International Journal of
Geo-Information

Article

Provenance in GIServices: A Semantic Web Approach

Zhaoyan Wu, Hao Li and Peng Yue *

School of Remote Sensing and Information Engineering, Wuhan University, 129 Luoyu Road, Wuhan 430079, China; zhaoyan_wu@whu.edu.cn (Z.W.); leehomm@whu.edu.cn (H.L.)
* Correspondence: pyue@whu.edu.cn

Abstract: Recent developments in Web Service and Semantic Web technologies have shown great promise for the automatic chaining of geographic information services (GIService), which can derive user-specific information and knowledge from large volumes of data in the distributed information infrastructure. In order for users to have an informed understanding of products generated automatically by distributed GIServices, provenance information must be provided to them. This paper describes a three-level conceptual view of provenance: the automatic capture of provenance in the semantic execution engine; the query and inference of provenance. The view adapts well to the three-phase procedure for automatic GIService composition and can increase understanding of the derivation history of geospatial data products. Provenance capture in the semantic execution engine fits well with the Semantic Web environment. Geospatial metadata is tracked during execution to augment provenance. A prototype system is implemented to illustrate the applicability of the approach.

Keywords: provenance; geospatial semantic web service; semantic execution engine; GIService; geoprocessing workflow

1. Introduction

As Earth-observing technologies develop, the amount of geospatial data will grow to multi-exabytes very soon. For example, the volume of data collected by Landsat-7, Landsat-8, MODIS, and Sentinel satellites has reached 5 petabytes (PB) per day [1], far more than Earth scientists can hope to analyze. Approaches for semi-automated or automated discovery and dissemination of geospatial knowledge for Earth science applications are urgently needed. One major approach is to promote the use of Semantic Web and Web Service technologies. The Web Service technologies can significantly reduce the data volume, computing steps, and resources required by the end-user [2], while the Semantic Web technologies allow the semantics of data and services to be machine-understandable for more effective discovery, integration, and reuse of geospatial data and services [3]. With ontology support, systems using these technologies can automatically construct an executable workflow (also called a service chain in a service-oriented environment) given the users' descriptions of what they want and the available services, as well as the input data distributed over the Web, and execute this workflow to generate a user-specific product. Some contributions in this area are the proposal for the Semantic Web Service [4], the automatic discovery and composition of Semantic Web Services [5], the Open Geospatial Consortium (OGC)'s Geospatial Semantic Web Interoperability Experiment, and geospatial applications of Semantic Web Service technologies [6,7].

With the advancement of e-Science or Cyberinfrastructure, Foster [8] uses the term *Service-Oriented Science* to refer to the scientific research supported by distributed networks of interoperating services. Web service technologies are now widely used to support the Cyberinfrastructure and lead to the development of a series of geographic information services (GIService) [9]. The development of Cyberinfrastructure-based geographical information systems (CyberGIS) will also drive the adoption of Web services [10]. Traditionally, Earth science data products are produced in the scientific data centers with pre-established

check for updates

Citation: Wu, Z.; Li, H.; Yue, P. Provenance in GIServices: A Semantic Web Approach. *ISPRS Int. J. Geo-Inf.* **2023**, *12*, 118. https://doi.org/10.3390/ijgi12030118

Academic Editors: Wolfgang Kainz and Dev Raj Paudyal

Received: 6 January 2023
Revised: 27 February 2023
Accepted: 7 March 2023
Published: 9 March 2023

processing steps or workflows. In the distributed information infrastructure, processing steps become standard-compliant, chainable service components, which are often dynamically discovered and automatically chained together as workflows to derive data and high-level information products [11]. In order for users to have an informed and trusted understanding of products generated automatically by distributed services, provenance, the processing history of data products, must be provided to them even before such user-specific products are generated. Moreover, provenance can help users find dependencies among physically existing data products and support data quality analysis such as error source identification and propagation. Yue [12] has presented an approach to capturing the provenance of geospatial data before service chains are executed. Such provenance information as source data and processing steps provides a context in which end users can evaluate the fitness of service chains and interpret the data products that service chains will deliver. However, the provenance information collected focuses on the ancestor relationships between geospatial data products and does not include the causal relations between execution parameter values and data products. There is still a demand for capturing the provenance of the geospatial data products that have already been generated. This paper extends the original work by automatically recording provenance during the execution of semantic GIService chains. Semantic service chains are generated based on the automatic discovery and composition of Semantic Web Services, which have been demonstrated in the previous work [13,14]. The work in this paper goes one step further to capture the provenance of semantic service chains in their executions. It proposes a three-level view of provenance in the context of automatic GIService composition. A semantic execution engine is extended to track the provenance. The semantic execution engine is a workflow engine that can execute semantic descriptions of workflows, such as semantic service chains represented using ontologies. It takes ontological instances as inputs, transforms them into syntactic descriptions for invoking individual services, and generates new ontological instances as outputs of executions. In addition, the work presents provenance queries based on the three-level view that combine domain ontologies and rules for inferring provenance. The query and inference of provenance rely on the current Semantic Web technologies including Resource Description Framework (RDF), the Web Ontology Language (OWL), and the SPARQL Protocol and RDF Query Language (SPARQL). The foundation of knowledge representation formalism for OWL is the Description Logic (DL). However, DL ontologies have expression limitations in inferring implicit relations [15]. Here, the rule-based approach is combined with DL ontologies to infer new assertions and add more declarative expressivity, which has not been attempted in the previous work.

The contribution of this paper is the provenance model tailored to the automatic GIService composition and automatic provenance capture in the semantic execution engine. The three-level conceptual view of provenance (i.e., knowledge provenance, service provenance, and data provenance) corresponds to the three phases of automatic GIService composition: process modeling, process model instantiation, and workflow execution. It provides an information context in which users can check query results against different levels of provenance, thus supporting plan adjustment in the different phases of service composition. Domain-specific information, such as the semantics of geospatial data and services, is important to interpret data products. To impose a domain-specific view of provenance, geospatial semantics, including geospatial data and service ontologies, are referred to in the provenance ontology through multiple links. Semantic execution engines are used for the execution of workflows in the Semantic Web environment. Domain-specific provenance can be enriched through geospatial metadata tracking during execution. Unlike most available workflow provenance approaches, which use engines running on syntactic descriptions of workflows, the workflows running in the semantic execution engine are semantic service chains, which can take advantage of mediation mechanisms in the semantic execution engine and make the approach fit well with the Semantic Web environment.

The remainder of the paper is organized as follows. Section 2 introduces a geospatial example to help in understanding the work and derive requirements for provenance.

Section 3 introduces background concepts and previous work on the semantic descriptions of GIServices. Section 4 presents a conceptual view of provenance, its ontological modeling, and rule declaration. Using provenance ontologies, the paper describes an approach that automatically captures provenance in a semantic execution engine in Section 5. A prototypical implementation is presented in Section 6. The work is compared with related work in Section 7, and conclusions and pointers to future work are given in Section 8.

2. Provenance for Geospatial Data Products in a Distributed Service Environment

2.1. An Example: Landslide Susceptibility Scenario

This section introduces a landslide susceptibility use case to illustrate the requirements for provenance when automatically deriving data products using distributed geospatial data and services. A disaster manager, John, wants to know: "What is the susceptibility of Dimond Canyon, California, United States to a landslide?" Such information is not directly available, yet it can be generated on-demand by automatic service chaining using available data and geoprocessing services. In the geospatial domain, the OGC has developed a series of standards for GIServices. The geospatial data can be provided by the OGC Web Coverage Service (WCS). WCS is the OGC service standard that defines a standard interface and protocol to access coverage data on the Web. Coverage data such as a raster image is a kind of digital geospatial information representing space-varying phenomena.

Both data and geoprocessing services can be published and discovered using a standards-compliant metadata catalogue service. Assume two simple computation models for landslide susceptibility index are available as landslide susceptibility services: the simpler one (the model in the top left of Figure 1a is based only on terrain slope and aspect, while the other (the model in the top right of Figure 1a also considers land cover type and vegetation growing condition. Each of the computation models also involves other models: deriving terrain slope and aspect from the Digital Terrain Model (DEM), calculating the Normalized Difference Vegetation Index (NDVI) as an indicator of vegetation growing conditions, and generating the land cover types using the classification of the Earth-observing imagery provided by a Web Image Classification Service (WICS). Thus, the answer to the geospatial question is potentially derivable from the semantic relationships among data and services, such as the semantics of input/output data in a service operation. For example, the computation models can be chained together using an input-output *DataType* match, i.e., by chaining two services so that the first service can output the semantically-matched data as an input to the second one. Figure 1a shows that different computation models can generate different results. The landslide susceptibility image generated by the model with four types of input data has more detail than the model with two types of input data. The provenance information for these data products, therefore, can help analyze the quality of the data products.

The scenario in Figure 1a is comparatively simple, since it assumes that all inputs are already processed into a form ready for analysis using the so-called data reduction and transformation services, including data format conversion, coordinate system transformation, and resolution conversion (i.e., resampling/interpolation/regridding). Figure 1b further illustrates an example process for data reduction and transformation. The provenance, then, provides information to position the DEM data in the original coverage dataset, e.g., by comparing the spatial bounding boxes of the result DEM and the original dataset. This kind of provenance can be called spatial provenance. The example here is used throughout the paper for illustration purposes. It is noted, however, that the approach is designed to be general and not restricted to only this example.

The Semantic Web Service, a combination of the Semantic Web and Web service technologies, is 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". The semantic representation in the Semantic Web Service provides an intelligent mechanism for organizing information and services, allowing human queries to be correctly structured for the available application services. It is then possible to determine automatically the relationships between the data and services available and build scientific workflows to derive geospatial information and knowledge from Earth science data distributed over the Web. A prototypical system for management and automatic composition of geospatial services using Semantic Web Services, called OWLSManager, has been developed [6]. Using OWL-S, a Web Ontology Language based Web Service Ontology, OWLSManager can generate and execute a service chain to derive the landslide susceptibility index automatically.

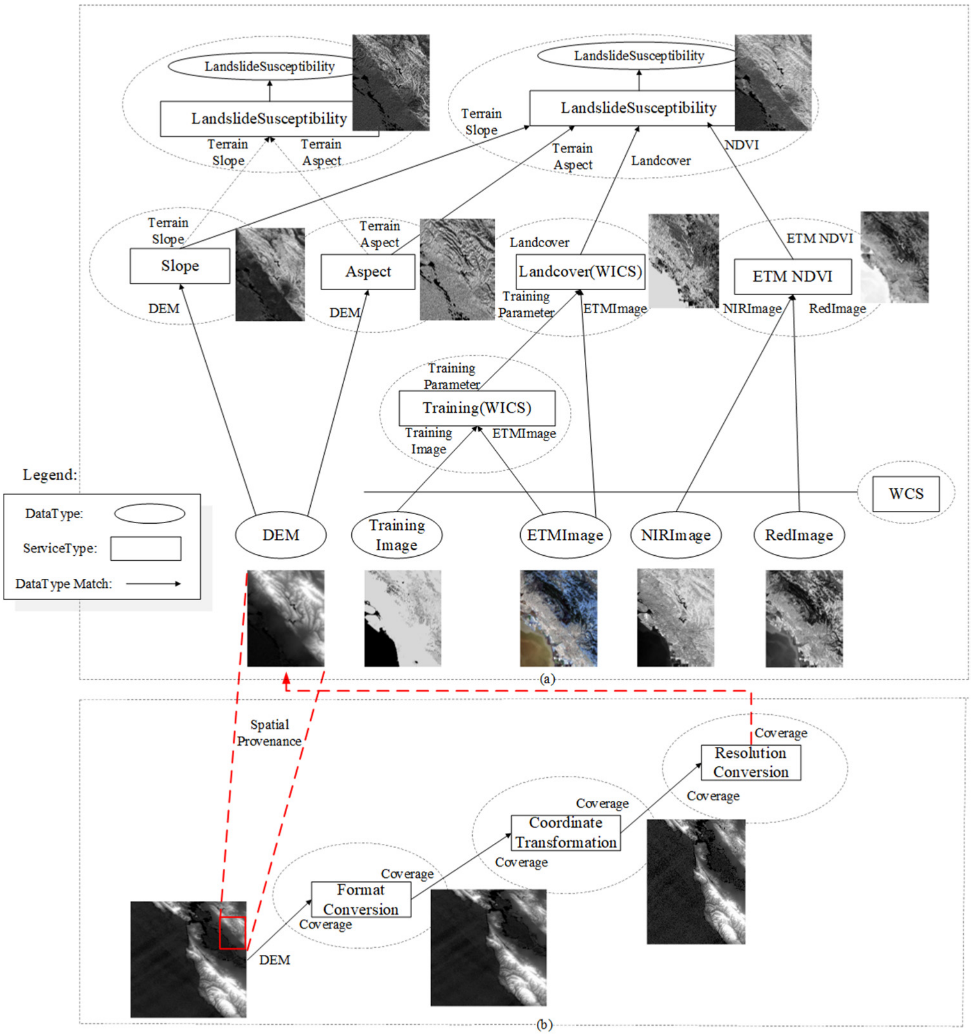


Figure 1. Landslide susceptibility case: (**a**) two computation models for landslide susceptibility index; (**b**) computation model for transforming DEM data into a form ready for analysis.

2.2. Requirements for Provenance

In the scenario described above, although geospatial products can be automatically generated to answer John's question, John may have the following questions:

- Before I can trust it for my decisions, how was the landslide susceptibility index derived?
- What are the source data and their spatial and temporal ranges?
- Is there an error in the source data and geoprocessing services involved?
- Can I use a different computational model for the landslide susceptibility index?

The answers to the above questions can support users in making decisions. In answering these questions, sufficient provenance information for the derived products must be available. In a system such as OWLSManager for automatic composition of services, provenance must satisfy the following important requirements, which can be satisfied in a Semantic Web environment, as addressed in the following sections.

2.2.1. The Different Levels of Provenance in Automatic GIService Composition

In order to provide appropriate provenance information for data products generated by automatic GIService composition, the modeling of provenance information should embody different levels of information generated in the three-phase procedure for automatic GIService composition. Automatic service composition, or called automatic service chaining, has been studied extensively. A number of examples in the literature demonstrate a three-phase procedure for automatic service composition [16], and can be used in GIService composition [7,13]: process modeling, process model instantiation, and workflow execution. The first phase is to construct an abstract process model, which consists of control flow and data flow among process nodes. A process node represents one type of individual service that shares the same functional behaviors: functionality, input, and output. The second phase is to transform the abstract process model into an executable service chain. And the third phase is to execute the service chain in a workflow engine to generate the requested data products. The provenance in these three phases, therefore, can provide end users with an informed understanding of different phases in the derivation history of geospatial data products.

2.2.2. Capturing Provenance in the Semantic Execution Engine

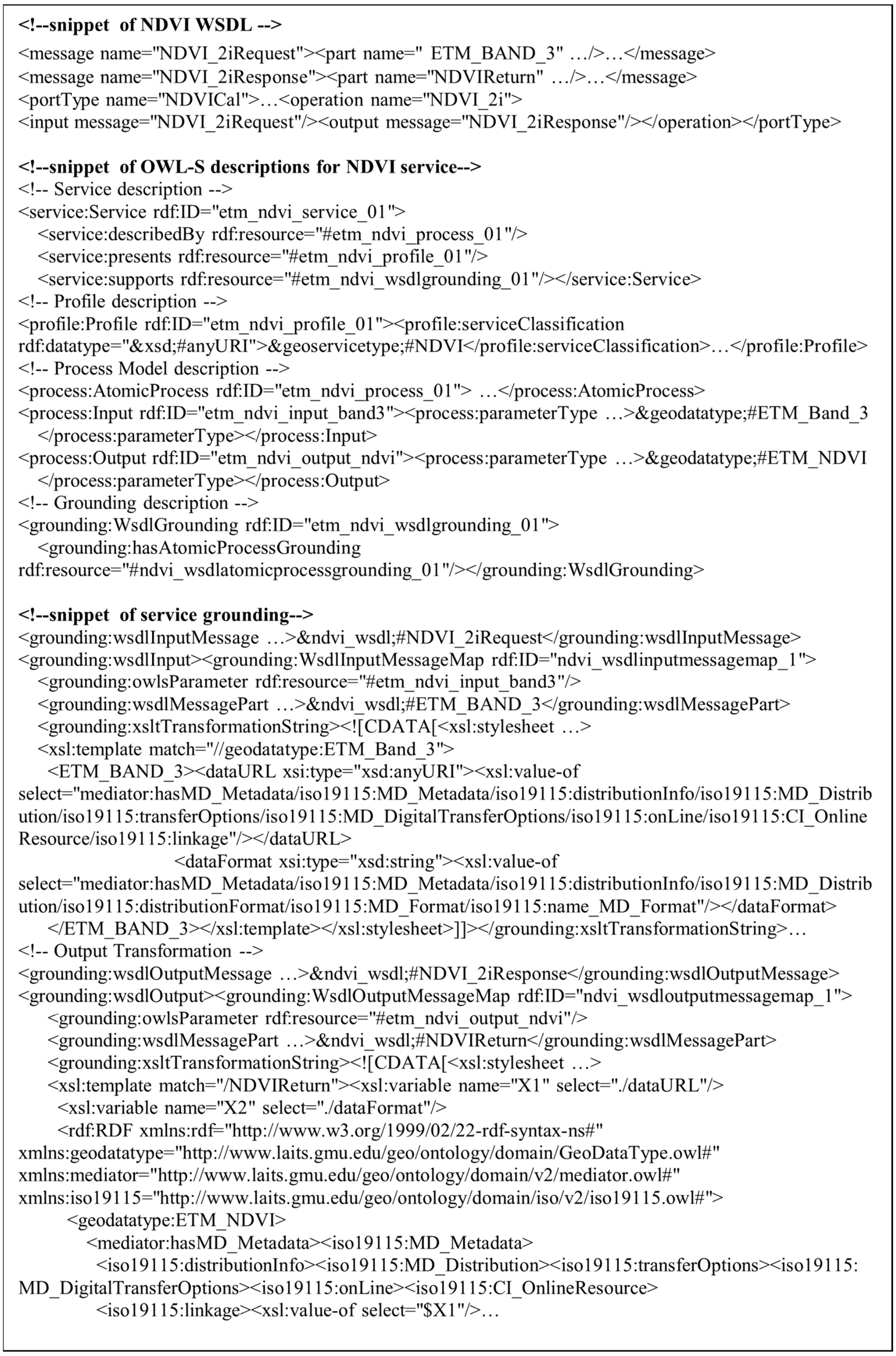
The semantic execution engine fits the Semantic Web environment by its nature of running semantic service chains. For example, OWLSManager can execute semantic service chains represented using OWL-S. Syntactic execution engines for service chains should instead support the execution of syntactic descriptions of service chains such as Web Services Business Process Execution Language based descriptions. The use of a semantic execution engine with semantic descriptions of service chains using OWL-S or Web Service Modeling Ontology can take advantage of the mediation capabilities of the Semantic Web Service technologies [17]. The execution engine is used to derive geospatial data products and therefore can be used to collect provenance.

2.2.3. Tracking Domain-Specific Metadata

The applications in the geospatial domain often require multiple modeling or processing steps involving heterogeneous data provided by different vendors in distributed locations. Ontologies for provenance should therefore link complex metadata for source or intermediate data products, such as data format, spatial projection, and region. The metadata for data products is not typically considered to have provenance since it is not related to how the data products are created. However, when linked with metadata for workflows and their executions, the metadata can be used to augment provenance. While metadata can be represented using available metadata ontologies linked in the provenance, generation, and capture of this metadata as instances in the provenance ontologies need investigations. The execution engine itself does not generate geospatial metadata for data products unless the metadata is specified as execution parameters, such as input and output file formats. More geospatial metadata should be tracked and added as facts to the provenance knowledge base to provide comprehensive provenance information.

3. Semantic Descriptions of GISservices

Although there are some up-to-date proposals for semantic descriptions of services, OWL-S is used as the vehicle in this paper for semantic representation of GIServices. An OWL-S file consists of three main parts: the service profile, the service model (i.e., process), and the service grounding. Figure 2 shows a snippet of the Web Services Description Language (WSDL) and OWL-S for the NDVI computation service. A geospatial *DataType* (*ETM_NDVI*) and a geospatial *ServiceType* (*NDVI*) are linked to the OWL-S descriptions. Geospatial *DataType* entities conceptualize the scientific meanings of distributed geospatial data, such as the science keyword collection of the Global Change Master Directory (GCMD); thus, they can be used to represent the semantics of input and output data in a geospatial service operation. Geospatial *ServiceType* entities are defined according to the scientific problems that the geospatial services focus on solving. They can be developed, for example, by conceptualizing service keyword collection in GCMD. WSDL is a standard for the syntactic description of Web services. The service grounding part of OWL-S provides information on how the syntactic and semantic worlds are bridged, e.g., by grounding the input/output ontology concepts to the input/output message of WSDL using Extensible Stylesheet Language (XSL) Transformations (Figure 2). A process can be either atomic or composite. An atomic process in OWL-S describes the behavior of an atomic service. A composite process is a collection of subprocesses or atomic processes with control and data flow relationships. Both atomic and composite processes can be advertised through a service profile ontology by their functionalities, inputs, outputs, preconditions, and effects (IOPE).



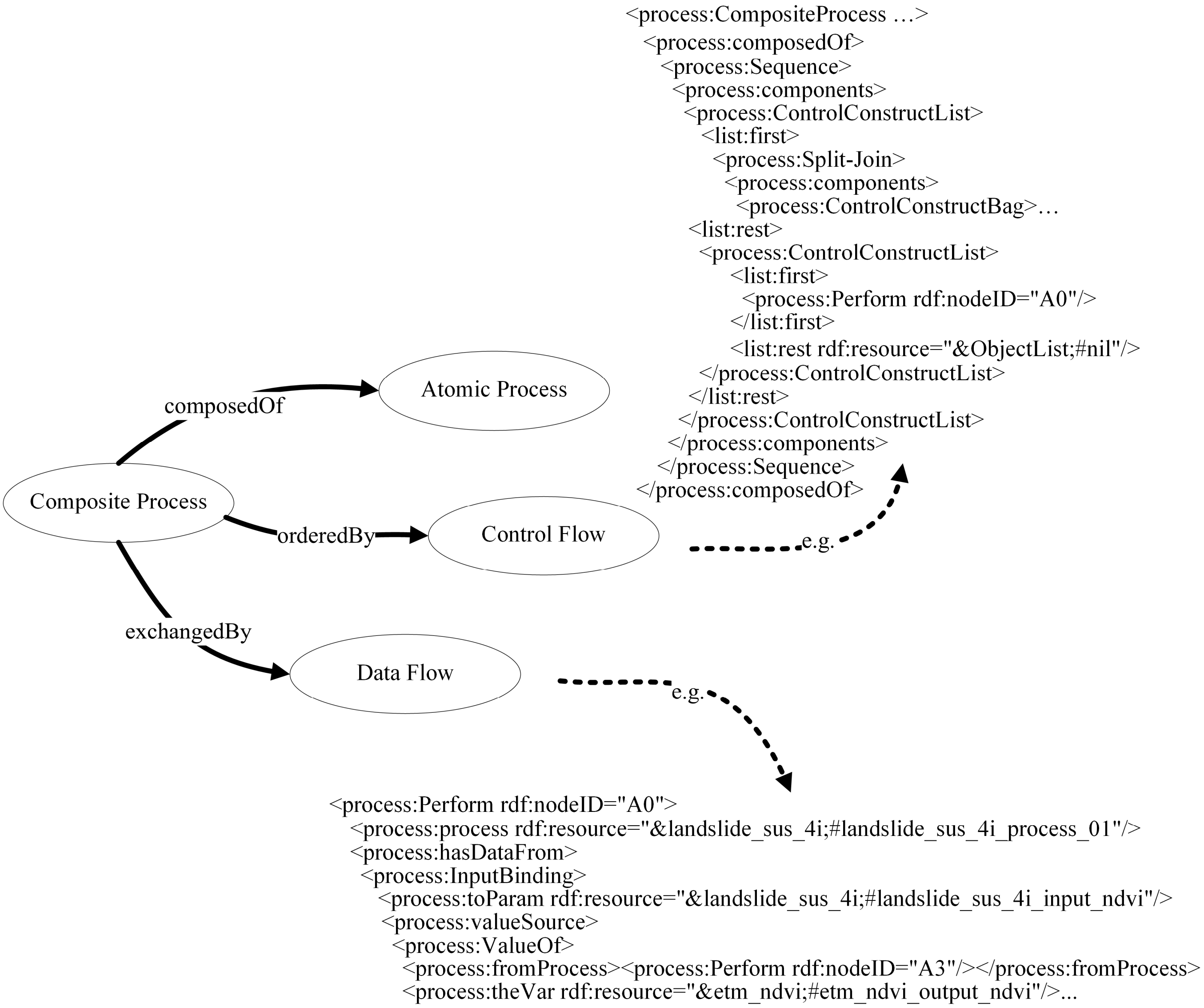
```
<!--snippet  of NDVI WSDL -->
<message name="NDVI_2iRequest"><part name=" ETM_BAND_3" …/>…</message>
<message name="NDVI_2iResponse"><part name="NDVIReturn" …/>…</message>
<portType name="NDVICal">…<operation name="NDVI_2i">
<input message="NDVI_2iRequest"/><output message="NDVI_2iResponse"/></operation></portType>

<!--snippet  of OWL-S descriptions for NDVI service-->
<!-- Service description -->
<service:Service rdf:ID="etm_ndvi_service_01">
  <service:describedBy rdf:resource="#etm_ndvi_process_01"/>
  <service:presents rdf:resource="#etm_ndvi_profile_01"/>
  <service:supports rdf:resource="#etm_ndvi_wsdlgrounding_01"/></service:Service>
<!-- Profile description -->
<profile:Profile rdf:ID="etm_ndvi_profile_01"><profile:serviceClassification
rdf:datatype="&xsd;#anyURI">&geoservicetype;#NDVI</profile:serviceClassification>…</profile:Profile>
<!-- Process Model description -->
<process:AtomicProcess rdf:ID="etm_ndvi_process_01"> …</process:AtomicProcess>
<process:Input rdf:ID="etm_ndvi_input_band3"><process:parameterType …>&geodatatype;#ETM_Band_3
  </process:parameterType></process:Input>
<process:Output rdf:ID="etm_ndvi_output_ndvi"><process:parameterType …>&geodatatype;#ETM_NDVI
  </process:parameterType></process:Output>
<!-- Grounding description -->
<grounding:WsdlGrounding rdf:ID="etm_ndvi_wsdlgrounding_01">
  <grounding:hasAtomicProcessGrounding
rdf:resource="#ndvi_wsdlatomicprocessgrounding_01"/></grounding:WsdlGrounding>

<!--snippet  of service grounding-->
<grounding:wsdlInputMessage …>&ndvi_wsdl;#NDVI_2iRequest</grounding:wsdlInputMessage>
<grounding:wsdlInput><grounding:WsdlInputMessageMap rdf:ID="ndvi_wsdlinputmessagemap_1">
  <grounding:owlsParameter rdf:resource="#etm_ndvi_input_band3"/>
  <grounding:wsdlMessagePart …>&ndvi_wsdl;#ETM_BAND_3</grounding:wsdlMessagePart>
  <grounding:xsltTransformationString><![CDATA[<xsl:stylesheet …>
  <xsl:template match="//geodatatype:ETM_Band_3">
   <ETM_BAND_3><dataURL xsi:type="xsd:anyURI"><xsl:value-of
select="mediator:hasMD_Metadata/iso19115:MD_Metadata/iso19115:distributionInfo/iso19115:MD_Distrib
ution/iso19115:transferOptions/iso19115:MD_DigitalTransferOptions/iso19115:onLine/iso19115:CI_Online
Resource/iso19115:linkage"/></dataURL>
                   <dataFormat xsi:type="xsd:string"><xsl:value-of
select="mediator:hasMD_Metadata/iso19115:MD_Metadata/iso19115:distributionInfo/iso19115:MD_Distrib
ution/iso19115:distributionFormat/iso19115:MD_Format/iso19115:name_MD_Format"/></dataFormat>
   </ETM_BAND_3></xsl:template></xsl:stylesheet>]]></grounding:xsltTransformationString>…
<!-- Output Transformation -->
<grounding:wsdlOutputMessage …>&ndvi_wsdl;#NDVI_2iResponse</grounding:wsdlOutputMessage>
<grounding:wsdlOutput><grounding:WsdlOutputMessageMap rdf:ID="ndvi_wsdloutputmessagemap_1">
   <grounding:owlsParameter rdf:resource="#etm_ndvi_output_ndvi"/>
   <grounding:wsdlMessagePart …>&ndvi_wsdl;#NDVIReturn</grounding:wsdlMessagePart>
   <grounding:xsltTransformationString><![CDATA[<xsl:stylesheet …>
   <xsl:template match="/NDVIReturn"><xsl:variable name="X1" select="./dataURL"/>
    <xsl:variable name="X2" select="./dataFormat"/>
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:geodatatype="http://www.laits.gmu.edu/geo/ontology/domain/GeoDataType.owl#"
xmlns:mediator="http://www.laits.gmu.edu/geo/ontology/domain/v2/mediator.owl#"
xmlns:iso19115="http://www.laits.gmu.edu/geo/ontology/domain/iso/v2/iso19115.owl#">
     <geodatatype:ETM_NDVI>
       <mediator:hasMD_Metadata><iso19115:MD_Metadata>
        <iso19115:distributionInfo><iso19115:MD_Distribution><iso19115:transferOptions><iso19115:
MD_DigitalTransferOptions><iso19115:onLine><iso19115:CI_OnlineResource>
        <iso19115:linkage><xsl:value-of select="$X1"/>…
```

Figure 2. A snippet of WSDL and OWL-S for the NDVI computation service.

The semantics of a GIService chain can be represented using composite process ontology. Figure 3 illustrates semantic descriptions for the landslide susceptibility case, using workflow ontologies in OWL-S. The control flow is represented by control constructs such as Sequence and Split-Join. 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. For example, as shown in Figure 3, the output (etm_ndvi_output_ndvi) of the NDVI computation process is linked to the input (landslide_sus_4i_input_ndvi) of the landslide susceptibility atomic process. Note that the purpose is not to propose new ontologies for semantic descriptions of service chains. Rather, the existing set of example ontologies from Yue [6] is used to illustrate automatic provenance capture in the semantic execution engine.



```
<process:CompositeProcess …>
  <process:composedOf>
    <process:Sequence>
      <process:components>
        <process:ControlConstructList>
          <list:first>
            <process:Split-Join>
              <process:components>
                <process:ControlConstructBag>…
        <list:rest>
          <process:ControlConstructList>
            <list:first>
              <process:Perform rdf:nodeID="A0"/>
            </list:first>
            <list:rest rdf:resource="&ObjectList;#nil"/>
          </process:ControlConstructList>
        </list:rest>
      </process:ControlConstructList>
    </process:components>
  </process:Sequence>
</process:composedOf>
```

```
<process:Perform rdf:nodeID="A0">
  <process:process rdf:resource="&landslide_sus_4i;#landslide_sus_4i_process_01"/>
  <process:hasDataFrom>
    <process:InputBinding>
      <process:toParam rdf:resource="&landslide_sus_4i;#landslide_sus_4i_input_ndvi"/>
      <process:valueSource>
        <process:ValueOf>
          <process:fromProcess><process:Perform rdf:nodeID="A3"/></process:fromProcess>
          <process:theVar rdf:resource="&etm_ndvi;#etm_ndvi_output_ndvi"/>...
```

Figure 3. Semantic descriptions for GIService chains.

4. The Three-Level View of Provenance

This section presents the model used for expressing provenance in GIServices. The model is designed within the application context of automatic GIService composition (Section 4.1). However, to support interoperability and situate the work with respect to the PROV-O, a forthcoming W3C standard on ontology for provenance, Section 4.2 presents mappings from the provenance model to PROV-O.

4.1. Knowledge, Service, and Data Provenance

The proposed three-level view of provenance corresponds to the three phases of automatic GIService composition. In the process modeling phase, the provenance is at the knowledge level, namely, knowledge provenance. It contains the data and service ontologies used, as well as process models for complex process modeling. A process model consists of the control flow and data flow among atomic processes. The data flow focuses on the data exchange among atomic processes, while the control flow concerns the order in which atomic processes are executed. An atomic process represents one type of processing service that shares the same functional behaviors such as functionality, input, and output. Process models are based on the knowledge of domain modelers. Such knowledge can be captured in the process, represented using ontologies, and shared and reused by other domain modelers. From this perspective, process models can be regarded as knowledge-oriented provenance. Thus, the knowledge provenance addressed in this paper is from a workflow perspective. Using provenance at this level, users can check the correctness of the process model and select an alternative model when necessary. The second level is the service level, namely service provenance. It includes executable services and chains that can be invoked many times. Using this information, it is possible for users to re-select services based on the performance evaluation of those services. The final level is the data level, namely data provenance level, which concerns execution instances and physically generated data products. The runtime-specific details for each execution, such as values for input parameters, execution data and time, belong to this level.

Based on the conceptual view of provenance, a set of provenance-related ontological entities and their relations can be defined at three levels as shown in Figure 4a. The relations consist of internal relations among entities at the same level and external relations among entities at different levels. Figure 4b shows a lightweight ontology for the purpose of demonstration. The ontology is represented using OWL. More relations and entities can be added to provide richer provenance information.

The knowledge provenance is the process model consisting of geospatial *DataType*, geospatial *ServiceType*, and workflow entities. Linking geospatial *DataTypes*, *ServiceTypes*, and workflow entities such as control flow and data flow together can represent process knowledge. Workflow ontologies can use the ontological entities for control and data flow in OWL-S. Domain experts can then use a model builder [18] to drag and drop geospatial *DataTypes* and *ServiceTypes* in a working panel and link them according to their control and data flow. After the model design is finished, the model, represented using process ontologies in OWL-S, is registered in a geospatial catalogue for sharing [19]. Such a catalogue service allows semantics-enhanced discovery of geospatial data, services/service chains, and process models. More complex models can be built upon those existing models in the model builder.

The service provenance consists of individual services and service chains. Semantic Web Service technologies such as OWL-S can be used to represent both services and service chains. A service chain as a whole can be considered a service and represented using the service ontology in OWL-S.

And the data provenance consists of data products (*ProvenanceGeoDataType*), atomic service executions (*AtomicServiceExecution*), and service chain executions (*CompositeServiceExecution*). Internal relationships in this level include ancestry relationships among data products (*hasGeoDataTypeAncestor* and *hasGeoDataTypeParent*) (1–2), input/output relationships between parameter values and service executions (*hasInput*, *hasOutput*) (3–4), the generation relationship between data products and service executions (*producedBy*) (5), and component relationships between atomic service executions and service chain executions (*isContainedBy*) (6). The range of *hasInput* and *hasOutput* is the *ParamValueBinding*, which binds parameters (using the *param* property) to their values (using the *objectValue* or *literalValue* property) (7).

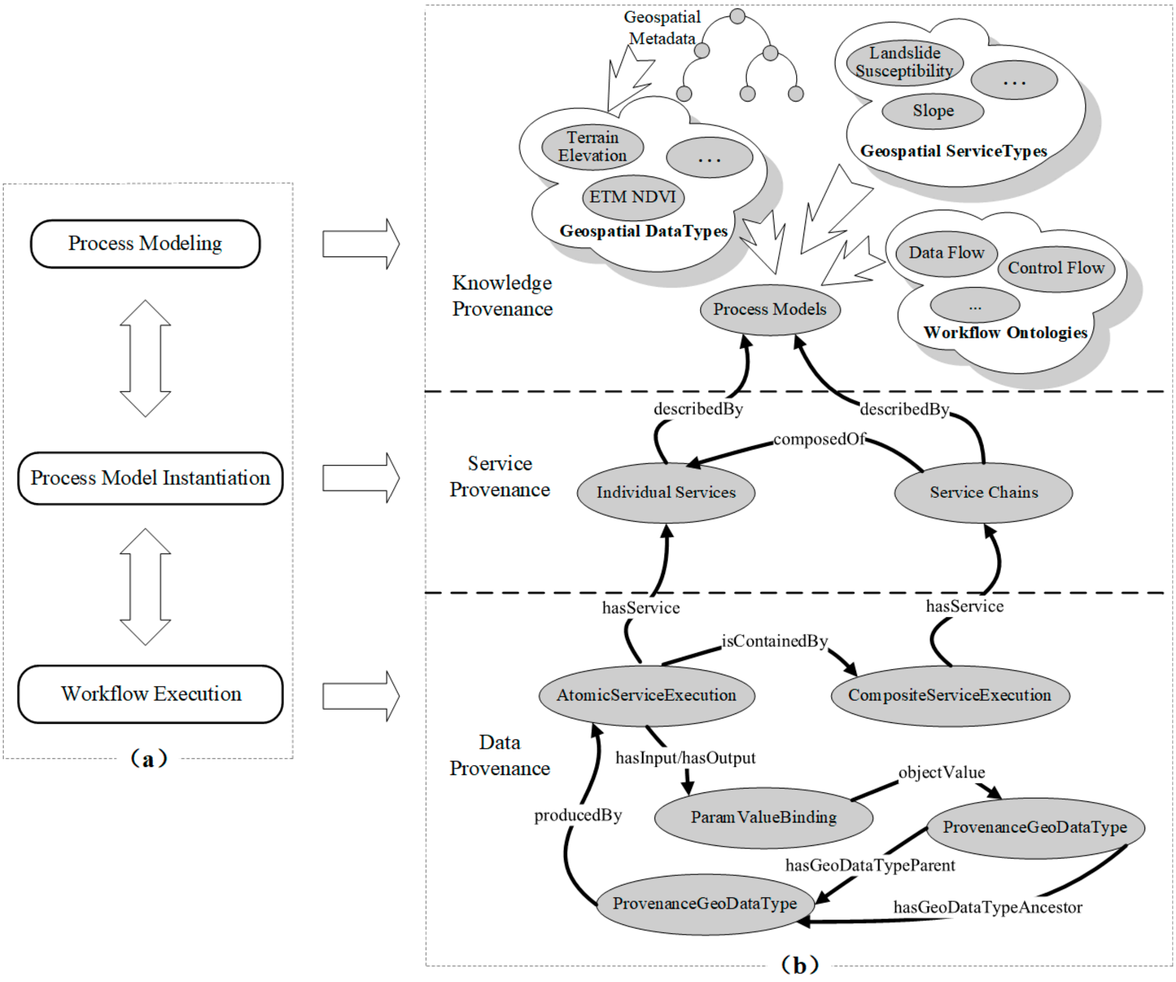


Figure 4. The three-level view of provenance: (**a**) three phases of automatic GIService composition; (**b**) knowledge, service, and data provenance.

The external relations among the three levels are as follows: the knowledge provenance addresses the knowledge aspects of the service provenance, e.g., the *describedBy* relation that links the service chain to its process model (8), while the service provenance addresses the service information for the execution instances in the data provenance, e.g., the *hasService* relation linking a service execution to its service description (9). Since entities at the knowledge and service levels have been defined by existing geospatial data and services ontologies, they are linked to the provenance ontology to create a domain context and to enable the three-level view of provenance. In Description Logic, these relations can be formalized as follows using DL notations.

$$\begin{aligned} \exists\, hasGeoDataTypeAncestor.\ &\top \sqsubseteq ProvenanceGeoDataType \\ &\top \sqsubseteq \forall\, hasGeoDataTypeAncestor.ProvenanceGeoDataType \end{aligned} \tag{1}$$

$$\begin{aligned} \exists\, hasGeoDataTypeParent.\ &\top \sqsubseteq ProvenanceGeoDataType \\ &\top \sqsubseteq \forall\, hasGeoDataTypeParent.ProvenanceGeoDataType \end{aligned} \tag{2}$$

$$\begin{aligned} \exists\, hasInput.\ &\top \sqsubseteq ServiceExecution \\ &\top \sqsubseteq \forall\, hasInput.ParamValueBinding \end{aligned} \tag{3}$$

$$\begin{aligned} \exists\, hasOutput.\ &\top \sqsubseteq ServiceExecution \\ &\top \sqsubseteq \forall\, hasOutput.ParamValueBinding \end{aligned} \tag{4}$$

$$\begin{aligned} \exists\, \mathit{producedBy}. \quad & \top \sqsubseteq \mathit{ProvenanceGeoDataType} \\ & \top \sqsubseteq \forall\, \mathit{producedBy.ServiceExecution} \end{aligned} \tag{5}$$

$$\begin{aligned} \exists\, \mathit{isContainedBy}. \quad & \top \sqsubseteq \mathit{AtomicServiceExecution} \\ & \top \sqsubseteq \forall\, \mathit{isContainedBy.CompositeServiceExecution} \end{aligned} \tag{6}$$

$$\begin{aligned} \exists\, \mathit{objectValue}. \quad & \top \sqsubseteq \mathit{ParamValueBinding} \\ & \top \sqsubseteq \forall\, \mathit{objectValue.ProvenanceGeoDataType} \end{aligned} \tag{7}$$

$$\begin{aligned} \exists\, \mathit{describedBy}. \quad & \top \sqsubseteq \mathit{Service} \\ & \top \sqsubseteq \forall\, \mathit{describedBy.Process} \end{aligned} \tag{8}$$

$$\begin{aligned} \exists\, \mathit{hasService}. \quad & \top \sqsubseteq \mathit{ServiceExecution} \\ & \top \sqsubseteq \forall\, \mathit{hasService.Service} \end{aligned} \tag{9}$$

The rationale for differentiating three levels of provenance is drawn from the automatic service composition and is applicable to the broad general information domain. The domain-specific view of provenance is imposed by linking semantics in geospatial data and services into provenance. For example, when transversing linked provenance data, machines can understand that the NDVI input required by the computation service for landslide susceptibility is provided by the Landsat Enhanced Thematic Mapper (ETM) NDVI and this use is semantically valid. The complexity of geospatial data is addressed by enriching geospatial *DataType* entities with ISO19115-based geospatial metadata ontologies so that geospatial data can be described more precisely. ISO19115 Geographic Information—Metadata is an international standard that defines a set of metadata elements, including elements for identification, data quality, spatial/temporal representation, and content.

The dependency among data products and the dependency between data products and service executions differ. The former is described by the *hasGeoDataTypeAncestor* and *hasGeoDataTypeParent* properties and can provide a clear understanding of data product dependencies. The latter is represented using the *hasInput* and *hasOutput* properties. The *hasGeoDataTypeParent* property links a geospatial data product to its direct ancestor geospatial data product, while *hasGeoDataTypeAncestor* is a transitive property so that the DL reasoner can infer an indirect ancestry relation between data products. However, DL has limited expressivity. For example, it cannot assert that the parent of a product's parent is an ancestor of this product. A combination of DL ontologies and rules can overcome such expression limitations in DL. In addition, using dependencies among data products and service executions, new assertions can be inferred about the dependencies among data products by using rules. For example, if an *AtomicServiceExecution a* has an input *ParamValueBinding b*, and *b* has an object value *ProvenanceGeoDataType c*, and *a* has an output *ParamValueBinding d*, with *d* having an object value *ProvenanceGeoDataType e*, then *e* has parent *ProvenanceGeoDataType c* (10). Using an execution a_i of the terrain slope computation service as an illustration, assume that the dataset c_i is an input parameter value to the execution a_i, and a_i outputs a new dataset e_i. These facts, once combined with the rule, can be used by a reasoner to infer that c_i is the parent of e_i.

$$\begin{gathered} \mathit{AtomicServiceExecution}(a) \wedge \mathit{ParamValueBinding}(b) \wedge \mathit{hasInput}(a, b) \wedge \\ \mathit{ProvenanceGeoDataType}(c) \wedge \mathit{objectValue}(b, c) \wedge \mathit{ParamValueBinding}(d) \wedge \\ \mathit{hasOutput}(a, d) \wedge \mathit{ProvenanceGeoDataType}(e) \wedge \mathit{objectValue}(d, e) \rightarrow \\ \mathit{hasGeoDataTypeParent}(e, c) \end{gathered} \tag{10}$$

4.2. Mapping the Provenance Model to PROV-O for Interoperability

The W3C Provenance Working Group has created an ontology for the W3C PROV Data Model (PROV-DM), named PROV Ontology (PROV-O). The W3C PROV model is intended to be a generic model that allows domain- or application-specific representations of provenance to be translated into the model and interchanged between systems. Mapping the domain-specific provenance model to PROV can support interoperability.

PROV-O encodes the PROV model using OWL. The core of the PROV model is based on three types—entity, activity, and agent—and their relations (Figure 5). An entity is "a physical, digital, conceptual, or other kind of thing" that we want to describe the provenance of. An activity is something that occurs over a period of time and acts upon or with entities. Activities use or generate entities through properties such as *prov:wasGeneratedBy* and *prov:used*. An agent could be a software agent, an organization, or a person. Agents are responsible for activities or entities. Figure 5 shows relations between the GIService provenance model and PROV. In the figure, the core structures of the PROV model are shown, along with types and relations from the GIService provenance model.

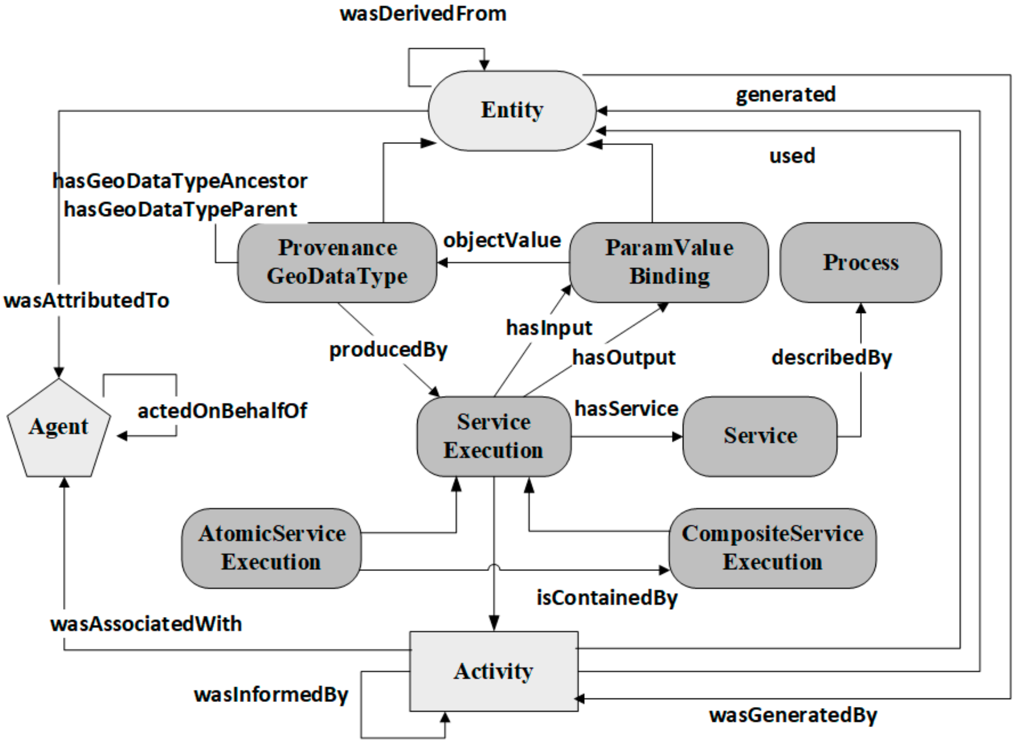


Figure 5. Relations between the GIService provenance model and PROV.

The concepts and relationships in the GIService provenance model that can be mapped to PROV are listed as follows:

geop:ProvenanceGeoDataType ⊑ prov:Entity
geop:ParamValueBinding ⊑ prov:Entity
geop:ServiceExecution ⊑ prov:Activity
geop:hasInput ⊑ prov:used
geop:hasOutput ⊑ prov:generated
geop:hasGeoDataTypeParent ⊑ prov:wasDerivedFrom
geop:hasGeoDataTypeAncestor ⊑ prov:wasDerivedFrom
geop:producedBy ⊑ prov:wasGeneratedBy

In GIServices, the provenance of geospatial data products is a set of assertions about the processing steps that generated them and the data used in those steps. A data product (*ProvenanceGeoDataType*) is a type of entity. A *ServiceExecution* uses or generates *ParamValueBindings*. Thus a *ServiceExecution* is an activity and *ParamValueBinding* can be considered a type of entity. A *ServiceExecution* can use (*used*) a *ParamValueBinding* as input (*hasInput*) and generate (*generated*) a *ParamValueBinding* as output (*hasOutput*). Both *hasGeoDataTypeParent* and *hasGeoDataTypeAncestor* link a data product to its ancestry data product, so they are derivation relations (*wasDerivedFrom*). A data product, once produced by (*producedBy*) a *ServiceExecution*, can be seen as generated by (*wasGeneratedBy*) the *ServiceExecution*. Additional concepts or relationships are not mapped, since they are application-specific, and outside the scope of PROV. With this mapping, it is then possible to generate PROV queries and statements for interoperability.

5. Extending a Semantic Execution Engine to Capture Provenance

A semantic execution engine executes directly semantic descriptions of service chains. Semantic provenance can be collected from the semantic execution engine. Section 5.1 shows how provenance is captured in the execution of semantic service chains. In addition, domain-specific metadata is tracked to enrich the provenance at the data level (Section 5.2).

5.1. Provenance Capture in Executing a Semantic Service Chain

Execution of a semantic service chain requires the specification of the service chain and input parameter values, both of which are semantically represented. For example, the service chain is represented using the composite process ontology of OWL-S, and input parameter values can be represented as bindings between OWL individuals and the process parameters of OWL-S. The ontologies (or ontological knowledge base) such as the Geospatial *DataType* and *ServiceType* ontologies, referred to in the semantic service chains, are another necessary input to execution since they are required when creating ontological instances from ontological concepts.

Figure 6 shows a general execution flow for a semantic service chain. The execution is based on the control flow in the composite process. Different types of control flow exist. For example, the sequence control construct in OWL-S specifies that subprocesses are executed sequentially, while the split control construct in OWL-S describes subprocesses that can be executed in parallel. The execution engine must identify the type of control flow in the composite process and be capable of understanding and parsing a particular type of control flow. The execution of subprocesses in a control flow consists of the execution of each subprocess and the data flow among these subprocesses. A subprocess can be either an atomic process or a composite process. If the subprocess is an atomic process, the input data binding specified using data flow ontologies such as *ValueOf* in OWL-S is used. The input data for the current process is retrieved using the output values of previous processes on which the current process depends or the input values of a parent composite process. Such a semantic input value is further transformed into the syntactic invocation message of the Web service and triggers the execution of the service. The syntactic output message is converted into ontological instances as both facts in the knowledge base and inputs to other processes in the next execution step. If the subprocess is a composite process, the execution will go through the control flow again. Such an execution flow continues until all subprocesses are executed. Finally, the output value binding, which consists of the service parameter in the composite process and its corresponding value, is available.

The mediation mechanism is a distinguishing feature of the semantic execution engine. The mediation includes mediation between input/output ontological instances and mediation between ontological instances and syntactic messages. The former mediation is due to the semantic match between ontological concepts, which results in the data flow among semantically matched input/output. For example, in automatic service composition, using the subsumption reasoning of DL in the ontology, a service composition system determines that the *ETM_NDVI*, the output of the previous service, can feed the input (*NDVI*) of the next service, since *ETM_NDVI* is subsumed by *NDVI*, or they are semantically matched. As a result, the final semantic service chain encodes the input-output relations between *ETM_NDVI* and *NDVI* using the ValueOf entity from OWL-S. In the execution process of the semantic chain, once an ontological instance of the *ETM_NDVI* is generated, the semantic execution engine needs to transform it into an instance of the *NDVI* so that the next service can be executed smoothly. Such a transformation is what the mediation capability should include. It allows on-the-fly transformation between ontological instances, transforming the output ontological instance from the previous process or parent process input to the input ontological instance of the current process. Such transformation requires the support of a knowledge base and highlights the key difference between the semantic execution engine and the syntactic execution engine. As shown in Figure 7, each value that needs to be transferred from *ETM_NDVI* to *NDVI* is represented using a *contextual path* [19], a term denoting a single concept that is in the context of one other concept through a series of properties. For example, "*ETM_NDVI.hasMD_Metadata.referenceSystemInfo. referenceSystemIdentifier.code*" in Figure 7 is such a path. The paths enable registration mappings and facilitate structural transformation of data [19], thus supporting transformation between ontological instances.

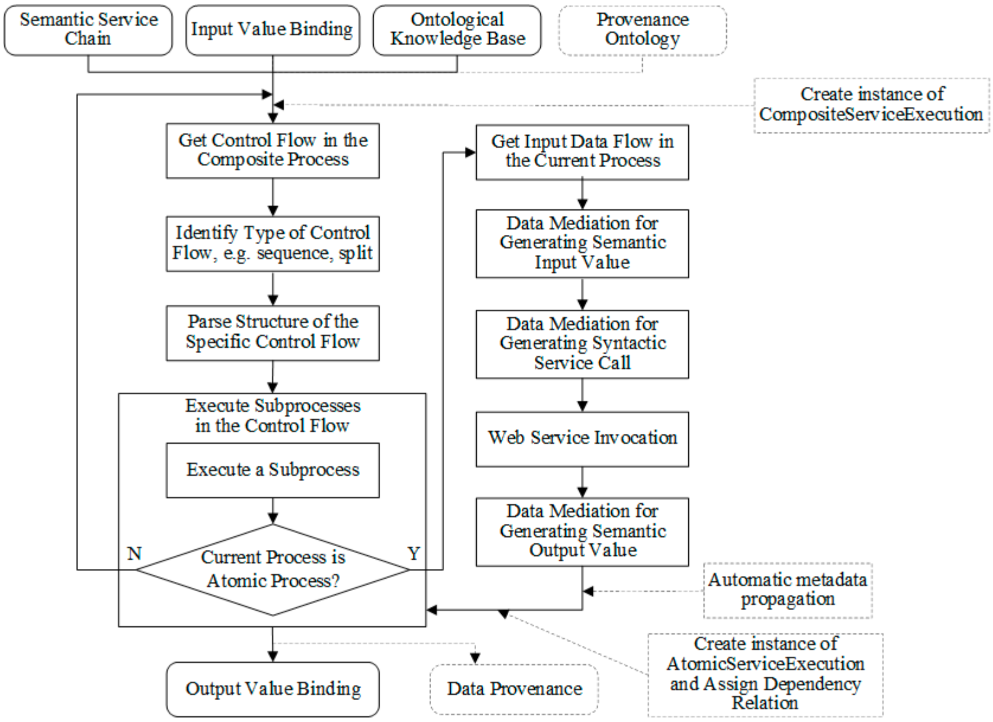

Semantic Service Chain
Input Value Binding
Ontological Knowledge Base
Provenance Ontology
Create instance of CompositeServiceExecution
Get Control Flow in the Composite Process
Identify Type of Control Flow, e.g. sequence, split
Parse Structure of the Specific Control Flow
Execute Subprocesses in the Control Flow
Execute a Subprocess
Current Process is Atomic Process?
N
Y
Get Input Data Flow in the Current Process
Data Mediation for Generating Semantic Input Value
Data Mediation for Generating Syntactic Service Call
Web Service Invocation
Data Mediation for Generating Semantic Output Value
Automatic metadata propagation
Create instance of AtomicServiceExecution and Assign Dependency Relation
Output Value Binding
Data Provenance

Figure 6. Extensions to the execution flow of semantic execution engine.

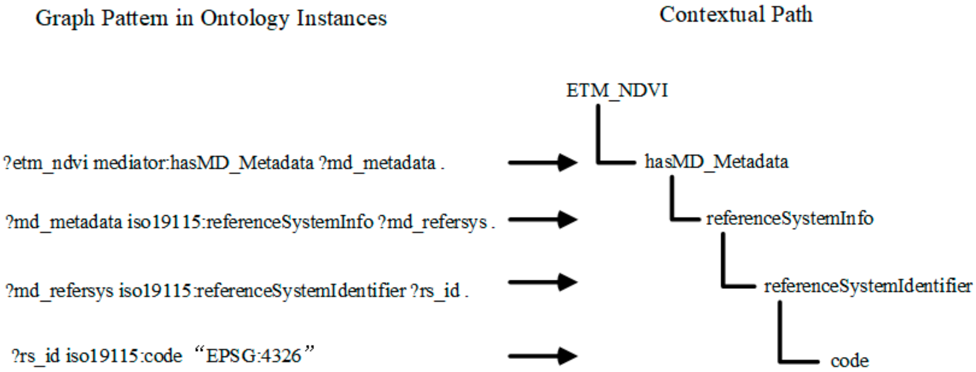

Graph Pattern in Ontology Instances
Contextual Path
?etm_ndvi mediator:hasMD_Metadata ?md_metadata .
?md_metadata iso19115:referenceSystemInfo ?md_refersys .
?md_refersys iso19115:referenceSystemIdentifier ?rs_id .
?rs_id iso19115:code "EPSG:4326"
ETM_NDVI
hasMD_Metadata
referenceSystemInfo
referenceSystemIdentifier
code

Figure 7. A contextual path in supporting transformation between ontological instances.

The mediation between ontological instances and syntactic messages is to bridge the semantic and syntactic world. Semantic Web technologies allow the information semantics to be machine-understandable and enable the wide automation of information retrieval and processing, while syntactic specification has its advantage in having concrete and industry-wide tools. Mediation uses the best of both. Such mediation can be implemented by taking advantage of the XSL transformations in the service grounding part of OWL-S, as shown in Figure 2.

Provenance capture in the semantic execution engine is supported by extending the execution flow of the engine. The dashed lines in Figure 6 show extensions. The provenance ontology, which acts as the provenance knowledge base for provenance generation, is used when running semantic service chains. The execution of a composite process is recorded using an instance of the ontological class *CompositeServiceExecution*, recording the corresponding service chain using the *hasService* relation. Each execution of an atomic process is extended by tracking domain-specific metadata and generating an instance of the ontological class *AtomicServiceExecution*. The service executed is added to the *AtomicServiceExecution* instance through the *hasService* relation. The *AtomicServiceExecution* instance is linked to the parent composite service execution by using the *isContainedBy* relation. Instances of *ProvenanceGeoDataType* are generated for the output geospatial data products of atomic service execution, while the input geospatial data products of atomic service execution are linked to the *ProvenanceGeoDataType* instances generated in the previous processes. Dependencies among these *ProvenanceGeoDataType* instances, such as *hasGeoDataTypeParent* also can be added.

Figure 8 illustrates a portion of the provenance information that was recorded during the execution of a format conversion service to a coordinate transformation service. The provenance ontology entities are instantiated to provide a visualization of the recorded information. Both the input and output *ParamValueBinding* instances of the format conversion service contain a *ProvenanceGeoDataType* instance, which records the geospatial metadata of the data product. For example, one of the provenance information records indicates that the input and output formats were GeoTiff and HDF-EOS, respectively. The input *ParamValueBinding* instance of the coordinate transformation execution is linked to the *ProvenanceGeoDataType* instance that was generated during the previous execution of the format conversion service. The provenance information of the reference system (MD_ReferenceSystem) is emphasized in the coordinate transformation execution. Moreover, additional provenance dependencies, such as ancestor relations, can be added, such as the *hasGeoDataTypeParent* between the output and input *ProvenanceGeoDataType* instances.

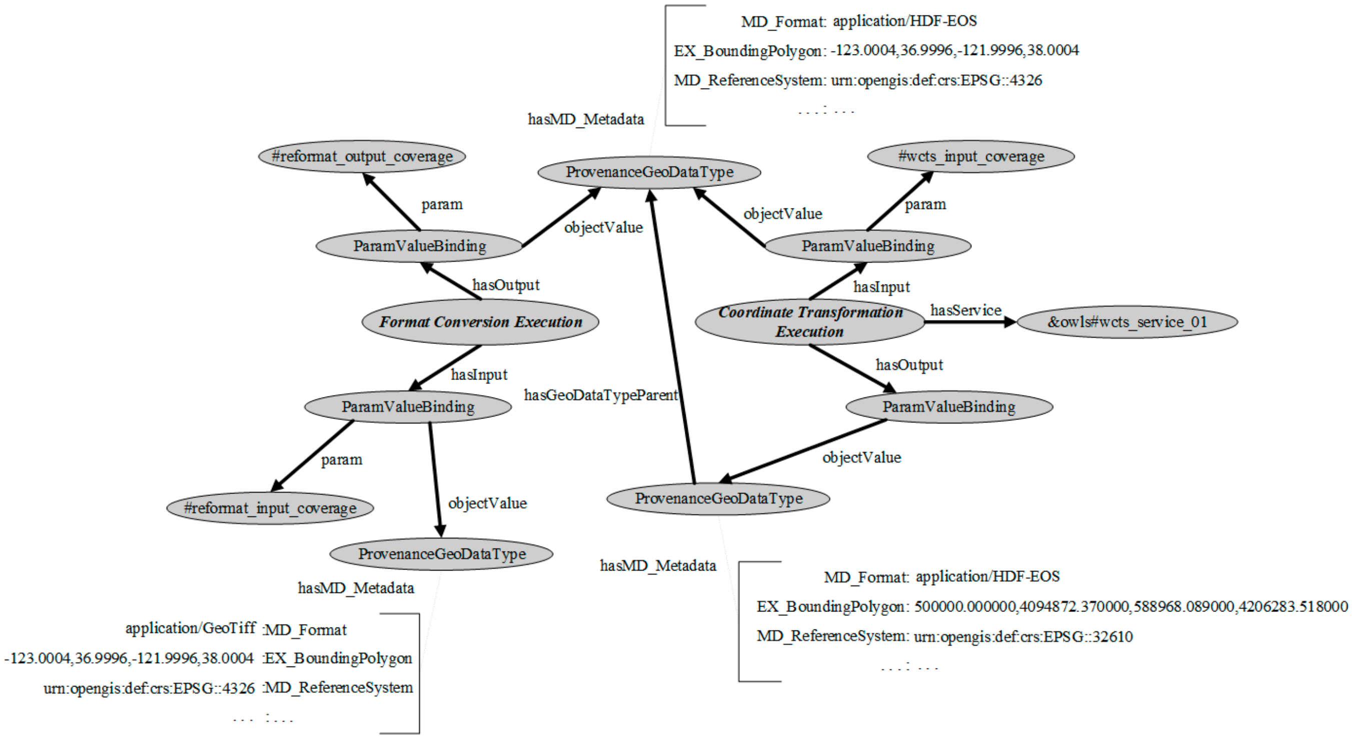

MD_Format: application/HDF-EOS
EX_BoundingPolygon: -123.0004,36.9996,-121.9996,38.0004
MD_ReferenceSystem: urn:opengis:def:crs:EPSG::4326
. . . : . . .
hasMD_Metadata
#reformat_output_coverage
ProvenanceGeoDataType
#wcts_input_coverage
param
objectValue
objectValue
param
ParamValueBinding
ParamValueBinding
hasOutput
hasInput
Format Conversion Execution
Coordinate Transformation Execution
hasService
&owls#wcts_service_01
hasInput
hasOutput
ParamValueBinding
hasGeoDataTypeParent
ParamValueBinding
param
objectValue
objectValue
#reformat_input_coverage
ProvenanceGeoDataType
ProvenanceGeoDataType
hasMD_Metadata
hasMD_Metadata
application/GeoTiff :MD_Format
-123.0004,36.9996,-121.9996,38.0004 :EX_BoundingPolygon
urn:opengis:def:crs:EPSG::4326 :MD_ReferenceSystem
. . . : . . .
MD_Format: application/HDF-EOS
EX_BoundingPolygon: 500000.000000,4094872.370000,588968.089000,4206283.518000
MD_ReferenceSystem: urn:opengis:def:crs:EPSG::32610
. . . : . . .

Figure 8. A snippet of provenance information.

Once all atomic processes in the service chain have been executed, the instances of provenance entities generated are added to the provenance repository, which can be represented using an RDF triple store and queried using SPARQL.

5.2. Tracking Domain-Specific Metadata

In the geoinformatic domain, the ISO19115 Geographic Information—Metadata standard defines a complete set of metadata elements for geospatial data products. However, the complete set of metadata is extensive, and a subset is often used in applications. Therefore, ISO 19115 has identified a set of core metadata elements (either mandatory or optional). Based on this reference, Table 1 lists the core metadata to be tracked throughout the execution of service chains. 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.

Table 1. Core geospatial metadata information to be tracked.

Geospatial Metadata	Tracking
identification	M
constraints	O
data quality	O
maintenance	O
spatial representation	C
reference system	M
content	O
portrayal catalogue	O
distribution	M

The tracked metadata aims to provide the basic minimum metadata information needed for interpreting and evaluating a derived data product. In a simple case such as the format conversion, reprojection, and subsetting for processing the DEM data, or the service deriving terrain slope from DEM data, identification (e.g., spatial extent), reference system (e.g., spatial projection), and distribution information (e.g., file format) are enough for the data product tracked. In cases where both vector and raster data are involved, spatial representation information is mandatory. Other metadata information, such as constraints (e.g., legal restrictions), data quality, and maintenance, is optional. For example, accuracy (e.g., errors) as part of data quality information could be tracked. However, it is better to use the provenance information to analyze error propagations after the execution, instead of tracking errors during the execution. This is outside the scope of this paper. For a detailed account on how to use provenance for data quality analysis, please see [20].

Some ideas on geospatial metadata tracking have been demonstrated in the Semantic Web Challenge of the 5th International Semantic Web conference in Athens, GA, USA. Semantics-enabled metadata are generated and propagated throughout a service chain. This metadata can be employed to validate a service chain, e.g., to determine whether metadata preconditions (such as a specific file format or projection) on the input data of services can be satisfied. In case of failed validation, some data processing services (file format conversion service or coordinate transformation service) can be inserted to modify the data to satisfy the metadata preconditions [21].

The work in the context of this paper focuses on metadata tracking during the execution of service chains. It does not consider the validation and satisfaction of metadata preconditions. The assumption on automatic metadata propagation can be applied, i.e., the output can automatically have some metadata information propagated from the input data. Only those metadata elements affected by the service operation are updated. For example, the output of the slope service, i.e., terrain slope data, can automatically have some metadata information propagated from the input DEM data, such as file format or bounding box.

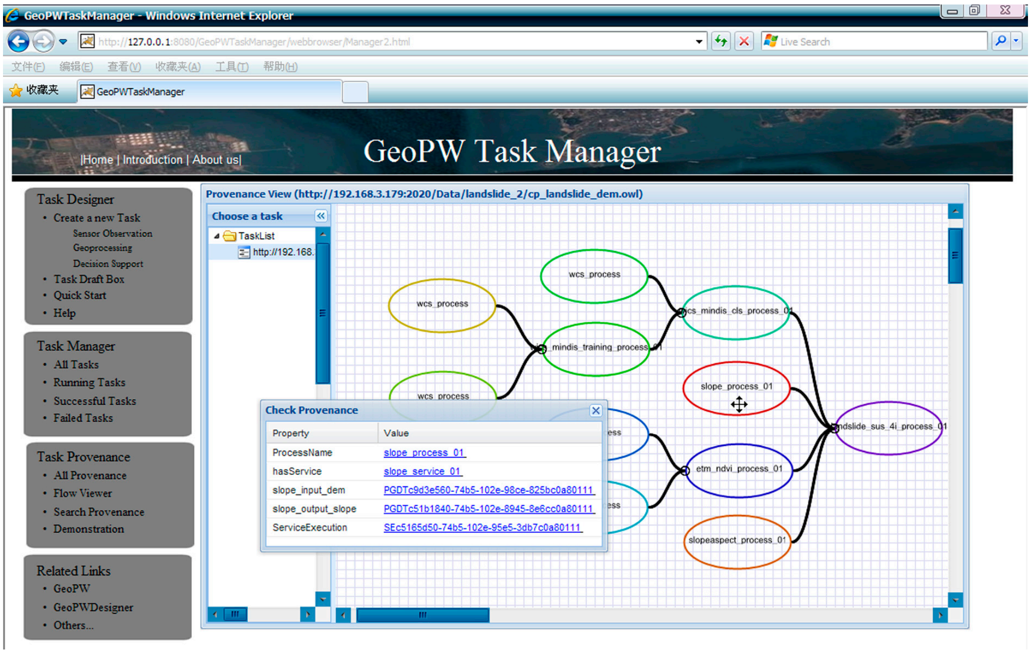
To support the flexibility of the metadata elements tracked, a metadata tracking profile, which can be updated by users and loaded in the execution engine at runtime, is created. The profile specifies the metadata elements to be tracked during the execution. The metadata tracked is guaranteed to be complete only as required by this profile. Figure 8 shows an example where file format, spatial projection, and spatial extent are updated in a series of *ProvenanceGeoDataType* instances.

6. Implementation

OWL-S is used for semantic descriptions of geospatial Web services and service chains. The execution of semantic services and chains uses the OWL-S API. The OWL-S API provides a Java API for programmatic access to read, execute, and write OWL-S descriptions. The API provides an ExecutionEngine that can invoke AtomicProcesses with WSDL groundings and CompositeProcesses that uses control constructs such as Sequence, and Split-Join. The OWLS-Manager, developed using the OWL-S API for automatic composition of geospatial services, has been presented in [6]. It can work as a semantic execution engine by using the ExecutionEngine of the OWL-S API. The ExecutionEngine only supports mediation between ontological instances and syntactic messages. It has been extended in the work to support mediation between input/output ontological instances. Provenance capture in the semantic execution engine, therefore, is implemented by extending the execution flow in the ExecutionEngine of the OWL-S API according to the approach in Section 5.1.

To run the landslide susceptibility case in this system, related Web services and their OWL-S descriptions are developed. The provenance captured is represented in RDF triples and can be loaded in either the Protégé or Jena RDF triple stores, and published on the Web using Joseki [22]. Using the three-level view of provenance ontologies, three types of provenance queries are implemented using SPARQL. The domain semantics are used to help formulate the queries. The first type of query is at the data provenance level, e.g., finding the spatial projection and bounding box of a provenance data product. The second type of query is at the service provenance level, e.g., querying the service and parameter information used in generating ancestor data products of a specific data product. The third type of query is at the knowledge provenance level, e.g., finding the process model for the service chain, which is used to generate a specific data product. Both DL reasoners and rule engines can be applied to the ontological knowledge base to infer new facts. For example, if a DL reasoner that can perform transitive closure using transitive properties is selected, more ancestor data products can be returned using the hasGeoDataTypeAncestor property. Rules make it possible to infer new facts from the existence of multiple properties, such as the hasInput, hasOutput, and hasGeoDataTypeParent properties in the rule defined in Equation (1). Rules are represented using the Semantic Web Rule Language (SWRL) and can be executed using the Jess rule engine.

Figure 9 shows a user interface for provenance navigation. The user interface is developed using JavaScript and runs in the Web browser. The application is an extension to an existing geoprocessing model builder that shows a graphic composite process model using a set of linked process nodes. Knowledge provenance is reloaded by the model builder for alternative selections of process models. Figure 9 shows the result of knowledge provenance based on the landslide susceptibility case. Clicking each process node will open a new window in the Web browser, showing the service and data provenance. The provenance is retrieved by using the Joseki Web query interface to access the RDF provenance dataset.



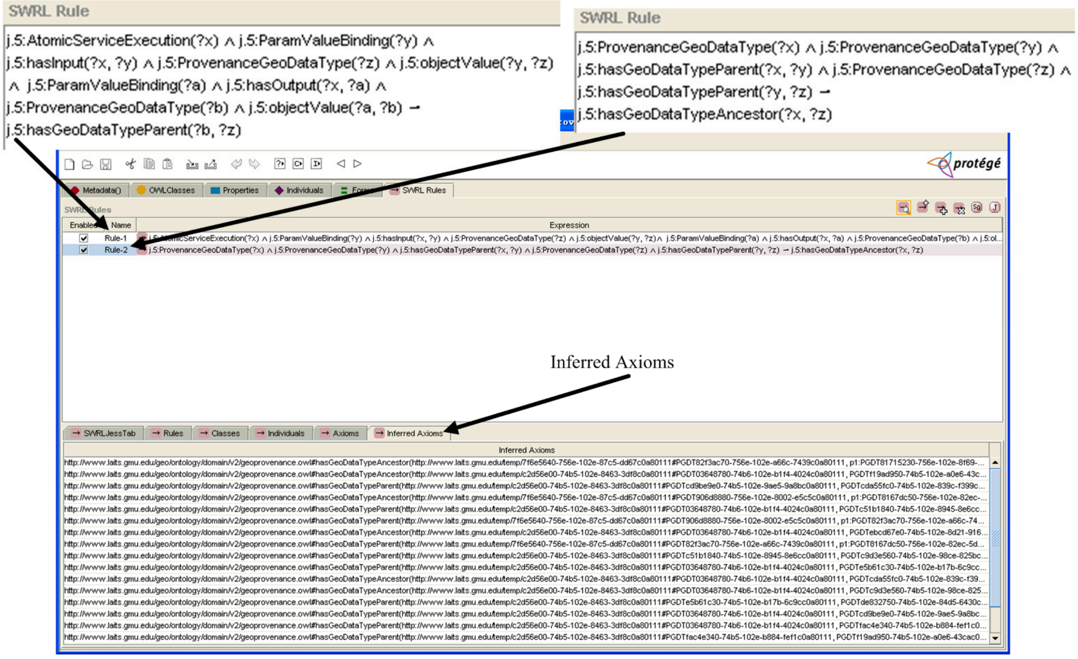
Provenance View (http://192.168.3.179:2020/Data/landslide_2/cp_landslide_dem.owl)

Check Provenance

Property	Value
ProcessName	slope_process_01
hasService	slope_service_01
slope_input_dem	PGDTc9d3e560-74b5-102e-98ce-825bc0a80111
slope_output_slope	PGDTc51b1840-74b5-102e-8945-8e6cc0a80111
ServiceExecution	SEc5165d50-74b5-102e-95e5-3db7c0a80111

Figure 9. Provenance navigation in the Web browser.

The combination of ontologies and rules to infer new assertions is tested using the SWRLtab in Protégé. The OWL file for provenance is loaded into Protégé. As shown in Figure 10, the rule in (10) is represented as an SWRL rule and transferred with OWL to the Jess rule engine using the SWRLJessTab in the Protégé. The second rule in Figure 10 states: if a *ProvenanceGeoDataType a* has parent *ProvenanceGeoDataType b*, and *ProvenanceGeoDataType b* has parent *ProvenanceGeoDataType c*, then *a* has ancestor *ProvenanceGeoDataType c*. After running the inference engine, the assertions on the *hasGeoDataTypeParent* relation, inferred from the *hasInput* and *hasOutput* relations in the first rule, and the *hasGeoDataTypeAncestor* relation, inferred from the second rule, are listed in the "Inferred Axioms" tab in Figure 10.



SWRL Rule

j.5:AtomicServiceExecution(?x) ∧ j.5:ParamValueBinding(?y) ∧ j.5:hasInput(?x, ?y) ∧ j.5:ProvenanceGeoDataType(?z) ∧ j.5:objectValue(?y, ?z) ∧ j.5:ParamValueBinding(?a) ∧ j.5:hasOutput(?x, ?a) ∧ j.5:ProvenanceGeoDataType(?b) ∧ j.5:objectValue(?a, ?b) → j.5:hasGeoDataTypeParent(?b, ?z)

SWRL Rule

j.5:ProvenanceGeoDataType(?x) ∧ j.5:ProvenanceGeoDataType(?y) ∧ j.5:hasGeoDataTypeParent(?x, ?y) ∧ j.5:ProvenanceGeoDataType(?z) ∧ j.5:hasGeoDataTypeParent(?y, ?z) → j.5:hasGeoDataTypeAncestor(?x, ?z)

SWRL Rules

Enabled	Name	Expression
☑	Rule-1	j.5:AtomicServiceExecution(?x) ∧ j.5:ParamValueBinding(?y) ∧ j.5:hasInput(?x, ?y) ∧ j.5:ProvenanceGeoDataType(?z) ∧ j.5:objectValue(?y, ?z)∧ j.5:ParamValueBinding(?a) ∧ j.5:hasOutput(?x, ?a) ∧ j.5:ProvenanceGeoDataType(?b) ∧ j.5:ol...
☑	Rule-2	j.5:ProvenanceGeoDataType(?x) ∧ j.5:ProvenanceGeoDataType(?y) ∧ j.5:hasGeoDataTypeParent(?x, ?y) ∧ j.5:ProvenanceGeoDataType(?z) ∧ j.5:hasGeoDataTypeParent(?y, ?z) → j.5:hasGeoDataTypeAncestor(?x, ?z)

```
Inferred Axioms
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeAncestor(http://www.laits.gmu.edu/temp/7f6e5640-756e-102e-87c5-dd67c0a80111#PGDT82f3ac70-756e-102e-a66c-7439c0a80111, p1:PGDT81715230-756e-102e-8f69-...
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeAncestor(http://www.laits.gmu.edu/temp/c2d56e00-74b5-102e-8463-3df8c0a80111#PGDT03648780-74b6-102e-b1f4-4024c0a80111, PGDTf19ad950-74b5-102e-a0e6-43c...
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeParent(http://www.laits.gmu.edu/temp/c2d56e00-74b5-102e-8463-3df8c0a80111#PGDTcd9be9e0-74b5-102e-9ae5-9a8bc0a80111, PGDTcda55fc0-74b5-102e-839c-f399c...
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeAncestor(http://www.laits.gmu.edu/temp/7f6e5640-756e-102e-87c5-dd67c0a80111#PGDT906d8880-756e-102e-8002-e5c5c0a80111, p1:PGDT8167dc50-756e-102e-82ec-...
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeParent(http://www.laits.gmu.edu/temp/c2d56e00-74b5-102e-8463-3df8c0a80111#PGDT03648780-74b6-102e-b1f4-4024c0a80111, PGDTc51b1840-74b5-102e-8945-8e6cc...
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeParent(http://www.laits.gmu.edu/temp/7f6e5640-756e-102e-87c5-dd67c0a80111#PGDT906d8880-756e-102e-8002-e5c5c0a80111, p1:PGDT82f3ac70-756e-102e-a66c-74...
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeAncestor(http://www.laits.gmu.edu/temp/c2d56e00-74b5-102e-8463-3df8c0a80111#PGDT03648780-74b6-102e-b1f4-4024c0a80111, PGDTebcd67e0-74b5-102e-8d21-916...
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeParent(http://www.laits.gmu.edu/temp/7f6e5640-756e-102e-87c5-dd67c0a80111#PGDT82f3ac70-756e-102e-a66c-7439c0a80111, p1:PGDT8167dc50-756e-102e-82ec-5d...
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeParent(http://www.laits.gmu.edu/temp/c2d56e00-74b5-102e-8463-3df8c0a80111#PGDTc51b1840-74b5-102e-8945-8e6cc0a80111, PGDTc9d3e560-74b5-102e-98ce-825bc...
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeParent(http://www.laits.gmu.edu/temp/c2d56e00-74b5-102e-8463-3df8c0a80111#PGDT03648780-74b6-102e-b1f4-4024c0a80111, PGDTe5b61c30-74b5-102e-b17b-6c9cc...
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeAncestor(http://www.laits.gmu.edu/temp/c2d56e00-74b5-102e-8463-3df8c0a80111#PGDT03648780-74b6-102e-b1f4-4024c0a80111, PGDTcda55fc0-74b5-102e-839c-f39...
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeAncestor(http://www.laits.gmu.edu/temp/c2d56e00-74b5-102e-8463-3df8c0a80111#PGDT03648780-74b6-102e-b1f4-4024c0a80111, PGDTc9d3e560-74b5-102e-98ce-825...
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeParent(http://www.laits.gmu.edu/temp/c2d56e00-74b5-102e-8463-3df8c0a80111#PGDTe5b61c30-74b5-102e-b17b-6c9cc0a80111, PGDTde832750-74b5-102e-84d5-6430c...
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeParent(http://www.laits.gmu.edu/temp/c2d56e00-74b5-102e-8463-3df8c0a80111#PGDT03648780-74b6-102e-b1f4-4024c0a80111, PGDTcd9be9e0-74b5-102e-9ae5-9a8bc...
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeParent(http://www.laits.gmu.edu/temp/c2d56e00-74b5-102e-8463-3df8c0a80111#PGDT03648780-74b6-102e-b1f4-4024c0a80111, PGDTfac4e340-74b5-102e-b884-fef1c0...
http://www.laits.gmu.edu/geo/ontology/domain/v2/geoprovenance.owl#hasGeoDataTypeParent(http://www.laits.gmu.edu/temp/c2d56e00-74b5-102e-8463-3df8c0a80111#PGDTfac4e340-74b5-102e-b884-fef1c0a80111, PGDTf19ad950-74b5-102e-a0e6-43cac0...
```

Figure 10. Inference using SWRL Rule in the Protégé.

7. Related Work and Discussion

The advocate of services appears in various initiatives for building geospatial information infrastructures [23]. The Spatial Data Infrastructures, with their initial purpose of geospatial data sharing and services, can use geoprocessing services to process data into information. The Grid and Cloud computing can provide only a set of low-level middleware and a small part of the functionality required for Cyberinfrastructure. There are substantial research challenges to developing high-level intelligent middleware services and domain-specific services for problem-solving and scientific discovery in Cyberinfrastructure [10]. Chaining geospatial service components in the Cyberinfrastructure helps provide intelligent geospatial services for geospatial information processing. In order to promote the wide deployment of GIServices, interoperability needs specific investigations. While syntactical interoperability has been addressed by a number of OGC specifications on geospatial services, a lot of research uses ontologies to ensure semantic interoperability. Bröring et al. [24] added semantic matchmaking functionality to the OGC Sensor Web Enablement framework to support semantically enabled sensor plug and play. Athanasiou et al. [25] provided middleware for exploring geospatial information using the OGC catalogue service specification. Prudhomme et al. [26] presented an automatic approach for geospatial data integration based on ontology matching, which corresponds to a semantic interpretation process.

Much work in the general information domain has contributed to determining the provenance of workflow-oriented data products [27]. A full-fledged provenance-aware application should take into consideration provenance representation, capture, storage, query, visualization, and applications. The major focus of this paper is on the representation and automatic capture of provenance. The provenance information is linked to content specific to the geospatial domain. Provenance information can be captured by tracing the execution of the workflow engine, aggregating provenance information generated by distributed service providers as a workflow executes, or a combination of the previous two methods. The approach in this paper uses a semantic execution engine to capture provenance. The workflow engine used, different from previous approaches, is Semantic Web oriented and runs semantic services and chains. The tracking of domain-specific metadata and its linkage to provenance impose a domain-specific view of provenance. The semantic execution engine has the advantage of directly supporting the execution of Semantic Web Services. The on-the-fly mediation during execution overcomes the syntactic heterogeneity of services and brings about their semantic interoperability. Provenance capture in the semantic execution engine enjoys the advantages of a semantic execution engine while at the same time allowing the provenance to be captured directly in the form of RDF triples. Therefore, the use of provenance capture in the context of Semantic Web Services and the execution engine for the semantic service chain is Semantic Web-oriented and fits naturally into the Semantic Web environment.

The use of Semantic Web technologies allows provenance, data, services, and workflows to be linked together, providing users with query and inference abilities on the distributed information. The geospatial data, services, workflow, and provenance ontologies using OWL present machine-processable semantics and provide a shared understanding of concepts and their relationships. Some efforts have been devoted to the use of Semantic Web technologies for provenance management and applications. Chebotko et al. [28] use the VIEW, a visual scientific workflow management system, to capture provenance in RDF format during the execution of workflows and propose the design of a relational RDF store for provenance management. Yue et al. [11] proposed a provenance model for linked data in web geoprocessing workflows. Ornelas et al. [29] adopted web services technology to capture both types of provenances (prospective and retrospective). Brown [30] used semantic web technologies to capture provenance meta-data and the data curation processes. In addition, most existing work focuses on adding semantic annotations to the syntactic representation of provenance generated by workflow engines instead of directly generating semantic provenance. This focus is due to the intrinsic nature of workflow engines they use, which were originally designed to work in their proprietary environment, such as a script-based processing environment, instead of the Semantic Web environment.

Although the work in this paper presents its own application domain ontology, it can be migrated to PROV-O with less effort since PROV-O is intended to be a generic provenance model that can accommodate different application contexts of provenance. This could be completed by adding application terms that extend classes and properties from PROV-O, as shown in Section 4.2. To add domain-specific information to the provenance ontologies, Sahoo et al. [31] propose two levels of provenance: the abstract upper-level ontologies, named Provenir ontology, and the detailed domain-specific provenance ontologies. An important concern in the geospatial domain is how geospatial semantics, such as semantics for geospatial data and services, are linked to provenance. Geospatial users can formulate provenance queries that incorporate geospatial semantics using domain-specific concepts such as spatial region or projection. He et al. [32] propose extensions to W3C PROV-XML, which can accommodate geospatial feature provenance at different levels of granularity. The provenance data can be published as part of the Web of Data by making them Linked data-compliant and accessible on the Web. The semantic provenance in our work, therefore, can be extended into the Web of Data in the future. Other works include the creation of provenance using distributed services and provenance management using Semantic Web technologies [33], the use of Proof Markup Language for modeling provenance and provenance search and visualization, and the integration of provenance and Web geoprocessing workflows for semantic discovery of heterogeneous geospatial resources [11]. However, how geospatial provenance information can be captured within the context of GIServices and chains has not been addressed in the literature. In addition, the use of Semantic Web technologies for linking provenance, geospatial data, and semantic descriptions for GIServices and chains and discovering dependencies provides an informed understanding of geospatial provenance. Motivated by provenance demand from automatic GIService composition, our work distinguishes a three-level view of provenance. Such a view is important because it can increase understanding of the derivation history of geospatial data products. For example, in using alternative process models when checking knowledge provenance for re-planning services.

8. Conclusions and Future Work

Semantic Web and Web Service technologies have made it possible to automatically derive information and knowledge from geospatial data in an effective and timely way. In order for users to have an informed understanding of such automatically derived data products, the provenance of these data products has to be recorded. The paper proposes a three-level view of provenance. The provenance of the data, services, and knowledge levels increases understanding of provenance and could contribute to the evolution of GIService chains. The use of ontologies and rules allows for effective linking, querying, and inferring of provenance-related information entities. Provenance capture by the semantic execution engine is compliant with existing systems and fits well with the Semantic Web environment.

Semantic Web Service technologies have been used before to address the semantic heterogeneity of GIServices. Once domain semantics for GIServices are described using Semantic Web Services, provenance can be linked to geospatial data, services, workflows, and semantics for them using Semantic Web technologies. The links from provenance to other data items, in a machine-readable format, also provide possibilities for automated mining of provenance and related contents. While most previous work on provenance focuses on capturing sources and the process steps used in deriving data products, the work in this paper proposes to trace knowledge. The geoprocessing process models, encoded using process model ontologies from Semantic Web Service technologies, could be captured as a kind of domain knowledge, which also benefits geospatial knowledge sharing. They are linked to the provenance ontology. Similarly, geospatial metadata collected from the engine is also linked to enrich the provenance at the data level. The work also shows promise for linking more kinds of geospatial knowledge involved in the derivation history.

For example, the seasonality of NDVI will affect the analysis results. Some may want to capture the knowledge on how to choose appropriate temporal NDVI data for the landslide susceptibility use case. If such knowledge could be well-defined, it would be possible to link it to the provenance.

The work in this paper proposes a Semantic Web approach at various levels of the system: engine, knowledge, service, and data, to meet the needs of distributed geospatial data processing. It provides a comprehensive prototype system that integrates all these technologies, Semantic Web, provenance, and workflow, and makes it work in practice. However, there are still some issues that could be further investigated. The provenance and related contents in the RDF format, although machine-readable, should be visualized in a user-friendly way. Although semantic interoperability has been addressed before, semantic annotations to GIServices still need some extra effort. Since the benefits of explicit semantic descriptions are clear, tools for creating these semantic descriptions can be provided. Finally, while most current work focuses on capturing and presenting provenance, applications on the use of provenance will justify the benefits of provenance. For example, provenance applications in service composition can be developed by extending the legacy system for automatic service composition, allowing the re-planning of service chains at different phases using different levels of provenance, and creating a more flexible and complex system for service composition.

Author Contributions: Zhaoyan Wu contributed to the design of the experiment and manuscript writing. Hao Li analyzed the results, and contributed manuscript writing. Peng Yue contributed idea and manuscript writing. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Chongqing Technology Innovation and Application Development Project (No. cstc2021jscx-dxwtBX0023), and Hubei Provincial Natural Science Foundation of China (No. 2020CFA001).

Data Availability Statement: The ontology that support the findings in this study is available on Github at https://github.com/leehommlee/ProvenanceinGIServices (accessed on 6 March 2023).

Acknowledgments: We are grateful to Lianlian He for the discussion and assistance on the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gomes, V.; Queiroz, G.; Ferreira, K. An overview of platforms for big earth observation data management and analysis. *Remote Sens.* **2020**, *12*, 1253. [CrossRef]
2. Di, L.; McDonald, K. Next generation data and information systems for earth sciences research. In Proceedings of the First International Symposium on Digital Earth, Beijing, China, 29 November 1999; pp. 92–101.
3. Yue, P.; Gong, J.; Di, L.; He, L.; Wei, Y. Integrating semantic web technologies and geospatial catalog services for geospatial information discovery and processing in cyberinfrastructure. *GeoInformatica* **2011**, *15*, 273–303. [CrossRef]
4. McIlraith, S.A.; Son, T.C.; Zeng, H. Semantic web services. *IEEE Intell. Syst.* **2001**, *16*, 46–53. [CrossRef]
5. Sycara, K.; Paolucci, M.; Ankolekar, A.; Srinivasan, N. Automated discovery, interaction and composition of semantic web services. *J. Web Semant.* **2003**, *1*, 27–46. [CrossRef]
6. Yue, P.; Di, L.; Yang, W.; Yu, G.; Zhao, P. Semantics-based automatic composition of geospatial Web services chains. *Comput. Geosci.* **2007**, *33*, 649–665. [CrossRef]
7. Sun, Z.; Di, L.; Hao, H.; Wu, X.; Tong, D.Q.; Zhang, C.; Virgei, C.; Fang, H.; Yu, E.; Tan, X.; et al. CyberConnector: A service-oriented system for automatically tailoring multisource Earth observation data to feed Earth science models. *Earth Sci. Inform.* **2018**, *11*, 1–17. [CrossRef]
8. Foster, I. Service-oriented science. *Science* **2005**, *308*, 814–817. [CrossRef]
9. Miller, H.J. Geographic information science III: GIScience, fast and slow–Why faster geographic information is not always smarter. *Prog. Hum. Geogr.* **2020**, *44*, 129–138. [CrossRef]
10. Wang, S.; Wilkins-Diehr, N.R.; Nyerges, T.L. CyberGIS—Toward synergistic advancement of cyberinfrastructure and GIScience: A workshop summary. *Spat. Inf. Sci.* **2012**, *4*, 125–148. [CrossRef]
11. Yue, P.; Guo, X.; Zhang, M.; Jiang, L.; Zhai, X. Linked Data and SDI: The case on Web geoprocessing workflows. *ISPRS J. Photogramm. Remote Sens.* **2016**, *114*, 245–257. [CrossRef]
12. Yue, P.; Gong, J.; Di, L. Augmenting geospatial data provenance through metadata tracking in geospatial service chaining. *Comput. Geosci.* **2010**, *36*, 270–281. [CrossRef]

13. Yue, P.; Di, L.; Yang, W.; Yu, G.; Zhao, P.; Gong, J. Semantic web services based process planning for earth science applications. *Int. J. Geogr. Inf. Sci.* **2009**, *23*, 1139–1163. [CrossRef]
14. Zhuang, C.; Xie, Z.; Ma, K.; Guo, M.; Wu, L. A task-oriented knowledge base for geospatial problem-solving. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 423. [CrossRef]
15. Mehla, S.; Jain, S. Rule languages for the semantic web. In *Emerging Technologies in Data Mining and Information Security*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 825–834.
16. Rao, J.; Su, X. A survey of automated web service composition methods. In Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition, San Diego, CA, USA, 6 July 2004; pp. 43–54.
17. Norton, B.; Pedrinaci, C.; Domingue, J.; Zaremba, M. Semantic execution environments for semantics-enabled SOA. *Inf. Technol.* **2008**, *50*, 118–121.
18. Zhang, M.; Bu, X.; Yue, P. GeoJModelBuilder: An open source geoprocessing workflow tool. Open Geospatial Data. *Softw. Stand.* **2017**, *2*, 8.
19. Bowers, S.; Ludäscher, B. An ontology-driven framework for data transformation in scientific workflows. In Proceedings of the International Workshop on Data Integration in the Life Sciences, Leipzig, Germany, 25–26 March 2004; Springer: Berlin/Heidelberg, Germany; pp. 1–16.
20. Veregin, H.; Lanter, D.P. Data-quality enhancement techniques in layer-based geographic information systems. *Comput. Environ. Urban Syst.* **1995**, *19*, 23–36. [CrossRef]
21. Yue, P.; Gong, J.; Di, L.; He, L. Automatic geospatial metadata generation for Earth science virtual data products. *GeoInformatica* **2012**, *16*, 1–29. [CrossRef]
22. Joseki. Hewlett-Packard Labs Semantic Web Programme. 2009. Available online: http://www.joseki.org/ (accessed on 6 December 2009).
23. Zhao, P.; Di, L. (Eds.) *Geospatial Web Services: Advances in Information Interoperability*; Information Science Reference (IGI Global): Hershey, PA, USA, 2010; p. 552.
24. Bröring, A.; Maue, P.; Janowicz, K.; Nüst, D.; Malewski, C. Semantically-enabled sensor plug & play for the sensor web. *Sensors* **2011**, *11*, 7568–7605.
25. Athanasiou, S.; Georgomanolis, N.; Patroumpas, K.; Alexakis, M.; Stratiotis, T. TripleGeo-CSW: A Middleware for Exposing Geospatial Catalogue Services on the Semantic Web. In Proceedings of the EDBT/ICDT Workshops, Brussels, Belgium, 27 March 2015; pp. 229–236.
26. Prudhomme, C.; Homburg, T.; Ponciano, J.J.; Boochs, F.; Cruz, C.; Roxin, A.M. Interpretation and automatic integration of geospatial data into the semantic web. *Computing* **2020**, *102*, 365–391. [CrossRef]
27. Zhang, M.; Jiang, L.; Zhao, J.; Yue, P.; Zhang, X. Coupling OGC WPS and W3C PROV for provenance-aware geoprocessing workflows. *Comput. Geosci.* **2020**, *138*, 104419. [CrossRef]
28. Chebotko, A.; Lu, S.; Fei, X.; Fotouhi, F. RDFProv: A relational RDF store for querying and managing scientific workflow provenance. *Data Knowl. Eng.* **2010**, *69*, 836–865. [CrossRef]
29. Ornelas, T.; Braga, R.; David, J.M.N.; Campos, F.; Castro, G. Provenance data discovery through Semantic Web resources. *Concurr. Comput. Pract. Exp.* **2018**, *30*, e4366. [CrossRef]
30. Brown, C. Semantic web technologies for data curation and provenance. In Proceedings of the 19th International Congress of Metrology, Paris, France, 24–26 September 2019.
31. Sahoo, S.S.; Barga, R.; Sheth, A.P.; Thirunarayan, K.; Hitzler, P. PrOM: A Semantic Web Framework for Provenance Management in Science. Available online: http://corescholar.libraries.wright.edu/knoesis/445 (accessed on 5 May 2021).
32. He, L.; Yue, P.; Di, L.; Zhang, M.; Hu, L. Adding geospatial data provenance into SDI—A service-oriented approach. *IEEE Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *8*, 926–936. [CrossRef]
33. Golbeck, J.; Hendler, J. A semantic web approach to the provenance challenge. *Concurr. Comput. Pract. Exp.* **2007**, *20*, 431–439. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.