Geoinformatica
DOI 10.1007/s10707-011-0123-x

Automatic geospatial metadata generation for earth science virtual data products

Peng Yue · Jianya Gong · Liping Di · Lianlian He

Received: 30 April 2010 / Revised: 3 January 2011
Accepted: 21 January 2011

Abstract Recent advances in Semantic Web and Web Service technologies has shown promise for automatically deriving geospatial information and knowledge from Earth science data distributed over the Web. In a service-oriented environment, the data, information, and knowledge are often consumed or produced by complex, distributed geoscientific workflows or service chains. In order for the chaining results to be consumable, sufficient metadata for data products to be delivered by service chains must be provided. This paper proposes automatic generation of geospatial metadata for Earth science virtual data products. A virtual data product is represented using process models, and can be materialized on demand by dynamically binding and chaining archived data and services, as opposed to requiring that Earth science data products be physically archived. Semantics-enabled geospatial metadata is generated, validated, and propagated during the materialization of a virtual data product. The generated metadata not only provides a context in which end-users can interpret data products before intensive execution of service chains, but also assures semantic consistency of the service chains.

Keywords Geospatial web service · Semantic web · Metadata generation · Virtual data product · Service chain · Geoprocessing workflow

P. Yue (✉) · J. Gong
State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing,
Wuhan University, 129 Luoyu Road, Wuhan, China 430079
e-mail: geopyue@gmail.com

L. Di
Center for Spatial Information Science and Systems (CSISS), George Mason University,
4400 University Drive, MS 6E1, Fairfax, VA 22030, USA

L. He
Department of Mathematics, Hubei University of Education,
Nanhuan Road 1, Wuhan, Hubei, China 430205

Published online: 02 February 2011

1 Introduction

The scientific research is increasingly supported by distributed networks of interoperable services [1]. Services allow the easy plug-and-play of new functions by following the Service-Oriented Architecture and should be utilized to support the geospatial Cyberinfrastructure [2]. A number of interoperable services have been available to the geospatial community, most notably the Open Geospatial Consortium (OGC) standards-compliant Web services. In a service-oriented environment, geospatial data are often consumed or produced by complex, distributed scientific workflows or service chains that span discipline boundaries [3]. For example, a data product for landslide susceptibility can be derived from a Digital Elevation Model (DEM) and Landsat Enhanced Thematic Mapper (ETM) imagery by chaining distributed geoprocessing services such as slope computation and the Normalized Difference Vegetation Index (NDVI). Using Semantic Web [4] technologies, the semantics of data and services can be machine-understandable so that geospatial data and individual geoprocessing services can be discovered and chained automatically to generate on-demand data products.

Yue et al. [5] have presented the relations among geoprocessing workflows, geospatial process models, and virtual data products. A virtual data product represents a geospatial data type (e.g. terrain slope) that a process model can produce, not an instance (e.g. an individual dataset for terrain slope). It can be materialized as an executable geoprocessing workflow or a service chain when all required geoprocessing methods and inputs, often discovered through a geospatial catalogue service, are available. By defining domain concepts to represent the semantics of geospatial Web resources (whether data, Web services, or service chains), the linkage among geospatial data, services, and geoprocessing service chains can be used for the implementation of virtual data products, thus supporting automatic or semi-automatic geospatial service chaining.

In order for the chaining results to be consumable, sufficient metadata for data products delivered by service chains must be provided. For example, geospatial users would like to know before execution of a service chain the spatial projection of a geospatial data product. Existing work focuses on metadata generation for digital resources that are physically archived instead of being generated on demand [6, 7]. In a service-oriented environment, generation of geospatial metadata for virtual data products provides a context for evaluating the applicability of the data products before intensive execution of service chains.

The contribution of this paper is the proposed approach on metadata generation before execution of service chains, which is subsequently referred to as automatic generation of geospatial metadata for Earth science virtual data products rather than archived data. The service chains are generated during the materialization of virtual data products. Unlike existing work that performs semantic evaluation of service chains only during the generation of process models [8, 9], the approach introduces metadata generation, propagation, and validation when process models are transformed into service chains, which can assure semantic consistency of the service chains. The specification of global and local geospatial metadata constraints supports automatic constraints validation and satisfaction for service chains. The proposal of unary and *n*-ary metadata propagation functions formalizes the metadata propagation model, and the inclusion of metadata profile specifies core metadata to be propagated. The results address the issue on semantic evaluation of a chain result in the OGC abstract service architecture [10] by examining metadata of inputs/outputs data of services.

The remainder of the paper is organized as follows. Section 2 describes briefly the concept of virtual data products. Two running examples are introduced to help in understanding the concept, and then motivations are identified. Section 3 introduces background concepts and previous work related to the approach. Section 4 describes critical issues for automatic geospatial metadata generation for Earth science virtual data products. Solutions are addressed in Section 5. Section 6 presents the implementation of the prototype system and result analysis. Conclusions and pointers to future work are given in Section 7.

2 Metadata requirements for earth science virtual data products

2.1 Earth science virtual data products

Geoprocessing workflows are important activities in geoscientific problem solving. In a service-oriented environment, Earth science data products are often consumed or produced by workflow-based geoprocessing service chains. Compared to the physically existed data products, Earth science virtual data products are generated on-demand by taking advantages of automation and dynamism from automatic service composition.

A number of approaches in the literature for automatic service composition distinguish between the generation of a process model and its instantiation into an executable service chain [11–13]. A process model consists of the control flow and data flow among process nodes. The data flow focuses on the data exchange among process nodes, while the control flow concerns the order in which process nodes are executed. A process node represents one type of processing services that share the same functional behaviors: functionality, input, and output. Process models are based on the knowledge of geoscientific modelers. Such knowledge can be captured in the process and represented using ontologies, and be shared and reused by other geoscientific modelers. Using process models, users can produce a required Earth science data product although the product does not really exist in any archive. Therefore, a process model produces a virtual data product, comparable to the physically archived data products. An Earth science data product with which a user is concerned is always subject to spatial and temporal constraints. Such constraints can be used in materializing a virtual data product. By propagating these specifications down to each process node of a process model binding archived data and services, this whole process model is instantiated into a service chain. The instantiation process is called the materialization of a virtual data product.

2.2 Running examples

Two Earth science applications are used as examples to explain the research motivations and to illustrate the proposed solution. To study metadata propagation, there should be examples with different forms of process model. The first involves functions that have only one input data set and produce one output data product (Fig. 1a). The second involves the merging of multiple data sets to produce a single output data product (Fig. 1b).

Figure 1a shows the process model for deriving a data product for terrain slope (Terrain_Slope) from input DEM data (Terrain_Elevation). The materialization of the virtual data product for terrain slope, using metadata specification for Dimond Canyon, CA, United States on 2005 January 10, is shown in Fig. 1c. The slope computation service has preconditions: the input terrain elevation must be in the GeoTIFF data format with the EPSG:4326 geographic coordinate reference system. The archived DEM data serving as input data is available in HDF-EOS file format with the EPSG:32610 Universal Transverse Mercator projection. The processing of data sets in a ready form requires the use of some

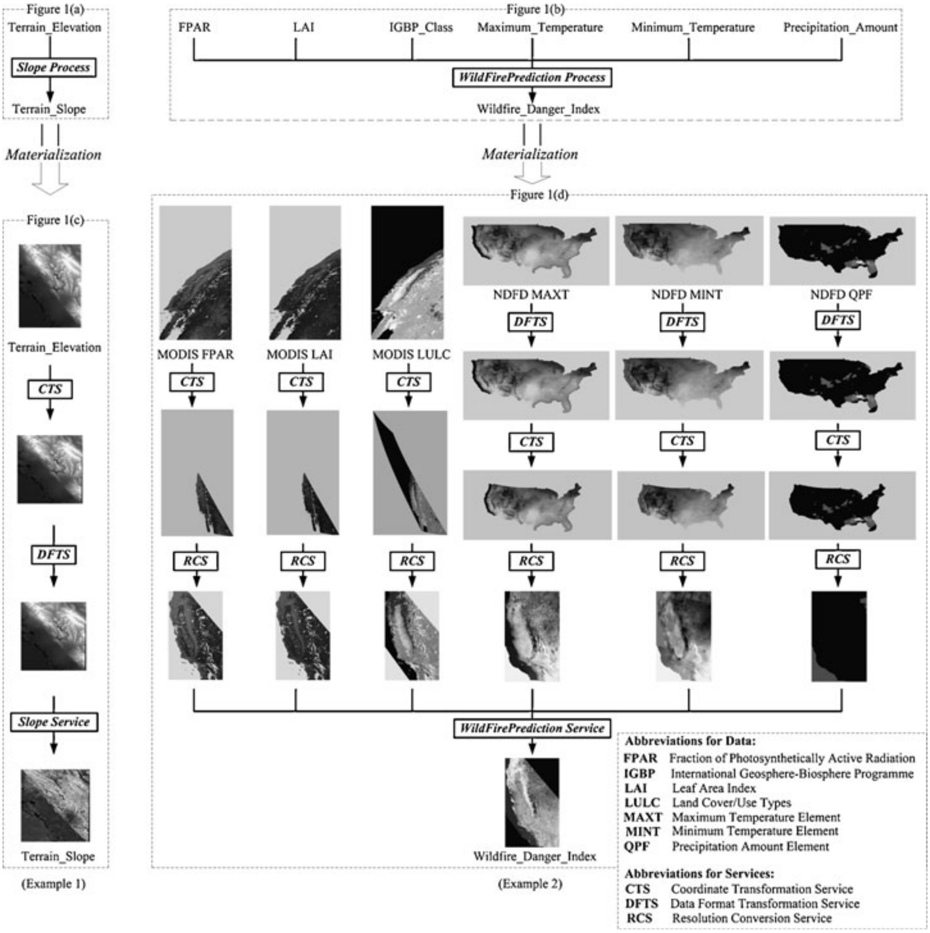


Fig. 1 Graphical representation of process models and materialization results: **a** a process model with one input data set and one output data set; **b** a process model with multiple input data sets and one output data set; **c** a materialization result for the virtual data product in (a); and **d** a materialization result for the virtual data product in (b)

general geospatial data processing services, the so-called data reduction and transformation services, including data format conversion, coordinate system transformation, and resampling/interpolation/regridding. In this example, the Coordinate Transformation Service (CTS) and Data Format Transformation Service (DFTS) are introduced into the service chain in order to transform the DEM data into a form that can be readily accepted by the slope computation service.

Figure 1b shows a process model for wildfire prediction as a virtual data product yielding the wildfire danger index (Wildfire_Danger_Index). This input data consists of the weather and remote sensing data: leaf area index (LAI), fraction of photosynthetically active radiation (FPAR), land cover/use types (IGBP_Class[1]), daily maximum temperature (Maximum_Temperature), daily minimum temperature (Minimum_Temperature), and

[1] Land cover classes defined by the International Geosphere-Biosphere Programme (IGBP)

precipitation (Precipitation_Amount). Figure 1d is an instance of this virtual data product when a user provides spatial and temporal metadata specifications Bakersfield, CA, United States and 2006 August 26, respectively. The National Oceanic & Atmospheric Administration (NOAA) National Digital Forecast Database (NDFD) can provide the weather data: Maximum Temperature Element (MAXT), Minimum Temperature Element (MINT), and Precipitation Amount Element (QPF). The operational NDFD data are stored in the GRIB2 data format with a Lambert conformal coordinate reference system and a spatial resolution of 5-km. The National Aeronautics and Space Administration (NASA) Earth Observing System (EOS) Moderate Resolution Imaging Spectroradiometer (MODIS) products can provide FPAR, LAI, and Land Cover/Use Types (LULC). The operationally available NASA data in the Land Processes Distributed Active Archive Center are stored in HDF-EOS data format, in a sinusoidal grid coordinate reference system at a spatial resolution of 1-km. The MODIS grids are stored as tiles, each covering approximately 1200 by 1200 km^2. These data are accessible through a standards-compliant service, the Web Coverage Service (WCS) [14]. As illustrated in Fig. 1d, the following data reduction and transformation services, CTS, DFTS and Resolution Conversion Service (RCS), are introduced into the service chain to transform the NDFD and MODIS data into a form that can be readily accepted by the wildfire prediction service.

2.3 The need for metadata for earth science virtual data products

In the examples described above, the materialization results of virtual data products involve multiple Web-based geoprocessing steps on diverse sources of geospatial data, which raises an issue of generating metadata for virtual data products to evaluate the applicability of the derived data products. The generated metadata provides a context to answer questions from geospatial users, for example

(1) Can you tell me some detailed metadata information—like the temporal and spatial coverage, resolution, format, or map projections—about the Earth science data products (e.g., terrain slope or wildfire prediction data in Section 2.2) that will be generated by service chains? Metadata allows a provider to describe geospatial data and services so that users can understand the assumptions and limitations and evaluate the applicability of those data and services for their intended use [15]. Therefore, providing enough metadata for Earth science products generated by the service chains is very important. It can ultimately determine the applicability of service chains for delivering the products a user needs.

(2) How can you ensure that geoprocessing service chains are semantically consistent if some individual geoprocessing services in the service chain can deal with geospatial data only in a certain file format or with a certain map projection? Assume a geospatial expert, Jack, knows that a WCS can provide DEM data, and he wants to get a data product for terrain slope. Jack selects a slope process model that can derive terrain slope from input DEM. For the slope process model, many individual services may be available as functional implementations. However, each service may have its own metadata constraints for the input data. For example, the slope computation service in the first use case has preconditions for its input DEM data: a particular file format and spatial projection. Such a situation is often observed in the geospatial domain, and is due to the heterogeneity of geospatial data. In Jack's scenario, when the WCS providing DEM data and slope computation service used in the first use case are selected, Jack needs to ensure that the geoprocessing service chain consisting of

these services is semantically consistent, so that the execution of this service chain can generate meaningful results. Therefore, metadata constraints should be checked before the execution of the service chain, and in case of failed validation, may be satisfied by inserting some data processing functions.

(3) Is it possible to identify some sub-chains whose output data already exist, thus preventing those sub-chains from being executed? Using the metadata generated for the intermediate data products, a metadata catalogue service such as the OGC Catalogue Services for the Web (CSW) [16], can be used to check whether or not specified data products are available. If these data products are available, the available data can be used instead of executing sub-chains.

With the metadata generated for virtual data products, geospatial users can now have more information on both derived data products and semantic consistency of service chains. For example, Jack would know before intensive execution of a service chain that the terrain slope data is in the GeoTIFF data format with the geographic coordinate reference system.

3 Background and related work

3.1 Semantic web approach for earth science virtual data products

An important aspect of developing Earth science virtual data products is the capture, ontological representation, and use of process model knowledge. An ontology is “a formal, explicit specification of a conceptualization” [17] that provides a common vocabulary for a knowledge domain and defines the meaning of the terms and the relations between them. Thus, ontologies can be used to represent semantic knowledge. The semantics of virtual data products are linked the semantics of geospatial data, services, and service chains, so that semantic relations among geospatial data, services, and geoprocessing service chains can be used for the dynamic discovery and composition of services to materialize virtual data products.

Several approaches have been proposed for semantic descriptions of geospatial data, services, and service chains, mainly using the Description Logic (DL) [18] based ontology approach, e.g. Kolas et al. [19], Lemmens et al. [8], Lutz [9]. The basic elements of description logic are *concepts*, *roles*, and *constants*. In the Web ontology context, they are commonly named *classes*, *properties*, and *individuals* respectively. Concepts group individuals into categories, roles stand for binary relations of those individuals and constants stand for individuals. The logical reasoning, called TBOX (Terminological Box) reasoning, supports determination of the subsumption, equivalence, and disjointness relations between concepts. The other type of reasoning, ABOX (Assertional Box) reasoning, determines whether a particular individual is an instance of a given concept description or relations between individuals.

The Web Ontology Language (OWL) [20], recommended by the World Wide Web Consortium (W3C) as a standard Web ontology language, is designed to enable the creation of ontologies and the instantiation of these ontologies in a description of Web resources. Description logic is the foundation of the OWL knowledge representation formalism. OWL is an extension of the Resource Description Framework (RDF) [21]. RDF is a basic data model that identifies objects (resources) and their relations to allow information to be exchanged between applications without loss of meaning. It is based on a graph model composed of triples (Subject, Predicate, and Object).

In Yue et al. [22], geospatial DataType and ServiceType ontologies are defined to provide semantic descriptions of geospatial data and services. Geospatial DataType ontology conceptualizes scientific meanings of distributed geospatial data; thus, it can be used to annotate the semantics of input and output data in a geospatial service operation. Furthermore, the DataType ontology can be enriched with metadata ontologies to allow more precise description of geospatial data, and serve as a relay structure to convey the value of exchanged Web Services Description Language (WSDL) messages as in the service grounding part of OWL-S [23], an OWL-based Web Service ontology. WSDL is a standard for syntactic description of Web services [24]. Geospatial ServiceType ontologies are defined according to the scientific problems that the geospatial services focus on solving. Geospatial DataTypes and ServiceTypes can be used to represent the data, functional, and execution semantics of geospatial services [25] (Fig. 2). Data semantics are the semantics of the input and output data of a geospatial service operation, and thus are represented using geospatial DataTypes. The execution semantics of a geospatial service can be specified using the metadata statement in the preconditions and effects. For example, the preconditions for a slope computation service in Fig. 2 specify that the input terrain elevation should be in the GeoTIFF data format with the EPSG:4326 geographic coordinate reference system. Geospatial ServiceTypes can be used to annotate the functionality of a geospatial service operation.

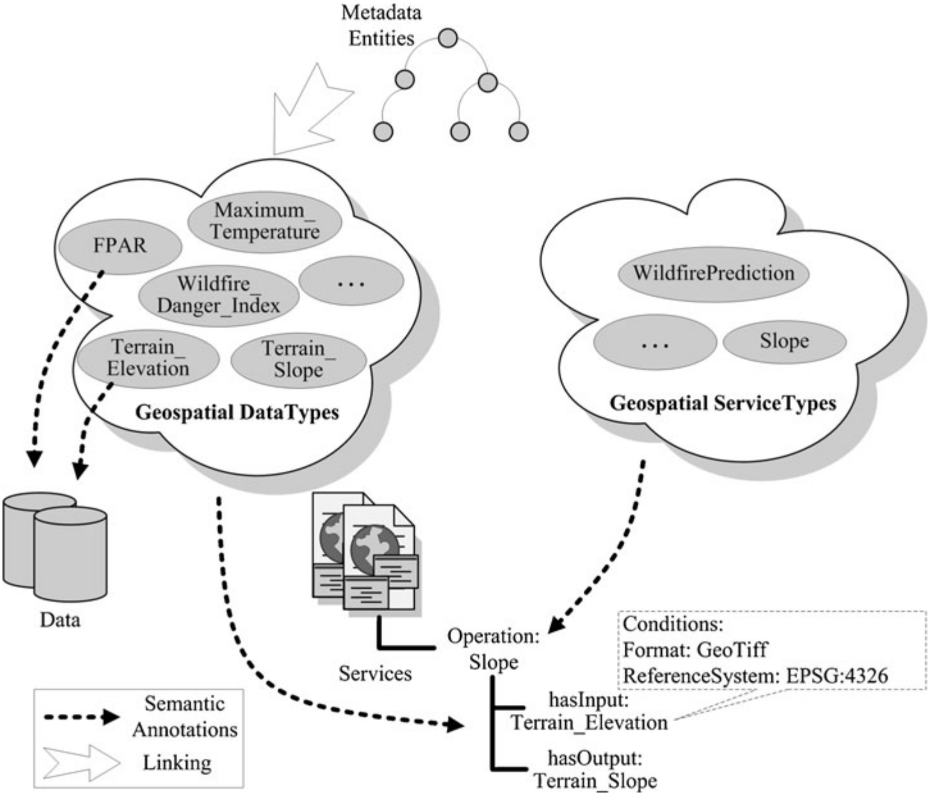


Fig. 2 Semantic descriptions for geospatial data and services (revised from Yue et al. [5])

Semantic Web technologies, in particular OWL and OWL-S, are used to represent semantics for geospatial data, services, and service chains. OWL-S consists of three main parts: service profile, service model (i.e. process), and service grounding. For the wildfire prediction service,[2] the geospatial DataType (Wildfire_Danger_Index) and ServiceType (WildFirePrediction) are linked into the OWL-S descriptions. The service grounding part of OWL-S provides information on how to bridge the input/output ontology concepts to the WSDL syntactic input/output message using Extensible Stylesheet Language Transformations (XSLT) [26]. A process can be either atomic or composite. Atomic process ontology in OWL-S describes the behavior of an atomic service, while a composite process is a collection of subprocesses or atomic processes with control and data flow relationships. Both atomic and composite processes can be described through service profile ontology by their functionalities, inputs, outputs, preconditions, and effects. In Fig. 3, the control flow is represented by control constructs such as Sequence and Split-Join, and the data flow is specified by input/output bindings using an OWL class such as ValueOf to state that the input to one subprocess should be the output of the previous one within a sequence. Examples on how control flow and data flow are encoded in OWL-S descriptions are also illustrated in Fig. 3. The semantics for a geospatial service chain can be represented using composite process ontology. It is noted that the purpose is not to propose new ontologies for semantic descriptions of data, services, and service chains. Rather, the existing set of example ontologies is exploited to illustrate automatic metadata generation for virtual data products.

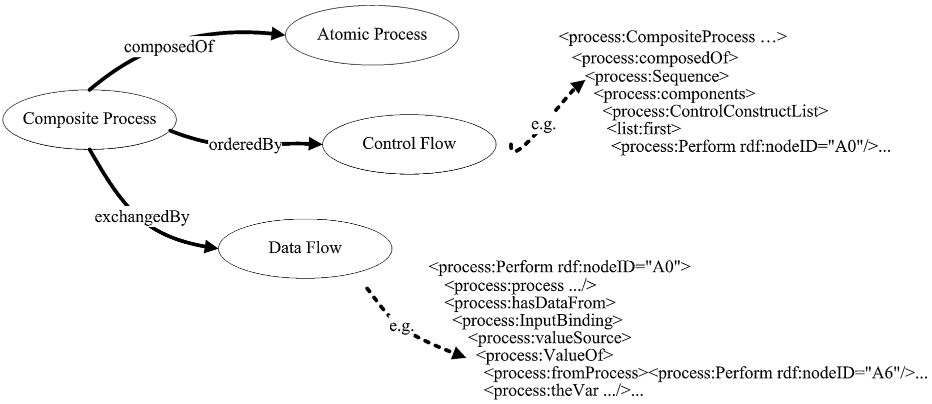
Both atomic and composite processes can be used to represent virtual data products. They can be can be bound to a concrete geoprocessing service chain through data and services discovery. The data and services discovery are enabled by a semantics-enhanced geospatial catalogue service [5].

3.2 Semantic evaluation of service chains

The importance of semantics on accessing and integrating geospatial information has long been recognized [27–29]. There are two levels of interoperability in the services: syntactic interoperability and semantic interoperability [10].The former requires that there is a technical connection, i.e., that the data can be transferred between Web services. It does not provide an interpretation for the content transferred in the connection. The latter assures that the contents of data and services are correctly understood when data/services are connected [22]. Sometimes one can argue the structural interoperability by highlighting the schematic heterogeneity (different message types). In the interoperability literature [27], semantic heterogeneity can be divided into cognitive and naming heterogeneities, and naming heterogeneity sometimes can be further subdivided into syntactic (different symbols) and structural (different expressions) [29].

So what is the semantic problem addressed in this paper? As mentioned in Section 3.1, geospatial DataType and metadata ontologies have been used for semantic descriptions of geospatial data, and they can address the cognitive and naming heterogeneities. The focus in this paper is on semantic issues of services/chains rather than geospatial data. The emphasis is on using the semantics of inputs, outputs, preconditions, and effects of services, together known as IOPE semantics [11–13], for semantic evaluation of service chains.

[2] An example OWL-S file for the wildfire prediction service is available at http://www.laits.gmu.edu/geo/ontology/owls/ap/v1/wps_wildfireprediction.owl.



```
<process:CompositeProcess ...>
  <process:composedOf>
   <process:Sequence>
    <process:components>
     <process:ControlConstructList>
      <list:first>
       <process:Perform rdf:nodeID="A0"/>...
```

```
<process:Perform rdf:nodeID="A0">
  <process:process .../>
  <process:hasDataFrom>
   <process:InputBinding>
    <process:valueSource>
     <process:ValueOf>
      <process:fromProcess><process:Perform rdf:nodeID="A6"/>...
      <process:theVar .../>...
```

Fig. 3 Semantic descriptions for geospatial service chains

Semantic evaluation of service chains can exist at the type and instance level when using process models for representing virtual data products and generating concrete service chains from process models. At the type level, the process models should be semantically correct. For example, the input/output binding of geospatial DataTypes in the data flow among process nodes should be semantically matched. Such a match can use relations between DL concepts and is restricted to the TBOX reasoning. During the materialization of virtual data products, individual datasets and services are bound to process models. When checking metadata constraints in the preconditions of individual services using available data instances, ABOX reasoning is used. In the paper, this approach is termed “two levels of semantic evaluation” in the conceptual and instance levels. The work in the paper assumes that process models are already semantically correct and delves into the semantic evaluation of service chains at the instance level. Such an evaluation ensures the so-called semantic consistency of service chains by satisfying the metadata constraints.

3.3 Related work

Generating metadata automatically is an important research problem in the digital library domain, because of the growth in digital resource repositories and rapid growth of accessible digital resources over the Web [6]. Most existing Automatic Metadata Generation Applications (AMeGAs) work has focused on generating metadata for Web pages, checking and analyzing the existing meta tags in the Web site hosting the Web page, or creating metadata from an analysis of the digital resource using, for example, extracting, harvesting and converting tools [7]. Some automation of geospatial metadata generation for archived data has been carried out by combining data preparation, filing, and documentation workflows in a proprietary Geographical Information System (GIS) [30]. The work in this paper focuses on the service-oriented environment and automatic generation of metadata for virtual data products instead of archived data. The emphasis on the semantic questions of the service chain is different from most existing automatic metadata generation applications.

The advocate of services/chains appears in various initiatives for building information infrastructures. The Spatial Data Infrastructures (SDI), with its initial purpose on geospatial data sharing and services [31], can use standards-compliant services to support distributed geoprocessing applications [32]. Grid technology, a distributed computing technology that involves the coordination and sharing of computing, application, data, storage, and network resources across dynamic and geographically dispersed organizations [33], tries to provide middleware services for coordinated problem solving using myriad resources [34]. Cloud Computing hides the underlying complexity of using information technology (IT) resources, and can provide services in different ways including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [35]. Both grid and cloud computing technologies can consolidate the development of the Cyberinfrastructure [2], and SDI and Cyberinfrastructure complement each other [36]. Chaining geospatial service components in the Cyberinfrastructure helps provide transparent and opaque platforms for scientific problem solving, which can advance geospatial Cyberinfrastructure to geospatial cloud computing [2].

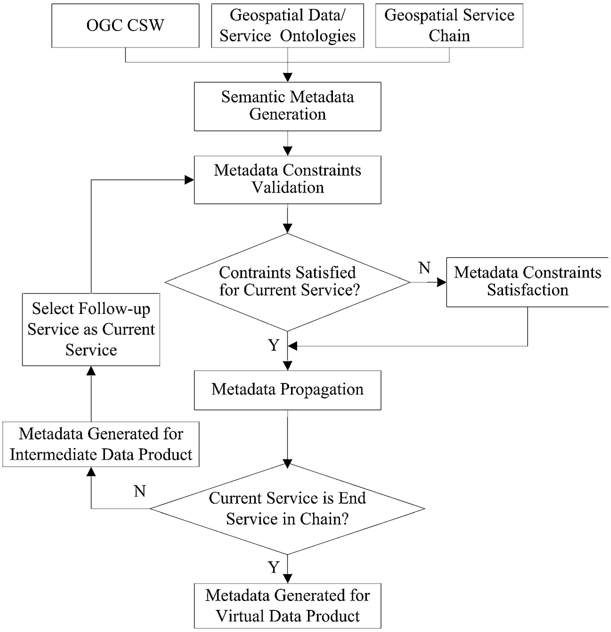
The question of semantics in the service chain has been addressed extensively in relation to automatic service composition [11–13]. Broadly speaking, service composition can address many aspects of Web service provision and use, including discovery, selection, composition, mediation, and invocation. Most efforts published in the literature focus on the methods for automatic service discovery and composition in automatic generation of composite service. Usually, Semantic Web Service technologies are employed in describing services. The semantics for inputs, outputs, preconditions, and effects addressed in the Semantic Web Service technologies are widely used in most Artificial Intelligence (AI) planning methods for automatic service composition [37–39]. As a result, some initial efforts have been made to introduce Semantic Web Service technologies, such as Web Service Modeling Ontology (WSMO), OWL-S, or Web Service Semantics (WSDL-S), into the geospatial domain [8, 22, 40]. Lutz [9] develops a lightweight ontological representation for describing IOPE semantics by combining description logic and First Order Logic (FOL). The matchmaking for service discovery is at the conceptual level and the approach is restricted to TBox reasoning and computationally expensive theorem proving. The service description in this paper follows the description logic in the OWL-S. The execution semantics are described using metadata statements and ABOX reasoning is used. Lemmens et al. [8] use OWL-S for generating abstract composition of services. The abstract composition is transformed into a concrete service chain. The matchmaking happens only when generating abstract composition. The approach in this paper, however, can conduct the semantic evaluation when generating the concrete service chains from abstract composition. Zaharia et al. [40] focus on the mediation during the execution of semantic service chains. Thus, the concrete service chains in the context of this paper could be further refined during the execution. The work in this paper builds on and extends the previous work on Semantic Web Service and service composition by highlighting the role of semantic metadata. The “two levels of semantic evaluation” approach advocates the semantic match at the conceptual level and semantic consistency at the instance level, while existing work emphasizes the matchmaking at the conceptual level. The introduction of metadata generation, propagation, and validation for semantic evaluation of a service chain is a core contribution of this paper.

The concept of virtual data was put forward in the Chimera virtual data system [41], where virtual data refers to data that can be produced on-demand, although not yet materialized. The virtual data can be materialized by applications of computational transformations to the archived data. In NASA funded GeoBrain project [42], the geo-object and geo-tree concepts were proposed. A geo-object consists of data itself, a set of attributes (metadata), and a set of transformation and creation methods that can operate on it. A geo-object can be either archived or virtual. A geo-tree represents a tree structured process workflow that can derive new geo-objects from a set of archived geo-objects. These new, derived geo-objects are called virtual geo-objects. Such virtual geo-objects are equivalent to the virtual data products in the context of this paper. Using metadata constraints, virtual geo-objects or virtual data products can be materialized on-the-fly. From the workflow perspective, the metadata for data used and generated by workflows, can contribute the data products' provenance by linking themselves to the metadata for workflows and their executions [43]. Although most work focuses on analyses of provenance information that was created during execution, such as analyzing ancestry relationships among data products, generation of metadata for input and output data products before execution can support workflow validation [44] and also contribute to the data products' provenance. The use of the components for metadata generation and propagation to augment geospatial data provenance have been explored [45]. The emphasis of this paper is on detailed specification of working mechanism in metadata generation and propagation. Although the present work has some similarity with [44] in that it deals with metadata generation and validation, the context of the work is different. It focuses more on the geospatial Web services and service chains. The OGC CSW and semantic descriptions of geospatial data, services, and service chains are used for geospatial metadata generation, propagation, and validation. If metadata constraints are not satisfied, data reduction and transformation services are chained automatically to create a semantically consistent service chain. Thus the work can complement the work in [44] by suggesting measures to be taken in case constraints are violated.

4 Critical issues for automatic geospatial metadata generation for earth science virtual data products

The flow diagram of Fig. 4 shows a generic process to automatically generate geospatial metadata for Earth science virtual data products. Geospatial data and service ontologies are used to support the generation of semantic metadata. A geospatial service chain can be

Fig. 4 Flow diagram illustrating automatic generation of geospatial metadata for virtual data products


OGC CSW
Geospatial Data/ Service Ontologies
Geospatial Service Chain
Semantic Metadata Generation
Metadata Constraints Validation
Contraints Satisfied for Current Service?
N
Metadata Constraints Satisfaction
Y
Select Follow-up Service as Current Service
Metadata Propagation
Metadata Generated for Intermediate Data Product
N
Current Service is End Service in Chain?
Y
Metadata Generated for Virtual Data Product

represented by linking data and service ontologies together. Using a semantically described service chain, semantics-enabled geospatial metadata is generated, validated, and propagated from the source inputs to the final data product derived by the service chain. The semantic metadata for source inputs is generated by interacting with the OGC CSW. Such metadata is used to check the semantic consistency of metadata constraints added to the current service. If constraints are not satisfied, metadata constraints satisfaction is employed to validating that constraints are satisfied. The metadata for the output of the current service is then derived through metadata propagation. After that, the follow-up service in the service chain is selected as the current service for constraints validation, satisfaction, and metadata propagation. Such a process continues until the metadata for the final data product is derived. During this flow, several critical issues must be explored.

Representing metadata constraints The metadata constraints are defined as metadata specifications that constrain the selection of instances of geospatial data and services during the materialization of process models into service chains. The scope of these constraints in the process model may vary. Some constraints apply to all the data and individual services involved in a process model (e.g. spatial area of interest). In this case, they are called *global metadata constraints*. Others may focus on a specific data set or service (e.g. file format supported); they are termed *local metadata constraints*. In a Semantic Web environment, all these constraints should refer to a semantics-enabled metadata representation. Therefore, an OWL description of geospatial metadata can be used to specify both global and local metadata constraints.

Semantics-enabled metadata generation and propagation Web Service technologies follow the *publish-find-bind* paradigm in the Service-Oriented Architecture. Metadata for geospatial data and services are published at a registration center to support the discovery of geospatial data and services. In the geospatial context, an interoperable service standard for a geospatial registration center is already available, namely OGC CSW. Therefore, generation of metadata should use this existing service standard. The metadata for the source input of service chains can be obtained from OGC CSW, while metadata for intermediate and final data products can be generated through metadata propagation. A collection of core metadata entities to be tracked should be identified. Some metadata information can be propagated along the service chain from the source data to the derived data products, because each atomic processing function/service may change values of only selected metadata elements in source data. All metadata generated should be represented in a formal semantic representation using ontologies so as to support the validation and satisfaction of the metadata constraints.

Validation and satisfaction of the metadata constraints Validation of metadata constraints checks the consistency of the service chains. For example, a geoscientific model wrapped in a geoprocessing service may be valid only in a certain area; therefore, this service must state its working area so that a meaningful result can be produced. The objects that global and local metadata constraints are imposed on differ, and therefore, different strategies must be adopted for validation. Semantic Web Services, the combination of the Semantic Web and Web Services [46], provide mechanisms for modeling services and building process models, and support the correct correlation between data and services. They thus can be used to determine the consistency of constraints added to services. If those constraints are not satisfied, automatic constraint satisfaction strategies to modify the service chain must be developed.

5 Automatic geospatial metadata generation

5.1 Representing metadata constraints

As noted in Section 4, there is a difference between global and local metadata constraints. Both global and local metadata constraints can be represented using the ISO 19115 metadata ontology developed at Drexel University, USA [47]. Global constraints are part of the users' goal, the spatial and temporal constraints of the requested data product produced by the process model, e.g. a wildfire prediction for Bakersfield, CA on the next day. Table 1 is an example of such a request in OWL generated for the wildfire prediction case. This ontological representation specifies the temporal/spatial ranges for which the virtual data product is requested.

Local constraints are the metadata constraints that the input data of an individual service must follow. Such constraints are represented as OWL-S preconditions. OWL-S

Table 1 An example of global constraints for a desired data product

```
<geodatatype:Wildfire_Danger_Index …>
<mediator:hasMD_Metadata>
<iso19115:MD_Metadata>
 <iso19115:identificationInfo>
  <iso19115:MD_DataIdentification><iso19115:dataExtent><iso19115:EX_Extent>
   <iso19115:geographicElement><iso19115:EX_GeographicBoundingBox>
    <iso19115:westBoundLongitude>
     <iso19103:Angle><iso19103:value>-130.69</iso19103:value></iso19103:Angle>
    </iso19115:westBoundLongitude>
    <iso19115:eastBoundLongitude>
     <iso19103:Angle><iso19103:value>-107.84</iso19103:value></iso19103:Angle>
    </iso19115:eastBoundLongitude>
    <iso19115:southBoundLatitude>
     <iso19103:Angle><iso19103:value>32.58</iso19103:value></iso19103:Angle>
    </iso19115:southBoundLatitude>
    <iso19115:northBoundLatitude>
     <iso19103:Angle><iso19103:value>41.67</iso19103:value></iso19103:Angle>
    </iso19115:northBoundLatitude>
  </iso19115:EX_GeographicBoundingBox></iso19115:geographicElement>
  <iso19115:temporalElement>
   <iso19115:EX_TemporalExtent><iso19115:exTemp><iso19108:TM_Period>
    <iso19108:beginning>
     <iso19108:TM_Instant><iso19108:position><iso19108:TM_Position_DateTime8601>
      <iso19108:dateTime8601>2006-08-26T00:00:00Z </iso19108:dateTime8601>
     </iso19108:TM_Position_DateTime8601></iso19108:position></iso19108:TM_Instant>
    </iso19108:beginning>
    <iso19108:ending>
     <iso19108:TM_Instant><iso19108:position><iso19108:TM_Position_DateTime8601>
      <iso19108:dateTime8601>2006-08-26T23:59:59Z</iso19108:dateTime8601>
     </iso19108:TM_Position_DateTime8601></iso19108:position></iso19108:TM_Instant>
    </iso19108:ending>
   </iso19108:TM_Period></iso19115:exTemp></iso19115:EX_TemporalExtent>
  </iso19115:temporalElement>
</iso19115:EX_Extent></iso19115:dataExtent></iso19115:MD_DataIdentification>
</iso19115:identificationInfo></iso19115:MD_Metadata></mediator:hasMD_Metadata>
</geodatatype:Wildfire  Danger  Index>
```

preconditions can be presented using the Semantic Web Rule Language (SWRL) [48], the SPARQL Protocol and RDF Query Language (SPARQL) [49], or other expression languages identified in the syntax of OWL-S. These preconditions are used for checking semantic consistency of services. The local constraints check is, therefore, equivalent to querying the knowledge base, which is a set of facts represented by the OWL, to check whether some condition for the input OWL individual is satisfied. ABOX reasoning is then used to infer implicit knowledge from the knowledge that is explicitly contained in the knowledge base, e.g. whether the input data file format satisfies the *unionOf* the OWL classes *GeoTIFF* and *NetCDF* (i.e., the format should be either *GeoTIFF* or *NetCDF*). The query of the OWL knowledge base for precondition checking makes SPARQL a more appropriate choice for precondition representation, since SPARQL is the W3C recommended standard query language for the RDF. Using the wildfire service as the example, the file format for some input data are specified using SPARQL, as shown in Table 2.

5.2 Semantic metadata generation and propagation

Using a metadata catalogue service, the input data of the service chain, those that already physically exist in a data archive, can be queried to obtain detailed metadata information, using the global constraints as query filters. For example, the NOAA NDFD and NASA EOS MODIS data for input to the wildfire prediction service can be located using the temporal/spatial ranges identified in Table 1. If the metadata registered in the catalogue service does not have enough detail, a metadata generation component can be used to extract additional metadata from those data encoded in self-describing file formats such as HDF-EOS and GeoTIFF. The detailed metadata information for located data records in the catalogue service is transformed into OWL individuals (similar to the OWL individual in Table 1) and added into the OWL knowledge base to facilitate the precondition checking. Generation of metadata for intermediate and final data products depends on metadata

Table 2 A local constraint for wildfire prediction service using SPARQL

```
<expr:SPARQL-Condition rdf:ID="supportedFileFormat">
 <expr:expressionLanguage rdf:resource="&expr;#SPARQL"/>
 <expr:expressionBody rdf:parseType="Literal">
  <sparqlQuery xmlns="…">
   PREFIX  ...
   SELECT   ?coverage WHERE {
          ?coverage mediator:hasMD_Metadata ?md_metadata .
          ?md_metadata rdf:type iso19115: MD_Metadata .
            ?md_metadata iso19115:distributionInfo ?md_disinfo .
            ? md_disinfo rdf:type iso19115: MD_Distribution .
              ?md_disinfo iso19115:distributionFormat ?file_format .
              ?file_format rdf:type fileformat:HDFEOS }
  </sparqlQuery>
 </expr:expressionBody>
 <expr:variableBinding>
  <expr:VariableBinding>
   <expr:theVariable>coverage</expr:theVariable>
   <expr:theObject rdf:resource="#wildfireprediction_input_mint"/>
  </expr:VariableBinding>
 </expr:variableBinding>
</expr:SPARQL-Condition>
```

propagation. Metadata propagation from the source input data to the final data products throughout the service chain depends on the metadata propagation on each atomic service.

In a service chain represented in Fig. 5, let *slope*(DEM) denote the output TerrainSlope, *ndvi*(ETMBand3, ETMBand4) denote the ETMNDVI, and *landslide*(TerrainSlope, ETMNDVI) denote the LandslideSusceptibility. The functional form representing this chain is

$$\mathrm{LandslideSusceptibility} = \mathit{landslide}(\mathrm{s}\mathit{lope}(\mathrm{DEM}), \mathit{ndvi}(\mathrm{ETMBand3}, \mathrm{ETMBand4}))$$

This functional representation is useful in analyzing metadata propagation. If we define a metadata propagation function for each atomic service, then the propagation of metadata through a service chain, called a metadata propagation model, can be represented as follows:

$$\mathrm{MD}_{\mathrm{LandslideSusceptibility}} = \mathit{func\_landslide}(\mathit{func\_slope}(\mathrm{MD}_{\mathrm{DEM}}), \mathit{func\_ndvi}(\mathrm{MD}_{\mathrm{ETMBand3}}, \mathrm{MD}_{\mathrm{ETMBand4}}))$$

where MD is the metadata description and the *func*'s are a set of metadata propagation functions that modify the MD appropriately for each atomic service in the service chain. A metadata propagation function for each service modifies the metadata for the input data. The modified metadata is then passed to the output data. This functional representation helps in understanding metadata propagation. The metadata of the final data product can be described solely in terms of the source input data and a set of metadata propagation functions. The functional representation of the metadata propagation model described above also implies that, when metadata for source input data is generated from the metadata catalogue service, the metadata propagation functions need only be defined appropriately in order to derive the metadata for a final data product.

The metadata propagation functions themselves must be tailored to specific geo-processing services and particular metadata elements. For the purposes of metadata propagation, it is useful to identify two types of metadata propagation as shown in Fig. 6: unary and *n*-ary functions. A unary function has one input data set, and outputs the requested data product. An *n*-ary function takes *n* inputs to output the requested data product (where *n*> *1*). For cases mentioned in Section 2.2, the slope computation case and wildfire prediction case can be characterized as unary and *n*-ary functions respectively.

It is assumed that when a service processes input data, values for explicitly described metadata elements can be changed while values of other metadata elements are transferred unchanged to the data output by the service. Explicitly described metadata elements can be specified in the execution semantics of geospatial services, i.e. using OWL-S preconditions/ effects. In OWL-S, the effects use the OWL Expression class for representing their values, which is the same as preconditions do. Thus, the processing of both preconditions and

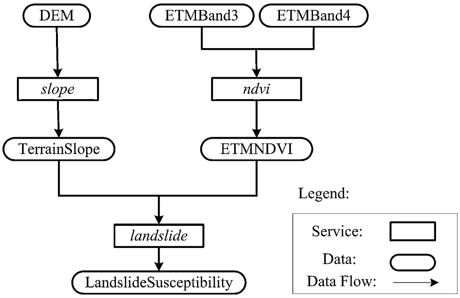


Fig. 5 Graphic representation of a service chain

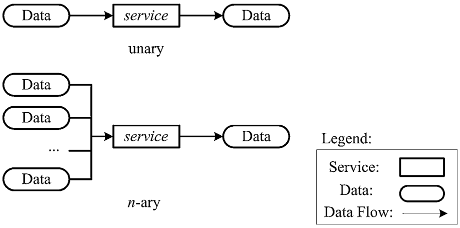


Fig. 6 Types of metadata propagation

effects is similar, except that the metadata in the Expression of effects is processed as the update of the metadata for the output data product. This assumption for metadata propagation is reasonable in real geospatial Web service applications. For example, a slope computation service changes only the thematic meaning of data and a data format transformation service changes only the file format of the data. For the unary case, except for the updated metadata, the metadata of the output data is the same as the metadata of the input data set. In the *n*-ary case, except for the updated metadata, the metadata of the output data is the same as the metadata of the principal input data set. The principal input is identified using the data model of the output and specified as the first input of the process in the OWL-S. For example, assume a geospatial expert wants to know the possibility of having wildfire(s) within a 300 km radius of interested area. An image cutting process, which uses an input polygon to cut the image, creating an image containing the values of the desired area only, is used. The input for image data serves as the principal input data since it uses the raster data model, the same as the output does. If there are multiple inputs using the raster data model, any input among them can serve as the principal input, because these inputs share some common characteristics after coregistration using the data reduction and transformation services.

ISO 19115 defines a full set of metadata elements. In typical applications, however, only a subset of the full number of elements is used. Therefore, ISO 19115 has identified a set of core metadata elements (either mandatory or optional). These elements are required to answer "what" (e.g., data topic), "where" (e.g., location), "when" (e.g., date), and "who" (e.g., contact) questions so that geographical data can be understood without ambiguity. Based on this reference, Table 3 lists the core metadata to be tracked throughout the service chain. An "M" indicates that the information is mandatory. A "C" indicates that the information is mandatory under certain conditions. An "O" indicates that the information is optional. The metadata tracked in the context of this paper focuses on the virtual data products instead of physical data products. Therefore, the metadata tracked has less

Table 3 Core metadata information to be tracked

Metadata information	Tracking demand
Identification	M
Constraints	O
Data quality	O
Maintenance	O
Spatial representation	C
Reference system	M
Content	O
Portrayal catalogue	O
Distribution	M

emphasis on the “when” and “who” questions, and aims to provide basic minimum metadata information for interpreting and evaluating a derived data product. Identification information is contained in the geospatial DataTypes to identify the topic of the data. The reference system and distribution information shows the spatial projection and file format metadata of the data products. In a simple case like the slope example, identification, reference system, and distribution information are enough metadata for the final data product for terrain slope. In a complex case like the wildfire prediction example, the wildfire prediction function in a service may work only in a certain area. Under this condition, bounding box and resolution as the spatial representation information are mandatory. Other metadata information such as constraints, data quality, and maintenance is optional. For example, if a general assessment of the quality of the derived data product is needed, the lineage information may be recorded during the metadata tracking. However, in order to allow automatic processing of lineage information, a new provenance information model, beyond the current lineage model in ISO 19115, has to be established. This is outside the scope of this paper. For a detailed account on how to recording lineage information, see [45]. To support the flexibility of the metadata elements tracked, a metadata tracking profile, which can be updated by users and loaded at runtime, is created. The metadata tracked is guaranteed to be complete only as required by this profile.

5.3 Validating that constraints are satisfied

Compliance with the global constraints is validated by setting query filters when locating the input data of the service chain from the metadata catalogue service. The local constraints are validated through the OWL-S precondition checking mentioned in Section 5.1. When local constraints are not satisfied, an automatic constraint satisfaction strategy can be employed to modify the service chain. In the Earth science domain, data reduction and transformation services such as coordinate transformation service, data format transformation service are common to most geospatial analysis, data mining, and feature extraction applications. They can modify the data to satisfy the metadata constraints. The rule for using these services applies for all geospatial users, i.e., they can be used whenever the corresponding metadata constraints are not satisfied.

The use of only an individual data reduction and transformation service to satisfy a particular metadata constraint is simple and requires only the insertion of this service before the constrained service. However, when multiple data inputs and multiple metadata constraints are involved, this problem becomes complex due to the possible interactions among these services and the context sensitivity of applications.

The flow diagram of Fig. 7 shows the process for automatically validating that geospatial metadata constraints are satisfied. It is made up of two loops. The outer loop is controlled by data input to the current service. For a given data input, the inner loop performs the precondition check (i.e., validating that metadata constraints are satisfied) for each precondition constraining this input.

When the precondition check fails, an appropriate data reduction and transformation service is inserted into the service chain. The insertion of the data reduction and transformation service to satisfy certain metadata constraints is implemented as a domain control procedure (e.g., some source code in the computer software program). The domain knowledge on the common usage of these data reduction and transformation services, therefore, is embodied in these procedures. To make the proposed flow work, preconditions for each geoprocessing service must be defined modularly. Each precondition should make only one type of constraint on one input parameter. Such a requirement facilitates the

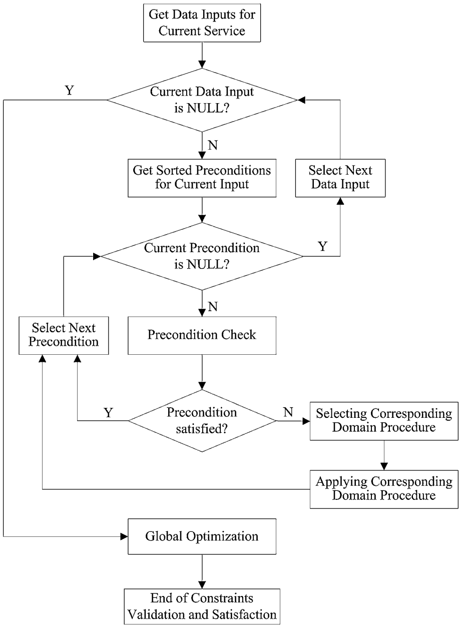

Get Data Inputs for Current Service
Current Data Input is NULL?
Y
N
Get Sorted Preconditions for Current Input
Select Next Data Input
Current Precondition is NULL?
Y
N
Select Next Precondition
Precondition Check
Precondition satisfied?
Y
N
Selecting Corresponding Domain Procedure
Applying Corresponding Domain Procedure
Global Optimization
End of Constraints Validation and Satisfaction

Fig. 7 Flow diagram illustrating automatic geospatial metadata constraints validation and satisfaction

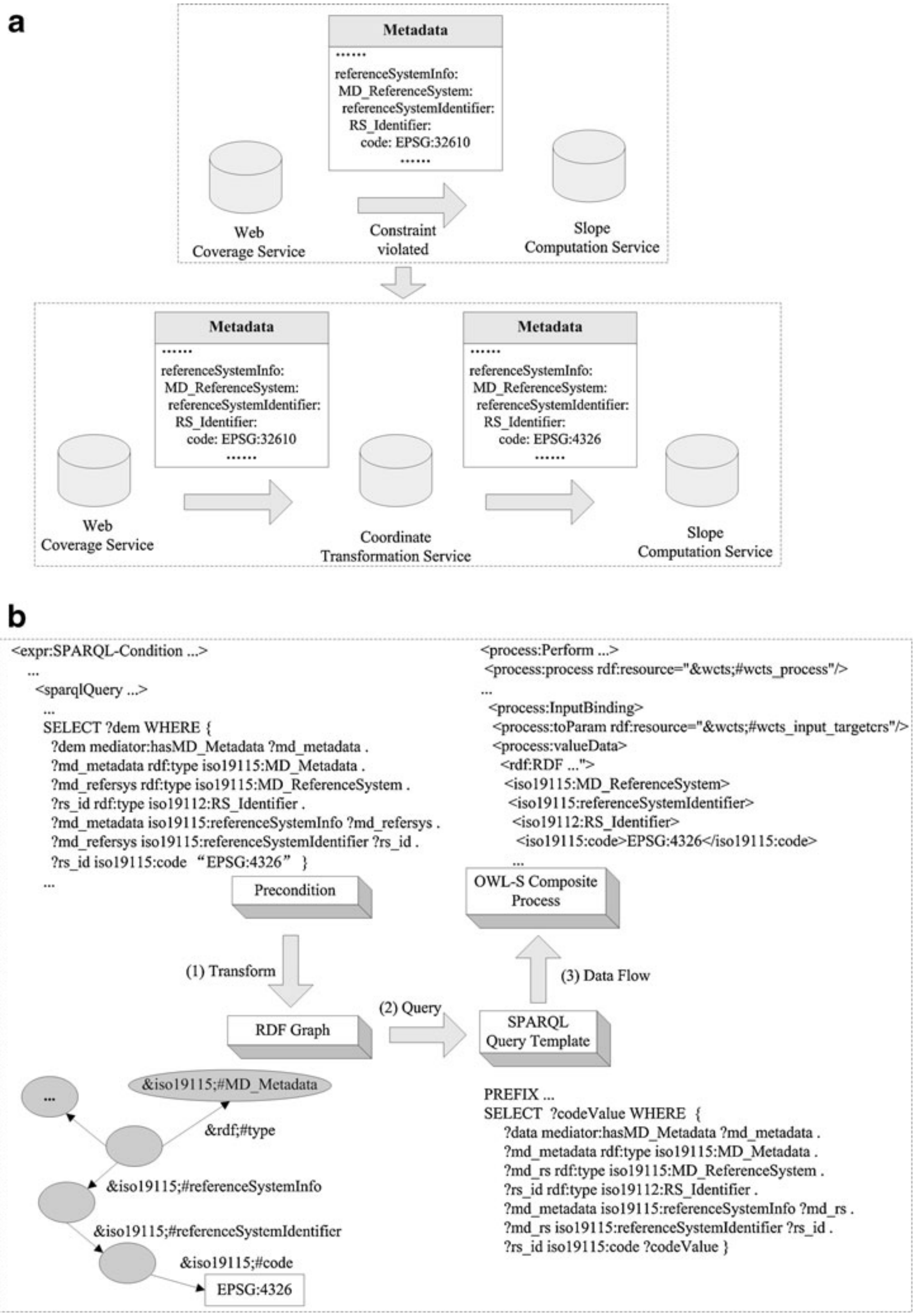
identification of a certain metadata constraint by examining each precondition, which can then help select the corresponding procedure if the specific constraint is violated. For example, two preconditions for the input data of the slope service are defined: one is for the coordinate reference system, and the other is for the file format.

Since the definition of preconditions is based on the ISO 19115 ontology, the common procedure is to identify the constraint and extract the information from the precondition (i.e., template queries can be defined to check preconditions). Figure 8a shows an example, where the constraint on projection is not satisfied in the top part of the figure. Figure 8b shows how to extract the projection code from the spatial projection precondition and then transfer it to the target projection parameter of the Coordinate Transformation Service. The procedure includes the following steps:

(1) Transforming a precondition into an RDF graph, which represents a knowledge base consisting of facts;
(2) Defining a template query to extract the parameter value from the knowledge base;
(3) Assigning the data flow in the updated service chain.

Figure 8b shows an RDF graph generated using a precondition in the SPARQL expression. Template queries such as SPARQL queries can be defined to extract the parameter values required in the modified service chain, as represented in the data flow of an OWL-S composite process shown in Fig. 8b. Thus each domain procedure modifies not only the control flow (through service insertion) but also the data flow (through input/output bindings). After applying the domain procedure, the metadata for input data of the slope service is updated as shown in the lower part of Fig. 8a.

The order of the preconditions, in the inner loop of the process shown in the Fig. 7, affects the order of services inserted. Consider Case 1 in Section 2.2. Figure 9 shows the services inserted between a Web coverage service and slope computation service when processing from the spatial projection precondition and file format precondition



```
Metadata
……
referenceSystemInfo:
 MD_ReferenceSystem:
  referenceSystemIdentifier:
   RS_Identifier:
     code: EPSG:32610
       ……
```

```
Metadata
……
referenceSystemInfo:
 MD_ReferenceSystem:
  referenceSystemIdentifier:
   RS_Identifier:
     code: EPSG:32610
       ……
```

```
Metadata
……
referenceSystemInfo:
 MD_ReferenceSystem:
  referenceSystemIdentifier:
   RS_Identifier:
     code: EPSG:4326
       ……
```

```
<expr:SPARQL-Condition ...>
  ...
   <sparqlQuery ...>
    ...
    SELECT ?dem WHERE {
     ?dem mediator:hasMD_Metadata ?md_metadata .
     ?md_metadata rdf:type iso19115:MD_Metadata .
     ?md_refersys rdf:type iso19115:MD_ReferenceSystem .
     ?rs_id rdf:type iso19112:RS_Identifier .
     ?md_metadata iso19115:referenceSystemInfo ?md_refersys .
     ?md_refersys iso19115:referenceSystemIdentifier ?rs_id .
     ?rs_id iso19115:code "EPSG:4326" }
    ...
```

```
<process:Perform ...>
 <process:process rdf:resource="&wcts;#wcts_process"/>
...
 <process:InputBinding>
  <process:toParam rdf:resource="&wcts;#wcts_input_targetcrs"/>
  <process:valueData>
   <rdf:RDF ...">
    <iso19115:MD_ReferenceSystem>
     <iso19115:referenceSystemIdentifier>
      <iso19112:RS_Identifier>
       <iso19115:code>EPSG:4326</iso19115:code>
        ...
```

```
PREFIX ...
SELECT ?codeValue WHERE {
    ?data mediator:hasMD_Metadata ?md_metadata .
    ?md_metadata rdf:type iso19115:MD_Metadata .
    ?md_rs rdf:type iso19115:MD_ReferenceSystem .
    ?rs_id rdf:type iso19112:RS_Identifier .
    ?md_metadata iso19115:referenceSystemInfo ?md_rs .
    ?md_rs iso19115:referenceSystemIdentifier ?rs_id .
    ?rs_id iso19115:code ?codeValue }
```

Fig. 8 Data flow in domain procedures: **a** metadata update after applying the domain procedure; and **b** steps included in the domain procedure

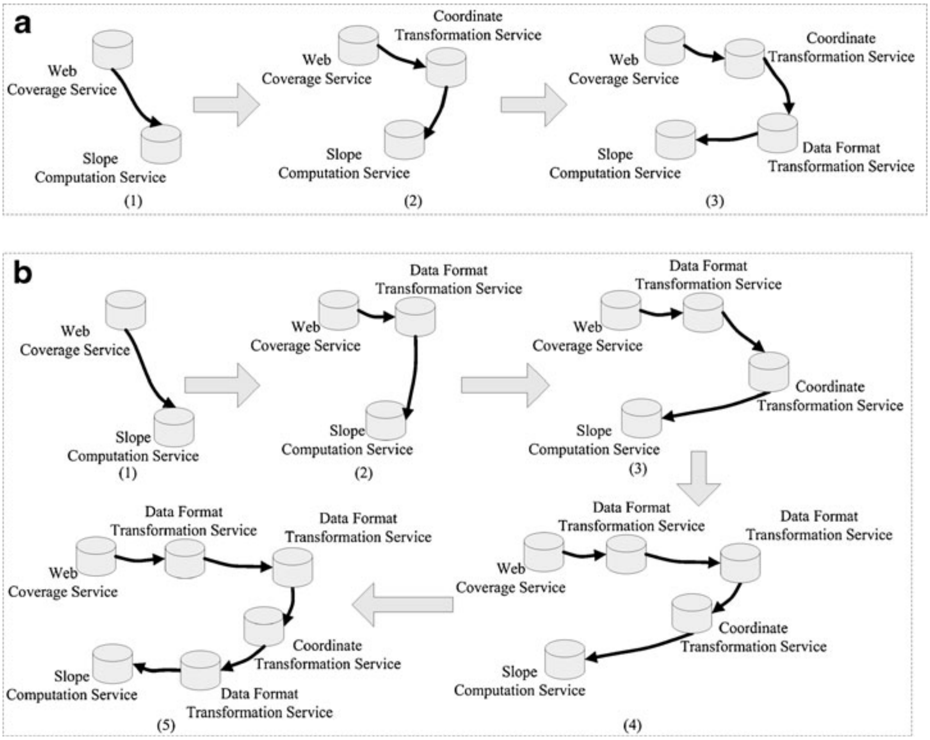

a
Web Coverage Service
Slope Computation Service
(1)
Coordinate Transformation Service
Web Coverage Service
Slope Computation Service
(2)
Web Coverage Service
Coordinate Transformation Service
Data Format Transformation Service
Slope Computation Service
(3)
b
Web Coverage Service
Slope Computation Service
(1)
Data Format Transformation Service
Web Coverage Service
Slope Computation Service
(2)
Data Format Transformation Service
Web Coverage Service
Coordinate Transformation Service
Slope Computation Service
(3)
Data Format Transformation Service
Data Format Transformation Service
Web Coverage Service
Coordinate Transformation Service
Slope Computation Service
(4)
Data Format Transformation Service
Data Format Transformation Service
Web Coverage Service
Coordinate Transformation Service
Slope Computation Service
Data Format Transformation Service
(5)

Fig. 9 Precondition satisfaction for multiple preconditions: **a** starting from spatial projection precondition; and **b** starting from file format precondition

respectively. In this use case, the coordinate transformation service can process the data only in the HDF-EOS file format. The indexes in Fig. 9 show the processing flow step by step during the employment of domain procedures. The first chaining result of the two (Fig. 9a and b) is preferred since it has fewer computational steps. Therefore, the inner loop can start from different preconditions and adopt the shortest path method, favoring conclusions resulting from shorter paths (i.e. fewer services) of the service chain. In addition, when getting the sorted preconditions before the inner loop, some preferences can be imposed on the domain procedures. For example, the spatial projection precondition has a higher priority than the resolution precondition because the regridding operation in the resolution conversion service must be executed in the correct coordinate reference system to meet the requirements of a fire prediction service.

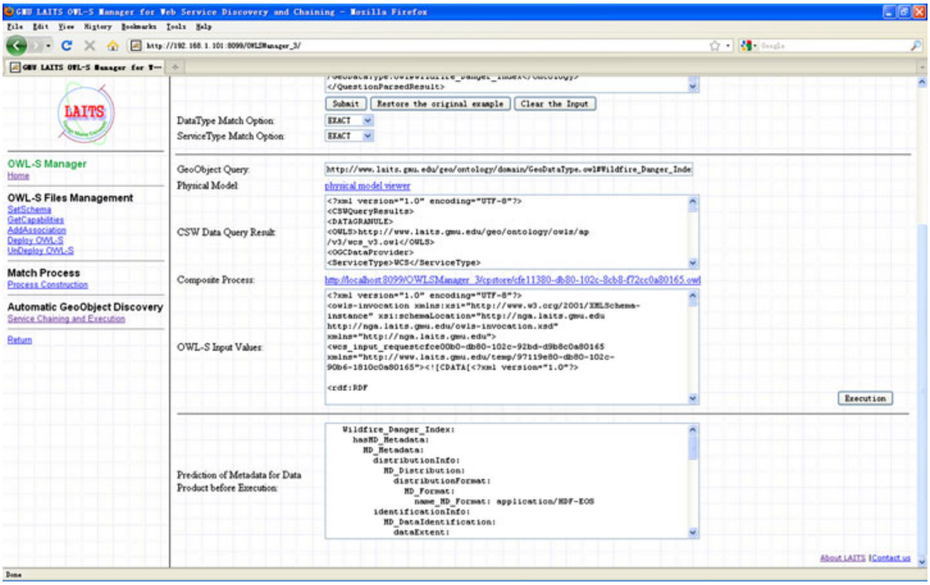
The service chain in Fig. 9b can be optimized to improve the execution efficiency. When two services transforming between data formats are joined sequentially, they can be replaced by a new service transforming between data formats, with its inputs those of the first service and output that of the second service. With metadata generated for intermediate data products, it is possible to use it as filters to generate queries to a metadata catalogue for data that will prevent those sub-chains from being unnecessarily executed. Therefore, as shown in Fig. 7, when precondition checking and action for all inputs are finished, global optimization can be employed. It consists of two steps: identifying sub-chains whose outputs already exist, to prevent repeated execution through reorganizing the service chain, and decreasing service redundancy by not chaining two services of the same functional type successively.

6 Prototype implementation and result analysis

6.1 Implementation

The prototype uses the OWLSMananger, a system for the management of geospatial OWL-S files that can deploy and undeploy OWL-S files into the knowledge base [22]. A semantics-enhanced CSW is integrated with the OWLSManager to support the generation of a virtual data product through automatic service composition [5]. OWL-S Application Programming Interface (API) [50] is used for parsing and traversing each service in the service chain represented using OWL-S. The API provides an ExecutionEngine that can invoke AtomicProcesses that have WSDL grounding and CompositeProcesses that use control constructs such as Sequence, and Split-Join. It has been extended by the Laboratory for Advanced Information Technology and Standards (LAITS) of George Mason University (GMU) to support the HTTP GET and POST invocations in addition to the SOAP invocation it already has. The most advanced version of OWL-S that OWL-S API supports currently is version 1.1. It has been extended by GMU LAITS to also support some new features in the pre-release version of 1.2, including support of the SPARQL precondition. OWL-S API is implemented on top of the Jena [51], a Java framework for building Semantic Web applications. Jena has provided a programmatic environment for RDF, OWL, and SPARQL, and includes a rule-based inference engine. Both TBOX and ABOX reasoning are supported by reasoners in Jena.

An interface for generating metadata for a virtual data product is integrated with OWLSManager (Fig. 10). The request for a virtual data product uses XML. This XML specifies the temporal/spatial ranges for which the information is requested. The Ontology element of the XML specifies the type of information (geospatial DataType). Through an XSLT transformation, the requested XML can be transformed internally into an ontology entity in Table 1. The middle part of Fig. 10 shows the materialization result of a virtual


GMU LAITS OWL-S Manager for Web Service Discovery and Chaining - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.1.101:8099/OWLSManager_3/
GMU LAITS OWL-S Manager for W...
LAITS
OWL-S Manager
Home
OWL-S Files Management
SetSchema
GetCapabilities
AddAssociation
Deploy OWL-S
UnDeploy OWL-S
Match Process
Process Construction
Automatic GeoObject Discovery
Service Chaining and Execution
Return
Submit
Restore the original example
Clear the Input
Execution
About LAITS
Contact us
Done

```
</QuestionParsedResult>
```

DataType Match Option: EXACT

ServiceType Match Option: EXACT

GeoObject Query: http://www.laits.gmu.edu/geo/ontology/domain/GeoDataType.owl#Wildfire_Danger_Inde

Physical Model: physical model viewer

CSW Data Query Result:

```
<?xml version="1.0" encoding="UTF-8"?>
<CSWQueryResults>
<DATAGRANULE>
<OWLS>http://www.laits.gmu.edu/geo/ontology/owls/ap
/v3/wcs_v3.owl</OWLS>
<OGCDataProvider>
<ServiceType>WCS</ServiceType>
```

Composite Process: http://localhost:8099/OWLSManager_3/cpstore/cfe11380-db80-102c-8cb8-f72cc0a80165.owl

OWL-S Input Values:

```
<?xml version="1.0" encoding="UTF-8"?>
<owls-invocation xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://nga.laits.gmu.edu
http://nga.laits.gmu.edu/owls-invocation.xsd"
xmlns="http://nga.laits.gmu.edu">
<wcs_input_requestcfce00b0-db80-102c-92bd-d9b8c0a80165
xmlns="http://www.laits.gmu.edu/temp/97119e80-db80-102c-
90b6-1810c0a80165"><![CDATA[<?xml version="1.0"?>

<rdf:RDF
```

Prediction of Metadata for Data Product before Execution:

```
  Wildfire_Danger_Index:
    hasMD_Metadata:
      MD_Metadata:
        distributionInfo:
          MD_Distribution:
            distributionFormat:
              MD_Format:
                name_MD_Format: application/HDF-EOS
        identificationInfo:
          MD_DataIdentification:
            dataExtent:
```

Fig. 10 OWLSManager interface, showing the metadata generated

data product including archived input data and an OWL-S-based service chain. The bottom part of the figure shows the resulting metadata for the virtual data product. The metadata tracking profile is configured in XML and loaded at the beginning of metadata generation. Both SWRL and SPARQL preconditions are supported and can be processed by domain procedures for data flow modification. The transformation from SWRL preconditions to an RDF graph uses the existing implementation of OWL-S API, while we added additional extensions to support transformation from SPARQL preconditions to an RDF graph. The domain procedures implemented in the component act as built-in domain rules to control satisfaction of metadata constraints.

Yue et al. [52] have presented functions of a metadata-tracking component to enable the instantiation of process models in this prototype system for geospatial service composition. This component interacts with a semantics-enhanced CSW for semantic metadata generation, propagation, and validation. However, how automatic validation and satisfaction of geospatial metadata constraints works is not specified. For example, approaches on data flows in the domain procedures are not formalized. In addition, the proposal of both unary and *n*-ary metadata propagation functions and specification of core metadata information in this paper provides theoretical base to support metadata propagation, which has not been done before.

To run the slope and wildfire prediction cases in this system, all related Web services have been implemented and OWL-S descriptions have been created for the services. The slope case uses W3C SOAP-based Web services, and an SWRL expression is used to represent preconditions. In the wildfire case, the 52n Web Processing Service (WPS) framework [53] is used to develop all geospatial processing services. WPS is an OGC standard defining interface and protocol for geospatial processing services. It uses the HTTP GET and POST protocol for operation requests. The SPARQL precondition is used in this case. Thus the implementation supports both OGC-compliant and non-OGC-compliant Web services and different language expressions for preconditions.

6.2 Analysis of results

The applicability of the automatic metadata generation approach is demonstrated through its support to both the slope and wildfire prediction examples in Section 2. A virtual data product request is represented using a geospatial DataType, Terrain_Slope, or Wildfire_Danger_Index, along with spatial and temporal constraints. Two preconditions are defined for the input data of the slope computation service, one for the file format and the other for the spatial projection. In the final service chain,[3] the coordinate transformation and data format transformation services are inserted sequentially between WCS and the slope computation service. Eighteen preconditions are defined for the wildfire prediction case. They cover the six input geospatial DataTypes and three different metadata entities: the file format, spatial projection, and grid resolution. Failure to satisfy any precondition leads to the insertion of one data reduction and transformation service, following the domain procedures of the metadata-tracking component. Figure 10 shows how the metadata generated for the data product for wildfire prediction appears to the user.[4] Table 4

[3] This case was demonstrated in the Semantic Web Challenge of the 5th International Semantic Web conference in Athens, GA, USA. The final service chain represented using OWL-S is available at http://www.laits.gmu.edu/geo/nga/demo2/cp.owl.

[4] This case was successfully demonstrated in July 2007 at Summer ESIP Federation meeting in University of Wisconsin, Madison, Wisconsin, USA. The final service chain represented using OWL-S is available at http://www.laits.gmu.edu/geo/ontology/owls/cp/wildfirecase.owl.

Table 4 Metadata generated for the data product for wildfire prediction

```
Wildfire_Danger_Index:
    hasMD_Metadata:
      MD_Metadata:
        distributionInfo:
          MD_Distribution:
            distributionFormat:
              MD_Format:
                name_MD_Format: application/HDF-EOS
        identificationInfo:
          MD_DataIdentification:
            dataExtent:
              EX_Extent:
                geographicElement:
                  EX_BoundingPolygon:
                    polygon:
                      GM_Envelope:
                        upperCorner:
                          DirectPosition:
                            coordinates: -1300330.654000,-38780.983000
                        lowerCorner:
                          DirectPosition:
                            coordinates: -2038330.654000,-1241780.983000
        referenceSystemInfo:
          MD_ReferenceSystem:
            referenceSystemIdentifier:
              RS_Identifier:
                code: AUTO2:42004,1,-100,45
        spatialRepresentationInfo:
          MD_GridSpatialRepresentation:
            axisDimensionProperties:
              MD_Dimension:
                dimensionSize: 1203
                dimensionName: row
            axisDimensionProperties:
              MD_Dimension:
                dimensionSize: 738
                dimensionName: column
```

lists the metadata generated. The metadata generated provides both an informed understanding of the virtual data product and a semantically consistent service chain. The metadata generated before the execution of the service chain can help users to evaluate the fitness of the service chain. Furthermore, when combined with CSW, the metadata can be assigned to the CSW query as query filters for identifying existing data products.

Both the syntactic and semantic issues about service chaining have been identified in the OGC abstract service architecture [10]. Syntactic issues are addressed by existing language standards for service composition such as the Web Services Business Process Execution Language (WSBPEL), known as BPEL for short. Semantic issues are related to the semantic evaluation of the results of a service chain, including whether the input data sets are suitable for the subsequent processing, the effect of individual services on the data, and how does the order of the services in the chain affect the results [10]. Such semantic evaluation depends on understanding the semantic information for both individual services and combinations of these services. From the perspective of this paper, the appropriateness of the starting data can be determined from whether they satisfy the metadata constraints. The effect of services on data can be specified using the execution semantics of a geospatial service (i.e., the metadata statement in the preconditions and effects). The sequence of the services is represented using OWL-S composite processes and can be modified by metadata constraints satisfaction. Therefore, the approaches in this paper contribute to the solutions of semantic issues of geospatial service chain.

So far we have only considered the cases in the Earth science domain, and tested the approach on raster data using two different types of process models. The automatic metadata generation and satisfaction of the input constraints for the Earth science processing and modeling are linked to the complex nature of Earth science data, which are highly multidisciplinary, heterogeneous, and distributed. For example, the data obtained from NASA or NOAA data centers are often incompatible in terms of the temporal and spatial coverage, resolution, format, and map projections. Experimenting against these characteristics of Earth science data, although relatively simple, can illustrate the rationale of the approach, and the concepts on automatic metadata generation and semantic consistency of service chains are not trivial. In the geospatial domain, different data models of input data exist (e.g., vector, raster, tabular). When more types of geoprocessing services and diverse data sources are involved, their interaction and effects on the metadata propagation are context sensitive. The goal here is to propose such vehicles as preconditions and effects to allow metadata change to be explicitly specified. The actual metadata to be tracked and metadata change after geoprocessing by various services should be specified case by case.

7 Conclusions and future work

This paper presents an approach for automatically generating metadata for Earth science virtual data products. The approach is built on previous work on Semantic Web Service and automatic geospatial service composition. Global constraints and local constraints play different roles in metadata generation. The approach uses OGC CSW for source metadata generation, OWL-S processes for metadata propagation, and service execution semantics for validating whether metadata constraints are satisfied. It uses the metadata generation results to optimize service chains. Both unary and *n*-ary metadata propagation functions are defined. A set of core metadata information from ISO 19115:2003 is to be tracked. If a precondition check fails, data reduction and transformation services are chained automatically. A detailed working flow for metadata generation, validation, and satisfaction of constraints is illustrated. The implementation incorporated into OWLSManager can be used to generate semantic metadata automatically for virtual data products. Two case studies of Earth science applications illustrate the applicability of the approach.

The approach demonstrates that semantic metadata can be used to validate the data prerequisites of individual geoprocessing services and create a semantically consistent service chain. Typical semantic issues of service chaining such as appropriateness of starting data, affect of services on data, and sequence of the services, as stated in OGC abstract service architecture, can be addressed in this approach. The metadata generated provides not only basic information to characterize the data products but also a context for end users to interpret and evaluate the data products delivered by service chains.

Future work includes the application of the approach in various use cases involving more types of geoprocessing services and generation of more complex metadata. Such an application should provide primarily semantic descriptions of all involved services. For example, over 100 geoprocessing services have been developed in GeoPW [54]. How these services determine the appropriateness of starting data or affect the input data should be described semantically, explicitly, and formally. And we believe that specifying such metadata requirements and metadata changes by an appropriate semantic description of geoprocessing services is important to allowing automatic service chaining and semantic evaluation of service chains in the future. The other problem is to extend the application to data products in other data models such as feature or network models. Current work is limited to raster data. The applicability of the approach should be tested in the future when a different model comes or various models exist simultaneous.

Another goal is to extend the existing approach on constraints satisfaction and include the planner based approach. The precondition checking for constraints satisfaction relies on queries on knowledge base. For example, SPARQL queries can be executed on OWL knowledge base. The inference is based on the OWL semantics. Adding extensions to the SPARQL query language to support spatial semantics has been proposed [55]. Incorporating these extensions in precondition checking is a worthwhile technique to support spatial reasoning for discovering implicit information that can validate metadata constraints. In addition, the insertion of data reduction and transformation services for satisfying metadata constraints is implemented as domain control procedures by providing functions in the computer source code. These domain control procedures embody domain knowledge, which can be expressed using some knowledge representation mechanism to facilitate AI planning. There are some planner-based works focusing on automating the processing of remote sensing images [56, 57]. Therefore, it is possible to transform the problem of satisfying metadata constraints to an AI planning domain so that traditional AI planners can be used with OWL-S Web service descriptions to generate a plan for service chains [38].

We will also improve the existing prototypical implementation by providing user-friendly tools. The semantic descriptions in this paper are developed manually, which is labor intensive and error prone. In order for Semantic Web Service technologies to be widely adopted, the complexity of creating semantic descriptions for services should be reduced. Future work will therefore address how much of complexity of creating these semantic descriptions can be hided from knowledge engineers by developing user-friendly tools. In addition, the current implementation on metadata generation can only provide partial and XML-encoded information to end users. A visual workbench for metadata tracking is necessary for end users with convenient browsing of metadata generated for intermediate and final data products, and a basic understanding of evolution of service chains. When the metadata is linked with service chains and their executions, the workbench can also allow the free navigation among data products and their provenance information. Therefore, future work includes developing such a visual workbench, which can provides new information generated, provide possibilities for provenance navigation, and allow users to edit appropriate process components and re-enact the metadata generation.

Acknowledgments We are grateful to the four anonymous reviewers, and to Dr. Barry Schlesinger for their detailed comments that helped improve the quality of the paper. This work was funded jointly by National Basic Research Program of China (2011CB707105), Project 40801153 and 41023001 supported by NSFC, LIESMARS and SKLSE (Wuhan University) Special Research Fundings.

References

1. Foster I (2005) Service-oriented science. Science 308(5723):814–817
2. Yang C, Raskin R, Goodchild M, Gahegan M (2010) Geospatial cyberinfrastructure: past, present and future. Comput Environ Urban Syst 34(4):264–277
3. Brodaric B, Fox P, McGuinness DL (2007) Geoscience knowledge representation in cyberinfrastructure. Comput Geosci 35(4):697–868
4. Berners-Lee T, Hendler J, Lassila O (2001) The semantic web. Sci Am 284(5):34–43
5. Yue P, Gong J, Di L, He L, Wei Y (2009) Integrating semantic web technologies and geospatial catalog services for geospatial information discovery and processing in cyberinfrastructure. GeoInformatica. doi:10.1007/s10707-009-0096-1
6. Greenberg J, Spurgin K, Crystal A (2006) Functionalities for automatic metadata generation applications: a survey of metadata experts' opinions. Int J Metadata Seman Ontologies 1(1):3–20
7. Albassuny BM (2008) Automatic metadata generation applications: a survey study. Int J Metadata Seman Ontologies 3(4):260–282
8. Lemmens R, Wytzisk A, Rd B, Granell C, Gould M, van Oosterom P (2006) Integrating semantic and syntactic descriptions to chain geographic services. IEEE Internet Comput 10(5):18–28
9. Lutz M (2007) Ontology-based descriptions for semantic discovery and composition of geoprocessing services. GeoInformatica 11(1):1–36
10. Percivall G (ed) (2002) The OpenGIS abstract specification, topic 12: OpenGIS service architecture, Version 4.3, OGC 02-112. Open Geospatial Consortium, Inc., 78pp
11. Rao J, Su X (2004) A survey of automated web service composition methods. In: Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004). San Diego, CA, USA, pp 43–54
12. Srivastava B, Koehler J (2003) Web service composition—current solutions and open problems. In: Proceedings of ICAPS 2003 Workshop on Planning for Web Services. Trento, Italy, pp 28–35
13. Peer J (2005) Web service composition as AI planning—a survey. Technical Report, University of St. Gallen, Switzerland, 63pp
14. Evans J (2003) Web Coverage Service (WCS), Version 1.0.0, OGC 03-065r6. Open Geospatial Consortium, Inc., 67pp
15. ISO/TC 211 (2003) ISO19115:2003, Geographic Information—Metadata
16. Nebert D, Whiteside A, Vretanos P (eds) (2007) OpenGIS® Catalog Services Specification, Version 2.0.2, OGC 07-006r1. Open GIS Consortium Inc. 218 pp
17. Gruber TR (1993) A translation approach to portable ontology specification. Knowl Acquis 5(2):199–220
18. Baader F, Nutt W (2003) Basic description logics. In: Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider P (eds) The description logic handbook. Theory, implementation and applications. Cambridge University Press, Cambridge, pp 47–100
19. Kolas D, Hebeler J, Dean M (2005) Geospatial semantic web: architecture of ontologies. In: Proceedings of the First International Conference on GeoSpatial Semantics (GeoS 2005). Mexico City, Mexico, pp 183–194
20. Dean M, Schreiber G (eds) (2004) OWL Web ontology language reference. World Wide Web Consortium (W3C). http://www.w3.org/TR/owl-ref. Accessed 19 November 2009
21. Klyne G, Carroll JJ (eds) (2004) Resource Description Framework (RDF): concepts and abstract syntax. World Wide Web Consortium (W3C), http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/. Accessed 19 November 2009
22. Yue P, Di L, Yang W, Yu G, Zhao P (2007) Semantics-based automatic composition of geospatial web services chains. Comput Geosci 33(5):649–665
23. Martin D, Burstein M, Hobbs J, Lassila O, McDermott D, McIlraith S, Narayanan S, Paolucci M, Parsia B, Payne T, Sirin E, Srinivasan N, Sycara K (2004) OWL-based web service ontology (OWL-S). http://www.daml.org/services/owl-s/1.1/overview/. Accessed 26 November 2009
24. Christensen E, Curbera F, Meredith G, Weerawarana S (2001) Web Services Description Language (WSDL) 1.1. World Wide Web Consortium (W3C), http://www.w3.org/TR/wsdl. Accessed 23 June 2006
25. Cardoso J, Sheth A (2005) Introduction to semantic web services and web process composition. In: Cardoso J, Sheth A (eds) Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004). Lecture notes in computer science, vol 3387. Springer, Berlin, p 14
26. Clark J (1999) XSL Transformations (XSLT). World Wide Web Consortium (W3C), http://www.w3.org/TR/xslt. Accessed 6 August 2006
27. Bishr Y (1998) Overcoming the semantic and other barriers to GIS interoperability. Int J Geogr Inf Sci 12(4):299–314

28. Sheth A (1999) Changing focus on interoperability in information systems: from system, syntax, structure to semantics. In: Goodchild MF, Egenhofer M, Fegeas R, Kottman CA (eds) The Interoperating Geographic Information Systems. Kluwer, New York, pp 5–30
29. Kuhn W (2005) Geospatial semantics: why, of what, and how? J Data Seman III LNCS 3534:1–24
30. Batcheller J (2008) Automating geospatial metadata generation—an integrated data management and documentation al approach. Comput Geosci 34(4):387–398
31. Mohammadi H, Rajabifard A, Williamson IP (2010) Development of an interoperable tool to facilitate spatial data integration in the context of SDI. Int J Geogr Inf Sci 24(4):487–505
32. Friis-Christensen A, Ostlander N, Lutz M, Bernard L (2007) Designing service architectures for distributed geoprocessing: challenges and future directions. Trans GIS 11(6):799–818
33. Foster I, Kesselman C, Tuecke S (2001) The anatomy of the grid: enabling scalable virtual organizations. Int J Supercomput Appl 15(3):200–222
34. Gahegan M, Luo J, Weaver SD, Pike W, Banchuen T (2009) Connecting GEON: making sense of the myriad resources, researchers and concepts that comprise a geoscience cyberinfrastructure. Comput Geosci 35(4):836–854
35. Vaquero LM, Rodero-Merino L, Caceres J, Lindner M (2009) A break in the clouds: towards a cloud definition. ACM SIGCOMM Comput Commun Rev 39(1):50–55
36. De Longueville B (2010) Community-based geoportals: the next generation? Concepts and methods for the geospatial Web 2.0. Comput Environ Urban Syst 34(4):299–308
37. Ponnekanti SR, Fox A (2002) SWORD: a developer toolkit for web service composition. In: Proceedings of the International World Wide Web Conference. Honolulu, Hawaii, USA, May 2002, pp 83–107
38. Sirin E, Parsia B, Wu D, Hendler J, Nau D (2004) HTN planning for web service composition using SHOP2. J Web Semant 1(4):377–396
39. Klusch M, Gerber A, Schmidt M (2005) Semantic web service composition planning with OWLS-Xplan. In: Proceedings of the Agents and the Semantic Web, 2005 AAAI Fall Symposium Series. Arlington, Virginia, USA, November, 2005, 8 pp
40. Zaharia R, Vasiliu L, Hoffman J, Klien E (2009) Semantic execution meets geospatial web services: a pilot application. Trans GIS 12(s1):59–73
41. Foster I, Vockler J, Wilde M, Zhao Y (2002) Chimera: A virtual data system for representing, querying, and automating data derivation. In: Kennedy J (ed) Proceedings of the 14th International Conference on Scientific and Statistical Database Management (SSDBM'02). Edinburgh, Scotland, IEEE Computer Society, pp 37–46
42. Di L (2004) GeoBrain-a web services based geospatial knowledge building system. In: Proceedings of NASA Earth Science Technology Conference 2004. June 22–24, 2004. Palo Alto, CA, USA, 8 pp
43. Zhao J, Goble C, Greenwood M, Wroe C, Stevens R (2003) Annotating, linking and browsing provenance logs for e-Science. In: Proceedings Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data, Sanibel Island, Florida, USA, 6 pp
44. Kim J, Gil Y, Ratnakar V (2006) Semantic metadata generation for large scientific workflows. In: Proceedings of the 5th International Semantic Web Conference. Athens, Georgia, USA, Lecture notes in computer science, vol 4273. Springer, Berlin, pp 357–370
45. Yue P, Gong J, Di L (2010) Augmenting geospatial data provenance through metadata tracking in geospatial service chaining. Comput Geosci 36(3):270–281
46. SWSI (2004) Semantic Web Services Initiative (SWSI). http://www.swsi.org/. Accessed 21 March 2008
47. Drexel (2004) ISO 19115 metadata ontology. Drexel University, USA, http://loki.cae.drexel.edu/~wbs/ontology/. Accessed 17 October 2005
48. Horrocks I, Patel-Schneider PF, Boley H, Tabet S, Grosof B, Dean M (2004) SWRL: a semantic web rule language combining OWL and RuleML. W3C Member Submission, http://www.w3.org/Submission/SWRL/. Accessed 12 March 2007
49. Prud'hommeaux E, Seaborne A (eds) (2006) SPARQL query language for RDF. World Wide Web Consortium (W3C), http://www.w3.org/TR/rdf-sparql-query/. Accessed 21 November 2009
50. OWL-S API (2004) OWL-S API. Maryland Information and Network Dynamics Lab Semantic Web Agents Project (MINDSWAP), http://www.mindswap.org/2004/owl-s/api/. Accessed 19 November 2009
51. Jena (2006) Jena. Hewlett-Packard Labs Semantic Web Programme, http://jena.sourceforge.net. Accessed 19 November 2009
52. Yue P, Di L, Yang W, Yu G, Zhao P, Gong J (2007) Semantics-enabled metadata generation, tracking and validation in the geospatial web service composition for distributed image mining. In: Proceedings 2007 IEEE International Geoscience and Remote Sensing Symposium (IGARSS07). 23 July–27 July 2007, Barcelona, Spain, pp 334–337
53. 52n WPS (2006) 52n Web Processing Service (WPS). https://www.incubator52n.de/twiki/bin/view/Processing/52nWebProcessingService. Accessed 16 October, 2006

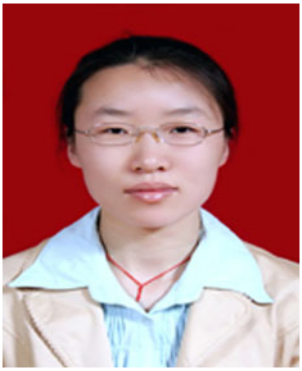
54. Yue P, Gong J, Di L, Yuan J, Sun L, Wang Q (2009) GeoPW: towards the geospatial processing web. In: Proceedings of the 9th International Symposium on Web and Wireless Geographical Information Systems (W2GIS 2009). 7 & 8 December 2009, Maynooth, Ireland, Lecture notes in computer science, vol 5886. Springer, Berlin, pp 25–38
55. Kolas D (2008) Supporting spatial semantics with SPARQL. Trans GIS 12(s1):5–18
56. Golden K (2003) A domain description language for data processing. In: Proceedings of the International Conference on Automated Planning and Scheduling, Workshop on the Future of PDDL. Trento, Italy, June 9–13, 2003, 10pp
57. Chien S, Fisher F, Lo E, Mortensen H, Greeley R (1999) Using artificial intelligence planning to automate science image data analysis. Intell Data Anal 3(3):159–176

Dr. Peng Yue holds a Ph.D. in GIS from the Wuhan University (2007). He is an associate professor at State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing of Wuhan University, China. His research interests include GIS interoperability, Web GIS, and Geospatial Semantic Web. He has been involved in many related research projects, including Choreographed Intelligent Web Services for Automated Geospatial Knowledge Discovery funded by U.S. NGA NURI, GeoBrain project funded by U.S. NASA REASoN program, Metadata Tracking in Geospatial Service Chaining and Geospatial Data Provenance funded by NSF of China, Grid GIS and Semantic Web-based Intelligent Geospatial Web Service funded by Ministry of Science and Technology of China.

Dr. Jianya Gong is the professor and director of the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, China. He studied as a PhD candidate at Wuhan Technical University of Surveying and Mapping and the Technical University of

Denmark from 1988 to 1992 and received his Ph.D. in 1992. His research interests include geospatial data structure and data model, geospatial data integration and management, geographical information system software, geospatial data sharing and interoperability, Photogrammetry, GIS and remote sensing applications.

Dr. Liping Di holds a Ph.D. in Remote Sensing and GIS from the University of Nebraska-Lincoln. He is the professor and director of the Center for Spatial Information Science and Systems (CSISS) (formerly LAITS), George Mason University. His research interests include GIS, remote sensing, interoperability, Semantic Web, global climate and environmental changes.



Lianlian He holds a B.Sc. degree in Informatics and Computational Sciences (2002) and a M.Sc. degree in Computational Mathematics from the Wuhan University (2005). She is currently working as a lecturer in the department of Mathematics, Hubei University of Education, China. Her research interest is on the applications of computational mathematic methods in Geoinformatics and Bioinformatics.