

Article

A Real-Time and Open Geographic Information System and Its Application for Smart Rivers: A Case Study of the Yangtze River

Zeqiang Chen ^{1,2}  and Nengcheng Chen ^{1,2,*} 

¹ State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 129 Luoyu Road, Wuhan 430079, China; ZeqiangChen@whu.edu.cn

² Collaborative Innovation Center of Geospatial Technology, 129 Luoyu Road, Wuhan 430079, China

* Correspondence: cnc@whu.edu.cn; Tel.: +86-27-6877-8969

Received: 18 January 2019; Accepted: 23 February 2019; Published: 28 February 2019



Abstract: The timely sharing and interoperation of multi-source cross-sectoral information is an important issue for a Geographic Information System (GIS). To study this issue, a real-time and open GIS model called GeoSensor is proposed in this work. GeoSensor integrates the real-time GIS model, real-time computation framework, and Open Geospatial Consortium services. This paper illustrates the system architecture and the implementation methods of the GeoSensor. One of the methods developed is the conceptual mapping of a real-time GIS data model to open GIS models and services and a real-time computation framework. The other method developed is the integration of open GIS services, a real-time computation framework, and hybrid databases. The GeoSensor was tested in a case study of building a smart river. In the case study, a comprehensive sensor web was constructed and integrated with the real-time information on rainfall, beacon, channel, sediment, and water levels derived from space-based sensors, air-borne sensors, and underground sensors from multiple sectors in the Yangtze River basin. The GeoSensor manages the comprehensive sensor web with 32 types of sensors and more than 10 billion observation records. Three application systems were developed based on the GeoSensor to manage flood control, hydropower production, and navigation of the Yangtze River. The results of the three application systems show that the real-time and open system improves the management efficiency of the Yangtze River. This system is promising for managing smart rivers.

Keywords: real-time and open GIS; smart rivers; Yangtze River; flood control; hydropower production; navigation

1. Introduction

With the development of information technologies and application requirements, Geographic Information System (GIS) technology is undergoing profound changes. One profound change is the development of real-time GIS. Real-time geographic information is acquired by omnipresent space-based sensors, air-borne sensors, underground sensors, and human observers [1]. Thus, geographic information and its services become real-time, dynamic, and online [2]. Some researchers have claimed that GIS has evolved from static GIS to temporal GIS and then to real-time GIS from the perspective of time [3,4]. Real-time GIS is a type of GIS that can acquire, manage, share, process, and service real-time or near real-time geographic objects/information as well as simulate and process dynamic geographic processes [5]. Goodchild [6] noted that real-time GIS is an important component for the future of GIS modelling. Another profound change is the development of an open GIS. In reality, the requirements of an application are usually very complex.

A GIS application system often involves many resources [2] (for example, data, sensor, model, and system resources). These resources are always heterogeneous, distributed, cross-industrial, cross-sectoral, or even restricted. Traditional file or data format exchange methods can no longer meet the requirements of the sharing and interoperation of resources. Therefore, open GIS has emerged. An open GIS is a GIS established according to open standards and interfaces under a certain computer and communication environment. In this scenario, the mentioned standards and interfaces mainly refer to the specifications defined by the Open Geospatial Consortium (OGC). The OGC standards emerge not only as industry standards but also through intersectoral collaboration among industrial, academia/research, and governmental organizations.

Great efforts have been made toward developing real-time GIS and open GIS. Many existing GIS applications focus on dealing with or analyzing real-time data for an application, such as ice forecasting [7], accident prevention on toll motorways [8], traffic systems [9], high-sulfur gas emergency responses [10], mapping of weather data [11], digital mapping for local governments [12], processing of marine environment data [13], and hydrography applications [14]. Then, real-time models and technologies were adopted to realize certain real-time functions or aspects of a GIS [3,15,16]. The OGC is an international non-profit organization committed to making quality open standards for the global geospatial community. OGC has defined many services and encoding specifications, such as the Web Mapping Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS), Web Processing Service (WPS), Sensor Web Enablement, and Publish/Subscribe interface standards [17]. The OGC Sensor Web is an infrastructure that bridges the gap between sensor resources (sensors, sensor systems, etc.) and their applications using a standard interoperable method that enables sensor resource discovery, access, and tasks in a uniform manner [18]. The OGC Sensor Web Enablement initiatives have developed a set of information models and interface models. Information models define the encoding method of sensor observations by the Observations and Measurements (O&M) conceptual model and sensor metadata by the Sensor Model Language (SensorML) [17]. Interface models define the interfaces of Sensor Web services, including the Sensor Observation Service (SOS) [19], which offers access to sensor observations and metadata, and the Sensor Planning Service, which tasks sensor resources. OGC defines the Publish/Subscribe interface standard to support the Publish/Subscribe message exchange pattern complementing the Request/Reply messaging pattern with OGC Web Services. The OGC specifications are widely used, e.g., by the Infrastructure for Spatial Information in Europe (INSPIRE) program [20] and the Global Earth Observation System of Systems (GEOSS) program [21], and developed, e.g., the 52° North software [22]. These efforts bring us closer to real-time GIS and open GIS, but a real-time and open GIS that integrates both real-time technologies and open specifications in a GIS is seldom developed.

To develop a real-time and open GIS, emerging related information technologies and geospatial information technologies, e.g., real-time GIS data models, real-time computation technology, and OGC service technology may provide us with a new way to think about this issue. Accordingly, real-time GIS data models and real-time computation technologies were investigated.

For a real-time GIS, the challenge is managing real-time data and moving features, dynamic spatiotemporal processes, and online services compared with traditional GIS. A suitable GIS model is at the core of managing these features. A real-time GIS data model was proposed to manage real-time GIS features and their relationships [5]. The characteristics of the data model are as follows: (1) being compatible with traditional GIS models, (2) expressing dynamic data and supporting the dynamic simulation of spatiotemporal process, (3) storing and retrieving real-time data from various sensors, and (4) representing the relationships among the sensor, observation, geographic object, object, spatiotemporal process, simulation, event, state, and change function. Therefore, a real-time GIS data model considerably supports the real-time and open GIS.

Real-time computation technologies, e.g., real-time computation frameworks, are developing due to the urgent need for real-time big data streaming. Real-time computation frameworks are emerging, such as the Apache Software Foundation, which provides top open-source, big data, real-time

processing frameworks, including Apache Storm, Apache Samza, Apache Spark, and Apache Flink [23]. Apache Spark is a large data processing framework built around speed, ease of use, and complex analysis. Compared with other big data and MapReduce technologies, such as Hadoop [24] and Storm [25], Spark has the following advantages: it provides a comprehensive and unified framework for managing the large data processing needs of data sets and data sources (batch data or real-time streaming data) with different properties (text data, chart data, etc.). Some studies show that Spark can increase the running speed of Hadoop cluster applications [26]. Since Spark has outstanding processing capabilities for massive real-time data, Spark has attracted greater attention among developers. Meanwhile, the current trend is to use real-time computation technologies in a GIS [27]. Therefore, Spark also supports real-time and open GIS.

The objective of this paper is to integrate the real-time GIS data model, the Spark framework, and some OGC specifications to develop a real-time and open GIS. Meanwhile, we apply the real-time and open GIS for constructing a smart river. The contributions of this paper are as follows: (1) designing and implementing a real-time and open GIS, called GeoSensor and (2) providing a case study for constructing the Yangtze River into a smart river with the GeoSensor.

The rest of this paper is organized as follows: methods are introduced in Section 2, useful cases of the system are described in Section 3, the results and discussions are offered in Section 4, and conclusions are summarized in Section 5 along with outlooks for future work.

2. Methods

2.1. System Architecture

A GIS is a layer-based system for processing logical data information, services, and applications. The characteristic of the real-time and open GIS framework is the full integration of real-time computing and open services. Therefore, a layer-based architecture of a real-time and open GIS, called the GeoSensor, was proposed, as shown in Figure 1.

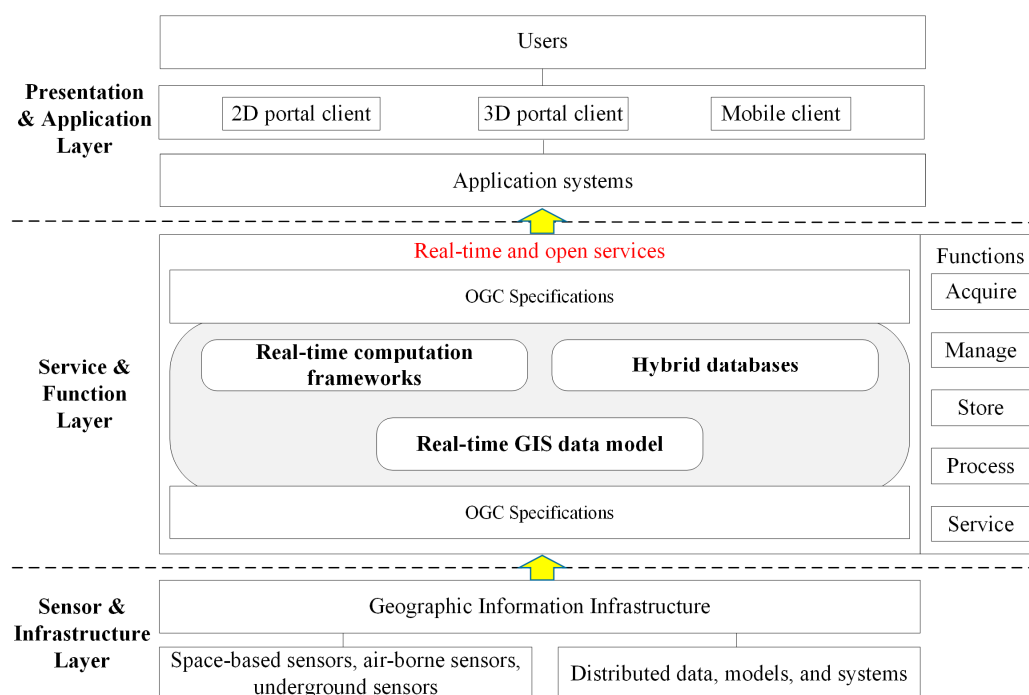


Figure 1. The architecture of the GeoSensor.

In the sensor and infrastructure layer, space-based sensors, air-borne sensors, and underground sensors are increasingly important infrastructures for capturing real-time information. The distributed

data, models, and systems are the resources required and could be integrated with real-time and open services.

The presentation and application layer displays the data, information, and knowledge that result from the application systems of the real-time and open information system with a 2D, a 3D, and a mobile client for users (e.g., governmental sectors, experts, and citizens).

The service and function layer is the core of the system. The real-time GIS data model manages heterogeneous resources and their relationships and helps design the storage structure of the database. The hybrid database stores the real-time data stream from sensors and data systems by a relational database and a Not only SQL (NoSQL) database. Real-time computation frameworks support the rapid processing of real-time data. With the help of a real-time GIS data model, professional models, hybrid databases, and real-time computation frameworks, the functions (acquire, manage, store, process, and service) of the system are developed and exposed to many OGC specifications, e.g., OGC services. The OGC services are WMS, WFS, WCS, SOS, WPS, etc.

Integrating the real-time GIS data model, hybrid databases, and real-time computation frameworks is the core of the service and function layer. In the next section, the integrated methods will be illustrated.

2.2. System Method

2.2.1. The Conceptual Mapping of the Real-Time GIS Data Model to Open GIS Models, Services, and the Real-Time Computation Framework

The proposed real-time GIS data model is a conceptual model. To manage and process real-time resources and processes in practice, the model requires mapping within the real-time GIS data model, the open GIS models and services, and the real-time computation framework. After careful study, the abstract mapping principles of the real-time GIS data model to open GIS services and the real-time computation framework are as follows (as in Figure 2):

- Sensor mapping to the SensorML and the SOS: The SensorML described Sensor information, and the SOS serviced the observation result of a sensor.
- Spatiotemporal process mapping to the WPS and Spark: A spatiotemporal process is encapsulated as a service by the WPS with the Spark computing framework.
- Observation mapping to the O&M model and the SOS: An observation is encoded by O&M, and the observation is provided by the SOS.
- Simulation mapping to the WPS and Spark: The simulation is the result of a spatiotemporal process. The simulation can be encoded by the O&M model and serviced by the SOS.
- GeoObject mapping to the Geography Markup Language (GML): A GeoObject is encoded by using the GML, and then it becomes the Object.
- Change function mapping to the WPS and Spark: A change function is a process, and it can be encapsulated as a service by the WPS with the Spark computing framework.

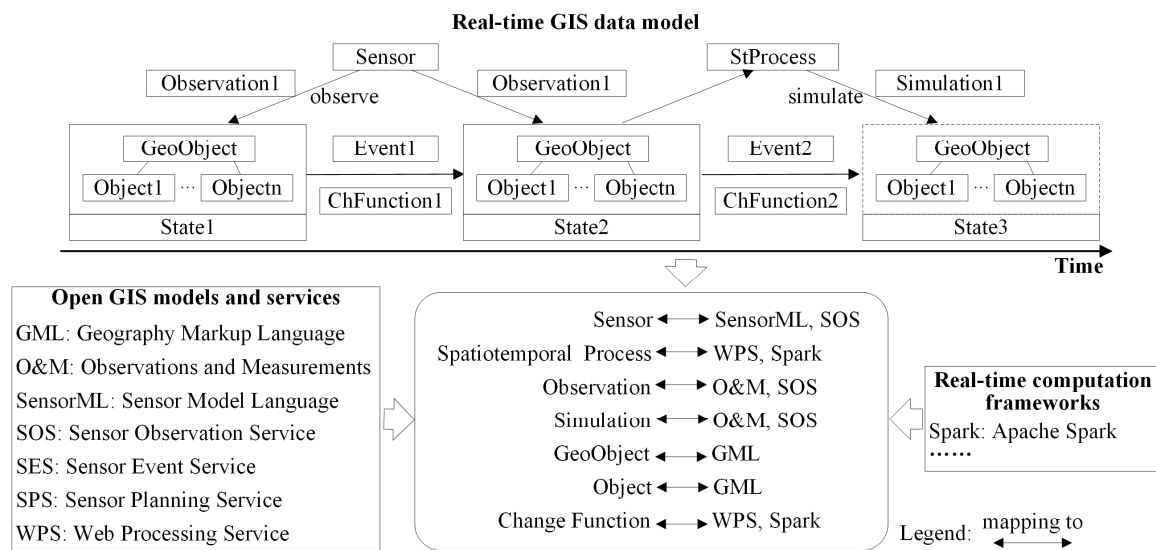


Figure 2. The mapping relationships among real-time GIS data models, open GIS models and services, and real-time computation frameworks.

2.2.2. The Integration of Open GIS Services, Real-Time Computation Framework, and Hybrid Databases

With the development of information technology, the sharing and interoperation of resources have always used the web service method. OGC services have certain applications, but many of them are mainly developed for single servers or service static resources [28,29]. The integration of open GIS services, real-time computation frameworks, and hybrid databases needs to be studied.

OGC open services expose their interfaces to users, while users do not know and do not need to know how they are implemented. Therefore, the key problem of implementing a real-time OGC service is choosing a real-time computing framework and a suitable database to support its basic implementation, as shown in Figure 3. Integrating the OGC services seamlessly with a real-time computation framework and hybrid databases requires the design of a system architecture, in which the “shell” is an OGC service interface and the “kernel” is a real-time computation framework and a hybrid database. Each OGC service may need to adapt a hybrid database. Chen describes an adaptation method [30]. The adapter includes configuration files and an abstract factory design pattern intended to “provide an interface for creating families of related or dependent objects without specifying their concrete classes” [30].

In Figure 3, the choice of the real-time computation framework depends on the developer. The system adopts Apache Spark to address high-frequency, massive-volume, stream-based sensor data since it is one of the best distributed real-time computation systems [23]. For the processing function, the key issue is how to integrate Spark with the WPS. The WPS defines standard interfaces for geospatial processing services. Following the architecture of the real-time computation framework-supported OGC service in Figure 3, WPS exposes standard interfaces, such as GetCapabilities, DescribeProcess, and Execute, to the client. Spark is responsible for logic processing and saves real-time data with the hybrid database organization method.

The core processes of integrating the WPS with Spark are shown in Figure 4. One of the mandatory inputs of the WPS is real-time data streaming from live sensors or the SOS. Another mandatory input of the WPS is a processing job. Real-time data streaming is divided into several DStream data fragments according to the specified size. DStream is the basic abstraction of Spark Streaming and consists of a series of resilient distributed datasets (RDDs). The logic of a processing job for WPS should be developed by executing RDD transformations in the Spark framework. In the Spark framework, the directed acyclic graph (DAG) scheduler splits a job’s logical graph into stages of tasks. The tasks were collected as the task set and then submitted to the TaskScheduler by DAGScheduler. DAGScheduler monitors the process, and if there is a failure in the scheduling phase, it will be

resubmitted. TaskScheduler receives the task set that was sent by DAGScheduler and distributes the task set as a task to an Executor on the cluster worker nodes for execution. If a specific task fails, the TaskScheduler is responsible for retrying, and the TaskScheduler may schedule other executors to perform the same task. Executor in the worker node receives tasks sent by the TaskScheduler that run in a multi-threaded manner, where one thread is responsible for one task.

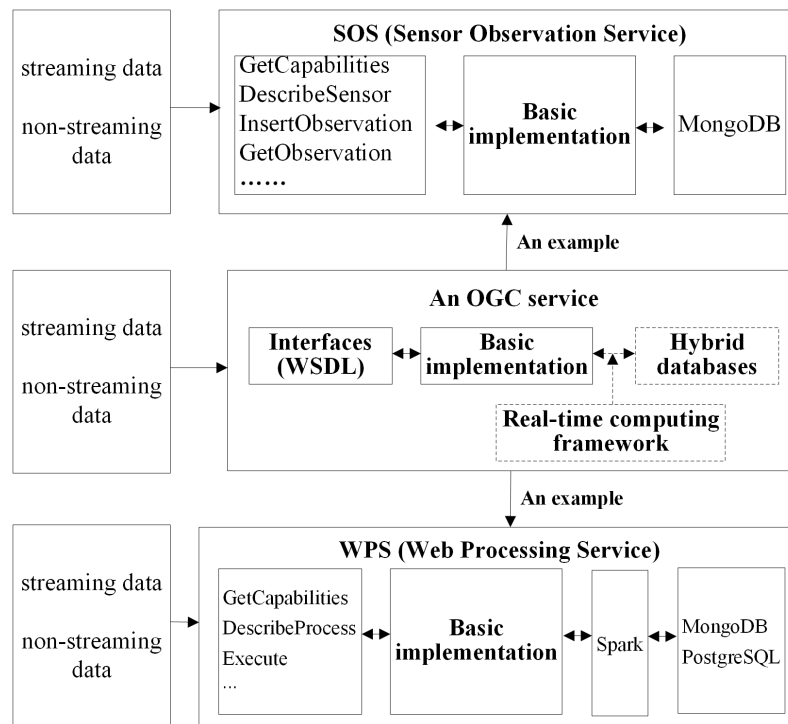


Figure 3. The architecture of a real-time computation framework and hybrid databases supported by the OGC services.

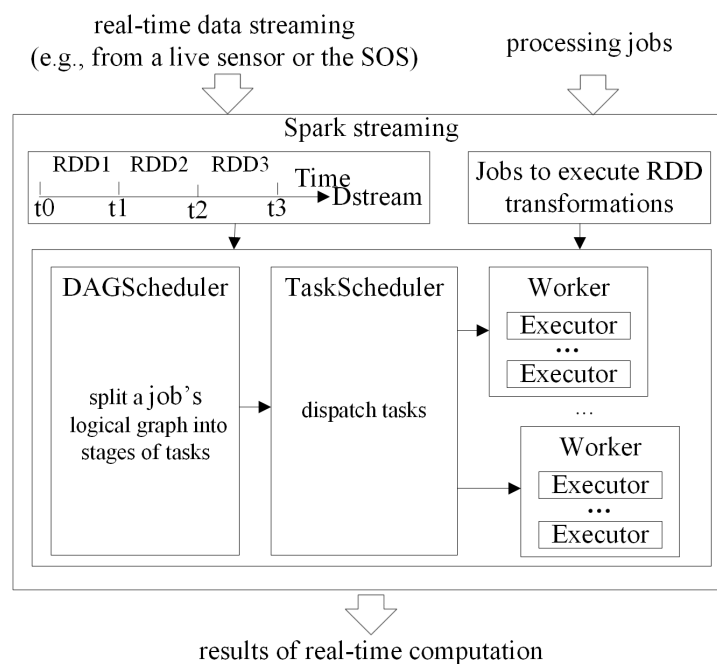


Figure 4. The core processes of integrating the WPS with Spark.

The performance of Spark was tested in an example. A Spark cluster with Moderate Resolution Imaging Spectroradiometer (MODIS) data (6634 files, a total of 1000 gigabytes (GB)) was designed to retrieve sea surface temperatures by using the split window algorithm with a common master server and five slave servers [31]. In stand-alone mode, a server can achieve approximately 0.04 GB per second. With the help of the Spark real-time processing framework, the clusters can achieve approximately 0.40 GB per second. The efficiency of image processing in a cluster mode is approximately 10 times that in a single machine. When the data volumes are small, the processing efficiency of the Spark is not clear. However, when the data volumes are large, the advantage of efficiency is clear.

The SOS provides standard interfaces to manage and retrieve metadata and observations from heterogeneous sensor systems. The SOS exposes the standard interfaces, such as GetCapabilities, DescribeSensor, InsertObservation, and GetObservation, to the client. MongoDB [32] is responsible for saving real-time data. The performance of the MongoDB SOS was tested using floating car sensor data. The software information and the computer server parameters were in Tables 1 and 2, respectively.

Table 1. Software.

Name	Version
CentOS (Linux system)	6.2 (64-bit)
Java JDK	1.7
Apache JMeter	2.13
MongoDB	3.0
PostgreSQL	9.2
Tomcat	7.0.61

Table 2. Server parameters.

Name	Parameters
CPU	Intel Xeon E5-2620 (2*6 core, 2.00 GHz)
Memory	32 GB (1333 MHz)
Hard drive	405 GB (SAS, 15000 rpm)
Network	1 Gbps

A floating car sensor records four observations at one time, with an average request size of 3.8 kilobytes (KB) and an average response size of 0.6 KB. For the SOS InsertObservation operation, the performance comparisons of the PostgreSQL SOS and the MongoDB SOS are shown in Table 3.

Table 3. The performance comparisons of the SOS InsertObservation operation.

PostgreSQL SOS		MongoDB SOS	
Data Size (Tens of Thousands)	Response Time (Milliseconds)	Data Size (Tens of Thousands)	Response Time (Milliseconds)
0	58	0	51
10	67	100	49
50	78	500	50
100	122	1000	51
300	280	3000	46
1000	411	10000	41
/	/	30000	42

The speed of the PostgreSQL [33] SOS decreases with increasing data volume. When the data volume reaches 10 million insertions, the insertion speed drops to approximately 146 times per second, which is approximately 10% of 1555 times per second at the beginning. However, the InsertObservation performance of the MongoDB SOS is basically consistent from 0 to 300 million insertions, and there is no clear decline in speed. The InsertObservation performance of the MongoDB SOS is basically stable between 50 and 70 ms, and the total insertion speed at this time is maintained at more than 4000 times/s.

For the GetObservation operation, the performance comparisons of the PostgreSQL SOS and the MongoDB SOS are shown in Table 4.

Table 4. The performance comparisons of the SOS GetObservation operation.

Concurrent Users	Response Time (Milliseconds)		
	PostgreSQL SOS	MongoDB SOS	MongoDB SOS/PostgreSQL SOS
1	37	13	35.14%
10	72	11	15.28%
100	1559	34	2.18%
200	2687	74	2.75%
1000	12564	437	3.48%

The MongoDB SOS stores 300 million observations, and the PostgreSQL SOS stores 10 million observations. When 1, 10, 100, 200, and 1000 users request the GetObservation operation concurrently, the response time of the MongoDB SOS is 35.14%, 15.28%, 2.18%, 2.75%, and 3.48%, respectively, but the basic data volume is 30 times that of the PostgreSQL SOS. The MongoDB SOS has a shorter response time than the PostgreSQL SOS.

Therefore, the GeoSensor can provide real-time open GIS services by integrating a real-time computation framework and hybrid databases.

2.3. System Implementation

The GeoSensor was implemented based on the ideas proposed in Sections 2.1 and 2.2. The GeoSensor is a service-oriented architecture and has four sub-systems known as the GeoSensorRegistry, GeoSensorManager, GeoSensorService, and GeoSensorViewer, as shown in Figure 5.

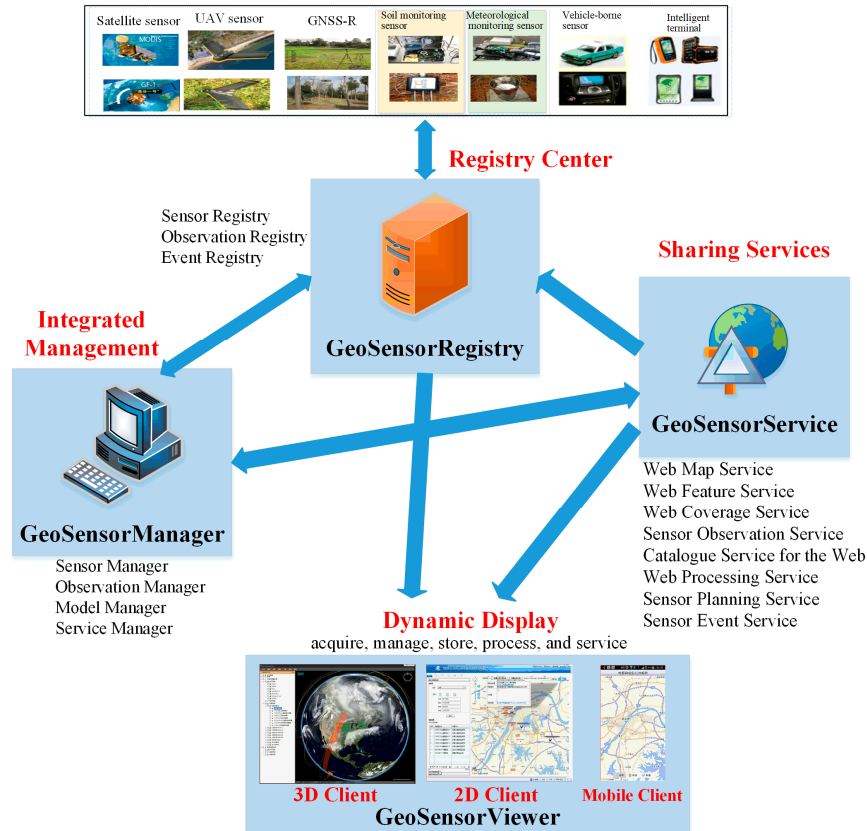


Figure 5. The components of the GeoSensor.

The GeoSensorRegistry is a registry center for sensor, observation, and event registry. The GeoSensorManager is responsible for the integrated management of sensors, observations, models, and services. The GeoSensorService is an OGC service server for WMS, WFS, WCS, SOS, WPS, and so on. The GeoSensorViewer is a 2D/3D/mobile client for dynamically displaying the requests and responses of common GIS functions, such as acquisition, manage, store, process, and service. The system was developed by the Java [34] programming language and the Java Spring Framework. The NoSQL database and the relational database are composed of MongoDB [32] and PostgreSQL [33], respectively. The real-time computation framework is used in Apache Spark. The GeoSensor is not only an integrated management system for sensor network resources but also a platform for geographic information services and a platform for analysis and simulation assisted by OGC service technologies and real-time computing technologies. The GeoSensor has the following features:

- Integrates with professional observation systems and integrates heterogeneous sensors from many industries, observation platforms, and protocols.
- Shares the whole process of observation and implements the real-time sharing of the observation environment, observation platform, observation data, and observation products in the web environment.
- Integrates domain applications with geography information systems through data, models, events, and services.
- Has the functions of assistant decision-making and simulation and implements the simulation of complex processes by combining professional assistant decision-making models.

3. Case Study

The Yangtze River has the largest watershed in China, and the total length of the river is more than 6300 km. The Yangtze River is the “Golden Waterway” that crosses the eastern, central, and western parts of China and poses three great challenges including flood control, hydropower production, and navigation. These challenges require a real-time and open information system that can fully manage the information on water, rain, sediment, navigation aids, and more in the whole river in a timely manner and the results are derived from the information. The temporal frequency of information collection and maintenance is daily or monthly, which cannot meet the daily or hourly requirements. Different missions are managed by different governmental sectors. The resources, e.g., the data, sensors, models, and systems, are heterogeneous and dispersed in different governmental sectors, usually with too little sharing and interoperation. Most existing systems are traditional information systems with many artificial interventions, and their results lack real-time and efficient processing capacity. Governments and experts think that there is an urgent need to adopt advanced open and real-time GIS technology to manage and analyze real-time dynamic resources to solve problems. They think that it is time to construct a smart river. A smart river is a vision similar to that of a smart city that adopts emerging information technologies to solve problems that could not or were not easily solved before. Therefore, several governmental sectors, universities, scientific research institutions, and companies are working together to build a smart river with the GeoSensor.

In keeping with the smart river concept, a comprehensive web-based sensor based on the GeoSensor was constructed and integrated with the real-time information on rainfall, beacon, channel, sediment, and water levels derived from space-based sensors, air-borne sensors, and underground sensors in the Yangtze River basin, as shown in Figure 6. There are 32 types of sensors, more than 5000 navigation aids, 300 hydrologic stations, and 100 river basin-monitoring satellites. The number of observation records is more than 10 billion. Based on the comprehensive sensor web and the GeoSensor, three application systems were developed to meet the challenges. The systems are a hydrology and sediment database and an information management analysis system for the cascade hydropower stations in the lower reaches of the Jinsha River, which is an information sharing system of reservoirs in the upper reaches of the Yangtze River, and the Yangtze River channel dynamic monitoring system (shown in Figure 6). In the systems, the SOS collects real-time data, the SPS controls sensors that

observe real-time data, the Sensor Event Service (SES) filters real-time observations, and the WPS encapsulates models from experts. All operations are performed in open ways.

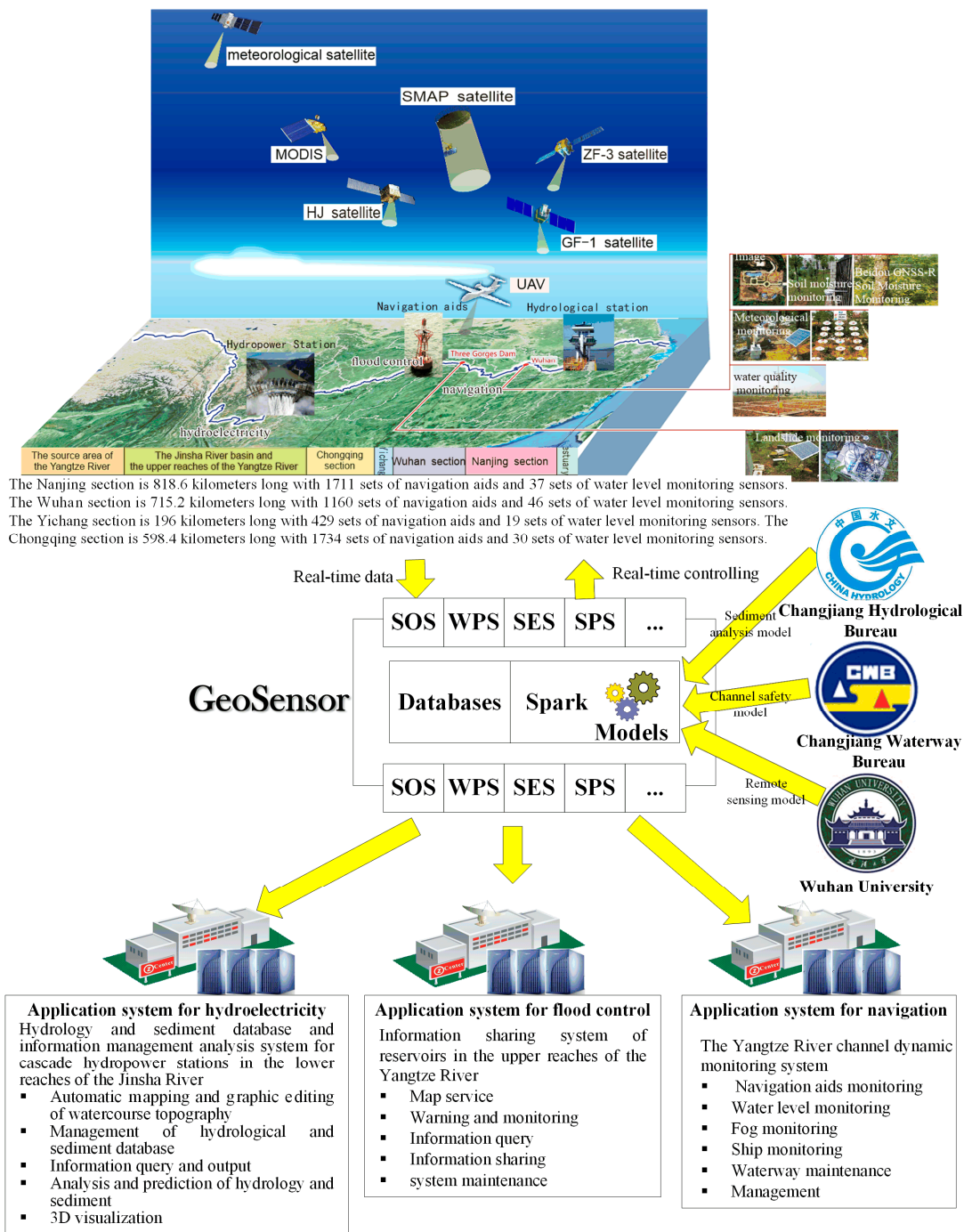


Figure 6. The comprehensive web-based sensor based on the GeoSensor in the Yangtze River basin.

4. Results and Discussions

4.1. Applications for Hydropower Production

The real-time and open system has been applied to manage and analyze the hydrology and sediment data and information of cascade hydropower stations in the lower reaches of the Jinsha River. The real-time and open system can perform a number of actions: accessing more than 300 hydrological and meteorological stations in the Jinsha River in real time, integrating dynamic

hydrological, sediment, meteorological, and remotely sensed observation data, encapsulating the hydrology and sediment analysis and calculation model, the hydrodynamic model calculation, and the river evolution prediction model from different experts, e.g., from the Changjiang Bureau of Hydrology and Wuhan University, as shown in Figure 6. It can also support the analysis of river evolution, sedimentation, and flood evolution by 3D visualization and simulation in a timely manner and in an open way, as shown in Figure 7. The real-time and open system plays an important role in monitoring and providing early warnings for the waterpower generation of the Wudongde, Baihetan, Xiluodu, and Xiangjiaba hydropower stations in the downstream Jinsha River. For example, the Xiluodu and the Xiangjiaba scouring and silting power generation realized real-time scouring, silting from scouring, and silting at regular times. In the flood seasons of 2013, 2014, and 2015, 0.8 billion, 1.6 billion, and 1.8 billion cubic meters of water were saved ahead of schedule, respectively. The economic benefit was remarkable.

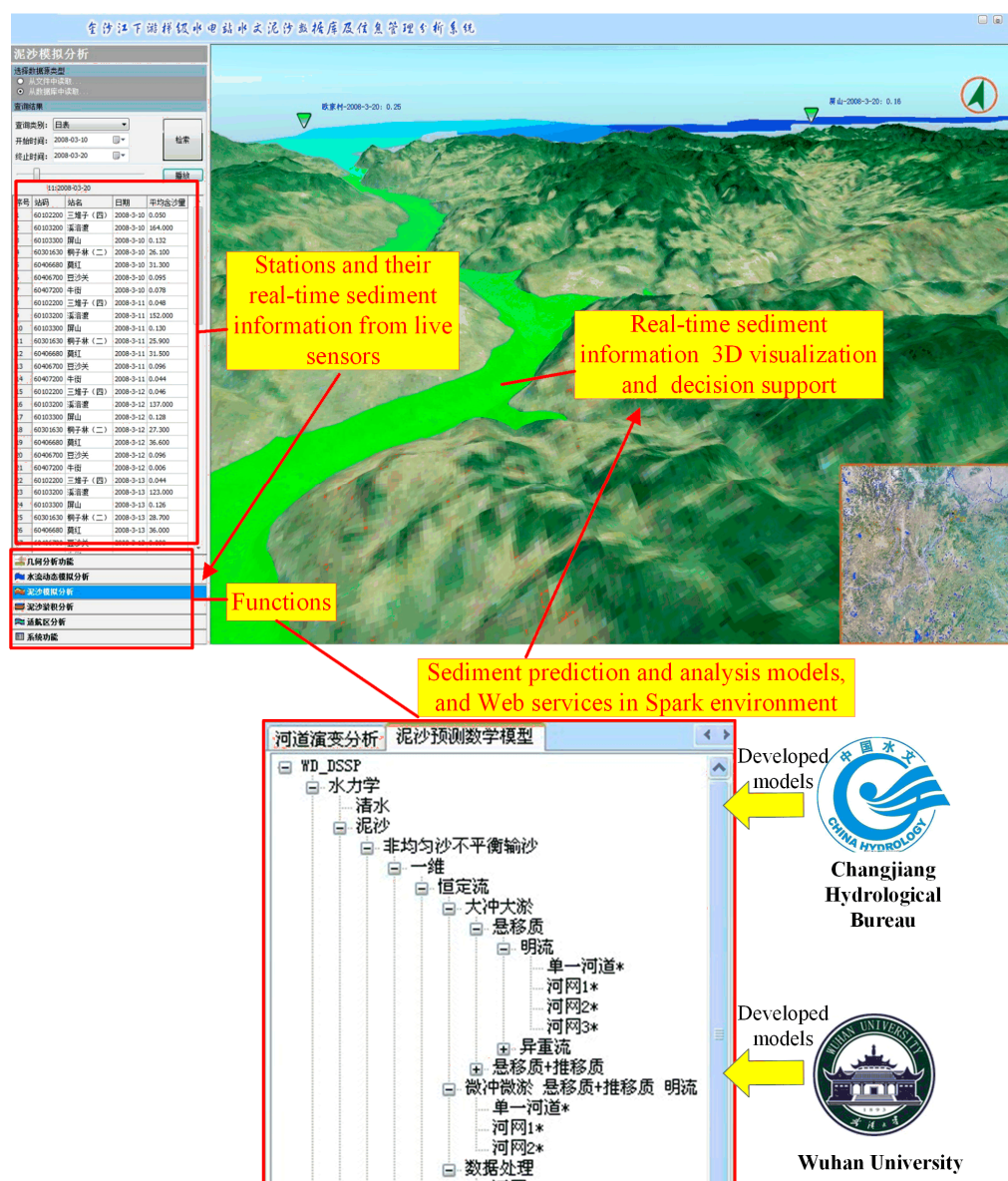


Figure 7. Visualization of the hydrology and sediment data and information with real-time data and information in an open way.

4.2. Application for Flood Control

The real-time and open system has been applied to share the information of the reservoirs in the upper reaches of the Yangtze River, which is shown in Figure 8. The system provides integrated, remotely sensed data from meteorological satellites and dynamic observation data. It can access the rainfall and flow rates of the sensor data in a timely manner in an open way from more than 8000 hydrological and meteorological stations in the 21 reservoir groups. It uses Spark models to enable the real-time calculation of the observation information, forecasting, and dispatching information, and rainfall information of the large controlled reservoirs in the upper reaches of the Yangtze River. All of the comprehensive real-time information can be sent to the Yangtze River Flood Control and Drought Relief Headquarters in less than 20 min and to the National Flood Control and Drought Relief Headquarters in less than 30 min. Real-time calculations and information sharing helps with making flood control decisions by the joint operation of the 21 reservoirs in the upper reaches of the river, which meets the demand for flood control.

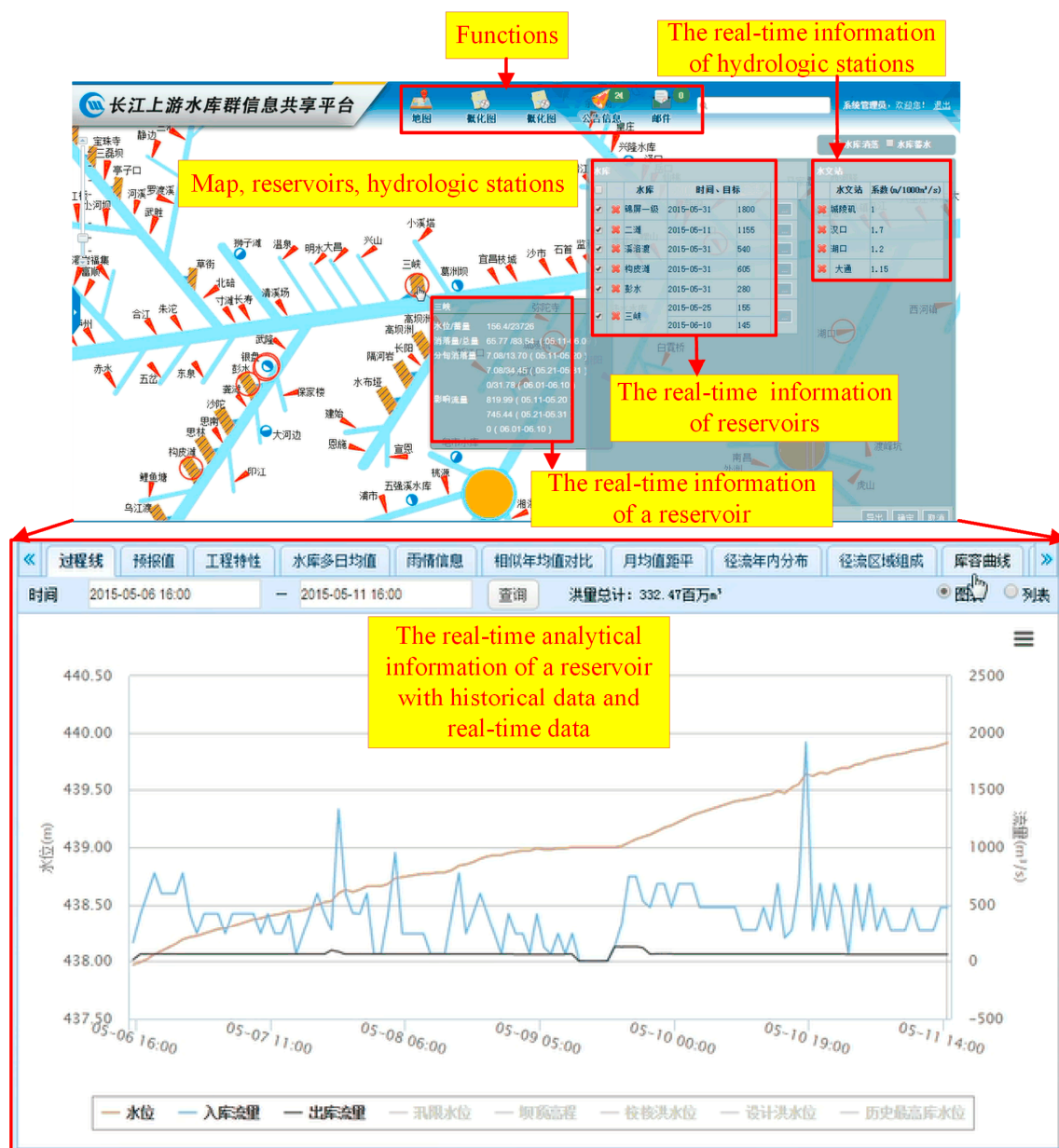
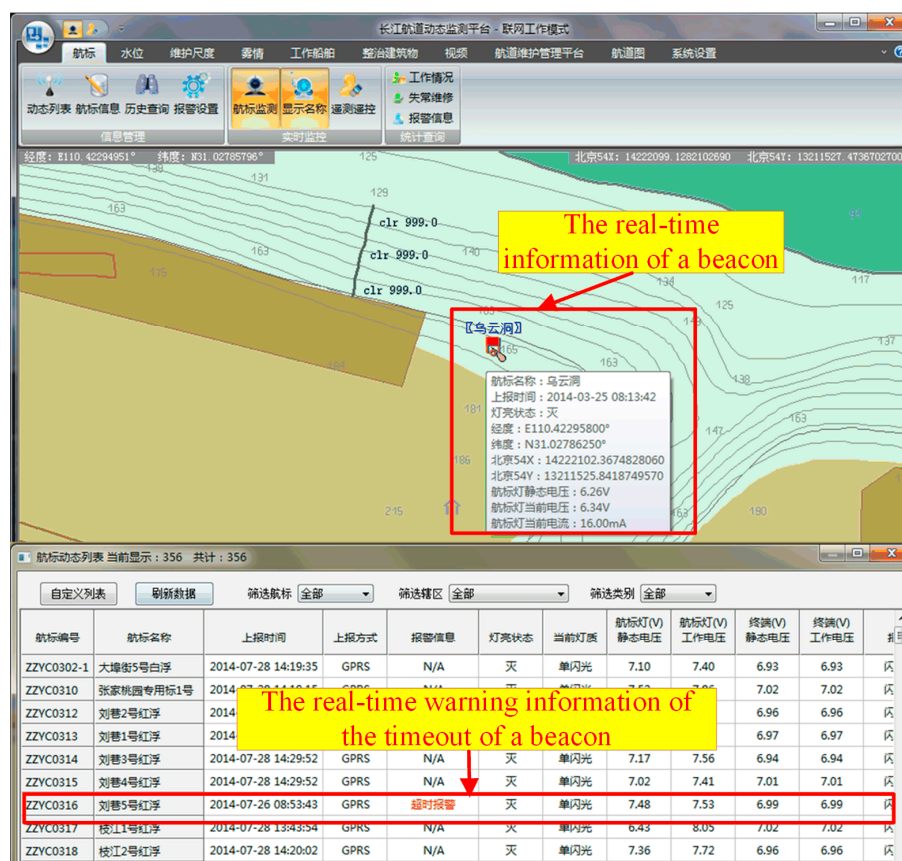


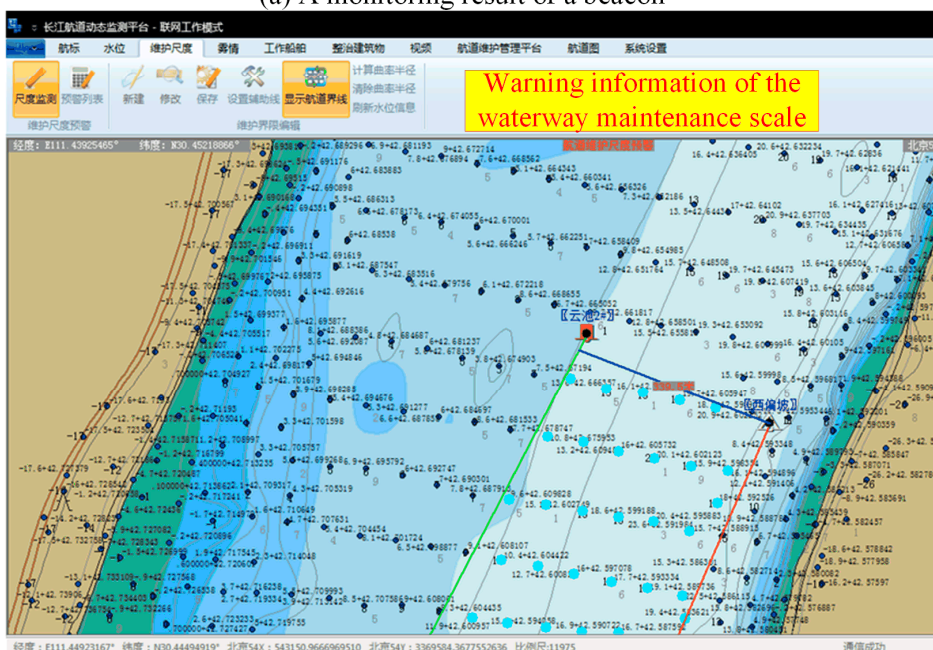
Figure 8. The GeoSensor shares real-time data and information of the reservoirs in an open way.

4.3. Application for Navigation

The real-time and open system has been applied to monitor and analyze the main channel of the Yangtze River, as shown in Figure 9.



(a) A monitoring result of a beacon



(b) A monitoring result of the waterway maintenance scale

Figure 9. The GeoSensor monitors and analyzes the main channel of the Yangtze River.

The GeoSensor integrates 5034 sets of telemetry monitoring systems and 132 sets of water level remote sensing report systems along 2328.2 km of the Yangtze River from Lanjiatuo to Liuhekou. The systems have the following characteristics: access to real-time information from water level buoys, telemetry monitors and alarm systems, water level prediction and automatic alarms for abnormal levels, channel maintenance scale monitoring and early warning systems, and navigation channel evolution analysis. The system provides certain advantages, such as transforming the beacon routing inspection into unattended monitoring, shortening the average mark recovery time from 8 hours to 2 hours, improving the water level sampling frequency from once per day to once per hour, and changing the Yangtze River waterway management mode from traditional to the “Internet Plus” mode. The system enhances the efficiency of the daily management of the Yangtze River waterway, the public service level, and emergency response capabilities, which meet the demands for navigation safety.

4.4. A Way to Have a Real-Time and Open GIS

The GeoSensor offers a way to have a real-time and open GIS. From the aspect of theory, the proposed methods can support the realizations of real-time GIS functions. A real-time GIS data model was proposed to organize real-time GIS features and resources. Compared with general data models, the real-time GIS data model considers sensor and spatio-temporal processes, where the sensor is the most important way to derive real-time data and the spatio-temporal process is the most important characteristic of a geographic phenomenon. With the explosion of geographic information, a real-time GIS captures and stores geographic information and faces great challenges since information becomes highly efficient and has a large volume and diverse data structure. Thus, any kind of database cannot meet the requirements of the real-time GIS, and hybrid databases should be used. The hybrid database method makes full use of the advantages of the relational database and the NoSQL database and integrates them to meet these challenges. The real-time computation framework synthetically supports the acquisition, the process, and the service functions of a real-time GIS. In the GeoSensor, the Apache Spark, which is known as a kind of real-time computation framework, is used to implement three functions with real-time characteristics. Meanwhile, all the data services and processing services are encapsulated by using OGC standards. The GeoSensor is an organic integration of real-time computing technology and OGC standard services.

From the aspect of experiments, the Yangtze River use case is shown, and the results exhibit the real-time and open features of the GeoSensor. In the comprehensive observation network of the Yangtze River basin, the GeoSensor captured and managed the real-time information of rainfall, beacon, channel, sediment, and water levels derived from space-based sensors, air-borne sensors, and underground sensors. The GeoSensor reconciles the contradiction among flood control, hydropower production, and navigation, according to comprehensive real-time information. As shown in Figures 7–9, the GeoSensor visualizes the hydrology and sediment data in a 3D scene in a timely manner, shares real-time information on the reservoirs, and monitors the main channel of the Yangtze River with real-time beacon and water information.

5. Conclusions and Outlooks

This paper describes the design and implementation of a real-time and open GIS system called the GeoSensor. The GeoSensor can manage and process real-time data in web applications with open services. The GeoSensor is used to address information sharing and interoperation in flood control, hydropower production, and navigation applications in the Yangtze River. The used cases illustrate an example of constructing a smart river. The results show that the system improved the management efficiency of the Yangtze River and that real-time and open systems would be promising for smart rivers.

The GeoSensor mainly integrates the real-time Apache Spark computation framework. Future work will focus on integrating other real-time computation frameworks and comparing their application scopes, advantages, and disadvantages.

Author Contributions: Conceptualization, Z.C. and N.C. Methodology, Z.C. and N.C. Software, Z.C. and N.C. Validation, Z.C. and N.C. Formal analysis, Z.C. and N.C. Investigation, Z.C. and N.C. Resources, Z.C. and N.C. Data curation, Z.C. and N.C. Writing—Original Draft preparation, Z.C. Writing—Review and Editing, N.C. Visualization, Z.C. and N.C. Supervision, N.C. Project administration, N.C. Funding acquisition, Z.C.

Funding: The National Key Research and Development Program of China funded this research, grant number 2017YFB0503704, the National Nature Science Foundation of China programme, grant number 41771422, the Nature Science Foundation of Hubei Province, grant number 2017CFB616, and the Fundamental Research Funds for the Central Universities, grant number 2042017kf0211.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study, in the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

References

1. Goodchild, M.F. Citizens as sensors: The world of volunteered geography. *GeoJournal* **2007**, *69*, 211–221. [CrossRef]
2. Zhang, X.; Chen, N.; Chen, Z.; Wu, L.; Li, X.; Zhang, L.; Di, L.; Gong, J.; Li, D. Geospatial sensor web: A cyber-physical infrastructure for geoscience research and application. *Earth Sci. Rev.* **2018**, *185*, 684–703. [CrossRef]
3. Xu, W.; Zhu, Q.; Zhang, Y.; Ding, Y.; Hu, M. Real-time GIS and its application in indoor fire disaster. In *Proceedings of the ISPRS 8th 3D Geoinfo Conference & Wg II/2 Workshop, Istanbul, Turkey, 27–29 November 2013*; Isikdag, U., Ed.; Copernicus Gesellschaft MBH: Gottingen, Germany, 2013.
4. Li, X. Real-Time GIS Data Model supporting dynamic data management and spatiotemporal process simulation. Ph.D. Thesis, Wuhan University, Wuhan, China, 30 June 2014.
5. Gong, J.; Geng, J.; Chen, Z. Real-time GIS data model and sensor web service platform for environmental data management. *Int. J. Health Geogr.* **2015**, *14*. [CrossRef] [PubMed]
6. Looking Forward: Five Thoughts on the Future of GIS. Available online: <http://www.esri.com/news/arcwatch/0211/future-of-gis.html> (accessed on 18 January 2019).
7. Koonar, A.; Ou, Z.; Scarlett, B. A real-time geo-spatial information system: An integration of GIS technologies for ice forecasting. In *Proceedings of the ISPRS Workshop on Spatial Analysis and Decision Making, Hong Kong, China, 3–5 December 2003*; Li, Z., Zhou, Q., Kainz, W., Eds.; A.A. Balkema Publishers: Rotterdam, The Netherlands, 2003.
8. Tanzi, T.; Servigne, S. A real-time GIS for accident prevention on toll motorways. In *3rd Joint European Conference and Exhibition on Geographical Information—From Research to Application through Cooperation, Vienna, Austria, 16–18 April 1997*; Hodgson, S., Rumor, M., Harts, J., Eds.; IOS Press: Amsterdam, The Netherlands, 1997.
9. Real-time Geographical Information Systems for ITS. Available online: <https://pdfs.semanticscholar.org/23f4/b895fe6ac870b73769754e7fa2daf907f249.pdf> (accessed on 18 January 2019).
10. Li, M.; Liu, H.; Yang, C. A real-time GIS platform for high sour gas leakage simulation, evaluation and visualization. In *Proceedings of the ISPRS International Workshop on Spatiotemporal Computing, Fairfax, VA, USA, 13–15 July 2015*; Yang, C., Clarke, K., Yuan, M., Yu, M., Li, M., Guan, W., Sun, M., Huang, B., Eds.; Copernicus Gesellschaft MBH: Gottingen, Germany, 2015.
11. A Web-Based Application for Real-Time GIS. Available online: <http://www.isprs.org/proceedings/xxxv/congress/yf/papers/934.pdf> (accessed on 18 January 2019).
12. Arai, C.; Matsuda, N.; Shikada, M. An application of remote sensing and real time GIS to digital map for local government. In *23rd International Geoscience and Remote Sensing Symposium (IGARSS 2003), Toulouse, France, 21–25 July 2003*; Caldararu, F., Patrascu, S., Caldararu, M., Paraschiv, A., Eds.; IEEE: New York, NY, USA, 2003.
13. Dabrowski, J.; Kulawiak, M.; Moszynski, K.; Bruniecki, L.; Kaminski, A.; Chybicki, A.; Stepnowski, A. Real-Time Web-based GIS for Analysis, Visualization, and Integration of Marine Environment Data. In *Proceedings of the Information Fusion and Geographic Information Systems, Berlin, German, 17–20 May 2009*; Popovich, V.V., Claramunt, C., Schrenk, M., Korolenko, K.V., Eds.; Springer: Berlin/Heidelberg, Germany, 2009.
14. Beaubouef, T.; Breckenridge, J. Real-world issues and applications for real-time geographic information systems (RT-GIS). *J. Navig.* **2000**, *53*, 124–131. [CrossRef]
15. Laurini, R.; Servigne, S.; Noel, G. Soft Real-Time GIS for Disaster Monitoring. In *Geo-Information for Disaster Management*; van Oosterom, P., Zlatanov, S., Fendel, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2005.

16. Yun, J.; Kim, D.; Hong, D.; Kim, M.; Han, K. A real-time mobile GIS based on the HBR-tree for location based services. *Comput. Ind. Eng.* **2006**, *51*, 58–71. [CrossRef]
17. OGC® Standards. Available online: <http://www.opengeospatial.org/standards> (accessed on 18 January 2019).
18. Bröring, A.; Echterhoff, J.; Jirka, S.; Simonis, I.; Everding, T.; Stasch, C.; Liang, S.; Lemmens, R. New generation sensor web enablement. *Sensors* **2011**, *11*, 2652–2699.
19. OGC Sensor Observation Service Interface Standard. Available online: https://portal.opengeospatial.org/files/?artifact_id=47599 (accessed on 18 January 2019).
20. Infrastructure for Spatial Information in Europe (INSPIRE) Program. Available online: <https://inspire.ec.europa.eu/> (accessed on 18 January 2019).
21. Christian, E. Planning for the Global Earth Observation System of Systems (GEOSS). *Space Policy* **2005**, *21*, 105–109. [CrossRef]
22. The 52° North Software. Available online: <https://52north.org/software/software-projects/> (accessed on 18 January 2019).
23. Top Big Data Processing Frameworks. Available online: <http://www.kdnuggets.com/2016/03/top-big-data-processing-frameworks.html> (accessed on 18 January 2019).
24. Hadoop Website. Available online: <http://hadoop.apache.org/> (accessed on 18 January 2019).
25. Storm Website. Available online: <http://storm.apache.org/> (accessed on 18 January 2019).
26. Zaharia, M.; Chowdhury, M.; Franklin, M.J.; Shenker, S.; Stoica, I. Spark: Cluster computing with working sets. *HotCloud* **2010**, *10*, 95.
27. Zhou, L.; Chen, N.; Chen, Z. Efficient Streaming mass spatial-temporal vehicle Data Access in urban sensor networks based on Apache Storm. *Sensors* **2017**, *17*, 815. [CrossRef] [PubMed]
28. Chen, N.; Chen, Z.; Hu, C.; Di, L. A capability matching and ontology reasoning method for high precision Open GIS Service discovery. *Int. J. Digit. Earth* **2011**, *4*, 449–470. [CrossRef]
29. Chen, Z.; Chen, N.; Di, L.; Gong, J. A Flexible Data and Sensor Planning Service for Virtual Sensors Based on Web Service. *IEEE Sens. J.* **2011**, *11*, 1429–1439. [CrossRef]
30. Chen, N.; Di, L.; Yu, G.; Min, M. A Flexible Geospatial Sensor Observation Service for Diverse Sensor Data Based on Web Service. *Isprs J. Photogramm. Remote Sens.* **2009**, *264*, 234–242. [CrossRef]
31. Liu, H.; Chen, N.; Chen, Z. Retrieving Method for MODIS Sea Surface Temperature with Apache Spark. *Comput. Syst. Appl.* **2018**, *27*, 112–117.
32. MongoDB Website. Available online: <https://www.mongodb.com/> (accessed on 18 January 2019).
33. PostgreSQL Website. Available online: <https://www.postgresql.org/> (accessed on 18 January 2019).
34. Java Website. Available online: <https://www.oracle.com/technetwork/java/index.html> (accessed on 18 January 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).