2012 International Conference on Medical Physics and Biomedical Engineering

Operation Composition Based on Linear Logic

Yanbin Peng

School of Information and Electronic Engineering
Zhejiang University of science and technology
Hangzhou, china

Abstract

While most literatures concentrate on composing services to reach complex service requirement, they all neglect one fact: service may has several operations which can also be composed to accomplish a complex function. This paper considers operation composition and imports linear logic to reason the correctness of composed operation. In the new framework, operation is expressed as linear logic axiom and the composed operation is expressed as linear logic theorem. With the help of correctness and completeness linear logic, we can verify the composed operation can meet the requirement. Experiment results show that the proposed method can improve recall rate of service discovery.

© 2012 Published by Elsevier B.V. Selection and/or peer review under responsibility of ICMPBE International Committee.

Keywords: web service; operation composition; linear logic;

1. Introduction

The software industry has experienced three stages: 1) during 60 years of 20th century, object-oriented technology innovate the development model of software product. The reusable of class reduces the complexity of software development. 2) During 80 years of 20th century, component-based technology and system integration technology enlarge the scale of software industry. 3) With the development of computer and internet technology, SOA (service oriented Architecture) and SOC (service oriented computing) has become the leading technology of the development of software industry. The essence of SOC and SOA [1, 2] is to implement large scale resource sharing and collaborative problem solving through standard protocol based on the foundation of loosely coupled distributed system. Thereafter, SOC and SOA will be the core technology to support software industry development.

To build service-oriented system and implement service-oriented computing mainly involve three aspects: 1) web service discovery. After built service library, system should provide rapid and efficient algorithm to find the target service to meet requirement of service requester. 2) Web service composition.
Service requirement will not always simple. Most of the time, single service can not meet the service query requirement. Therefore, system must provide some means to compose atomic services to achieve the requirement of service requester. 3) Composition service verification. How can we believe the composed service can meet the requirement? We should prove it. Therefore, some means should be provided to prove the correctness of composed service.

To solve the above problems, many research efforts [1-16] have been done. However, in service composition field, they neglect one fact: service may have several operations which can also be composed to accomplish a complex function. Without considering this fact will make service oriented system can not find as much target composed service as possible, which in turn will reduce recall rate of service discovery. Based on our former research works [10, 11, 13-15], this paper considers this fact and provides a linear logic based method to prove the correctness of composed operation. Experiments show that the new proposed method can improve recall rate of service discovery.

2. Architecture Linear Logic Enhanced System

Based on our former research works, we design a new architecture for service discovery and integration. Firstly, we assume that a service is not atomic unit. There are several operations in a service. Therefore, we can compose operations in service to form an advanced service which can implement more complex functions. Based on advanced service library, we can match service with service requirement, and compose service to meet complex service requirement. To guarantee the correctness of composed operation in advanced service, we express operation as linear logic axiom and composed operation as linear logic theorem. Based on the correctness and completeness of linear logic, we can prove the correctness of composed operation. The architecture of linear logic enhanced system is shown in figure 1.

![Architecture of linear logic enhanced system](image1)

3. Linear Logic

Linear logic (LL), proposed by the J.Y.Girard, is an improvement on the classical logic. LL is a computable, resource-oriented logic which is widely used in theorem proving and other fields.

Syntax of linear logic used in this paper is as follows:

\[ A ::= P | \neg A | A \otimes A | A \oplus A | ! A | 1 \]
Where, P represents proposition variables, A represents formula. ⊗ is called multiplicative conjunction, like “and” in classic logic. ⊕ is called additive disjunction, like “or” in classic logic. → is logic implication.

Based on the syntax of linear logic, service function can be expressed as linear logic Theorem: ⊩, Δ ⊩ I→⊂O, where ⊩ is operation set of service, Δ stands for non-functional constraint, I stands for input parameter set of service function, O stand for output parameter set of service function, I→⊂O express the function of composed operations in ⊩. For any operation, it can express as a linear logic axiom: ⊩, Δ ⊩ I→⊂O, where O stands for operation name, the meaning of Δ, I and O is the same as above description. For simplicity, we neglect non-functional constraint Δ, therefore, operation is expressed as: ⊩ ⊩ I→⊂O, service function is expressed as: ⊩ ⊩ I→⊂O.

The common reference rules are as follows:

1) shift rule:
\[
\frac{\Omega, A ⊩ B}{\Omega, A ⊩ B} \quad \frac{\Omega, A ⊩ B}{\Omega, A ⊩ B}
\]

2) cut rule:
\[
\frac{\Omega, A ⊩ B, \Omega, A ⊩ B}{\Omega, A ⊩ B}
\]

3) R ⊗ rule:
\[
\frac{\Omega, A ⊩ B}{\Omega, A ⊩ B} \quad \frac{\Omega, A ⊩ B}{\Omega, A ⊩ B}
\]

4) R→→ rule:
\[
\frac{\Omega, A ⊩ B}{\Omega, A ⊩ B} \quad \frac{\Omega, A ⊩ B}{\Omega, A ⊩ B}
\]

### 4. Verification of Composed Operation

This paper mainly concerns the verification of composed operation. Therefore, in this section, an example is provided to show the process of composed operation verification based on linear logic.

Assume a service named refrigerator-selling, which has three operations:

1) model-searching operation. Operation used for the consumer to search model for his favorite refrigerator, expressed as: model-searching ⊩ F→⊂M.

2) price-searching operation. Operation used for the consumer to search price for his favorite model, expressed as: price-searching ⊩ M→⊂P$.

3) price-transforming. Operation for the consumer to transform price of dollar to price of Chinese money, expressed as: price-transforming ⊩ P$→⊂P ¥.

In this example, we have a complex service function beneath these operations, expressed as: ⊩ ⊩ F→⊂P ¥. The verification process using rules is expressed as follows:

Model-searching ⊩ F→⊂M

Price-searching ⊩ M→⊂P$

price-searching, F ⊩ M

price-searching, M ⊩ P$

shift
Finally, we have proved complex service function \( MS, PS, PT \vdash P$ \rightarrow PY \) through linear logic reasoning process. MS stands for model-searching, PS stands for price-searching, PT stands for price-transforming.

5. Experiment

In order to certify the effects of our proposed method, we have done a series of experiments. For short of standard service library, we generate service library by ourselves. For one service, firstly, we generate an operation number at random, and then generate operation one by one. For every operation, we generate its input parameters and output parameters at random. Therefore, we random generate six service library which has 1000, 2000, 3000, 4000, 5000 and 6000 service in it. We compare our method with the method of CZERWINSKI and KLUSCH. The experiment results are as follows:

<table>
<thead>
<tr>
<th>Service number</th>
<th>1000</th>
<th>2000</th>
<th>3000</th>
<th>4000</th>
<th>5000</th>
<th>6000</th>
</tr>
</thead>
<tbody>
<tr>
<td>CZERWINSKI</td>
<td>0.73</td>
<td>0.72</td>
<td>0.68</td>
<td>0.66</td>
<td>0.64</td>
<td>0.63</td>
</tr>
<tr>
<td>KLUSCH</td>
<td>0.76</td>
<td>0.75</td>
<td>0.73</td>
<td>0.71</td>
<td>0.68</td>
<td>0.66</td>
</tr>
<tr>
<td>Our method</td>
<td>0.81</td>
<td>0.79</td>
<td>0.79</td>
<td>0.77</td>
<td>0.75</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Seeing from table 1, we can conclude that because of form complex service function from combining operations in service, we can gain better recall rate than other service discovery method.

6. Conclusion

Aiming at mining the internal ability in service, this paper considers that service has several operations, which can be combined to achieve more complex function. To verify the correctness of composed operation, this paper import linear logic as a basis to verify it. Experiment results show the new proposed method can improve recall rate of service discovery. In the following, we will compose operation from different services.
Acknowledgment

Supported by the research program supported by the department of education in ZheJiang Province(Y201016929); Zhejiang Provincial Natural Science Foundation of China（No.Y1100036 & No. Y1080366）.

References


