Incremental Composition of Geographic Web Services: An Emergency Management Context

Carlos Granell, José Poveda, Michael Gould
Department of Information Systems (LSI)
Universitat Jaume I
E-12071 Castellón, Spain
canut@uji.es

SUMMARY

We describe an incremental, semi-automated method for composition of atomic or composite web services in a geographical domain. It has distinct advantages over more complex web service choreography languages such as BPEL. This method is being tested in an emergency management scenario as part of the ACE-GIS (EU) project.

KEYWORDS: *Web services composition, incremental composition, services interoperability, emergency management*.

INTRODUCTION

Electronic cooperation, from e-government to e-commerce has become so important over the past few years that it promises to become the global work paradigm of excellence in the near future. For this reason, the development of web service technologies for enabling e-services has become an important emerging research area. A web service can be defined as a set of related functionalities that can be accessed via WWW through XML standards (Tsur *et al*. 2001). This concept is becoming increasingly important because business, government, and consumers in general are making the web a place for all types of activities, that is, consolidating the basis of a global information society.

For the purpose of our study, a general web service can be considered either an atomic or a composite service. An atomic service is an Internet-based software component that does not rely on other web services to fulfill user requests. An example of an atomic service is a weather service that, give a location, provides weather-related information such as wind direction and magnitude. On the other hand, a composite service, or a so-called opaque service chain in ISO/OGC terminology (ISO/TC211, 2002), is defined as a set of such general services (atomic or composite) working together to offer a value-added service. As part of a pilot application in the ACE-GIS project (Adaptable and Composable E-Commerce and Geographic Information Services, http://www.acegis.net), a toxic gas dispersion model has been selected as an example of a composite service which is composed of *GetNearestAirport*, *GetWind, GetGasDispersionPlume*, and *GetGasDispersionMap* services to provide key information for a realistic emergency response plan. An important issue for mature e-services will be to offer not only atomic services to be used *ad hoc* but also value-added composite services based on the chaining of other services (atomic or also composite) which are then invocated using web technologies (Alameh 2003).

The web services paradigm was designed with an eye on heterogeneous platform creation (mix of languages, applications), however to enable creation of a value-added composite service, it would be necessary to provide interoperability between specific web services. In the geographical domain, the concept of interoperability is not a new issue. The conceptual architecture used by Open GIS Consortium (OGC) defines several atomic services, such as Web Map Server (WMS), Web Feature Service (WFS), geocoders or gazetteers, in order to allow the assembly of composite services. In addition, the taxonomy of geographic services defined in ISO/TC 211 supposes a high level of heterogeneity among diverse these services. Therefore, to enable real service chaining it is necessary that these services are interoperable among themselves. In this way, OGC has launched the Open Web Services initiative phase 1.2 (OWS 1.2) to address interoperability requirements in an open and distributed environment such as is the web. This initiative analyses current specifications and standards in the context of web services in order to decide the future specifications of aforementioned OGC services using web services technology to achieve the desired interoperability. Limited experiments were carried out in this way such as for SOAP (OGC 2003a) and UDDI (OGC 2003b). Thus far, OGC has demonstrated client-server interoperability via the implementation specifications, for example multi-vendor interactions between WFS or WMS clients and servers, however interactions between WFS, WMS and other services are not yet well defined. Therefore, this current (conformance-based) interoperability is not enough in the context of generic interoperability of OGC services.

For this reason, in this paper we discuss a novel approach for going from one-to-one conformance to true interoperability among diverse web services. We show a model for composing web services in the geographical domain, based on declarative and interoperability aspects, and present how our approach takes the first steps toward composition and invocation of diverse OGC services, in a "real" emergency plan scenario. This example is proposed as a use case in the ACE-GIS project, which proposes to build a developers platform for model-driven design and invocation of compound geographic information services (in addition to basic e-commerce services such as authentication and eventually payment).

INCREMENTAL COMPOSITION

The incremental composition model (described in detail in Granell *et al.* 2003) outlines a declarative composition model for web services, which is centered around descriptive aspects, on interoperability and on scalability. Basically, a web service composition is defined as a set of atomic or composite web services which interact according to certain logical rules. These logical rules specify the patterns of composition (serial, parallel, etc.) that describe the execution order of the services involved in the composition, and the connection flows established for the data flow between services. This composition differs from WS choreography languages such as BPEL, which attempt to define an entire workflow graph in one go. The incremental approach taken here is more cautious, in the sense that the developer moves ahead pair by pair, assuring herself along the way that the composition is functional at each step before moving ahead.

Both atomic and composite services should appear to the user as a single entity with which to interact. This affords several benefits including encapsulation of the underlying service complexity and straightforward reuse of a composition in future compositions. Composing web services requires the description of each service so that other services can interact with it. The language for describing operational features of web services, in most all composition languages, is WSDL. In treating a composition as a single web service, however, this composition should also carry a WSDL describing the entire functionality of the composition. In these composition descriptions only the abstract description is present, without details on concrete implementation: things such as access protocols or location points. In addition to the abstract WSDL description, a composite service incorporates the specification of composition pattern and connection flow, which are encoded declaratively in an XML WSDL extension. Additional aspects such as the quality of service (QoS) desired in a composition, semantic attributes or security considerations, may also be embedded declaratively in the description. This together -- abstract WSDL plus XML extension -- forms an abstract interoperability interface of the composite service. This composite WSDL description is fundamental for the incremental composition process, as the approximation follows an abstract component-based model, which facilitates interoperability and connectivity between services.

The approximation presented is essentially component-oriented and uses a backtracking-mode algorithm for invocation. The resulting composition graph, as logical relations between services, is defined implicitly by the very structure of the composition. This graph may be thought of as a tree structure, whose leaf nodes are linked to atomic services, and whose internal nodes represent virtual intermediate compositions, where the root node represents and encapsulates the entire composition. The composition is virtual in the sense that each atomic service functions independently, without the need to know that it is being used by other services. In addition, each internal node (or composition) does not exist physically as an atomic service, but it can be considered as a logical or virtual service. Therefore, the final composition encapsulates the complexity of the model in a manner similar to that of class hierarchy in object oriented languages; the main difference being that here we refer to abstract interfaces and not instantiations of classes.

According to the conceptual architecture that supports the incremental composition model (Granell *et al.* 2003), we establish two functional views of the system in composition and invocation of services:

Composition view. During this process, a user establishes a composition pattern and connection flow for the messages involved in linking a pair of web services. This model follows a component-oriented composition which consists of decomposing complex compositions into basic patterns of pairs (avoiding the need for a flow language). In this manner the desired composition is constructed in multiple iterations of composing two at a time: for example, simple to simple forming complex, then complex plus another simple, etc., in an *incremental*-mode. Even through current composition languages allow construction of several services simultaneously (Aissi *et al.* 2002), in our geographic service context it is not realistic that a user would combine, for example, 10 Web Map Services at the same time. Therefore, in principle, we prefer a simple and consistent manner to create compositions instead of increasing the cardinality of services composition each time. This is similar to recommended procedure for debugging software a piece at a time and recompiling.

Invocation view. In general terms the invocation of a composition requires an analysis of the associated description to encode the logical composition graph as a tree structure. Then, once this structure is created it is traversed in backtracking-mode, invoking directly the atomic web services and chaining the results obtained according to the connection flow defined in the composite WSDL description. As a shared point in other proposals on compositional languages, backtracking-mode invocation through the incremental composition model achieves a dynamic binding with concrete web service instances at execution time. The access mechanism to a service is not known until the moment of invocation. In this way, the composition using the abstract interfaces provides maximum flexibility compared to binding at design time.

As an initial, didactic example, we consider a scenario for composing simple mathematical functions (figure 1). Assume that a catalog provides basic functions (sum, subtract, square, cube root, etc.) as atomic services. A typical example would be a composition of arbitrary number of these simple functions to solve a complex function, say *C*, which would be totally inefficient to attempt to create a separate and permanent web service for each possible mathematical composition. Figure 1 depicts a logical graph of a hypothetical function *C* composed in turn of three compositions (*C1*, *C2*, *C3*). All these composition are virtual (dotted line rectangle), that is, there are not directly executed. While *C1* and *C3* are seen as a black box by the user, *C2* can be examined in order to access its children atomic services (*Add* and *Square root*). This simple example exposes how the construction process of compositions follows an incremental design which hides composition complexity from the user (encapsulation) and allows selective deep-level views of the graph, facilitates independent invocation of these intermediate combinations (for example executing and testing only *C1*), and facilitates the debugging and error detection in the final composition.

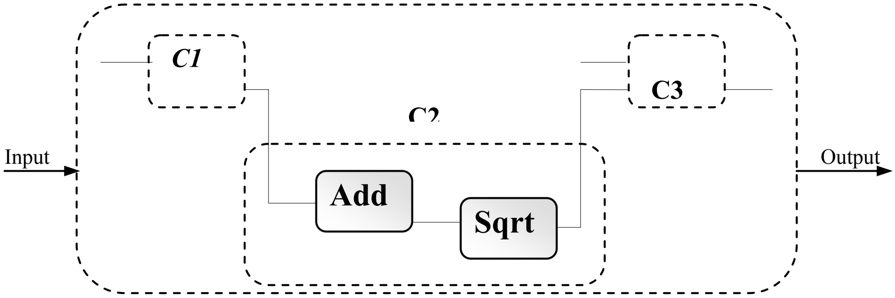

Figure 1: Selective view of the logical composition graph of mathematical functions case.

USE CASE: EMERGENCY SCENARIO

In this section we present an emergency pilot as use case to investigate the interoperability among diverse and more complex web services. This use case is one of the pilot applications proposed in the ACE-GIS project. The emergency pilot deals with accidents involving toxic gas releases from a chemical plant located at an arbitrary location. Suppose that there is an accident and fire officers, police officers or other technicians need information in near real time. For example, the fire officers can retrieve emergency plans or police can track a gas dispersion model to determine population at risk and for evacuation if necessary.

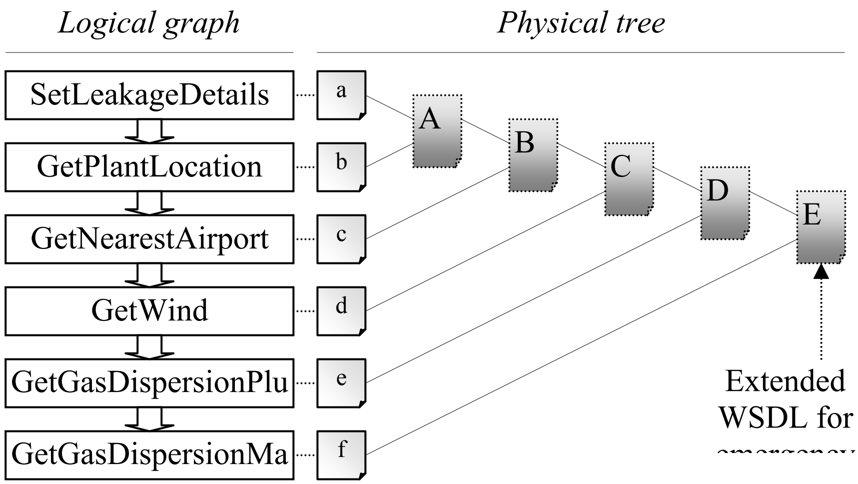

Figure 2: Incremental composition for emergency response service.

Figure 2 shows a serial sequence of web services (on the left) from the occurrence of the accident until the emergency response plan is generated. In this example, the last two services, *GetGasDispersionPlume* and *GetGasDispersionMap* are wrappers containing OGC services, WFS and WMS respectively. All atomic web services involved in the composition are associated with WSDL file descriptions (solid shaded files named with lower case letters). According to incremental composition, we create the desired composition (*emergency response service*, *WSDL E*) in multiple iterations of composing two at a time. So, first *SetLeakegeDetails* (*WSDL a*) and *GetPlantLocation* (*WSDL b*) are composed to obtain a new virtual composition described with an extended WSDL, *WSDL A* (dotted shaded files named with capital letters). The process continues until we obtain the required composition (*WSDL E*). The logical sequence of web services represents the logical composition graph, whereas the result of applying the incremental composition produces a tree structure.

This approximation allows a means for composing services in a simple, consistent manner. Therefore, this model holds the advantage that as complex as the process model may be, the logical graph may be decomposed into basic patterns of pairs of services. In addition, any virtual composition can be executed and reused independently in other compositions, that is, acts as an atomic web service. In some sense, this highly controlled pair-wise composition mirrors the incremental approach to processing pairs of map layers (e.g. overlay of binary rasters). Because many GIS allow a maximum of 2 layers to be processed at a time, this creates a string of intermediate products (map files), which are then again processed with another map. At each stage of the map processing the intermediate result is something simple and clear, to be able to judge whether or not the process is working thus far. This is opposed to the so-called object-oriented continuous map model, which is computationally more elegant perhaps, however which can lead to inordinate numbers of intersections and processes, such that when a strange resulting map appears at the end, it is often difficult to determine what might have caused the error.

The incremental composition approach described here offers a similar, simplistic platform for composing web services, which we believe will lead to more satisfactory results for a wide range of GI-based compositions, such as the emergency response case posed here and in the ACE-GIS project.

CONCLUSIONS

We have presented a novel approach for composition and invocation of web services within the context of the ACE-GIS project, on-going research on geographic web services interoperability. This approach is aimed at becoming a first step in resolving some of the key interoperability problems among GI services.

The incremental model proposes a definition of web service compositions which are loosely coupled based only on abstract syntactic descriptions in WSDL. As a distinct benefit over known composition languages, this approach does not define a language or a small set of well-known compositional languages. Just as the Web's success can be attributed to its simplicity, we propose a process in which services are composed in an incremental and controllable manner, avoiding the need for a compositional or flow language to describe complex graphs of multiple service interactions. This incremental composition is specified declaratively in XML WSDL extensions, as an aggregation of external descriptions of the services as well as the composition pattern and connection flow. From the inherent structure of this incremental composition, a tree structure is generated which then is traversed in backtracking-mode order to invoke the composite service. Future work will include the promotion of the incremental model components under open source license and the presentation of the WSDL extensions to relevant standards bodies, beginning with OGC and OASIS, in order to improve its general acceptance and use.

Thus far we have developed a prototype capable of guiding the user in the creation of compositions formed by an arbitrary number of existing web services, such as the mathematical function composition scenario. Now, we are working to extend our incremental composition model to include any XML-Schema's user-defined data type interchanged between services, such as in the use case of the emergency scenario. That would generalize our model to compose any available web service, consequently permitting the composition of web services based on principles of interoperability and scalability.

ACKNOWLEDGEMENTS

The work described here has been supported in part by European Union project IST-2001-37724 (ACE-GIS) and by Spain Ministry of Science and Technology grant TIC-2000- 1568-C3. Feedback from ACE-GIS project partners has been especially helpful.

BIBLIOGRAPHY

Aissi A., Malu P., and Srinivasan K., E-business Process Modeling: The Next Big Step. *IEEE Computer*, 35(5), 55-62, 2002.

Alameh N., Chaining geographic information web services, *IEEE Internet Computing*, Sept-Oct 2003, 22-29.

Granell C., Poveda J, and Gould M., Incremental Weak Composition and Invocation of Geographic Web Services. In *Proceedings of the Intl. Workshop on Semantic Processing of Spatial Data (GEOPRO 2003),* Mexico City, November 4-5, 2003, 179-187.

ISO/TC211, Geographic information – Services (ISO/DIS 19119) v4.3. International Organization for Standardization. Also OpenGIS Consortium Abstract Specification 13. 2002.

OGCa, OWS 1.2 SOAP Experiment Report. OpenGIS Consortium, *OpenGIS Discussion Paper OGC 03-014*, 2003.

OGCb, OWS 1.2 UDDI Experiment. OpenGIS Consortium, *OpenGIS Discussion Paper OGC 03-028*, 2003.

Tsur S., Abiteboul S., Agrawal R, Dayal U., Klein J., and Weikum G., Are Web services the next revolution in e-commerce? (Panel). In *Proceedings of the VLDB Conference*, Rome, 614-617, 2001.