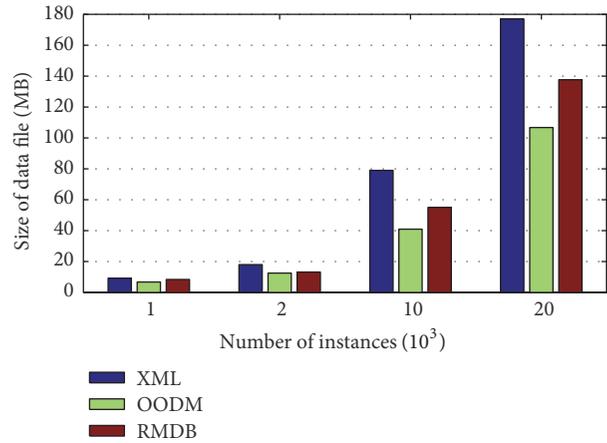


FIGURE 6: Comparison of data file sizes.

causes an inefficiency operation. In query method type C and ontology with size 1×10^4 and 2×10^4 , the data records are more than one million, which will exceed the total usable memory size of the testing machine after 49 s and 91 s, respectively. It reveals that, with the increase of atmospheric lidar data instances of ontology, XML storage model will consume much more time and memory compared to the other methods. Obviously, the OODM and RMDB storage methods are superior to the Native XML method in query time and data file size. Figures 3, 4, and 5 also show that the consuming time of the RMDB method is greater than that of the OODM in all three types of queries. The RMDB method is associated with the data in the form of the tables which joined with each other in the querying process. The OODM

method can easily obtain the object data due to its simplified data table.

7. Conclusions

In this paper, a new ontology storage model is developed for lidar data based on object-oriented model to maximize the query efficiency and minimize the data storage file size by the representation method and mapping mechanism. The model is very useful for lidar data as (1) the lidar data ontology structure can be easily transformed into the OODM structure; (2) the relationship of lidar data ontology can be clearly expressed and stored; and (3) the OODM of lidar data have high expansibility and query efficiency. Some

extensions, which are optimized method for processing the temporal data such as efficient query method and storage model, are further addressed to consider more practical factors.

Disclosure

This paper is an extended version of the International Conference on Computer Science and Technology (CST2016).

Competing Interests

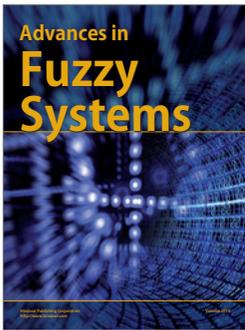
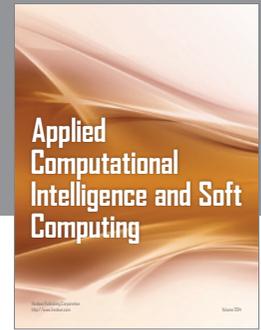
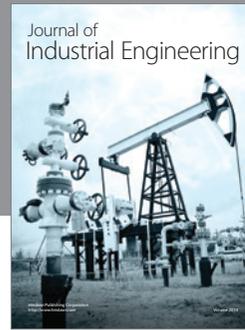
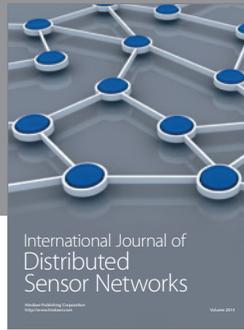
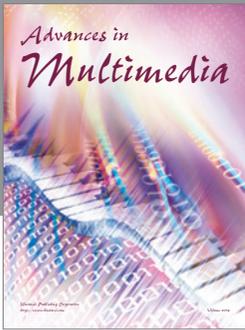
The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) (Grants nos. 61675166 and 61605157).

References

- [1] E. Yang, B. Matthews, and M. Wilson, "Enhancing the core scientific metadata model to incorporate derived data," *Future Generation Computer Systems*, vol. 29, no. 2, pp. 612–623, 2013.
- [2] A. Wiacek, J. R. Taylor, K. Strong et al., "Ground-based solar absorption FTIR spectroscopy: characterization of retrievals and first results from a novel optical design instrument at a new NDACC complementary station," *Journal of Atmospheric and Oceanic Technology*, vol. 24, no. 3, pp. 432–448, 2007.
- [3] H. D. Kambezidis and D. G. Kaskaoutis, "Aerosol climatology over four AERONET sites: an overview," *Atmospheric Environment*, vol. 42, no. 8, pp. 1892–1906, 2008.
- [4] N. Sugimoto, I. Matsui, A. Shimizu, and T. Nishizawa, "Lidar network for monitoring Asian dust and air pollution aerosols," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS '08)*, pp. II573–II576, July 2008.
- [5] Z. Daquan, Z. Lu, Y. Jie et al., "The impact of temperature and precipitation variation on drought in China in last 50 years," *ACTA Physica Sinica*, vol. 59, no. 01, pp. 655–663, 2010.
- [6] L. Zhou, Z.-Q. Gong, R. Zhi, and G.-L. Feng, "Study on the regional characteristics of the temperature changes in China based on complex network," *Acta Physica Sinica*, vol. 58, no. 10, pp. 7351–7358, 2009.
- [7] S. Li, D. Hua, Y. Wang, F. Gao, Q. Yan, and X. Shi, "Fiber-optic spectroscopic rotational Raman lidar with visible wavelength fiber Bragg grating for atmospheric temperature measurement," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 153, pp. 113–118, 2015.
- [8] V. Mateos, V. A. Villagr a, F. Romero, and J. Berrocal, "Definition of response metrics for an ontology-based Automated Intrusion Response Systems," *Computers and Electrical Engineering*, vol. 38, no. 5, pp. 1102–1114, 2012.
- [9] M. D. M. R. Garc a, J. Garc a-Nieto, and J. F. Aldana-Montes, "An ontology-based data integration approach for web analytics in e-commerce," *Expert Systems with Applications*, vol. 63, pp. 20–34, 2016.
- [10] S. Abburu and S. B. Golla, "Ontology storage models and tools: an authentic survey," *Journal of Intelligent Systems*, vol. 25, no. 4, pp. 539–553, 2016.
- [11] D.-W. Jeong, H.-Y. Shin, D.-K. Baik, and Y.-S. Jeong, "An efficient web ontology storage considering hierarchical knowledge for jena-based applications," *Journal of Information Processing Systems*, vol. 5, no. 1, pp. 11–18, 2009.
- [12] L. Al-Jadir, C. Parent, and S. Spaccapietra, "Reasoning with large ontologies stored in relational databases: the OntoMinD approach," *Data & Knowledge Engineering*, vol. 69, no. 11, pp. 1158–1180, 2010.
- [13] B. Sujatha and S. V. Raju, "Ontology based natural language interface for relational databases," in *Proceedings of the 2nd International Conference on Intelligent Computing, Communication and Convergence (ICCC '16)*, pp. 487–492, Odisha, India, January 2016.
- [14] F. Zhang, Z. M. Ma, and W. Li, "Storing OWL ontologies in object-oriented databases," *Knowledge-Based Systems*, vol. 76, pp. 240–255, 2015.
- [15] X.-Y. Du, M. Li, and S. Wang, "Survey on ontology learning research," *Journal of Software*, vol. 17, no. 9, pp. 1837–1847, 2006.
- [16] E. Hammann, A. Behrendt, F. Le Mounier, and V. Wulfmeyer, "Temperature profiling of the atmospheric boundary layer with rotational Raman lidar during the HD(CP)2 Observational Prototype Experiment," *Atmospheric Chemistry and Physics*, vol. 15, no. 5, pp. 2867–2881, 2015.
- [17] C. Xie, M. Zhao, B. Wang et al., "Study of the scanning lidar on the atmospheric detection," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 150, no. 1, pp. 114–120, 2015.
- [18] H. Chen, D.-X. Hua, Y.-K. Zhang, and C.-X. Zhu, "A method of vertical and horizontal plus cubic spline interpolation for Mie scattering lidar profile data," *Acta Physica Sinica*, vol. 63, no. 15, Article ID 154204, 2014.
- [19] Z. Zhou, D. Hua, H. Chen, and Y. Zhang, "Study on the regional atmospheric lidar data storage model based on ontology," in *Proceedings of the The International Conference on Computer Science and Technology (CST '16)*, pp. 589–596, Shenzhen, China, January 2016.
- [20] E. M. Sanfilippo and S. Borgo, "What are features? An ontology-based review of the literature," *Computer-Aided Design*, vol. 80, pp. 9–18, 2016.
- [21] M. Chen, H. Lin, D. Liu, H. Zhang, and S. Yue, "An object-oriented data model built for blind navigation in outdoor space," *Applied Geography*, vol. 60, pp. 84–94, 2015.
- [22] H. H. Wang, N. Gibbins, T. R. Payne, and D. Redavid, "A formal model of the Semantic Web Service Ontology (WSMO)," *Information Systems*, vol. 37, no. 1, pp. 33–60, 2012.
- [23] X. Jianwu and B. Yi, "Based on the object-oriented database ontology storage method research," *Advances in Information Sciences and Service Sciences*, vol. 4, no. 15, pp. 383–389, 2012.
- [24] J. Kim, D. Jeong, and D.-K. Baik, "An XPath-based OWL storage model for effective ontology management in Semantic Web environment," *Journal of Zhejiang University: Science A*, vol. 10, no. 6, pp. 843–857, 2009.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

