Design for Spatiotemporal Information Cloud Platform of Smart City Based on OSGi

Tianpu Zhao, Zhibo Yu, Jiale Han, Min Zhang and Jingyang Wang*
Hebei University of Science and Technology, Shijiazhuang, China

*Corresponding author e-mail: ever211@163.com

Abstract. In 2009, IBM put forward the development policy of building “smart city” before building “smart earth”. In response to this policy, the National Bureau of Surveying and Mapping issued “The Notification on the Development of Smart City Spatiotemporal Information Cloud Platform”, which formally took the construction spatiotemporal information cloud platform of smart city as a pilot project. In this paper, the OSGi (Open Service Gateway Initiative) technology is used as the framework. Developing component on demand is realized by the SOA architecture and OSGi technology, which improves the efficiency of developing software, shortens the cycle of developing software and saves labor costs. Resource access is achieved through RESTful services, and distributed coordination services are implemented through Zookeeper.

1. Introduction

In recent years, the number of urban residents has increased dramatically. According to the statistics of the United Nations, it is estimated that the world's urban population will reach 66% in 2050 [1]. With the arrival of the population, there are also some urban diseases, such as traffic jams, increasingly serious environmental pollution, resource shortage and so on. People always feel helpless in the face of these urban diseases, but with the advent of smart cities, the government and people in dilemma have new expectations for future urban life. The United States which first proposed the slogan of “Smart City” released its first national-level smart city development strategy in 2015[2]. Chennai city managers in India have made a lot of efforts in policy to successfully implement smart city by taking advantage of various IT opportunities and economic & environmental challenges[3]. Through the Padova smart city system, Italy has collected a large amount of data and analyzed the relationship between data and data, which provides a basis for real implementation of smart city in the future[4]. Compared with the efforts made by foreign countries for smart city, China has also put forward the countermeasure for building the cloud platform of spatiotemporal information of smart city.

The spatiotemporal information cloud platform of smart city constructed in this paper has two characteristics: (1) Provide a universal platform that can be applied to multiple departments to facilitate communication.(2) Information sharing is flexible. In the previous digital city construction, massive data has been collected, while in the smart city, how to really apply the data, how to use it well and how to use it widely have become the theme. Since the proposal of “smart city” to the construction of “smart city” is a relatively short period of time, many things are still in a relatively “virtual” stage, so in the process of building spatiotemporal information cloud platform of smart city, we should combine it more with reality, put people first, highlight the dominant position of people,
create value for smart city through citizen participation, and take on the role of the smart city skeleton[5].

2. System Module Design

The spatiotemporal information cloud platform of smart city is composed of three parts: operation & maintenance management, cloud resource and homepage. Operation & maintenance management is mainly provided for the administrator of the platform, which is used to manage the basic information of the platform. Cloud resource is mainly provided for some professionals to use, for publishing a variety of services for homepage display. The homepage is mainly a platform for display, and the content displayed is different depending on the permissions of the visitors and platform users. The system module design diagram is shown in Fig. 1.

![System Module Design Diagram](image)

**Figure 1.** The system module design diagram

3. System Architecture

The construction plan of this platform is to build a one-stop service platform under the support of the security assurance system and standard specification system and infrastructure services to realize not only the integration of urban data software, management, services & applications but also the sharing of data & information. The one-stop service platform provides a basic support platform for building smart city with extensive interconnection, application and information services. This general-purpose platform can also be described by five keywords: “on demand service”, “full awareness”, “interconnection”, “integration sharing” and “business collaboration”. The platform designed in this paper is based on the ArcGIS cloud platform. The architecture of the system is shown in Fig. 2.
As shown in Fig. 2, the architecture of the platform can be divided into: infrastructure layer, data layer, service layer and application layer.

(1) Infrastructure layer: The infrastructure layer is a collection of software and hardware used in the platform, including firewall, computer room infrastructure, PostgreSQL database and other resources.

(2) Data layer: The data layer corresponds to the data of the platform. There are a lot of data stored on the platform, which can be mainly divided into two categories: system support data and application data. The system support data can also be called basic data, mainly corresponding to user management data, group management data, audit management data and other operation & maintenance management data. The application data mainly corresponds to the data that need to be registered and published into service in cloud resource management. It is also classified into toponym & address data, basic vector data, basic image data, planning data and thematic data of various departments.

(3) Service layer: The service layer includes published services and implemented functional modules based on data in the data layer. Function modules include user management, group management, etc. Published services include toponym & address services, thematic services, etc. Users can publish services through ArcGIS Server according to their own needs and register services through service registration module. After being audited and approved by the administrator of the platform, the service can be successfully released to the platform. The user who registers the service as the administrator of the service has the privilege control of this service, and can freely choose whether to share this service or to be privately owned. The administrator of this service can also set other users to become administrators of this service. If the service is updated or modified in the future, as long as the administrator of the service can do the corresponding operation.

(4) Application layer: The application layer is the basic components and business components, which is comprised of the data in the data layer and the services in the service layer. By calling these components to implement the various functions of the platform, it is convenient for users in different departments.

4. Key Technologies

4.1. Integration of SOA Architecture and OSGi Technology

OSGi is a popular modular framework in Java nowadays, and SOA (Service-Oriented Architecture) is a component model. The SOA architecture under the OSGi framework can take advantage of the characteristics of high cohesion and low coupling to decouple components from each other, and then recombine the required components according to the requirements of project functions to form new applications, so as to achieve project customization [6]. Components are divided into basic components
and business components. Business components depend on basic components and the basic components serve the business components.

In the platform, user management, application management, information management, information feedback, audit management, directory management, background management, secondary development of operation & maintenance management, group management, audit management and resource management of cloud resource are defined and implemented by interfaces. Since the entire platform is a maven project, the maven-bundle-plugin plug-in is required to build the bundle. The platform uses ServiceMix as a container. Because ServiceMix has integrated cxf, each interface can generate different jar packages directly using "maven install" command. The jar package here is the component. At the same time, the jar package generated by each module is deployed to the deploy file in ServiceMix. Each component exists independently, so when one of the business components needs to be modified, such as user management, only the modified interface of user management needs to be repackaged and deployed to ServiceMix, other components will not be affected. And when one of the components goes wrong, the efficiency and stability of OSGi can ensure that it does not affect the overall situation and does not cause global errors because of local errors.

4.2. RESTful Service

REST is a term coined by Dr. Roy Thomas Fielding in his paper in 2000. Its core idea is to abstract all information in Web applications into different resources. REST itself can be seen as a constraint or a rule, not a framework. All services that conform to the rule or constraint of REST can be called RESTful services[7]. In RESTful services, users can request data that is resources from the background through a specified URL (the interface address of the service request).

As mentioned above, user management, application management, information management and other services in the platform are defined and implemented by interfaces, and then packaged and deployed. This process is also the process of publishing RESTful services. After the successful publication of services, they are accessed through the prescribed URL address. Here's a point to note. Previously, these interfaces were packaged into components and put into ServiceMix. So when accessing these services, you need to start ServiceMix. It's better to enter a “list” to see if each component started successfully. In the platform, the address has three parts: (1) Public part: http://localhost:8080/ CreatMap/rest/service. (2) Each interface will set the service attribute in the bundle-context.xml file. Including name, mark and interceptors, which represent the service name, service comment and interceptor encountered. The value of name is the URL address of the second part. (3) There are many <action>s in the methodParameter.xml file corresponding to the interface.
Each <action> corresponds to an implementation class. One attribute in <action> is name, and the value of name is the URL of the third part. (1) +(2)+(3) is the complete access address of a RESTful service. The specific process of accessing platform data through a URL is shown in Fig. 3.

The main implementation process of accessing platform data through a URL is as follows:

1) The RESTful service sends a request to the platform via the URL address.
2) Interceptor. In introducing the second component of the URL, it is mentioned that in the bundle-context.xml file, there is an attribute called interceptors, whose value is what the interceptors encounter. In this platform, the interceptors which are set up is token interceptor. So the parameters of the request should include token and the correct token value.
3) Call the concrete implementation class through the service layer.
4) The cache layer. If the same service has been accessed before, the result will be retained in the cache layer within a certain period of time. At this time, it is better to directly retrieve the result of the cache layer. Otherwise go to the next step.
5) PostgreSQL database, this layer is to call the contents of the platform database, the results of the query in the database can be returned.

4.3. Zookeeper Distributed Coordination Service

Zookeeper is a distributed cluster management framework, which provides a directory tree-based data storage based on a root node-like file system. However, this storage method not only stores data, but also maintains and monitors changes in these data[8]. Once the change of the service is monitored, the user of the service is notified in time and the corresponding operation is performed. There are three roles within the Zookeeper cluster: leader, follow and observer. Leader is used to update the status of the system, dispatch transactions and process transaction requests; follow is used to read & write the client and have the right to vote for leader; observer does not participate in the election of the leader, only synchronizes the state of the leader and improves the speed of the request. Zookeeper cluster maintains the consistency of data and synchronizes the state of servers through the ZAB protocol. The ZAB protocol has two modes of existence, namely recovery mode and broadcast mode. When the leader is unresponsive or down, the ZAB protocol will be in recovery mode. The election of leader will be re-conducted within zookeeper cluster. When most followers and observers synchronize the leader status, the recovery mode will be terminated and the broadcast mode will be entered. Broadcasting mode is the normal mode of transaction processing.

The application of Zookeeper distributed coordination service in the platform is shown in Fig. 4.

In Fig. 4, the client can be regarded as a consumer of the service or a service gateway. ServiceMix is regarded as the provider of the service. Client can call services from ServiceMix according to the invocation rules of RESTful services, while Zookeeper which is the core of the whole distributed service registry, plays the role of decoupling and discovering services. In addition, the Zookeeper cluster internally synchronizes the status of follow and observer with the leader through the ZAB protocol.

When the service starts, ServiceMix registers the service name, service version information and server address in the form of nodes to the service configuration center. The client is assigned to a service node through the service configuration center. This service node is the sub-node under the root node. In order to ensure the high availability of services, multiple services with the same functions can be allocated under each service node. Therefore, the correct services available under the service node will be invoked through the invocation strategy of error retry.
5. Conclusion
This platform integrates massive data and information on the achievement of digital city, solves the problem of “information isolated island”, and combines with the achievements of some existing smart city. It fully considers the domain characteristics of geospatial science and the application characteristics of smart city, integration of the existing cloud computing technologies to build spatiotemporal information cloud platform of smart city from four levels: infrastructure layer, data layer, service layer and application layer. The platform integrates GIS technology with other industry applications, provides various intelligent services based on geographic information for people's lives and national economic production, realizes the use of data and provides visible conveniences for people's medical treatment, travel and other aspects. It lays the foundation for the application of large data, and realizes the economical utilization of information resources.

6. Acknowledgments
The authors would like to thank the editor and referees for their careful review and valuable critical comments. This work was supported by the fund of Hebei University of Science and Technology No. 2014PT21.

References
[1] Hannele Ahvenniemi, Aapo Huovila, Isabel Pinto-Seppä, Miimu Airaksinen, What are the differences between sustainable and smart cities?, Cities, 2017, pp. 234-245.
[2] Su J. J. and Meng X. L., Development status of smart city at home and abroad, Smart City, 2016, 2 (12), pp. 7-9.
[3] Rashmi Krishnamurthy, Rajeev Mishra, Kevin C. Desouza, City profile: Pune, India, Cities, 2016, pp. 98-109.
[4] Cenedese A., Zanella A., Vangelista L., Zorzi, M., Padova Smart City: An urban Internet of Things experimentation, A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on.
[5] Sergio Trilles, Andrea Calia, Óscar Belmonte, Joaquín Torres-Sospedra, Raúl Montoliu, Joaquín Huerta, Deployment of an open sensorized platform in a smart city context, Future Generation Computer Systems, 2016, pp. 221-233.
[6] Zhong L., Design and implementation of intelligent campus management system based on OSGi technology, Jiangxi University of Finance and Economics, 2018.
[7] Gu Q. B., Design and implementation of land and resources archives management system based on SOA, Ningbo University, 2017.
[8] Gu Q. B. and Pan S., Design and Implementation of Distributed Query Framework for Land and Resources Archives Based on Zookeeper, Mobile Communications, 2017, 41(05), pp. 84-90.