International Journal of Hybrid Information Technology
Vol.9, No.4 (2016), pp. 183-198
http://dx.doi.org/10.14257/ijhit.2016.9.4.17

A Survey of Quality Prediction Methods of Service-oriented Systems

Jinyu Kai[1,2], Huaikou Miao[1,2] and Honghao Gao[3]

[1] Shanghai University ,School of Computer Engineering and Science
[2] Shanghai Key Laboratory of Computer Software Evaluating & Testing,Shanghai,, P.R. China
aykjy@shu.edu.cn
[1] Shanghai University ,School of Computer Engineering and Science
[2] Shanghai Key Laboratory of Computer Software Evaluating & Testing,Shanghai,, P.R. China
[3]Computing Center, Shanghai University, Shanghai, P.R. China

Abstract

Service quality prediction plays a very important role in many fields related of service-oriented systems. Now, there are large numbers of approaches to predict the quality of service-oriented systems. However, it is impossible to apply one method to all prediction tasks without distinction. In reverse, unclearly to recognize the category of the prediction tasks will result in losing in all kinds of prediction approaches. In order to discover which methods are suitable for their corresponding service quality prediction scenarios, it is necessary to classify the prediction tasks and to classify the prediction approaches clearly. In this paper, a classification framework for predicting tasks for QoS of service-oriented systems is defined, four types of the prediction technologies are mainly introduced, and the research findings of quality prediction approaches for service-oriented system in literature using those four types of prediction technologies are listed and the increasing use of the method based on probabilistic model checking are specially concentrated on, and finally the result of this survey are given and the open research questions about predicting service quality by probabilistic model checking method are discussed.

***Keywords**: Service quality prediction; Statistical forecast approaches; Machine learning prediction approaches; prediction methods based on formal methods.*

1. Introduction

Component-based software engineering identifies in the concept of component as a building block the elementary unit in software architecture modeling, assessment, development, and management [1]. This follows from three main principles of software engineering: separation of concerns, modularization, and reuse. According to such perspective, a software system is an aggregate of (software) modules or components.

Service-oriented architecture (SOA) is increasingly adopted as a paradigm for building highly dynamic, distributed service-oriented systems. The same as the paradigm of component-based software engineering, the basis of modern service-oriented architecture software system is the component-based paradigm. A service-oriented system is realized by composing individual web services and its components assume the form of Web services. But contrasting to a traditional software component, the components of SOA software are not only their development, quality assurance, and maintenance but also their execution and their management which can be under the control of third-parties. As a consequence, the capabilities and quality of service-oriented systems more and more will

ISSN: 1738-9968 IJHIT

depend on the quality of its third-party services. Specifically, this means that service-oriented systems have to become resilient against failure of their third-party services. Quality prediction, together with self-adaptation, allows service-oriented systems to anticipate the need for adaptation and thus to prevent the actual occurrence of failures or to mitigate upcoming failures [2].

Making decision on self-adaptation is the key and service quality prediction is the basis. A good prediction basis is half the success of making decision on proactive self-adaptation. The overall decision making process can benefit greatly from accurate predicting. Therefore, service quality prediction plays a very important role in the proactive adaptation of service-oriented systems.

Now, there exist many prediction approaches. Each approach is applied in its application context. It is impossible to apply one method to all prediction tasks without distinction. In reverse, unclearly to recognize the category of the prediction tasks will result in losing in all kinds of prediction approaches. To recognize the category of the prediction task accurately and select an effective method appropriately, it is necessary to classify the prediction tasks and to classify the prediction approaches clearly.

Existing surveys on the prediction methods of quality of service oriented systems are most from the viewpoint of software architectures, in this paper, we give a more overall survey of the prediction methods from the view of prediction technology, we especially notice that the increasing use of prediction technology based on probabilistic model checking.

Our contribution is on the basis of surveying the research achievements on service quality prediction, we devote to classify the prediction tasks and to classify the prediction approaches, so as to discriminate the category of the prediction task quickly and identify the suitable prediction method appropriately when a service quality prediction task needs to be solved and this second is to discuss the scenarios that a prediction method based on probabilistic model checking are fitted for.

The paper is organized as follows. Section 2 defines a survey framework for predicting methods for service-oriented systems quality; Section 3 gives the Overview the prediction techniques, and Section 4 lists the research findings of the quality prediction approaches for service-oriented system, specially concentrates on the increasing use of method based on probabilistic model checking and finally; section 5 and section 6 give the result of our survey and discuss the open research questions about predicting service quality by probabilistic model checking method.

2. A Classification Framework for Predicting Tasks for QoS of Service-Oriented Systems

The prediction task exists in all fields, such as whether forecast, earthquake prediction, economic projection, lottery prediction, psychological prediction, population prediction, market forecast and so on. There still exist the prediction tasks in the field of quality of service-oriented systems. Similar to the study framework of quality of traditional component-based systems, the study of prediction of quality of service-oriented systems can be from the following perspectives.

2.1 From the View of the Transparency of Service Architecture

A component [3] is a self-contained, replaceable part of a software system that fulfils a clear function or a group of related functions in the context of software architecture. The components used can be developed in-house or be third party components, such as commercial off-the-shelf, modified off-the-shelf and open source components. The use of third party components has a number of potential benefits for component integrators, such as lower costs, better-tested products and maintenance support.

Software architecture reveals the components which may be used to compose whole software and the interaction behavior among components described by certain architecture relation which can roughly classified into three types, such as control, data, and use [4].

The paradigm of component-based software engineering is at the basis of modern service-oriented architectures (SOAs), in which the components assume the form of Web services selected and aggregated into a composed process which can itself become a service and can be recursively used in other service's composition. Specially, Service-oriented workflows in BPEL export and import information by using web service interfaces exclusively. As we see that IAAS (Infrastructure as a Service), PAAS (Platform as a Service) and SAAS (Software as a Service), everything in the system exposes itself and communicates with others in the form of service which is independent on the application platform and the programming language. For sake of convenient, in the rest of this paper, we let ***component service*** denote the services which are selected to build a service-oriented software and let ***composite service*** denote the service which be built by component services i.e. the service-oriented software.

Form the view of the transparency of service architecture, a service can be classified two categories, one is no transparency meaning without the service's internal structure and dependencies, and the other is total transparency meaning knowing the service' s internal structure and dependencies. Therefore, a no transparency service may be treated as a black box even if it is really a composite service.

Thus, the study of prediction of quality of service-oriented systems can be classified into the prediction methods of no-transparency service and that of transparency service; no matter it is a component service or a composite service. The prediction methods of transparency service are also called prediction method based on software architecture.

2.2 From the View of the Measure Manner of the Service Quality Attributes

To predict software quality, a quality model which consists of several quality attributes that are used as a checklist should be defined. The quality model is dependent of the type of software and forecasts can either use a fixed already defined quality model or define his own. There are several example of quality models in literature, such as the McCall's Quality Model, Boehm's Quality Model, FURPS/FURPS+ Model, Dromey's Quality Model, ISO 9000:2000 standards, ISO9126 Quality Model, and ISO/IEC 15504, IEEE' s several standards, Capability Maturity Models and Six Sigma etc.. Those models are focused around either processes or capability level or a set of attributes used to distinctively assess quality by making quality a quantifiable concept. Quality model actually reduce the notion of quality to a few relatively simple and quantifiable attributes [5].

To predict software quality, forecasts also need software measurements. A same quality attribute in different quality models may be defined by different measure approaches. For example, in ISO 9126:1 there are three approaches to software quality; internal quality, i.e. quality of code, external quality i.e. quality of execution and quality in use i.e. to which extent the user needs are met in the user's working environment. Therefore, a same quality attribute will be measured in different way according to its definition. Except that a same quality attribute can be defined differently in different quality models, a quality attribute may be directly measurable or indirect measurable. For the purpose to predict software quality, the indirect measurements must be derived by using directly measurable attributes. For example, creditability is an indirect measurable quality attribute; however, its measurement can be derived by measuring those directly measurable attributes, such as CPU utilization, Memory usage, and Network interface traffic volume etc.

Thus, the study of prediction of service quality attributes can be classified into the prediction methods of direct measurable quality attribute and that of indirect measurable quality attribute from the view of the measure manner of the service quality attributes.

2.3 From the View of the Feature of the Pertinent Historical Data

Prediction can be broadly considered as an activity to quantitatively calculate or estimate some future events or future aspects of a particular object for a specific goal, usually as a result of rational study or analysis of pertinent historical data on variables of the service quality attributes.

The pertinent historical data may be classified into two categories; one is the time series on past data of the variable that is being predicted, the other is the time series on past data of the variable that is related with being predicted ones.

Thus, correspondingly, the prediction tasks are classified into two categories. The first type is that which can be solved by using the past trend of a particular variable of the service quality attribute to base the future forecast of that variable. As this category of prediction tasks can be solved simply by using time series on past data of the variable that is being forecasted, these tasks are called time series prediction tasks.

The second category of prediction tasks also uses historical data to solve. But in predicting future values of a variable, the predictor examines the cause-and-effect relationships of the variable with other relevant variables.

This category of prediction tasks is solved by using past time series on many relevant variables to produce the forecast for the variable of interest. Prediction tasks falling under this category are called causal prediction tasks, as the basis to solve such prediction tasks is the cause-and-effect relationship between the variable predicted and other time series selected to help in generating the prediction.

Ignore the knowledge of the architecture of a transparency service, a transparency service is a no-transparency service, then, the prediction task of transparency service can also be classified into time series prediction tasks and causal prediction tasks.

However, in the opposite case, with the information about transparency service including of the data about the quality of component services and the information about the architecture relations which are used to describe the interaction behavior among components from different aspects such as control, data, and use, it is funny to predict the quality of transparency service of a certain moment, many key models such as Litterwood model, Cheung model, Laprie model, Kubat model, Gokhale et al. model, Ledoux model etc., have been developed and can be used to predict the quality of certain quality attribute or certain set of quality attributes of a transparency service. About this class of prediction methods, literature [3][6] give a detail survey; In this paper, we focus on the applying of three types of technologies introduced in section 3 on the different prediction tasks of a service; we don't classify the prediction tasks of a service according to its architecture.

2.4 From the View of the Problem Domain of Prediction Tasks

As introduced above, quality prediction provides important value reference to help the effective adaption of the system. Except for this, quality prediction can help to solve the problem like QoS-driven service selection or QoS-driven service recommendation, QoS-driven healing, QoS-driven optimization, or QoS-driven protection etc. those problems are general called QoS-*.

To predict the point quality of a component service helps to solve the problem in the area of QoS-driven service selection or QoS-driven service recommendation, and to predict the quality trend of a service no matter it is transparency or not helps to solve the problem in the area of QoS-driven service adaption, and to predict the quality trend of a composite service and to predict the point quality of a composite service and to analysis the influence that the quality of the component services do on that of the composite service help to solve the problem in the area of QoS-driven reconfiguration, optimization, protection, or healing etc..

But after all, all problems about QoS-* is correlated, thus, from the view of the problem domain, we do not classify the prediction tasks.

Figure 1 is a classification-perspective tree about the prediction tasks of quality of service-oriented software systems. Each branching represents from one perspective to classify the prediction tasks, the first branching represents from the view of the transparency of service architecture, the second branching represents from the view of the measure manner of the service quality attributes, the third branching represents from the view of the feature of the pertinent historical data on variables of the service quality attributes, and all those prediction tasks are to solve the problems in the areas of QoS-*.

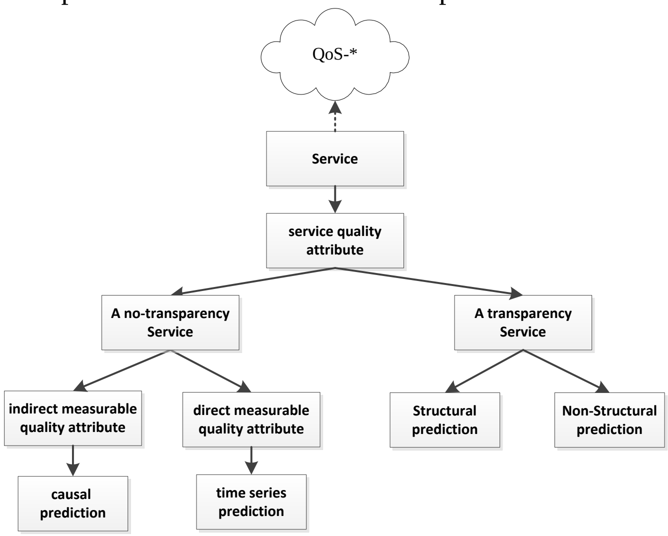

Figure 1. A Classification Tree about the Quality of Service Prediction Tasks

To sum up, the problems of quality prediction for service-oriented software systems can be typically classified into the following categories.

- To predict the quality of a service attribute or a set of service attributes of a causal prediction task of a no-transparency service.
- To predict the quality of a service attribute or a set of service attributes of a time series prediction task of a no-transparency service.
- To predict the quality of a service attribute or a set of service attributes of a causal prediction task of a transparency service.
- To predict the quality of a service attribute or a set of service attributes of a time series prediction task of a transparency service.

3. Overview the Prediction Techniques

There are numerous techniques that can be used to accomplish the goal of prediction. Those prediction techniques can be divided into two broad categories: qualitative and quantitative. In this paper, we limit our discussion to quantitative prediction techniques and manly discuss four types of prediction techniques, statistical forecast approaches, machine learning prediction approaches, aggregation functions prediction approaches and prediction methods leveraging of formal methods.

3.1 Statistical Forecast Approaches

In statistical discipline, the notion of forecasting is the same as that of prediction.

Quantitative forecasting methods are used when historical data on variables of interest are available. According to the category of the historical data, quantitative forecasting methods are classified into two classes.

The first class methods are based on an analysis of historical data concerning the time series of the specific variable of interest and possibly other related time series. As this category of forecasting methods simply uses time series on past data of the variable that is being forecasted, these techniques are called time series methods.

The second category of quantitative forecasting techniques also uses historical data. But in forecasting future values of a variable, the forecaster examines the cause-and-effect relationships of the variable with other relevant variables. Thus, this class of forecasting technique uses historical data concerning past time series on many relevant variables to produce the forecast for the variable of interest.

Forecasting techniques falling under this category are called causal methods, as the basis of such forecasting is the cause-and-effect relationship between the variable forecasted and other time series selected to help in generating the forecasts.

Time Series Prediction Models

Time series prediction models [7] work on a series of past observations data: $m_1, m_2, ..., m_i, ..., m_{n-1}, m_n$, with m_i being the observed data at time point i . They predict the future value for time point $n+1$, i.e., $\hat{m}_{n+1}$.

Some typical examples for such prediction models include:

- Last: This model uses the last observed value as the prediction value: $\hat{m}_{n+1} = m_n$
- Windowed Mean, BM(k): In this model, the arithmetic mean of the past k values $(1 \leq k \leq n)$ is used as the prediction value, where k is chosen to minimize prediction error: $\hat{m}_{n+1} = \frac{1}{k}\sum_{i=0}^{k-1} m_{n-i}$
- Simple Exponential Smoothing, SEM(α): BM treats past observations e-qually. Conversely, SEM places more weight on more recent observations $(\alpha \in [0,1])$: $\hat{m}_{n+1} = \alpha \cdot m_n + (1-\alpha) \cdot \hat{m}_n$

Causal Method of Forecasting

As mentioned earlier, causal methods use the cause-and-effect relationship between the variable whose future values are being forecasted and other related variables or factors. The widely known causal method is called regression analysis, a statistical technique used to develop a mathematical model showing how a set of variables are related. This mathematical relationship can be used to generate forecasts. In the terminology used in regression analysis contexts, the variable that is being forecasted is called the dependent or response variable. The variable or variables that help in forecasting the values of the dependent variable are called the independent or predictor variables.

Regression analysis that employs one dependent variable and one independent variable and approximates the relationship between these two variables by a straight line is called a simple linear regression. Regression analysis that uses two or more independent variables to forecast values of the dependent variable is called a multiple regression analysis. The forecasting technique using regression analysis for the simple linear regression case is briefly introduced. Regression analysis uses additional ways of analyzing the effectiveness of the estimated regression line in forecasting.

3.2 Machine Learning Prediction Approaches

Machine learning prediction approaches construct predict model based on machine learning techniques. There is a closed relationship between machine learning and statistics. In the literature [8], Larry Wasserman believes that both statistics and machine learning are research work based on data, while statistical emphasis on the prediction about the low-dimension space, the machine learning emphasis on the high-dimension space.

This class of approaches leverages data mining and machine learning capacities to train prediction models using historic monitoring data. Proposed solutions include variants of statistical methods, such as regression, as well as support vector machine, Bayesian classifier, multi-layer artificial neural network etc.

3.3 Aggregation Functions Prediction Approaches

Using aggregate functions to make predictions about the quality of the component system is a kind of widely-used quality prediction method based on white-box technology and commonly used to forecast the quality of structural system. The specific steps are as follows: firstly, the aggregate functions for each kind of service quality attributes are given in the three kinds of control structures, such as sequential structure, selection structure and loop structure, and then the quality of the system is calculated according to the structural characteristics.

3.4 Prediction Methods Leveraging of Formal Methods

Formal methods [9] are mathematical techniques, which are supported by formal system models developed by formal language and are with the aid of automated or semi-automated tools, for developing software and hardware systems. Mathematical rigour enables users to analyse and verify these models at any part of the program life-cycle: requirements engineering, specification, architecture, design, implementation, testing, maintenance, and evolution. In this paper, we focus on the researches using the formal methods to solve the problems encountered when developing high quality software systems.

Formal verification [10] is a technique which using the formal method to check the correctness of a software system during all phases in its life-cycle i.e. using mathematical reasoning on the formal system models to guarantee their absence of errors. When system errors do exists, formal verification can also serve as an effective bug-hunting technique.

To check the consistency of the software system specifications with its requirements within one or all phases in its life-cycle is one of the problems that formal verification may solve. Let $Impl$ denotes the software system specifications and $Spec$ denotes the requirements on the specifications. To prove the consistency of the software system specifications with its requirements is to prove one of the below relations is to be satisfied [11]:

1) The equivalence of the software system specifications with its requirements: i.e.
$Impl \equiv Spec$;

2) The implication of the software system specifications with its requirements: i.e.
$Impl \Rightarrow Spec$;

3) The semantic derivation of its requirements from the software system specifications: i.e.
$Impl \models Spec$;

The existing approaches to solve the above consistency checking problem can be classified into four categories [11], namely, Theorem proving, Model checking, Equivalence checking, and language containment. Of these four types of methods, model checking has received much attention and been widely used due to many advantages, for example the process of its checking does not need manual intervention and its high automatization and the process of its checking gives the counter examples when the software system specifications cannot satisfy its requirements .

Concretely speaking, model checking [12] is a formal verification method, which has been applied with great success to check the correctness of hardware device drivers and both cryptographic and communication protocols. It provides a methodology for exploring all possible behaviors of a system and comparing them to a requirement specification that describes the expected or allowed behaviors of the system [13].First, the correct behavior of these systems is formally specified and then a mathematical model that captures system is systematically constructed and analyzed in order to verify that the correctness properties are satisfied. In some cases, these are abstract models, designed by hand and based on expert knowledge of the system; in others, they can be extracted directly, from source code or a high-level design document. A key appeal of model checking is that, once the model and its correctness properties have been specified, the verification process itself is typically fully automated. Due to the exhaustive and formal nature of model checking, model checking applications have shown great promise for analyzing and debugging systems, especially in situations where traditional testing and analysis techniques have failed to provide conclusive results.

Probabilistic model checking [14] is a generalization of model checking for verifying quantitative properties of systems which exhibit stochastic behavior, for example due to failures or uncertainty about the environment. Like other forms of model checking, probabilistic model checking applies formal methods on a rigorous mathematical behavior model of a system to automatically prove certain precisely specified properties, e.g. "the probability of both sensors failing simultaneously is less than 0.0001 " and in this case the model is probabilistic, i.e. it captures the stochastic or uncertain aspects of a stochastic system. Therefore, probabilistic model checking can be used as a method to predict the quality of software. In this paper, we will emphasize on applying this method in predicting the quality of service-oriented software system.

4. State of the Art of the Quality Prediction Approaches for Service-Oriented System

4.1 The Applying of Statistical Prediction Methods

The statistical prediction methods, such as deduction from investigation, time series forecast, regression forecast and others, are widely applied in the area of prediction of quality of Web Services.

Collaborative filtering based approaches have been employed in service computing to predict the QoS of previously unknown services.

Shao et al. proposed a forecasting method to predict personalized service quality which is based on collaborative filtering. This method gives personalization service quality prediction to the services which are required by users and it is mainly based on the experience of similar consumers, and it is often used to recommend personalized services during services selection. And their paper focuses on the response time and availability.

To add the ability to predict the quality of service missing in UDDI, UX [16]and UDDIe [17] are proposed, which are UDDI systems with the ability to predict the service's future performance by average function or arithmetic mean of QoS data which is submitted by consumers to set the average QoS value to facilitates requests to discover service with good qualities.

Zheng [18] proposes a Web Service response time dynamic prediction approach where self- correlation analysis is used to identify the basic characteristics of the time series of response time data, and then, according to the feature pattern, the corresponding prediction model is constructed, thus the response time of the service invoked in the next period is dynamically predicted.

To help users select appropriate Web Service, Hua [19] proposes a Web Service QoS prediction method based on the models of ARMA(autoregressive and moving average) and ARIMA(autoregressive integrated moving average). According to the history collected data to determine the parameters of the models of ARMA and ARIMA and then using the models to predict the QoS of a service in the short future time.

Rodrigo N. Calheiros, Enayat Masoumi, Rajiv Ranjan, Rajkumar Buyya [20]adopt Autoregressive Integrated Moving Average(ARIMA) model to predict the workload of a cloud application.

Wang etc. in literature [21] proposes an approach for component level online time series reliability prediction based on probabilistic graphical models.

Andreas Metzger, Osama Sammodi, and Klaus Pohl [2] discuss an approach that aims to collect additional data points by using online testing method to overcome the limitations of time series prediction so as to improve prediction accuracy.

Hongbing Wang, Cheng Wan [22]adopt Stochastic Differential Equation to predict the quality failure of Service-oriented system of systems; they adopt Brownian motion to model the quality track, which does not need to estimate the distribution and similarity of other services and use direct incremental model to estimate the features of historical quality track, and use user’s conditional preference to balance multi-attributes’ influences on risk warning, and apply weighted conditional preference net to model and calculate the attributes’ influences on quality failure prediction.

Chen Wang and Jean-Louis Pazat [23] adopt the Program Evaluation and Review Technique (PERT) as an efficient tool for rapid runtime estimation of global execution time of the workflow.

4.2 The Applying of Machine Learning Prediction Methods

The machine learning prediction methods, such as Support Vector Regression, Bayesian Approach and others, are widely applied in the area of prediction of quality of Web Services.

After investigating the relationship between TCP throughput and measurements of path properties including available bandwidth, queueing delays, and packet loss, Mirza et al. [24] uses Support Vector Regression which is a powerful machine learning technique that has shown good empirical performance in many domains for throughput prediction.

Different modeling pattern recognition and machine learning techniques are employed in software categorization and Nizar Bouguila, Jian Han Wang and A. Ben Hamza [25] introduce an unsupervised Bayesian algorithm based on finite Dirichlet mixtures, employed to software prediction by categorizing modules into fault-prone and non-fault-prone.

C.Jin [26] proposes a GA-SA-SVR model where genetic algorithm(GA) and simulated annealing (SA) are integrated into a new optimize algorithm, called GA-SA, which is applied to support vector regression model (SVR) for predicting software reliability.

Mirza Silic etc.2012 in the literature [27] proposes a model for reliability prediction of atomic web services for recommendation which considers user-, service- and environment-specific parameters of the invocation context and aggregate the available previous invocation data using K-means clustering algorithm.

4.3 The Applying of Aggregation Functions Prediction Methods

As the control structure of the component system in the design stage is in line with the structured design thought, therefore, it is often used of the aggregation function for quality prediction in the design phase of the service system.

In this paper [28], they present a predictive QoS model which is better suited for production workflows [29] since they are more structured, predictable, and repetitive. Production workflows involve complex and highly-structured processes, whose execution requires a high number of transaction accessing different information systems. These characteristics allow the construction of adequate QoS models for workflow tasks.

The QoS model is the collection of the aggregation functions which are used to compute the quality of service for workflows automatically based on atomic task QoS attributes

4.4 The Applying of Formal Model Checking Prediction Methods

Just as the use of aggregate function to predict the quality of the service system is a kind of prediction method based on white-box technology, the forecasting method based on probabilistic model checking is also a prediction method based on white box technology; unlike the approach based on aggregate functions which can only predict the quality of the structural system, the prediction method based on probabilistic model checking can predict the quality of both the structured system and the unstructured system; for the service system in the running stage, influenced by system running environment and client operation tools (browsers), in the course of running, even if the service system that is in conformity with the design idea of structure control may appear the unstructured operation situation. Therefore, it is very suitable for the use of state machine to model the service system in the running phase.

Gallotti et al in [31] propose an approach named ATOP (from Activity diagrams TO Prism models) which derives a probabilistic model from an abstract description of service compositons (activity diagram) to feed the PRISM tool for the evaluation phase and this approach is a methodology conceived for evaluating system at design-time.

Gao et al in [32] the authors propose a two-phase method involving monitoring and prediction with the aim of monitoring at run-time the reliability of compositional web services which exhibit random behavior. This method takes advantage of the probabilistic model checking technique; it focuses mainly on reliability by providing a DTMC-based Markovian model.

In [33] the authors propose a general CTMC probabilistic model for performance indicators in which both state and transition are parameterized resulting in a model adaptable at run-time. They have realized and presented a validating prototype----built on top of the PRISM Model Checker.

5. Result of Survey

In this paper, the overviews of the methods of prediction the quality of the service-oriented system is presented from three kinds of technologies. All of those approaches can be used to predict the quality of service-oriented systems. Essentially, those approaches can be classified into two categories; one is to adopt the black-box methodology, which treats a service-oriented software system as a monolithic whole, to model the quality attribute assessment model of the system. During the course of modeling the quality attribute of a service-oriented system, only its interactions with external environment is considered, without an attempt to model its internal structure. Usually, in such model, no information other than the data related of special quality attribute is used. The common feature of adopting black-box methodology to analysis the quality attribute of a service-oriented software system is to model the stochastic failure process, developing some parametric model of cumulative number of failures over a finite time interval. data obtained while the system is tested or collected from its log file when it is running are used to estimate model parameters or to calibrate the model, then exploit this model to analysis the quality attribute of the system.

The other class of approaches to assess the reliability of component based software is based on white-box technology which takes into account the information about the architecture of the software made out of components. The approaches based on white-box technology to assess the reliability of component-based software are also called architecture based approaches which model the reliability of the software combining its architecture with its failure behavior.

As can be seen from the above survey, prediction methods based on machine learning technique and statistical forecast methods based on causality are often used to predict the quality of services that cannot be measured directly. Exploiting the association or causal relationship between the service qualities attributes that can be measured directly and this service quality attribute, the attribute value of the service quality at some time can be predicted by using the machine learning method or statistical causal prediction method. Because this class of methods does not involve the relationship between the components of the service system, those methods belongs to the approaches based on black-box technique.

As well as prediction methods based on machine learning and statistical forecasting method based on causality, statistical forecasting method based on time series also belongs to the approaches based on black-box technology which are often used to predict the future service quality attribute values according to the attribute data of an the quality of service in historical time series and these service quality attributes can be directly measured, or can be obtained by machine learning method or causal association method.

The prediction methods based on statistics and learning machine belong to the category based on black-box technique. Although those approaches based on black-box methodology may be used to analysis and predict the reliability of a component service or a composite service respectively, obviously, they can’t be used to analysis the influence that the reliability of the component services do on that of the composite service.

Different from the above two kinds of methods, the prediction method based on formal probabilistic model checking forecast service quality attributes involved in the service system of the relationship between the internal components, therefore, formal forecasting method of probabilistic model checking is a kind of prediction method based on the technology of white box. To use formal probabilistic model checking forecasting service quality attributes need to satisfy the following conditions, 1) service quality attributes need to be characterized by probability; 2) the overall quality of service is determined by the quality of components and the background of measurement of the quality attributes of component services conforms to the Markov property. Figure 2 shows the correspondent method for different prediction tasks.

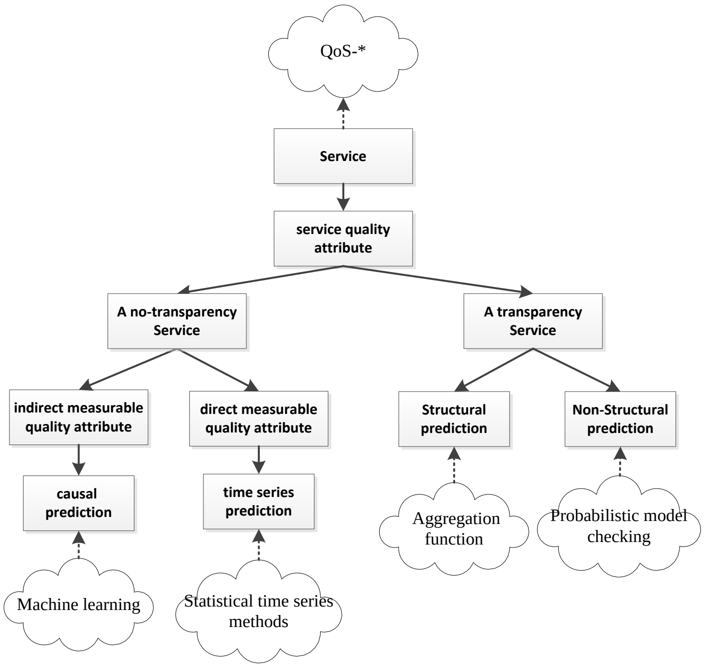

Figure 2. A Classification Tree about the Quality of Service Prediction Tasks

This survey report clarifies the relationship between the forecast methods and the prediction tasks, and lays a good foundation for the selection of suitable forecasting methods for the prediction task.

6. Discussion

Although the use of formal probabilistic model checking to predict the quality of service quality attributes need to meet certain conditions, however, there is a large number of scenarios that are in line with this condition in the service system, Therefore, the use of formal probabilistic model checking to forecast the quality of service attributes has become an important method to predict the quality of service system.

However, it is worth further exploration and research in the process of the prediction theory and technology combined with a formal method used to guide the quantitative forecast of service system quality based on probability model checking method, three aspects are listed as follows:

(1) The construction of service quality model

Service quality model directly determines the significance of the results.

Although the underlying theory of the approaches based on probability model checking technique of quantitatively prediction of the quality of the service system is the same, take the different influencing factors on the quality of the service system into consideration and construct different the service quality model and get different results.

Therefore, it is an important research topic for quantitative forecasting of the quality of service system based on probabilistic model checking to further investigate the factors that affect the quality of the service system and construct the service quality model which is in line with the intention of the predictor.

(2) The description of service quality requirements

During the course of using the approach based on probability model checking to quantitatively forecast the quality of the service system, system service quality requirements to be tested need to be described in a probabilistic temporal logic. The temporal logic language proposed by Israel computer theorists Pnueli in 1977 is a formal language supported by rigorous mathematical proposition logic which skillfully combines the temporal operators with the first-order logic proposition and abandons the trivial processing of concrete time however it can describe the nature of the system behavior change over time. Probabilistic temporal logic is an extension of temporal logic. Although the probabilistic temporal logic formula is readable and easy to handle by machine, using probabilistic temporal logic language requires more professional knowledge. Therefore, put forward higher request to the workers who use this language. Thus how to describe the quantitative system service quality in order to achieve the balance the maneuverability between the operator and machine is a problem worthy of further research.

(3) The efficiency of probability model checking

The reason why the model checking technique cannot be applied widely is due to the state explosion. Probabilistic model checking extends the technology of model checking, but it also faces the problem of state explosion. However, the probabilistic model checking has its own particularity, can use its particularity to solve the problem of the efficiency of the quantitative properties inspection probability model is a problem worthy of further research.

The solutions of these problems will further promote the application of formal probabilistic model checking techniques in the prediction of service system quality attributes.

Acknowledgement

This work is supported by National Natural Science Foundation of China (NSFC) under grant No. 61170044 and 60970007, and Young University Teachers Training Plan of Shanghai Municipality under Grant No. ZZSD13008.

References

[1] D.-B. Zhang, Z.-B. Sun and C.-Q. Zhu, "Reliability analysis of retaining walls with multiple failure modes", Journal of Central South University, vol. 20, no. 10, **(2013)**, pp. 2879-2886.

[2] A. Metzger, C. H. Chi, Y. Engel and A. Marconi, "Research challenges on online service quality prediction for proactive adaptation", Software Services and Systems Research - Results and Challenges (S-Cube) Workshop on European IEEE, **(2012)**, pp. 51-57.

[3] A. Immonen and E. Niemelä, "Survey of reliability and availability prediction methods from the viewpoint of software architecture", Software & Systems Modeling, vol. 7, no. 1, **(2008)**, pp. 49-65.

[4] A. Purhonen, E. Niemelä and M. Matinlassi, "Viewpoints of DSP software and service architectures", Journal of Systems & Software, vol. 69, no. 69, **(2004)**, pp. 57-73.

[5] P. Berander, L.-O. Damm, J. Eriksson, T. Gorschek, K. Henningsson, P. Jönsson, S. Kågström, D. Milicic, F. Mårtensson, K. Rönkkö and P. Tomaszewski, "Software quality attributes and trade-offs", Blekinge Institute of Technology, **(2005)**.

[6] K. Goševa-Popstojanova and K. S. Trivedi, "Architecture-based approach to reliability assessment of software systems", Performance Evaluation, vol. 45, no. 2-3, **(2001)**, pp. 179–204.

[7] A. Metzger, O. Sammodi and K. Pohl, "Accurate Proactive Adaptation of Service-Oriented Systems", Assurances for Self-Adaptive Systems, Springer Berlin Heidelberg, **(2013)**, pp. 240-265.

[8] B. Anne-Laure and S. Matthias, "Machine learning versus statistical modeling", Biometrical Journal, vol. 56, no. 4, **(2014)**, pp. 588-593.

[9] J. Woodcock, P. G. Larsen, J. Bicarregui and J. Fitzgerald, "Formal Methods: Practice and Experience", Acm Computing Surveys, vol. 41, no. 4, **(2009)**, pp. 1729-1739.

[10] S. Chaki, E. M. Clarke, A. Groce, S. Jha and H. Veith, "Modular Verification of Software Components in Software Engineering, IEEE Transactions, vol. 30, no. 6, **(2004)**, pp. 388-402.

[11] C. Kern and M. R. Greenstreet, "Formal verification in hardware design: a survey", Acm Transactions on Design Automation of Electronic Systems, vol. 4, **(1999)**, pp. 123-193.

[12] O. Grumberg, D. Peled and E. M. Clarke, "Model checking", Cambridge: MIT Press, **(1999)**.

[13] R. Kumar, "Using live sequence chart specifications for formal verification of systems", Dissertations & Theses-Gradworks, **(2008)**.

[14] D. Parker, "Implementation of Symbolic Model Checking for Probabilistic Systems", University of Birmingham, **(2002)**.

[15] L. S. Shao, Z. Li, J. F. Zhao, B. Xie and M. Hong, "Web Service QoS Prediction Approach", Journal of Software, vol. 20, **(2009)**, pp. 2062-2073.
[16] C. Zhou, L.-T. Chia, B. Silverajan and B.-S. Li, "Ux-An Architecture Providing QoS-Aware and Federated Support for UDDI", Proceedings of the International Conference on Web Services, ICWS, **(2003)**.
[17] A. S. Ali, R. J. Al-Ali, O. F. Rana and D. W. Walker, "Uddie: an extended registry for web service", Proceedings of the Service Oriented Computing: Models, Architectures and Applications, SAINT, **(2003)**.
[18] X.-X. Zheng, J.-F. Zhao, Z.-W. Cheng and B. Xie, "Web Service Response Time Dynamic Prediction Approach", Journal of Chinese Computer Systems, vol. 32, no. 8, **(2011)**, pp. 1570-1574.
[19] Z. Hua, M. Li, J. F. Zhao and B. Xie, "Web Service QoS Prediction Method Based on Time Series Analysis", Journal of Frontiers of Computer Science and Technology, **(2013)**, pp. 218-226.
[20] R. Calheiros, E. Masoumi, R. Ranjan and R. Buyya, "Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications' QoS", IEEE Transactions on Cloud Computing, vol. 1, **(2014)**, p. 1.
[21] L. Wang, H. Wang, Q. Yu, H. Sun and A. Bouguettaya, "Online reliability time series prediction for service-oriented system of systems. Lecture Notes in Computer Science", vol. 8274, **(2013)**, pp. 421-428.
[22] H. Wang and C. Wan, "Quality Failure Prediction for the Self-Healing of Service-Oriented System of Systems", Web Services (ICWS), IEEE International Conference on IEEE, （**2014)**, pp. 566-573.
[23] Wang, Chen and J. L. Pazat, "A Two-Phase Online Prediction Approach for Accurate and Timely Adaptation Decision", Services Computing (SCC), IEEE Ninth International Conference on IEEE, **(2012)**, pp. 218 - 225.
[24] M. Mirza, J. Sommers, P. Barford and X. Zhu, "A machine learning approach to tcp throughput prediction", Networking IEEE/ACM Transactions, vol. 18, no. 4, **(2010)**, pp. 1026-1039.
[25] N. Bouguila, J. H. Wang and A. Hamza, "A Bayesian approach for software quality prediction", Intelligent Systems, 4th International IEEE Conference, IEEE, **(2008)**, vol. 11-54.
[26] C. Jin, "Software reliability prediction based on support vector regression using a hybrid genetic algorithm and simulated annealing algorithm", Let Software, vol. 5, **(2011)**, pp. 398-405.
[27] M. Silic, G. Delac and S. Srbljic, "Prediction of atomic web services reliability based on k-means clustering", Proceedings of the 9th Joint Meeting on Foundations of Software Engineering, **(2013)**.
[28] J. Cardoso, J. Miller, A. Sheth and J. Arnold, "Modeling quality of service for workflows and web service processes", Analyze \& Monitor Workflow Qos Process Qos Compute Web Service Qos Quality of Service, vol. 1, no. 3, **(2002)**, pp. 281--308.
[29] S. McCready, "There is more than one kind of workflow software", Computerworld November, vol. 2, **(1992)**, pp. 86-90.
[30] V. Grassi and S. Patella, "Reliability prediction for service-oriented computing environments", Internet Computing IEEE, vol. 10, no. 3, **(2006)**, pp. 43-49.
[31] S. Gallotti, C. Ghezzi, R. Mirandola and G. Tamburrelli, "Quality Prediction of Service Compositions through Probabilistic Model Checking. Quality of Software Architectures", Models and Architectures, **(2008)**.
[32] H. Gao, H. Miao and H. Zeng, "Predictive web service monitoring using probabilistic model checking", International Journal on Applied Mathematics & Information Sciences, vol. 6, no. 1, **(2013)**.
[33] G. Cicotti, L. Coppolino, S. D'Antonio and L. Romano, "Big data analytics for qos prediction through probabilistic model checking", Eprint Arxiv, **(2014)**.
[34] A. Cuomo, M. Rak and U. Villano, "Journal Performance prediction of cloud applications through benchmarking and simulation", International Journal of Computational Science & Engineering, vol. 11, no. 1, **(2015)**.

Authors

Jinyu Kai. She is currently a Ph.D. student in the School of Computer Engineering and Science of Shanghai University, China. Her research interests include formal methods, software engineering, Web service application development, web application development and quantitative model checking.

Huaikou Miao is currently a professor in Computer Engineering and Science at Shanghai University, China. His research interests include formal methods and software engineering.

Honghao Gao received the Ph.D degree in the School of Computer Engineering and Science of Shanghai University, Shanghai, China, in 2012. His research interests include Web service and model checking.

