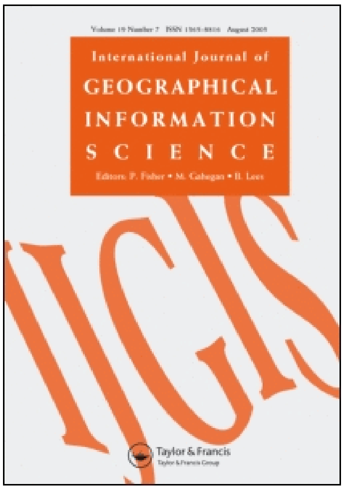

International Journal of Geographical Information Science
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713599799

Semantic Web Services-based process planning for earth science applications

Peng Yue [ab]; Liping Di [a]; Wenli Yang [a]; Genong Yu [a]; Peisheng Zhao [a]; Jianya Gong [b]
[a] Center for Spatial Information Science and Systems (CSISS), George Mason University, Greenbelt, MD 20770, USA [b] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China

First Published:September2009

To cite this Article Yue, Peng, Di, Liping, Yang, Wenli, Yu, Genong, Zhao, Peisheng and Gong, Jianya(2009)'Semantic Web Services-based process planning for earth science applications',International Journal of Geographical Information Science,23:9,1139 — 1163

To link to this Article: DOI: 10.1080/13658810802032680

URL: http://dx.doi.org/10.1080/13658810802032680

International Journal of Geographical Information Science
Vol. 23, No. 9, September 2009, 1139–1163

Taylor & Francis
Taylor & Francis Group

Research Article

Semantic Web Services-based process planning for earth science applications

PENG YUE†‡, LIPING DI*†, WENLI YANG†, GENONG YU†, PEISHENG ZHAO† and JIANYA GONG‡

†Center for Spatial Information Science and Systems (CSISS), George Mason University, 6301 Ivy Lane, Suite 620, Greenbelt, MD 20770, USA

‡State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 129 Luoyu Road, Wuhan 430079, China

(*Received 5 March 2008; in final form 5 March 2008*)

In a Web service-based distributed environment, individual services must be chained together dynamically to solve a complex real world problem. The Semantic Web Service has shown promise for automatic chaining of Web services. This paper addresses semi-automatic geospatial service chaining through Semantic Web Services-based process planning. Process planning includes three phases: process modeling, process model instantiation and workflow execution. Ontologies and Artificial Intelligence (AI) planning methods are employed in process planning to help a user dynamically create an executable workflow for earth science applications. In particular, the approach was implemented in a common data and service environment enabled by interoperable standards from OGC and W3C. A case study of the chaining process for wildfire prediction illustrates the applicability of this approach.

Keywords: Geospatial web service; Service composition; Service chain; Semantics; Ontology; AI planning

1. Introduction

Service-Oriented Architecture (SOA) has shown prospects for providing valuable geospatial data and processing functions for worldwide open use. SOA is 'a way of reorganizing a portfolio of previously siloed software applications and support infrastructure into an inter-connected set of services, each accessible through standard interfaces and messaging protocols' (Papazoglou 2003). With this information architecture, large volumes of data and powerful computing resources are available to all users, thus significantly enhancing their ability to use online/near-line data over the Web and allowing the widespread automation of data analysis and computation. Scientists can use services to contribute their original content or value-added products to the community. This cyber community will evolve and become a collective knowledge base. In fact, the scientific research enabled by the SOA, the so-called service-oriented science (Forster 2005), has been explored across multiple disciplines in different countries, such as the US Department of Energy's Earth System Grid (ESG),[1] US National Science Foundation (NSF) funded GEONGrid[2] and the UK e-science program (Hey and Trefethen 2005). Web service technologies,

*Corresponding author. Email: ldi@gmu.edu

International Journal of Geographical Information Science
ISSN 1365-8816 print/ISSN 1362-3087 online © 2009 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/13658810802032680

a set of technologies for the implementation of SOA, have gained wide application around the information world. A Web service is 'a software system designed to support interoperable machine-to-machine interaction over a network'.[3] It has a standard interface to enable the interoperation of different software systems, so that Web services developed by different organizations can be combined to fulfill users' requests. The interoperable services can be published, discovered, chained and executed through the Web. A number of interoperable services have been available to the geospatial community, most notably the Open Geospatial Consortium (OGC) standards-compliant services, including Web Feature Service (WFS), Web Map Service (WMS), Web Coverage Service (WCS), Sensor Observation Service (SOS), Catalogue Services for Web (CSW) and Web Processing Service (WPS).

In a service-oriented environment, where highly diversified data and versatile processing functions are accessible as services, an 'intelligent' mechanism is required to facilitate information discovery and integration over the network and to automate the assembly of service chains (i.e. service composition) to provide value-added products. To achieve this goal, Web services must be semantically meaningful, as well as syntactically expressive. Semantic descriptions of Web services and semantic interoperability ensure that the right services are invoked to produce the right outcomes, as opposed to syntactical interoperability, which ensures only that services are invoked using the correct form. Semantic Web (Berners-Lee *et al.* 2001) technologies, which give machine-processable meanings to the documents, allow the semantics of data and services to be machine-understandable and thus are being applied to Web services. With the emergence of the Semantic Web, Semantic Web Service has become an area of active research. It is essentially a combination of the Semantic Web and Web service technologies, designed to maximize 'automation and dynamism in all aspects of Web service provision and use, including (but not limited to) discovery, selection, composition, negotiation, invocation, monitoring and recovery'.[4] This paper will address the geospatial service chaining using Semantic Web Services.

Yue *et al.* (2007a) have presented an architecture and approach for automatic service chaining in earth science applications using Semantic Web Services. However, the chaining method is limited to the concept match of input–output between services. This produces some limitations. For example, the OGC Web Coordinate Transformation Service (WCTS) performs a geometrical operation that changes spatial reference coordinate systems without transforming the content or theme of input data. As a result, this service cannot be chained correctly into the service chain based only on the concept match of input–output. Artificial Intelligence (AI) planning is a promising approach (Srivastava and Koehler 2003; Rao 2004; Peer 2005), and it can be incorporated into our system framework. In this paper, we present an extension of original work. We address the use of Semantic Web Services-based process planning for semi-automatic geospatial service chaining. A process can be either an atomic process, which is a description of the behavior of one service type, or a composite process, which is a composition of atomic processes. Process planning consists of three phases: process modeling, process model instantiation and workflow execution.[5] This paper will show how ontologies and AI planning methods are employed in the process planning to help a user dynamically create an executable workflow for Earth science applications. In particular, the approach was implemented in a common data and service environment enabled by interoperable standards from OGC and World Wide

Web Consortium (W3C). A case study of the chaining process for wildfire prediction illustrates the applicability of the approach.

2. Earth science application in the service-based environment

2.1 *A motivating example*

To illustrate the proposed solution, we use the following example throughout the paper. Assume a disaster manager, John, wants to know: 'What is the possibility of having wildfire(s) within a 300 km radius of Bakersfield, CA tomorrow?' He would go through the following steps to answer the question using the distributed heterogeneous data and various geoprocessing services:

(1) Specify the metadata description of the desired data product: through a service registry, John has access to a service registry/catalog (e.g. CSW) providing descriptions of the available services. There might be several wildfire prediction services available. John knows that earth science applications are always subject to spatial constraints, e.g. a certain wildfire prediction service may be limited to producing wildfire prediction data for a certain place. Thus, John has to first get the bounding box of the area of interest, and then use it as the filter to get a qualified wildfire prediction service. John knows that a Geocoder service, a Coordinate Transformation Service (CTS) (projecting geographic coordinates to buffer processing coordinate system), a geometry buffer service, a CTS (transforming the projected coordinates to geographic coordinates) and a geometry envelope calculation service can be chained to generate the bounding box for the area with which he is concerned. John uses that service chain first to create the bounding box for the area of concern. In addition, John also specifies the projection for the wildfire prediction image data product: a Lambert Azimuth Equal Area projection (LAMAZ), centered at latitude 45° and longitude 100°.[6]

(2) Process modeling: John knows that, to get a wildfire prediction product for the region within 300 km of Bakersfield, he usually requires the output of a buffer process to get the part of the wildfire prediction product of an available, qualified wildfire prediction service that he needs. Thus he must rely on an image cutting service (a service which uses a polygon to cut the image, creating an image containing the values of the desired area only) to create the data product for the area with which he is concerned. Thus, John constructs an abstract process that consists of feeding the output of a buffer process and the wildfire prediction product into an image cutting process.

(3) Create the executable workflow: John now wants to create an executable service chain that can be stored to routinely create the desired wildfire prediction data product. John finds a wildfire prediction service that, given maximum temperature, minimum temperature, precipitation amount, Leaf Area Index (LAI), Fraction of Photosynthetically Active Radiation (FPAR) and land cover/use types (LULC) as input, can generate wildfire prediction data products for California. John searches the catalog (e.g. CSW) to find the input data for the wildfire prediction service, using the next day's date as the temporal filter and a bounding box constraint of this wildfire prediction service as the spatial filter. John finds that the National Oceanic & Atmospheric Administration (NOAA) National Digital Forecast Database[7]

(NDFD) can provide the weather data[8] and National Aeronautics and Space Administration (NASA) Earth Observing System (EOS) Moderate Resolution Imaging Spectroradiometer (MODIS)[9] products can provide the FPAR, LAI and LULC.[10] Tables 1 and 2 show the data and services used.

John needs several general geospatial data processing services to coregister the data sets, the so-called data reduction and transformation services, including data format conversion, coordinate system transformation and resampling/interpolation/regridding. In some cases, these general services may also be available as optional functions in data request services, such as the WCS. In this example, the WCS does not provide these optional functions. The operationally available NASA data in the Land Processes Distributed Active Archive Center (LPDAAC) are stored in HDF-EOS data format and in a sinusoidal grid coordinate reference system at a spatial resolution of 1 km. The MODIS grids are stored as tiles, each covering approximately $1200 \times 1200\,\mathrm{km}^2$. 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 fire prediction service takes input data in HDF-EOS format, with LAMAZ projection and 1 km spatial resolution. Preprocessing is

Table 1. Services used in this example.

Service	Description
Wildfire prediction	OGC WPS process that uses a logistic regression algorithm to provide the computational model for wildfire prediction. It takes into consideration the maximum temperature, minimum temperature, precipitation, Leaf Area Index (LAI), Fraction of Photosynthetically Active Radiation (FPAR) land cover/use types (LULC)
Image cutting service (IMCS)	OGC WPS process that uses a polygon to cut an image. The input polygon follows the Geography Markup Language (GML) schema
Geocoder	OGC WPS process that provides the geographic coordinates for the geographic address. The input and output data follow the OGC OpenGIS Location Service (OpenLS) schema
Buffer	OGC WPS process that performs the spatial operation of buffer. The input and output follow the GML schema
GetEnvelope	OGC WPS process that calculates the bounding box of a feature. The input and output data follow the GML schema
CTS	OGC WPS process that performs the reprojection computation. It can transform the data from one spatial projection to another spatial projection. The input and output data follow the OGC WCTS schema
Data format translation service (DFTS)	OGC WPS process that performs the reformating computation. It can transform the data from one file format to another file format
Resolution conversion service (RCS)	OGC WPS process that performs the operations of resampling/interpolation/regridding
OGC CSW	OGC Web-based geospatial catalog service for publication, discovery and access of geospatial data and services
OGC WCS	Provides the available geospatial data (MODIS and NDFD) in the data archives

Table 2. Data used in this example.

Data	Description
FPAR	MODIS/Aqua Fraction of Photosynthetically Active Radiation data product. Operational NASA EOS data (MYD15A2.4), available from the NASA LPDAAC
LAI	MODIS/Aqua Leaf Area Index data product. Operational NASA EOS data (MYD15A2.4), available from the NASA LPDAAC
LULC	MODIS/Terra Land Cover Type data product. Operational NASA EOS data (MOD12Q1.4), available from the NASA LPDAAC
Maximum temperature	NOAA NDFD maximum temperature element
Minimum temperature	NOAA NDFD minimum temperature element
Precipitation amount	NOAA NDFD precipitation amount element

needed to transform the NASA and NDFD data into the form that can be readily accepted by the service.

2.2 *Knowledge representation in supporting service-based earth science applications*

Several parts of the workflow described in Section 2.1 can be supported by advanced knowledge representation technologies. This support can be of great value to scientific users and contribute to the evolution of scientific research in a service-based environment.

2.2.1 Capturing knowledge. As demonstrated by Lutz and Klien (2006) and Yue *et al.* (2007a), semantic concept annotation for geospatial data and services can help the computer automatically combine the data and services while creating a service chain. This case is the same: given the semantic descriptions in the wildfire prediction service, the computer can automatically locate the input data with the corresponding semantic concepts. In addition to the semantic description of data and services, the process model can be captured and represented as another kind of domain knowledge. For example, given the geographic address and the width of a buffer, a process model that creates the buffer, called DistanceBuffer, can be formulated by the composition of the Geocoder process, CTS and Buffer process. The process model not only provides how subprocesses are composed, but also contains knowledge about how to implement an abstract process model, e.g. the DistanceBuffer process model represents a concrete implementation of an abstract buffer process. Additionally, the rules for using the data reduction and transformation services to derive user products are usually simple and commonly accepted in geospatial domain. For example, if the available data's spatial projection is different from the requested data's spatial projection, a CTS can be introduced to finish this reprojection process. When enough metadata information is tracked, such services can be dynamically chained, using domain rules.

2.2.2 Reasoning. Given the information presented above, the following types of reasoning can be identified:

(1) Semantic match. With a semantic knowledge base, data and services discovery will be more accurate and efficient as compared to keyword matching because semantic relationships can be used in the discovery process.

(2) Process decomposition. Some users may have only some high-level abstract process models. The process model, as the composite process, is a structured set of subprocesses that may be further decomposed into existing process models. Thus, process decomposition is to transform the process model to a set of atomic processes that can facilitate the generation of an executable service chain.
(3) The reasoning for employing rules. Different data reduction and transformation services can be inserted into the service chain according to the rules employed.

2.2.3 Grounding. Creating an executable service chain requires mappings from the process model to the concrete specification of the service description (i.e. syntactical description) required for invoking the service. In this case, the mapping for geospatial Web services will consider multiple service specifications, including WCS, WPS, WCTS, OpenGIS Location Service (OpenLS) and Geography Markup Language (GML).

With the knowledge capturing, reasoning and grounding process implemented, general users do not need to know concrete steps of deriving knowledge from data and services. For example, if a wildfire prediction process model and a DistanceBuffer process model are constructed previously, John needs only to specify an abstract model requiring an Image Cutting Service (IMCS) process. The input data consist of the wildfire prediction data product and buffer polygon, which is the output of another process taking at least the geographic address as one input. Then it is possible to create an executable service chain to provide the data product with specific metadata descriptions.

3. Background

In this section, we introduce some concepts related to our approach.

3.1 *Common data and service environment*

As identified by Di (2005), a framework for intelligent geospatial knowledge systems requires interoperability of both geospatial data and services for a system to be able to pull out and chain data and services from providers to complete user requests for geospatial information and knowledge. To facilitate interoperability, two standards-based interoperability environments are needed: the common data environment and the common service environment.

The *common data environment* is 'a set of standard interfaces for finding and accessing data in data archives of varied sizes and sources. This environment allows geospatial services and value-added applications to access diverse data provided by different providers in a standard way without knowing their internal handling of data' (Di 2005). Currently, the most notable interface standards for the common data environment are the OGC Web Data Services Specifications, including WCS, WFS, WMS, and CSW. OGC also provides a data-encoding standard, GML, which is well developed to describe geometries and geographical relations.

The *common service environment* is a set of standard interfaces for service declaration, description, discovery, binding, chaining and execution (Di 2005). This environment allows geospatial knowledge systems dynamically to generate user-specific geospatial information/knowledge by discovering and chaining standards-compliant services supplied by service providers. The requirements for this set of standards in a geospatial knowledge system are very similar to the requirements in mainstream Web services technology. Therefore, the standards used in the mainstream Web service arena can be adopted for geospatial knowledge systems. In this paper, we rely on the Web Services Description Language (WSDL)[11] for the concrete specification of all geospatial Web service descriptions. In particular, we include WPS, a forthcoming OGC specification. A WPS can provide 'any sort of GIS functionality to clients across a network, including access to pre-programmed calculations and/or computation models that operate on spatially referenced data' (Schut and Whiteside 2005). The three mandatory operations included in the WPS interface are GetCapabilities, DescribeProcess and Execute. For more details, please refer to Schut and Whiteside (2005).

The degree of interoperability of geospatial data and services in a Web-based environment determines the levels of effort towards the implementation of automation. Standards-based interoperable geospatial Web service technologies (e.g. OGC interoperability standards) have already demonstrated their capability to enhance access and delivery of heterogeneous geospatial information, although major efforts focus on the syntactical interoperability. There is a need to explore how to align existing standards-based geospatial Web services with Semantic Web Services technologies to allow automatic discovery and chaining of geospatial Web services. We should build our work on existing efforts towards the interoperability of Web service, and a common data and service environment would greatly lower complexity of problems caused by the heterogeneity of geospatial data and services.

3.2 *Ontology approach*

To provide the semantic concepts, we can use the ontologies. An ontology is 'a formal, explicit specification of a conceptualization' (Gruber 1993) that provides a common vocabulary for a knowledge domain and defines the meaning of the terms and the relations between them. The Web Ontology Language (OWL)[12], recommended by W3C as a standard Web ontology language, is designed to enable the creation of ontologies and the instantiation of these ontologies in the description of Web resources. OWL is an extension of the Resource Description Framework (RDF)[13], which defines a flexible approach to representing data based on a graph model composed of triples. The foundation of knowledge representation formalism for OWL is the description logic (DL) (Baader and Nutt 2003). DL is more like an object-oriented approach to knowledge representation. The basic elements of description logics 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 expressive power of different DL languages is subject to the set of constructors and axioms in that language (Baader and Nutt 2003). Generally, the particular selection of constructors and axioms is made so that inference procedure is decidable (Volz 2004). Constructors are a set of symbols formalized for the definition of concepts and roles. There are two types of constructors: concept-forming constructors and role-forming constructors. These constructors can be used to construct complex concepts and roles from atomic concepts and atomic roles.[14]

A DL knowledge base (KB) comprises two components: *TBOX* and *ABOX* (Baader and Nutt 2003). TBOX consists of a set of terminological axioms which make statements about how concepts or roles are related to each other. ABOX introduces individuals, i.e. instances of a class, into the knowledge base and asserts the properties of these individuals (Volz 2004).

There are two types of reasoning in DL: TBOX reasoning and ABOX reasoning. In TBOX reasoning, a basic type of reasoning is to determine whether or not a concept is subsumed by another concept (i.e. subsumption reasoning). In ABOX reasoning, a basic type of reasoning is to determine whether or not a particular individual is an instance of a given concept description. In practice, according to the generality of concepts, the DL knowledge base can be organized hierarchically with 'a special treelike data structure' (Brachman and Levesque 2003a), called a taxonomy. New facts can be added to a taxonomy through an efficient classification process. This taxonomy allows queries to be answered efficiently and thus makes it practical to consider extremely large knowledge bases (Brachman and Levesque 2003a).

In our work, we use both TBOX and ABOX reasoning, and incorporate them into the different phases of process planning.

3.3 *AI planning*

Russel and Norvig (2003) define planning as follows: 'The task of coming up with a sequence of actions that will achieve a goal is called planning'. An important representation of planning problems related to the Web service field is using concepts of the state, goal and action from the classical planning domain. The world or a specified domain is modeled as a set of states that can be divided into initial states and goal states. Goals are partially specified states that can be achieved through actions from the initial states of the world. An action is specified in terms of the preconditions and the effects (post-conditions). The preconditions are the states that must hold before the action can be executed, and the effects are the state changes when the action is executed (Russel and Norvig 2003). Thus, the assumption for Web service composition as a planning problem is that a Web service can be specified as an action (Rao and Su 2004). As a software component, a Web service takes input data and produces output data. Thus, the input and the output parameters can be treated as the preconditions and effects, respectively. Furthermore, the Web service might change the states of the world after its execution. Then, the world states before service execution are the preconditions, and the new states generated after execution are the effects (Rao and Su 2004). If the metadata constraints are the world states, then a service calculating the terrain slope from DEM data may require the HDF-EOS data format as a precondition for DEM data. OWL-S,[15] an OWL-based Web service ontology, provides a formal knowledge representation for the inputs, outputs, preconditions, and effects of Web services. Thus, it is possible that the state of the world is represented in the OWL knowledge base, and the OWL reasoner can be used to reason about the state of the world (Sirin *et al.* 2004). Precondition checking is equivalent to querying the knowledge base, and applying effects is equivalent to adding and deleting facts from the knowledge base (Sirin *et al.* 2004).

The most straightforward planning method in the classical planning domain is state-space search (Russel and Norvig 2003). It includes regressive planning and progressive planning. Regressive planning consists of backward state-space search (i.e. searching from the effects to the preconditions when considering an action), repeatedly simplifying the goal until the goal is achieved in the initial state (Brachman and Levesque 2003b). The first action considered in the planner is the last one in the plan. Progressive planning, instead, is searching forward, i.e. from the preconditions to the effects. Our goal is a user-specified data product with metadata descriptions. Thus, the system can continually search regressively, using service input–output concept matching, until all input data are available for the service chain. Although this approach is straightforward, it does not take into consideration the service functionality in service chaining. Thus it is suitable only for those geospatial Web services where the semantic concept of a service output can embody the functionality of the service, e.g. a service outputting terrain slope data can be identified as a terrain slope calculation service in most situations. For those services whose functionalities are not conveyed by their input/output parameters, this approach might cause much uncertainty.

Apart from constructing the description of an action in terms of its preconditions and effects only in the classical planning domain, Hierarchical Task Network (HTN) planning focuses on the process of making an action concrete, i.e. action decomposition. 'Plans are refined by applying action decompositions. Each action decomposition reduces a high-level action to a partially ordered set of lower-level actions. Action decompositions, therefore, embody knowledge about how to implement actions' (Russel and Norvig 2003). A plan library could contain several decompositions for a high-level action, and lower-level actions could have additional preconditions and effects beyond those in the high-level actions. Each decomposition should be a correct plan. A key issue in HTN planning is detecting interactions and resolving conflicts, since the decomposition will expose the hidden information of lower-actions (i.e. preconditions and effects) which might direct to an incorrect plan. The key advantage of HTN is that, at each level of the hierarchy, an action is reduced to a small number of actions at the next lower level, so that the computational cost of finding the correct way to arrange those actions for the current problem is small (Russel and Norvig 2003). Thus, HTN reduces the complexity of reasoning by removing a great deal of uncertainty about the world.

Traditional planners cannot handle the large amount of data over the semantic Web. And many traditional AI planning methods use a closed-world assumption (i.e. unknown states are assumed to be false) when representing the state of the world (Russel and Norvig 2003). In contrast, Semantic Web and particularly OWL, have an open-world assumption (Sirin *et al.* 2004). And as mentioned in Section 3.2, the taxonomy in the DL knowledge base makes it practical to consider extremely large knowledge bases. Sirin *et al.* (2004) conducted an initial experiment to incorporate the OWL reasoner into SHOP2, a domain-independent HTN planning system whose theorem prover makes a closed-world assumption. The state of the world is represented in the OWL knowledge base, and the OWL reasoner can be used to reason about the state of the world. Although some open issues exist as identified by Sirin *et al.* (2004), it is still a promising way. In our work, we also use the OWL knowledge base to represent the state of the world.

In Section 5, we will show how our work borrows ideas from regressive planning and action decomposition and incorporates them into the different phases of process planning.

4. OWL-S perspective for geospatial Web service

Current research on Semantic Web Services technologies provides the choice of OWL-S, Web Service Modeling Ontology[16] (WSMO), Web Service Semantics[17] (WSDL-S), Semantic Annotations for WSDL[18] (SAWSDL) and Semantic Web Services Framework[19] (SWSF). WSMO and SWSF do not limit their knowledge representation to description logic. Thus their definitions are not built upon OWL in the way OWL-S is. Apart from defining an ontology framework for Web services, WSDL-S and SAWSDL aim to extend existing WSDL elements with semantic annotations; thus, they are not defining a complete ontology framework for Web services. Most previous work uses OWL-S, and many tools are available. This work also uses OWL-S as the vehicle for semantic representation of geospatial Web service.

OWL-S is structured in three main parts:

(1) Service profile: what a service does (advertisement).
(2) Service model: how a service works (detailed description), e.g. a series of necessary input parameters identified in the service model.
(3) Service grounding: how to assess a service (execution), e.g. grounding the input/output ontology concepts to the output message of WSDL operation using Extensible Stylesheet Language Transformations (XSLT).

We have defined the geospatial 'DataType' and 'ServiceType' ontologies to address the data and functional semantics of services respectively. Data semantics annotate the semantics of input and output data in a Web service operation. Functional semantics represent the semantics for a service function. The 'DataType' ontology, in the service grounding, can be used to define a set of bidirectional mappings between the schemas of the OGC-compliant services and the ontologies. However, Yue *et al.* (2007a) did not consider the execution semantics of services. The execution semantics specify the requirements of a service such as the preconditions and effects (Sheth 2003). When a thematic concept match (TBOX reasoning) based on the 'DataType' ontology and 'ServiceType' ontology is available, a geospatial service might still have multiple metadata constraint requirements such as file format, data projection on the input data. We propose to define these metadata constraints in the OWL-S preconditions. Before execution, a precondition check is required for the available data instances. Thus, precondition checking is in fact an ABOX reasoning problem. As is stated in Section 3.3, precondition checking is equivalent to querying the knowledge base. This motivates the use of SPARQL,[20] a promising query language for RDF. And it is also supported in the forthcoming version 1.2 of OWL-S.[21] Table 3 is an example of OWL-S precondition on the file format of input data.

The present work also extends previous work by enriching the 'DataType' ontology with the ISO 19115 ontology and GML ontology.[22] As illustrated in the Figure 1, additional properties are defined, including 'hasMD_Metadata' and 'hasGML', so that each 'DataType' has standards-based semantic metadata and formalized geometry concepts. 'GeoDataType' serves as the top level concept of 'DatatType' ontology. XSLT[23] between GML and the GML ontology can be imported in the service grounding of any OWL-S descriptions for geospatial Web services with GML parameters.

Table 3. An example of OWL-S precondition on the file format of input data.

```
<expr:SPARQL-Condition rdf:ID="supportedFileFormat">
<expr:expressionLanguage rdf:resource="&expr;#SPARQL"/>
<expr:expressionBody rdf:parseType="Literal">
<sparqlQuery xmlns="http://www.w3.org/2002/ws/sawsdl/spec">
  PREFIX iso19115: &lt;http://loki.cae.drexel.edu/~wbs/ontology/2004/09/iso-19115#&gt;
  PREFIX mediator: &lt;http://www.laits.gmu.edu/geo/ontology/domain/v3/mediator_
  v3.owl#&gt;
  PREFIX fileformat: &lt;http://www.laits.gmu.edu/geo/ontology/domain/v2/fileformat.
  owl#&gt;
  PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;
  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_maxt"/>
  </expr:VariableBinding>
</expr:variableBinding>
</expr:SPARQL-Condition>
```

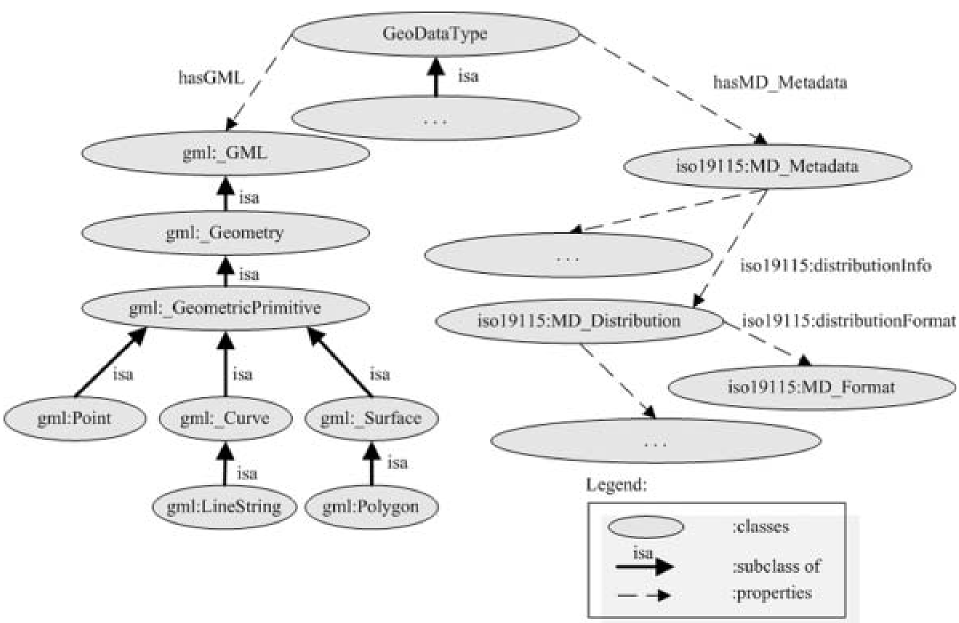

Figure 1. 'DataType' ontology.

Currently, OGC Web services are not equivalent to the W3C SOAP-based Web services. Most OGC Web service implementations provide access via HTTP GET and HTTP POST. They do not support SOAP. Since WSDL can describe the HTTP GET/POST bindings in addition to the SOAP binding, the HTTP GET and POST bindings can still be supported in the service grounding. However, the WSDL grounding in OWL-S is still under development. It cannot well handle the mapping of the multiple OWL-S inputs to a complex WSDL schema type in the message. Following the XML Message handling in Business Process Execution Language for Web Services (BPEL4WS)[24], a de facto standard for the description of service composition, the WSDL grounding is extended by an additional property 'wsdlMessagePartElement' which contains the XPATH to locate the certain element in the complex type. Table 4 shows a snippet of WSDL and service grounding for the WPS buffer process.

5. Process planning

OGC Abstract Service architecture (Percivall 2002) identifies three types of service chaining:

(1) User-defined (transparent) – the human user defines and manages the chain.
(2) Workflow-managed (translucent) – the human user invokes a service that manages and controls the chain. The user is aware of the individual services in the chain.
(3) Aggregate (opaque) – the human user invokes a service that carries out the chain. The user has no awareness of the individual services in the chain.

For most users who have little or no specific domain knowledge, user-defined chaining is inappropriate. Opaque and translucent chaining is much more important to promote wide utilization of geospatial information resources. Through process planning, it is possible to address the translucent and opaque chaining. An overview of Semantic Web Services-based process planning is illustrated in Figure 2. It shows how Semantic Web Services can be used to support the process planning. 'DataType' and 'ServiceType' ontologies are used to address the inputs, outputs and functionalities of semantic description for geospatial Web services. Metadata constraints are specified in the preconditions and effects of Semantic Web Services.

The process planning consists of three phases:

(1) Process modeling, which generates a composite process model consisting of the control flow and data flow among atomic processes.
(2) Process model instantiation, where the composite process is instantiated into a concrete workflow or executable service chain.
(3) Workflow execution, where the chaining result or workflow is executed in the workflow engine to generate the on-demand data product.

We use action decomposition in the process-modeling phase. The preconditions and effects of Semantic Web Services are not considered in the action decomposition. That limits the reasoning in the process-modeling phase to the discovery of a composite process, which is in fact a TBOX reasoning problem. Action decomposition is indeed a process of discovery of a composite process and expanding the abstract process according to the matching result from the discovery.

Table 4. A snippet of WSDL and service grounding for the WPS buffer process.

```
<!--snippet of WPS WSDL-->
<message name=“Execute_POST”><part name=“payload” element=“wps:Execute”/></message>
<message name=“ExecuteResponse”><part name=“payload” element=“wps:ExecuteResponse”/></message>
<portType name=“WPS_HTTP_POST_PortType”>
…
<operation name=“Execute”><input message=“wps:Execute_POST”/>
<output message=“wps:ExecuteResponse”/></operation></portType>

<!--snippet of service grounding-->
<grounding:wsdlInputMessage rdf:datatype=“&xsd;#anyURI”>&wps_buffer_wsdl;#Execute_POST</grounding:wsdlInputMessage>
<grounding:wsdlInput>
<grounding:WsdlInputMessageMap rdf:ID=“wps_buffer_wsdlinputmessagemap_gml”>
<grounding:owlsParameter rdf:resource=“&buffer_profile;#buffer_input_gml”/>
<grounding:wsdlMessagePart rdf:datatype=“&xsd;#anyURI”>&wps_buffer_wsdl;#payload</grounding:wsdlMessagePart>
<groundingx:wsdlMessagePartElement rdf:datatype=“&xsd;#string”><![CDATA[
<context type=“xpath” xmlns=“http://www.laits.gmu.edu/geo/ontology/domain/groundingx.owl” xmlns:wps=“http://www.opengeospatial.net/wps”
xmlns:xlink=“http://www.w3.org/1999/xlink”
xmlns:wcts=“http://www.opengis.net/wcts”>/wps:Execute/wps:DataInputs/wps:Input[position()=1]/wps:ComplexValue</context>]]></
groundingx:wsdlMessagePartElement>
<grounding:xsltTransformationString><![CDATA[
<xsl:stylesheet version=“2.0” xmlns:xsl=“http://www.w3.org/1999/XSL/Transform” xmlns:rdf=“http://www.w3.org/1999/02/22-rdf-syntax-ns#”
xmlns:mediator=“http://www.laits.gmu.edu/geo/ontology/domain/v3/mediator_v3.owl#” xmlns:iso19115=“http://loki.cae.drexel.edu/~wbs/ontology/2004/
09/iso-19115#”
xmlns:gml-ont=“http://loki.cae.drexel.edu/~wbs/ontology/2004/09/ogc-gml#” xmlns:ows=“http://www.opengeospatial.net/ows” xmlns:xlink=“http://
www.w3.org/1999/xlink”
xmlns:geodatatype=“http://www.laits.gmu.edu/geo/ontology/domain/GeoDataType.owl#” xmlns=“http://www.opengis.net/gml”>
<xsl:import href=“http://www.laits.gmu.edu/geo/ontology/owls/xslt/owl2gmlpacket.xsl”/>
</xsl:stylesheet>
]]></grounding:xsltTransformationString>
</grounding:WsdlInputMessageMap></grounding:wsdlInput>
<grounding:wsdlInput>
```

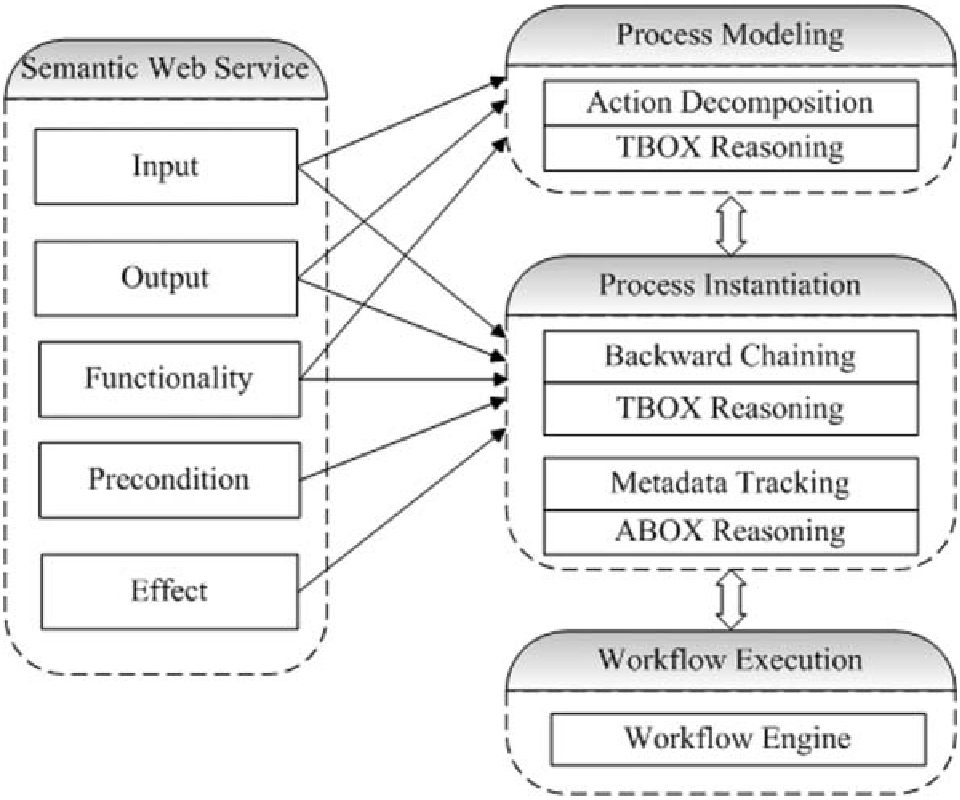

Figure 2. An overview of semantic web service based process planning.

This simplifies the planning in the process-modeling phase and it is not necessary to consider the conflict resolving in the decomposition. Reasoning about the preconditions and effects is left to the instantiation phase for the process model, where the metadata-tracking component can deal with it. Therefore, our action decomposition is simplified and the problem need not be translated to the domain of traditional planning systems. In the process model instantiation phase, a regressive search using data type concept matching, as mentioned in Section 3.3, is used to locate the input data available for the process model.

We show details of the process planning in Section 5.1 through 5.3, using the wildfire prediction use case as an example.

5.1 *Process modeling*

The 'Composite Process' ontology in the OWL-S is used to represent the available process models. A composite process can be characterized as a collection of subprocesses with control and data flow relationships. The control flow specifies the ordering and conditional execution of subprocesses, while the data flow focuses on data exchange among the subprocesses. In OWL-S, the control flow is represented by the control constructs such as Sequence and Split. The data flow is specified through input/output bindings using a class such as ValueOf to state that the input of one subprocess is getting values from the output of a previous subprocess. Using the available composite processes, an abstract process model can be reduced to a

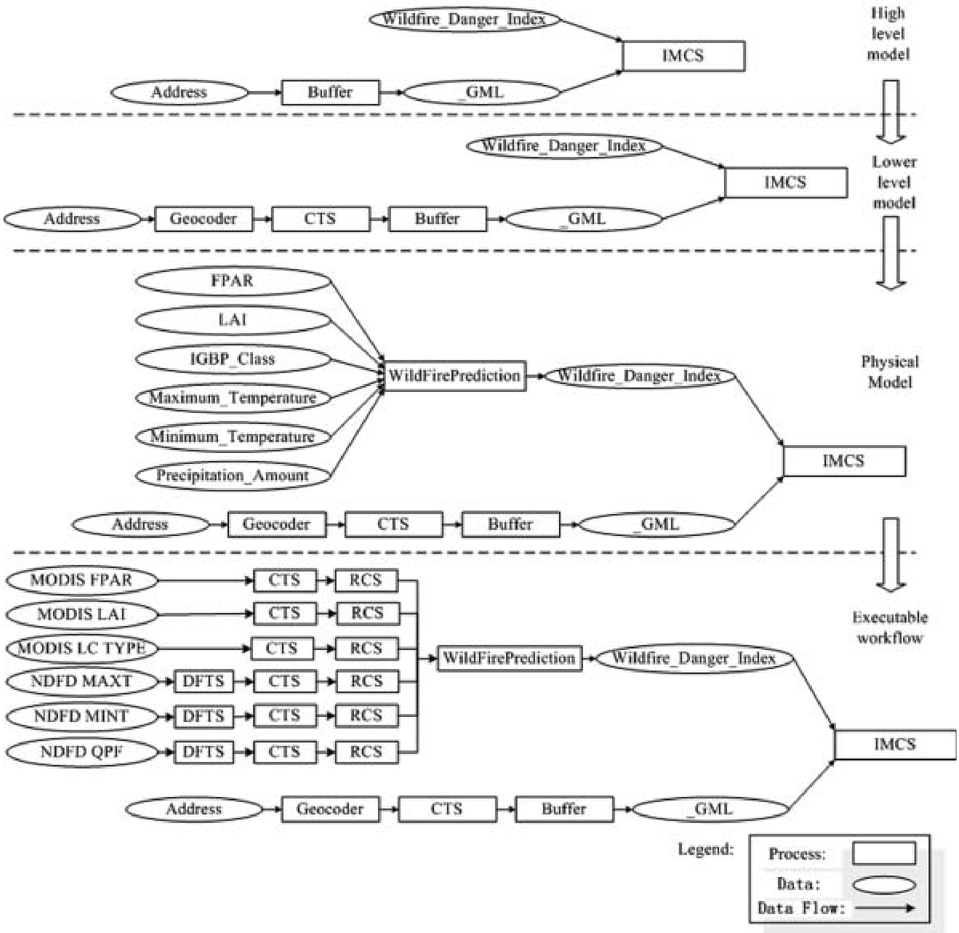

Figure 3. Process planning for the wildfire prediction case.

structured set of subprocesses, which may be further decomposed. The goal is to find a collection of atomic processes for some high-level composite process.

A high-level process model can be built by the user using the 'DataType' and 'ServiceType' ontologies. In Figure 3, John specifies in the high-level process model that a process to create the buffer, with 'Buffer' as the 'ServiceType', needs to take the 'Address' as the input 'DataType'. Action decomposition is used in this phase. The DistanceBuffer composite process as a whole has the matched 'ServiceType' and input 'DataType'. The match is based on the subsumption reasoning. We define process templates based on the data and functional semantics of services. A process template is defined as a tuple (F, I, O), where F is the semantic concept addressing the function of the process, I is a finite set of input semantic concepts and O is a finite set of output semantic concepts. The match process can be divided into two phases. The first is based on the concept match of functionality. It can reduce a large number of processes to a small set containing matched processes. The second phase finds the match of the input/output based on the result set of the first phase. Based on the matching result, the data flow and control flow in the high-level process model can be expanded to include the concrete processes. An example of a lower-level process model is generated in Figure 3.

5.2 *Process model instantiation*

Instantiation creates an executable service chain by binding the service instances and available data to the result of the process-modeling phase. It consists of two steps: generating the physical model and creating the executable workflow.

5.2.1 Physical model generation. The 'WildFire_Danger_Index' data in the result from the process-modeling phase must be bound with the available data. The available data may be either readily obtainable from some data provider or generated at run-time through a service chain. A geospatial catalogue provides the information of data availability. In addition to the 'DataType' constraint, more filtering requirements, such as spatial and temporal extents, are added to the query on the catalogue. If the requested data cannot be found, a process can be selected to produce the requested data. Then the data query is moved on to the input 'DataType' of the selected process. If the selected process has the spatial constraints (e.g. wildfire prediction service), the spatial extent of the query for the input data of the selected process should be adjusted correspondingly. The process continues regressively until all input data available for the service chain are found. We call it 'DataType' driven backward chaining (Yue *et al.* 2007a). It is based on the subsumption reasoning of input–output between services or data-input between data and services. However, this does not exclude the possibility of needing human intervention in complex applications where service functionality needs to be considered. Therefore, in Figure 2, we point out that service functionality could be considered in the process model instantiation phase. The resultant chain is called the 'Physical Model' (e.g. Figure 3). Correspondingly, the model in the process-modeling phase is called the 'logical model'. In relatively simple cases such as creating a service chain to generate a terrain slope data product (first a WCS providing the DEM data and then a slope calculation service generating the slope data product), 'DataType' driven backward chaining is enough to derive an intended data product automatically without human intervention, and the chaining can be characterized a method to enable opaque chaining.

5.2.2 Creating the executable workflow. In a physical model, the input data are available and subprocesses are atomic processes. However, in earth science applications, many processing services have metadata constraints on the input data, such as spatial resolution, file format and coordinate reference system. As mentioned in Section 2.2, the data reduction and transformation services can modify the data to satisfy the metadata constraints. We use a metadata tracking component to automatically insert such services whenever the corresponding metadata constraints are not satisfied. The domain knowledge needed to determine the data reduction and transformation services is implemented as built-in rules. The local constraints and global constraints in the service chain are identified using the ISO 19115 ontology (Yue *et al.* 2007b). Global constraints are those metadata constraints that are applicable to all input and output data of atomic processes, e.g. the place and date of interest (i.e. spatial and temporal constraints). They are put forward in the users' goal (i.e. the metadata descriptions in a user-specified data product). The local constraints identify those metadata constraints that the input data of individual service must follow, e.g. file format. Local constraints might also contain some metadata constraints that are the same as the global constraints, such as the spatial constraints of the wildfire prediction service, yet the global constraints usually have values spanning a larger range (e.g. larger area than that of the users' goal). In this situation, the values in the global constraints can be modified by the corresponding local constraints to apply those sub-chains located just before the service that hosts the local constraints. A metadata-tracking component that can automatically perform the following functions is implemented (Yue *et al.* 2007b):

- Semantic metadata generation. With the availability of CSW, the input data needed by the service chain and resided physically in a data archive can be queried using the global constraints, and their detailed metadata information can be obtained, either directly from the catalog or through extracting from the data files (e.g. from self-describing files such as HDF-EOS and GeoTIFF file).
- Metadata validation. When all the metadata of the input data are ready, the local constraints on the dependent service can be validated with the generated metadata, to see whether the OWL-S precondition is satisfied or not.
- Metadata satisfaction. The data reduction and transformation services are inserted to modify the data so that preconditions are satisfied.
- Metadata tracking. Keeping track of metadata is particularly important in geospatial service chaining (Alameh 2003). Using the service capabilities advertised in OWL-S, the output metadata can be derived from the input metadata by modifying, deleting or inserting metadata elements affected by the operation of the service.

This metadata component provides metadata constraints to validate the physical model, thus avoiding attempts to execute invalid service chains and computing resources waste. A final executable service chain or workflow is shown in the bottom part of Figure 3.

5.3 *Workflow execution*

The chaining result is represented as the 'Composite Process' ontology of OWL-S with service groundings. It can be executed in an OWL-S engine. There are many XML-based service composition languages, such as BPEL4WS, the Web Services Flow Language (WSFL) and the Web Service Choreography Interface (WSCI). van der Aalst (2003) compared the control flow of these common service composition languages. Twenty flow control constructs, such as sequence, parallel split and choice, were identified as those most often required when designing a service composition language. The 'Composite Process' ontology of OWL-S has control constructs for these pattern definitions. Since most control construct definitions originate from the service composition languages, a composite process can be converted to any of the service composition languages to enable execution in the existing engines for these languages (Yue *et al.* 2007a).

6. Prototype implementation and result analysis

6.1 *Implementation*

A prototypical system[25] composed of the following components has been developed (Figure 4).

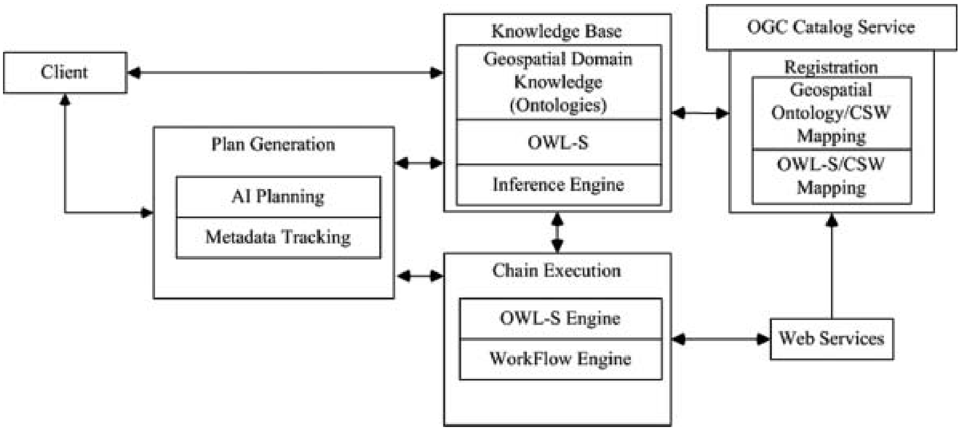

Figure 4. System architecture.

(1) *Client:* assists users in formulating goals on the basis of the ontology supported by the system. Users can also use the client to contribute a composite process or atomic process of OWL-S into the knowledge base.
(2) *Knowledge base:* includes the definitions of geospatial domain ontology and services ontologies. An inference engine is attached for reasoning.
(3) *Plan generation*: uses AI planning approaches to generate the process model to achieve the users' goals, i.e. data products. The metadata-tracking component is used to validate the process model and generate the executable service chain. The final result is produced as the OWL-S composite process and sent to the chain execution component for execution.
(4) *Chain execution:* executes an OWL-S composite process by invoking each individual service and passing the data between the services according to the flow specified in the composite process. The individual services are invoked using the service groundings. The composite process can also be converted to a workflow language and executed in the corresponding workflow engine.
(5) *Catalog:* provides a mapping description between ontologies and the CSW registration information model to register ontologies into the CSW to facilitate discovery of data and services.

This system operates on the common data and service environment addressed in Section 3.1, with additional OWL/OWL-S ontology descriptions. The OWL-S API[26] is used for OWL-S parsing and execution. It has been further extended to support the invocation of HTTP GET and POST as well as just SOAP. Jena Transitive and the OWL-Micro Reasoner[27] have been selected for reasoning. The former is efficient, but has limited capabilities and cannot handle ABOX reasoning in the precondition checking of the case described here. The latter is used for the OWL-S precondition check. The ontologies are registered in the CSW (Yue *et al.* 2006a) and can be queried through the CSW interface. OWLSManager, a component for the management of OWL-S files that can deploy and undeploy OWL-S files into the knowledge base, is developed (Yue *et al.* 2007a).

To run the wildfire prediction scenario in this system, we have implemented all Web services listed in Table 1 and created OWL-S descriptions for these online services.[28] The 52n WPS framework[29] is used to develop all geospatial processing services. WCTS also is treated as a processing service and aligned with WPS by defining a WPS complex data type with a schema following WCTS. As a forthcoming OGC specification, WPS focuses on a standard for data processing rather than just data provision. Therefore, the implementation described here is completely compatible with OGC standards and the interoperability is achieved.

6.2 *Result analysis*

In this scenario, when formulating the goal, John needs to rely on the services to derive a spatial bounding box for the desired wildfire data product, so that the query of the catalog service can have the correct spatial filter. The client helps John to find the DistanceBuffer process. He appends a CTS process and a GetEnvelope process to get the geographic bounding box information. The role of the DistanceBuffer, CTS and GetEnvelope processes here is that of information-providing processes, which refers to services that can be used to gather information during the planning (Sirin *et al.* 2004). From the perspective of AI planning, in offline planning, the process model for service composition is generated before the execution of the service component, while in online planning, the actual process model is created at run time to adapt dynamic environments. For compute-intensive applications such as earth science applications, offline planning is efficient (Ponnekanti and Fox 2002), and thus is preferred to online planning. Although we use information-providing processes and include workflow execution in the process planning, our approach bears more characteristics of offline planning.

Although improvements are needed for our system to be used in real world applications,[30] the process planning for the use case is valid and has its generality. In the first step of this case (see Section 2.1), the user relies on the information-providing processes to get the geographic bounding box as the spatial filter of a CSW query. In the second step (process modeling), a lower-level model is generated by the plan generation component from the user's initial model. The user could also specify a wildfire prediction process with input and output 'DataType's in the initial model if the user is concerned with particular 'DataType's. In the third step (process model instantiation), the user specifies the input 'DataType's of the lower-level process model that are needed to be bound (e.g. 'Wildfire_Danger_Index'). The plan generation component interacts with the CSW and generates the physical model. In the physical model, NOAA NDFD data and NASA MODIS data are bound to the service chain. In the wildfire prediction service, eighteen preconditions are defined. They cover the six input 'DataType's and three types of different metadata entities: the file format, coordinate reference system and grid resolution. The dissatisfaction of any precondition will lead to the insertion of one data reduction and transformation service, based on the built-in rules in the metadata-tracking component. The physical model is transformed to an executable workflow after all preconditions for all services, including data reduction and transformation services, are satisfied. The result is sent to the chain execution component for execution. In all these steps, the user is assisted by the ontologies from the knowledge base in selecting services and does not need to deal with syntactical service descriptions and message element mappings among possibly chainable services. Our method can help the domain expert focus more on the domain knowledge contribution instead of delving into the technical details. In our system, both individual data processing services and valid process models can be shared.

The system is thus a self-evolving system whose capability will increase significantly as more individual services and modeling processes are inserted and/or developed.

7. Related work

Sirin *et al.* (2004) introduced HTN planning into Web service composition, because the concept of action decomposition in HTN planning is similar to the concept of composite process decomposition in OWL-S process ontology (Sirin *et al.* 2004). OWL-S descriptions are translated to a SHOP2 domain. Thus, service composition is transformed into a SHOP2 planning problem. However, SHOP2 does not handle concurrency currently. Therefore, those OWL-S composite processes with 'Split' and 'Split + Join' control constructs could not be transformed. Also using SHOP2's theorem prover in the planning process loses the inferencing capability in the OWL DL reasoner. Sirin *et al.* (2004) conducted an initial experiment to incorporate the OWL reasoner into SHOP2. We focus on the geospatial domain and use action decomposition only in the process-modeling phase. The preconditions and effects are used only in the process-instantiation phase. The problem need not be translated to the domain of traditional planning systems. The power of the OWL DL reasoner is also incorporated.

Some efforts on geospatial Web service composition have been reported. One example is the Geosciences Network (GEON) (Jaeger *et al.* 2005). Geospatial Web Services, including data (GML representation) provider services and customized services with vector data processing functionalities, are sampled to compose a workflow manually in the KEPLER system (Ludäscher *et al.* 2005). The KEPLER system provides a framework for workflow support in the scientific disciplines. The major feature of the KEPLER system is that it provides high-level workflow design, while at the same time hiding the underlying complexity of technologies from the user as much as possible. Both Web service technologies and Grid technologies are wrapped as extensions in the system. For example, individual workflow components, such as data movement, database querying, job scheduling and remote execution, are abstracted into a collection of generic, reusable tasks in the Grid environment (Altintas *et al.* 2004). Thus, combining a knowledge representation technique (e.g. OWL and OWL-S), with the lower-level, generic, common scientific workflow tasks in the KEPLER system, is a worthwhile technique for minimizing or eliminating human intervention in the generation and instantiation of workflow. Using semantically augmented metadata to annotate data and services is important to automatic service and data discovery (Lutz *et al.* 2003; Lutz and Klien 2006; Klien *et al.* 2006). Ontologies, related in both simple taxonomic and non-taxonomic ways, are employed using subsumption reasoning to improve the discovery of services and the recall and resolution of data. Since 2005, OGC has issued the Geospatial Semantic Web Interoperability Experiment (GSW IE) (Kolas *et al.* 2005, 2006; Kammersell and Dean 2006; Lutz and Kolas 2007). In this experiment, five types of ontologies are identified, including base geospatial ontology, feature data source ontology, geospatial service ontology, geospatial filter ontology and domain ontology. Based on these ontologies, a user's query can be translated to the data source semantic queries via semantic rules, and then transformed to a WFS query through XSLT. The query is represented using SPARQL, and the semantic rules are represented using SWRL.[31] Some additional efforts (Raskin and Pan 2005; Fox *et al.* 2006) focus on the geoscience ontology and data discovery using the ontology approach. All of them address the usage of ontologies to enhance the capability of the data discovery. For geospatial Web services, Roman *et al.* (2006) plan to use WSMO to facilitate discovery and invocation of semantically described geospatial Web services. Lemmens *et al.* (2006) have experimented with WSDL-S in their use case implementation. And Yue *et al.* (2006b) introduce the path planning for chaining Geospatial Web Services.

8. Conclusions and future work

Semantic Web Services provide a prospect to automate the earth science applications in a service-oriented environment. In this paper, we have shown how Semantic Web Services can be used to address the data semantics, functional semantics and execution semantics of interoperable geospatial Web services. Furthermore, we present an approach, namely, Semantic Web Services-based process planning, through which Geospatial Web Services can be chained semi-automatically. The approach allows ontologies to be combined with AI planning methods to help a user dynamically create an executable workflow for distributed earth science applications. TBOX and ABOX reasoning are incorporated into process modeling and process model instantiation phases to help create a plan. To enable metadata tracking, metadata constraints are defined in the preconditions of OWL-S. Experiments were performed with an ISO 19115 ontology and a GML ontology, in particular, in a common data and service environment conforming to the interoperable standards and specifications from OGC and W3C. Schema mapping among multiple geospatial standards including WCS, WPS, WCTS and GML is demonstrated. A case study of the chaining process for wildfire prediction illustrates the applicability of the approach.

A distinct advantage of ontology is that it bridges the gap between the concepts people use and the data machines interpret. Our research and prototypical implementation indicate that ontology alone is not sufficient for service chaining. It must be combined with rules and AI planning to achieve the automation of service chaining. The full automation of process planning is still an ongoing research activity. However, human-involved planning is practical for sophisticated domain applications. And our goal is enable maximum automation. It should be noted that although in our current implementation, the three phases in the process planning are executed uni-directionally, they can also interact multi-directionally. The failure of execution at the posterior phase can notify the upper phase or a human user to select an alternative plan. This recursive process continues until a plan is applicable. We are also implementing our approach to the GeoBrain (Di 2004), an open, interoperable, Web-based geospatial information services and modeling system for higher-education teaching and research.

In addition to the extension of WSDL grounding described in Section 4, some additional extensions might be needed in the future. For example, an OWL-S composite process is executed by invoking each individual service and passing the data between the services according to the flow specified in the composite process. The invocation of individual service is based on the service grounding. An OWL-S engine needs to transform the input values (i.e. OWL instances) of an atomic process to a WSDL message for the invocation of services. The service output message after invocation is further transformed into the output value (i.e. new OWL instances) of that atomic process. According to the data flow, the new OWL instances are cast to the input values of the subsequent atomic process in the workflow. While this cast can be handled by the OWL-S engine, it is better for the service grounding to deal with it so that the user has more control. An initial attempt has been made in our implementation to add the 'sparqlTransformationString' to the service grounding in addition to the 'xsltTransformationString'. The design was inspired by the lowing schema mapping from SAWSDL. Another example is the modularity of precondition definitions. As a preliminary implementation attempt, eighteen preconditions for the wildfire prediction service were defined. We believe that there might be a better and simple way to define such constraints. A more flexible definition of preconditions, one that goes beyond the existing expressive limitations of OWL-S, needs to be found. We will pay close attention to the development of Semantic Web Services as our research goes on.

Acknowledgements

This work is supported by a grant from US National Geospatial-Intelligence Agency NURI program (HM1582-04-1-2021, PI: Dr Liping Di). We are grateful to the two anonymous reviewers, Dr Barry Schlesinger and Professor Mark Gahegan, for their detailed comments that helped improve the quality of the paper.

Notes

1. Earth System Grid (ESG) (http://www.earthsystemgrid.org/).
2. The Geosciences network (GEON). A Research Project to Create Cyberinfrastructure for the Geosciences. NSF/ITR (www.geongrid.org).
3. Web Services Architecture. W3C Working Group Note 11 February 2004, W3C (http://www.w3.org/TR/ws-arch/).
4. Semantic Web Services Initiative (SWSI) (http://www.swsi.org/).
5. We include workflow execution in the process planning because it interacts with other phases and the failure of execution can trigger a new round of planning.
6. This is a projection best for the visualization of continental USA.
7. http://www.nws.noaa.gov/ndfd/
8. Daily maximum temperature, minimum temperature and precipitation amount can be retrieved from the NDFD weather elements.
9. http://edcdaac.usgs.gov/datapool/datatypes.asp
10. Each FPAR and LAI grid is considered as valid for 7 days, until it is replaced with the next. For prediction purpose, they can be valid on that time. The newest LULC data are only available at day 289 of year 2004. The data are assumed to be valid on that time.
11. Web Services Description Language (WSDL) 1.1. W3C Note 15 March 2001 (http://www.w3.org/TR/2001/NOTE-wsdl-20010315).
12. OWL Web Ontology Language Reference, W3C (http://www.w3.org/TR/owl-ref).
13. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation 10 February 2004 (http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/).
14. **Example (HDFEOS):** *Given the atomic concept* MD_Format *and the atomic role* name_MD_Format, *we can describe a HDFEOS File Format using constructors as* MD_Format $\sqcap$ $\exists$name_MD_Format.application/HDF-EOS
15. OWL-based Web Service Ontology (OWL-S) (http://www.daml.org/services/owl-s/).
16. Web Service Modeling Ontology (WSMO) (http://www.w3.org/Submission/WSMO/).
17. Web Service Semantics (WSDL-S) (http://www.w3.org/2005/04/FSWS/Submissions/17/WSDL-S.htm).
18. Semantic Annotations for WSDL (SAWSDL) (http://www.w3.org/TR/sawsdl/).

19. Semantic Web Services Framework (SWSF) (http://www.w3.org/Submission/2005/SUBM-SWSF-20050909/).
20. SPARQL, Query Language for RDF (http://www.w3.org/TR/rdf-sparql-query/).
21. OWL-S 1.2 Pre-Release (http://www.ai.sri.com/daml/services/owl-s/1.2/).
22. http://loki.cae.drexel.edu/~wbs/ontology/
23. Reusable XSLT files are available at http://www.laits.gmu.edu/geo/nga/wildfirecase.html
24. Business Process Execution Language for Web Services (BPEL4WS) (http://www-128.ibm.com/developerworks/library/specification/ws-bpel/).
25. This system was demonstrated in the Semantic Web Challenge of the 5th International Semantic Web conference in Athens, GA, USA.
26. http://www.mindswap.org/2004/owl-s/api/
27. http://jena.sourceforge.net/inference/index.html
28. All our Web services, ontologies and related resources are available at http://www.laits.gmu.edu/geo/nga/wildfirecase.html
29. http://www.incubator52n.de/twiki/bin/view/Processing/52nWebProcessingService
30. The training process to create the meaningful model for the wildfire prediction process did not work well due to the memory requirement and training speed of the logistic regression component of the WEKA data mining package. We used the historical weather data of California from NASA AMES' Ecological Forecast Lab (http://ecocast.arc.nasa.gov/) and historical MODIS data ordered from the NASA LPDAAC. WEKA is a collection of machine learning algorithms for data mining developed by the University of Waikato (http://www.cs.waikato.ac.nz/~ml/weka/). We used a simplified prediction model instead by assigning empirical weight values to each predict variable and made the model as a simple weighted average model.
31. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission (2004) (http://www.w3.org/Submission/SWRL/).

References

ALAMEH, N., 2003, Chaining geographic information web services. *IEEE Internet Computing*, **7**(5), pp. 22–29.

ALTINTAS, I., BIRNBAUM, A., BALDRIDGE, K., SUDHOLT, W., MILLER, M., AMOREIRA, C., POTIER, Y. and LUDÄSCHER, B., 2004, A framework for the design and reuse of grid workflows. *Lecture Notes in Computer Science*, **3458**, pp. 119–132.

BAADER, F. and NUTT, W., 2003, Basic description logics. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi and P. Patel-Schneider (Eds). *The Description Logic Handbook: Theory, Implementation and Applications*, pp. 47–100 (Cambridge: Cambridge University Press).

BERNERS-LEE, T., HENDLER, J. and LASSILA, O., 2001, The semantic web. *Scientific American*, **284**(5), pp. 34–43.

BRACHMAN, R.J. and LEVESQUE, H.J., 2003a, *Knowledge Representation and Reasoning*, p. 172 (Los Altos, CA: Morgan Kaufmann Publishers).

BRACHMAN, R.J. and LEVESQUE, H.J., 2003b, *Knowledge Representation and Reasoning*, pp. 314–316 (Los Altos, CA: Morgan Kaufmann Publishers).

DI, L., 2004, GeoBrain – a web services based geospatial knowledge building system. In *Proceedings of NASA Earth Science Technology Conference 2004*, 22–24 June 2004, Palo Alto, CA, USA (CD-ROM).

DI, L., 2005, A framework for developing web-service-based intelligent geospatial knowledge systems. *Journal of Geographic Information Sciences*, **11**(1), pp. 24–28.

FORSTER, I., 2005, Service-oriented science. *Science*, **308**(5723), pp. 814–817.

FOX, P., MCGUINNESS, D., MIDDLETON, D., CINQUINI, L., DARNELL, A., GARCIA, J., WEST, P., BENEDICT, J. and SOLOMON, S., 2006, Semantically-enabled large-scale science data repositories. *Lecture Notes in Computer Science*, **4273**, pp. 792–805.

GRUBER, T.R., 1993, A translation approach to portable ontology specification. *Knowledge Acquisition*, **5**(2), pp. 199–220.

Hey, T. and Trefethen, A.E., 2005, Cyberinfrastructure for e-science. *Science*, **308**(5723), pp. 817–821.

Kammersell, W. and Dean, M., 2006, Conceptual search: incorporating geospatial data into semantic queries. In *Proceedings of Terra Cognita 2006: the 5th International Semantic Web Conference*, 5–9 November 2006, Athens, GA, USA.

Klien, E., Lutz, M. and Kuhn, W., 2006, Ontology-based discovery of geographic information services – an application in disaster management. *Computers, Environment and Urban Systems*, **30**(1), pp. 102–123.

Kolas, D., Hebeler, J. and Dean, M., 2005, Geospatial semantic web: architecture of ontologies. In *Proceedings of the 1st International Conference on GeoSpatial Semantics (GeoS 2005)*, pp. 183–194 (Berlin: Springer).

Kolas, D., Dean, M. and Hebeler, J., 2006, Geospatial semantic web: architecture of ontologies. In *Proceedings of 2006 IEEE Aerospace Conference*, 4–11 March 2006, Big Sky, MT, USA.

Jaeger, E., Altintas, I., Zhang, J., Ludäscher, B., Pennington, D. and Michener, W., 2005, A scientific workflow approach to distributed geospatial data processing using web services. In *Proceedings of 17th International Conference on Scientific and Statistical Database Management (SSDBM'05)*, 27–29 June 2005, Santa Barbara, CA, USA, pp. 87–90.

Lemmens, R., Wytzisk, A., By, R.D., Granell, C., Gould, M. and van Oosterom, P., 2006, Integrating semantic and syntactic descriptions to chain geographic services. *IEEE Internet Computing*, **10**(5), pp. 18–28.

Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E., Tao, J. and Zhao, Y., 2005, Scientific workflow management and the Kepler system. *Concurrency and Computation*, **18**(10), pp. 1039–1065.

Lutz, M. and Kolas, D., 2007, Rule-based discovery in spatial data infrastructures. *Transactions in GIS*, **11**(3), pp. 317–336.

Lutz, M. and Klien, E., 2006, Ontology-based retrieval of geographic information. *International Journal of Geographical Information Science*, **20**(3), pp. 233–260.

Lutz, M., Riedemann, C. and Probst, F., 2003, A classification framework for approaches to achieving semantic interoperability between GI web services. *Lecture Notes in Computer Science*, **2825**, pp. 200–217.

Papazoglou, M.P., 2003, Service-oriented computing: concepts, characteristics and directions. In *Proceedings of the 4th International Conference on Web Information Systems Engineering (WISE 2003)*, 10–12 December 2003, Roma, Italy, pp. 3–12.

Peer, J., 2005, *Web Service Composition as AI Planning – A Survey* (St Gallen: University of St Gallen).

Percivall, G. (Eds), 2002, The OpenGIS abstract specification, topic 12: OpenGIS service architecture. Version 4.3. OGC 02-112 (Wayland, MA: Open Geospatial Consortium Inc.).

Ponnekanti, S.R. and Fox, A., 2002, SWORD: a developer toolkit for web service composition. In *Proceedings of the International World Wide Web Conference*, May 2002, Honolulu, HI, USA, pp. 83–107.

Rao, J., 2004, Semantic web service composition via logic-based program synthesis. PhD thesis (Trondheim: Norwegian University of Science and Technology).

Rao, J. and Su, X., 2004, A survey of automated web service composition methods. In *Proceedings of the 1st International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, July 2004, San Diego, CA, USA, pp. 43–54.

Raskin, R. and Pan, M., 2005, Knowledge representation in the semantic web for earth and environmental terminology (SWEET). *Computer and Geosciences*, **31**(9), pp. 1119–1125.

Roman, D., Klien, E. and Skogan, D., 2006, SWING – a semantic web services framework for the geospatial domain. In *Proceedings of Terra Cognita 2006: International*

Semantic Web Conference ISWC'06 Workshop, 5–9 November 2006, Athens, GA, USA.

RUSSEL, S. and NORVIG, P., 2003, Artificial intelligence: a modern approach. 2nd edn, pp. 375–458 (Upper Saddle River, NJ: Prentice-Hall Inc.).

SCHUT, P. and WHITESIDE, A., 2005, *OpengGIS® Web Processing Service*, Version 0.4.0, OGC 05-007r4 (Wayland, MA: Open Geospatial Consortium Inc.).

SHETH, A., 2003, Semantic web process lifecycle: role of semantics in annotation, discovery, composition and orchestration. Invited talk at WWW 2003 Workshop on E-Services and the Semantic Web, Budapest, Hungary.

SIRIN, E., PARSIA, B., WU, D., HENDLER, J. and NAU, D., 2004, HTN planning for web service composition using SHOP2. *Journal of Web Semantics*, **1**(4), pp. 377–396.

SRIVASTAVA, B. and KOEHLER, J., 2003, Web service composition – current solutions and open problems. In *Proceedings of ICAPS 2003 Workshop on Planning for Web Services*, June 2003, Trento, Italy, pp. 28–35.

VAN DER AALST W., 2003, Don't go with the flow: web services composition standards exposed. *IEEE Intelligent Systems*, **15**, pp. 72–76.

VOLZ, R., 2004, Web ontology reasoning with logic databases. PhD thesis (Fak. f. Wirtschaftswissenschaften: Universität Karlsruhe).

YUE, P., DI, L., YANG, W., YU, G. and ZHAO, P., 2007a, Semantics-based automatic composition of geospatial Web services chains. *Computers and Geosciences*, **33**(5), pp. 649–665.

YUE, P., DI, L., YANG, W., YU, G., ZHAO, P. and GONG, J., 2007b, Semantics-enabled metadata generation, tracking and validation in the geospatial web service composition for distributed image mining. In *Proceedings of the 2007 IEEE International Geoscience and Remote Sensing Symposium (IGARSS07)*, 23–27 July 2007, Barcelona, Spain, pp. 334–337.

YUE, P., DI, L., ZHAO, P., YANG, W., YU, G. and WEI, Y., 2006, Semantic augmentations for geospatial catalogue service. In *Proceedings of the 2006 IEEE International Geoscience and Remote Sensing Symposium (IGARSS06)*, 31 July–4 August 2006, Denver, CO, USA, pp. 3486–3489.

YUE, P., DI, L., YANG, W., YU, G. and ZHAO, P., 2006b, Path planning for chaining geospatial web services. *Lecture Notes in Computer Science*, **4295**, pp. 214–226.