Journal of Hydrology 587 (2020) 124973

Contents lists available at ScienceDirect

Journal of Hydrology

journal homepage: www.elsevier.com/locate/jhydrol

Research papers

A data sharing method in the open web environment: Data sharing in hydrology

Jin Wang[a,b,c], Min Chen[a,b,c,*], Guonian Lü[a,b,c], Songshan Yue[a,b,c], Yongning Wen[a,b,c], Zhenxu Lan[a,b,c], Shuo Zhang[a,b,c]

[a] Key Laboratory of the Virtual Geographic Environment, Ministry of Education of PRC, Nanjing Normal University, Nanjing, Jiangsu, China
[b] Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing Normal University, Nanjing, Jiangsu, China
[c] State Key Laboratory Cultivation Base of Geographical Environment Evolution (Jiangsu Province), Nanjing Normal University, Nanjing, Jiangsu, China

ARTICLE INFO

Keywords:
Data sharing
Data configuration
Hydrological data
Hydrological modeling and simulation

ABSTRACT

Data sharing plays a fundamental role in providing data resources for geographic modeling and simulation. Although there are many successful cases of data sharing through the web, current practices for sharing data mostly focus on data publication using metadata at the file level, which requires identifying, restructuring and synthesizing raw data files for further usage. In hydrology, because the same hydrological information is often stored in data files with different formats, modelers should identify the required information from multisource data sets and then customize data requirements for their applications. However, these data customization tasks are difficult to repeat, which leads to repetitive labor. This paper presents a data sharing method that provides a solution for data manipulation based on a structured data description model rather than raw data files. With the structured data description model, multisource hydrological data can be accessed and processed in a unified way and published as data services using a designed data server. This study also proposes a data configuration manager to customize data requirements through an interactive programming tool, which can help in using the data services. In addition, a component-based data viewer is developed for the visualization of multisource data in a sharable visualization scheme. A case study that involves sharing and applying hydrological data is designed to examine the applicability and feasibility of the proposed data sharing method.

1. Introduction

Geographic modeling is an effective way to explore geographical processes and understand geographical rules (Demeritt & Wainwright, 2005; Granell et al., 2013; Lü et al., 2019). Geographic data record the intrinsic information related to geographic phenomena and processes, which serve as the inputs for geographic simulations. With the development of geographic modeling, researchers and governments have produced huge geographic data in various disciplines and domains. Due to the high cost of data collection and the complexity of data processing, data sharing has become one of the key tasks in the field of geomodeling (Beran and Piasecki, 2009; Chen et al. 2009a,b; Voinov and Cerco, 2010; Abdallah and Rosenberg, 2019; Xue et al., 2019).

With the development of information technology (IT), the methods of data acquisition by users have passed through following approximate stages. (1) In the early days of computer development, data was usually stored in files, and there was no effective way to manage large amounts of data. Users typically shared data by copying the data files (Peng, 2005). Although this method of data sharing is direct and flexible, it is inefficient and inconvenient for managing and copying big datasets. (2) With the popularization of databases, more and more data resources have been stored in databases. Databases provide not only an effective way to manage massive data resources but also unified interfaces through which users can access these resources (Gogu et al., 2001; Horsburgh et al., 2008). However, a specific database normally has its own specific data description model, which requires data contributors or users to follow specific specifications for sharing or accessing the data. For example, ArcGIS (https://www.esri.com/zh-cn/home), which is a widely used software system in geographic information systems (GIS), provides two lightweight databases (i.e., file and personal geodatabases). Although ArcGIS provides interfaces through which users can share and access data in these databases, users should have some amount of familiarity with these databases to access the required data. (3) With the development of web technology, users can easily access distributed data resources through the Internet (Peng, 2005; Markert et al., 2019; McDonald et al., 2019). In the web environment, data

* Corresponding author.
E-mail address: chenmin0902@163.com (M. Chen).

https://doi.org/10.1016/j.jhydrol.2020.124973

Available online 18 April 2020

contributors upload data files into a data storage server and describe these data resources with specific description specifications (Morsy et al., 2017). Then, users can search and access the data through metadata information. Examples include the Hydrologic Information System (CUAHSI-HIS, http://his.cuahsi.org/) developed by the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI). This system offers web services, tools, standards and procedures that enhance access to data for hydrological analyses (Whitenack, 2010). Hydroshare (https://www.hydroshare.org/landingPage/) is a well-known platform for sharing, discovering and tracing hydrological data (Horsburgh et al., 2016b). Furthermore, many studies have been conducted on web-based data sharing (Peng, 2005; Zhang et al., 2007; Ames et al., 2012, Han et al., 2012, 2014; Horsburgh et al., 2009; Zhu et al., 2017; Zhu and Yang, 2019). Due to the openness and convenience of web technology, more and more data has been shared in the web environment (Jones et al., 2016; Markert et al., 2019; McDonald et al., 2019).

Web-based data sharing methods have provided the foundation for data sharing in the open web environment and the possibility for users with different backgrounds to access vast data resources. However, current data sharing methods have mainly focused on data publication using metadata at the file level, which requires identifying, restructuring and synthesizing raw data files for further usage (Abdallah and Rosenberg, 2019). Moreover, these data configuration efforts are difficult to reuse, which causes repetitive labor. Hydrological modeling has also suffered from these problems in the data sharing process (Laituri and Sternlieb, 2014; Miller et al., 2004; Maidment, 2016; Abdallah and Rosenberg, 2019).

Hydrologic modeling and simulation often involve problem-driven research that typically requires multisource and multiformat data for different research purposes. The hydrological data (e.g., observed data, reanalysis data, etc.) may come from different data collection devices, models and data analysis systems and may be stored in different formats. Currently, the data storage or description methods can be divided into two categories. (1) Common data description specifications examples include WaterML (https://www.opengeospatial.org/standards/waterml), which is a standard information model developed by the Open Geospatial Consortium (OGC) for organizing hydrological observation data (Kadlec et al., 2015); the Observation Data Model (ODM), which provides a consistent format for point-based environmental observation data storage and retrieval in a relational database (Horsburgh et al., 2008, 2016a); and the Water Management Data Model (WaMDaM), which organizes and stores water management data from multiple sources using contextual metadata and controlled vocabularies (Abdallah and Rosenberg, 2019). (2) Proprietary file formats are used by examples including the Storm Water Management Model (SWMM, https://www.epa.gov/water-research/storm-water-management-model-swmm), HydraPlatform (http://umwrg.github.io/HydraPlatform/) and Riverware (Zagona et al., 2001), and all of these models use their own rules to organize the data. Customized .TXT files, comma-separated value (CSV) files and Microsoft Excel files are usually employed by individual researchers to develop their own models due to the different research backgrounds among researchers.

The above two data description methods mainly focus on the data description itself, which cannot emphasize the specific hydrological information contained in the data. For example, depending on the decisions of the data producers, hydrologic variables (e.g., runoff, evaporation and wind speed) can be stored in various data formats. Before users can extract the required information, they must be familiar with the corresponding data formats, which increases the cost for users to obtain specific data. Moreover, the heterogeneity and diversity of hydrological data make it difficult for users to customize their data requirements, and application-specific data customization efforts are difficult to reuse.

For example, a SWAT model, which is a comprehensive physically based hydrological model, requires a variety of data for its execution, such as DEM data from the study area, soil data, land use data and meteorological data (McDonald et al., 2019), (https://swat.tamu.edu/media/19754/swat-io-2009.pdf). Shared data is an important data source for users who lack in-situ observed data. Meteorological data (e.g., precipitation and temperature) are required as the drivers of the SWAT model, and the preparation process of such data is moderately tedious. Specifically, the National Climatic Data Center (NCDC, https://www.ncdc.noaa.gov/) provides an open ftp server from which users can download climatic data sets, e.g., temperature, precipitation, air pressure and dew point temperature data (ftp://ftp.ncdc.noaa.gov/pub/data/noaa/isd-lite/). First, users need to download the required data files and be familiar with the data organization structure according to their metadata. Second, the required information needs to be extracted from the downloaded raw data files and processed following specific data requirements (e.g., missing values should be denoted as −99, and the data must be temporally continuous). Finally, the extracted data need to be organized in a SWAT-compatible data format. These data customization processes are unavoidable for users relying on shared data. However, these application-related data customization efforts are highly repetitive for many users of the SWAT model running similar simulations.

This article introduces a data sharing method, which can improve the data usability in the open web environment. In this study, the information contained in hydrological data can be described using the Universal Data eXchange (UDX) model in a clear and structured way and can provide data to users through data services rather than through downloading raw data files. Users can request the required data by invoking corresponding data services in the designed data server without considering the raw data sources. Moreover, the proposed data configuration manager provides an interactive programming tool for data customization. Additionally, a component-based data viewer is developed to display different types of hydrological information in a shareable visualization scheme.

The remainder of this article is structured as follows. The conceptual framework of the proposed data sharing method is introduced in Section 2. Section 3 introduces the design and implementation of the proposed data sharing method. In Section 4, a case study involving the sharing and application of hydrological data is applied. Finally, conclusions and future works are presented in Section 5.

2. Conceptual framework of the proposed data sharing method

The purpose of data sharing is to use available data to the greatest possible extent. Because the ease of use of data is rarely considered in most data sharing methods, a data sharing method that can improve the availability and usability of data in the open web environment is urgently needed. Fig. 1 shows the conceptual framework of the proposed data sharing method, which includes three components: a data server, a data configuration manager and a data viewer.

The UDX model is designed to describe heterogeneous data in a clear and structured way (Yue et al., 2015). Unlike other data models (e.g., ASCII GRID, GeoTIFF, NetCDF and GeoJson), the UDX model is a self-explanatory data description model with a hierarchical structure that is easy to understand. As shown in Fig. 2, raw evaporation data are stored in a plain text file that contains the site information (number, name and position) and evaporation values (at each site). The top and bottom panels show the structured expression of information in the UDX model. Compared to the raw data file, the information expressed by the UDX model can be easily interpreted by users. The information stored in the raw data files may use different organizational structures (different formats), but the structured expression of the information can remain the same. When the information is published as data services, users can access the required information without accessing the raw data files, which improves the efficiency of data access.

The sources of hydrological data are diverse and can be divided into three categories: data files, databases and data shared by online platforms. Each of the data sources contains abundant hydrological data.

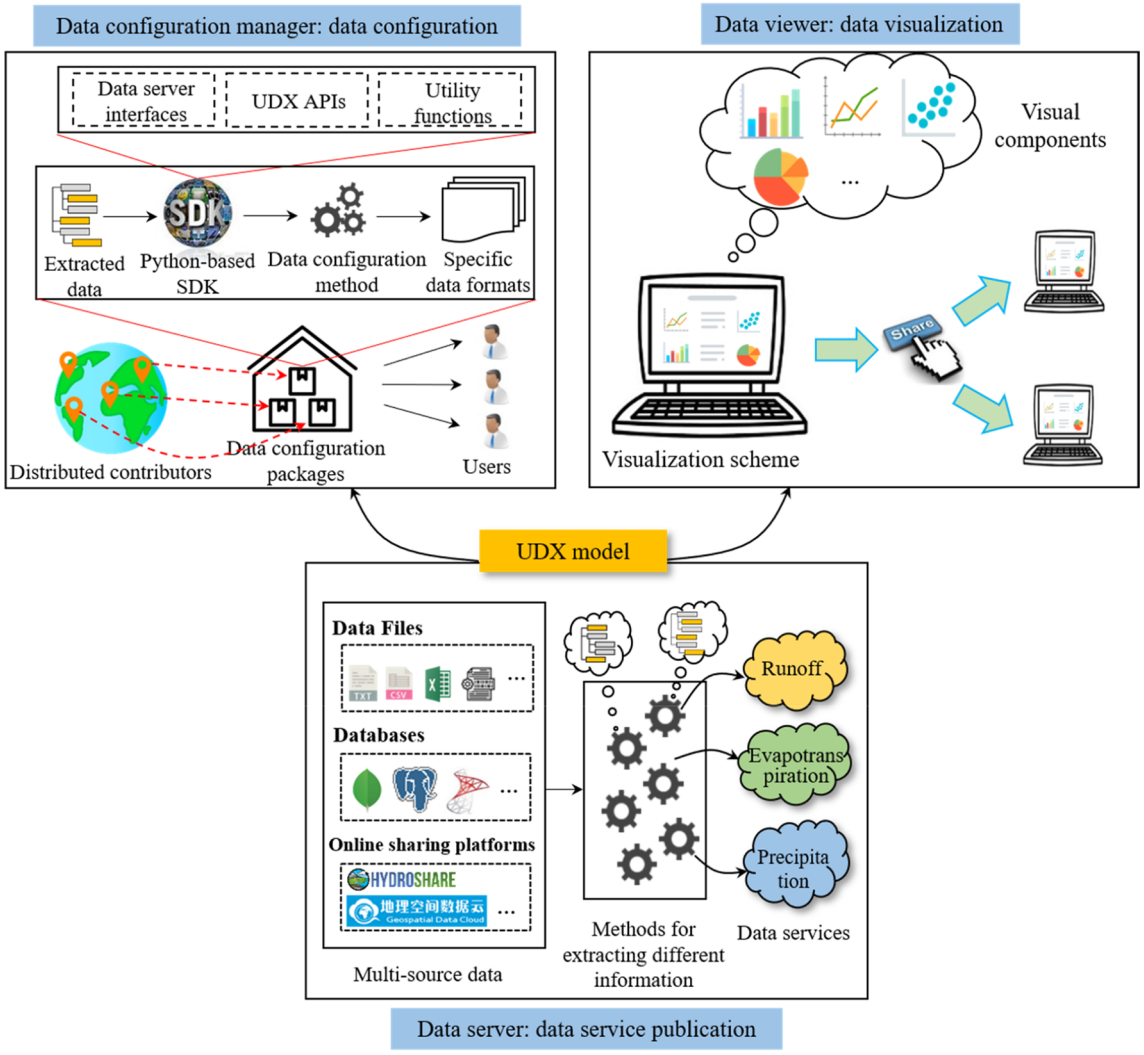

Fig. 1. Conceptual framework of the proposed data sharing method.

Generally, users must identify and extract the required data from these data sources through different access approaches, which makes it hard to reuse the data access logics. In this article, data servers are designed for publishing the information contained in multisource data sets as data services to provide a unified approach for users to access the required data. Additionally, data extraction methods are developed to extract specific information from raw data sources and publish the information via data services. For example, the runoff data extraction method, evaporation data extraction method and precipitation extraction method can be used to extract corresponding information from a climate data source. Users can selectively request the required data by invoking the corresponding extraction methods without considering the data organizational structure of the raw data sources. Compared to the process in upload/download mode, users can access the required data without downloading the files in this approach, which avoids unnecessary manual data manipulations of raw data files.

Due to the various data requirements of different applications, the data requested from data services should be reprocessed before use in specific applications. The current practices for processing data are mostly associated with application-specific data manipulations that are difficult to perform multiple times. In this article, the information provided by data services is described by the UDX model, which can easily be operated with UDX Application Programmers Interfaces (APIs). Considering the flexibility and independence of data processing, the data configuration manager designs a Python-based software development kit (SDK) for users to customize their own data requirements. The SDK includes data server interfaces, UDX APIs and utility functions, which provide the standard interfaces for data configuration. A data configuration process can be packaged as a data configuration package that can be reused by others. In this design approach, the data configuration package can be developed by distributed contributors. Data configuration managers can manage these packages and provide an open platform for users to reuse and share data configuration packages, which avoids duplicating data configuration efforts.

Data visualization is a key step in hydrological modeling and simulation and provides a direct way to visualize hydrological data (Kao et al., 2011; Horsburgh and Reeder, 2014; Jones et al., 2016). For both observation data and model output data, there is a need to visualize the information associated with these data. Different information types require different visualization methods. For example, observation sites can be visualized with 2D maps, and runoff data can be visualized with charts, such as bar graphs and line graphs. There is not a one-size-fits-all visualization method for all types of hydrological information. The data viewer provides a component-based data visualization method for visualizing multiple types of information in a visualization scheme. Each component can implement a kind of visualization requirement, and several visual components can be combined to form a visualization scheme. The data viewer also supports sharing the visual scheme with the click of a button. With the data viewer, the data visualization methods can be reused, which avoids the need for tedious efforts for

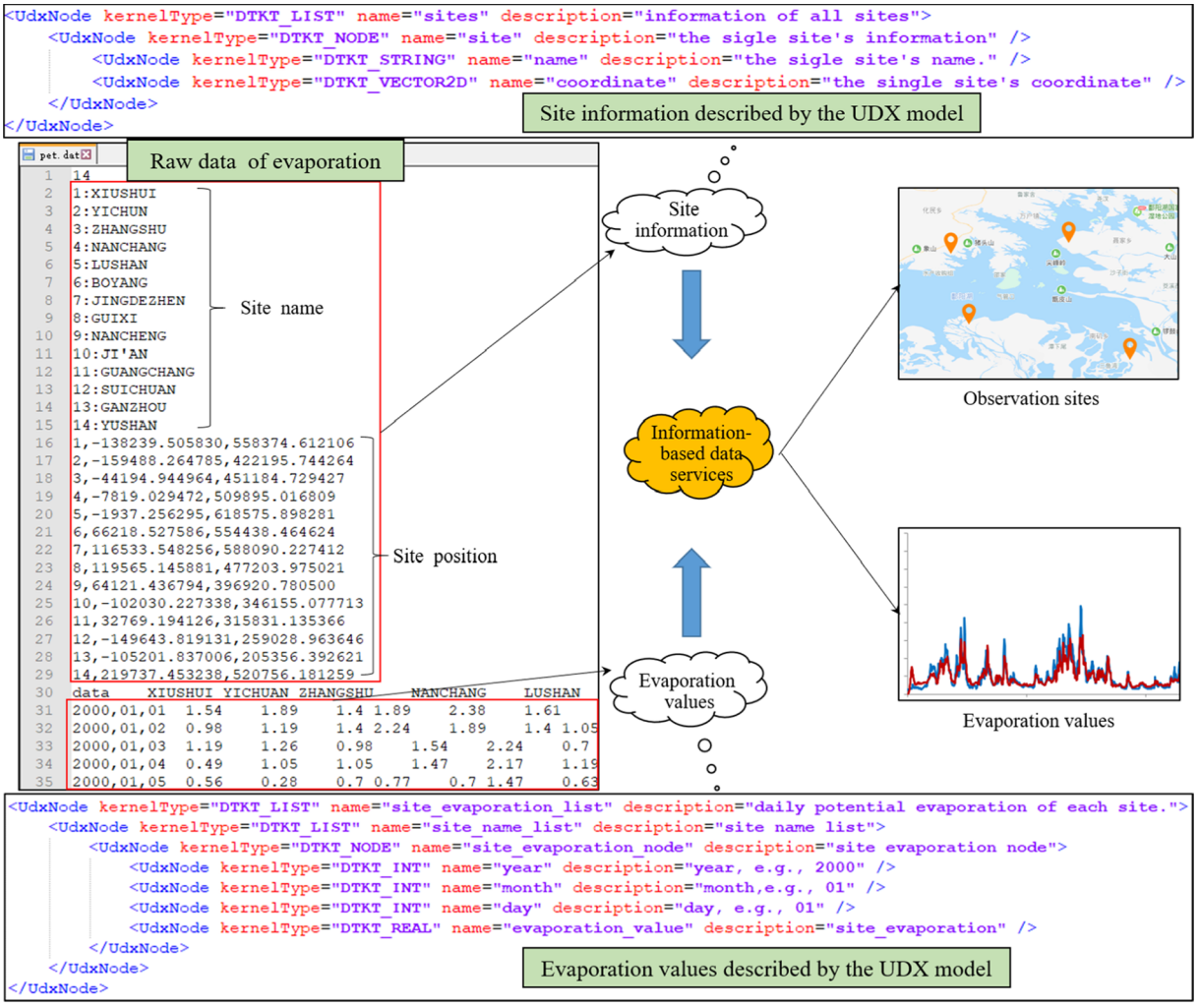

Fig. 2. The raw evaporation data expressed by the UDX model.

visualization customization.

In summary, the proposed data sharing method decouples the processes of data publication, data configuration and data application, which enables these three subprocesses to be independently developed; thus, they can be flexibly combined to meet the data requirements of various applications.

3. Implementation of the data sharing method

This section introduces the framework design for the data sharing method. Yue et al. (2019) proposed the UDX model for describing the heterogenous data in a structured way, which provides a common data view through which users can understand the data. Wang et al. (2018) employed a UDX model to describe the data requirements of models and provided a series of methods for operating UDX data (e.g., mapping between raw data and UDX data and transforming different types of UDX data), finally producing specific UDX data for driving models. However, this article focuses on a data sharing method that helps users easily access the hydrological information contained in raw data resources. The UDX model is employed as a medium for describing specific hydrological information, which provides the foundation for users to customize their data requirements and conduct visualization analysis. The UDX data operation mechanisms were introduced in previous studies; thus, this article will not cover these topics.

3.1. Data service publication

In a comprehensive hydrological simulation, the required data usually come from multiple data sources. These data sources may be shared through different data sharing methods, requiring time-intensive efforts for user access. The commonly used data sources in hydrology are summarized below.

(1) Data files. A data file is the storage medium for most data. In hydrology, data is usually stored in common data description specifications or proprietary file formats, as mentioned in Section 1. Although most of these data formats are expressed as data files, the data organization structures are different. For the common data formats, there are generally several specialized parsing libraries to read/write the data, such as the Geospatial Data Abstraction Library/OGR Simple Features Library (GDAL/OGR, http://www.gdal.org/), which is used to read raster and vector data. NetCDF provides corresponding libraries (https://www.unidata.ucar.edu/downloads/netcdf/index.jsp) for data reading and extraction. For proprietary file formats, users need to manually write code to read the data and extract the required information according to the data organizational structure.

(2) Databases. Due to the powerful data management and retrieval capabilities of databases, geographic data are usually stored in databases. Different types of databases are suitable for storing different types of information. Data that are associated with data fields

are usually stored in a relational database, such as a PostgreSQL (https://www.postgresql.org/), Oracle (https://www.oracle.com/index.html) or MySQL (https://www.mysql.com/) database. In hydrology, ODM and WaMDaM use a relational database to store hydrological data. Nonrelational databases are typically used to store "big data" that require a rapid response, and such databases include MongoDB (https://www.mongodb.com/), HBase (https://hbase.apache.org/) and Neo4j (https://neo4j.com/). Generally, the Structured Query Language (SQL) is used to query data stored in databases.

(3) Data provided by online data sharing platforms. Online sharing platforms are among the most important data sources for scientific research. In addition to the web-based data sharing platforms introduced in Section 1, there are several online real-time data sharing platforms available, such as that operated by the Water Resources Department of Jiangxi Province, China (http://www.jxsl.gov.cn/slxxhw/jhsq/), which provides real-time rainfall and water level data. The National Urban Air Quality Real-time Release Platform of China (http://106.37.208.233:20035/) provides real-time air monitoring data. Some platforms provide file-based data sources that require users to download data files, while others provide interfaces through which users can request data. For the latter, users must request a data stream through a Hypertext Transport Protocol (HTTP) request and then extract the required information.

Based on the above analysis, whether writing code to parse customized data formats or invoking APIs to extract specific information, these data manipulation processes are related to specific data sources and are difficult for others to reuse. Because data sources are heterogeneous and variable, the information contained in data sources can be extracted and published via data services, which would reduce the difficulty of accessing data. There are three steps in the data service publication process: information identification, data extraction and description, and data publication, as shown in Fig. 3.

First, specific hydrological information can be identified from raw data sources. In Fig. 3, for example, land use, evaporation and soil depth information are stored in data files with different formats. The river flow and water level data are from online data sharing platforms. The precipitation and dew point temperature data are stored in different databases. Different users may require different hydrological information for various applications; thus, they need to identify the required data according to the metadata descriptions of these data sources. Second, the information is extracted and described via the UDX model. Fig. 4 shows the process of extracting information by the proposed data extraction method. Different data sources can employ different data reading/writing interfaces to extract specific information, as mentioned above. Then the extracted data can be described by the UDX model and operated by UDX APIs. Through the data extraction method, users can request different types of hydrological data and different levels of information. The method can be developed using mainstream programming languages (such as Java, C++, and Python) with the corresponding versions of the UDX APIs developed in this article (https://gitee.com/OGMS/UdxAPIs.git). In addition, the data extraction methods can be reused by different users, which reduces the duplicated labor required for accessing the data. Third, the information is published via a data service in the data server. Through data services, users can access the required hydrological information without considering the raw data sources.

The data server is a lightweight server for hosting data services and data extraction methods in the open web environment. Users can deploy the data server on their own servers and upload their data sources and corresponding data extraction methods to establish data services for sharing their data. Then, others can invoke the data extraction method to request the required data from the data server.

3.2. Data configuration

3.2.1. Python-based SDK

The data configuration SDK provides reusable data manipulation tasks for users to customize their data requirements for specific applications. Python (https://www.python.org/) is a widely used object-oriented language that provides a flexible and concise architecture for efficient processing. In this article, the data configuration SDK is based on Python and includes three parts, as shown in Fig. 5.

Data server interfaces are designed to establish an information exchange channel between data servers and applications. "UdxServer.py" mainly provides a connection to a data server and obtains the data service list from that data server. "DataService.py" represents the basic metadata of a data service, such as the specific hydrologic information that the data service can provide. "ExtractMtd.py" contains the code invoking the data extraction method hosted on the data server and returns the required data. "Config.py" maintains the global variable and

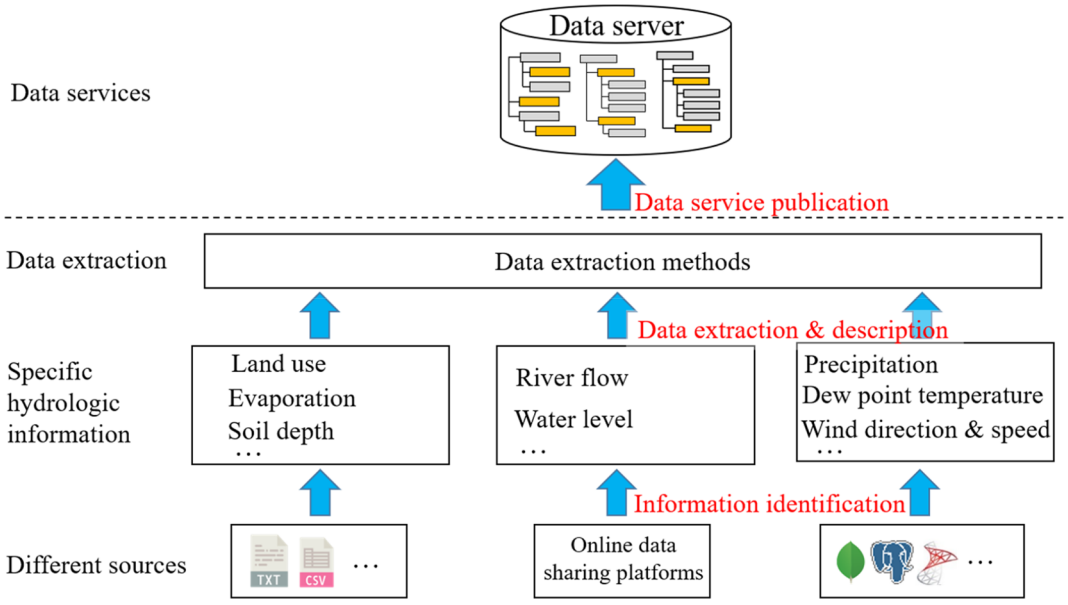

Fig. 3. The process of data service publication.

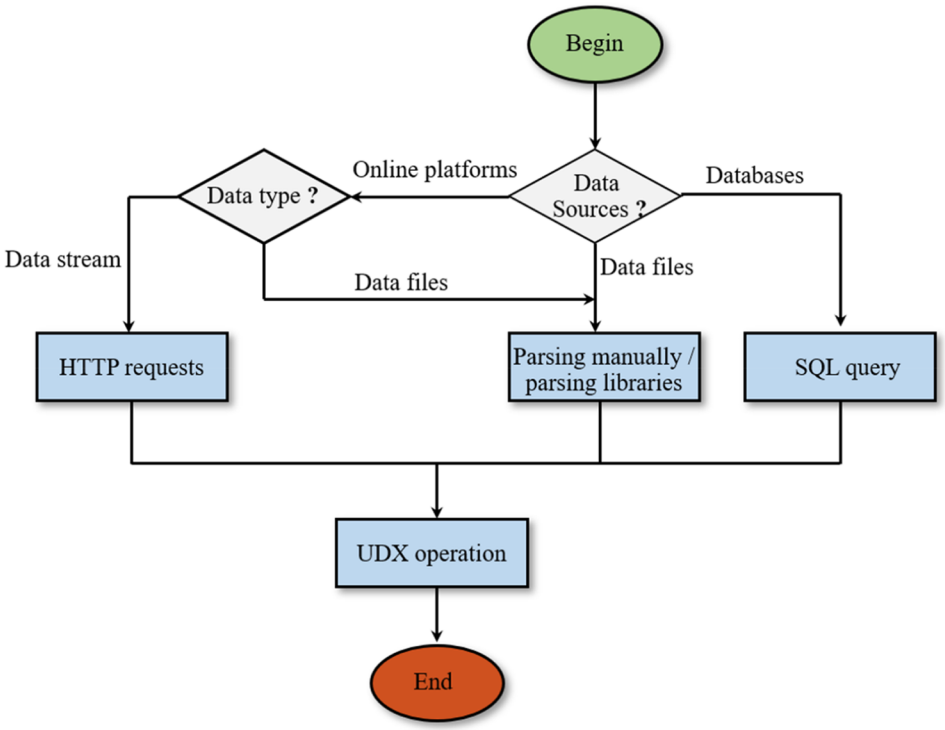

Fig. 4. The process of extracting information by the data extraction method.

configuration information of the SDK.

The auxiliary functions contain three modules, which provide practical functionalities that can significantly improve the efficiency of data configuration. In the proposed data sharing method, user-related data is stored in their own "user space" in the data server. The "FileIO" module provides interfaces for writing data files into their "user space" or reading data files from their "user space". To improve the inter-operability of the UDX model and Python, the "DataTypeConvertor" module provides interfaces for information exchange between UDX data and Python data structures. For example, the function "toPythonList(udxType)" can easily transform the "real_array" type in UDX data to the "list" type in Python. The "Utils" module includes several utility functions, such as math-related functions and functions for data processing.

The UDX APIs contain the interfaces for UDX data operation, as shown in Table 1. These APIs enable flexible operations for UDX data nodes, which can be classified into two categories: operations related to the UDX node structures and operations related to the UDX node values. The former can locate a specific data node or change the UDX data structure to form a new one. The latter allows users to read/write data node values, such as by implementing reprojection or data calibration processes. With these APIs, users can flexibly operate UDX data nodes, which is a more efficient process compared to operations involving raw data files.

3.2.2. *Data configuration manager*

The data configuration SDK provides the "standard" interfaces for accessing data services and operating the requested data. To reduce the difficulty of data configuration and reuse data configuration methods, a data configuration manager based on the data configuration SDK is designed in this article, as shown in Fig. 6. The data configuration manager has three functionalities: providing an interactive programming workspace for data configuration, managing data configuration packages, and providing customized data for various applications (e.g., model invocation, data analysis and data visualization).

Jupyter Notebook (https://jupyter.org) is a web app that provides

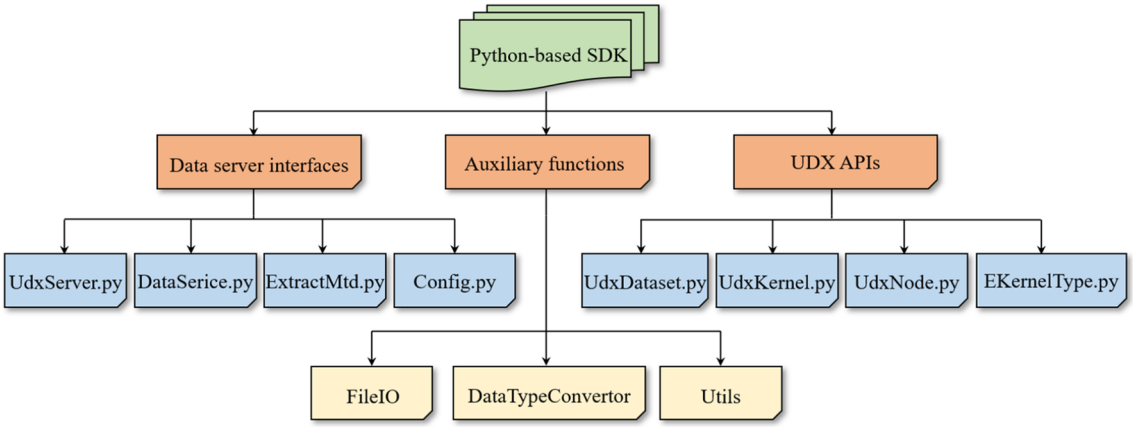

Fig. 5. The design of the Python-based SDK structure.

Table 1
The primary operation APIs of the UDX model.

Category	Reading/writing interfaces	UDX APIs
UDX node structure operation	Reading interfaces	UdxNode get_node_by_name(string name) UdxNode get_child_node_by_index(int index) int get_child_node_count() int get_node_length()
	Writing interfaces	UdxNode add_child_node(string name, UdxKernelType type) bool remove_child_node(UdxNode node)
UDX node value operation	Reading interfaces	EKernelType get_typed_value() EKernelType get_typed_value_by_index()
	Writing interfaces	bool add_typed_value(EKernelType value) bool set_typed_value_by_index(EKernelType value, int index)

an interactive programming environment for editing, testing and running Python (as well as R, ruby, and C++) code in real time. Because data configuration tasks are labor- and time-consuming and are difficult to repeat, the Jupyter Notebook is introduced, which allows users to customize their data requirements online and then publish the data configuration methods as data configuration packages to be reused by others. To customize specific data requirements, users must follow three steps (data requesting, data operation and data saving) based on the SDK. Here, an example is given to illustrate the process of data configuration. To extract the specific study area from DEM data, polygon Shapefile data can be used to clip the area. The DEM data and the Shapefile data can be requested from the data server using data server interfaces. Then the clipping operation can be implemented by a third-party data-processing library, such as the Python version of the GDAL. The GDAL can be installed by users in the Jupyter programming environment. Technically, users can install any third-part library to implement their configuration tasks. Finally, the output files (i.e., the specific study area) of the data-clipping task are stored on the data server and can be downloaded directly for specific applications. According to users' data application requirements, the output files can be saved as any data format. For example, if the application can receive UDX data as input, users can output UDX data (expressed by XML in this article); otherwise, they can output specific data formats to drive their applications. The data configuration process can be saved as a data configuration script, which can be invoked independently in the Python environment. Notably, users can download the data configuration script and run it, as long as the Python environment is installed.

The data configuration scripts can be packaged into a data configuration package for reuse. Any contributors can upload their data configuration packages to the data configuration manager and share them with other users. In addition, these data configuration packages can be integrated into other applications, which is a benefit of the modular design concept.

3.3. Data visualization

Data visualization is one of the important steps in hydrological analyses. Additionally, data visualization is an effective way to examine

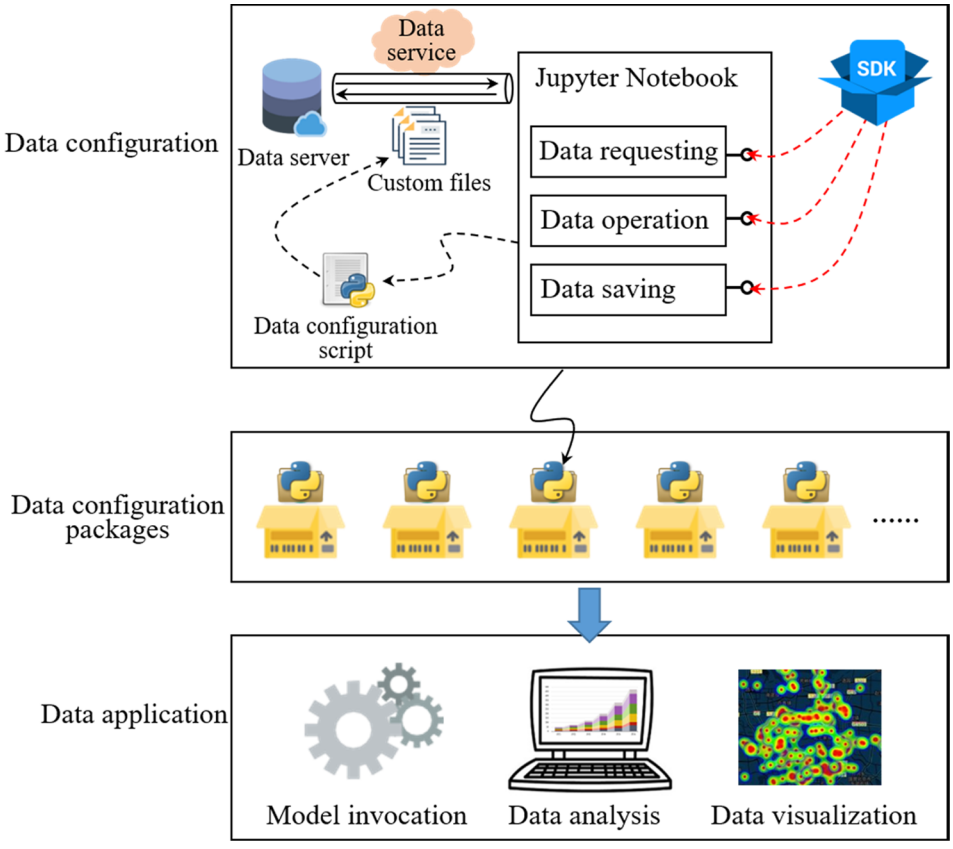

Fig. 6. The design of the data configuration manager.

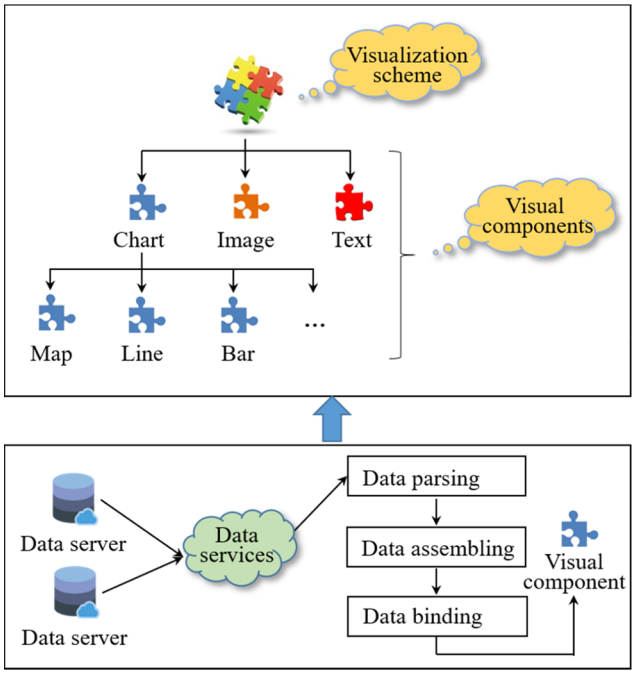

Fig. 7. The design of the data viewer.

simulation results. The different hydrological information contained in hydrological data sets must be visualized in different styles. Although there are many excellent web visualization libraries and tools available, no single platform can meet all potential visualization requirements. Generally, users must customize data visualization methods for specific visualization requirements. However, due to the diversity of hydrological applications, visualization customization results are often difficult to reuse. The data viewer provides a component-based data visualization method to enable multiple types of hydrological information to be visualized in a visualization scheme, as shown in Fig. 7.

The data services provided by distributed data servers are employed as input data for the data viewer. Inside each visual component, the data provided by data services can be parsed into detailed visual elements, such as single data values, one-dimensional arrays and multi-dimensional arrays. These elements can be assembled as specific data structures and then associated with the visual engine of the current visual component. For example, the "bar component" may employ a two-dimensional array as the input. Inside the component, the two-dimensional data are parsed from incoming data and then assembled in the data format that the visual engine supports. Finally, a two-dimensional array with a specific data format will be bound to the "bar chart visual engine". Many visualization libraries and tools, such as Echarts (https://www.echartsjs.com/zh/index.html), D3.js (https://d3js.org/), Leaflet (https://leafletjs.com/), and Three.js (https://threejs.org/), can be encapsulated into visual components as visual engines to drive data visualization. The visualization capability of the visual component is dependent on its developers. Technically, developers can customize arbitrarily complex visual components, as long as they conform to the designed data exchange interfaces.

The visualization component is the basic component of the data viewer, and each component can visualize one specific type of information. There are three types of visual components designed in the data viewer: chart components, image components and text components. A chart component is used to visualize data, and image and text components are used to provide supplementary information for the visualization scheme. According to the visualization requirements, multiple visual components can be coupled to form a comprehensive visualization scheme, which can display multiple information types in one visualization view. For example, in a hydrological simulation, the output data include river flow, evaporation, and runoff information. The "bar component" can be used to display the river flow, and the "line component" can be used to visualize river runoff information. The image component and the text component can provide descriptive information about the simulation. These related visual components can form a visualization scheme of the simulation results, which can be easily shared with others with a keystroke.

4. Case study

In this section, a case study that includes hydrological data sharing, configuration and visualization is introduced to examine the practicality of the proposed data sharing method. In Section 4.1, hydrological data from different sources are published as services on the data server. Section 4.2 focuses on data configuration for driving the Water Flow for Lake Catchments (WATLAC) model, and the simulated results that contain different types of hydrological information are then visualized in a visualization scheme. The prototype system designed in this case study can be accessed at https://ogms.gitee.io/datasharing/.

4.1. Hydrological data publication

Poyang Lake is the largest freshwater lake in China (Jiangxi Province). Many hydrological studies have been conducted in the Poyang Lake basin to prevent the ecological environment from deteriorating in recent years. In this section, we publish a series of related hydrological data on a data server to provide fast access to these data for hydrological simulations. As shown in Fig. 3 in Section 3.1, these data come from different sources. The land use, evaporation and soil depth data are stored in data files. The Water Resources Department of Jiangxi Province (http://www.jxsl.gov.cn/slxxhw/jhsq/) provides real-time river flow and water level data at different sites in Jiangxi Province. Additionally, MongoDB provides annual precipitation data, dew point temperature data, and wind data for China, as well as other data sets (from 2010 to 2018).

In Fig. 8, observation site information and precipitation data (stored in MongoDB) publication are taken as an example. Three data extraction methods ("SiteInfoExtraction", "PrecipitationExtractionBySitePerYear" and "PrecipitationExtractionByYearPerSite") are developed for extracting the site data and precipitation data from MongoDB, and then the extracted data are described by the UDX model. Subsequently, these methods are published to the data server, which can be accessed by others. Users can simply call the data extraction methods to acquire the corresponding data without considering the sources of the data. In this case, the data extraction methods are developed in Python, and the Python versions of the interfaces are used to access MongoDB and operate the UDX model.

4.2. Data configuration and data visualization

4.2.1. Data configuration

The WATLAC model was developed by the Nanjing Institute of Geography & Limnology, Chinese Academy of Sciences (Zhang and Li, 2009; Ye et al., 2011). As shown in Fig. 9, the model mainly focuses on the simulation of surface, soil and underground hydrological processes. In this section, the WATLAC model is employed to simulate hydrological process in the Poyang Lake basin and changes in the volume of lake water. The model is published on the OpenGMS platform (Chen et al., 2011, 2019; Wen et al., 2013, 2017; Yue et al., 2015, 2016; Zhang et al., 2019), which can be accessed online (http://geomodeling.njnu.edu.cn/modelItem/e4f3f129-de8f-4721-96db-8c50e8054fa5).

There are three types of input data (geospatial data, meteorological data and groundwater data) for the model. In Section 4.1, we have published some of these data as data services (land use, soil data, site position, rainfall and evaporation data). In this section, these data

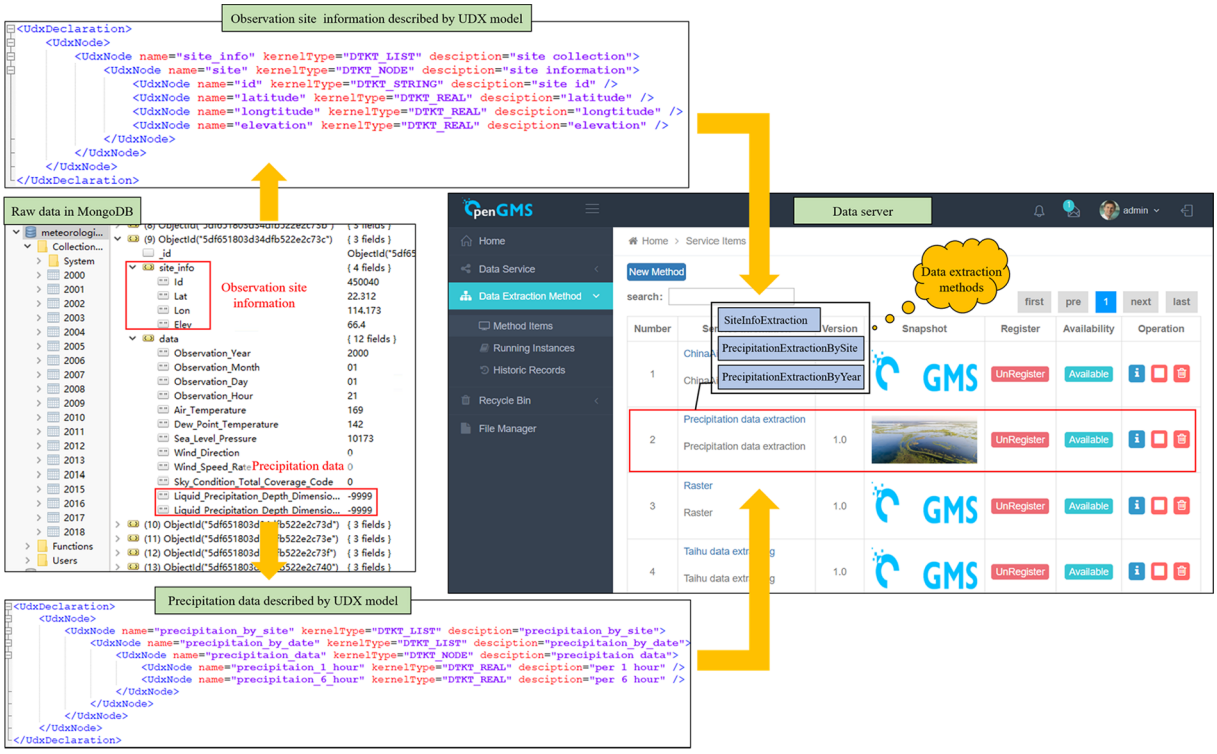

Fig. 8. Example of site information and precipitation data publication.

services will be used to configure specific data formats for driving the WATLAC model. In Fig. 10, evaporation data configuration is taken as an example. The site position and evaporation data are requested from the data server and are employed to customize the input data for driving the WATLAC model, as shown in the left panel. The data configuration scripts can be published as data configuration packages that can be shared with others in the data configuration manager, as shown in the upper right panel. In this section, the land use, soil, site position, rainfall and evaporation data are configured in the data configuration manager. The data files generated by the data configuration methods can be used to drive the WATLAC model directly.

4.2.2. Data visualization

In this study, we have designed several visual components in the data viewer, such as a 2D map component, line chart component and pie chart component. In Fig. 10, the lower right panel shows the data

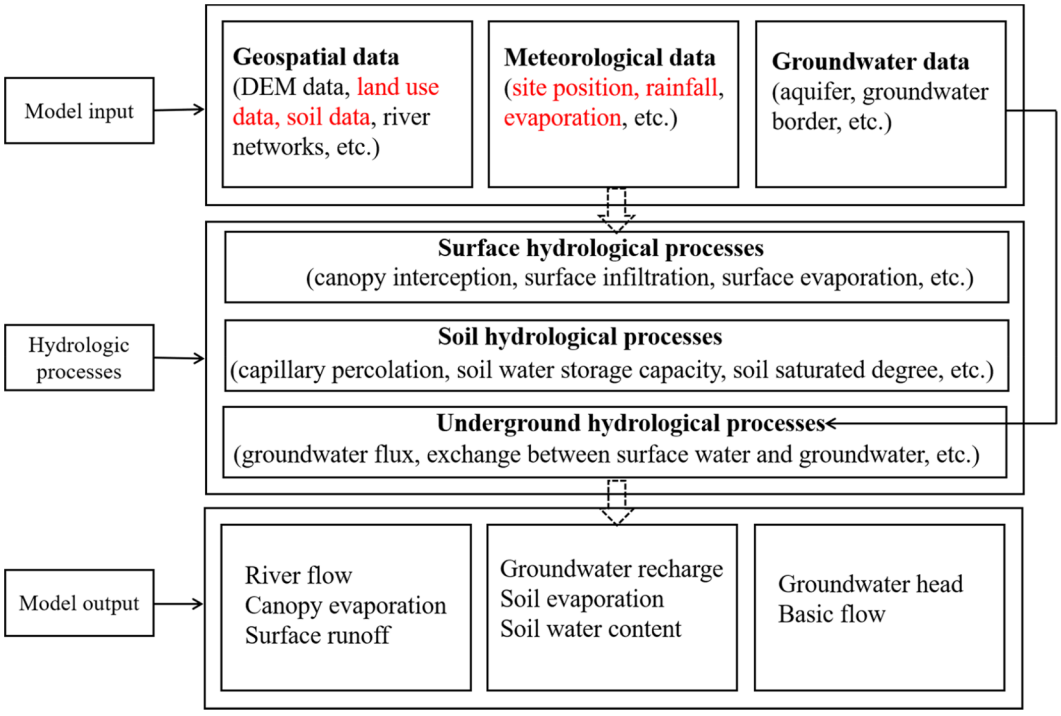

Fig. 9. The framework of the WATLAC model.

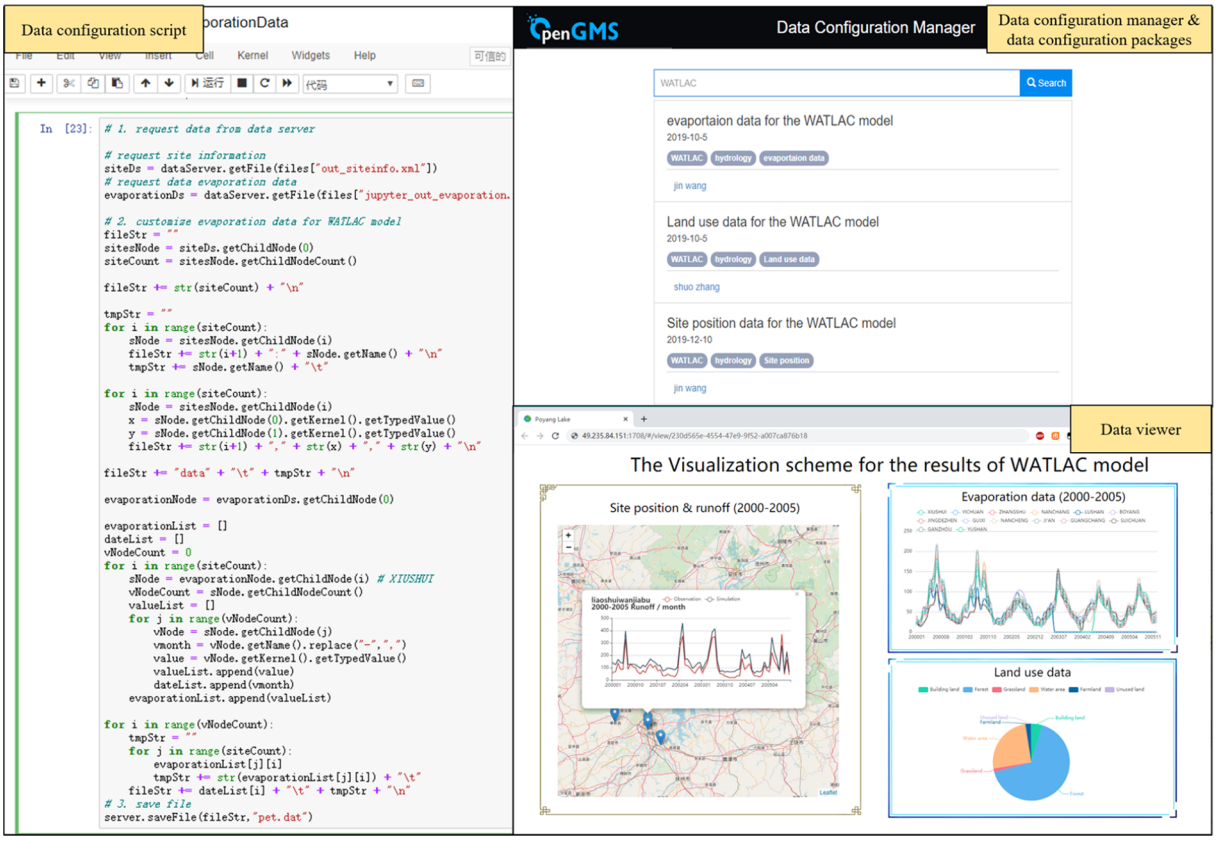

Fig. 10. Example of data configuration and data visualization.

viewer, which visualizes a portion of the output results from the WATLAC model, such as the site position, runoff, land use and evaporation. In the 2D map component, the line chart containing the runoff data is displayed when the user clicks the "blue markers", which refer to site positions. The evaporation data at each site is displayed in the line chart, and the land use data is displayed in the pie chart component.

With the data configuration manager, users can easily reuse the packages contributed by others, which reduces the difficulty of preparing data for hydrological simulations. The data viewer also provides a flexible and effective way to visualize multiple information types using visual components, which enables the possibility of sharing visualization methods.

5. Conclusions, discussions and future work

The purpose of the proposed data sharing method is to improve the usability of hydrological data resources. This method provides the following specific contributions to data sharing studies. First, this study helps users gain easy access to their required data through data services provided by the data server. The designed data configuration manager and data viewer provide effective methods for users to configure their data requirements and visualize the data, which could increase data usability. Moreover, the proposed data sharing method can separate the whole data sharing process into three subprocesses (i.e., data publication, data configuration and data application), which would lead to the improvement of sharing and interoperability among distributed data resources.

Although the WATLAC model is employed to examine the feasibility of the proposed data sharing method, the shared data can also be applied to other models and applications. On one hand, the designed framework enables the public to participate in the data sharing process. In the foreseeable future, not only data resources but also data processing methods could be shared in the framework. Massive data resources (data and corresponding processing methods) provide the possibility of customizing specific data requirements. On the other hand, the UDX model is employed to provide a common data view through which users can understand the data. Users can focus on their own data requirements, rather than on complex data parsing and transformation works. These works can be implemented in data extraction methods or third-party data processing libraries provided by professionals. In this way, the framework simplifies the data preparation process required for users to drive specific applications. In short, the proposed data sharing method has improved the usability of shared data, which would improve the efficiency of hydrological modeling and simulation.

The following issues are worth exploring in future studies to support comprehensive hydrological modeling and simulation.

(1) High-level interfaces are required to improve the efficiency of data manipulation tasks. The underlying interfaces designed in the data configuration SDK require large chunks of code for manipulating the UDX data, which is not straightforward for data configuration package contributors.
(2) Reusable and automated data manipulation tools need to be developed to improve the efficiency of data processing. Time-consuming data manipulations are unavoidable in using shared data. Although the proposed data sharing method provides a more efficient way of reusing data manipulation efforts, the processes of data extraction and configuration are still time- and labor-intensive.
(3) Robustness of the data viewer is necessary. Because hydrological data are usually "big data", their visualization is generally difficult in a web application. Although specific visual components can be developed for data according to their specific types, the method still has its maximum limit for visualization. To handle "big data", a series of data-reduction technologies are necessary for efficient data visualization, such as the data compression, level of detail (LOD)

and progressive transmission strategies for data.

CRediT authorship contribution statement

Jin Wang: Methodology, Writing - original draft, Software, Writing - review & editing. **Min Chen:** Conceptualization, Funding acquisition. **Guonian Lü:** Supervision. **Songshan Yue:** Investigation, Resources. **Yongning Wen:** Project administration. **Zhenxu Lan:** Visualization. **Shuo Zhang:** Data curation, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We appreciate the detailed suggestions and comments from the secretariat and the anonymous reviewers. A special thanks to Qi Zhang (State Key Laboratory of Lake Science and Environment, Nanjing Institute of Geography and Limnology) for providing the WATLAC model in the study case. We also express heartfelt thanks to the other members of the OpenGMS team (http://opengmsteam.com/). This work was supported by the NSF for Excellent Young Scholars of China under Grant number 41622108, the Natural Science Foundation of China (Grant No. 41701441 and U1811464), and Priority Academic Program Development of Jiangsu Higher Education Institutions under Grant 164320H116.

References

Abdallah, A.M., Rosenberg, D.E., 2019. A data model to manage data for water resources systems modeling. Environ. Modell. Softw. 115, 113–127.
Ames, D.P., Horsburgh, J.S., Cao, Y., Kadlec, J., Whiteaker, T., Valentine, D., 2012. HydroDesktop: Web services-based software for hydrologic data discovery, download, visualization, and analysis. Environ. Modell. Softw. 37, 146–156.
Beran, B., Piasecki, M., 2009. Engineering new paths to water data. Comput. Geosci. 35, 753–760.
Chen, M., Sheng, Y., Wen, Y., Su, H., 2009a. Geographic problem-solving oriented data representation model. J. Geo-Inf. Sci. 11, 333–337.
Chen, M., Sheng, Y., Wen, Y., Tao, H., Guo, F., 2009b. Semantics guided geographic conceptual modeling environment based on icons. Geogr. Res. 28, 705–715.
Chen, M., Tao, H., Lin, H., Wen, Y., 2011. A visualization method for geographic conceptual modelling. Ann. Gis 17, 15–29.
Chen, M., Yue, S., Lü, G., Lin, H., Yang, C., Wen, Y., Hou, T., Xiao, D., Jiang, H., 2019. Teamwork-oriented integrated modeling method for geo-problem solving. Environ. Modell. Softw. 119, 111–123.
Demeritt, D., Wainwright, J., 2005. Models, Modelling, and Geography. Questioning Geography: Fundamental Debates. Blackwell Publishing, Oxford, UK, pp. 206.
Gogu, R., Carabin, G., Hallet, V., Peters, V., Dassargues, A., 2001. GIS-based hydrogeological databases and groundwater modelling. Hydrogeol. J. 9, 555–569.
Granell, C., Schade, S., Ostländer, N., 2013. Seeing the forest through the trees: a review of integrated environmental modelling tools. Comput. Environ. Urban Syst. 41, 136–150.
Han, W., Di, L., Zhao, P., Shao, Y., 2012. DEM Explorer: an online interoperable DEM data sharing and analysis system. Environ. Modell. Softw. 38, 101–107.
Han, W., Yang, Z., Di, L., Yagci, A.L., Han, S., 2014. Making cropland data layer data accessible and actionable in GIS education. J. Geogr. 113, 129–138.
Horsburgh, J.S., Aufdenkampe, A.K., Mayorga, E., Lehnert, K.A., Hsu, L., Song, L., Jones, A.S., Damiano, S.G., Tarboton, D.G., Valentine, D., Zaslavsky, I., Whitenack, T., 2016a. Observations Data Model 2: a community information model for spatially discrete Earth observations. Environ. Modell. Softw. 79, 55–74.
Horsburgh, J.S., Morsy, M.M., Castronova, A.M., Goodall, J.L., Gan, T., Yi, H., Stealey, M.J., Tarboton, D.G., 2016b. Hydroshare: sharing diverse environmental data types and models as social objects with application to the hydrology domain. J. Am. Water Resour. Assoc. 52, 873–889.
Horsburgh, J.S., Reeder, S.L., 2014. Data visualization and analysis within a hydrologic information system: integrating with the R statistical computing environment. Environ. Modell. Softw. 52, 51–61.
Horsburgh, J.S., Tarboton, D.G., Maidment, D.R., Zaslavsky, I., 2008. A relational model for environmental and water resources data. Water Resour. Res. 44.
Horsburgh, J.S., Tarboton, D.G., Piasecki, M., Maidment, D.R., Zaslavsky, I., Valentine, D., Whitenack, T., 2009. An integrated system for publishing environmental observations data. Environ. Modell. Softw. 24, 879–888.
Jones, A.S., Horsburgh, J.S., Jackson-Smith, D., Ramírez, M., Flint, C.G., Caraballo, J., 2016. A web-based, interactive visualization tool for social environmental survey data. Environ. Modell. Softw 84, 412–426.
Kadlec, J., StClair, B., Ames, D.P., Gill, R.A., 2015. WaterML R package for managing ecological experiment data on a CUAHSI HydroServer. Ecol. Inform. 28, 19–28.
Kao, S., Ranatunga, K., Squire, G., Pratt, A., Dee, D., 2011. Visualisation of hydrological observations in the water data transfer format. Environ. Modell. Softw. 26, 1767–1769.
Laituri, M., Sternlieb, F., 2014. Water data systems: science, practice, and policy. J. Contemporary Water Res. Educ. 153, 1–3.
Lü, G., Batty, M., Strobl, J., Lin, H., Zhu, A.X., Chen, M., 2019. Reflections and speculations on the progress in Geographic Information Systems (GIS): a geographic perspective. Int. J. Geogr. Inf. Sci. 33, 346–367.
Maidment, D.R., 2016. Open water data in space and time. J. Am. Water Resour. Assoc. 52, 816–824.
Markert, K.N., Pulla, S.T., Lee, H., Markert, A.M., Anderson, E.R., Okeowo, M.A., Limaye, A.S., 2019. AltEx: an open source web application and toolkit for accessing and exploring altimetry datasets. Environ. Modell. Softw. 117, 164–175.
McDonald, S., Mohammed, I.N., Bolten, J.D., Pulla, S., Meechaiya, C., Markert, A., Nelson, E.J., Srinivasan, R., Lakshmi, V., 2019. Web-based decision support system tools: the Soil and Water Assessment Tool Online visualization and analyses (SWATOnline) and NASA earth observation data downloading and reformatting tool (NASAaccess). Environ. Modell. Softw. 120, 104499.
Miller, R.C., Guertin, D.P., Heilman, P., 2004. Information technology in watershed management decision making 1. J. Am. Water Resour. Assoc. 40, 347–357.
Morsy, M.M., Goodall, J.L., Castronova, A.M., Dash, P., Merwade, V., Sadler, J.M., Rajib, M.A., Horsburgh, J.S., Tarboton, D.G., 2017. Design of a metadata framework for environmental models with an example hydrologic application in HydroShare. Environ. Modell. Softw. 93, 13–28.
Peng, Z.R., 2005. A proposed framework for feature-level geospatial data sharing: a case study for transportation network data. Int. J. Geogr. Inf. Sci. 19, 459–481.
Voinov, A., Cerco, C., 2010. Model integration and the role of data. Environ. Modell. Softw. 25, 965–969.
Wang, J., Chen, M., Lü, G., Yue, S., Chen, K., Wen, Y., 2018. A study on data processing services for the operation of geo-analysis models in the open web environment. Earth Space Sci. 5, 844–862.
Wen, Y., Chen, M., Lu, G., Lin, H., He, L., Yue, S., 2013. Prototyping an open environment for sharing geographical analysis models on cloud computing platform. Int. J. Digit. Earth. 6, 356–382.
Wen, Y., Chen, M., Yue, S., Zheng, P., Peng, G., Lu, G., 2017. A model-service deployment strategy for collaboratively sharing geo-analysis models in an open web environment. Int. J. Digit. Earth. 10, 405–425.
Whitenack, T., 2010. CUAHSI HIS CENTRAL 1.2. Technical report, Consortium of Universities for the Advancement of Hydrologic. Science Inc.
Xue, Z., Couch, A., Tarboton, D., 2019. Map based discovery of hydrologic data in the HydroShare collaboration environment. Environ. Modell. Softw. 111, 24–33.
Ye, X., Zhang, Q., Bai, L., Hu, Q., 2011. A modeling study of catchment discharge to Poyang Lake under future climate in China. Quat. Int. 244, 221–229.
Yue, S., Chen, M., Wen, Y., Lu, G., 2016. Service-oriented model-encapsulation strategy for sharing and integrating heterogeneous geo-analysis models in an open web environment. ISPRS-J. Photogramm. Remote Sens. 114, 258–273.
Yue, S., Chen, M., Yang, C., Shen, C., Zhang, B., Wen, Y., Lü, G., 2019. A loosely integrated data configuration strategy for web-based participatory modeling. GISci. Remote Sens. 56, 670–698.
Yue, S., Wen, Y., Chen, M., Lu, G., Hu, D., Zhang, F., 2015. A data description model for reusing, sharing and integrating geo-analysis models. Environ. Earth Sci. 74, 7081–7099.
Zagona, E.A., Fulp, T.J., Shane, R., Magee, T., Goranflo, H.M., 2001. Riverware: A generalized tool for complex reservoir system modeling 1. J. Am. Water Resour. Assoc. 37, 913–929.
Zhang, F., Chen, M., Ames, D.P., Shen, C., Yue, S., Wen, Y., Lü, G., 2019. Design and development of a service-oriented wrapper system for sharing and reusing distributed geoanalysis models on the web. Environ. Modell. Softw. 111, 498–509.
Zhang, C., Li, W., Zhao, T., 2007. Geospatial data sharing based on geospatial semantic web technologies. J. Spat. Sci. 52, 35–49.
Zhang, Q., Li, L., 2009. Development and application of an integrated surface runoff and groundwater flow model for a catchment of Lake Taihu watershed. China. Quat. Int. 208, 102–108.
Zhu, Y., Yang, J., 2019. Automatic data matching for geospatial models: a new paradigm for geospatial data and models sharing. Ann. Gis 25, 283–298.
Zhu, Y., Zhu, A.X., Feng, M., Song, J., Zhao, H., Yang, J., Zhang, Q., Sun, K., Zhang, J., Yao, L., 2017. A similarity-based automatic data recommendation approach for geographic models. Int. J. Geogr. Inf. Sci. 31, 1403–1424.