

# Remote Sensing Images Data Integration Based on the Agent Service

Binge Cui, Chuanmin Wang, Qiang Wang

College of Information Science and Engineering, Shandong University of Science and Technology, 266510 Qingdao China  
cuibinge@yahoo.com.cn

**Abstract.** Remote sensing images have been fully utilized in disaster detection and other domains. However, many researchers cannot find and access the appropriate remote sensing images as they needed. In this paper, we proposed an effective approach to integrate and share the image resources over the Web. Firstly, the image metadata are exposed based on Grid services and the standard metadata specification; secondly, Agent service is introduced to discover and invoke the metadata services dynamically; thirdly, researchers can query and browse or locate the remote sensing images through the service interface. We have developed a service-oriented remote sensing images integration platform, which supports the parallel query and browse of multi-source remote sensing images. Moreover, it provides better availability and extensibility.

## 1 Introduction

Satellite remote sensing technologies are developed rapidly in recent years. Remote sensing images have four basic characteristics: large-area, quasi-synchronous, multi-temporal and low-cost. Due to such benefits, remote sensing images have been utilized extensively in disaster detection, meteorological prediction, resources investigation, environment protection [1]. Remote sensing images are usually stored and managed by different departments, which is difficult for researchers to access or to acquire them [2]. In order to promote the research in ocean sciences, it is urgent for us to integrate and share the remote sensing images over the Web [3].

Remote sensing image metadata describe the important characteristics of image files, such as the longitude and latitude, coordinate system, projection method, sensor mode, orbit number [4]. All metadata are extracted from the image files or header file. Based on the image metadata, users can determine whether the remote sensing images satisfy their requirements. If we can integrate the image metadata distributed in different departments and share them, then researches will know where the remote sensing images that they needed are located. If users have the access privileges, they can even download the image files from the corresponding stations.

In this paper, we proposed an approach to integrate and share the remote sensing images over the Web. The image providers extract and store the image metadata into the metadata repository using the Metadata Extracting and Archiving service. And

then, the image metadata are exposed using the Metadata Retrieving service. In order to enable researches to access multiple remote sensing images stations simultaneously, we designed an Agent service to take over users' queries and invoke the corresponding Grid services. We developed a platform for the integration of remote sensing images.

## 2 The Conceptual Model for Remote Sensing Images Integration

Metadata integration and sharing is the foundation and precondition of the remote sensing images integration. In this paper, we proposed a novel conceptual model for metadata integration and servicing as shown in Figure 1.

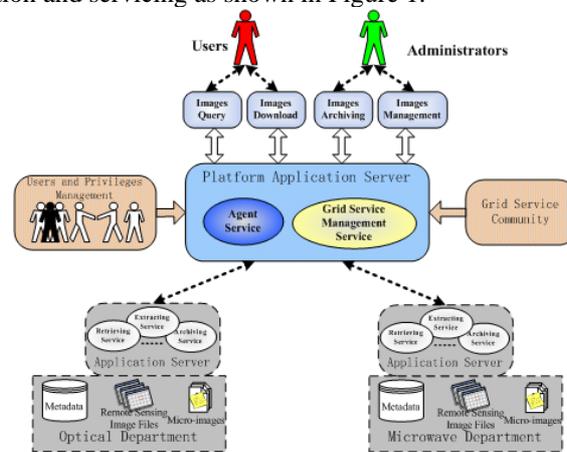


Fig. 1. Conceptual Model for Metadata and Images Integration

In Figure 1, the conceptual model is comprised of five modules:

1. Resource Providers: Each department owns and manages a lot of remote sensing data, including image files, metadata and micro-images. Among them the image files are acquired from some toll stations or network, metadata are extracted or computed from image files, micro-images are made by researchers manually based on the remote sensing images.

2. Metadata Processing Services: These services are deployed on the local application server of each department. Among them Retrieving service retrieves image metadata according to users' queries, Extracting service extracts image metadata from the header files, and Archiving service stores the metadata into the metadata repository. Other services include Metadata Management service, Cloud Cover Computing service, etc.

3. Redirecting Services: Agent service acts as an agent between users (or administrators) and the individual Grid services. For example, users want to query the remote sensing images within a certain zone. When Agent service receives such a request, it will search all available Retrieving services by referring to the service community, and invoking them to return the image metadata.

4. Grid Service Community: It will answer for the description, organization, storage and management of Grid services. Currently, we use a datasheet to store the registration information of all services. The service community is accessed through the Grid Service Management service.

5. Users and Privileges Management: The system users are classified into four roles: platform administrator, data administrators, registered users and public users.

### 3 Metadata Retrieving Service and Metadata Specification

Remote sensing images can be queried through the common query interface of the integration platform. However, some users want to search the remote sensing images by specifying the range of longitudes and latitudes; others want to specify the location using paths and rows. Moreover, users want to query the metadata for multiple satellites and sensors at one time, and they want to specify different parameter sets for each satellite or sensor. To meet the above requirements, we defined two search operations with extensible parameters based on XML schema. One operation is “MetadataSearchByLL”, and its input parameters are shown in Table 1.

**Table 1.** Input Parameters for the Service Operation “MetadataSearchByLL”

Parameter Name	Date Type	Comment
StartDateTime	DateTime	
EndDateTime	DateTime	
WestBoundaryLongitude	Float	
EastBoundaryLongitude	Float	
NorthBoundaryLatitude	Float	
SouthBoundaryLatitude	Float	
ResultSetScope	Int	0: Images within the selected zone wholly; 1: Images center within the selected zone; 2: Images within the selected zone partially or wholly.
Source	Varchar	
OrbitNumber	Varchar	
SpecialParams	Varchar	Its format is a XML document, which will store the additional query conditions for all selected satellites and sensors.

The input parameters for the operation “MetadataSearchByPR” is similar to that in Table 1, and they use paths and rows instead of longitudes and latitudes to describe the images range.

An example for the format for the parameter “SpecialParams” is as follows:

```
<SpecialParams>
  <SpecialParam>
    <SatName>SPOT_1</SatName>
    <SatType>Optical</SatType>
    <SensorName>HRVIPAN</SensorName>
    <SensorMode>PAN</SensorMode>
    <CloudCover>10</CloudCover>
  </SpecialParam>
</SpecialParams>
```

The output of the operation “MetadataSearchByLL” is also a XML document. Each section of the document corresponds to the result set returned from one metadata table. A typical example for the output parameter is as follows:

```
<DataSets>
  <DataSet SatName="LANDSAT-5" SensorName="TM">
    <Records>
      <Record>
        <Path>400</Path>
        <Row>22</Row>
        <Orbit_Number>95874</Orbit_Number>
        <Average_Cloud_Cover>0</Average_Cloud_Cover>
        .....
      </Record>
    </Records>
  </DataSet>
</DataSets>
```

The output of the operation “MetadataSearchByLL” contains multiple datasets as usual, and each dataset has different data structure. Thus, it is impossible to define the output of the operation using simple data types. By introducing the XML schema for the input and output parameters, we can extend the operation interface optionally. Other client programs can process and display the output results based on the XML schema.

To eliminate the heterogeneities among various types of remote sensing data, we defined the standard metadata specification for optical images and microwave images, and built the mappings between the image metadata and the standard metadata. A partial mapping file is as follows:

```
<Satellite satname="LANDSAT-5" sensorname="TM">
  <Parameters>
    <Parameter>
      <ColumnName> Landsat4WRSPath </ColumnName>
      <DataType> int </DataType>
      <MappingColumn> TrackNumber </MappingColumn>
    </Parameter>
    <Parameter>
      <ColumnName> Landsat4WRSRow </ColumnName>
      <DataType> int </DataType>
      <MappingColumn> FrameNumber </MappingColumn>
    </Parameter>
  </Parameters>
</Satellite>
```

In the above XML document, Landsat4WRSPath and Landsat4WRSRow are the field names defined in the header file of LANDSAT-5; TrackNumber and FrameNumber are the field names defined in the standard metadata specification. The query condition and query result are described in the standard metadata, which assures the independence of the upper layer applications.

#### 4 Agent Service and Dynamic Invocation of Grid Service

In order to improve the access efficiency and data security, most Grid services are deployed on the local application server of each department. Thus, there may be multiple identical Grid services on the Web. Which Retrieving service will be invoked when users submit a metadata query? We believe that all available Retrieving services should be invoked simultaneously. Which Extracting and Archiving service will be invoked when users want to archive one scene remote sensing image? We believe that only the Grid services deployed on the same department as the administrator should be invoked. In this paper, we proposed and implemented an Agent service to implement the dynamic discovery and invocation of Grid services.

Firstly, the Grid services deployed on every application server should be registered using the Grid Service Management service; secondly, Agent service discovers the WSDL address for the Grid service with the specific service name and station identifier; thirdly, Agent service receives and processes the input parameter set for the Grid services, and then invokes the Grid service with these parameters. At last, Agent service takes over the outputs of the invoked Grid service, and returns them to the upper layer Web applications. If the invoked service is Retrieving service, then Agent service should merge the result set firstly. The input for Agent service is shown in Table 2.

**Table 2.** Input Parameters for Agent Service

Parameter Name	Date Type	Comment
ServiceName	Varchar	Grid Service Name
Operation	Varchar	Operation Name
Station	Varchar	It denotes the station that users belong to. For public or registered users, the value should be ""
InputParams	Varchar	Its format is a XML document, which will store the input parameters for the invoked Grid service.

In Table 2, "ServiceName" denotes the invoked service; "Operation" denotes the operation name of the service. "Station" is used to identify the one and only Grid service that should be invoked. For example, an administrator worked in the optical department can only invoke the Extracting service and Archiving service running on the optical application server. If "Station" is assigned a null string, then all related Grid services running on every application server should be invoked, such as for metadata retrieving operation.

Assume that the invoked service is Extracting service, an example for the "Input-Params" is as follows:

```
<InputParams>
  <InputParam datatype="Varchar">
    <ParamName>Satellite</ParamName>
    <ParamValue>LANDSAT-5</ParamValue>
  </InputParam>
  <InputParam datatype="Varchar ">
    <ParamName>Sensor</ParamName>
    <ParamValue>TM</ParamValue>
  </InputParam>
  <InputParam datatype="Varchar">
```

```

        <ParamName>Station</ParamName>
        <ParamValue>OP</ParamValue>
    </InputParam>
    <InputParam datatype="Varchar ">
        <ParamName>HeaderFile</ParamName>
        <Param-
Value>ftp://192.168.130.1/p121r34_5t19920824.met</ParamValue>
    </InputParam>
</InputParams>

```

In C#, we add a Web reference to the Agent service with the reference name “Agent”, and invoke the Extracting service based on the Agent service as follows:

```

Agent.AgentService as = new Agent.AgentService();
as.ServiceInvoke("ExtractingService", "Extract", "OP", "<InputParams>");

```

The effect of these statements is equivalent to that of the following statements:

```

Extracting.ExtractingService es = new Extracting.ExtractingService();
es.Extract("LANDSAT-5", "TM", "OP",
"ftp://192.168.130.1/p121r34_5t19920824.met");

```

Based on the above discussion we can see that, by using Agent service, Grid service can be discovered, bound and invoked dynamically. Thus, any remote sensing data can be accessed and processed through the integration platform as long as they are encapsulated using Grid services and registered in the service community. Moreover, Agent service provides parallel queries and faults tolerance to improve the efficiency and reliability of the Web applications.

## 5 Architecture for Remote Sensing Images Integration Platform

In order to implement the organization, storage, management and sharing of remote sensing data, we designed a deployment architecture for the integration platform. The guiding idea of the design is to improve the performance and availability of the platform, and to assure the security of the remote sensing data that possessed by different departments.

In Fig. 2, Agent service and Grid Service Management service are deployed on the platform application server. The SQL Server 2000 database is deployed on the platform server, which will store the registration information for Grid services, users and administrators. In practice, the IIS Web server, platform application server and database are deployed on the same server to reduce the hardware cost and to improve the system performance.

Retrieving service, Extracting service, Archiving service and other metadata services are deployed on the Tomcat application server in each department. Remote sensing image files and micro-images are stored on the FTP server, which provides strict access control and high security. The images access privileges and metadata are stored in the SQL Server 2000 database. Each user can be granted the access privileges to each kind of remote sensing images. Tomcat application server and database are deployed on the same server, but FTP server is usually a separate physical server to maximize the security of remote sensing images.

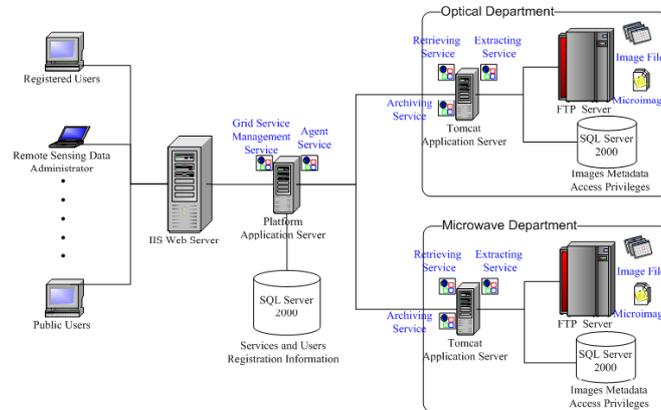


Fig. 2. Deployment Architecture for the Integration Platform

Registered users and public users can query the remote sensing image through the integration platform. The query result includes the image metadata and micro-images, and the user interface is similar to the book information list in Amazon. Registered users can also download the remote sensing image files with appropriate privileges. Data administrators use the integration platform to archive and manage their remote sensing images. The metadata are extracted and stored during the archiving process, and they are integrated and shared through the Retrieving services. The five-tier deployment architecture not only assures the availability and extensibility of the platform, but also improves the performance for the remote sensing images access.

## 6 Related Works

OGSA-DAI is a middleware product which supports the exposure of data resources, such as relational or XML databases, onto grids [5]. Various interfaces are provided and many popular database management systems are supported. The shortcoming for OGSA-DAI is that it doesn't support the parallel query of multi-source data, and our integration platform can query the metadata of satellites from distributed departments.

Wang Na proposed three-tier architecture based on J2EE platform to implement the real-time publication of ocean remote sensing data [6]. In this architecture, remote sensing image files are stored in the large-scale relational database, and they are accessed using ArcSDE spatial database engine. The real-time remote sensing data publication system provides the functions of querying, browsing and downloading. However, users can only query the remote sensing data within one department, and they cannot browse and compare the image data among multiple departments. In other words, there is no integration platform to associate one image provider with another.

Lin Yu-xian et al. proposed architecture for the sharing of remote sensing data in digital city, which consists of data layer, service layer and application layer [7]. In the service layer, it provides data service interface and resource directory. Resource di-

rectory provides navigation, retrieving and locating of remote sensing data resources. The sharing and service platform builds a central node to synchronize the resource directory on each embranchment node. The service platform can support the city application system through the data service interface. The design of the platform doesn't consider the archiving of remote sensing images and metadata, which is the foundation and prerequisite of the sharing of remote sensing data.

## 7 Conclusions

In this paper, we designed and implemented a remote sensing data integration and sharing platform. The image metadata are extracted and exposed using Grid services, and all services are managed by the service community. We proposed an Agent service, which can discover, bind and invoke the corresponding Grid service dynamically in a large-scale, open and ever-changing Grid environment. The partial input and output parameters are described using XML document, which enhances the extensibility of the user interface for service operations. Our work differs from those existing systems in that the integration platform supports the parallel query of multi-source remote sensing images, which improves the query efficiency remarkably.

## References

1. Wang Xiaohua, Wang Shudong, Wang Wei. Study on Remote Sensing Image Metadata Management and Issue. IEEE International Geoscience and Remote Sensing Symposium, 2005
2. F Zhu, M Turner, I Kotsiopoulos, et al. Dynamic Data Integration: A Service-Based Broker Approach. Int. J. Business Process Integration and Management, 2006, 1(3): 175~191
3. J Yu, Y Han. Service Oriented Computing – Principles and Applications. Beijing: Qinghua University Press, 2006, 261~298
4. China Remote Sensing Satellite Ground Station. Online Catalogue System. [http://cs.rsgs.ac.cn/cs\\_en/query/query\\_map.asp](http://cs.rsgs.ac.cn/cs_en/query/query_map.asp), 2004
5. OGSA. the OGSA-DAI Project. <http://www.ogsadai.org.uk/>, 2005
6. Wang Na. Design and Implementation of Real-Time Publication System of Ocean Remote Sensing Data: [Master Thesis]. Hangzhou: Zhejiang University, 2006
7. LIN Yuxian, Li Qi. the Services Sharing of Remote Sensing Data in Digital City. Computer Science, 2007, 34(4): 121~125

## Acknowledgement

This work is supported by Project of "Taishan Scholar" Funded by Government of Shandong Province.