KAF: Kalman Filter Based Adaptive Maintenance for Dependability of Composite Services

Huipeng Guo, Jinpeng Huai, Yang Li, and Ting Deng

School of Computer Science and Engineering, Bei Hang University, Beijing
guohp@act.buaa.edu.cn, huaijp@buaa.edu.cn,
{liyang,dengting}@act.buaa.edu.cn

Abstract. Service composition is fundamental in development of Web service oriented applications. Dependability of composite services is of significant importance since it directly impacts users' experience. However, dependability of a composite service may change over time as a result of inevitable changes in component services. In addition, users may also pose varying dependability requirements to meet different needs. It has become a big challenge to dynamically maintain the dependability of composite services. This paper proposes an innovative system called KAF that constructs a closed-loop control for adaptive maintenance of composite services. Modeling the control process as a Markov decision process (MDP), we further design an efficient Kalman-Filter based algorithm for service state prediction. With the availability of the precise prediction, optimal control decisions can be made. We evaluate the performance of KAF against other alternative approaches through comprehensive experiments and results demonstrate that KAF is capable for adaptive dependability maintenance.

Keywords: service composition, dependability, adaptive maintenance, Kalman filter.

1 Introduction

With the maturity of key standards such as SOAP, WSDL, and UDDI, Web service has been widely recognized as a promising technology for distributed application development. Following the service-oriented architecture, Web service supports better interoperability, higher usability, and increased reusability compared to traditional middleware technologies such as RMI and CORBA. Service composition is the fundamental process for Web service based application development, which constructs a composite service by composing basic Web services that may be distributed over the world. As this trend continues, more and more composite services will be deployed in the next few years, serving various sectors of our society.

The dependability property of composite services is of significant importance to users, which includes many critical factors, such as availability, reliability and so on. A service with higher availability promises more functional time and a more reliable service reduces the probability to fail when it is invoked. The trust property of a service is also important since a user may risk invoking a malicious service.

Z. Bellahsène and M. Léonard (Eds.): CAiSE 2008, LNCS 5074, pp. 328–342, 2008.

A key issue to service composition is that the dependability of a composite service may change dynamically over time. Users however, desire that the composite service delivers stable dependability. A stable dependability makes it possible for the users to expect the predictable results. This is particularly important when the application is mission-critical, e.g., a disaster management. When the dependability of the composite service degrades, the users may severely suffer from the decreased performance.

The intrinsic dynamics of composite services stems from the fact that a composite service is composed of many component services that potentially belong to different providers distributed in the open Internet. Each component service is subject to different environments and changes, such as varying system load and available bandwidth. The properties of such a service may therefore change from time to time. In addition, existing component services may become unavailable or new services may appear to serve. All these factors lead to the inevitable dynamic changes of the composite service. On the other hand, the dependability posed by the users or the applications may evolve to meet different needs in reality.

To obtain stable composite services with desirable dependability, the system must provide adaptive control to offset the unpredictable changes in either environments or application requirements. The Web service and service composition has been studied and several attempts have been made to construct high dependable composite services [1-4]. However, they fail to support the adaptive maintenance of dependability. To the best of our knowledge, there is no successful work for this problem.

In this paper we propose an innovative system called KAF to dynamically maintain the dependability of a composite service. KAF constructs a closed-loop control for the maintenance. By continuously monitoring the system status, KAF makes progressive control over the composition of the component service. We model this as a Markov decision process (MDP) that facilitates control decision making. Since it is highly desirable to maintain the dependability of the composite service over the one required by the users, it becomes imperative to estimate the next status of the service. Exploiting the MDP modeling, we propose a Kalman-Filter Based adaptive maintenance algorithm that precisely predicts and adjusts the service state.

We have made the following original contributions in this paper.

- We make the attempt to explore the important issue of adaptive maintenance of composite services. The innovative system KAF is proposed which implements a closed-loop control over service dependability.
- By modeling the control process as a MDP, we design a Kalman-Filter based algorithm that accurately predicts the state of the composite service such that optimal control decisions are thereafter made.
- We have preliminarily implemented KAF and conducted comprehensive trace-driven experiments. The performance of KAF is studied comparatively against other alternative approaches.

The rest of this paper is organized as follows. Section 2 introduces the background and problem. Section 3 states the KAF architecture. Section 4 describes the adaptive control method of service composition dependability maintenance. Adaptive control mechanism implement of service composition dependability is discussed in Section 5. In Section 6 we evaluate the proposed approach. And we review the related work in section 7. At last, conclusions and future work are presented in Sections 8.

2 Background and Problem Statement

In this paper, we model a composite service as a tuple (*ST*, *SC*), where *ST* is a collection of tasks combined according to the four basic service composition modes including *sequence* (•), *parallel* (||), *choice* (+) and *iteration* (◦), and *SC* is the set of component services for each task in *ST*.

There are many attributes associated with a Web service. In this paper we particularly focus on four key attributes for component services: *availability*, *reliability*, *trust* and *price*. The approach can support more flexible definition of QoS other than these attributes. As discussed before, dependability can be considered as an aggregate property of availability, reliability and trust [4-10]. These attributes of services reflect from different perspectives the ability of a service providing a specific functionality. Price directly indicates the cost that a user has to pay and thus is indispensable to be included for study in a practical setting. Therefore, each component service can be represented by a tuple ($A(t)$, $R(t)$, $T(t)$, $P(t)$), where $A(t)$, $R(t)$, $T(t)$ and $P(t)$ represent the numerical values of availability, reliability, trust and price at time t, respectively.

Service availability is defined as the probability that a service is properly functioning in a specified period of time under certain conditions. According to [10], service availability can be measured as:

$$A\left(s_{ij},t\right)=\frac{T_s\left(t\right)}{T_a\left(t\right)}\ , \tag{1}$$

where $A(s_{ij}, t)$ denotes the availability of service s_{ij} in time t; $T_s(t)$ is the total available time and $T_a(t)$ is the total measurement time.

Service reliability is the ability of executing the specific functionality under certain conditions and within a particular period. And it can be measured by the ratio of the unsuccessful execution number and the total number of executing:

$$R\left(s_{ij},t\right)=\frac{N_s\left(t\right)}{N_a\left(t\right)}\ , \tag{2}$$

where $R(s_{ij}, t)$ denotes service reliability; $N_s(t)$ is the number of the successful execution instances, $N_a(t)$ is the total number of execution.

How to calculate the trust value is a hot research topic, and there are many works [6-8]. Generally, there are two types of trust value: direct trust and recommendation trust. Based on Beth's works [6], taking into account practicability and measurability, we describe trust value of the service mainly with recommendation trust. In this paper we assume that the trust of a Web service is based on recommendations of the service information manager (SIM), which collect the feedback results of some selected honest sampling clients. In addition, other trust value calculation methods [7, 8] can also be supported.

According to [4], the availability of a composite service with redundant services can be computed given the availability information of its component services. Similarly, we can also compute the reliability and the trust of the composite services. Given the three attributes computed as above, we are able to compute the dependability *d* as a function of them.

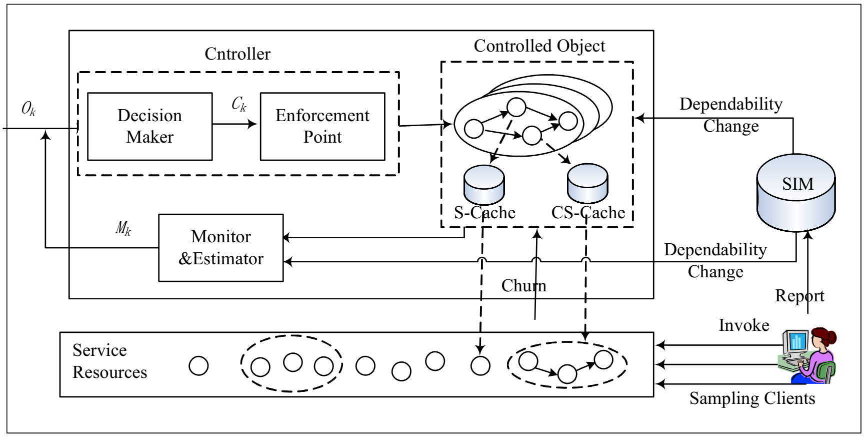


Fig. 1. KAF architecture

$$d = \alpha a + \beta r + (1 - \alpha - \beta)\tau,\ \alpha, \beta, a, r, \tau, d \in [0,1]\ . \tag{3}$$

Hereinto, a, r and τ are the availability, the reliability and the trust of a composite service in a certain period, respectively. Note that α , β and $(1-\alpha-\beta)$ are the weights of availability, reliability and trust. The assignment of weights can be determined by the users to reflect different emphasis on these attributes.

On the other hand, we take into account maintenance costs of a composite service. In this paper, based on our previous work [4], the attribute can be dynamically adjusted based on the instrument of redundancy provision. It is intuitive that more services are used, higher cost are introduced.

The objective of the maintenance for composite services is as follows. For a given composite service (*ST*, *SC*), and the given desired dependability (denoted as *O*), we need to keep the instant dependability of the composite service *d*(*t*) as close to *O* as possible for any time *t*. Meanwhile, we should minimize the cost of the composite as much as possible over time.

3 KAF Architecture

In this section we first introduce the system architecture of KAF and then describe the modeling of the decision making process for dynamic maintenance in KAF.

3.1 System Architecture

The KAF Architecture is shown in Figure 1, which consists of three main components. The Monitor and Estimator component monitors the current attributes of all services and also estimate their further states based on the current and historical information. The information provides the basis to the Decision Maker that produces maintenance strategy. The decision output is feed into Enforcement Point that implements the decision and dynamically adjusts the composition of the composite service.

In the KAF, composite service based on redundancy, including component services cache named S-Cache and composite services cache named CS-Cache, is the controlled object. Guarantee strategy of composite services is the corresponding controller and is performed by Decision Maker and Enforcement Point. Monitor& Estimator, SIM and sampling clients form a feedback loop. Composite services, guarantee strategy and feedback loop constitute a closed-loop feedback control system.

In figure 1, O_k ($0 \le k \le N$) are the desired value of the dependability in a composite service system, represents dependable attributes of a composite service. M_k ($0 \le k \le N$) are feedback values of a composite service, denote actual values of dependability attribute. M_k are estimated by Estimator based on summarization of SIM from the report of sampling clients. C_k ($0 \le k \le N$) are control values and are determined by the difference between expectations and estimating value. C_k denote the dependability value needed to be adjusted. Based on adjustment strategies, the Enforcement Point choose candidate services and complete the reconstruction of services composition.

In KAF architecture, the feedback loop is composed of two mechanisms: execution monitoring and reporting. Execution monitoring gains the services available state information from the execute engine by the monitor. Reporting mechanism summarizes the using result of sampling clients and sampling clients are some honest client nodes selected by SIM. Estimator estimates dependability of component service base on the results of SIM and then computes dependability of composite service.

SIM is an extended service registry supporting the description and update of QoS information. SIM is responsible for the reporting mechanism, collecting and managing the information of service. Some function of SIM includes service dependability value monitoring, service available state detection under different strategies (such as different frequencies, different incentive incentives, and so forth), evaluation and statistical functions of sampling client monitoring reports.

To adjust the dependability effectively, the Decision Maker determines service options and update strategy, such as increasing number of backup service, replacing service decreased most and so on. In this paper, guarantee strategy is a kind of meta-strategies. According to a meta-strategy, Enforcement Point runs the APB algorithm [4] selecting new service and replacing declining or failed services to implement the composite service's re-structure. In addition to the dependability of services, we must also consider the cost of replacement and rewards of expectation, so that to make the long-term revenue maximize. For the realization of this goal, in this paper we use Markov decision process theory to support strategy choices.

3.2 MDP Modeling

Due to the dynamic nature of component services, such as verity of load, network conditions and churn, the dependability of composite service will be affected. In these conditions, we need selecting some redundant service to ensure the composite service can still satisfying the desire even dependability of some component services drop.

We design dependability factor $f = lnO/lnD$, where O is the desired value for dependable properties need to satisfied and D means the design value of dependability. Dependability factor reflecting the important degree of dependability is used to guide the choice of component service. Generally f is larger than 1, which means D usually is larger than O. We can see that, when the dependability of component services declines rarely, low dependability factor can reduce the cost of constructing composite services. When dependability dropped significantly, increasing of dependability factor can improve the quality of composite service, and the cost may increase.

To adjust the dependability factor of composite service reasonably, we select MDP (Markov Decision Process)[11] to modeling the composite service maintenance process: RSC = $\langle S, A, \{A(x) \mid x \in S\}, Q, RV \rangle$, where S is the set of all possible states and A is the set of decision strategies, $\{A(x) \mid x \in S\}$ is the strategy when the state is x, $Q(B \mid x, a) = Q\{X_{t+1} \in B \mid X_t = x, A_t = a\}$ specifies the probability to the next state X_{t+1} when the state is $X_t = x$ and after strategy $A_t = a$ is executed, and RV is the reward of executing the strategy. If we assume that the state of system in moment t was $X_t = x \in S$, while the decision is $A_t = a \in A(x)$, under the decision of transfer function Q, system transfers to the state X_{t+1} with the reward of $RV(a,x)$. After the transfer, the system has entered a new state, and then choose a new decision to continue decision-making process. Therefore we establish Markov decision process model of component service dependability's maintenance as follows.

Definition 1. In a period of time, dependability of a composite service CS is $d = \alpha a + \beta r + (1-\alpha-\beta)t$, and [0,1] is divided into k parts. When $d \in [\frac{i\text{-}1}{k}, \frac{i}{k})$, we define that the state of the composite service is s_i, marked as $s_i = [\frac{i\text{-}1}{k}, \frac{i}{k})$. And the state set of composite service dependability is denoted by $S = \{s_1, \ldots, s_m\}$.

By dividing [0, 1], we can reduce the state space and reduce the complexity of decision-making in service composition.

Definition 2. Let Δf be the variety amount of dependability factor, We define the adjustment strategy set as $A = \{-\Delta f, 0, +\Delta f\}$, where the strategy $-\Delta f$ and $+\Delta f$ mean the amount of reduction or addition values of dependability factor respectively, and the strategy 0 means no changing to the dependability factor.

Definition 3. For a composite service CS, suppose that d, ψ, w and η represent the dependability, revenue, service adjustment cost and the importance degree of the CS, where η is a positive real number, we denote by $rv = \eta(\text{-}\ln(1-d)) + \psi - w$ the reward of composite service CS.

The definition 3 represents if composite service properties such as availability, reliability and trust are higher or the maintenance cost of composite service is smaller, we can get more reward. Then composite service developers will be more satisfied with the composite service, that is, the value of v will be larger. Among them, the income is related to the number of service execution and the prices of services.

Definition 4. For a composite service CS, at the moment t suppose that CS is in the state s_i, we denote by $rv = \eta(\text{-}\ln(1-d)) + \psi - w(A_t, s_i)$ the immediate reward of composite service CS after adopting strategy A_t, where $w(A_t, s_i)$ represents the cost of dependability adjustment.

While using Markov decision process, if the state transferred probability function and reward function are known, by dynamic programming methods, we can construct value function and achieve optimal decision strategy. However, in the service-oriented software development process, it is difficult to observe all the historical actions of component services. That means that transition probability function and reward function are unknown, we can not use dynamic programming techniques to determine optimal decision strategy. Hence we design Kalman-Filter based approach to estimate systemic state and choose optimal decision strategy.

4 KAF Adaptive Control

In the distributed and dynamic environment, we can not get the parameters of service state instantly and need estimate them online. So to adjust controller parameters and to keep the performance satisfying the design demand, we apply Kalman-Filter based approach to estimation service's state, then adjust controller following the estimated value.

4.1 Kalman Filter-Based Estimation

Kalman filter [12] uses the recurrence of the state equation to achieve the optimal estimate of state variables in the linear dynamic system. The Kalman filter is unbiased and has the smallest variance. It is easy to realize by computer and be suitable for on-line analysis. Furthermore, the extended Kalman filter (EKF) can be used in nonlinear systems. Therefore, this paper introduces the extended Kalman filter to estimating the dependability of component services in order to implement the adaptive control maintenance of composite service's dependability.

At moment t, the state of a service is a $x(t)$ and represent the dependability of the service. The state equation of the component service is:

$$x(t+1)=\Phi(t, x(t))+w(t) . \tag{4}$$

In this equation, Integer $t\geq 0$ is discrete-time variable, $x(t)\in R^{n\times 1}$ is the dependability of the component service. $\Phi(t, x(t))$is the nonlinear transition matrix of service state. We suppose that $w(t)$ is the disturbance to the dependability of the component service.

The measurement equation for the dependability of the component service is:

$$y(t)=C(t,x(t))+v(t) . \tag{5}$$

Here, $y(t)\in R^{m\times 1}(m\leq n)$ is the measurement value of the dependability. $C(t,x(t))$is measurement matrix. $v(t)\in R^{m\times 1}$ is the measurement noise.

And assume that $w(t)$, $v(t)$ and the starting states $x(0)$ have characteristic: $E\{w(t)\}=0$, $E\{v(t)\}=0$, $E\{x(0)\}=x_0$, $E\{w(t)w^T(j)\}=Q_i(t)\delta_{t,j}$, $E\{v(t)v^T(\mathrm{j})\}=P_i(t)\delta_{t,j}$, $E\{[x(0)-x_0][x(0)-x_0]^T\}=P_0$ and $E\{x(0)w^T(t)\}=0$, $E\{x(0)v^T(t)\}=0$, $E\{w(t)v^T(j)\}=0$. $\delta_{t,j}=\begin{cases}1 & t=j\\ 0 & t\neq j\end{cases}$, $t, j\geqslant 0$.

We get State transition matrix:

$$\Phi(t+1,\ t)=\frac{\partial\Phi(t,x)}{\partial x}|_{\mathrm{x}=\hat{\mathrm{x}}(\mathrm{t}|\mathrm{y}_\mathrm{t})}\ . \tag{6}$$

Measurement matrix:

$$C(t)=\frac{\partial C(t,x)}{\partial x}|_{x=\hat{x}(t|y_{t-1})} \ . \tag{7}$$

The observation values are *y(1),y(2),…,y(t)*, so optimal predict value of *x(t+1)* is:

$$\hat{x}(t+1|y_t)=\Phi(t+1,\ t)\hat{x}(t|y_t) \ . \tag{8}$$

Suppose that we have gotten the optimal predictive value $\hat{x}(t|t\text{-}1)$ and the corresponding prediction error covariance *K(t|t-1)* before getting the dependability measure value *y*(*t*) of the component service. Then we can get the predict value $\hat{x}(t+1|t)$ based on the overall information and the corresponding prediction error covariance matrix *K(t+1|t)* at moment *t* by using Kalman filter and *y*(*t*).

The Kalman gain is:

$$G(t)=K(t|t\text{-}1)C^T(t)[C(t)K(t|t\text{-}1)C^T(t)+Q_2(t)]^{\text{-}1} \ . \tag{9}$$

Assume innovation $\rho(t)=y(t)\text{-}C(t)\hat{x}(t|y_{t-1})$ and the optimal predictive value based on overall information of state *x(t)* , we can apply *ρ(t)* to modify the predictive value $\hat{\mathrm{x}}(t+1|t-1)$:

$$\hat{x}(t|y_t)=\hat{x}(t|y_{t-1})+G(t)\rho(t) \ . \tag{10}$$

And the corresponding predict error for the covariance is:

$$K(t+1|t)=\Phi(t+1,\ t)K(t)\Phi^T(t+1,\ t)+Q_1(t) \ . \tag{11}$$

$$K(t)=[I\text{-}G(t)C(t)]K(t,\ t-1) \ . \tag{12}$$

4.2 Adaptive Control Algorithm

According to the aforementioned analysis, we design the adaptive control algorithm for the dependability maintenance of the composite service. The basic idea is as follows. First we collect the sampling client’s using report data about service execution, summarize the state and value of component services and calculate the dependability of composite services and compare with the expectation value. Then estimate the value of every selected component service by the Kalman filter formula, compute the immediate reward of every action and choose corresponding action following the MDP framework. At last we execute the APB algorithm [4] and the strategy execute module to select new service and replace some degenerate service.

The input of KAF adaptive control algorithm includes constructed composite service, available component services’ properties measure value, expected dependable value D_k of composite service. The output includes dependability factor and selected services.

The first part of KAF algorithm (lines 1-4) is initialization, including read the expectation value O_k of composite services, collect parameter of component services

KAF adaptive control algorithm

```
input: constructed composite service, available component service
properties measurement value, expected dependable value $D_k$ of composite service.
output: dependability factor, selected services
1. Read the $O_k$, determine the weight of every attributes;
2. Read the initial dependable attribute values from SIM;
3. Calculate the dependable attribute values for each component service respectively;
4. Calculate $d^*_k$ of composite service;
5. Collect sampling measure values $M_k$ of component services;
6. Predict the value of the next period by Kalman filter following the measure values;
7. Compute the immediate reward according to MDP strategy;
8. Determine the action and modify the dependable factor $\alpha$;
9. Calculate updated $d_k$ ;
10. Calculate $\Delta d_k = d_k - d^*_k$ ; // $\Delta d_k$ is the $C_k$
11. Select new service following the APB algorithm;
12. $d^*_k = d_k$;
13. goto step 5.
```

from SIM, calculate the dependable attribute values, and calculate real dependability of composite service.

Then obtain sampling measure values of component services, and according to Kalman filter calculation formulas estimate the values in the next period. Afterwards in the MDP, compute the immediate reward and determine the action of modifying the dependable factor α (line 5-8).

In the third part (lines 9-13), compute the dependable attribute verity values Δd_k and select new service following the APB algorithm [4]. At last, to measure the component services and start another prediction and control process.

5 KAF Implementation

The KAF platform provides environments where (i) composite service developers may create a new composite service and adaptive control the choice of candidate services, (ii) Monitor and SIM may get and validate the state and value verity of component services, and (iii) Estimator may predict the value of component services with the Kalman filter and then constructor implement the adaptive strategy of the controller. KAF is the extension of the WebSASE [24]. WebSASE supports Web service development, deployment, composition and is based on established standards such as SOAP, WSDL and BPEL. KAF is implemented in Java using IBM eclipse, and implement of middleware is based on the AXIS 2.0.

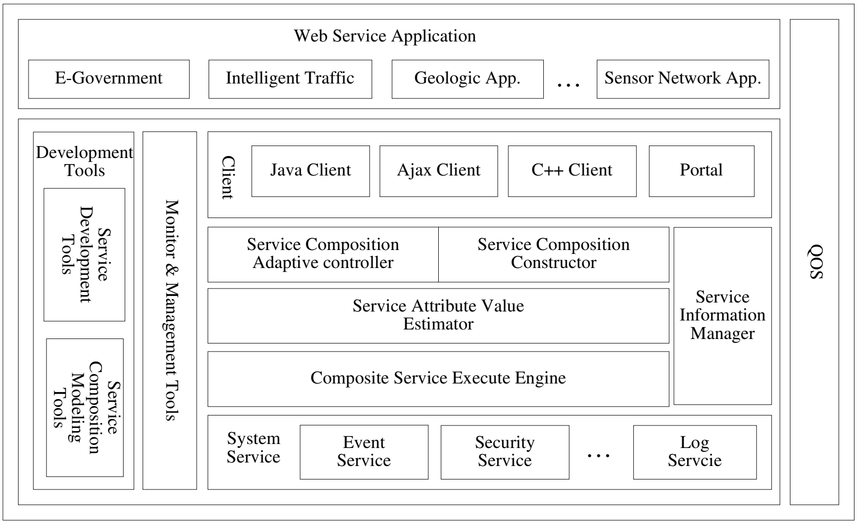

Web Service Application
E-Government
Intelligent Traffic
Geologic App.
...
Sensor Network App.
Development Tools
Service Development Tools
Service Composition Modeling Tools
Monitor & Management Tools
Client
Java Client
Ajax Client
C++ Client
Portal
Service Composition Adaptive controller
Service Composition Constructor
Service Information Manager
Service Attribute Value Estimator
Composite Service Execute Engine
System Service
Event Service
Security Service
...
Log Servcie
QOS

Fig. 2. The KAF implementation architecture

The KAF system introduces the notion of service composition adaptive control. The adaptive control is implemented by four main components: the *estimator*, the *controller*, the *constructor* and the *monitor*.

The *estimator* is responsible for estimating the value of services in the next period with the measure value of the currently period by the Kalman filter formula. The SIM is implemented as an extension to the UDDI to support the measure and evaluation of service quality. Through validation of the monitor, SIM chooses some honest clients as sampling nodes, collects the report of them and summarized as measure value.

According to dependability requirements and estimated value of the next period, the *controller* computes the immediate reward of every action and choosing corresponding action following the MDP framework. The amount of action to adjust the dependable factor can be edited through a visual interface by the composite service developer. The *controller* also determines actions to be taken in response to events such as service fail.

The *constructor* (Enforcement Point) constructs the composites according to component information and quality requirements from the model created by the modeling tools. Redundant services are selected and component service cache is constructed to store information of redundant components. Composite service cache is built and the BPEL documents of backup composites are created. Most important, the *constructor* modify the composite service through adding some more services or replace some services according to the adjustment of dependable factor using APB algorithms presented in [4].

The *monitor* performs the task of receiving the sampling measure value of services from SIM, receiving the event report from the execute engine and validating honesty of sampling clients. After measure value or other information is received, the *monitor* transmits these data to the *estimator* for the new estimation and control.

6 Performance Evaluation

In this section, we introduce experiments we conducted for performance evaluation and discuss evaluation results.

6.1 Simulation Setting and Metrics

To evaluate the effectiveness of our method, we design a set of simulation experiments to analysis the verity of dependability. First we generate a certain number of services and queries randomly. Each service is marked by its name, version, type, availability, reliability, trust and price etc, and these data are updated dynamically.

We adopt the following metrics and strategies: *dependability*, *cumulative reward*, *Random strategy* and *Kalman-Filter based adaptive control strategy*.

Dependability denotes an integrative ability of a composite service to provider certain functionality in a certain period. In this paper we select detailed properties including availability, reliability and trust.

Cumulative reward defines the reward of composite service execution in a certain period of time (Compute following definition 3. And in this paper the time is a day.).

Random strategy: If some component service in the composite service fails, then we select random another service to replace it.

Kalman-Filter based adaptive control strategy: In execution process of composite service, we use Kalman filter to estimate the dependability of the component services and adjust the dependable factor and then reconstruct a composite service.

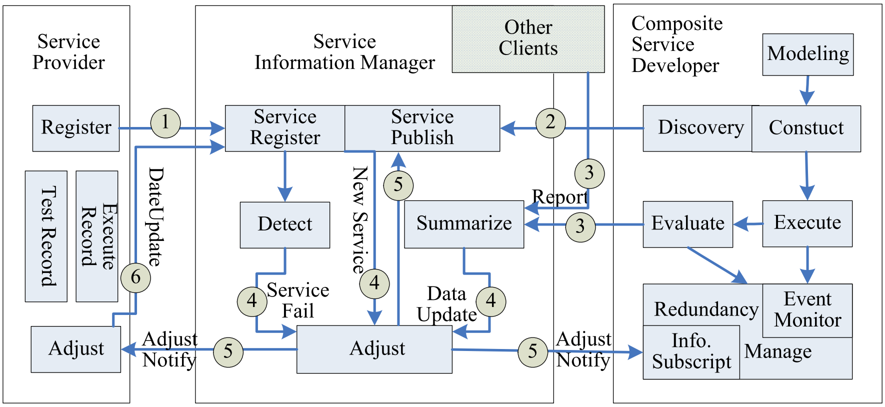
Base on the above metrics and strategies, we design the adaptive control mechanism of composite service dependability, the execute process is shown in figure 3. The main function of SIM include service information registering and publishing, service available state monitoring, service dependability evaluation, service dependability properties analysis adjustment. The main function of Service Provider includes service information registering, service execution record, service information update adjustment. The main function of Composite Service Developer includes service discovery, service modeling, service construction, service execution, service execution evaluation and redundant service management.

6.2 Results Analysis

In our simulation, 3000 services are generated and distributed into selected nodes. 2.3 percent of the services fail every day, which is based on the observation of [13].

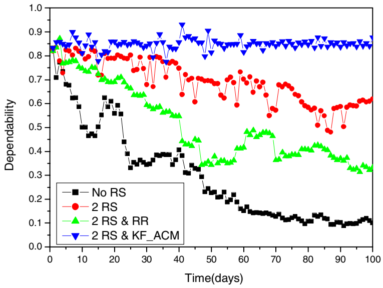
The first simulation is implemented to analyze the effect of adaptive control maintenance mechanism. 50 composite services is constructed and the dependability of execute for 100 days is recorded. The results in figure 4 show that dependability of composite services without backup services drops to a very low level after some days.

With the help of backup services, the dependability drops slowly. Results also show that relatively higher dependability can be achieved with the random replacement of failed service. With the help of adaptive maintenance mechanism based on KAF, the dependability stays steadily in a relatively high level.



1. Service Registry, 2. Service Discovery, 3. User Report,
4. Service Data Update, New Service Register, Service Fail Message Exchange,
5. Service Information Adjust Notify, 6. Service Information Update

Fig. 3. Process of services' dependability information feedback



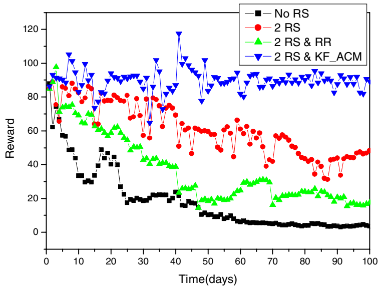
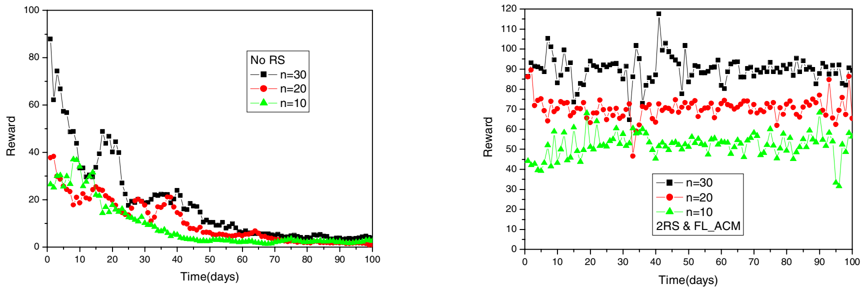


Fig. 4. Dependability maintain dynamically

Fig. 5. Reward of maintain dynamically

The second experiment is to analyze the effect of adaptive maintenance mechanism on the Reward of composite service. The results in Figure 5 show that reward of composites without backup services drops to a very low level after some days. With the help of backup services, the reward drops slowly. Results also show that relatively higher dependability can be achieved with the random replacement of failed service. With the help of adaptive maintenance mechanism based on Kalman filter, the reward stays steadily in a relatively high level.

The third experiment is to analyze effect of dependability's important degree on the Reward of composite service. The results in Figure 6(a) show that cumulative reward of composites without backup services drops to a very low level after some days in different important degree. With the help of backup services and adaptive maintenance mechanism based on Kalman filter, the Cumulative reward stays steadily in a relatively high level. And the higher important degree is, the reward will be much.



(a) Reward Without Adaptive Maintenance

(b) Reward with KF-based Adaptive Maintenance

Fig. 6. Dependability's important degree vs. reward

7 Related Work

To achieve dependable composite service, many efforts have been made. Related work mainly involves dynamic service composition, dynamic adaptation technology, monitoring and recovery technology, evaluation, replication technology and so forth.

F. Casati and H. Sun et al. [14, 15, 16] proposed a dynamic service composition method, only required functions are defined in the definition period and component services bind and instantiate at runtime. Composite service need communicating with service registry dynamically, find necessary component services according to the pre-established strategies. Possible strategies include choice based on the QoS [9, 17], or based on the semantic [18] and so on. This method can improve the flexibility of composition and dynamic adaptability. But dynamically searching and several remote interactions with the service registry, the efficiency will be low. And the quality of services cannot be guaranteed because the existing service registries cannot guarantee the authenticity of data and the state of registered services.

J. Harney and P. Doshi et al. [1][19] proposed adaptive methods -VOC and VOC $^{\varepsilon}$. VOC method can avoid unnecessary inquiries and reduce overhead by calculating the potential value of changed information. VOC$^{\varepsilon}$ is the improvement of VOC. VOC$^{\varepsilon}$ only monitors the service out of expiration time so as to gain better efficiency. They mainly concerned achieving verify of attributes efficiently and economically. While in KAF we consider the adaptive adjustment of services selection to meeting desired dependability steadily and economically in a volatile environment.

S. Guinea et al. [20, 21] proposed a self-healing service composition method through service state monitoring and recovery of fails. F. Tartanoglu et al. [22] based on the concept of cooperative atomic action and web service composition action, proposed a forward error recovery method to achieve fault tolerance and dependable composite service. K. Birman, et al. [23] proposed an extended WS architecture to achieve the high dependability of services, to support the error monitoring, reliable message transferring through components such as health control, reliable message, event notification and so on. However, these monitoring and recovery technologies do not consider monitoring and recovery efficiency, costs and rewards, and also they do not assess the effect of the monitoring and recovery in quantitative forms.

P. Godfrey et al. [25] presented the definition and calculating method of Churn, compared and analyzed different failure node replacing strategy based on some real traces. X. Li and K. Nahrstedt [26] presents a service composition approach with QoS guarantee to minimize the interference. R. Jurca et al. [2] proposed a reliable service quality monitoring mechanism based on customers' feedback, reduce lying and anti-conspiracy.

In addition, J. Salas et al. [27] proposed a framework to increase service availability. X. Ye and Y. Shen [28] proposed a middleware supporting reliable web services based on active replication technology. These studies concern availability and reliability of service individuals and their methods are qualitative. While KAF considers the dependability of composite service and supports quantitative assessment.

8 Conclusions and Future Work

In this paper, we have studied the important problem of dynamic maintenance of desirable dependability of composite services. We have proposed KAF, a system that provides closed-loop feedback based control over the dependability of composite services. The adaptive dependability control over a composite service is modeled as a Markov Decision Process. To maintain the dependability constantly over a given desirable value, it is particularly imperative to predict the state of the composite service. We have designed a Kalman-Filter based algorithm for accurate estimation. Comprehensive experiments have been conducted and the results have demonstrated that KAF is able to achieve the dynamic maintenance of dependability. At the same time, the cost of maintenance is reduced significantly on the long run. As a result, KAF is viable and valuable for Web service based applications.

Although significant achievements have been made so far, there are several interesting directions in our future work. On the one hand, we will investigate more comprehensive and finer grained QoS models for composite services, e.g., to define parameters more specific to Web services in more flexible way. On the other hand, we will extend to richer composition model other than supporting the general mode of sequence, parallel, choice and iteration, e.g., more complex relationships, such as compensation and transaction, will be considered.

Acknowledgments. We would like to thank Yanmin Zhu of Imperial College London for his fruitful suggestions and kind help in the writing of this paper. We would also like to thank the anonymous reviewers for their constructive comments and suggestions. This research was supported in part by China NSF grants (No.90412011 and No.60525209), China 863 High-tech Programme (No.2006AA01Z19A) and China 973 Fundamental R&D Program (No.2005CB321803).

References

1. Harney, H., Doshi, P.: Speeding up Adaptation of Web Service Compositions Using Expiration Times. In: Proc. of WWW (2007)
2. Jurca, R., Binder, W., Faltings, B.: Reliable QoS Monitoring Based on Client Feedback. In: Proc. of WWW (2007)

3. Wu, K., David, J., et al.: The Applicability of Adaptive Control Theory to QoS Design: Limitations and Solutions. In: Proc. of IEEE IPDPS (2005)
4. Guo, H., Huai, J., et al.: ANGEL: Optimal Configuration for High Available Service Composition. In: Proc. of ICWS (2007)
5. Avizienis, A., Laprie, J., et al.: Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transaction on Dependable and Secure Computing 1(1), 11–33 (2004)
6. Beth, T., Borcherding, M., Klein, B.: Valuation of Trust in Open Networks. In: Proc. of Conference on Computer Security (1994)
7. Golbeck, J.: Computing and Applying Trust in Web-based Social Networks. PhD thesis, University of Maryland (2005)
8. Abdul-Rahman, A., Hailes, S.: A distributed trust model. In: Proc. of NSPW (1997)
9. Liu, Y., Ngu, A.H.H., Zeng, L.: QoS Computation and Policing in Dynamic Web Service Selection. In: Proc. of WWW (2004)
10. Ran, S.: A model for web services discovery with QoS. ACM SIGecom Exchanges (2003)
11. Puterman, M.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley-Interscience, Chichester (1994)
12. Haykin, S.: Adaptive Filter Theory, 4th edn. Pearson Education, London (2002)
13. Kim, S., Rosu, M.: A survey of public web services. In: Proc. of WWW (2004)
14. Baresi, L., Guinea, S.: Towards Dynamic Web Services. In: Proc. of ICSE (2006)
15. Sun, H., et al.: Research and Implementation of Dynamic Web Services Composition. In: Zhou, X., Xu, M., Jähnichen, S., Cao, J. (eds.) APPT 2003. LNCS, vol. 2834, pp. 457–466. Springer, Heidelberg (2003)
16. Mennie, D., Pagurek, B.: An Architecture to Support Dynamic Composition of Service Components. In: Proc. of WCOP (2000)
17. Casati, F., et al.: eFlow: A Platform for Developing and Managing Composite e-Services. In: Proc. of AIWORC (2000)
18. Verma, K., et al.: METEOR–S WSDI: A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web Services. Journal of Information Technology and management (2005)
19. Harney, J., Doshi, P.: Adaptive Web Processes Using Value of Changed Information. In: Dan, A., Lamersdorf, W. (eds.) ICSOC 2006. LNCS, vol. 4294, pp. 179–190. Springer, Heidelberg (2006)
20. Guinea, S.: Self-healing web service compositions. In: Proc. of ICSE (2005)
21. Baresi, L., et al.: Towards Self-healing Service Compositions. In: Proc. of PRISE (2004)
22. Tartanoglu, F., Issarny, V., et al.: Coordinated forward error recovery for composite Web services. In: Proc. of SRDS (2003)
23. Birman, K., Renesse, R., Vogels, W.: Adding High Availability and Autonomic Behavior to Web Services. In: Proc. of ICSE (2004)
24. Ge, S., et al.: WebSASE: A Web Service-based Application Supporting Environment. In: Proc. of the 5th Northeast Asia Symposium (2002)
25. Godfrey, P., et al.: Minimizing Churn in Distributed Systems. In: Proc. of SIGCOMM (2006)
26. Li, X., Nahrstedt, K.: Minimum User-perceived Interference Routing in Service Composition. In: Proc. of INFOCOM (2006)
27. Salas, J., et al.: WS-Replication: A Framework for Highly Available Web Services. In: Proc. of WWW (2006)
28. Ye, X., Shen, Y.: A Middleware for Replicated Web Services. In: Proc. of ICWS (2005)