Integrating semantic and syntactic descriptions for chaining geographic services

Rob Lemmens¹, Carlos Granell², Andreas Wytzisk¹, Rolf de By¹, Michael Gould², Peter van Oosterom³

¹ International Institute for Geo-Information Science and Earth Observation (ITC)
Enschede, The Netherlands
² Universitat Jaume I (UJI)
Castellón, Spain
³ Delft University of Technology
Delft, The Netherlands

ABSTRACT

Accelerating the development of complex and heterogeneous geographic services requires improved methods that integrate service discovery, composition, and reuse. We present an integrated use of semantic and syntactic service descriptions for service chaining, by combining an application that supports service discovery and abstract composition, with another that supports concrete composition and execution of services. This facilitates the use of XML-based service description languages for building a geo-service-reuse architecture based on common ontologies and shared service descriptions.

KEYWORDS: semantic annotation, service reuse, geographic services description, discovery and composition methodology

The field of Geographic Information Systems (GIS) has been highly influenced by advances in web service technology. The resulting proliferation of specialized geographic services (e.g., for visualizing vector cartographic data, locating a map view given a toponym, and more recently, specific geo-processing functions) has created an interesting challenge: integrating multiple geographic services, each from a specific information community and spatio-linguistic region. The objective is to produce viable alternatives to the common practice of downloading and geo-processing massive datasets using traditional desktop GIS. Desktop GIS workflow historically has required complex manual data sourcing and reformatting before arriving at even the simplest analysis such as visualization of geodata themes in context (e.g., spatial relationship between factories and schools). Access to chains of remote geographic services promises more flexible, just-in-time analysis of geographic data that is updated in situ, however in practice the chaining of geographic services is non-trivial. Geographic data are special in that multiple versions of the same entities on the earth’s surface can differ radically in terms of data model, scale, generalization of data, and the conceptual models used by the data collectors. Also, most geographic data are collected by different government agencies, and therefore important semantic differences are also found at administrative borders at all levels. Moreover, geographic data objects may have multiple geometric and/or graphic representations, depending on the service type and on the client accessing them.

Alameh [1], in her conceptual description of geographic service chaining, ends by highlighting unresolved research topics such as semantics and dynamic service chaining. Here we show concrete progress in a methodology that combines service discovery, abstract composition (identifying service chain functionality), concrete composition (controlling data flow) and execution. The first two are being studied by semantics researchers and the latter two are common syntactic research issues, however most approaches to geo-service chaining have addressed semantic and syntactic issues superficially and separately.

Here an integrated approach identifies syntactic and semantic relations among possible components involved in geo-service chaining, by combining two independently developed applications. One supports service discovery and abstract composition (‘GeoMatchMaker’), the other concrete composition and execution of services (‘Integrated Component Designer’). Users submit queries based on a geographic semantic framework to the GeoMatchMaker, which identifies appropriate candidate services (some of which may be service compositions). Then semantic and syntactic descriptions are combined by the Integrated Component Designer to permit incremental building of a concrete service composition from the candidate services list. This article discusses the various steps in the integrated approach, for application in typical scenarios involving geographic information services.

Service Description Background (possibly as sidebar)

Service description standards for geographic services are evolving toward the use of general web service standards, such as WSDL for syntactic service descriptions [2] and OWL-S for semantic service descriptions [3]. Annotation approaches have emerged as a way to bridging the gap between the syntactic and semantic worlds.
Syntax-based descriptions
Web Service Description Language (WSDL) is a widely accepted standard for describing web service interfaces. During discovery and composition phases we focus on the abstract part of a WSDL description - operation and input/output messages. Implementation details will be needed during the service execution. At that stage, we make use of OASIS Web Services Business Process Execution Language (WSBPEL), that expresses how a set of web services are to be invoked. Both specifications are expected to become recommendations under their respective committees (W3C and OASIS), yet they treat web services only at the syntactical level, necessary but insufficient for creating meaningful descriptions of web services.

Semantic-based descriptions
To improve semi-automatic discovery methods, services have to 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. The Web Ontology Language (OWL) is a recommended specification of W3C that facilitates the creation of (web-based) machine ontologies. OWL draws upon the formal theory of Description Logics, which has roots in first-order predicate logic and provides highly expressive concept-forming constructs [4]. OWL-S [3] is an upper ontology 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, i.e., the service profile (what the service does), the service grounding (how the service can be accessed) and the service model (how to use the service in terms of semantic content, including its workflow). OWL-S provides classes that can be instantiated by a service provider to create specific service descriptions. Because OWL-S is an upper ontology, it obviously does not provide domain ontologies. These must be established by information communities themselves.

Annotation approaches
At present, the integration of syntactic and semantic descriptions is provided by two major approaches: OWL-S grounding and WSDL-S [5]. OWLS-S provides abstract constructs for input and output parameters of processes. It does not 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. In the OWL-S specification version 1.1, WSDL is used 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, other parameters, such as operation name and a URI, pointing to the actual WSDL document, are specified. The use of 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 [6].

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, and permitting the relation between existing WSDL constructs and ontology concepts.

RiskMap service scenario
A typical geographic service chaining scenario might involve planning for possible emergency situations, such as in the following example, called ‘RiskMap’ service. This service should generate a map with the real-time locations of potentially hazardous substances such as ammonia or explosives, and then centre the map around a user-specified location (“hazardous substances near my city”).

Consider a scenario in which a service engineer has to build the above service from smaller distributed services for an end user who only interacts with the composite service. Assume that the geographical aspects of the hazard information are provided by a Web Map Service (WMS) as defined by the OpenGeospatial Consortium (OGC; see http://www.opengeospatial.org). A WMS GetMap request is formulated by a URL containing input parameters for specific geographic features (e.g. points representing hazardous sites) and the geographic extension of the map view. This geographic extension may be determined indirectly by translating a toponym (using a gazetteer service) to its corresponding bounding box (via the BBOX service).

The service engineer is tasked to provide a service chain that allows the user to enter a city name and that shields him/her from the detailed WMS parameter construction. The elements of the RiskMap service chain and its output are depicted in respectively figures 1a and b.
In this scenario the resolution of a city name and determination of its location may be handled by one of several gazetteer services, each of which having 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. Key to the methodology described here, is that multiple service candidates may be considered 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 non-trivial exercise. The approach described here 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 to support the discovery and abstract composition of geographic services. Figure 2 shows the proposed semantic framework, which is composed of three formal ontologies grouped into information and operational model:
Figure 2: UML class diagram, depicting an overview of the ontologies as part of the proposed semantic framework for geo-information and geo-operations.

- A feature concept ontology formally defines the conceptualisations of real world phenomena and the relationships between them. For example, ‘Building’ is a feature type that is (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 ISO19109 standard [7]. This model distinguishes three abstraction levels, i.e., meta-level, implementation level and data level.
- A geo-operation ontology formally defines operation types in terms of their behaviour and is based on OWL-S. Each type is characterised by the behaviour of a well-known atomic GIS operation (inspired by the ISO19119 service taxonomy [8]) and its typical input and output parameters.

Information and operation model
Semantic service metadata can be represented in the ontology by classes or by individuals (class instances). A (partial) class definition of a gazetteer operation (which serves as a candidate operation for our RiskMap chain) in a Description Logic axiom is shown below. The prefixes refer to the hosting ontology. ‘LocSpat’ stands for the operation that reads a location attribute type (e.g., an address) and produces a spatial attribute type (e.g., a geometric object), a standard gazetteer operation.

```
opera::LocSpat β

(opera::AcrossAttributeTypes 6
 ( (⋯ opera::hasInputPar. ( ( opera::hasParType.symbol:GF_LocationAttributeType)) 6
 (⋯ opera::hasOutputPar. ( ( opera::hasParType.symbol:GF_SpatialAttributeType))

with:

β ‘is subclass of’
|⋯ conjunction of ‘there exists at least one’ and ‘for all’
6 ‘intersected with’
. separator between role and role-filler
```
The above definition describes the ‘LocSpat’ operation type as subclass of the ‘AcrossAttributeTypes’ operation type and puts input and output restrictions on it. A more specific definition is created for an example gazetteer, the Alexandria Digital Library (ADL) Gazetteer (http://middleware.alexandria.ucsb.edu/client/gaz/adl/index.jsp). The definition specifies that the gazetteer takes as input an ‘address’ that only consists of a city name. The omitted ‘forall’ quantifier means that the operation can also take other input types (but they are all GF_LocationAttributeType). The output is of type ‘point’.

\[
\text{opera:ADLGazetteer} \beta \\
\text{opera:LocSpat} 6 \\
( | \text{opera:hasInputPar.} ( | \text{opera:hasParType.} ( | \text{opera:typeBijection.opera:OP.CityNameAddress))) 6 \\
( | \ldots \text{opera:hasOutputPar.} ( | \text{opera:hasParType.} ( | \text{opera:typeBijection.opera:OP.Point])))
\]

Another representation can be given with so-called ‘individuals’ that instantiate the concepts used in these class definitions. Both class and individual definitions are encoded in 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 a set of common geo-ontologies (derived from the semantic framework) is shared by all participants. Also, service providers annotate their services using such geo-ontologies. The service discovery finds annotated services that are directly consumed by the composition process to build a concrete composition. As new compositions are published in the web services repository, not only single services are discovered but also compositions, thus increasing service reuse.

**Discovery and abstract composition**

Geo-service discovery in general involves the identification of service advertisements that may match a service request, which we refer to as matchmaking. Consider the service chain with \(n\) services:

\[
\text{chain}\ (S_1, ..., S_n)
\]

We seek cross-matches between the output parameters of a service and the input parameters of a subsequent service, and evaluate the behavioural aspects of the combination. For 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 against an ontological advertisement \(A\) (representing a candidate service \(S_{i+1}\)). In ontologies, a concept (e.g., ‘Building’) is interpreted as a set of individuals (e.g., ‘Louvre’, ‘Taj Mahal’, etc.). When ontologies are materialised as knowledge bases, concepts and their relationships are separated from the individuals. They
are contained in the so called TBox, respectively ABox\(^1\). The TBox (T stands for ‘Terminology’) holds declarations of concepts and the ABox contains assertions (hence the term ‘ABox’), specific to individuals (instances of the concepts) [4].

Depending on whether we use concept-based or individuals-based definitions of the operations, there are four possibilities to perform the matchmaking. Concepts are denoted with upper case, individuals with lower case.

- Match type I involves concept descriptions only. This is done by TBox reasoning. Match types II, III, IV are performed with individuals by ABox reasoning. Differences between TBox and ABox reasoning in the context of this paper have been discussed in [9]. In our current GeoMatchMaker prototype, we have opted for type II matches, because the entry of advertisements and the interpretation of the results are more straightforward than for the other match types. The matching has been performed with the (RacerPro, www.racer-systems.com/) reasoner (see Figure 3) by inferring all candidate individuals \(a\) in the knowledge base that instantiate \(R\).

RacerPro is a knowledge representation system that can be used for reasoning with ontologies. It can directly read OWL documents and represent them as TBoxes and ABoxes in DL knowledge bases. Through a Java API, called JRacer, RacerPro provides numerous functions for managing the knowledge base and reasoning with its TBoxes and ABoxes. A small subset has been used in the kernel of GeoMatchMaker to provide reasoning capabilities.

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

![Figure 4 (a): Output of the GeoMatchMaker prototype (discovery part).](image)

\(^1\) These terms have no relationship whatsoever with the term BBox (bounding box parameter of a map server)
Figure 4 (b): The RiskMap service chain structure as a result of discovery and abstract composition.

The GeoMatchMaker prototype integrates the Protégé ontology editor (http://protege.stanford.edu/) and provides an interactive environment to compose the service chain. The chain can be exported for execution purposes in different forms, such as an OWL-S document, which supports nine control flow patterns. Figure 4b shows the structure of the service chain modelled as an OWL-S graph of individuals. The boxes represent instances of OWL-S process concepts. Amongst them are the discovered geo-operations (ADLGazetteer, BBoxCreate, MakeGetMapRequest) and supporting control constructs (Sequence, Perform, etc). The sequence pattern can be recognised by following the ‘first-rest’ control flow and is portrayed as a UML activity in Figure 1a.

Concrete composition and execution

The Open Geospatial Consortium, and the ISO technical committee for Geographic Information and Geomatics (ISO TC211, see http://www.isotc211.org/) have defined three design patterns for geographic service composition according to the degree of transparency of the web service chain complexity to the client [8]: transparent or user-defined chaining, opaque or aggregate service chaining, and translucent or workflow-managed chaining. As the name suggests, translucent chaining is midway between transparent and opaque chaining, offering balanced benefits as compared with the other two patterns [1].

Our concrete composition approach relies on the translucent chaining to reduce the complexity of design of geographic service chains to the user by means of the notion of integrated component [10], which is the fundamental building block 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 may 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 available already in catalogues. Each new integrated component encapsulates the functionality of the contained integrated components. Two interfaces control the access to an integrated component: the public interface openly expresses an integrated component’s functionality (described in WSDL-S), whereas the private interface encapsulates how an integrated component performs its functionality. For example, the snippet code below shows some features of WSDL-S to semantically annotate operations and parameters for the Gazetteer integrated component (public interface). The annotation (by the WSDL-S modelReference attribute) for the operation getCoordinates refers to the concept LocSpat 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.

```xml
<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#Point"/>
</wsdl:message>
```
The notion of integrated component in terms of encapsulation and of providing integrated services of geospatial information has similarities with the translucent chaining pattern described previously. Once an integrated component meets certain user requirements its description is thus transformed into an executable WSBPEL process document, which actually contains concrete and executable geographic web services.

The centre and right hand side of Figure 3 shows the concrete composition and execution. Service discovery produces an OWL-S document that contains an abstract chain, i.e., a list of appropriate web services (or compositions) for composition (Figure 4b). The link between service discovery and concrete composition consists of creating integrated components from such a list. For that, we offer three different possibilities (Figure 3). The first one automatically creates the corresponding integrated component from a WSDL-S description (Automatic IC Creation box, Figure 3). Given a WSDL description, the second possibility allows users to 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, as one goal is to improve service reuse, the service discovery can also discover existing compositions (seen as integrated components) to be used in new compositions. In this third case, the creation process is not necessary because the integrated component already exists. The composition process (IC Composition box, Figure 3) then constructs, using composition patterns, complex integrated components by incrementally reusing existing ones taken from the repository [10].

Figure 5 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 5 shows the graphical editor for defining the private interface of the RiskMap integrated component (represented by getRiskMap function). This component combines (reuses) two other
integrated components already available –LocationAttrToBox and UMN WebMapService-- by using the composition pattern sequence (red box in Figure 5). Each of them in itself is a composition. 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 (IC transformation box, Figure 3). This process serialises the integrated component description representing our RiskMap composition into a WSBPEL process document. The right hand side of Figure 3 shows the service execution, which takes the WSBPEL process and produces the risk map. We have tested the resulting process in the Oracle BPEL Process Manager (www.oracle.com/technology/products/ias/bpel/index.html).

Related Work (possibly as sidebar)
Current OGC Web Service Common Specification (OWS) efforts [11] within the Open Geospatial Consortium are aligning basic geographic services such as Web Map Service (WMS) and Web Feature Service (WFS) with the mainstream publish-find-bind paradigm represented with SOAP, WSDL and UDDI. In this context, the recent 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 initialisation and data output [12], however without service chaining support. 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) [8]. Currently there is no OGC specification that deals with semantics in support of service (and data) discovery. An attempt has been made in the OGC Geo Semantic Web Interoperability Experiment (GSW IE) [13], however with very limited results. Einspanier et al. [14] identify the need for the integrated use of syntax and semantics in service chaining. Other related research (although mainly on the semantic aspects of service chaining) is reported by Klien et al., [15] which addresses geographic ontology design, client interfacing and reasoning in the application field of disaster management.

Conclusion
One of the strengths of the presented integrated approach 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. Our GeoMatchMaker prototype does not support automatic discovery, but rather semi-automatic (human controlled) discovery. Another limitation lies in the exchange of workflow information between the prototypes. Currently, there is no single common format that holds workflow elements, ontology concepts and WSDL parameters. However, this can be implemented by a relatively simple style sheet transformation allowing, in this case, the reuse of existing compositions that are already annotated semantically in a semi-automatic way. From our implementation experiences, the WSDL-S approach has been implemented with less effort than the OWL-S grounding. Although OWL-S supports the whole range of discovery-composition-chaining, there are fewer enactment engines for it, compared to other standards such as WSDL and WSBPEL. From a practical point of view, a hybrid solution is therefore still preferred.

The strength of the approach described here lies in the way syntactic and semantic service descriptions are combined for chaining geographic services, allowing us to take advantage of the composition of the semantics used in the service discovery process and, in turn, permitting service reuse by discovering existing annotated compositions. Indeed, reuse becomes an important point in the concrete composition process because it is essential to rapidly create complex geo-processing services. Also, reusable services and compositions offer developers service description reuse and also knowledge reuse as borrowed from previous solutions and experiences applied to similar problems, of special interest in geographic applications where multiple users work in and study the same geographical region.

Acknowledgements
This work has been partially supported by the EU AWARE project SST4-2004-012257.

References


Carlos Granell is currently a PhD candidate at the Universitat Jaume I, Castellón, Spain. He holds a MSc in Computer Engineering from the same university. His research interests focus on the interoperability on GIS, and web service reuse and composition integrated in Spatial Data Infrastructure (SDI). Contact: carlos.granell@lsi.uji.es

Rob Lemmens is assistant professor at ITC. His research activities focus on interoperability issues in spatial data infrastructures and application development, based on ISO and Open Geospatial Consortium (OGC) specifications. Lemmens holds an MSc in Geodesy from Delft University of Technology and is currently pursuing his PhD on semantic interoperability of distributed geo-services. Contact: lemmens@itc.nl.

Michael Gould is a senior lecturer in Information Systems at the Universitat Jaume I, Castellón, Spain, where he teaches Geographic Information Systems and Systems Integration. His research interests include Spatial Data Infrastructures (SDI) and web services interoperability. Contact: gould@lsi.uji.es

Andreas Wytzisk is assistant professor at ITC. His research activities focus on interoperability issues in spatial data infrastructures, distributed simulations and sensor webs. Wytzisk holds a PhD in Geoinformatics from the University of Münster, Germany. Contact: wytzisk@itc.nl.

Rolf de By is associate professor at ITC, in the department of Geoinformation Processing. His research interests are in the fields of designing large and advanced information systems that handle geospatial data, spatial database technology and methods, and novel applications. De By holds an M.SC. Degree in Applied Mathematics and a Ph.D. degree in Computer Science, both from the University of Twente. Contact: deby@itc.nl

Peter van Oosterom obtained a MSc in Technical Computer Science in 1985 from Delft University of Technology, The Netherlands. In 1990 he received a PhD from Leiden University for this thesis "Reactive Data Structures for GIS". Since 2000 he is professor at the Delft University of Technology (OTB) and head of the section ‘GIS Technology’. He is European editor for the International Journal on Computers, Environment and Urban Systems CEUS. Contact: P.J.M.vanOosterom@tudelft.nl