Scientific
Research

Microkernel and Middle-Ware Based GIS Platform Design

Tingyan Xing[1], Shang Zhang[1,2], Fan Zhang[3]

[1]Faculty of Information Engineering, China University of Geosciences, Beijing, China
[2]College of Computer and Information Technology, China Three Gorges, Yichang, China
[3]Zondy Cyber Group Co., Ltd., Optical Valley Software Park, Wuhan, China
Email: xtygis@163.com, wetoo@163.com

Abstract

By contrasting the traditional way in which GIS was completed that comes with less flexible, low efficiency, and lack redundancy which cause high entry cost, the fast development of microkernel plug-in technology provides the lightweight, efficient and scalable solution to Geographical Information System (GIS). This paper is to reveal the potential of microkernel plug-in geospatial information processing technology in GIS design and practice, with the acceptance and usage of function model called Resource loading manager(RLM) for GIS applications, which provides a possible solution to overcome the GIS's high cost issue. After the short review of microkernel plug-in technology, the possibility of GIS design with microkernel is analyzed. This paper also introduced the composition of the whole system and the design of GIS service platform based on middle-ware in detail.

Keywords

1. Introduction

The concept of Microkernel (micro kernel) was first put forward by Richard Rashid. It was originally made for building microkernel operating system based on message passing mechanism, not for software architecture services [1]. In recent years, as the development of software systems, especially with the "component" and "hierarchical" concept and micro-kernel technology gradually was introduced into software design framework for the "decoupling possible relationships between components" [2]. More and more modern software systems adopt the idea of microkernel by using plug-ins to realize software architecture. But in the field of GIS, microkernel-based

technology standardized plug-in system has not yet been established.

At same time, several drawbacks hindered communities who choose the commercial software for implementing GIS projects. Organizations lacking of enough financial resource cannot afford the cost of purchasing software. Furthermore, commercial packages typically are large and tightly integrated packages, organizations should pay for the whole package even though limited functional components are necessary. With the fast changing pace of technology, lack of standards in the GIS industry inevitably leads to plenty of false and wasted effort in GIS development. The difficulty, high cost and technical complexity of developing GIS have always been underestimated, which lead to the downright disappointment of pre-bubble promises on GIS deployment [3].

In this paper, we have established a geospatial information support platform based on microkernel architecture technology, including remote sensing data processing, open geographic information processing and mutual integration process, which provide a lightweight runtime environment. In this case, not only a lot of software purchase costs can be saved, but also a variety of personalized geospatial information processing demand can be met. This article presents a lightweight, efficient and scalable microkernel architecture geospatial information technology and method [3].

2. Brief Review of Microkernel and RLM

2.1. Microkernel

The concept of Microkernel (micro kernel) was originally made for building microkernel operating system based on message passing mechanism, not for software architecture services [4]. In recent years, as the development of software systems in style of "component" and "hierarchical", micro-kernel technology has gradually been introduced into software design framework for the "decoupling possible relationships between components". But in the field of GIS, microkernel-based system has not yet been established.

Actually, the microkernel based system could provide several benefits that traditional GIS platform don't have:
a) There is no need to develop new application systems to meet user's changing needs;
b) Easy to modify the existing similar software systems.

In traditional way the efficiency of scheme a), b) is impossible. Even small or new functional change resulted in system, the engineers often need to develop new application or modify the code. This will resulted in high development cost, and poor scalability.

For the government, enterprises, individuals and other different types of users which need a lightweight, efficient and scalable geospatial information procession support environment, this paper presented such a platform based on microkernel architecture technology, including remote sensing data process and open geographic information process and mutual integration process, which made a lightweight runtime environment. In this case, not only a lot of software purchase costs could be saved, but also a variety of personalized geospatial information processing demand can be met [4].

2.2. Brief State of the Art of RLM

In order to ensure the efficient resource management, a geospatial information application system based on micro kernel technology is designed in this paper.

The key of micro kernel technology in this paper is Geospatial information resource loading manager (RLM) which used to load the function module of program could realize the business logic of the application program. The structure of the RLM is shown in **Figure 1** [5].

From **Figure 1**, FSRL (functional service requirements list) is used to count the resources requirement by the client software. This information at least includes: client software security level, the features of required resources, and the method of administration.

According to FSRL, FR (function resolver) puts the functional requirements of client software in line with the rules of the system function modules to subdivide, and create functional modules list according to the requirements.

FMRL (functional modules requirements list) is used to describe the function modules refined from the functional service requirements of client software.

LC (list counters) is used to arrange function modules according to FMRL.

RCF (resource configuration file) is used to configure all functional modules.

AM (access manager) is used to check the permissions of LC calls for functional modules.

FC (function combiner) is used to package the function modules called by LC for plug-in, using for client software.

The Timer is used in one or more functional blocks occupied by other programs, and cannot be immediately called. The timer periodically detects the required permissions whether the function modules were released, and dynamically loaded them into the function plug-in when they were released.

3. Implementation of Microkernel Architecture

The basic idea of the method described above is: The process logic of programming system is implemented through a plug of expression of microkernel architecture [5]. FR is used to translate the functional requirements of client software into functional modules requirements. LC re-organizes and dispatches modules, and provides plug-ins for client software through FC. Furthermore, in order to satisfy the time limits and accuracy of function calls, we use RCF and AM to manage the resources. Meanwhile, the timer periodically detects the function module which is unable to be used temporarily. When the function mode is available, the FC immediately loads it into the plug-in functionality for client software to use [5]. The process of the RLM is shown in **Figure 2**.
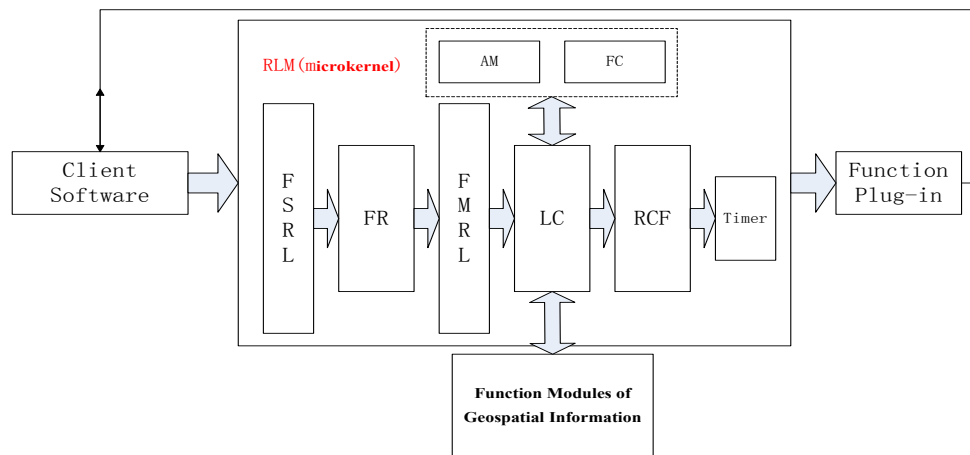

Figure 1. The structure of the RLM.

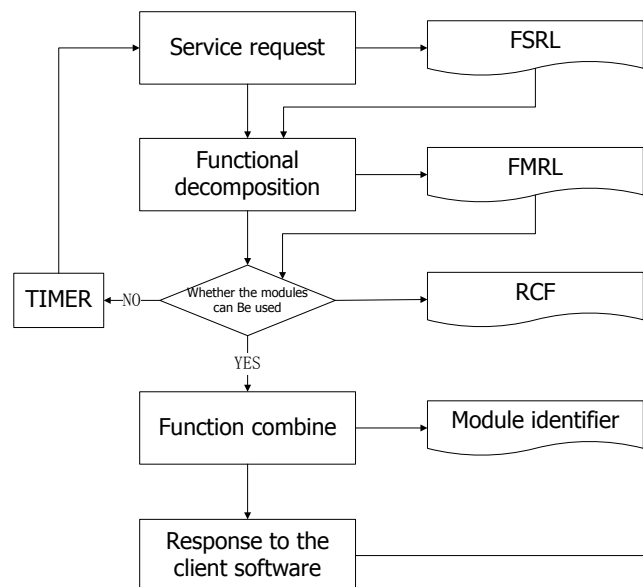**Figure 2.** Workfolw of the RLM.

The workflow is as follows:

a) Client software confirms the service requests to create FSRL, and submit to the RLM;

b) RLM confirms the FSRL submitted by client software. Then FR translates the required service into small granularity of function module, and generates the LC;

c) LC views the RCF according to FMRL to decide whether the required function modules could be called immediately. If possible, LC will call the corresponding function modules and update the status of function modules in the RCF;

d) FC package function service modules as function of plug-ins. This plug-ins is feed to client software as respond. Meanwhile, FC creates function module identification, and returns it to the client software for handle.

e) Client software calls plug-ins by module identification and after that returns it to the RLM. Then RLM updates RCF.

According to the above scheme, step b) also includes the following steps: After RLM receives FSRL submitted by client software; RLM translates the required service into small granularity of function module by FR. Meanwhile, RLM divides the requests of each functional module into read-only requests and marks them in generated function modules [5].

As the same, step c) also includes the following steps: After LC received FMRL, AM is read at first, in corroding to decide whether the client software has the permission to use the required level of the selected function modules. Then the results are identified in FMRL. For insufficient permissions to demand the use of functional modules, RLM will suspend the next steps. Finally, RLM checks RCF, confirms whether the required function modules could be used at moment. If the module cannot be called immediately, then the request will return temporarily, and the status will be identified in RCF [5].

Step d) should also include the following steps: The plug-ins provided to client software adopt the loose packaging mode. For exclusive functional resources such as write operation, when the user finish the write operation in the form of plug-in functional module, the instant loading manager will return to RLM, and updates RCF; Due to the function modules that are occupied by the other programs without using plug-in function modules, the demand of the client software will be set with a time interval repeat in step c), until the function modules were dynamic loaded as the plug-in function.

Step e) should also include the following steps: Client software will receive FMRL about status of functional plug-ins package which suggests that the reason why some function modules could not be available at moment.

4. A Protype GIS Service Platform Design

Data is the foundation of all applications and services. The uniqueness and accuracy are the important characteristics of a geographic entity, therefore, in this LBS platform differences and ambiguity caused by the multiple data sources must be overcome. This platform (**Figure 3**) in the GIS based service is to deal with all kinds of basic data, including the navigation electronic maps of all kinds of manufacturers, all kinds of industry data and remote sensing image data. GIS based service mainly has two parts, one is an online service support system, and another is a middle-ware system. The design, implementation, operation of these two parts require rigorous data standard and constrained service specification [6].

4.1. Online Support Service System

Online service support system is based on the data management module which conducts the most basic data management, multi-source data fusion, data processing and distribution, the diversity of data services, third-party data service support and so on. The design is to build a quick compatibility and integration of a variety of data sources of standard geographic data management system.

By establishing such a unified geographic data management system, flexible scheduling and data dispatch module are needed. Due to the diversity of the LBS service and the growing demand for the industry, different service requirements for data format are also diverse. Through data scheduling and dispatch module, and according to the data content choice, the user can specify its own data format needs, and carry on the content data release, data integration and updating. Therefore, all data are standard unified and constant. The diversification is also the important guarantee of accurate GIS service. According to industry application and characteristics of the public service, while through the data access middle-ware and services middle-ware interface, the LBS platform can be customized [6].
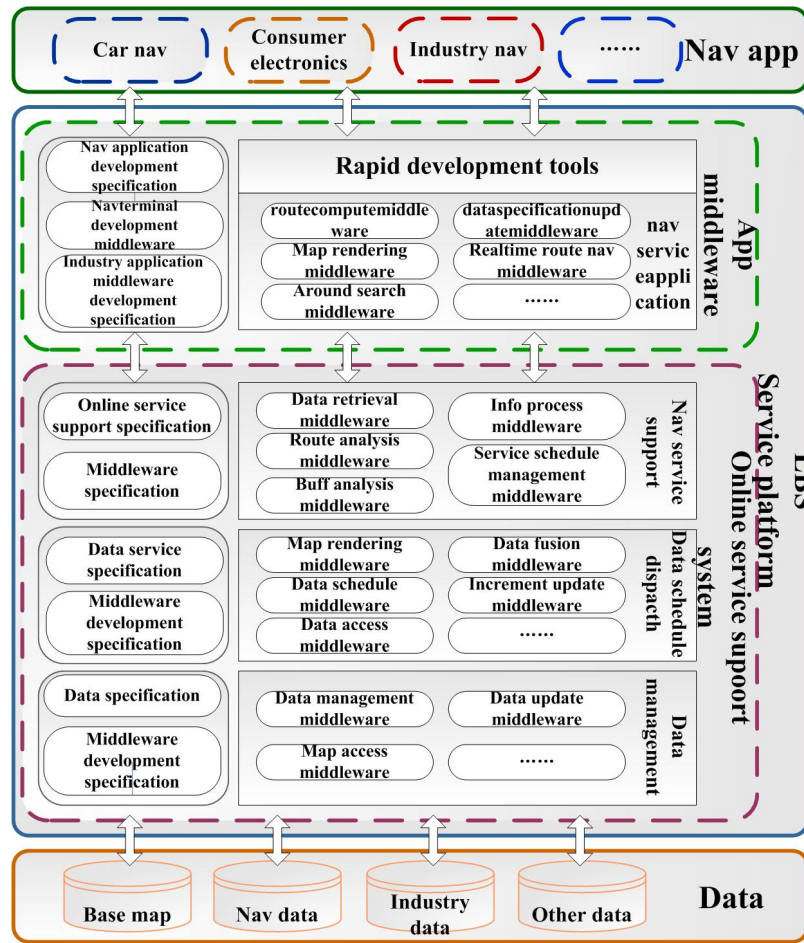
Figure 3. The architecture of GIS service platform.

The Nav service support module is a platform to support various types of online services, it mainly includes various navigation electronic map service functions, such as spatial analysis, buffer analysis, data scheduling, information processing, online service application, user management, and other functions. Those functions are designed to provide online function module for the user of third party, and also could for the middle-ware design and development, which can be composed for all kinds of navigation application.

4.2. The Middle-Ware Platform

Middle-ware system is the core of GIS service, which throughout the underlying data management to online services support, and the application development, middle-ware technology process.

The service platform composed of the various types of middle-ware. In order to achieve structure flexible and functional independence, function modules are based on middle-ware implementation, which are basement of the infrastructure of the services platform.

In the field of electronic map navigation application, middle-ware is the main form and content of navigation terminal development, their complementation are completely followed by the specification shown in the **Figure 1**. In development, the use of middle-ware can have better classification of all kinds of functions and extensions, which can assist the user to build and deploy the GIS services [7].

Otherwise the rapid development tools are, moreover, providing an important feature of GIS platform, which increase the difficulty of software development, shorten the development cycle, and solve the problem of multiplication differences. The tool can support multiple mobile devices in terms of building type development; the development of the core functionality can be implemented directly by the middle-ware in the form of drag and drop. Rapid development tools make full use of the navigation terminal development, middle-ware and industry

application which could improve the efficiency of software development, enhance its extensibility.

5. Conclusions

GIS public service will not only change the ordinary way of life, but also bring enormous business opportunities. At present, the high cost of service influences the LBS public service. The development and application of microkernel or middle-ware based product will provide GIS service with a high efficiency and low cost.

Meanwhile microkernel technologies which can absorb scientific and technological achievements in the field of information technology synchronously keep up with advanced concept of software architecture, so as to promote the development of GIS technology and software.

At the same time, based on micro kernel technology, we build the atom granularity function module of management and automatic loading, which has realized the function of resource-efficient scheduling and ordering management, which would allow GIS application to have a laid foundation.

Acknowledgements

References

[1] Sun, B., Wu, H., Zhao, H. and Hu, X. (2010) Research and Application on Plug-In Technology in OpenSim. *Proceedings of International Conference on Audio Language and Image Processing* (*ICALIP*), Shanghai, 23-25 November 2010, 1392-1396.

[2] Wang, M., Nie, Y.M. and Yuan, H. (2011) Research on the Urban Land Use Status and Potential Evaluation Information System Solution Based on OSGI Plug-In—A Case Study of Zhaoyuan, Shandong Province. *Advanced Materials Research*, **356-360**, 2812-2815.

[3] Laplante, P., Gold, A. and Costello, T. (2007) Open Source Software: Is It Worth Converting? *IT Professional*, **9**, 28-33. http://dx.doi.org/10.1109/MITP.2007.72

[4] Levina, O. and Stantchev, V. (2009) A Model and an Implementation Approach for Event-Driven Service Orientation. *International Journal on Advances in Software*, **2**, 288-299.

[5] Zhang, F. and Wu, X.C. (2013) A Microkernel-Based Resource Loading Manager on Geographic Information Technology Platform. *Proceedings of* 3*rd International Conference on Consumer Electronics*, *Communications and Networks* (*CECNet*), Xianning, 20 November 2013, 15-18.

[6] Zhang, S., Xing, T.Y., Tao, L.F. and Zhang, F. (2013) OSGIS & FOSS in LBS Design. *Proceedings of the* 2013 *International Conference on Software Engineering and Computer Science*, Yichang, 27 September 2013, 261. http://dx.doi.org/10.2991/icsecs-13.2013.57

[7] Zhang, S., Xing, T.Y. and Tao, L.F. (2014) FOSS-Based WSN Indoor Localization Platform Design. *Proceedings of the* 2014 *International Conference on Information GIS and Resource Management* (*ICGRM* 2014), Guangzhou, 3-5 January 2014, 401-408.