Research of an Integrated Organization and Management Methods for Natural Resource Data

Xueyuan Li[a], Changqing Liu[b] * ,Chen Fan[c], Wei Tang[d], Yizhou Yang[e]

Beijing GEOWAY Software Co., Ltd. 100043 Shijingshan, Beijing, China,

[a]lixueyuan@geoway.com.cn, [b]liuchangqing@geoway.com.cn, [c]fanchen@geoway.com.cn, [d]tangwei@geoway.com.cn, [e]yangyizhou@geoway.com.cn

Abstract. With the continuous growth in the diversity, volume, and complexity of natural resource data, a singular approach to organize, store, and manage spatial databases has become insufficient for managing multi-source data . Against this backdrop, this research conducted to build upon traditional spatial data storage and management mechanisms by constructing a file system and a non-relational database tailored for distributed environments. This involves studying the logical structure and modeling methods of an integrated data model for natural resources, formulating integrated data storage strategies, and designing unified data access interfaces. The ultimate objective is to achieve the unified storage, organization, management, and access of multi-source heterogeneous data. To validate the effectiveness of this integrated data model, an empirical study was conducted using natural resource data from a specific region. The results demonstrate that the integrated data model can fully leverage the comprehensiveness and efficiency advantages of big data management for natural resources. Specifically, it meets the requirements for precise query statistics, efficient application analysis, flexible data distribution, and rapid delivery of 2D and 3D products and services related to natural resource data.

Keywords : Natural resources, Integrated data models, Organizational management, Storage strategies, Access interfaces

1 Introduction

A singular data storage paradigm cannot fully meet the diverse requirements of various application scenarios, such as browsing and display, extraction and distribution, computational analysis, and service support, regarding data storage, organization, and access performance[1]. Hence, it is imperative to conduct research on storage technologies for natural resource data based on the study of spatial data conceptual models. The aim is to achieve one-time data modeling through integrated organization and management of natural resource data, while supporting diverse data applications. The cornerstone of accomplishing these objectives is the construction of an integrated data model for natural resources.

K. Subramaniyam et al. (eds.), *Proceedings of the 3rd International Academic Conference on Blockchain, Information Technology and Smart Finance (ICBIS 2024)*, Atlantis Highlights in Computer Sciences 21,
https://doi.org/10.2991/978-94-6463-419-8_19

Using an object-oriented approach, natural resource management objects are encapsulated in terms of data content and data access, enabling comprehensive management of both data and methods through an integrated data model(Figure 1). The data content of these objects serves as attribute members within the integrated data model, encompassing full time-series data in various storage formats along with associated data relationships. Meanwhile, data access for natural resource management objects is defined as method members within the model, supporting the retrieval and scheduling of data in any storage format and at any temporal version[2].

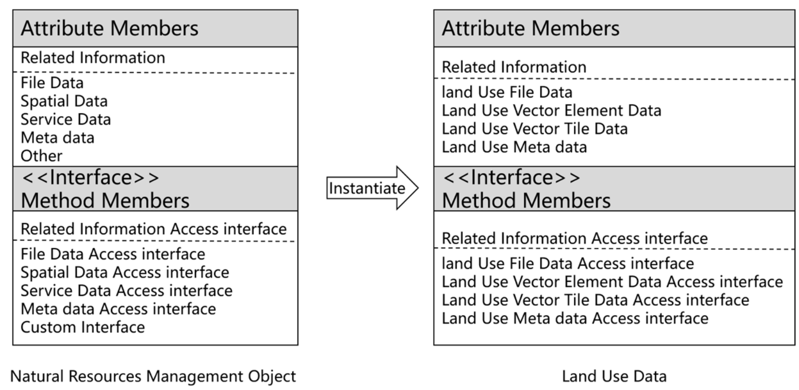

Fig. 1. Integrated Data Model Logic Composition

2 Research methodS

2.1 Construction of Integrated Data Model

By coupling basic spatial data models such as feature datasets, mosaic datasets, and tile datasets, and extending temporal attributes, the data members of the integrated data model (IModelData) are constructed, guided by data applications.

$$IModelData = (D, R) \quad (1)$$

$$D = \{d_i | i \in D_{core}\} \quad (2)$$

$$D_{core} = \{File\ format, Spatial\ form, Service\ form, Metadata\} \quad (3)$$

R defines the associative relationships between elements in D.

The integrated data model comprises data members that are constituted by five interdependent and interrelated components: file-form data, spatial-form data, service-form data, metadata, and associative information(Figure 2), all specific to natural resource management entities[3].

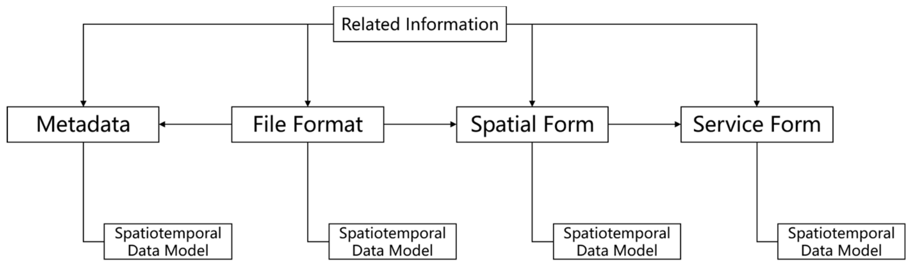

Fig. 2. Composition of Data Members in the Integrated Data Model

Association information defines the coupling relationships among the various components of the data members within the integrated data model[4]. It provides detailed descriptions of the storage locations for different forms of data, as well as the associative and temporal relationships between them. Serving as the entry point for accessing the content of the integrated data model, association information is utilized in specific application scenarios to identify the optimal form of data and route to its corresponding storage location.

$$R = \{r_i | i \in D_{core}\} \quad (4)$$

$$r_i = (e_i, t_i, p_i, o_i, a_i, c_i) \quad (5)$$

Each data storage form is described using a six-tuple(Table 1). When the data of form i exists, e_i is set to 1; otherwise, it is 0. When e_i equals 1, the other components of the r tuple contain valid values. When e_i equals 0, the other components of the r tuple are set to null.

Table 1. The Meaning of Metadata Items in Association Information

Itemize	Meaning
e	The Existence of Data in a Specific Storage Form
t	The adopted spatio-temporal data model, whose specific values depend on parameter e
p	The data access path, with value dependency on parameters e and t
o	The online status of data (online, offline), with value dependency on parameter e
a	Data access privileges, with value dependency on parameter e
c	Custom Content

Compared to a single data model, the integrated data model comprehensively meets application requirements such as browsing and display, extraction and distribution, computational analysis, and service support. Combined with distributed storage technology, the same natural resource management object can simultaneously support multiple data applications, fully leveraging the performance advantages of physical devices. In scenarios involving comprehensive data applications such as advanced analysis and knowledge mining, the advantages of the integrated data model, such as comprehensiveness of data and flexibility of access, become even more prominent.

2.2 Storage Strategy for Integrated Data Model

A hybrid storage strategy is adopted[5], which comprehensively utilizes the advantageous features of spatial databases, NoSQL databases, and distributed file systems to achieve integrated management of structured, semi-structured, and unstructured data such as vectors, images, terrain, tables, tiles, and files(Figure 3). Each type of data is matched with the optimal physical storage format[6]. For file-based data, the physical files are stored in the file system, while the positional relationships between files are stored in the spatial database. Among spatial data, vector data adopts a feature dataset model, while image and terrain data utilize a mosaic dataset model; both are stored in the spatial database. Service-oriented data is stored in a tile dataset model within a NoSQL database[7]. Metadata and the associative information between various forms of data within the integrated data model are stored in a relational table model within the spatial database[8].

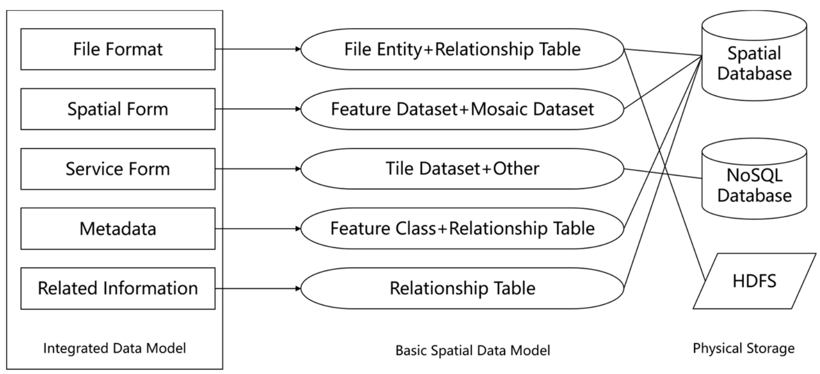

Fig. 3. Storage Strategy for the Integrated Data Model

2.3 Access Interfaces for the Integrated Data Model

The Integrated Data Model encompasses multiple storage systems such as spatial databases, NoSQL databases, and distributed file systems, along with various data forms including structured, semi-structured, and unstructured data. This multitude of storage systems and data forms leads to a diverse set of data access interfaces within the hybrid storage architecture, increasing the complexity of data storage and management. In the context of data access within this hybrid storage, the Integrated Data Model provides data access interfaces that are independent of natural resource management objects for each form of data(Figure 4). These interfaces generate detailed access information based on the associative data within the Integrated Data Model and then automatically select appropriate low-level APIs through a parameter generator to manipulate the data stored in the hybrid storage[9].

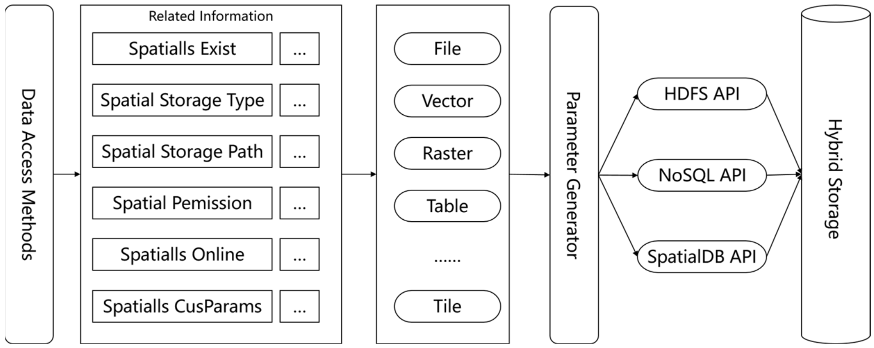

Fig. 4. The Logic of Data Access Interfaces for the Integrated Data Model

The integrated data model adheres to the principle of separating the logical model from physical storage. The application layer facilitates data access and management based on the logical model, delivering user-friendly and transparent data access services through the integrated data model's methodologies(Table 2). This approach minimizes the technical learning curve for users, eliminating the need to delve into the complexities of underlying data storage, and enabling them to concentrate more on the design and implementation of business logic.

Table 2. Integrated Data Model Data Access Interface

Interface Type	Interface	Describe
DataSource	Connect	Connecting diverse data sources such as spatial databases, NoSQL databases, and file systems, which are related to specific land and resource management objects.
	DisConnect	Disconnecting from various data sources, including spatial databases, NoSQL databases, and file systems, pertaining to specific land and resource management objects.
	Create	Create a dataset of a specified type and name within a designated type of data source.
	Desc	Delete the dataset of a specified type and name from a designated type of data source.
	AutoExecute	Based on the application scenario, adapt to the optimal data storage format and return the data in the form of feature datasets, mosaic datasets, tile datasets, file datasets, or relational tables.
	Execute	Based on the input parameters, return the specified data storage format, and provide the data in the form of feature datasets, mosaic datasets, tile datasets, file datasets, or relational tables.
FileDataset	Open	Open the file dataset, whose compositional information is stored in the spatial database while the actual file entities are stored in the file system.
	Import	Ingest the external files into the database.
	Del	Delete all or some of the files within the file dataset.
	Query	Query the file dataset and return the retrieved files.
VectorDataset	Open	Open the feature dataset in the spatial database.

Interface Type	Interface	Describe
	Import	Write external library file format vector data into the feature dataset.
	Construct	Generate feature dataset content based on the ingested file format data.
	Del	Delete all or part of the feature dataset content.
	Query	Query the feature dataset and return the search results.
RasterDataset	Open	Open the mosaic dataset in the spatial database.
	Import	Ingest the external raster data in file format into the database.
	Construct	Generate mosaic dataset content based on the ingested file-based data.
	Del	Delete all or part of the mosaic dataset content.
	Query	Query the mosaic dataset and return the search results.
TileDataset	Open	Open the tile dataset in the NoSQL database.
	Import	Write external tile data in file format into the tile dataset.
	Construct	Generate tile dataset content based on the ingested spatial data.
	Del	Delete all or some of the tiles within the tile dataset.
	Query	Query the tile dataset and return the retrieved tiles.
TableDataset	Open	Open the relational table (storing metadata and association information) in the spatial database.
	Import	Ingest the row data from external files into the database.
	Del	Delete all or some of the records in the relational table.
	Query	Query the file dataset and return the retrieved records.
Feature	Geometry	Return the geometric data of the specified feature.
	Fields	Return the attribute data of the specified feature.
Raster	Band	Return data from a specific spectral band.
	Cell	Return specific pixel data.

3 Verification methodS

Construct a natural resource database that integrates relational databases, non-relational databases, and distributed file systems for unified data storage, organization, and management. Validate the construction process and effectiveness of the integrated data storage by selecting data ingestion as the verification point. Identify typical business scenarios such as data distribution, data browsing, and overlay statistics. Conduct a comparative analysis of the performance of both standalone and integrated storage solutions in supporting these scenarios to verify the technical specifications and advantages of the integrated storage approach(Table 3).

Table 3. Test Scenarios

Sequence Number	Scene Classification	Test Scene Description
1	Model Building	Conduct integrated model building process testing for vector and raster data
2	Vector Data Distribution	Perform distribution testing on vector data in different storage formats
3	Image Data Distribution	Carry out distribution testing on image data in varied storage forms

Sequence Number	Scene Classification	Test Scene Description
4	Vector Tile Data Browsing	Conduct browsing tests on data sources with diverse storage patterns
5	Vector Superposition Statistical Analysis	Execute overlay statistical testing on data in various storage configurations

4 Results

4.1 Integrated Data Model Construction for Vector Data

Taking land cover patch data as the management object, storage and management are carried out through an integrated data model to verify its effectiveness. The construction sequence of the integrated data model for land cover patch management objects is as follows: "manual file data ingestion" -> "spatial data association and ingestion" -> "metadata association and ingestion" -> "service data association and ingestion". After the model construction is completed, application verification is conducted on data in different forms(Table 4).

Table 4. Time Consumption Table for Integrated Data Model Construction of Vector Data

Data Format	Storage Method	Construction Time (seconds)
File Format	HDFS	11.99
Spatial Format	Oracle-Spatial	1032
Metadata	Oracle	1.2
Map Tiles	MongoDB	876

4.2 Integrated Data Model Construction for Imagery Data

Taking imagery data as the management object, storage and management are carried out through an integrated data model to verify its effectiveness. The construction sequence of the integrated data model for imagery data management objects follows: "ingestion of file data paths and organizational information" -> "spatial data association and ingestion" -> "metadata association and ingestion" -> "service data association and ingestion". After the completion of model construction, application verification is conducted on data in different forms(Table 5).

Table 5. Time Consumption Table for Integrated Data Model Construction of Imagery Data

Data Format	Storage Method	Construction Time (seconds)
File Format（File Storage）	HDFS	2.3
Spatial Format(Mosaic Dataset)	Oracle-Spatial	33
Metadata	Oracle	0.8
Map Tiles	MongoDB	7200

4.3 Practical Verification and Analysis of the Integrated Data Model

Conducted tests on a specific area for vector data distribution, imagery data distribution, vector data browsing, and overlay statistical analysis of vector data. The vector data (12GB) and imagery data (361GB) are stored in Oracle-Spatial and HDFS respectively, while the vector tile data are stored in both Oracle and MongoDB. The two vector layers involved in the overlay statistical analysis are stored in HDFS and Oracle, with a total of 14GB of raw data. The test verification results from the comparative experiments will be summarized(Table 6).

Table 6. Bilingual Review - Academic Paper Version

Scene Classification	SQL DB	NoSQL DB	HDFS	Integrated Data Model
Vector Data Distribution	181min	N/A	122min	122min
Image Data Distribution	262min	N/A	70min	70min
Vector Tile Data Browsing	2.6s	1.77s	N/A	1.77s
Vector Superposition Statistical Analysis	220min	N/A	8.6min	8.6min

Note: N/A denotes that the specific database configuration does not support the operation, has performance limitations, or is not suitable for comparative analysis.

From the experimental results, it can be seen that for data distribution services, distributed file systems have certain performance advantages over traditional relational databases. Analysis suggests that the data redundancy characteristics of distributed file systems provide higher access efficiency for national fundamental data. In terms of data browsing, non-relational databases have higher browsing efficiency than relational databases, mainly because the key-value model adopted by non-relational databases has higher query efficiency and is more suitable for rapid access to vector tile slicing. For overlay statistics involving smaller data scales, the non-relational database method based on MongoDB outperforms traditional relational database storage strategies in terms of performance, mainly due to MongoDB's lightweight model design and in-memory database characteristics[10]. It directly stores serialized data involved in spatial calculations in the full memory space, which is more efficient than disk access. However, compared to relational databases, distributed file systems have lower efficiency because the data scale is too small to fully leverage the advantages of distributed systems, and the distributed architecture increases data communication costs during the calculation process. Correspondingly, in large-scale overlay statistics experiments, it can be seen that distributed file systems have better performance than traditional relational databases, while non-relational databases are limited by data storage scale and data structure, and cannot complete large-scale overlay statistical calculation tasks.

5 Conclusion

In summary, the integrated approach for organizing and managing natural resource data proposed in this study, along with its comparative analysis against traditional relational databases, non-relational databases, and distributed file systems, has shown its unique advantages. It can support functionalities that are not possible with other single storage methods and boasts a relatively high level of data management efficiency. When applied to a range of operational business scenarios, the integrated data model introduced in this research can fully exploit the comprehensiveness and efficiency benefits of big data management in the context of natural resources. It demonstrates the ability to handle complex business requirements and holds practical significance for the comprehensive utilization of natural resource data.

References

1. De'an L ,Liqiong L .A cadastral spatial data storage structure based on relational database[J].Geo-spatial Information Science,2012,4(3):15-20. DOI：10552214
2. Meine N V ,B G V ,Jan G H , et al.Relational versus instrumental perspectives on values of nature and resource management decisions[J].Current Opinion in Environmental Sustainability,2023,65101374-. DOI：10.1016/J.COSUST.2023.101374
3. Malte J .A flexible likelihood-based neural network extension of the classic spatio-temporal model[J].Spatial Statistics,2024,59100801-. DOI：10.1016/J.SPASTA.2023.100801
4. Bin W Z ,Yong G ,Hexiang B , et al.Spatial Association from the Perspective of Mutual Information[J].Annals of the American Association of Geographers,2023,113(8):1960-1976. DOI：10.1080/24694452.2023.2209629
5. Duan Z .Research on Distributed Storage Technology for Big Data[J].Engineering Advances,2023,3(3): 10.26855/EA.2023.06.020
6. Pierre-Alain F ,Markus F .A Survey of Methods for Converting Unstructured Data to CSG Models[J].Computer-Aided Design,2024,168103655-.DOI：10.1016/J.CAD.2023.103655
7. Győrödi A C ,Turtureanu T ,Győrödi Ş R , et al.Implementing a Synchronization Method between a Relational and a Non-Relational Database[J].Big Data and Cognitive Computing,2023,7(3): DOI：10.3390/BDCC7030153
8. Li S ,Liang Y ,Deng X , et al. Exploring the Spatial Heterogeneity of Rural Development in Laos Based on Rural Building Spatial Database[J].Land,2023,12(11). DOI：10.3390/LAND12112008
9. Mora G C ,Garrigós I ,Zubcoff J , et al.Model-based Generation of Web Application Programming Interfaces to Access Open Data[J].Journal of Web Engineering,2020,19(7-8):1147-1172. DOI：10.13052/JWE1540-9589.197810
10. Inês C ,Filipe S ,Jorge B .Performance Evaluation of NoSQL Document Databases: Couchbase, CouchDB, and MongoDB[J].Algorithms,2023,16(2):78-78. DOI：10.3390/A16020078