

MODEL-DRIVEN GEOSPATIAL WEB SERVICE COMPOSITION

Wu Du^a, Hong Fan^{a,*}, Jian Li^a, Haibo Wang^b

^aState Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University,
Wuhan 430079, China – d.y.wu.du@gmail.com, geohfan@163.com, keenswordlj@163.com

^bSchool of Management, Hubei University of Technology, Wuhan 430072, China - wang_whb@163.com

KEY WORDS: geospatial service, model-driven composition, quality of service, service composition model

ABSTRACT:

A lot of work has been done on the geospatial service composition to support advanced processing, spatial calculation, and invoking of heterogeneous data. However, the quality of service chain is rarely considered and the process model cannot be reused. A model-driven way of geospatial web service composition is proposed in this work, the service composition is treated as an optimization problem by *GwscFlow* model and dynamic binding mechanism. The case of facility location analysis is provided to demonstrate the improvements in geospatial service composition through optimization algorithms.

1. INTRODUCTION

With the emergence and development of the Service-Oriented Architecture(SOA), web service is playing a more and more important role in the data interoperability and cross-platform share. The geospatial web service, which is accessible in the spatial data infrastructure, provides a unified and easy way to decouple modules in GIS applications. A series of standards and specifications for geospatial web services, including web map service(WMS), web feature service(WFS), web coverage service(WCS), web processing service and so on, have been put forward to facilitate the geo-data interoperability by the Open Geospatial Consortium(OGC).

The composition service, which is composed by atomic services according to certain patterns, is able to accomplish complicated tasks that cannot be done by single atomic service. A lot of work has been done on the methods of web service composition. According to the fields that the composition method belongs to, these methods can be classified as follows: methods based on workflow, eg. Fabio Casati et al. establish the eflow platform to orchestrate and manage services; methods based on the artificial intelligence, eg. Evren Sirin, et al. decompose the task of service composition based on the shop2 planner that transform the problem of service composition to a planner problem; methods based on the graph theory, Snehal Thakker proposes the backward chaining algorithm, Lerina Aversano proposes the forward chaining algorithm, and Joonho Kwon et al. put forward a two-phase method by combining the backward chaining and forward chaining algorithms. The workflow method needs human assistance in the composition process for the low-level automation. Methods based on AI is able to complete the service composition automatically in certain extent, however, the task has to be decomposed in advance. The graph based methods is apt to cause problems of combination explosion.

Recently semantics based service composition has been adopted in the geospatial web service with the development of the semantic web. Peng Yue et al. put forward a spatial data-driven way to construct geospatial service chain, which describes the data through ontology and represents the services through owls. Friis-Christensen proposes a framework of distributed geographic information processing to manage the risks of forest fire, and summarizes crucial problems in

constructing service chain by OGC services. Sergio A.B. Cruz et al. focus on the spatial data and come up with service composition based on AI planning, which describes the requirements of spatial data quality in rules. Farnaghi proposes an automatic solution for composing OWs for disaster planning, which annotates the OGC services by SAWSDL and wsmo-lite, and demonstrate that the proposed automatic composition approach can improve disaster management process.

The issues that geospatial web service composition handles include complicated processing, spatial calculation, and invoking of heterogeneous data. The existing problems are as follows: first, the quality of geospatial service is rarely considered during the construction of service chain; second, the process model of geospatial service chain cannot be shared and reused, since the process model generated by automatic composition is dynamic and does not have a fixed process model. The processing of geographic information usually follows some routine workflow, while in the execution of the workflow, an atomic process in the procedure can be replaced by another process that employ a new algorithm.

In this work, we propose model-driven composition architecture for the geospatial service, and demonstrate the application by the classic facility location analysis. Furthermore, the evaluation criteria of geospatial service are provided to promote the construction of service chain. The process model of composite service is able to be shared and reused because it's independent on the specific language that describes the web service, such as wsd, owls, wsmo and so on.

2. SERVICE COMPOSITION MODEL

2.1 The *GwscFlow* model

GwscFlow model, an abstract and extensible workflow model, is designed to describe the composition of geospatial service. The core elements of *GwscFlow* model include *AbstractGwscElement*, *FlowNode*, *GwscFlow*, *Transition*, *WebService* and *FlowGate*. All elements are inherited from *AbstractGwscElement*, and all nodes in the model are inherited from *FlowNode*. *Transition* represents the flow from one node to another, and *FlowGate* represents the branch or join of flows, which is essential for constructing the process flow of

*Corresponding author

GwscFlow model, because there're no so called control structures in *GwscFlow* model like BPEL4WS, all control structures are implemented indirectly through the *FlowGate* nodes. *FlowJoin* is the only node where flows are merged, while *FlowSplit* is the node where flows are split, they are all inherited from *GwscFlow*, which simplifies the modeling process. *GeoInformationService*, *GeoProcessingService* and *GeoWorkflowService* are all inherited from *WebService*, which respectively represent the geo-info service, geo-processing service and geo-workflow service.

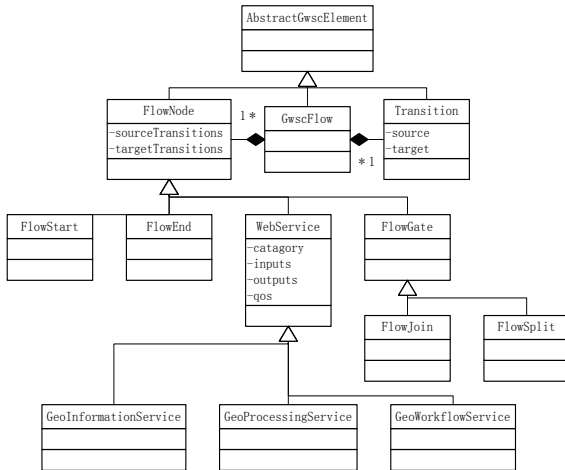


Figure 1. The uml graph of core *GwscFlow* elements

WebService, as the superclass of all geospatial services, is neutral to specific fields, and it includes four basic attributes: category, inputs, outputs and qos. We mainly focus on five evaluation criteria about non-functional properties: execution time, cost, reputation, reliability and availability, according to the theory of service selection(Zeng LZ, 2004). The reliability refers to the ratio of successful calls to all, and the availability refers to the degrees that services can be accessed since some services are not available in 24 hours. The category relies on specific fields.

2.2 dynamic binding of geo-processing services

Every *WebService* defined in the *GwscFlow* model is an abstract web service, which will be bind to a specific web service during the actual construction of the service chain. A specific web service is selected from alternatives through the matchmaking results of the inputs and outputs. The most commonly used way for service matchmaking is matching the service I/O by the concepts hierarchy of ontology(Massimo Paolucci, 2002). The drawback is that there're only four result types, including *exact*, *plugin*, *subsumes* and *fail*, results in the same type cannot be compared again. Ayse B. Bener et al. propose another way to match service IOPE through bipartite graph, in which the preconditions and effects are matched by swrl rules, and the results are continuous. It's not effective to math geospatial services based on just inputs and outputs, actually the main function of service reflected in the service category. So we match the geo-processing services by service inputs, outputs and categories, based on the bipartite graph method and semantic distance.

- (1) Semantic descriptions of geo-processing services. Geo-processing services are annotated by owls and the owls ontology is extends to support the quality of service(QoS)(Wu Du, 2010). Geo-processing services are classified according to the ISO 19119.
- (2) Matchmaking based on bipartite graph. Any pair of

concepts in the ontology has a semantic distance, with a range of [0, 1]. The parent concept is denoted by p, and the child concept is denoted by c. The semantic distance between concept p and c is calculated as follows:

$$weight_{p,c} = 1 / \text{number of direct subclasses}_p \quad (1)$$

The semantic distance between p and c is the product of all edge weights.

$$weight_{a,b} = weight_{a,t} * weight_{t,k} * \dots * weight_{x,y} \quad (2)$$

- (3) The abstract service is denoted by R, and the service to be matched is denoted by O, a, b refers to the concepts from R and O, the value of matching is calculated by the following formula:

$$V_{R,O} = \sum_{a \in R, b \in O} weight_{a,b} * subsumes_{a,b} \quad (3)$$

The *subsumes_{a,b}* is valued as follows: exact=1, plugin=0.6, subsumes=0.4, fail=0.

2.3 the platform of model-driven service composition

The platform of model-driven service composition, based on *GwscFlow* model, includes four layers, the *GwscFlow* model layer, the model-driven layer, the application layer and the optimized algorithm layer.

The *GwscFlow* model is at bottom of the platform architecture, independent from specific flow design frameworks. The model-driven layer includes visual tools and frameworks for developments. The graphical editor framework(GEF) is used to build a bridge from service composition model to workflow designers. The application layer consists of three parts: workflow pattern, dynamic bindings and data simulation. The *GwscFlow* model is transformed to workflow patterns, including sequence construction, parallel branches and synchronizing join construction. The service composition problem is transformed to the dynamic binding problem, in which the abstract service node is bound to specific service by matchmaking algorithms based on bipartite graph. Data simulation refers to the module that generated simulated data and services for meta-heuristic algorithms. Specific service composition algorithms are implements in the optimized algorithm layer, including genetic algorithm, ant colony optimization and local optimization.

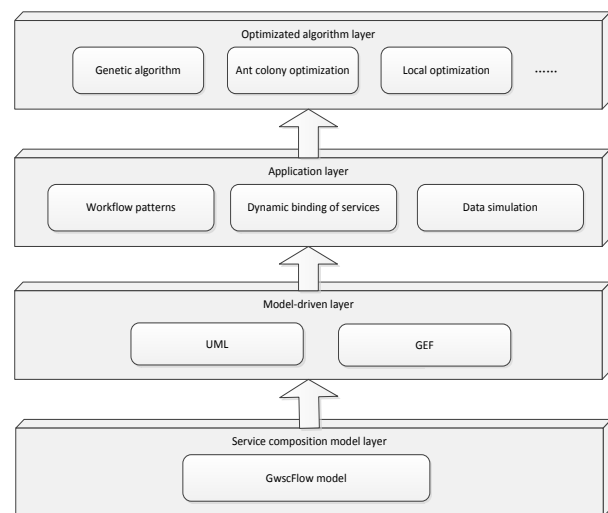


Figure 2. Architecture of the platform of model-driven service composition

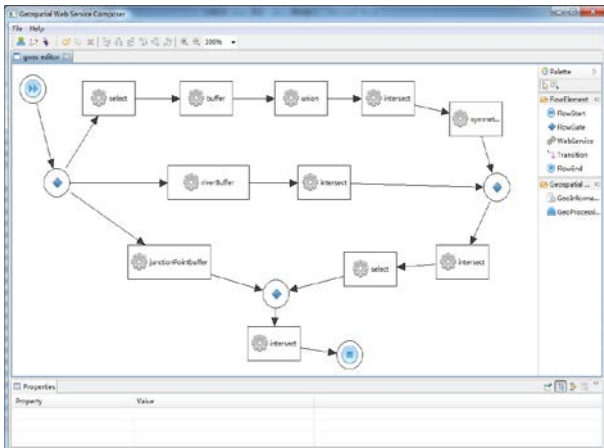


Figure 3. The geospatial web service composer

2.4 mappings between GswcFlow model and workflow patterns

The *GswcFlow* model need to be transformed to workflow patterns for two reasons: firstly, Web Service Choreography Description Language such as BPEL4WS and OWLS support control flows, which means that the *GswcFlow* model can be described by existing frameworks; secondly, the evaluation of composite service quality relies on the workflow patterns, which is also employed in the experiments in section 3. The mapping rules between *GswcFlow* model and workflow patterns are as follows:

- (1) The *GswcFlow* model corresponds to the three patterns in workflow: sequence construction, parallel branches and synchronizing join construction.
- (2) The model corresponds to sequence construction if there's no *FlowGate* element; when the *FlowSplit* is as the start node and the transition numbers is greater than 1, the model corresponds to parallel branches. When the *FlowJoin* is as the end node and the transition numbers is greater 1, the model corresponds to synchronizing join construction. When the *FlowGate* is as the start node or end node, and the transition number is equal to 1, then the *FlowGate* model corresponds to sequence construction since the *FlowGate* acts just as a connector.

3. A USER CASE

3.1 The location analysis problem

The location analysis of sewage treatment plant is a classical problem in GIS, we model the location analysis process from the point of geo-processing service, aimed to select the suitable parcels that meet following conditions: (a)the altitude should be below 365m to cut down the cost; (b)outside the flooded area; (c)around the river within 1000m to cut down the pipeline cost; (d)away from urban residents at least 150m; (e)better in exploitable vacant lot, to cut down the expropriation cost; (f)around sewage concentrated center within 1000m.

3.2 service composition model of location analysis

Five types of services are used in the location analysis, including buffer, intersection, selection, symmetrical difference and union.

(1)According to the condition a, get 1000m buffer area of the river layer, denoted as *riverAroundArea*. Overlay the *riverAroundArea* by lower altitudes, then we get the areas that

are around the river and below 365m, denoted as *lowlandAroundRiver*.

(2)Select the residents and do the buffer analysis, denoted as *ResidentAroundArea*, then union the *ResidentAroundArea* and flooded area, the result is denoted as *residentUnionFloodArea*, which is overlaid by urban parcels with the result *residentFloodCityParcel*. At last, we can get the urban parcels that meet the condition b and d through the operation *symmetricalDifference* by *residentFloodCityParcel* and urban parcels.

(3)We can get the parcels that meets conditions a, b, c and d by overlaying the results from step 1 and step 2, then select the vacant lot out of them. Do the buffer analysis of sewage concentrated center, and overlay the result with vacant lot, then we get the result parcels which are suitable to build the sewage plant.

The *WebService* models that the entire flow need is as follows, and the inputs and outputs correspond to the specific concepts in the ontology which is demonstrated in fig5:

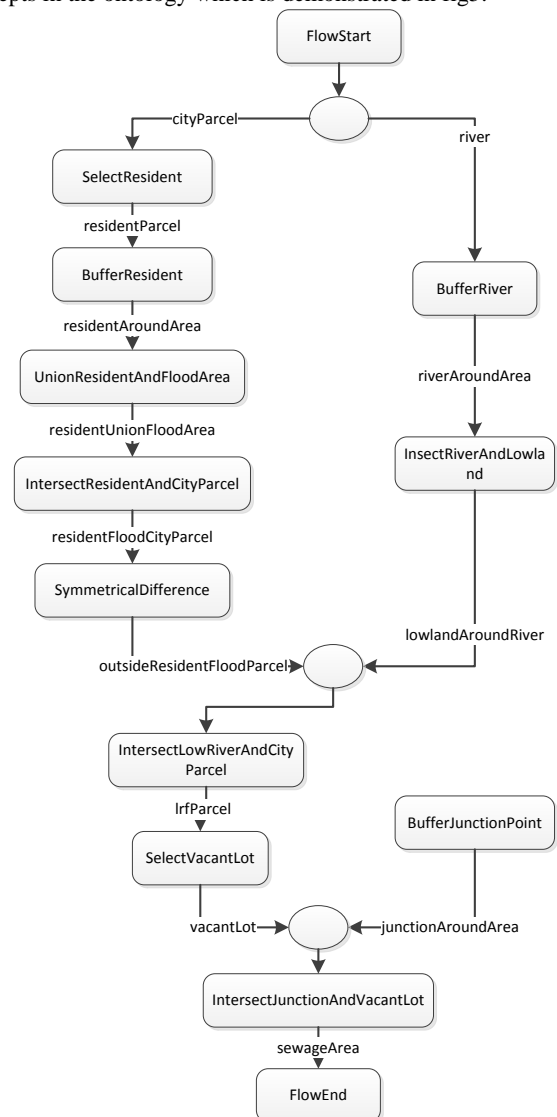


Figure 4. The composition model of location analysis

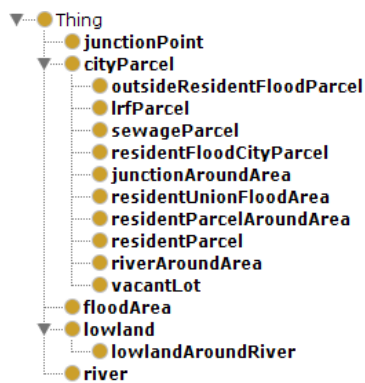


Figure 5. the application ontology in the location analysis

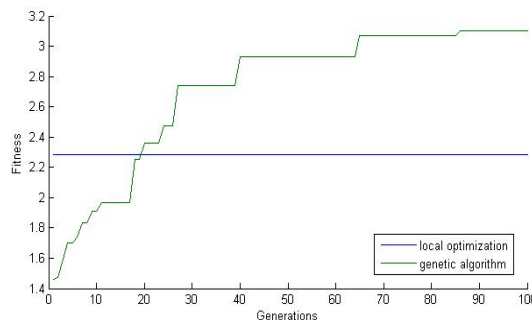


Figure 5. The result of composition optimization upon generic algorithm

3.3 service composition optimization by considering QoS

The geo-processing service chain is constructed through *GwscFlow* model and dynamic binding strategies, however, there's still room for improvement since the quality of service is not considered during the composition process. In particular, how to select specific service that guarantees the best quality of the entire service chain when there're alternative services in *WebService* node. Michael C.Jaeger put forward a service quality model that evaluates the entire service chain upon the workflow patterns. Liangzhao Zeng presents a middleware platform which addresses the issue of selecting web services in a way that maximizes user satisfaction over QoS attributes; Gerardo Canfora uses the genetic algorithms to complete the QoS-aware composition, and compare with the integer programming method. In this work the genetic algorithm is adopted to optimize the service composition process by maximizing the quality of service chain.

1, Genome encoding. Each *WebService* node corresponds to an item in genome encoding, the value is the serial number in the alternative services set. The genome length equals to the number of *WebService* nodes.

2, the selection operation, in which two individuals are selected from the population based on the roulette algorithm; the mutation operation, in which the value of a position in genome encoding is changed to another; the crossover operation, in which two service chains exchange parts of genome encoding according to certain crossover rate.

3, the population size is 100, the mutation rate is 0.3, and the crossover rate is 0.7, the evaluation function is as follow:

$$fitness = (t_1 * wschain_{availability} + t_2 * wschain_{reliability} + t_3 * wschain_{reputation}) / (t_4 * wschain_{cost} + t_5 * wschain_{time})^p \quad (4)$$

For every *WebService* node there're 10 alternative services that are generated randomly by data simulation module. The experiment is implemented upon the open source watchmaker framework, which speeds up the algorithm convergence via elitist strategy. The comparison result between basic local optimization and genetic algorithm is demonstrated in fig4.

4. CONCLUSION

The *GwscFlow* model is presented to model the service composition process upon model-driven theory. Compared to common web service that conforms to w3c standard, the geo-processing service is more complicated and relies more on the quality of service chain, therefore, the genetic algorithm is employed to optimize the service chain by satisfying the quality constraints. The bipartite graph is used to match the geo-processing services, however, the attributes of *WebService* node is incomplete, only the inputs, outputs, category and QoS have been considered so far. We focus on the geo-processing service, not consider other data provider service such as WMS, WFS and WCS, since the geospatial service is more complex than common web service. The extension of attributions of *WebService* node and the service composition including both geo-processing service and data provider service will be researched in future.

Acknowledgements

This work is supported by national natural science foundation of China(40971243) and fundamental research funds for the central universities(2011161902020011).

References

- Aversano, L., G, 2004. Canfora and A. Ciampi. An algorithm for Web service discovery through their composition. in IEEE International Conference on Web Services.
- Ayse B. Bener, Volkan Ozadali, Erdem Savas Ilhan, 2009. Semantic matchmaker with precondition and effect matching using SWRL. *Expert Systems with Application* 36(5):p. 9371-9377.
- Cruz, Sérgio A.B, Antonio M.V. Monteiro, and Rafael Santos, 2012. "Automated Geospatial Web Services Composition Based on Geodata Quality Requirements." *Computers & Geosciences* 47: 60–74.
- Deng, S., et al., 2007. Automatic Web service composition based on backward tree. *Ruan Jian Xue Bao/Journal of Software*. 18(8): p. 1896-1910.
- Fabio Casati, S.I.L.J., 2000. An open, flexible, and configurable system for service composition, in *Advanced Issues of E-Commerce and Web-Based Information Systems: IEEE Computer Society Washington, DC, USA*, p. 125.
- Farnaghi, Mahdi, and Ali Mansourian, 2013. Disaster Planning Using Automated Composition of Semantic OGC Web Services:

A Case Study in Sheltering. *Computers, Environment and Urban Systems* 41: 204–18.

Friis-Christensen, Anders, Roberto Lucchi, Michael Lutz, and Nicole Ostländer, 2009. "Service Chaining Architectures for Applications Implementing Distributed Geographic Information Processing." *International Journal of Geographical Information Science* 23, no. 5: 561–80.

Gerardo Canfora, M.D.P.R., 2005. An approach for QoS-aware service composition based on genetic algorithms, in *GECCO '05 Proceedings of the 2005 conference on Genetic and evolutionary computation*: ACM New York, NY, USA

Jaeger, M.C., G. Rojec-Goldmann and G. Muhl, 2004, QoS aggregation for Web service composition using workflow patterns. *EIGHTH IEEE INTERNATIONAL ENTERPRISE DISTRIBUTED OBJECT COMPUTING CONFERENCE, PROCEEDINGS*, p. 149-159.

Kwon, J. and D. Lee, 2012. Non-redundant web services composition based on a two-phase algorithm. *DATA & KNOWLEDGE ENGINEERING*, 71(1): p. 69-91.

Massimo Paolucci, Takahiro Kawamura, et al., 2002, Semantic Matching of Web Services Capabilities. *The Semantic Web — ISWC 2002 Lecture Notes in Computer Science Volume 2342*, pp 333-347

Snehal Thakkar, C.A.K.J., 2002. Dynamically Composing Web Services from On-line Sources, in *In Proceeding of the AAAI-2002 Workshop on Intelligent Service Integration*.

Sirin, E., et al., 2004. HTN planning for Web Service composition using SHOP2. *Web Semantics: Science, Services and Agents on the World Wide Web*, p. 377-396.

Wu Du, Hong Fan, 2010. An automatic service composition algorithm for constructing the global optimal service tree based on QOS. *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, p.3976 – 3979.

Yue, P., et al., 2007. Semantics-based automatic composition of geospatial Web service chains. *Computers & Geosciences*, 33(5): p. 649-665.

Zeng, L.Z., et al., 2004. QoS-aware middleware for Web Services Composition. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*. 30(5): p. 311-327.