Remote Sensing Service chain Self-Evolution Method

Haifeng Li
Supervised by Qing Zhu
State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University
430079 Wuhan, China
hfli6135@lmars.whu.edu.cn

Abstract. In order to facilitate continuous and quick adaptation to the change of environment and use requirement, this paper presents a self-evolution method for the remote sensing information service chain to keep effective and robust of service chain with lesser modification, and to maximize all users' utilization under concurrent user requirements. According to the driver forces of service chain change, we partition self-evolution method into three levels: (1) a fuzzy semantic based relatedness analysis and min-conflict heuristic based service chain reconfigure methods to adapt to user requirement change; (2) a sensitivity analysis and robust optimization based method to keep maximum stability of service chain in dynamic environment; (3) a non-cooperative game approach for multi-service chain cooperation optimization under concurrent tasks concurrent condition.

1 Introduction

As satellites of Earth Observing System (EOS) currently beam down several hundred terabytes annually, the inconsistent between powerful data instruments and incompetent data process become ever more standing, which are making this field "data-rich but analysis-poor" [1]. The key reason lead to this is not that we lack of applications to handle with these data, but mechanisms how to aggregate the applications which distributing in internet extensively (hence be looked as remote sensing services) together and cooperate them to satisfied the need of the data analysis. This is so-called remote sensing service chain[2] through service composition. Compared with generic Web service composition, remote sensing services have some typical features as follows:

Complex in user requirements, for examples: real-time monitor for forestry fire, coast, and flood; concurrent in user requirements, for instance, in Sichuan Wenchuan Earthquake, we must evaluate earthquake damage and monitor coast, landslides, and barrier lakes at the same time.

Rich in data dimensions. Data in remote sense with dimensions of spatiality, temporal, image resolution, sensor type, and image spectrum, makes it harder to be descript and more complicated in processing flow. What's more, mass remote sensing images make the service chain more sensitivity to response time.

Complicated in remote sensing processing. Remote sensing service composition has been constrained by more strict process semantic; Computation-intensive feather in remote sensing also make remote sensing processing more time consumed.

So, remote sensing service chain should be flexible enough to effectively adapt to fast change of use requirements and environment, through frequently refine their structure. The existent methods to generate services have some disadvantages as following:

Lack of mechanism to adapt user requirements change via local reconfigure at function level which are known as abstract services[3]. The state-of-art service composition approaches[4] are facing more and more serious bottlenecks of effectiveness and stability, since new service chain must be generated from “scratch” for each requirement. Those methods are also known as “first principle”. Distinguish with it, another way is how to make use of relativity between remote sense service chains and reuse knowledge about similar, already solved problems successfully. This methods are always known as “second principle”[5], which aim to make service chain generate more effectively and execute more stably. Although there are some researches generate service chain by case based reasoning[6, 7], but they all do not take into account strict process constraints in remote sense. What’s more how to measure similarity between cases accurately and to refine service chain effectively are still open questions.

Lack of robust adapting to dynamic environment at capability level known as concrete services[3] which usually modeled as QoS constraint based optimization. The service chains are more sensitivity to services and transport network performance, because data-intensive and computing-intensive are essential features in remote sensing. A small perturbation in QoS dimension of services and transport network will make former optimization solution becomes infeasible. There are many researches dynamic modify service chain through runtime monitor and re-planning technology[8]. But, because of high dynamic and uncertain of services and transport network in nature, the dynamic modification may be too frequency, and lead to unstable and decrease of performance of service chain. Therefore, we still are short of quantization model to estimate the influence of QoS perturbation on service chain. The mechanism how to keep service chain robust in dynamic and uncertain environment is unclean.

Lack of optimal mechanisms to deal with concurrent user requirements. The existing optimal composition approaches search optimization solution[9] under QoS constraints (such as response time, cost, stability and available) via “selfish” way. Yet, these methods only take single used requirements into account, not adapt to applications like remote sensing emergency and disaster response where concurrent task happened frequently. Concurrent tasks competing optimal services lead to conflict problem and decreased performance of all service chains, which are known as “tragedy of the commons”. A key problem here is how to reduce the conflict cause by concurrency tasks to make all service chain reach optimization at the same time.

In conclusion, in the face of high dynamic environment and user requirements, and high concurrent of user requirements, the challenge of remote sensing service chain generation is: how service chains adapt to user requirements and environment to keep effective and robust of service chain with lesser modification and how to implement multiple service chain cooperation optimization under concurrent tasks to maximize all user’s utilization. Hence, we put forward a novel self-evolution method to solve it.

2 Remote Sensing Service chain Self-Evolution Method

The basic conception behind remote sensing service chain self-evolution method is: it is a self-adaptive behavior responding to exterior dynamics factors, through frequent revise structure, function and capability of service chain, with completeness, minimization and consistency.

Exterior factors dynamics refer to user requirements, services runtime environment which including service temporarily disabled, modification of services QoS and network QoS, and so on.

Completeness, refer to if it is feasible to change from current service chain to others, then, we always can find the post-evolution service chain.

Minimization, refer to achieve the evolution process with minimum service chain changing. The minimization has two means here: maximum reuse existent service chain and least revision that establish the upper and lower limits of the sensitivity interval and find a robust solution with lesser sensitivity to dynamic environment.

Consistency, include function consistency and capability consistency, i.e. evolution process must satisfied constraints such as function constraints and QoS constraints.

We first analyze driving forces of service chain evolution to understand which factors make it change.

2.1 Driving Forces of Service chain Evolution

We classify driving forces into two categories: user which provides information requirements and preference, and runtime environment of services, shown as fig. 1.

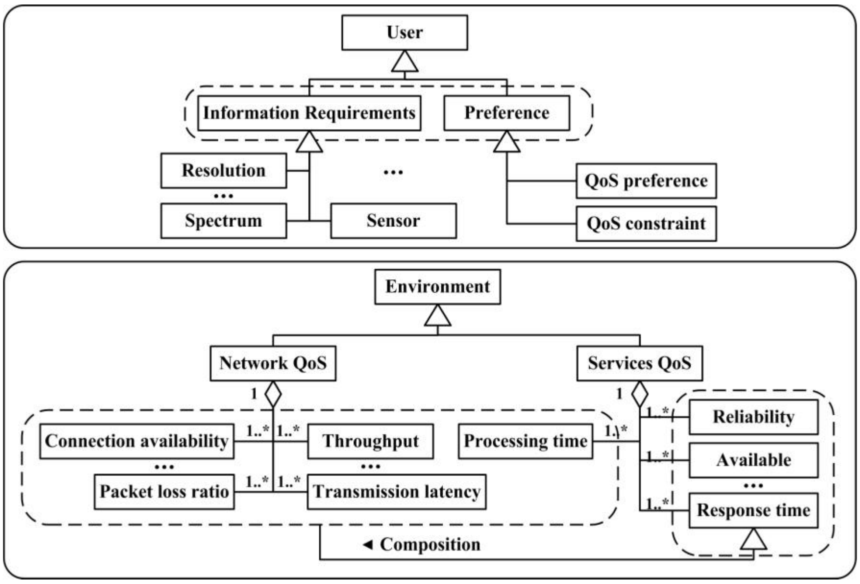

Fig. 1. Driving forces of service chain evolution

The driving forces of user are decomposed into information requirements which describe function demands and preference which describe non-function demands. The former associate with abstract and the latter associate with concretion service chain[3].

The information requirements describe the function about use demand. A typical requirement can be described as following:"2008-8-8 Beijing Olympic Country 1m geospatial resolution panchromatic IKONOS image". Another requirement changes to:"2008-8-8 Beijing Olympic Country 0.5m geospatial resolution panchromatic image". Now, the abstract service chain must be modified.

The preference describes the non-function about user demand. The preference can also divide into QoS preference and QoS constraint. The former describes how important QoS dimension means to user, the latter describes anticipant upper or lower limit of QoS dimension. A typical preference can be described as following:"response time less than ten minutes and weight equal to 0.5, cost less than 100 dollars and weight equal to 0.3, successful execution rate more than 80% and weight equal to 0.3". The non-function here mainly refer to QoS dimensions such as five dimensions model presented by[9]. The concrete service chain should modify with preference changing.

Runtime environment of services include network QoS (the response time is computed by the sum of the processing time and the transmission time) and services QoS such as the response time of service change from 20 minutes to 30 minutes. The alterations of them make performance (object function in optimization) of concretion service chain fluctuate frequently and irregularly.

What's more, concurrent tasks will lead to value of network and services QoS change more severity because of "completion of best resource".

2.2 Service chain Self-Evolution Method

We reduce self-evolution method to two levels and three hierarchies according to the driving forces motioned above, shown as table 1. The proposed research methods consist of three aspects as follows, also shown as fig. 2:

Table 1. The basic idea of service chain evolution.

Service chain level	Question	Basic idea
Single service chain	How to adapt user requirement changing	Choosing a most similarity service chain by user requirements relatedness analysis, and fast reconfiguring by local revising based on reuse knowledge about similarity, already solved problems successfully.
	How to keep service chain robust in dynamic and uncertain environment	Analysis the influence of QoS perturbation including QoS preference, QoS constraints, and services QoS on service chain; set up a robust optimization model to keep service composition optimal solution more stability.
Multi-service chain	How to make all service chain reach optimization at the same time in concurrent situation.	Modeling competition relationship between tasks by non-cooperative game, which assures maximizing all tasks' utilization under multi-task conflict condition.

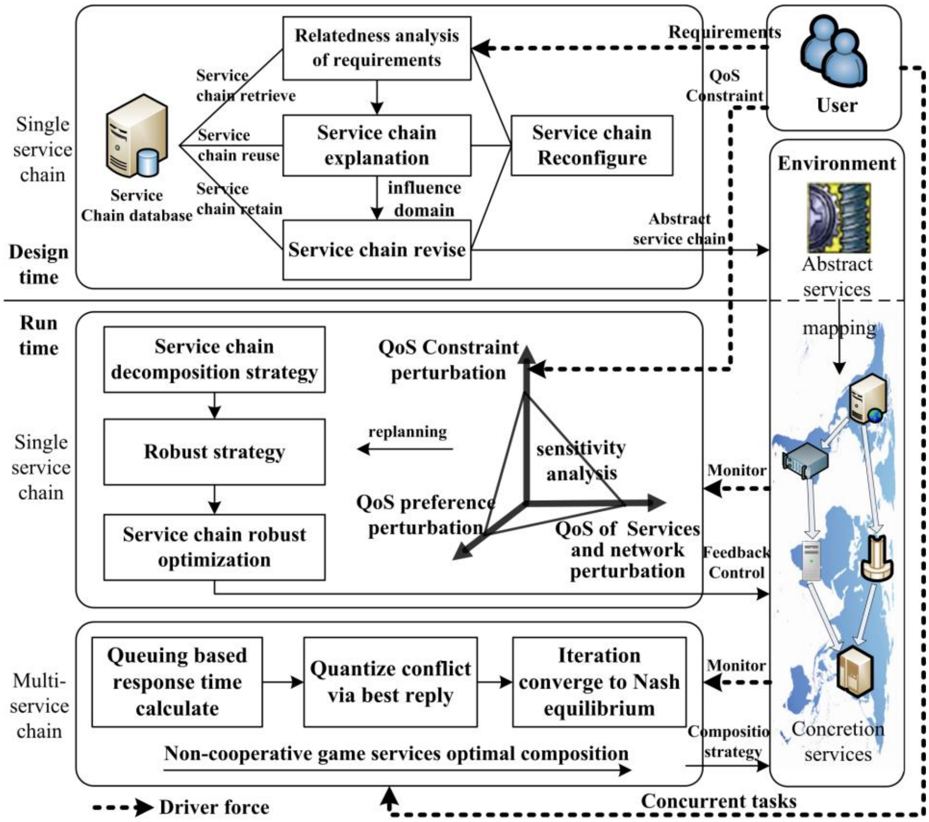

Fig. 2. Architecture of service chain self-evolution method

(1) Adapt to user requirement

We model remote sense requirements as and/or graph. To estimate relativity between two requirements, we proposal fuzzy semantic distance based method on node level, and Hausdorff distance based method on graph level.

After relatedness analyses, we accomplish service chain explanation by quantitative analysis the influence domain of each service. Finally a min-conflict heuristic based regression algorithm is presented to search a minimal influence domain solution to achieve service chain reconfigure, and prove be A*.

(2) Robust optimization in dynamic environment

We quantitative analysis the influence of QoS perturbation on service chain performance via mix integer linear programming model. Based on this, we cast QoS preference, QoS constraints, and QoS of services and network to profit parameters, left-hand side and left-hand side of constraints, respectively, and establish the upper and lower limits of the sensitivity interval through sensitivity analysis.

Then, we set up robust optimization model via minimax criterion[10] and decompose service chain to three execution stage (executed, executing, un executed) .

Based on this, we decrease service chain sensitivity to dynamic environment and reduce re-planning frequency by finding robust optimization solution.

(3) Concurrent tasks optimization

A non-cooperative game based mathematics model is proposed to analysis competition relationship between tasks through best reply function, which is defined to quantize conflict between tasks to assure each task finds optimal composition strategy adapting to other tasks'. Based on this, we present an iteration algorithm converging to Nash equilibrium, which maximizing all task's utilization under multi-task conflict condition.

3 Conclusion

Remote sensing service chain self-evolution method is a self-adaptive behavior to exterior factors dynamics, through frequent revise structure, function and capability of its. We have done some experiments on user requirements change and tasks concurrent scenarios and our methods show good performance in former and good convergence and better practice utility of all tasks in latter.

Next step, we will focus on our sensitivity analysis and robust optimization based method to test capability of keeping service chain stability in dynamic environment.

References

1. Clery, D. and Voss, D.: All for one and one for all. Science, 308(6):757, (2005)
2. Alameh, N.: Chaining geographic information Web services. IEEE Internet Computing, 7(5):22-29, (2003)
3. Canfora, G., Penta, M.D., Esposito, R. and Villani, M.L.: A framework for QoS-aware binding and re-binding of composite web services J. Syst. Softw., 81(10):1754-1769, (2008)
4. Sirin, E., Parsia, B., Wu, D., Hendler, J. and Nau, D.: HTN Planning for Web Service Composition Using SHOP2. In Proc. Proceedings of 2nd International Semantic Web Conference(ISWC'03). pp. 20-23. Springer, Sanibel Island, Florida, USA (2003)
5. Koehler, J.: Planning from second principles. Artificial Intelligence, 87(1-2):145-186, (1996)
6. Klusch, M. and Renner, K.-U.: Fast Dynamic Re-Planning of Composite OWL-S Services. In Proc. Proceedings of the 4th IEEE European Conference on Web Services (ECOWS 2006) pp. 134-137. IEEE CS Press, Zurich, Switzerland (2006)
7. Recio-García, J.A., Diaz-Agudo, B. and González-Calero, P.A.: A Distributed CBR Framework through Semantic Web Services. In: Bramer, M., Coenen, F.andAllen, T. (eds.): In Proc. Proceedings of the Twenty-fith SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, AI 2005. pp. 88-101. Springer (2005)
8. Canfora, G., Penta, M.D., Esposito, R. and Villani, M.L.: QoS-Aware Replanning of Composite Web Services. In Proc. Proceedings of the IEEE International Conference on Web Services (ICWS'05). pp. 121-129. IEEE Computer Society (2005)
9. Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J. and Chang, H.: QoS-Aware Middleware for Web Services Composition. IEEE Transactions on Software Engineering, 30(5):311-327, (2004)
10. Beyer, H.-G. and Sendhoff, B.: Robust optimization – A comprehensive survey. Comput. Meth. Appl. Mech. Eng., 196(33-34): 3190-3218, (2007)