ONTOLOGY-SUPPORTED AUTOMATIC SERVICE CHAINING FOR GEOSPATIAL KNOWLEDGE DISCOVERY

Liping Di, Professor and Director
Peng Yue, PhD student
Wenli Yang, Principle Scientist and Associate Director
Genong Yu, Post-doctoral Research Associate
Peisheng Zhao, Research Assistant Professor
Yaxing Wei, PhD student
Center for Spatial Information Science and Systems (CSISS)
George Mason University
6301 Ivy Lane, Suite 620, Greenbelt, MD 20770
ldi@gmu.edu
pyue@gmu.edu
wyang1@gmu.edu
gyu@gmu.edu
pzhao@gmu.edu
ywei@gmu.edu

ABSTRACT

With the advances in sensor and platform technologies, the capability for collecting geospatial data has significantly increased. Large volumes of data have been collected using remote sensing. While those data are potentially valuable for the benefit of society, they must be converted to geospatial knowledge before they are useful. The traditional methods — only geospatial experts analyze data — fall far short of today's increased demands for geospatial knowledge. As a result, significant amounts of data have not even once been analyzed after collection. Recent progress in the geospatial semantic Web has shown promise for developing automatic geospatial knowledge discovery methods for solving application problems, which otherwise require considerable resources. This paper presents an approach for automatically solving geospatial problems in the geospatial semantic Web environment. The approach simulates the process used by geospatial experts who first use backward reasoning from the required knowledge to the available raw data to select a set of available geo-processing functions, and then execute the functions sequentially, starting from raw data, to derive the desired knowledge. This backward reasoning effectively creates a path from raw geospatial data to the desired geospatial knowledge. With rich semantic descriptions of services and the support of ontology, the path can be formed automatically through backward reasoning from the desired result to raw geospatial data using semantic Web services. Such a path can be instantiated to become an executable workflow to generate the result automatically. A prototypical system is implemented to demonstrate the above concept and approach.

INTRODUCTION

With recent advances in sensor and platform technology, the capability for collecting geospatial data has significantly increased. Large amounts of geospatial data have accumulated over the years, through established satellite observations, emerging sensor networks, and the spread of location-sensitive data collectors. While those data are potentially valuable for societal benefits, they must be converted to geospatial knowledge before they become useful. A gap needs to be filled with proper geospatial processing before the end users can extract what they really want in terms of knowledge and information. For example, a fire chief for fire may be interested in information about where fires occur and how large an area is affected. A risk manager may be interested in knowing the potential risks of natural hazards in his territory. Such information does not pop out from the remotely sensed data directly, but can be achieved through a series of geospatial processing steps, i.e. image pre-processing, classifier training, classification, statistical summarizing, and result presentation. A series of processes may need to be invoked during the

processing. This can become problematic and laborious when the data are fragmented across different sectors of agencies(Brodaric and Gahegan, 2006). Expected increases in Web-based services for both information and processing capabilities will exacerbate difficulties in finding, integrating, and using such services to meet the increasing demand from different users. It is not realistic to prepare all the information beforehand considering the continuous feed of sensor data and the extremely huge amount of possible intermediate data. The traditional method of analyzing data - only by geospatial experts - falls far short of today's increased demands for geospatial knowledge. As a result, a significant amount of data has not even once been analyzed after collection.

Several efforts have emerged to deal with these problems. Among them, two concepts are the most relevant to the work presented in this paper. They are *interoperability* and *virtual product*. To reduce the human involvement in connecting the dots between geospatial Web services and geospatial data, interoperable Web services and data are needed. When Web-based geospatial components (services and data) are published following standard interfaces or formats, little or no knowledge of these components is required by users to be able to exchange information, execute services, and transfer data between components. Such Web components are called interoperable. To reduce storage requirements and to meet the real time or near real time requirements, a model to generate the final geospatial product under request, instead of all the intermediate geospatial data, is actually stored and managed. The underlying technology is Web services. Use of Web services can significantly reduce the data volume, computing steps and resources required at the end-user side(Di and McDonald, 1999). The model describes the composite process that connects required Web services and data. The model is also called virtual product.

The key step to realize this vision of the geospatial cyberinfrastructure and promote the ultimately extensive use of geospatial data is the composition of geospatial Web resources – services and data. The key problem is the heterogeneity of geospatial resources. There are two levels of heterogeneity – structural (schemas) and semantic (meaning)(Brodaric and Gahegan, 2006). Correspondingly, two levels of interoperability deal with such heterogeneity – syntactical and semantic. Integration and composition of heterogeneous Web services and data can be done by properly matching their inputs and outputs at a common knowledge level(Badea et al., 2004). With ontology, or a well-defined common (domain-specific) concept that gives formal (machine processable) descriptions of geospatial Web services and relationships between them, automation of service composition is possible. Automatic service composition can be of great value to the geospatial user community because it can greatly broaden the uses of geospatial knowledge in social and economic activities and it can automatically provide answers to users' questions(Di, 2004). The application of ontology to geospatial Web reasoning is a promising approach in parallel with other kinds of progress in the semantic Web. Studies have shown great promise in the automatic discovery of geospatial knowledge for solving application problems, which otherwise require tremendous resources. This paper presents an approach for automatically solving geospatial problems in the geospatial semantic Web environment.

There are three key issues for automatic service composition: interoperable Web services, discovery, and assembly(Di, 2005). We have studied all aspects of automatic Web service composition since early 2003 in a large project. Overall architecture were discussed in another paper(Yue et al., 2007). In this paper, we focus on a reasoning algorithm that automates the composition of Web services based on geospatial ontology. First, the concept of reasoning in the Web environment is presented. Second, supporting techniques, e.g., geospatial ontology, are discussed. Third, the procedure and algorithm by which backward reasoning is implemented. Fourth, application scenarios are discussed in detail. Fifth, related work on applying semantic Web technologies in integration and composition of heterogeneous geospatial resources is discussed. Finally, we give concluding remarks and future directions.

WORKING THEORY

A *Web service* is a program accessible over the Web through some standard interface. A service can be described by its *input* parameters and their properties (*pre-conditions*), *outputs* and their properties (*post-conditions*), and *side-effects*(Kona et al., 2007). This definition leads to a better understanding of service discovery and composition. Service discovery is defined as the process of finding a needed service by matching the query template of a service that has set inputs, pre-conditions, outputs, post-conditions, and side-effects(Kona et al., 2007). In the semantic Web, a Web service is described in some semantic-expressive language, such as OWL-S(Martin et al., 2004). Matching can be extended to include

subsumption, *equivalent*, and *implication*. In other words, besides an exact match, the following cases can also be considered as a match:

(1) For inputs and outputs, subsumed match occurs, or the found type is subsumed to or belongs to the required type, e.g. Landsat ETM (Enhanced Thematic Mapper) derived NDVI (Normalized Difference Vegetation Index) is subsumed to NDVI(Yue et al., 2007);
(2) For inputs and outputs, an equivalent term to describe the data type is found, e.g. landslide susceptibility, landslide potential, and landslide probability are semantically the same concepts as inferred from some given geospatial data ontology;
(3) For properties, the property found implies the required property, e.g. "within Maryland" for a dataset implies that it is also "within the US".

When considering the match of a service, extra outputs can be ignored, but extra inputs can be problematic. The extra inputs cannot be satisfied by the given conditions and, therefore, the match has to be rejected. This is one important motivation for discussing backtracking or a tracking process.

Service composition is required if a single (atomic) service can not be found, given the required inputs, outputs, pre-conditions, post-conditions, and side-effects. By this process, several Web services are invoked to meet requirements. Service composition can be conceptualized as a *directed acyclic graph* (DAG) where a vertex represents a service and an edge connects inputs and pre-conditions to the service or outputs and post-conditions to the service(Kona et al., 2007). The edge that connects inputs to a service is called an *incoming edge*. The edge that connects outputs to a service is called an *outgoing edge*. As pointed out by Kona et al.(Kona et al., 2007), the following conditions holds for a composite service

(1) Leaf nodes (atomic service) have zero incoming edges or given inputs;
(2) Head nodes (final output) have zero outgoing edges;
(3) Middle nodes (services) have at least one incoming edge and at least one outgoing edge.

Automatic composition of Web services has two stages. The first stage is the chaining of Web services considering only the matching of inputs and outputs. The second stage deals with the pre-conditions and post-conditions by automatically invoking some intermediate services, such as the invocation of re-projection service if the data projection property does not match, By taking these two stages, Web service composition can be automated in a rule-based system if each service is considered to be as a rule in the form of*"IF (inputs) THEN (outputs)"*. Once services are represented in a rule-based system, backward and forward reasoning can be applied to chain the services automatically. *Forward reasoning* is a bottom-up approach that starts with the initial facts and keep on deriving new conclusions until all the outputs are satisfied. *Backward reasoning* is a top-down (or goal-driven) approach that starts with a goal and looks for rules that generate the required outputs. For geospatial product requirements, the goal is defined. Trying all possible outputs that could be generated by given inputs from the catalog system is either unnecessary or too time-consuming. To adopt backward reasoning to reason through the limited number of rules is more efficient. Therefore, backward reasoning is used in the automation of geospatial Web service composition. Inputs and outputs can be matched at several levels. They can be matched at the instance level matching all the concrete properties and conditions. They can also be matched at the data type level, which ignores the details associated with each instance. Matching at the data type is considered more flexible during the first stage. This is easier to handle. With proper ontology, more meaningful and flexible matching can be done. The process of semantically matching is going to be discussed further in next section for geospatial ontology.

To illustrate the mechanism in applying backward reasoning in geospatial Web service composition, we start with a simple repository of Web services or rules. Existing geospatial data available through these Web services can be considered as initial facts in the rule-based reasoning system. In systems compliant with open standards, geospatial data are typically served through WCS (Web Coverage Service), WFS (Web Feature Service) or SOS (Sensor Observation Service). Processes are defined as compliant with WPS (Web Processing Service). These standard services make the interoperation between services possible. In our example, all data were made available through such standard interfaces and therefore they can be easily invoked and executed at later stages. A rule-based system contains rules, initial facts and requirements. Here we start with the following rules (Web services):

(1) Slope service: IF (exist DEM) THEN (exist slope)
(2) Aspect service: IF (exist DEM) THEN (exist aspect)
(3) Land cover classification service: IF (exist TrainingImage) AND (exist ETMImage) THEN (exist Landcover)
(4) ETM NDVI service: IF (exist NIRImage) AND (exist RedImage) THEN (exist ETM_NDVI)

(5) IF (exist slope) AND (exist aspect) AND (exist Landcover) AND (exist ETM_NDVI) THEN (exist LandslideSusceptibility)

In the catalog systems (CS/W), we have the following datasets served through WCS (the initial facts).

(a) (exist DEM)
(b) (exist TrainingImage)
(c) (exist ETMImage)
(d) (exist NIRImage)
(e) (exist RedImage)

We need to prove the following:

- (exist LandslideSusceptibility)

First we search if the goal state is in the initial facts (datasets). There is no such a rule in the initial facts. Then, we try to match it against the conclusions of each rule. It matches rule (5). We choose rule (5). We need to prove *(exist slope), (exist aspect)*, *(exist Landcover),* and *(exist ETM_NDVI)*. To prove *(exist slope)*, we find matching rule (1) and then need to prove *(exist DEM)*. This condition *(exist DEM)* is in the set of initial facts. We need to prove *(exist aspect)* and find rule (2) to be used. Proving *(exist Landcover)* leads to rule (3) and the need to prove *(exist TrainingImage)* and *(exist ETMImage)*. Both are in the set of initial facts. Last, we need to prove *(exist ETM_NDVI)* and find that rule (4) can be used. Both conditions for rule (4) are in the set of initial facts. Up to this point, the goal has been proved since then initial facts show that all sub-goals are proven.

In reality, it is possible to find more than one rule matching the given criteria. In such case, we have to try each rule exhaustively. A *backtracking* mechanism should be used. If one rule fails, we go back and continue to try the other rules at that point.

ONTOLOGICAL TYPING

There are three possible levels of matching and discovery: dictionary, thesaurus, and ontology. At the level of "dictionary", we could find the data by matching keywords. At the level of "thesaurus", we could find the data by expanding the keywords with anonyms and synonyms. At the level of "ontology", we could do a focused match with meaning and subsumed matching. The use of ontology extends the matching of inputs and outputs in Web service composition.

To enable the accurate discovery of services and data, it is very important to have a proper ontology for both service and data. The analysis above shows that the choice of service is determined by matching its inputs and outputs with those required. The crucial parameters for both inputs and outputs can be represented as data types. The correct definition of these data types is important for the success of the exact matching and beyond to enable more flexible matching, such as subsumption, equivalent inference and other relationships expressed in ontology.

OWL-S (Web Ontology Language for Web Service) is becoming the de-facto approach to define Web service semantically. The backbone OWL (Web Ontology Language) is capable of defining any relationship expressible in description logic. Both OWL-S and OWL are used here to represent semantics for geospatial service and data. Figure 1 shows a model that a service may be needed to be describe a service. For each data type in input and/or output set, an ontology for data types may be used to describe relationships between data types. Figure 2 shows one example for the data types used in the case of landslide susceptibility.

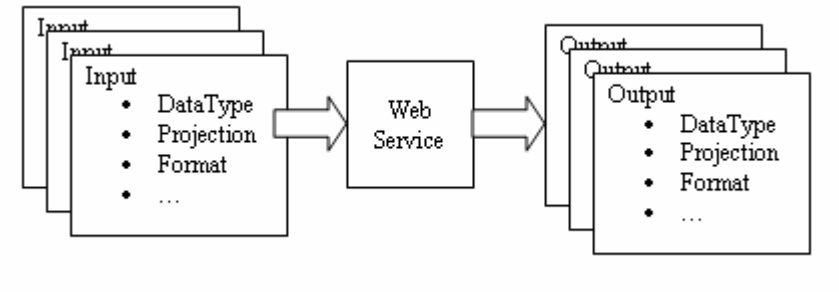

Figure 1. Service Model (inputs, outputs, pre-conditions, post-conditions, and side-effects)

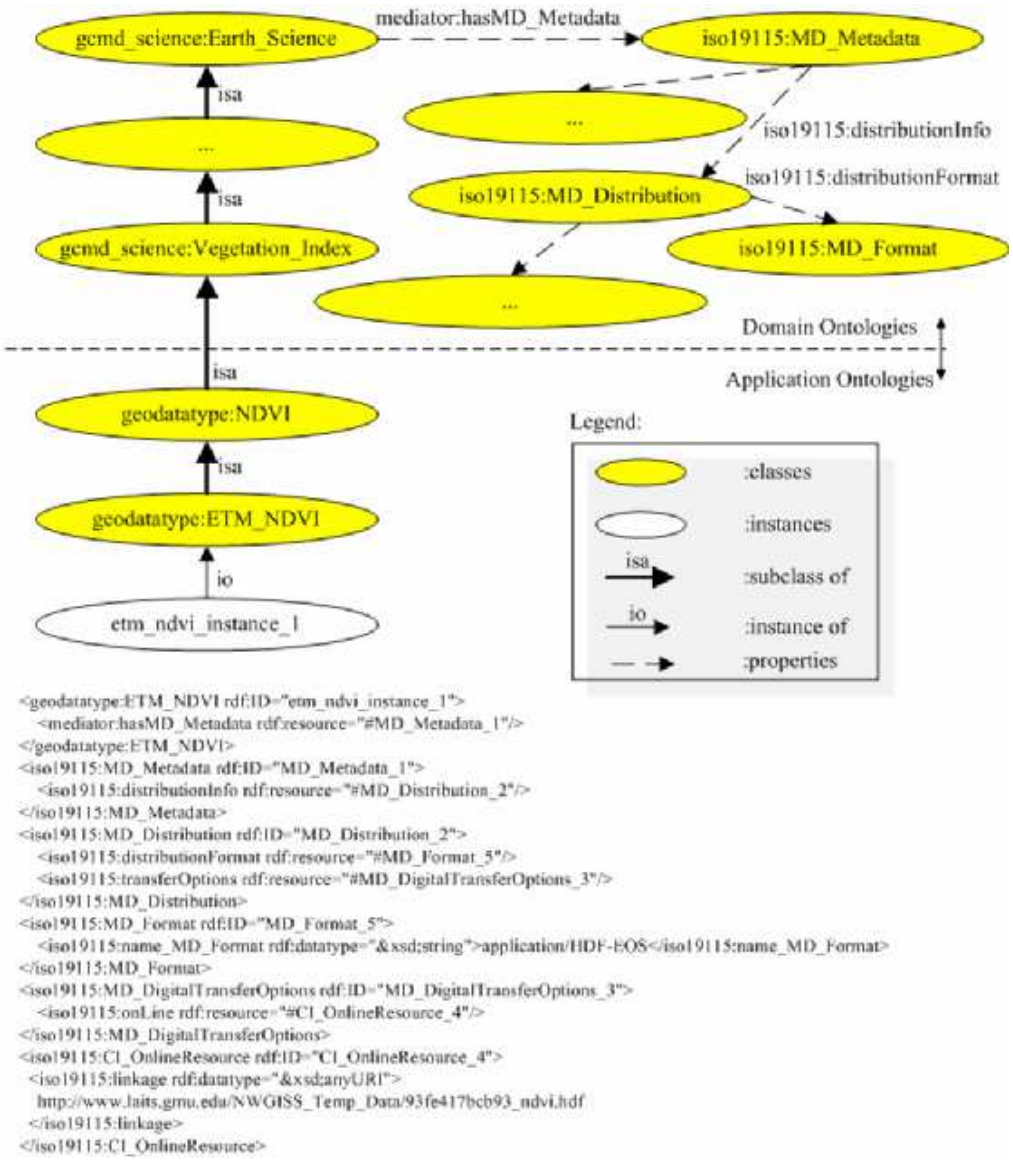

Figure 2. An example geospatial data type ontology using OWL

It is important to focus on the data type and ignore other properties during the Web service chaining. If a pair of inputs and/or outputs has the same data type, the pair is considered as a match. Their property differences will be handled later during the instantiation of the abstract Web service chain. This not only simplifies the process of service composition but also adds some flexibility.

During the second stage – instantiation, the required intermediate services, such as a cutting service to meet the boundary limits, a re-projection service to meet the project system, and a reformatting service to meet the data format, are automatically embedded in during the instantiation of the chain. The result of instantiation is expressed in BPEL4WS (Business Process Execution Language for Web Service). It can then be executed by a BPEL (Business Process Execution Language) engine.

BACKWARD REASONING

Once each individual Web service is semantically defined and described in a standard ontological language, it is possible to automatically harvest or register all the Web services. Their semantic expressions can be used with a semantic-aware reasoner. At this stage, backward reasoning is used to derive the abstract Web

service chain. Figure 3 shows the algorithm. Figure 4 gives a simple sample process to complete composition through a backward reasoning algorithm. Five recursions are needed to complete the reasoning for this simple example.

GP – goal geospatial product
AS – all data source in a data repository
RS – all Web services in a catalog or a federation of catalogs
Each rule in RS can be expressed as ID (data inputs) =R (rule)=>OD(data outputs)
If GP is directly retrievable from AS, STOP-with-success.
Else find all rule subset SRS from RS whose conclusions matches GP
For each rule R in SRS
Set ID to be the new sub-goals GPS
Run for each GP in GPS
STOP-with-fail
Else backtrack one level

Figure 3. Backward Reasoning Algorithm

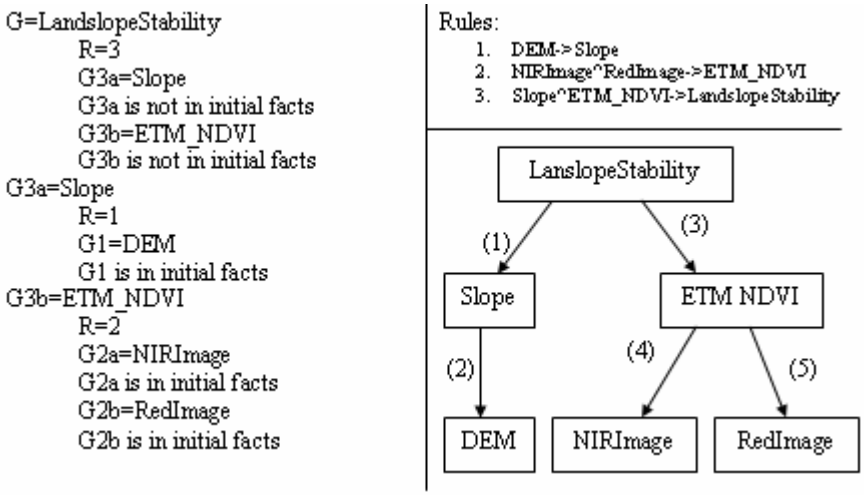

Figure 4. An exemplar Web composition process using backward reasoning

SCENARIOS

Two scenarios were examined using the backward reasoning algorithm. One is the derivation of a landslide susceptibility map from geospatial data served through WCS. Another is the presentation of live air quality using diverse Web-based geospatial services.

Landslide susceptibility

Data-driven backward reasoning was applied to chain the Web services necessary to generate landslide susceptibility. There are possibly many landslide susceptibility models. In the catalog service, we registered one model that takes four input parameters – slope, aspect, land cover, and NDVI. Figure 5 shows the result of composite service in a diagram. Once this abstract model is built, extra Web services are added to convert projection, change format, and associate default parameters. These were done automatically by the reasoner during the instantiation stage. The result is a BPEL process to be executed by the in-house BPELPower engine, a BPEL execution engine developed at the Center for Spatial Information Science and Systems, George Mason University.

The model is constructed properly if there are lots of available Web services even though there may be alternative services that match the post-conditions and outputs. One special case is that some Web service is generic. For example, the re-projection Web service is also registered in the catalog service. It takes one data type, such as slope, and produces the same data type but in another projection. The matching is based on data type only and delays the handling of the detailed property differences. When the reasoner finds

such a Web service, the reasoner ignores the inclusion of the service. The reason is that such services do not change the state of the system when only data types are considered without comparing their properties. This property is consistent with the design of the reasoner.

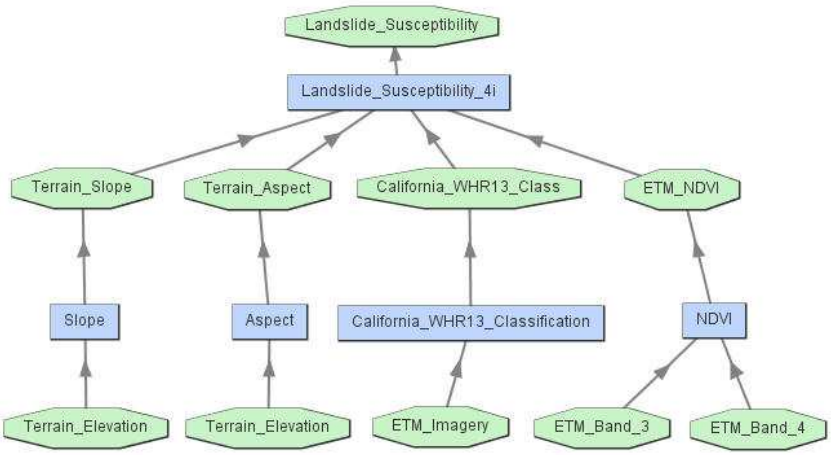

Figure 5. A composite model to extract landslide susceptibility

Live Air Quality

Production of a live air quality map was a complicated case. The goal is to analyze the sources and transportation of pollutants from Canadian forest fires to USA. The data are fed in live through sensor Web and geospatial data services, such as WCS, WFS, and WMS (Web Map Service). The final result integrates surface visibility and meteorological conditions from surface meteorological sensor networks, MODIS satellite data, and political boundaries. Several services were used to convert and portray the data. Figure 6 shows the abstract models built for this case. Figure 6, panels (a) and (b) show the manual process in building such a composition. Several data types needed to be subsumed to more specific categories to enable the final successful construction of workflow instances.

The backward reasoning approach is limited in dealing with an integration of Web services that involves many generic services. These generic services need to be handled in the abstract model construction stage. They cannot be delayed to the instantiation stage. One solution is to associate these generic Web services with specific meanings. Once that association is made, these generic Web services are specified and can be used in the composition by the reasoner. Figure 6 (c) shows the result. The resulting chain consists of a series of actions, which are live accessed to sensor networks and to other Web services. It first accesses a point monitoring data service to retrieve current air quality (AIRNOW, SURF_MET). The hourly data are then aggregated into daily averages. The result is portrayed as an image. Corresponding satellite data are retrieved. These data are stacked up together and presented to the user on the fly.

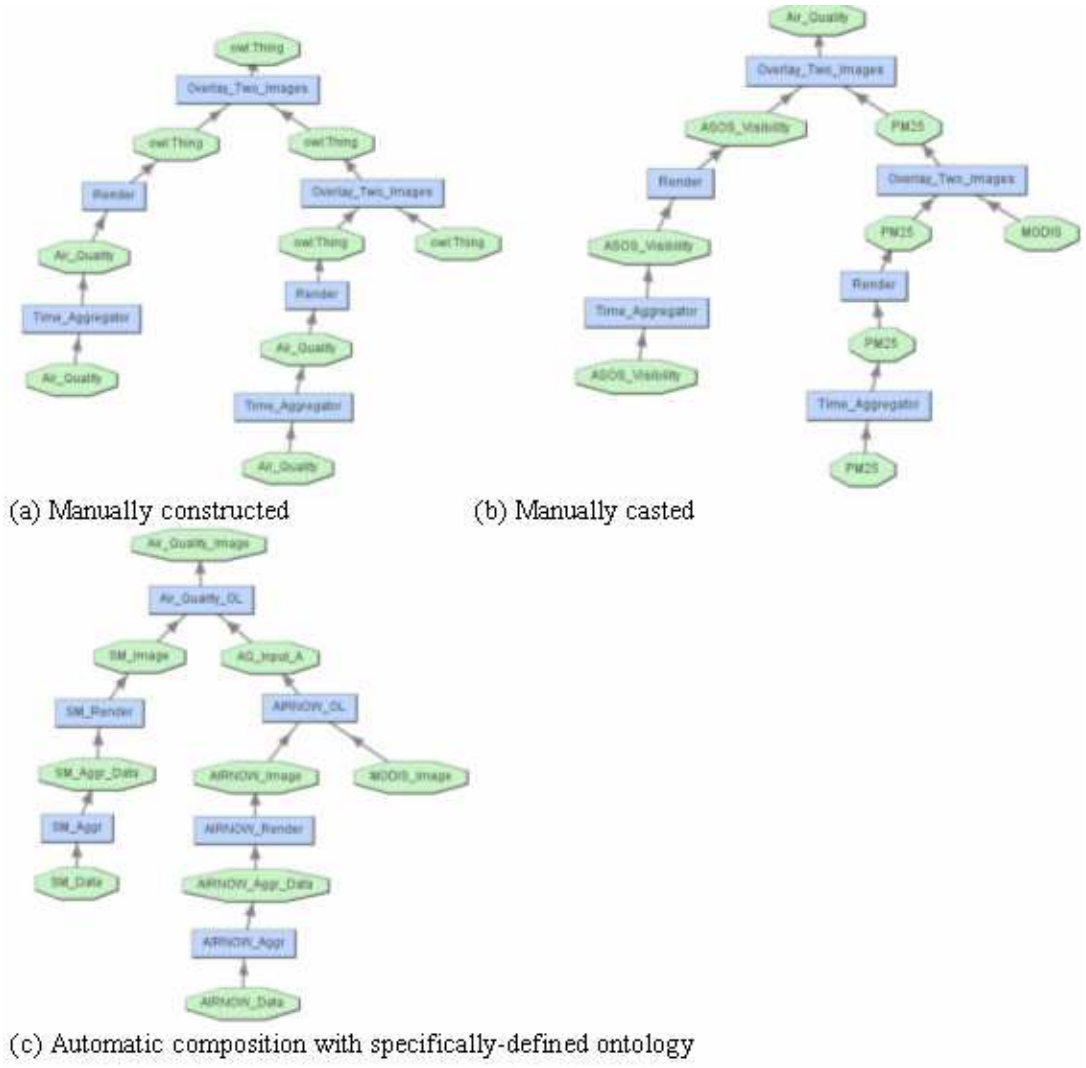

Figure 6. Composite geospatial Web service to present live air quality data

RELATED WORK

Invasive Software Composition (ISC) is an early example of the automatic service composition. It unifies view-based component development, generic component templates and aspect-oriented development. The key is that a software component is no longer treated as a "black box", but is opened up as a "gray box" with a specific interface for composition(Aβmann, 2003).

Many techniques have been used for Web service composition. Planning is one of the most popular used methods. Different planning techniques have been applied in the composition of Web services, including theorem proving, e.g. Prolog (McIlraith and Son, 2002; McIlraith et al., 2001), hierarchical task planning(Wu et al., 2003), and forward chaining with backward chaining for query (Constantinescu and Faltings, 2004; Constantinescu et al., 2004).

Rule-based systems have also been used in aiding the composition of Web services. The forward chaining technique was one of the first approaches attempted (Thakkar et al., 2002). The resulting system monitors the state transition between each action added. It starts with an empty set. At the first step, it adds all the binding patterns that have required inputs from the list of inputs. At the second step, it refreshes the list of outputs and list of inputs by adding all the new outputs from the selected patterns. At the third step, it

examines the list of all requested outputs against these presented in the list of available outputs. The following criteria are used to continue or to stop the reasoning:

(1) If all required outputs present, the process exits successfully.
(2) If there is no new output added to the list of available outputs, the process exits with failure return.
(3) If the list of required outputs is not completely met, the process loops again from step one.

Recently Web service composition has been developed along with the development of semantic Web technology and ontology technology. Traverso and Pistore (Traverso and Pistore, 2004) used OWL-S process models as the bases to describe individual services. A planner was used to compose Web services with nondeterministic and partially observable state transition systems. The generated plan was then converted to they type of process described in BPEL4Ws.

The project described here combines the rule-based system and the lately development of semantic Web technology in the automation of geospatial Web service composition. Backward reasoning is adopted, because it is a goal-driven approach that is more sufficient than forward reasoning in dealing with geospatial service composition.

CONCLUSIONS

An approach towards automatic composition that combines ontology technology with rule-based systems has been developed. Backward reasoning has been successfully adopted in the automation of geospatial Web services with semantic Web technology. The approach is data-driven. Two scenarios were discussed in detail. Trials showed that the composition could be done automatically if the proper ontology is built. One limitation for the data-driven approach is the lack of semantic representation for the algorithm itself. This leads to ambiguity when the services are generic. One special case is that one service takes in one data type and outputs the same data type. If these generic services need to be chained during the model construction stage, specific ontological representations are necessary. Alternative methods to consider the use of semantic information about the algorithm itself in the composition may be explored in the future.

Acknowledgements

This study is sponsored by National Geospatial-intelligence Agency (NGA) University Research Initiative (NURI) (HM1582-04-1-2021, PI: Dr. Liping Di).

Reference:

Aβmann, U., 2003. Invasive software composition Springer, Berlin; New York 334 pp.

Badea, L., Tilivea, D. and Hotaran, A., 2004. Semantic Web reasoning for ontology-based integration of resources. In: H.J. Ohlbach and S. Schaffert (Editors), Workshop on Principles and Practice of Semantic Web Reasoning LNCS. Springer-Verlag Berlin Heidelberg, St Malo, France, pp. 61-75.

Brodaric, B. and Gahegan, M., 2006. Representing geoscientifc knowledge in cyberinfrastructure: some challenges, approaches, and implementations. In: A.K. Sinha (Editor), Geoinformatics: Data to Knowledge. The Geological Society of America, Boulder, Colorado, USA, pp. 1-20.

Constantinescu, I. and Faltings, B., 2004. Large scale, type-compatible service composition, IEEE International Conference on Web Services IEEE Computer Society, San Diego, CA, USA.

Constantinescu, I., Faltings, B. and Binder, W., 2004. Type based service composition, 13th World Wide Web Conference, New York City, USA.

Di, L., 2004. Distributed geospatial information services - architectures, standards, and research issues, XXth ISPRS Congress. International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences. ISPRS, Istanbul, Turkey, pp. 187-193.

Di, L., 2005. A framework for developing Web-service-based intelligent geospatial knowledge systems. Journal of Geographic Information Sciences, 11(1): 24-28.

Di, L. and McDonald, K., 1999. Next generation data and information systems for Earth sciences research, First International Symposium on Digital Earth. Science Press, Beijing, China, pp. 92-101.

Kona, S., Bansal, A., Gupta, G. and Hite, T.D., 2007. Semantics-based Efficient Web Service Discovery and Composition, The University of Texas at Dallas, Richardson, Texas, USA.

Martin, D. et al., 2004. OWL-S: Semantic Markup for Web Services W3C.

McIlraith, S.A. and Son, T.C., 2002. Adapting Golog for composition of semantic Web services, Eighth International Conference on Principles of Knowledge Representation and Reasoning Toulouse, France

McIlraith, S.A., Son, T.C. and Zeng, H., 2001. Mobilizing the semantic Web with DAML-enabled Web services, Second International Workshop on the Semantic Web, Hongkong, China.

Thakkar, S., Knoblock, C.A., Ambite, J.L. and Shahabi, C., 2002. Dynamically composing web services from on-line sources, AAAI-2002 Workshop on Intelligent Service Integration, Edmonton, Alberta, Canada.

Traverso, P. and Pistore, M., 2004. Automated Composition of Semantic Web Services into Executable Processes In: S.A. McIlraith, D. Plexousakis and F. van Harmelen (Editors), The Semantic Web – ISWC 2004. Lecture Notes in Computer Science. Springer Verlag, pp. 380-394.

Wu, D., Parsia, B., Sirin, E., Hendler, J. and Nau, D., 2003. Automating DAML-S Web services composition using SHOP2, 2nd International Semantic Web Conference, Sanibel Island, Florida, USA.

Yue, P., Di, L., Yang, W., Yu, G. and Zhao, P., 2007. Semantics-based automatic composition of geospatial Web services chains. Computer and Geosciences, (accepted).