A Research on Error Handling Model of Service Fault-tolerant Strategy

Yongqing Wang
College of Transportation Management
Dalian Maritime University
Dalian, P.R China[1]
Bureau of Yunnan Highway Transport Administration
Kunming, P.R China[2]
jzr1986@hotmail.com

Yan Chen
College of Transportation Management
Dalian Maritime University
Dalian, P.R China
chenyan_dlmu@163.com

Ming Yang
College of Transportation Management
Dalian Maritime University
Dalian, P.R China
turner7257@163.com

***Abstract*—With the rapid development of web services, web services fault-tolerant mechanism has become a new research focus. This paper presents an innovative error handling model of service fault-tolerant strategy. For the goal of enhancing the fault-tolerant performance of web services, we analyze the key technologies of web services fault-tolerant in detail, propose the concept of fault-tolerant strategy, describe the service fault-tolerant model mechanism, show the life cycle and the state transition of service in this model, as well as the working mechanism of fault-tolerant mode. And the model has been applied in the project of "maritime and shipping integrated system of Yunnan Province". As this model has the characteristics of flexibility and platform independence, it has received a good practical application, greatly enhance the user experience. The feasibility rationality and effectiveness of the model has been proved.**

Keywords- error handling; fault-tolerant strategy;web service

I. INTRODUCTION

The Internet is a non-reliable heterogeneous, distributed autonomy and rapidly evolving network ecological environment [1]. Running in such a complex environment, web services may be affected by network failure, changes of the mode of communication, the infrastructure changes, and security issues, which may cause the business process functionality and quality of service difficult to be guaranteed. What's more, the web services those have complex business logic will contain more than one service component, so they need messaging and data exchange with each other to accomplish specific business needs. For the long-running service, there are complex interactions; the probability of error occurrence is much higher. The common error contains: software errors such as server crash, network failure and hardware error; memory leaks, data anomalies, and other types of errors as server shutdown, network congestion and CPU load is too high[2~4].

Therefore, the web service fault-tolerant mechanism has become the research focus. Currently, the service fault-tolerant research focuses on the extension of the standard protocol, fault tolerance of the constituent services and fault-tolerant platform of web service combination. WSRM [5] and WS Reliability [6] is the XML messaging fault-tolerant standard developed by the industry. Reference [7] proposed a concept of fault-tolerant SOA, which will use the web service replication and log methods to achieve fault recovery mechanisms. Reference [8] proposed an optimization method based on combined service process reengineering concept. Reference [9] proposed a single business-critical service recovery based on service copies to ensure the service availability. Reference [10] proposed a fault-tolerant model of composite web services based on mobile Agent which is actually using the service copy to provide fault tolerance.

In this paper, we design an error handling model of service fault-tolerant strategy.

II. THE ERROR HANDLING MODEL OF SERVICE FAULT-TOLERANT STRATEGY

This model uses the pre-defined fault-tolerant strategy in the description of the services for forecasting the error occurs, classify the errors and then define the approach to deal with all kinds of errors which can guide the web service implementation component, enable it to timely and correct handling errors in the process of service execution. A description of each service has its own definition of error and the corresponding error approach. This allows the web service implementation component easy to monitor the abnormal situation of the service, and make the appropriate treatment.

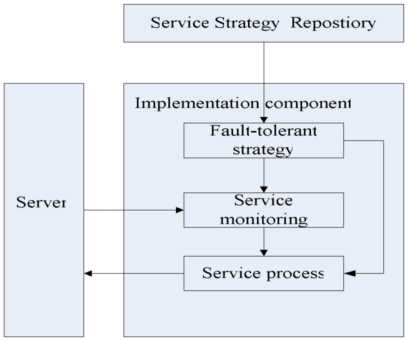


Figure 1. Error handling model based on fault-tolerant strategy

A. *Fault-tolerant strategy*

Service strategy repository stores the services fault-tolerant strategy which is described by XML. The service fault-tolerant strategy reflects the service registration information and it can be defined in the WSDL service description document or be extracted separately. When the developers give some option fields to service strategy in advance, it will choose different operations under different conditions. In service strategy, the description of exception handling condition usually uses a form as follows:

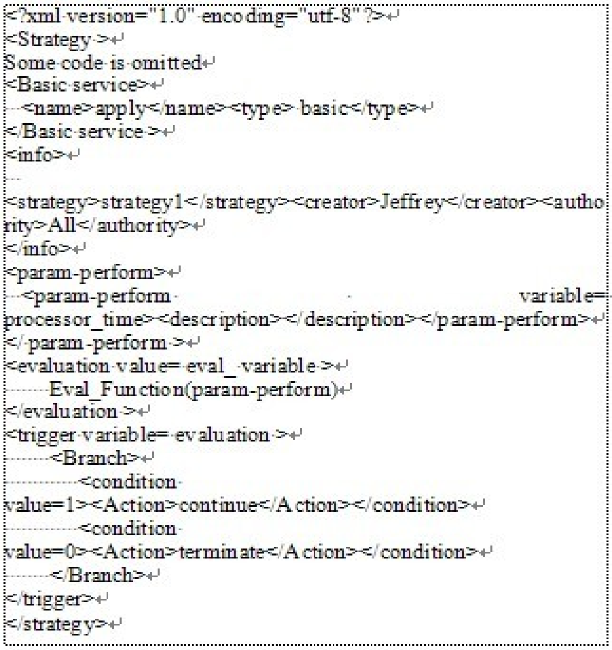
$$if \quad \langle condition \rangle \quad then \quad \langle action \rangle \tag{1}$$

As it is described by the format of XML, the editing and maintenance of this service strategy can be developed by some editing tools with more flexible and friendly user interface. For different services, you can define different error handling and recovery rules. For example: for a number of important services, you need to use the backup server. At the same time, a service can have more than one service fault-tolerant strategy. In addition, there are requirements of quality of service (Qos), thus increasing the robustness of the fault tolerance mechanisms. For example: when the server load reaches a certain value, it will result in a sharp decline of service performance or even cause the service to stop working, and these situations need to be detected and avoided.

In fault-tolerant strategy, we define some kind of evaluation function, and then make a comprehensive evaluation of relevant information to get the operating status of the current services and servers, such as:

$$Ef\,(st_1, st_2, ..., st_n) \tag{2}$$

In which, st_1, st_2, …, st_n represent the status parameters of running service. The results of this comprehensive evaluation can be used as a condition option to drive the implementation of the specific decision-making behavior. Here is an instance of the XML description of a service fault-tolerant strategy:



```
<?xml version="1.0" encoding="utf-8" ?>
<Strategy >
Some code is omitted
<Basic service>
  <name>apply</name><type> basic</type>
</Basic service >
<info>
…
<strategy>strategy1</strategy><creator>Jeffrey</creator><autho
rity>All</authority>
</info>
<param-perform>
  <param-perform                                    variable=
processor_time><description></description></param-perform>
</ param-perform >
<evaluation value= eval_ variable >
       Eval_Function(param-perform)
</evaluation >
<trigger variable= evaluation >
       <Branch>
            <condition
value=1><Action>continue</Action></condition>
            <condition
value=0><Action>terminate</Action></condition>
       </Branch>
</trigger>
</strategy>
```

This fault-tolerant strategy contains the following information:

- Service, the corresponding service information, such as service number, service name and service type;
- Info: service strategy information, such as strategy number, creator, service number;
- Param-perform: used to evaluate the service and the related parameters of the server running, such as: the effective input and output of services, CPU load and memory use;
- Evaluation: used for a comprehensive evaluation of local parameter, then get the driving condition of service strategy;
- The trigger: the definition of conditional options, it triggers the related operations according to different evaluation results,;
- Action: give a specific operating behavior for different conditions of options;
- Others: this includes other relevant information, such as quality of service requirements and safety.

B. *Service fault-tolerant model mechanism*

When the service starts, the implementation component issues a strategy request to service strategy repository to get the appropriate service strategy according to the service marks, it transforms the error description information of service and the processing action into information which can be identified locally, during the operation of service, implementation components monitor the error occurred based on error description of services.

If an error occurs, the implementation component will handle the according error based on error handling scheme given by fault-tolerant strategy. For example, restart the service or start the other service implementation procedures.

In service fault-tolerant model, we use the error handling mechanism to achieve fault-tolerant strategy, the fault-tolerant strategy describes the error characteristics and the corresponding processing mechanism, increase the flexibility of the system; and the error monitoring and treatment of implementation component, also make the advantages of automation to the system.

The life cycle of the services running including the following stages, as shown in Fig. 2:

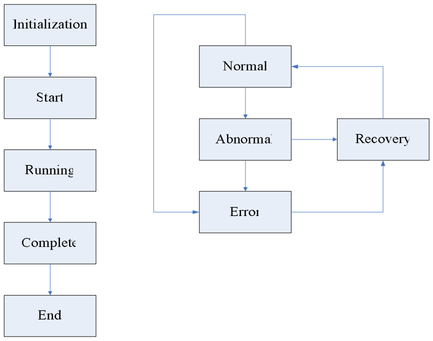


Figure 2. The life cycle and the state transition of service.

①Service initialization: the initialization process of service copy on the host server, include the configuration of service operating parameters service and the environment initialization.

②Service start: the copy program of service is running, the initialization process of program itself;

③Service running: the copy of service is running, it begins to provide normal services;

④ Service complete: logically, the copy of service completes a unit of work;

⑤ Service end: the copy program of service exits, the service end.

In which, the service status when service is running will be include:

①Normal;

②Abnormal;

③Error;

④Recovery.

It should be noted is: here the meaning of "abnormal" refers to a state between the "normal" and "wrong", such as the server that provide services has been running beyond its capacity, etc. So when the "abnormal" occurs, we need appropriate treatment, so that the status of the service could turn into a "normal", or it may appear "wrong". In the process of service running, we can collect some parameters information used to assess the current status, such as memory usage, CPU load.

In service fault-tolerant model, the implementation component provides the interface to establish communications with the service provider's server, to collect the operation situation of the service and the operation situation of the server.

In this paper, the key point technique is used to be an important fault-tolerant service recovery technology in the design of our fault-tolerant model. The key point library of model will record information of key point of service. When a service runs to the key points, the implementation components will judge according to the service strategy definition and description, and record the context information. But not all of the services get the key point information; there may be other mechanisms in the recovery of the service.

Figure 3 shows the working process of the service monitoring of implementation components:

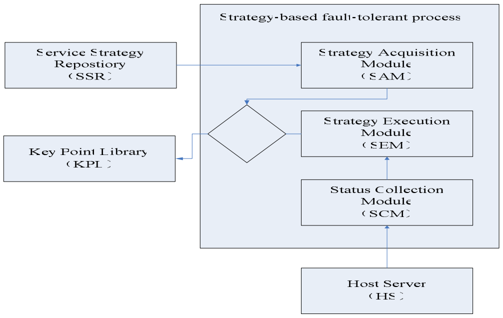

Strategy-based fault-tolerant process
Service Strategy Repostiory (SSR)
Strategy Acquisition Module (SAM)
Key Point Library (KPL)
Strategy Execution Module (SEM)
Status Collection Module (SCM)
Host Server (HS)

Figure 3. The working mechanism of fault-tolerant mode.

(1) Strategy Acquisition Module (SAM), is responsible for getting the related information from the service strategy repository, this information will be used to determine the status of the running service and whether it reaches the key point;

(2) States Collecting Module (SCM) is responsible for communication with the server to collect the operation information of service and host server;

(3) Strategy Execution Module (SEM), is based on current services and server status information and services strategy to determine whether the service has occurred error, or whether it has reached a key point (if the service provides key points information as a service recovery strategy), if an unexpected error occurred and it will execute corresponding operation to handle the error based on error strategy provided by service strategy.

The following is part of the fault-tolerant processing and service recovery operation:

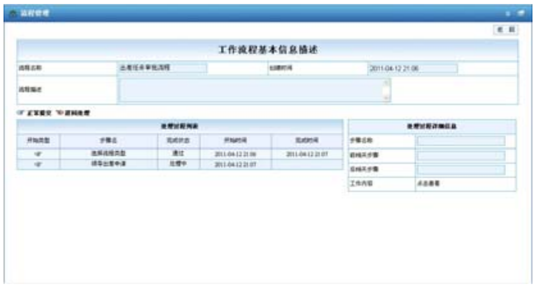
- Recovery services from the key points. When the service is running, the key point information of service is stored in the key point library. SEM can re-run the service, achieve the service recovery into a stable state after the last update of the key point information;
- Restart the service. when a serious service application error occurs (such as: service program automatically exit), it needs to re-start the service, at this time, it does not re-assign the service to run on other servers;
- Dynamic re-distribution services to other available servers. If the server load is too heavy or a serious error occurs which will lead the service error, the service should dynamically allocated to other available servers;
- Start the sub server as the primary server. If there is a same service running on the sub server at the same time, start the sub server as the primary server;
- Use the other implementations of the service. If the service has a number of different implementations, in the error of current implementation, you can start other implementations, re-run the service.

After the recovery of service, the implementation component continues to monitor the execution of the service until it completes.

III. A REAL CASE STUDY OF APPLYING MODEL

This section focuses on a real cooperation project--“maritime and shipping integrated system of Yunnan Province”, which is developed by Shipping Administration Bureau of Yunnan Province and Dalian Maritime University. This system mainly includes four sub-platforms: office platform, shipping platforms, maritime platforms, integrated service platform. It should be an integrated set of user actions, system integration and data integration into one common platform. The error handling model of service fault-tolerant strategy has been actually applied to the project.

The GUI of system which is used error handling model is shown in Fig 4:



工作流程基本信息描述

流程名称	出差任务审批流程	创建时间	2011-04-12 21:06
流程描述			

正常提交 退回处理

处理过程列表				
开始类型	步骤名	完成状态	开始时间	完成时间
	选择流程类型	通过	2011-04-12 21:06	2011-04-12 21:07
	填写出差申请	处理中	2011-04-12 21:07	

处理过程详细信息	
步骤名称	
前驱关步骤	
后续关步骤	
工作内容	点击查看

Figure 4. The GUI of system.

As an integrated data platform, there must be a large number of service calls. In the process of project implementation, the model has received a good practical application, greatly enhance the user experience.

IV. CONCLUSIONS

This paper introduces an innovative error handling model based on service fault-tolerant strategy. It proposes the concept of fault-tolerant strategy, describes the service fault-tolerant model mechanism, shows the life cycle and the state transition of service in this model, as well as the working mechanism of fault-tolerant mode. And the model has been applied in the project of "maritime and shipping integrated system of Yunnan Province". As this model has the characteristics of flexibility and platform independence, it has received a good practical application, greatly enhance the user experience. The feasibility rationality and effectiveness of the model has been proved.

ACKNOWLEDGMENT

This wok was supported by the Fundamental Research Funds for the Central Universities (2012QN072).

REFERENCES

[1] LIU Ling-xia WU Zhao-xue QIAN Yuan, et al. "Fault-tolerant Web Services" Computer Science, 2009, vol. 36, pp. 24-28.

[2] Schneider F B. Implementing fault-tolerance services using the state machine approach. ACM Computing Surveys, 1990, 22(4), pp. 299-320

[3] Stelling P, Foster I, Kesselman C, et al. A Fault Detect i on Service for Wide Area Distributed Computations. The 7th IEEE Symp. on High Per formance Distribut ed Computing. Chicago,USA, 1998

[4] ZHANG Li-min, GAO Chun-ming, CAI Mei-ling. Fault-tolerant processing mechanism of Web service composition in run-time.Computer Engineering and Applications, 2008, 44 (3),pp. 143- 147.

[5] Iwasa K, Durand J, Rutt T, et al.Web service reliable messaging,WS-Reliability 111.OASIS Specification, 2004.

[6] Bilorusets R.Web service reliable messaging protocol [S].IBM/BEA/Microsoft/TIBCO Specification, 2004.

[7] Liang D, Fang C L, Chen C, et al.Fault - tolerant Web service. Proceedings of the 10th Asia - Pacific Software Engineering Conference (APSEC' 03) , ChiangMai, Thailand, 2003,pp.310- 319.

[8] Zeng Liang-zhao, Benatallah B, Dumas M, et al.QoS- aware middle-ware for Web services composition.IEEE Transactions on Software Engineering, 2004, 30(5),pp. 311- 327.

[9] Salas J, P'erez-Sorrosal F, Pati Marta, et al. WS-replication: aframework for highly available Web services. Proceedings of the 15th International Conference on World Wide Web, Edinburgh, Scotland, May 2006, pp. 23- 26.

[10] XU Wei, JIN Bei-Hong , LI Jing, et al. A Mobile Agent2Based Fault2Tolerant Model for Composite Web Service. Chinese Journal of Computers, 2005, 28(4), pp.558-567.