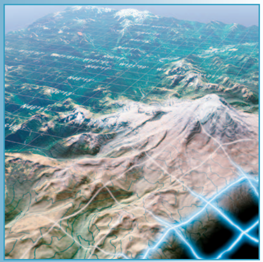
Web Services for GIS



Integrating Semantic and Syntactic Descriptions for Chaining Geographic Services

Integrating multiple geographic services from different information communities and spatiolinguistic regions is challenging because of its inherent complexity and heterogeneity. A geographic information systems workflow approach can use semantic and syntactic service descriptions to form service chains that can integrate service discovery, composition, and reuse. Service chaining links remote geographic services to help expert users form complex geoprocessing services and perform timely analysis of geodata. This method facilitates the use of XML-based service description languages to build a geoservice-reuse architecture based on common ontologies and shared service descriptions.

Rob Lemmens, Andreas Wytzisk, and Rolf de By
International Institute for Geo-Information Science and Earth Observation

Carlos Granell and Michael Gould
University Jaume I of Castellón

Peter van Oosterom
Delft University of Technology

The geographic information systems (GIS) field has been highly influenced by advances in Web services technologies. The result is a proliferation of specialized geographic services that, for example, help visualize vector cartographic data, locate a map view given a toponym (geographic name), and more recently, provide specific geoprocessing functions. Historically, desktop GIS workflow has required complex manual data sourcing and reformatting before arriving at even the simplest analysis, such as visualizing geodata themes in context to find spatial relationships between factories and schools. To avoid the common practice of downloading and processing massive data sets using traditional desktop GIS, the field is now challenged with finding a way to integrate multiple geographic services, each from a specific information community and spatiolinguistic region. Access to chains of remote geographic services promises more flexible, just-in-time analysis of geographic data that can be updated in situ.

In practice, however, chaining geographic services is nontrivial, mostly because geographic data have varied differences from other types of data. (See the “Why Geoinformation Is Special” sidebar for full details.)

In this article, we address the unresolved research topic of semantics and dynamic service chaining.[1] We've developed a methodology that combines ser-

 Published by the IEEE Computer Society 1089-7801/06/$20.00

Why Geoinformation Is Special

Five unique features distinguish geographic data from other types of scientific data:

- *Multiple versions*. Multiple versions of the same entities on the Earth's surface can differ radically in terms of data model, scale, data generalization, and the conceptual models the data collectors use. Important semantic differences are also involved in the data, which are mostly collected by different government agencies.
- *Implicit linking*. In general, explicit references must be present to combine information in a meaningful manner. However, geographic information enables linking without explicit references, but we can use implicit references (via coordinate reference systems). Some processing steps (packages as services) may be necessary to realize this. For example, one information source might only use city names (and not coordinates) so a gazetteer service must translate the city name into a geographic location. In another example, two sets of geoinformation might use coordinates but in a different reference system so a coordinate translation service is necessary.
- *Massive data sets*. Compared to general (administrative) information, geoinformation can be massive. Hence, in case of service chaining, we pay close attention where to process the different steps involved and be careful with shipping information. For example, one service can deliver a detailed route network and another can compute the most efficient path, an integrated service (or at least executed at the same node) that computes and delivers the shortest path is more efficient than shipping the relevant part of the network to another service (on another node in the network) and then computing the most efficient path from source to destination. In case of satellite imagery, the raster data volumes are also huge, so it might be far more efficient to do the image processing (object recognition) closer to the source and then ship the result.
- *Maps as implicit interfaces*. Everyone is familiar with reading maps, so they are a natural man-machine interface for the services interacting with the user and presenting (intermediate) results. Also zooming in and out is a natural way to show more or less data (or more or less aggregated) data. Services should exploit this natural advantage of geoinformation.
- *Geoinformation is geometry based*. Because geoinformation is geometry based, it's possible to apply a whole set of common mathematical tools in geoservices (such as to compute the distance between two objects or compute the buffer around an object) that weren't specifically developed for an application. Other information types might also use such nonapplication dependent tools (such as string manipulation and statistical operations), but with geoinformation, the set of these tools is even larger.

vice discovery, abstract composition (identifying service chain functionality with the help of conceptual parameters), concrete composition (controlling control and data flow among specific services), and execution. Semantics researchers are studying the first two, and the latter two are common syntactic research issues, but most approaches to geoservice chaining have addressed these issues superficially and separately. The strength of our approach lies in the way we combine syntactic and semantic service descriptions for chaining geographic services, letting us take advantage of the semantics composed in the service discovery process. In turn, we strive to reuse services by discovering existing annotated compositions, helping expert users rapidly create complex geoprocessing services.

RiskMap Service Scenario

Our integrated approach combines two independently developed applications to identify syntactic and semantic relations among possible components involved in geoservice chaining. GeoMatchMaker supports service discovery and abstract composition, and the Integrated Component Designer supports concrete composition and execution of services. Users submit queries based on a geographic semantic framework to GeoMatchMaker, which identifies appropriate candidate services (some of which might be service compositions). Then, the Integrated Component Designer combines semantic and syntactic descriptions to let users incrementally build a concrete service composition from the candidate services list. (See the "Related Work in Integrating Service Descriptions" sidebar for details on other research and approaches.)

A typical geographic service chaining scenario might involve planning for possible emergency situations. For example, the RiskMap service, which we've implemented using our approach, generates a map with the real-time locations of potentially hazardous substances, such as ammonia or explosives, and then center the map around a user-specified location.

Consider a scenario in which a service design-

Related Work in Integrating Service Descriptions

Current Open Geospatial Consortium Web Service Common Specification (OWS) efforts[1] are aligning basic geographic services such as the Web Map Service (WMS) and Web Feature Service (WFS) with the mainstream publish-find-bind paradigm represented with SOAP, Web Service Description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI). In this context, the recently released Web Processing Service (WPS) specification provides access to spatial operations, ranging from simple calculations to complex models by means of Web service interfaces exposing the parameters for data input, operation initialization, and data output,[2] but it provides no support for service chaining. Within OGC, service discovery is handled by a service registry that provides service metadata with details on service types, as defined in ISO 19119 (Services).[3]

Currently, no existing OGC specification deals with semantics in support of service (and data) discovery. The OGC Geo Semantic Web Interoperability Experiment (GSW IE)[4] has embarked on issuing semantic queries in relation to Web Feature Services (WFS), but not so much on discoverying services in general.

Udo Einspanier and his colleagues[5] have identified the need for the integrated use of syntax and semantics in service chaining. Other related research[6] addresses geographic ontology design, client interfacing, and reasoning in disaster management (although mainly on the semantic aspects of service chaining).

References

1. A. Whiteside, ed., *OGC Web Services Common Specification Version 1.0. OGC Implementation Specification*, OGC 05-008, Open Geospatial Consortium specification, 2005.
2. P. Schut, and S. Keens, eds., *Web Processing Service (WPS) Interoperability Experiment: Final report*, OGC Interoperability Experiment OGC 05-051, Open Geospatial Consortium, 2005.
3. *ISO 19119:2005, Geographic information – Services*, Int'l Organization for Standardization, 2005.
4. J. Lieberman, T. Pehle, and M. Dean, "Semantic Evolution of Geospatial Web Services," *Proc. W3C Workshop on Frameworks for Semantics in Web Services*, 2005; www.w3.org/2005/04/FSWS/Submissions/48/GSWS_Position_Paper.html.
5. U. Einspanier et al., "Toward a Process Model for GI Service Composition," *Proc. Münster GI-Days*, IfGI prints 18, 2003, pp. 31–46.
6. E. Klien, M. Lutz, and W. Kuhn, "Ontology-Based Discovery of Geographic Information Services: An Application in Disaster Management," *Computers, Environment and Urban Systems*, vol. 30, no. 1, 2006, pp. 102–123.

er must build such a service from smaller distributed services for an end user who will interact only with the composite service. Assume that the hazard information's geographical aspects are provided by a Web Map Service (WMS), as defined by the industry standardization body Open Geospatial Consortium (OGC; www.opengeospatial.org). The WMS needs a preceding service to provide it with a WMS `GetMap` request as a URL containing input parameters for specific geographic features (such as points representing hazardous sites) and the map view's *geographic extent* (the geographic area the map view spans). The service designer might determine this geographic extent indirectly by translating a toponym to its corresponding bounding box.

The service designer must provide a service chain that lets users enter a city name but that simultaneously shields them from the detailed WMS parameter construction. Figures 1a and 1b depict the elements of the RiskMap service chain and its output, respectively.

In this scenario, one of several *gazetteer services* (services that take a toponym as input and produce geographic coordinates) can handle resolving a city name and determining its location, and each service has its own geographic coverage, data resolution, and special semantic (in addition to syntactic) interface needs. The same is true for the other services in the chain. The key to the methodology we describe here is that a service designer might consider multiple service candidates at each of the four steps in Figure 1a. However, determining how one or another of two functionally similar candidates can interface with other services, in semantic and syntactic terms, is a nontrivial exercise. To address this challenge, we've developed an approach that provides concrete assistance in augmenting the semantic content of each service description and in discovering and combining services.

Semantic Framework

Our integrated approach adopts a semantic framework as a basis for semantic service descriptions, which support the discovery and abstract composition of geographic services. (See the "Service Descriptions" sidebar for background details on this topic.) Figure 2 shows our proposed framework, which consists of three formal ontologies grouped into an information and operational model:

- A *feature-concept ontology* formally defines the conceptualizations of real-world phenomena and the relationships between them. For

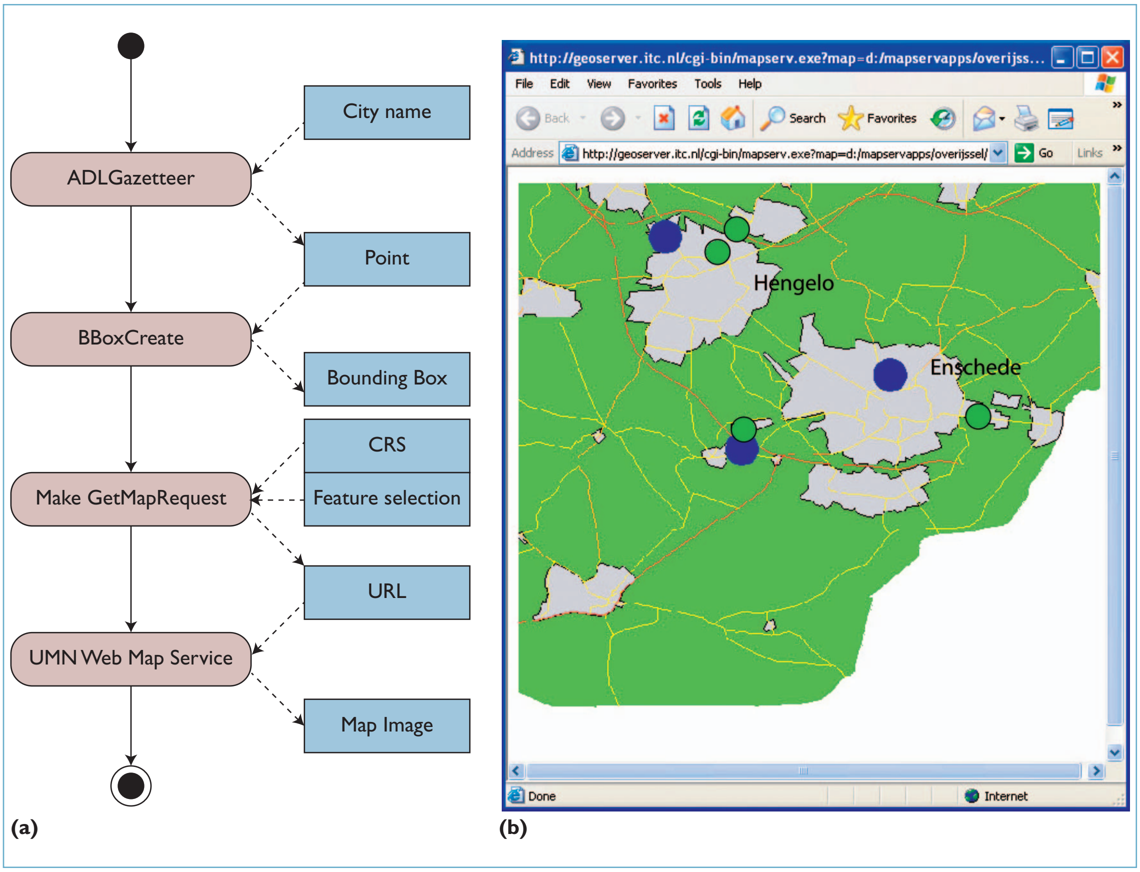


Figure 1. RiskMap service chain. (a)) UML sequence diagram representing the abstract service composition that accepts a city name and produces a map of nearby risks. (b) This sample output of one possible service chain gives a map showing the area around the city of Enschede with fireworks depots (small circles) and ammonia storage locations (large circles).

example, "building" is a feature type that's (partially) defined by its thematic and spatial attributes.

- A *feature-symbol ontology* formally defines the abstract elements that make up a feature in an object-field model, based on the ISO 19109 standard,[2] which distinguishes three abstraction levels: metalevel, application level, and data level.
- A *geo-operation ontology* formally defines operation types in terms of their behaviors and is based on the Web Ontology Language (OWL) based Web service ontology, OWL-S. Each type is characterized by the behavior of a well-known atomic GIS operation (inspired by the ISO 19119 service taxonomy[3]) and its typical input and output parameters.

We can represent semantic service metadata in the ontology with classes or class instances, which we call *individuals*. This is a (partial) class definition of a gazetteer operation (which serves as a candidate operation for our RiskMap chain) in a Description Logic (DL) axiom:

```
opera:LocSpat ⊑

  opera:AcrossAttributeTypes ⊓
  (∃∀ opera:hasInputPar. (∃ opera:has
   ParType.symbol:GF_Location
   AttributeType)) ⊓
  (∃∀ opera:hasOutputPar. (∃ opera:
    hasParType.symbol:GF_Spatial
    AttributeType))

with:
  ⊑   'is subclass of'
  ∃∀  conjunction of 'there exists at
    least one' and 'for all'
  ⊓  'intersected with'
  .  separator between role and role-
    filler
```

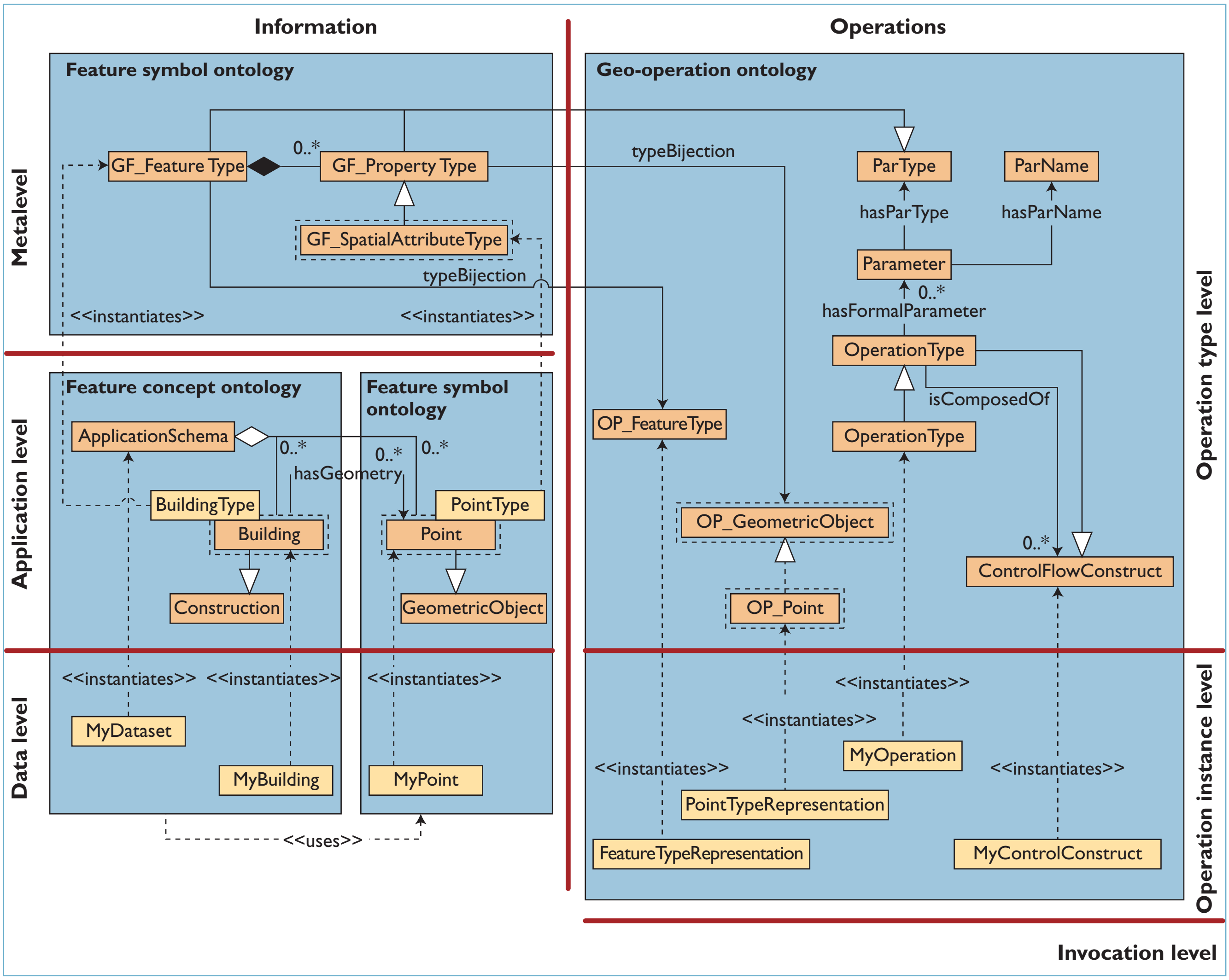


Figure 2. Proposed semantic framework for integrating geographic services. This Uniform Modeling Language class diagram depicts an overview of the three formal ontologies that form the proposed semantic framework for geoinformation and geo-operations.

The prefixes refer to the hosting ontology. `LocSpat` stands for the operation that reads a location attribute type (such as an address) and produces a spatial attribute type (such as a geometric object), a standard gazetteer operation. This definition describes the `LocSpat` operation type as a subclass of the `AcrossAttributeTypes` operation type and puts input and output restrictions on it. (We provide a more specific definition for the example Alexandria Digital Library [ADL] Gazetteer [http://middleware.alexandria.ucsb.edu/client/gaz/adl/index.jsp] later on.) The definition specifies that the gazetteer takes as input an address that consists of only a city name. The omitted `forall` quantifier means that the operation can also take other input types (but all are `GF_LocationAttributeType`). The output is of type `point`:

```
opera:ADLGazetteer ⊑

  opera:LocSpat ⊓
  (∃ opera:hasInputPar.(∃ opera:
     hasParType.(∃ opera:typeBijection.
     opera:OP_CityNameAddress))) ⊓
  (∃∀ opera:hasOutputPar.(∃ opera:
     hasParType.(∃ opera:typeBijection.
     opera:OP_Point)))
```

As an alternative, we can also represent this above operation with individuals that instantiate the concepts used in these class definitions. Both class and individual definitions are encoded in

Service Description

Service description standards for geographic services are evolving toward the use of general Web service standards, such as the Web Service Description Language (WSDL) for syntactic service descriptions[1] and the Web Ontology Language (OWL) for semantic service descriptions.[2] Annotation approaches have emerged as a way to bridge the gap between the syntactic and semantic worlds.

Syntax-Based Descriptions

WSDL is a widely accepted standard for describing Web service interfaces. During the discovery and composition phases of our approach, we focus on the abstract part of a WSDL description: operation and input–output messages. Implementation details are needed during the service execution. At that stage, we use the Oasis Web Services Business Process Execution Language (WSBPEL), which expresses how to invoke a set of Web services. Both specifications are expected to become recommendations under their respective committees (at W3C and Oasis), yet they treat Web services only at the syntactical level, which is necessary but insufficient for creating meaningful Web service descriptions.

Semantic-Based Descriptions

To improve semiautomatic discovery methods, services must be described with formal languages that allow for machine reasoning. A key role is played by machine ontologies, which are machine-accessible representations of conceptual models. OWL facilitates the creation of Web-based machine ontologies by drawing on the formal Description Logic theory, which has roots in first-order predicate logic and provides highly expressive concept-forming constructs.[3] OWL-S[2] is an *upper ontology*, (one that provides primitive concepts at a general level, spanning multiple application domains) based on OWL that models the characteristics of Web services and can be used to create semantically enriched Web service descriptions.

OWL-S provides three modelling constructs at the top level — that is, the service profile (what the service does), service grounding (how the service can be accessed), and service model (how to use the service in terms of semantic content, including its workflow). OWL-S provides classes that a service provider can instantiate to create specific service descriptions. Given that OWL-S is an upper ontology, it doesn't provide domain ontologies; information communities must establish these themselves.

Annotation approaches

Currently, two major approaches allow integration of syntactic and semantic descriptions: OWL-S grounding and WSDL-S.[4] OWL-S provides abstract constructs for input and output process parameters. It doesn't explicitly describe the concrete I/O messages, but rather specifies, in a so-called *grounding*, how they must be linked to parameters in a concrete message mechanism. The OWL-S specification version 1.1 uses WSDL as the grounding mechanism. For each OWL-S process, a mapping is created between each I/O parameter of the OWL-S process model and its corresponding target parameter in the WSDL document. Furthermore, OWL-S specifies other parameters, such as operation name and a URI, pointing to the actual WSDL document. Using an OWL-S processor such as the OWL-S virtual machine allows for the control of interaction between Web services, based on the combination of OWL-S process and grounding.[5]

WSDL-S annotates Web services by enriching WSDL descriptions, which otherwise lack semantic expressivity, with semantic tags (specifically the WSDL-S `modelReference` attribute for WSDL part and operation elements). WSDL-S suggests adding semantics to WSDL by using extensibility in the elements and attributes supported by the WSDL specification as well as permitting the relation between existing WSDL constructs and ontology concepts.

References

1. F. Curbera et al., "Unravelling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI," *IEEE Internet Computing*, vol. 6, no. 2, 2002, pp. 86–93.
2. D. Martin et al., "OWL-S: Semantic Markup for Web Services," W3C member submission, 22 Nov. 2004; www.w3.org/Submission/OWL-S/.
3. F. Baader et al., eds., *The Description Logic Handbook; Theory, Implementation and Applications*, Cambridge Univ. Press, 2003.
4. R. Akkiraju et al., "Web Service Semantics – WSDL-S," W3C member submission, 7 Nov. 2005; www.w3.org/Submission/WSDL-S/.
5. M. Paolucci et al., "A Broker for OWL-S Web Services", *Extending Web Services Technologies: The Use of Multi-Agent Approaches*, Springer, 2004, pp. 92–99.

OWL in the ontology, and they are stored in a knowledge base for reasoning purposes.

Integrated Architecture and Implementation

Figure 3 shows the integrated architecture for service chaining using syntactic and semantic descriptions. We assume that all participants share a set of common geo-ontologies (derived from the semantic framework), which service providers also use to annotate their services. The service discovery finds annotated services that the composition process directly consumes to build a concrete composition. As new compositions are published in the Web services repository, a service designer discovers single services as well as compositions, thus increasing service reuse.

Discovery and Abstract Composition

Geoservice discovery generally involves *match-*

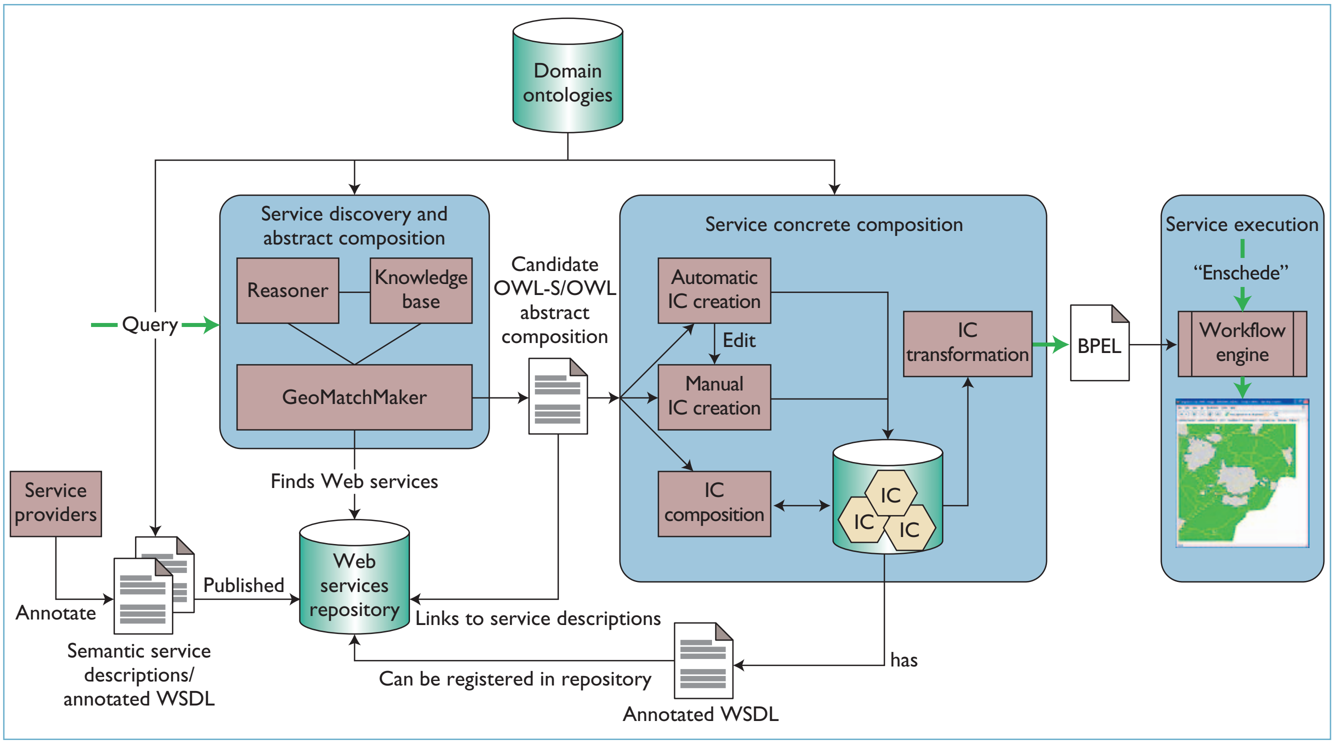


Figure 3. Integrated architecture for service chaining. The figure shows the architecture components and the service engineer's workflow (green arrows).

making, identifying service advertisements that might match a service request. Consider the service chain with n services: chain (S^1, ..., S^n). We seek cross-matches between a service's output parameters and a subsequent service's input parameters, and we evaluate the behavioral aspects of the combination with the help of reasoner software. When searching for a service S^{i+1} that follows a given service S^i, an ontological request R (representing the service S^i) is tested by the reasoner against an ontological advertisement A (representing a candidate service S^{i+1}). In ontologies, a *concept* (such as a Building) is interpreted as a set of *individuals* (such as the Louvre or Taj Mahal). When ontologies are materialized as knowledge bases, concepts and their relationships are separated from the individuals. They are contained, respectively, in the *terminology box* (TBox), which holds declarations of concepts, and the *assertions box* (ABox), which contains assertions specific to individuals (instances of the concepts).[4] (These terms have no relationship with a map server's bounding box parameter [BBox].)

Depending on whether we use concept- or individuals-based definitions of the operations, there are four possible ways to perform the matchmaking:

1. match ($R_{out}(S^i)$, $A_{in}(S^{i+1})$), (between concepts)
2. match ($R_{out}(S^i)$, $a_{in}(S^{i+1})$), (between concept R and individuals a)
3. match ($r_{out}(S^i)$, $A_{in}(S^{i+1})$), (between individuals r and concept A)
4. match ($r_{out}(S^i)$, $a_{in}(S^{i+1})$), (between individuals)

Concepts are denoted with upper case, and individuals with lower case.

Match type 1 involves concept descriptions only. This is done by TBox reasoning. Match types 2, 3, and 4 are performed with individuals by ABox reasoning. (An earlier work discusses the differences between TBox and ABox reasoning in this context.[5]) In our current GeoMatchMaker prototype, we've opted for type 2 matches because the formulation of advertisements (which the service designer does) and interpreting the results is more straightforward than for the other match types.

We use the RacerPro knowledge-representation system to reason with ontologies (www.racer-systems.com). The RacerPro reasoner (see Figure 3) performs the matching by inferring all candidate individuals a in the knowledge base that instantiate R. RacerPro can directly read OWL documents and represent them as TBoxes and

ABoxes in DL knowledge bases. Through a Java API, called JRacer, it provides numerous functions for managing the knowledge base and reasoning with its TBoxes and ABoxes. We used a small subset in the GeoMatchMaker kernel to provide reasoning capabilities.

For brevity, we elaborate here only on the search for a two-service chain (for which we selected the ADL Gazetteer service as the first service). Figure 4 shows the results in terms of a set of matching services. These services create a bounding box around the geometric point generated by the gazetteer. The service designer further evaluates them by refining the requesting concept until only one service is left. After the designer selects the BBoxCreate service, only one service, which must build a GetMapRequest from the bounding box, is left to complete the chain. Information, such as feature selection and coordinate-system metadata, which are needed by the GetMapRequest, are also added by the service designer in this service.

The GeoMatchMaker prototype integrates the Protégé ontology editor (http://protege.stanford.edu) and provides an interactive environment to compose the service chain. For execution purposes, he prototype can export the chain in different forms, such as an OWL-S document, which supports nine control-flow patterns. Figure 5 shows the structure of the service chain modeled as an OWL-S graph of individuals. The boxes represent instances of OWL-S *process* concepts. Among them are the discovered geo-operations (`ADLGazetteer`, `BBoxCreate`, and `MakeGetMapRequest`) and supporting control constructs (`Sequence`, `Perform`, and so on). The sequence pattern is the result of following the first-rest control flow and is portrayed as a Uniform Modeling Language (UML) activity in Figure 1a.

Concrete Composition and Execution

The OGC and the International Organization for Standardization (ISO) technical committee for Geographic Information and Geomatics (ISO/TC211; www.isotc211.org) have defined three design patterns for geographic service composition according to the degree of transparency of the Web service chain's complexity to the client:[3]

- *transparent*, or user-defined chaining;
- *translucent*, or workflow-managed chaining, and
- *opaque*, or aggregate service chaining.

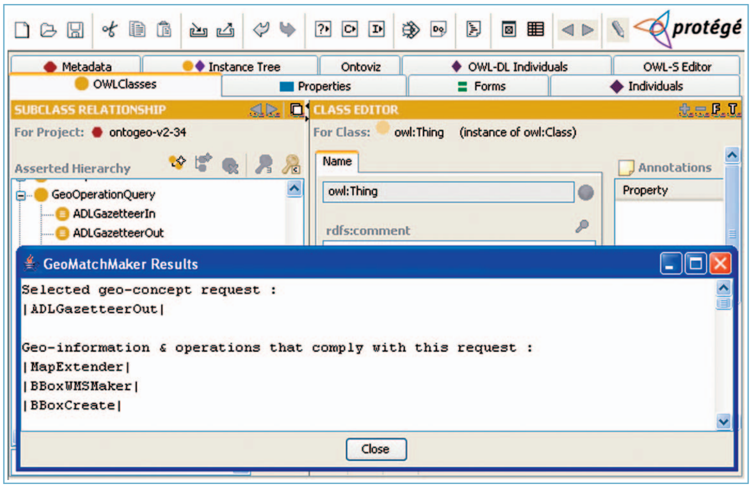


Figure 4. GeoMatchMaker prototype output (discovery part). The background shows the Protégé ontology editor interface, and the foreground includes a window that shows potential operations that are found to have an input that match the ADLGazetteer operation's output.

As the name suggests, translucent chaining is midway between transparent and opaque chaining, offering balanced benefits compared with the other two patterns.[1]

Our composition approach relies on translucent chaining to reduce the design complexity of geographic service chains to the user by using *integrated components*[6] as the fundamental building blocks for service composition. The idea consists of creating an integrated component from a set of candidate geographic Web services with the same functionality. For instance, an integrated component for Web mapping might comprise several concrete Web mapping services, improving the chain flexibility because several Web mapping services are available for carrying out the integrated component's functionality. Next, users create more complex and heterogeneous integrated components by reusing simpler integrated components already available in catalogs. Each new integrated component hides the complexity of (encapsulates) the contained integrated components' functionality.

Two interfaces control the access to an integrated component: the public interface openly expresses an integrated component's functionality (described in the Web Service Description Language for Web Service Semantics [WSDL-S]), whereas the private interface encapsulates how an integrated component performs its functionality. For example, the following code snippet shows some features of WSDL-S to semantically annotate operations and parameters for the Gazetteer inte-

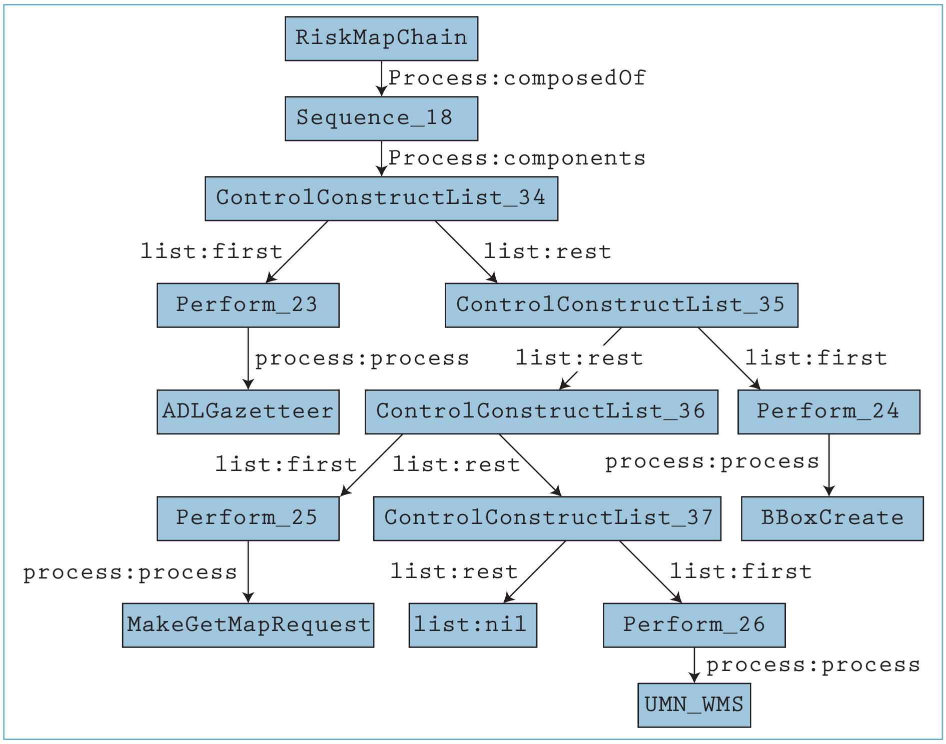


Figure 5. The RiskMap service chain structure. As a result of discovery and abstract composition, GeoMatchMaker produces an OWL-S document with this structure, representing the chain's sequence pattern.

grated component (public interface):

```
<wsdl:message name="getMsgResponse">
  <wsdl:part name="coordinates"
     element="xsd1:ResponseType"
     wssem:modelReference=
     "Ontology0#Point"/>
</wsdl:message>
<wsdl:message name="getMsgRequest">
  <wsdl:part name="name"
     element="xsd1:RequestType"
     wssem:modelReference=
     "Ontology0#CityName"/>
</wsdl:message>
  <wsdl:portType name="Gazetteer">
  <wsdl:operation name="getCoordinates"
     wssem:modelReference=
     "Ontology0#LocSpat"/>
</wsdl:portType>
```

The annotation (by the WSDL-S `modelReference` attribute) for the `getCoordinates` operation refers to the `LocSpat` concept in the geo-operation ontology, which formally defines an operation that returns a spatial attribute type, based on a location. WSDL `part` tags are annotated in the same manner.

In terms of encapsulation and providing integrated services of geospatial information, the notion of an integrated component has similarities with the translucent chaining pattern we described earlier. Once an integrated component meets certain user requirements, the IC Transformation process (see Figure 3) transforms its description into an executable Oasis Web Services Business Process Execution Language (WSBPEL)[7] process document, which contains concrete, executable geographic Web services.

Figure 3 illustrates the composition (center) and execution (right-hand side). Service discovery produces an OWL-S document that contains an abstract chain — that is, a list of appropriate Web services for composition (Figure 5). The link between service discovery and concrete composition consists of creating integrated components from such a list. For that, we offer three different possibilities (see Figure 3). The first automatically creates the corresponding integrated component from a WSDL-S description. Given a WSDL description, the second possibility lets users manually generate a new integrated component by annotating it with the concepts taken from shared geo-ontologies. In both cases, a new integrated composition is created from existing Web services. Yet, because one of our goals is to improve service reuse, the service discovery can also identify existing compositions (seen as integrated components) to be used in new compositions. In this third case, the creation process is unnecessary because the integrated component already exists. The composition process then uses composition patterns to construct complex integrated components by incrementally reusing existing ones taken from the repository.[6]

Figure 6 shows a screenshot of the Integrated Component Designer applied to the RiskMap scenario. This software tool is a set of Eclipse Plug-ins (www.eclipse.org) developed in Java. Figure 6 shows the graphical editor for defining the private interface of the RiskMap integrated component (represented by the `getRiskMap` function). This component combines and reuses two other integrated components already available — `LocationAttrToBox` and `UMN_WebMapService` — by using the composition pattern `sequence` (see the red box in Figure 6). Both `LocationAttrToBox` and `UMN_WebMapService` are themselves integrated components. The former contains the first two services in the abstract chain, `ADL Gazetteer` and `BBoxCreate`, forming an intermediate composition that takes a city name as input and produces a bounding box. The latter integrates the last two services, `MakeGetMapRequest` and `UMN MapServer`, encapsulating a full `GetMap` request to retrieve the final map image.

The user might execute a given composition through the transformation process (see the "IC transformation" box in Figure 3). This process serializes the integrated component description representing our RiskMap composition into a WSBPEL process document. Figure 3's right-hand side shows the service execution, which takes the WSBPEL process and produces the risk map.

We have tested the resulting WSBPEL process in the Oracle BPEL Process Manager (www.oracle.com/technology/products/ias/bpel/). Although the resulting WSBPEL code is more verbose than one created manually, it's fully compliant with to the standard to be executed by WSBPEL-compliant workflow engines. In our implementation experiences, we implemented the WSDL-S approach with less effort than the OWL-S grounding. Although OWL-S supports the whole range of discovery-composition chaining, fewer enactment engines are available for it compared to other standards such as WSDL and WSBPEL. From a practical point of view, a hybrid solution is therefore still preferable.

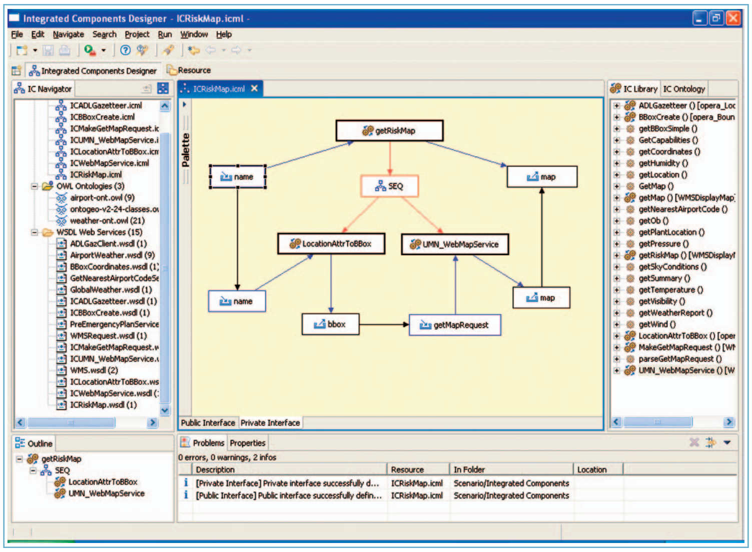


Figure 6. Snapshot of the target composition getRiskMap. Red-bordered boxes denote composition patterns, bold black-bordered boxes are integrated components, and black single-bordered boxes are input and output parameters.

One of the strengths of the integrated approach we present here is the use of common ontologies for the different steps in geographic service chaining. Web-based ontologies provide a formal yet flexible mechanism to describe Web services. Unfortunately, our GeoMatchMaker prototype supports only semiautomatic, human-controlled discovery. Another limitation lies in the exchange of workflow information between the prototypes. Currently, no single common format exists that holds workflow elements, ontology concepts, and WSDL parameters. However, this can be implemented with a relatively simple stylesheet transformation allowing, in this case, the reuse of existing compositions that are already annotated semantically in a semiautomatic way.

Our future work will look into methods for automatically aligning and maintaining ontologies that different information communities use (ontology mapping) and at developing more sophisticated algorithms for service metainformation propagation and result ranking. On the other hand, context-aware service composition and discovery should be included in the integrated prototype, letting users discover and compose services according to their location or preferences.

Acknowledgments

This work has been partially supported by the European Union Aware Project SST4-2004-012257.

References

1. N. Alameh, "Chaining Geographic Information Web Services," *IEEE Internet Computing*, vol. 7, no. 5, 2003, pp. 22–29.
2. *ISO 19109:2005, Geographic information – Rules for application schema*, Int'l Organization for Standardization, 2005; www.iso.org. >
3. *ISO 19119:2005, Geographic information – Services*, Int'l Organization for Standardization, 2005; www.iso.org
4. F. Baader et al., eds., *The Description Logic Handbook; Theory, Implementation and Applications*, Cambridge Univ. Press, 2003.
5. R. Lemmens and M. de Vries, "Semantic Description of Location Based Web Services using an Extensible Location

Ontology," *Proc. Münster GI-days 2004: Geoinformation and Mobility*, Univ. of Münster, IfGI prints 22, 2004, pp. 261–276.

6. C. Granell et al., "Improving Reuse of Web Service Compositions," *E-Commerce and Web Technologies*, LNCS 3590, Springer Verlag, 2005, pp. 358–367.
7. OASIS WSBPEL TC, "Web Services Business Process Execution Language Version 2.0," Committee Draft 17 May 2006; www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel

Carlos Granell is a researcher at the University Jaume I of Castellón, Spain. His research interests include interoperability in GIS and Web service reuse and composition integrated in spatial data infrastructures. He has a PhD in computer science from the Universitat Jaume I. Contact him at carlos.granell@uji.es.

Rob Lemmens is an assistant professor at the International Institute for Geo-Information Science and Earth Observation and is pursuing his PhD on semantic interoperability of distributed geoservices. His research activities focus on interoperability issues in spatial data infrastructures and application development, based on ISO and Open Geospatial Consortium specifications. Lemmens has an MSc in geodesy from Delft University of Technology. Contact him at lemmens@itc.nl.

Michael Gould is a senior lecturer in information systems at the University Jaume I of Castellón, Spain. His research interests include spatial data infrastructures and Web services interoperability. He received the PhD in geographic information systems from the NCGIA State University of New York at Buffalo . Contact him at gould@uji.es.

Andreas Wytzisk is an assistant professor at the International Institute for Geo-Information Science and Earth Observation. His research activities focus on interoperability issues in spatial data infrastructures, distributed simulations, and sensor webs. Wytzisk has a PhD in geoinformatics from the University of Münster, Germany. Contact him at wytzisk@itc.nl.

Rolf de By is an associate professor at the International Institute for Geo-Information Science and Earth Observation in the geoinformation processing department. His research interests are in designing large and advanced information systems that handle geospatial data, spatial database technology and methods, and novel applications. De By has a PhD in computer science from the University of Twente. Contact him at deby@itc.nl.

Peter van Oosterom is a professor at the Delft University of Technology and head of the GIS Technology section. He is European editor for the international journal *Computers, Environment, and Urban Systems*. Van Oosterom has a PhD in mathematics and natural sciences from Leiden University. Contact him at P.J.M.vanOosterom@tudelft.nl.