Computers & Geosciences 47 (2012) 60–74

Contents lists available at SciVerse ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Automated geospatial Web Services composition based on geodata quality requirements

Sérgio A.B. Cruz [a,b,*], Antonio M.V. Monteiro [a], Rafael Santos [a]

[a] National Institute for Space Research (INPE), P.O. Box 515, São José dos Campos - SP, Brazil
[b] Embrapa Agriculture Informatics, Embrapa, P.O. Box 6041, Campinas, SP, Brazil

ARTICLE INFO

Article history:
Received 6 May 2011
Received in revised form
17 September 2011
Accepted 21 November 2011
Available online 19 December 2011

Keywords:
Service-Oriented Architecture (SOA)
Geoprocessing
Web Service composition
Geospatial analysis
Geodata quality
Artificial intelligence planning method

ABSTRACT

Service-Oriented Architecture and Web Services technologies improve the performance of activities involved in geospatial analysis with a distributed computing architecture. However, the design of the geospatial analysis process on this platform, by combining component Web Services, presents some open issues. The automated construction of these compositions represents an important research topic. Some approaches to solving this problem are based on AI planning methods coupled with semantic service descriptions. This work presents a new approach using AI planning methods to improve the robustness of the produced geospatial Web Services composition. For this purpose, we use semantic descriptions of geospatial data quality requirements in a rule-based form. These rules allow the semantic annotation of geospatial data and, coupled with the conditional planning method, this approach represents more precisely the situations of nonconformities with geodata quality that may occur during the execution of the Web Service composition. The service compositions produced by this method are more robust, thus improving process reliability when working with a composition of chained geospatial Web Services.

1. Introduction

Service-Oriented Architecture (SOA) (Erl, 2004) is a distributed computing architecture supported by software components that are accessible by communication protocols. These loosely coupled and autonomous components set up basic functionalities. Various components or services can be combined to achieve high-level functionalities from basic ones. SOA can be used as a platform to perform geospatial analysis. In this case, the Web Services perform the sequence of activities carried out during the geospatial analysis, which encompasses the gathering and processing of geodata. The set of ordered invocations of Web Services needed to achieve the geospatial analysis goals can be considered as a process on the SOA architecture.

Several papers have discussed the automated building of process specifications by combining Web Services (Aggarwal et al., 2004; Milanovic and Malek, 2004; Papazoglou and Heuvel, 2007; Rao and Su, 2004; Sirin et al., 2005; Zhovtobryukh, 2007). Many proposed solutions are based on a static perspective of Web Services descriptions. Ontologies, such as the OWL Web Ontology Language for Services (OWL-S) (W3C, 2004), semantically describe Web Services based on descriptions of their input and output parameters, execution preconditions and effects. The functionalities of Web Services can be indicated implicitly in terms of these service properties or explicitly by associating the Web Services to service functionality taxonomies (Sirin et al., 2004; Sycara et al., 2003; Yue et al., 2007). However, the execution of the generated process may be unsuccessful at solving real-world problems, even for an optimized composition, as the dynamic components of the processes are not considered. We consider dynamic components in this case the characteristics intrinsic to each geodata instance which differentiates individuals of the same type. Satellite images from the same sensor type and from the same region may present differences due to, for example, the cloud coverage, which conditions its use. This lack of regard results in a lack of quality in the geospatial data produced by service compositions built using a static view. Geodata quality in this context, describes the fidelity with which a geodata instance represents the real world, enabling it to be used in geoprocessing procedures or in decision-making with maximum reliability.

The results produced by performing these compositions may contain errors arising from nonconformities to data quality requirements in the messages exchanged between the data producer and the consumer services. Each data instance exchanged by the services has particular characteristics that enable it to be used by a specific geoprocessing procedure. Such suitable characteristics can be described as geospatial data quality requirements that can be checked only at run-time.

In this work, we propose that semantic descriptions of the dynamic aspects of geospatial data quality requirements in the

* Corresponding author at: Embrapa Agriculture Informatics, Embrapa, P.O. Box 6041–13083-886, Campinas, SP, Brazil. Tel.: +55 19 3211 5700; fax: +55 19 3211 5754.
E-mail address: sergio@cnptia.embrapa.br (S.A.B. Cruz).

0098-3004/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cageo.2011.11.020

message exchanges of a Web Services composition can improve the robustness of the process. In our approach, we use quality requirements to perform a semantic annotation of geospatial data that highlights the nonconformities at run-time. Conditional planning then handles any identified nonconformities. If a nonconformity is identified, geographic analysis procedures are scheduled as a contingency plan to fit the geodata to the consumer services' data quality requirements. The nonconformity will be registered in a process report if a contingency plan is not available. In this way, we can create more reliable service compositions, which will produce better quality geodata.

2. Web Services composition and planning methods

The planning problem involves finding a sequence of actions that changes the initial world state to a final or goal world state (Russell and Norvig, 2003). Web Services can be used as actions in the planning methods to solve this problem in the context of Web Service composition. The input and output parameter data types represent states in the planning problem (Carman et al., 2003). These input and output parameters define the state transformations performed by the Web Service or, implicitly, its functionality (Sirin et al., 2004). Here we assume that the Web Service has only one operation and its functionality is associated with information transformations (Martin et al., 2007).

In the context of Web Services, the planning problem can be described as follows:

D is the set of data types, which correspond to the states.
D_{init} is the set of available data types, which describes the initial states.
$D_{req} \subseteq D$ is the set of required data types, which correspond to the goal state.
W is the set of Web Services, which correspond to the actions.
$\delta : D \times W \rightarrow D$ is a transition function. Each tuple implicitly describes a Web Service semantic.

For convenience, we can define a *GEN* function that provides all Web Services that can produce a data instance with a *dout* data type.

$$GEN(dout) = \{w \in W, din \in D \,|\, \exists \delta(din, w) = dout\}$$

$GEN : D \rightarrow P(W)$ where $P(W)$ is the power set of W.

The composition engine needs to find a sequence of Web Services that produce an instance of data type D_{req}. Listing 1 illustrates the goal-oriented procedure performed by the engine. For the geospatial domain, the D_{req} and D_{init} in this procedure represent, respectively, the required and available geodata types.

Listing 1. A service composition procedure based on input and output datatype matching.

```
function COMPOSE(D_init , D_req ) returns a composition
{
  service_fringe ← PREDECESSORS(D_req - D_init )
  if (service_fringe is empty )
    then return empty composition
  input_set ← ⋃ service_fringe inputs
  output_set ← ⋃ service_fringe outputs
  NewD_req ← D_req − output_set data types ∪ input_set data
types
  NewD_init ← D_init ∪ output_set data types
  plan ← MERGE(service_fringe, COMPOSE(NewD_init ,New_req ))
  return plan
}

function PREDECESSORS(requested_datatypes ) returns services
{
  services ← [ ]
  for each datatype in requested_datatypes
  {
    candidate_services ← GEN(datatype)
    selected_service ← SELECT(candidate_services)
    services ← services + selected_service
  }
}

function MERGE (service_fringe , new_service_fringe )
{  Connect new_service_fringe outputs to the corresponding
      service_fringe inputs
}

function GEN (datatype ) returns services
{  Find services providing datatype based on the semantic
      subsumption relationship of data types
}

function SELECT (candidate_services ) returns services
{  Select services from candidate_services based on a heuristic
}
```

This procedure performs a breadth-first search using a backward planning strategy to find the services needed to produce the required data type with a lower branching factor. The goal of this procedure is to discover and select services that satisfy the data type dependencies of the composition. The data type matches are evaluated on a semantic level to verify if a service output data type contains the required data type.

Problems arise with this procedure when the functionality of a service cannot be inferred from its input and output data types. This problem occurs when the input and output parameters have the same data type. Some procedures in the geospatial analysis domain have this property, such as data interpolations, rate smoothing and scale adjustments. These procedures act on the properties of the data instance instead of performing data type transformations. The approach shown in Listing 1 does not include such geospatial procedures in the service execution plan, because the algorithm is based on a classical planning approach (Russell and Norvig, 2003) that considers the output geodata to be in an ideal state for use by the consumer service. Geodata in an ideal state enables the execution of the consumer service with the generation of a new geodata instance with the inclusion of minimal errors, that can be suitably used in the rest of the process. However, this state cannot always be verified in the service interactions. The procedure in Listing 1 assumes that restrictions based only on data types express the necessary and sufficient condition for the use of geodata by a consumer service.

We can identify two types of unpredictable fault situations at process runtime. The first encompasses operational failures such as unresponsive and inaccessible services. The second type corresponds to instances in which unsuitable geodata is supplied to the geospatial services. Geodata with outlier values, random spatial distributions or cloud coverage are examples of unsuitable geodata, and these faults may be inherent to the geodata production. However, building new service compositions can dynamically define new and unforeseen contexts for the use of the geodata and can also cause these faults. We focus on the second type of unpredictable fault, which is related to dynamic aspects of the service interactions while performing the process. In this work, we propose modeling this dynamic perspective with geodata quality requirement rules, which can be used in a conditional planning strategy to automate the construction geospatial Web Service compositions.