RESTFul based heterogeneous Geoprocessing workflow interoperation for Sensor Web Service

Chao Yang a,b, Nengcheng Chen a, Liping Di b,*

a State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 129 Luoyu Road, Wuhan, Hubei 430079, China
b Center for Spatial Information Science and Systems, George Mason University, 4400 University Drive, MSN 6E1, Fairfax, VA 22030, USA

1. Introduction

1.1. Sensor Web Enablement

The OpenGIS Web Service (OWS) framework, which is specified by the Open Geospatial Consortium (OGC), provides a series of common Web service architectures and standards for geospatial information interchange and interoperation. Through these specifications, isolated information is expanded into accessible Web resources and user can obtain information through distributed Web services.

In a sensor Web environment, sensors and sensor resources are distributed and heterogeneous. The Sensor Web Enablement (SWE) specifications are the subset of OWS specifications that aims to create an integrated geospatial Sensor Web network where all types of sensors, instruments, imaging devices and repositories of sensor data are discoverable, accessible, applicable and controllable.

SWE consists of a sensor information model and a number of sensor Web specifications. The information model contains four parts: SensorML (Botts and Robin, 2007), Observation and Measurements (O&M) (Cox, 2007a, 2007b), Transducer Markup Language (TML) (Havens, 2007) and Event pattern Markup Language (EML) (Everding and Echterhoff, 2008). The service specifications for sensors connected to the Web are mainly the Sensor Planning Service (SPS) (Simonis, 2007), the Web Notification Service (WNS) (Simonis and Echterhoff, 2006), the Sensor Alert Service (SAS) (Simonis, 2006) and the Sensor Observation Service (SOS) (Na and Priest, 2007).

1.2. Workflow technique for Sensor Web 2.0

In the Sensor Web 2.0 (Mandl et al., 2008) software architecture, the Sensor Web combines sensors and sensor networks with a Service-Oriented Architecture (SOA), which allows users to discover, describe and invoke services from a heterogeneous software platform via a series of technology such as eXtensive Makeup Language (XML), Simple Object Access Protocol (SOAP) or Web Service Description Language (WSDL) and various workflow standards applied in converting the Sensor Web into a desirable geospatial workflow (or a geospatial process chain) (Chen et al., 2010a).

The workflow is designed to describe the whole sensor knowledge transfer process from raw sensor observations to informational products. This process connects a series of operation steps from serving planning and observations, through data processing and information managing (Chen et al., 2010b).

Currently, there are two main workflow industry structural and technical standards in the Web workflow applications field: an XML-based workflow description language (mostly in XML Process Definition Language (XPDL) and Web Service Flow Language (WF-XML)).

*Corresponding author. Tel.: +1 703 993 6114; fax: +1 703 993 6127. E-mail address: ldi@gmu.edu (L. Di).
with commands such as “receive”, “reply” and “invoke” for controlling or orchestrating Web Services. BPEL is an industry standard executable language for orchestrating Web Services into business processes.

A XpDL design workflow uses an XML-based syntax and is specified by an XML schema. The workflow’s components and actions are both described as “Elements”. As a standard workflow modeling language, XpDL endorses process definition visualization and the XpDL based process supports more than 20 kinds of workflow patterns.

Unlike XpDL, BPEL is primarily concerned with the Web service orchestration, BPEL has its own XML based programming syntax, with commands such as “receive”, “reply” and “invoke” for controlling the basic workflow process unit—“Activities”, which directs the interoperation of different Web services and is described only by a series of connection subelements in a BPEL process. BPEL itself defines the syntax for an XML-based programming language. Thus, the Web Services can be organized and directed in an order, which is significant in complex Web service integration. As they differ in structure and operation model, transfer between an XpDL process and BPEL process workflow is hard to achieve.

1.3. REST architecture and RESTful Web workflow

The Representational State Transfer (REST) (Fielding, 2000) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. Distributed Web content is described as a Web resource. The RESTful Web service (Richardson and Ruby, 2007) is gaining increased attention because of this simple Web service publishing and consuming: distributed Web Services are described and referenced as Web resources with a global identifier (e.g., a URI in HTTP), applying HTTP method (e.g., POST, GET, PUT or DELETE) to operate Web resources.

In the RESTful architecture, workflows are the collections of Web resources. A web resource has three defined attributes: the URI for resources, the Internet media type of the data supported by the workflow (like XML) and the set of HTTP operations supported by the workflow. After a workflow is deployed as an operable Web resource with a specified URL, users can create, modify and delete resources, passing data and information through different workflow platforms. The main advantages of applying RESTful architecture in workflow interoperation are ease of implementation, design and operation.

1.4. Problem and organization

The increasing adoption of the service-oriented architecture (SOA) paradigm enables service composition and scalable Web workflows as a means of building distributed applications (Rosenberg et al., 2008). Under the SOA environment, the workflows are applied as Web Services, the interoperability of workflows built on the interface chaining. Although there are numerous transgressions in daily Web practice, the consensus of the Web development community backs the resource-oriented paradigm proposed by the REST model. The simplicity of the single interface, single protocol (HTTP methods) approach has potential benefit in complex scenarios.

With the explosion of Earth observation information, the trend is to organize individual Sensor Web Services into aggregated geospatial workflows to offer advanced operations for complex application scenarios. Because the workflows vary in character it is hard to interoperate them in a unified workflow engine. Usually, the solution under the SOA framework is that workflows are sealed and exposed as Web service interfaces for future interoperation. This paper focuses on the interoperation of heterogeneous geospatial workflows in the Sensor Web environment using the RESTful method. As opposed to the Web service chaining model in SOA, this paper abstracts and presents the concrete workflow process as a virtual process. This process can be published and fed as a Web resource; the resource can be created, accessed and operated by users and then the process executed in a specific workflow engine.

In Section 2, workflow technology and the RESTful architecture style are introduced as the base of the RESTful geospatial Workflow Interoperation System (RWIS). In Section 3, workflow interoperation technique details of RWIS and heterogeneous Sensor Web workflow instances (XpDL based sensor information accessing workflow and the BPEL sensor information processing workflow separately) are described. Then, the benefits of the RWIS architecture, work pattern and evaluation of the on-demand environment observation and monitoring under the Sensor Web Environment are discussed. Finally, Section 5 summarizes this paper and presents the outlook for the future.

2. Methodology

2.1. System architecture

Workflow, or a chain of Web services, has been widely applied in integrating geospatial Web services (Di, 2005). A virtual workflow (or abstraction of the workflow) describes abstract Web Services and data composition and provides ontology information on how the workflow has been composed. The virtual workflow components need to be instantiated in a specific workflow engine. Both XpDL and BPEL support this kind of virtual workflow design—instantiation pattern.

The proposed RWIS (as Fig. 1 shows) has two components: the geospatial workflow model (GWM) for the resource index and a geospatial workflow instance (GWI) for the resource content.
The GWM is the abstract description of the data and geospatial workflows, which consists of a series of data ontology and workflow ontology steps; it has defined the formalizing concept of the relationship between workflows—definition of abstract vs. concrete workflows and data and workflow property roles as constraints or capabilities. Then the GWM can transfer to GWI, with its corresponding workflow engines, and can invoke independent and distributed “true” data and Web workflows.

For instance, an abstract sensor data processing and publication workflow wraps an SOS, a Web Processing Service (WPS) and a Web Coverage Service (WCS) and provides the user with sensor observation data access and processing. The virtual workflow can be published as a Web resource. Once the virtual workflow has been invoked by a user, it will be initialized by a distributed workflow engine and the workflow engine will execute the workflow by directing the Web service process. The sensor data resource is fed by SOS, then WPS is applied to handle the data and provide processed data product to WCS, finally, the WCS provides the user the sensor observation information with a specified spatial bounding area.

2.2. Concrete OGC Web Services into virtual Geospatial workflow

The scientific community can adopt the Kepler workflow to edit, execute, and manage other workflows (Zhang et al., 2007). Diverse workflow standards and models are currently applied in Web Services integration. Even though different workflows are distinct in framework and features, there still is similarity in the way they organize and invoke Web Services. First of all, the structure of a workflow depends on the invoking Web service interface; second, descriptive language is needed when organizing and coordinating Web Services. Most OGC Web services are based on the HTTP-GET/POST binding. The integration of OGC Web services as a service chain relies mostly on invoking HTTP-based services: one service’s output then can be the input of another service.

When designing a virtual Geoprocessing workflow, a series of ontology tags needs to be added: the workflow description languages (BPEL, XPDL, etc.) of integrating Web Services, the Web service’s attributes such as service description, message exchanging method or calling interface, network communication method (HTTP, FTP, etc.) and the input/output data type.

2.2.1. Workflow in the XPDL pattern

The XPDL specification uses XML as the mechanism for interchanging process definitions, which forms a common interchange standard that enables products to continue to support arbitrary internal representations of process definitions. A design workflow is described in an XML document that stores and exchanges the process descriptions and diagrams. XPDL includes two main functional packages: execution details and visualization characters. This study does not primarily concern the XPDL visualization characters. The execution details include standard workflow description elements: “Input/OutputSets”, “Assign”, “Switch”, “Loop”, etc. In the latest XPDL 2.0 version, the XPDL based virtual workflow can be directly interpreted by many XML-extension based workflow engines and can be instantiated in an entity engine or migrated between different workflow execution engines.

2.2.2. BPEL workflow: Web Service orchestration

The BPEL can meet the obligatory demands of a programming language because of its properties like loops (while), algorithm branch (switch) and explicit definition of variable type, e.g., Integer, Float (Chen et al., 2010c). The result of a complete process definition is a BPEL script that will be interpreted by an orchestration BPEL engine like BPEL Power or Active BPEL (Active Endpoints). The ingredient of a BPEL Web service workflow depend on the Web service’s WSDL description file, which provides a series of Web service attributes that includes the input/output messages, operation method, port type, etc. By mapping these Web service’s attributes, the BPEL script language can generate corresponding roles such as “partnerLink”, “Activities”, notification message, etc.

Using the BPEL script language, a whole workflow can be designed by orchestrating the Web service operations in an order: a “partnerLink” element receives messages from clients; a message passes from activities to a partner link. The message is controlled by a BPEL script. For instance, partner link A (SOS) is encoded as specified by OGC’s O&M for observation data; after this activity is done, the data reference will be handled as a message and pass to partner link B (WCS) for the publishing coverage information. The process is a standard Web service that allows programmatic invocation and also can be interpreted by other BPEL engines.

2.3. Workflow integration and interoperation in RESTful

2.3.1. Describing and publishing workflow resource

The Atom technology is applied under the REST architecture to feed and publishing Web resources. Atom applies a pair of related standards. The Atom Syndication Format (Nottingham and Sayre, 2005) is an XML language that describes lists of Web resources as “feed”. Feeds are composed of a number of items, known as “entries”, each with an extensible set of attached metadata. The Atom Publishing Protocol (Gregorio and de hOra, 2007) (APP) is an application level protocol for an HTTP-based approach for creating, updating and editing the “Web feed”.

Web feed, or “syndicated feed”, is a data format used for providing users with frequently updated content. In a Web feed, a workflow is described as an “entry” and the “entry” has a unique URI. To operate contents in that feed, users can issue a HTTP “POST” request to create new content, “GET” requests for retrieving content, “PUT” requests for editing contents and “DELETE” requests for deleting contents.

Atom protocols can publish a workflow as Web resource via the workflow’s WSDL file, which give a brief description of the workflow and provides the workflow recipe in an XML document. The WSDL file not only presents how services are integrated and executed in a workflow, but also provides workflow URL information and services information as “portType” elements. According to this information, the workflow can be wrapped as workflow “content” in a “feed”.

2.3.2. RESTful workflow manipulation pattern

After Atom protocols published workflow resources, the common HTTP method can be applied to operate the workflow resources. The workflow collections can be retrieved independently or as a whole. If there is more than one resource collection, the user can issue “GET/collectionsURI/” to retrieve all available collections; if the user wants to get a resource in a nested collection, they can issue “GET/collectionsURI /ingredients/*” to retrieve the content.

Fig. 2 shows the HTTP methods for the geospatial workflow resources operation under RESTful architecture: the “GET” request to obtain a specific representation of a workflow resource, the “PUT” request to create or update a resource with the supplied document, the “DELETE” request to delete a resource and the “POST” request to submit a workflow to be processed by the identified resource.
3. Experiments

When applying RESTful architecture as the base of coordinating sensor information observation and processing workflow, as Fig. 3 depicted, raw sensor observation information is accessed through SPS and SOS, and then is handled in WPS. The output information is delivered through the WCS portal to the users. SPS and SOS are organized as the Sensor Information Accessing (SIA) Workflow; WPS and WCS are known as the Sensor Information Processing (SIP) Workflow.

3.1. Sensor Information Accessing Workflow

3.1.1. Sensor Information Accessing Service

The SIA workflow aims to provide all steps for sensor planning, discovery, accessing sensor data and encoding data. The workflow consists of SPSs and SOSs. The SPS implementation contains four mandatory operations: “GetCapabilities”, “DescribeTasking”, “DescribeResultAccess” and “Submit”, and three optional operations: “GetFeasibility”, “GetStatus”, “Update” and “Cancel”. SPS “GetCapabilities” offers an operation for tasking management, by which the user can access sensor observation information in an area of interest or at a special time and can meet requests for dynamical sensor information retrieval. Then the user can send an observation task via “DescribeTasking” to SPS with request parameters. “Submit” is the interaction between SPS and sensors. After the task is finished, “DescribeResultAccess” exposes the observational data-accessing URL.

SOS has three mandatory core operations: “GetCapabilities”, “DescribeSensor” and “GetObservation”, and two optional transactional operations: “RegisterSensor” and “InsertObservation”. The three mandatory core operations are used to help consumer discover and retrieve sensor and observations; the two optional transactional operations are used to help provider register sensor and observations. SOS standard defines an API for managing deployed sensors and retrieving sensor observation data. SOS provides access to observations from sensors and sensor systems in a standard way that is consistent for all sensor systems.

3.1.2. Design Abstract Sensor Information Accessing Workflow

The abstract SIA process definition mostly relies on the SPS and SOS WSDL profile. Each Web service is described as an entity and has a unique ID; the inputs and outputs organized as a continuous operation in the “WorkflowProcess” element. The XPDL is applied to combine the SPS and SOS as a process chain. The whole process is divided into several “Activities”. As the left part of Fig. 4 shows, there are several process steps in the SIA workflow. First, the user submits a task request through the “GetFeasibility” operation from the SPS server and then the SPS server judges whether the task can be done; if so it sends the executable task information to user. After that, the user can send the results of the “DescribeTasking” operation with the sensors selected and observation task information to SPS, which parses the information and passes the parameters to the SOS server for retrieving the observation data. The observation data would be encoded by invoking “InsertObservation” operation and rewraps sensor information in the form specified by O&M. Then the information will be put into a database that connects with the SOS service, and the “GetObservation” response provides safe access for the user to observations.

The encoded process can be deployed by the OpenWFE engine, which is an open source workflow environment. As is shown in the right part of Fig. 4, the OpenWFE engine executes the SIA workflow for the user by invoking operations provided by SPS and SOS.
workflow and offers a combination function: the user queries the sensor information through SPS service, and then access sensor observation data from SOS.

3.2. Sensor Information Processing Workflow

3.2.1. Sensor Information Processing Service

The OGC WPS service is a standardized interface, which can be seen as accessible online GIS program, for providing Web based GIS computing resources. The WPS can be configured to offer any sort of GIS functionality to clients across a network, including access to pre-programmed calculations and/or computation models that operate on spatially referenced data. WPS has three operations: “GetCapabilities”, “DescribeProcess” and “Execute”. The “GetCapabilities” operation offers the users service level metadata to illustrate the service abilities, and then the user can create a detailed process description via “DescribeProcess”. Users get the final result by carrying out “Execute” with specified parameters.

In this system, the Geographic Resources Analysis Support System (GRASS) and Geospatial Data Abstraction Library (GDAL) programs are deployed on the WPS server (http://swe.whu.edu/wps/) to provide data processing. The operation code is wrapped as Web service and exposed as an interface to users.

As a processing function, WPS can be divided into two parts:

(A) The data processing service can support multi format data and handle diverse processes. GDAL supports all data processing functions: data format translation, data merging, coordinate transformation, etc. The data processed by the service can be delivered across a network, or stored on the local server.

(B) Geospatial analysis is provided by the GRASS program. In this system, the service provides vector processing and analysis operations, such as buffer area analysis, vector map generation and distance calculation.

The Web Coverage Service (WCS) is another OGC-formulated standard interface, which enables interoperable access to offer geospatial “coverages”. It has three operations: “GetCapabilities”, “GetCoverage” and “DescribeCoverageType”. The WCS serves WPS process outputs as inputs. The user or client can get the sensor data coverage map for the designated area through the “GetCoverage” operation. With the synthesis of WCS and WPS, the sensor processing workflow can provide a coverage service with processed geospatial information, which represents space-varying phenomena in forms that are useful for client-side rendering, multi-valued coverage and input to scientific models and other clients.

3.2.2. Design Abstract Sensor Information Processing Workflow

The Oracle JDeveloper BPEL designer can create a concrete sensor processing workflow. The BPEL designer tool enables enterprises to orchestrate and execute Web services and business processes. In the BPEL-based SIP workflow, the WPS and SOS are described as having a “partnerLink” role, the “portType” elements describe the Web service operations, the BPEL process is described in the “sequence” element and the input and output information are “variable” elements. By controlling these roles using the “assign” and “invoke” operations, the whole process can interoperate with corresponding Web Services.

As Fig. 5 shows, two partner links indicate WPS and WCS. After WPS has finished the data processing task, the call-back information will be generated and sent back to the engine so the next step of the workflow can be directed. The WPS output will feed back to WCS, then the WCS coordinates the data for the coverage information. The whole workflow completes the sensor processing and publishing task.

3.3. Heterogeneous workflows interoperation

A. Create and Publish Workflow Resources

The SIA and SIP workflows are published as workflow resources by sending the “POST” request to the collection URL in the APP server. After the full workflow has been created as a workflow collection, it can be retrieved by sending a HTTP GET request to the geospatial workflow service document URL. This XML-based document provides the user with three critical pieces of information: the status of the request, the metadata information of the workflow resource and the address of the workflow resource.

B. Workflow Resource Interoperation

Once the workflow entries have been deployed as workflow resource collections, the resource can be retrieved and instantiated in a concrete workflow engine. The APP wrapped...
Fig. 5. SIP workflow implementation based on BPEL.

Fig. 6. Execution and response of SIA workflow.
workflow can be invoked by issuing a “POST” request to the workflow entry and then instantiated in the workflow engine. Figs. 6 and 7 show the interoperation of the resource pattern workflow. In Fig. 6, user accesses the SIA workflow URI and sends a “POST” operation with demanded observation area and time information to invoke SIA workflow to get observation information. After the SIA workflow finishes, the sensor observation information is delivered by the output XML document. The SIA workflow output (observation information) is the input for invoking SIP (Fig. 7). Then the SIP workflow URI is invoked by another “POST” request and user can obtain the result URL in the SIP output.

4. Case study and discussion

Volcanoes are an important source of gases for the atmosphere. Injection of NO₂ by volcanic eruptions and its subsequent conversion to H₂NO₃ results in stratospheric aerosol formation and can cause significant variations of climate in a much longer time, about several days (Eisinger and Burrows, 1998). A spectacular volcanic eruption, like Mt. Pinatubo in 1991, is associated with Northern Hemisphere surface cooling of as much as 0.5°C on time scales of months to years (McCormick et al., 2005).

As most volcanoes are in uninhabited regions, it is difficult to detect the volcanic eruptions and assess their amounts of gaseous emissions. Satellite remote sensing measurements provide one well-suited opportunity to overcome this difficulty. Furthermore, access observation information from multiple satellites can eliminate space restriction; most meteorological satellites are operating in low earth orbit and cannot provide continuous observation of a particular area.

4.1. NO₂ mapping of Eyjafjallajökull volcano eruption in Iceland

The 2010 eruptions of Eyjafjallajökull are a series of volcanic events at Eyjafjöll in Iceland, which, although relatively small for volcanic eruptions, caused enormous disruption to air travel across western and northern Europe over a period of six days in April 2010. Then the volcano ejections spread into the atmosphere and could be monitored by various meteorological sensors. In this eruption NO₂ observation and mapping scenario, the Ozone Monitoring Instrument (OMI) aboard on Aura satellite was the sensor. It provides many trace gas observations. The OMI data has been published by the National Aeronautics and Space Administration (NASA) in HDF-5 format.

In the proposed NO₂ eruption monitoring experiment, the observation time is 2010 April 20 and 24, and the monitoring area is worldwide. First, the HDF-5 format NO₂ data was accessed by the SIA workflow and encoded as O&M specifies. Then the user’s observation information was processed and transferred using a SIP workflow. SIP supports two kinds of output format—Geotiff and KML. As is shown in Fig. 8, visualization of the KML-based NO₂ observation in Google Earth indicates that after three days of Eyjafjallajökull’s eruption, the NO₂ amount in the air in Europe has a dramatic increase.

4.2. Discussion

4.2.1. The advantages and disadvantages of the RESTFul Sensor Web

In Sensor Web 2.0, the ability to integrate Sensor Web services with other OGC services is emphasized. A workflow can fully fit the demand for conjoint operation of different live geospatial Web services as a single whole. In the geospatial research field, different organizations offer different Web services. The workflow can be used to organize distributed geospatial services as a service chain for multiprocessing. Workflow interoperation is based on invoking a Web services interface. This kind of interoperation is at the service level. When users want to interoperate between different workflow entities, the workflows must be executed individually or an extra computer program applied to connect the workflows.

Organizing workflows and Web services under the RESTFul architecture can be a proper solution for this obstacle. In planning the RESTFul Sensor Web environment in OWS-6, distributed Web
services and workflows were deployed as Web resources nodes; they can be described, discovered and accessed in a centralized workflow management system, which is known as GeoBPMS. Users can retrieve and access Web resource entities to built desirable process tasks; the HTTP-based workflow operation model is easy and open to all kind of Web content.

In an ideal workflow interoperation system, only the abstract workflows are stored, users can operate abstract workflow resources and initialize the resources when they are applied. As the current limitation of software architecture, the abstract workflows cannot be initialized in a commonly used workflow engine, XPDL allows process definitions only on the top level and BPEL does not support nested XPDL process definition. Consequently, a compound workflow cannot be structured using XPDL and applied in a BPEL engine. Although the BPEL is design for a complex business process and can meet the demand of nested workflows. So the transform process from abstract workflow to the specific workflow is restricted.

In this paper, the abstract workflow is designed by XPDL and BPEL and published as a workflow resource by the Atom protocols. Users can issue “GET”, “PUT” and “DELETE” requests to obtain/insert/delete the abstract workflow resources, and then issue a “POST” request for the workflow procedure in a specific workflow engine. As the containers of XPDL-based abstract workflows, OpenWFE and the BPEL engine carry out the role of SIA and SIP workflow instantiation separately.

4.2.2. Heterogeneous workflows for Observation and processing

In this test bed for the RWIS-driven Sensor Information Observation and Processing (O&P) task, users can access sensor data in a planned observation task from an SIA workflow, and then invoke the SIP workflow for mapping and publishing real-time observation. As it was depicted in the NO2 O&P experiment, the conjunction of an SIA workflow and a SIP workflow can respond to the user’s request containing parameters like observation time, data extent and specified output.

Compared to the traditional manual geospatial data processing pattern, which is time consuming and complicated, in this system, observational data can be used in a manual operation with commercial GIS and a Remote Sensing software (e.g., PCI-Geomatica, ENVI, Erdas Imagine, and ArcGIS) process to get the final information on demand. The user can coordinate a heterogeneous workflow not inferior to the commercial software to get a complex process function.

5. Conclusions and outlook

This paper proposes an interoperation model for distributed and heterogeneous geospatial workflows using RESTful architecture. Under the Sensor Web environment, the Sensor Web services can be organized as a geospatial workflow to provide a multi-function operation. In this paper, the SPS and SOS are organized as SIA workflow and the WPS and WCS are deployed as SIP workflow. The proposed RWIS achieves heterogeneous workflow interoperation in an OpenWFE-based SIA workflow and BPEL-based SIP workflow. In this RWIS-driven volcano NO2 eruption O&P task, the collaboration of SIA and SIP workflows performs efficiently in the NO2 monitoring scenario. The RWIS could be a proper system for coordinating and interoperating distributed and heterogeneous workflows.

In the future, a mechanism for semantic workflow description and registration should be considered. A semantic workflow description can offer users a more accurate result when retrieving a workflow resource. Then, the RWIS architecture needs to add the support of process models, which are also important computing resources and providing users with intelligent function. Finally, the development of a geospatial workflow interoperation platform that can merge and coordinate various workflows to provide valuable analysis functions in the environment research field should be promoted.

Acknowledgements

This work was supported by grants from the National Basic Research Program of China (973 Program) (No. 2011CB707101), National High Technology Research and Development Program of China (863 Program) (No. 2011AA010500), National Nature Science Foundation of China (NSFC) program (No. 41171315, 41023001 and 41021061), and the NASA AIST program of the United States (grant #NNX06AG04G, PI: Dr. Liping Di).

References
