Analysis of Internet Ware Performance Modeling Based on the Fusion of Process Group Component Model

Zhang Hongxiang

Xi'an University, 710065

Keywords: Performance; Process Group Component Model; Structural Analysis; Service

Abstract: The performance modeling analysis is an important subject for the Internet ware running in the open, dynamic, and difficult-to-control Internet environment. However, the performance computation in many researches at present is based on the fusion of black box, in which in-depth consideration of the system structure is not carried out, and the evaluation index is excessively single. Therefore, a kind of Internet ware performance modeling analysis based on the fusion of the process group component model is put forward in this paper. In this model, multi-layer Internet ware performance modeling analysis index system is established through the structural analysis of the Internet ware in accordance with its structural model. The experiment has shown that this model can clearly and objectively analyze the performance of the Internet ware. In addition, it can provide help to the design, development and deployment of the Internet ware.

1. Introduction

Entering the 21st century, network represented by the Internet has gradually integrated into all aspects of human society, which has greatly promoted the process of globalization and expanded the development space for the information technology and applications [1]. On the other hand, the Internet is growing into a "Unified computer" composed of a huge and increasing number of computing devices. Compared with the traditional computer systems, the support that Internet can provide for solving the problems in the application field has gained both quantitative and qualitative leap [2-3]. In order to adapt to the significant change in these application areas and information technology, new software forms that integrate the Internet environment have been formed under the technical support of the object-oriented and software components, namely, the Internet ware [4-5]. Since the Internet ware is a union of a set of software entities distributed in a wide area and provided by the second party in the form of service [5-6]. Among them, the source code and the internal structure of most services are usually not visible, which can only be accessed through the predefined interface specifications [7-8]. At the same time, in order to ensure the personalization and flexibility of the software, its development, deployment, operation and maintenance environment has changed from a traditional closed, static and controllable environment to an open, dynamic and difficult-to-control Internet environment. And the analysis on its performance modeling has become very important [9-10].

In this paper, the analysis method for the Internet ware performance modeling based on the fusion of process group component model is analyzed. On the basis of the analysis of the Internet ware system architecture, the system structure model is established. Based on this model, the Internet ware performance modeling analysis system based on the fusion of the process component model is established. At the same time, it provides good basis for the user to make the best choice for the optimal scheme to the third-party entity. In addition, it also has guiding significance for the design, development and deployment of the Internet ware.

2. Structural Analysis of Internet Ware

Compared with the software with the same component, the basic elements that make up the Internet ware are not the components or modules, but the software entities, which are the entities provided by the software developer or the ordinary users that can satisfy the services required by a

  DOI: 10.25236/isrme.2019.036

certain Internet ware. In addition, a kind of synergy can also be achieved autonomously between the entities. All the entities are distributed in the Internet software system. They are independent of each other, operate autonomously, are not under the unified control of any institution or organization. They collaborate in an open, dynamic and ever-changing environment. The Internet ware not only serves the users in different time zones but also serves as a software entity to meet the other requests for the Internet ware that arrives at any time. The above system determines that the structure of the Internet ware is very complicated. Therefore, in order to analyze the performance of the Internet ware system, a clear and proper analysis of its compositional structure is required. We define its structural dependency and semantic dependency in accordance with the characteristics of the Internet software.

It is assumed that the entity set of the software system can be divided into three sub-sets in accordance with the different functions. The sub-set A has entities *A1*, *A2*, *A3*, …; the sub-set B has the entities *B1*, *B2*, *B3*, …; the sub-set C is the basic entity layer, which has the entities *C1*, *C2*, *C3*, …, A_1 depends on B_1 and C_2 in the structure; while C_2 needs to be dynamically bound to A_1 in the data flow of the completed function; and so on. Therefore, the system structure dependency diagram can be obtained, as shown in Figure 1 (a). For the convenience of subsequent processing, it is transformed into three structurally dependent trees. Each of them takes the entities A_1, A_2 and A_3 as the root nodes to form Figure 1(b) ~ (d) as the following.

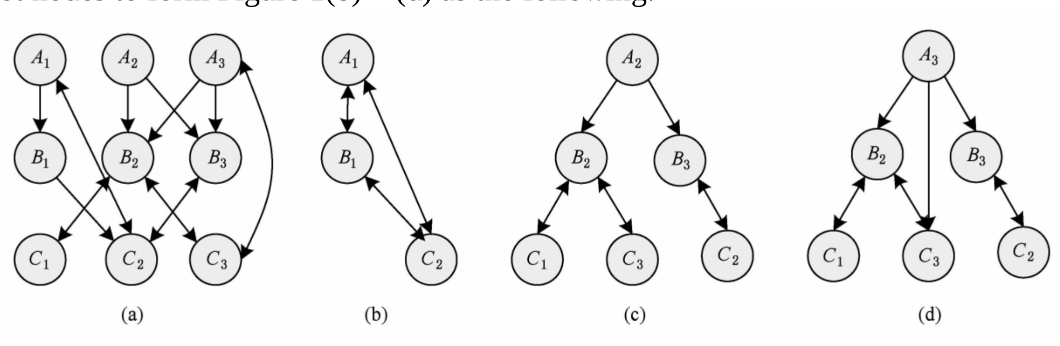

Figure 1 Structure Dependency Diagram and the Structure Dependency Tree

For the semantic dependency relationship within each entity layer, as there is a sequential relationship between the entities, they are represented by the matrix. If the entity layer has *N* entities, the matrix is *N*×*N* as the following.

$$\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix} a_{ij} = (0,1)$$

When a_i in the entity layer must be used before a_j, $a_{ij} = 1$; otherwise, $a_{ij} = 0$.

3. Analysis of Internet Ware Performance Modeling Based on the Fusion of the Process Component Model

On the basis of the analysis of the Internet ware structure, the Internet ware performance modeling analysis index system model based on the fusion of the process group component model is put forward in this paper. It has integrated the traditional index and the specific index of the Internet ware, covering the dynamic indexes and the static indexes. It has good guiding significance for the design, development, deployment and analysis of Internet ware.

As a kind of software system, the performance of the software is affected by a number of factors, which should be formed by a number of of feature attribute indexes. At the same time, the Internet ware is a kind of union of the software entities that are distributed in a wide area and provided by the second party. Among them, each basic entity is a component first, which can ensure that the individual entities can be independently deployed, operated and evolved. Therefore, it is applicable to the performance modeling analysis model of the components. However, the features of the Internet ware such as the autonomy, evolution, cooperativity, polymorphism, responsiveness and so on have determined that its basic entities cannot be completely equivalent to the components. The "Natural" heterogeneity of the Internet determines that the basic entities of the Internet ware have different characteristics and their performances are also not the same. The modeling analysis should also include the unique indexes. We fuse the concept of the performance feature attribute to establish a performance modeling analysis system. In reference to the ISO/IEC 9126 standard for the analysis of the software quality characteristics, the software performance features are decomposed and added with the measurement of the software components that is distinguished from the measurement of the reusability of the special quality features of general software. At the same time, the performance modeling analysis index system with the tree structure is established in accordance with the characteristics of the Internet ware, as shown in Figure 2. Due to the limited space, the indexes in the lowermost layer are not all listed. And only a few examples are taken for illustration in the following.

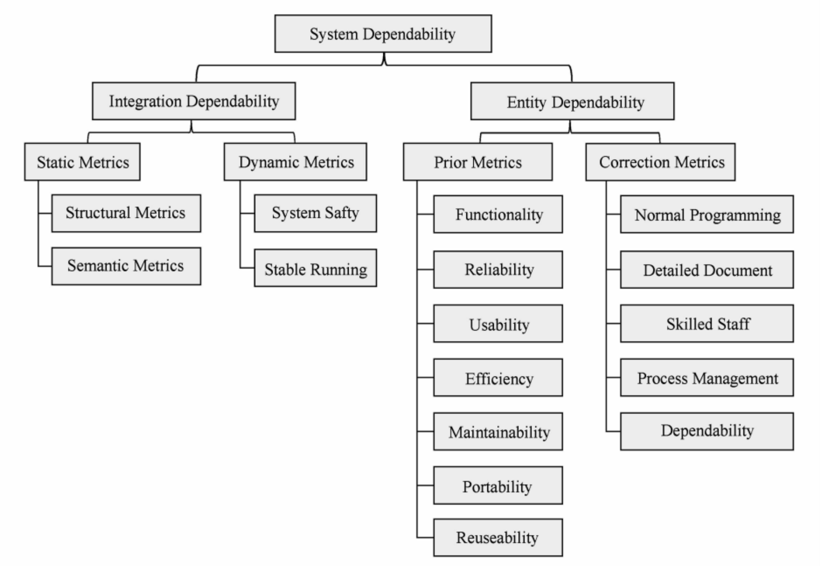

Figure 2 Internet Ware Performance Modeling Analysis System

4. Calculation Method for the Performance Modeling Analysis Index

In the structural model based on the fusion of the Internet, the static performance index and the dynamic performance index are defined. Performance is regarded as the conditional probability of the performance modeling analysis services that can be provided when the system meets the performance modeling analysis index. That is, it is assumed that the performance index is $P(M)=\{P(M_i)\, i=1,2,\cdots,\ n\}$, then the system performance $P(D_s \quad M_i)$ can be calculated by the Bayesian inference as the following:

$$P(D_s|M)=\frac{P(M|D_s)P(D_s)}{P(M)} \tag{1}$$

Bayesian reasoning has integrated the following assumptions: The quantity to be investigated follows a certain probability distribution. In addition, it can be inferred on the basis of the probabilities and the observed data so as to make the optimal decisions. When the Bayesian method is used, each training example observed can incrementally reduce or increase the estimated probability of a certain hypothesis, which can provide a more reasonable approach to learning than the other algorithms. The prior knowledge can be combined with the observation data to make the decision on the final probabilities of the assumptions, and its form can be as the following: 1) The prior probability of each candidate hypothesis; 2) The probability distribution of each possible hypothesis on the observable data. Predictions can be made for the elements at the upper layer with a number of hypotheses. And their probabilities can be used for the weighting.

4.1. Static Performance Index

The static performance index is mainly used to carry out measurement from the aspect of system structure dependency and the semantic dependency. Oriented towards the system design stage, analysis is carried out on the performance of the system after the assembly in accordance with the choice of different entities.

1) Entity performance. Each entity in the Internet ware can be viewed as a separate component. And its performance can be analyzed using the component performance indexes (such as the functionality, reliability, ease of use, efficiency, maintainability, portability, reusability and so on). At the same time, the condition probability required by its quality to meet the quality standards of the entity can be used for correction, that is, after the entity executes in accordance with the quality standards in the process of the development, testing, deployment and so on, the impact of the probability of its service performance modeling analysis on its own performance.

In Figure 3, the performance of the entity C_1 can be calculated from its priori index, and at the same time, it is corrected by using the analysis index. It is assumed that the performance of C_1 is $P(D_{C1} \quad M)$. Then from the equation (1), the following can be obtained:

$$P(D_{C1} \mid M) = \frac{P(M \mid D_{C1}) P(D_{C1})}{P(M)} \tag{2}$$

In which, $P(D_{C1})$ stands for the priori index, that is, the initial analysis of the entity performance that is composed of the traditional performance modeling and analysis indexes such as the functionality, reliability, ease of use, efficiency, maintainability, portability, reusability and so on. Both $P(M \quad D_{C1})$ and *P (M)* can be obtained by analyzing the indexes. It is assumed that each priori index is $P(M_p) = \{P(M_{pi}) \quad i = 1, 2, \cdots, n\}$, and each analysis index is $P(M_e) = \{P(M_{ei}) \quad i = 1, 2, \cdots, n\}$, hence the equation (2) is transformed into the following

$$P(D_{C1} \mid M) = \frac{P(M_e \mid D_{C1}) P(M_p)}{P(M_e)} \tag{3}$$

2) Integrated performance. In the Internet ware system, for the performance of the upper layer entity, the impact of its basic entity performance should also be taken into consideration, that is, the impact of the underlying entity that it depends on the structural dependency and the semantic dependency between the entities is expressed through the integrated performance index.

In Figure 3, for the performance of the entity B_2, in addition to the entity performance calculation by itself, it shall also be corrected through the conditional probability of the entities C_1 and C_3. It is assumed that the performance of the entity B_2 is $P(D_{B1} \quad M)$, and the performance of the entity that it depends on is $P(D_{Dp}) = \{P(D_{Dpi}) \quad i = 1, 2, \cdots, n\}$. In addition, the impact of the structural performance and the semantic performance should also be taken into consideration. The structural performance is the performance of the interdependence of the structures between the entities, that is, the performance of the system structure in all aspects, such as the language compatibility, the interface consistency and so on. The semantic performance is the performance of the semantic dependency between the entities, that is, the performance in the aspect of the system semantics, such as the data replication consistency, the fault detection, the fault recovery and so on. The group component model of this process is composed of the entities of the element $a_{i,j} = 1$ in the dependency matrix. Therefore, the values for the structural performance and the semantic performance are both 1. Then the performance calculation formula for the entity B_2 is as the following

$$p\left(D_{B2}|M\right)=\frac{P\left(M_e\mid D_{B2}\right)P\left(M_p\right)}{P\left(M_e\right)}\times P\left(D_{B2}\mid D_{Dp}\right) \tag{4}$$

$P\left(D_{B2}\ D_{Dp}\right)$ stands for the performance of B_2 upon the analysis of the base node performance modeling. And the value of $P\left(D_{B2}\ D_{Dp}\right)$ before the system is put into operation can be set to a certain initial value.

4.2. Dynamic Performance Index

Due to the openness, dynamics and hard-to-control nature of the environment in which the Internet ware is located, various situations that may cause the degradation of the performance may occur at any time during the process of deployment and operation. Therefore, in the performance modeling analysis of the Internet ware, the dynamic performance indexes such as the security performance, the stable operating performance and so on should be included as well. In the analysis of the current service quality of the entity through the dynamic performance indexes, when the performance degradation is identified, the system can switch from the current entity to the entity that has the same function and relatively high performance. This has fully reflected the autonomy, evolution, collaboration, polymorphism, reactivity and other features of the Internet ware. The security performance refers to the probability that a certain security policy can ensure that sensitive resources and data will not be accessed or obtained illegally by an attacker when the Internet ware system is maliciously attacked during the operation. The stable operation performance refers to the probability that the Internet ware system normally accomplish the specified function under the specified conditions and within the specified time. Both are the probability values in continuous interval of $[0,1]$, which can be measured by the error injection method. As the dynamic performance is an index that mainly examines the performance of the system during the operation, and the calculation method has little relationship with the main content of this paper, its value is set to 1 in the subsequent calculation process, that is, the impact of dynamic performance on the overall performance of the system is not taken into consideration.

4.3. Calculation Method for the Overall System Performance

Another layer of nodes is added above the top layer of the entities, that is, the performance of the entire system. The calculation starts from the performance of the lowest layer of the entities and is carried on layer by layer up until finally the performance of the topmost layer of entities can be calculated. And the performance of the entire system is determined by the performance of each entity at the topmost layer.

In accordance with the above method, the performance of the basic entities C_1, C_2 and C_3 can be calculated. And the performance of the entity B_2 can be calculated through the calculation of the C_1 and C_3. And the performance of the entity B_3 can be obtained from the entity C_2. Similarly, the performance of the entity A_2 can be calculated from the performance of the entities B_2 and B_3. Finally, the overall performance of the system is obtained. The calculation formula is as the following:

1) Entity performance of each basic entity

$$C_1: P\left(D_{C1}\mid M\right)=\frac{P\left(M_e\mid D_{C1}\right)P\left(M_p\right)}{P\left(M_e\right)};$$

$$C_2: P\left(D_{C2}\mid M\right)=\frac{P\left(M_e\mid D_{C2}\right)P\left(M_p\right)}{P\left(M_e\right)};$$

$$C_3: P\left(D_{C3}\mid M\right)=\frac{P\left(M_e\mid D_{C3}\right)P\left(M_p\right)}{P\left(M_e\right)}$$

2) The second layer entity performance

$$B_2: P\left(D_{B2}\mid M\right)=\frac{P\left(M_e\mid D_{B2}\right)P\left(M_p\right)}{P\left(M_e\right)}\times P\left(D_{B2}\mid D_{C1}\right)\times P\left(D_{B2}\mid D_{C3}\right);$$

$$B_3 : P(D_{B3} | M) = \frac{P(M_e | D_{B3}) P(M_p)}{P(M_e)} \times P(D_{B3} | D_{C2}).$$

3) Top layer entity performance

$$A_2 : P(D_{A2} | M) = \frac{P(M_e | D_{A2}) P(M_p)}{P(M_e)} \times P(D_{A2} | D_{B2}) \times P(D_{A2} | D_{B3}).$$

4) Overall system performance

$$P(D_s | M) = \frac{P(M_e | D_s) P(M_p)}{P(M_e, D_{A2})} \times P(D_s | D_{A2})$$

When the system has a variety of assembly options available, it is possible to calculate all kinds of situations and comprehensively select the optimal solution.

5. Examples and Results Analysis

We have simulated an actual enterprise e-commerce application system to carry out experiment on the above research results. The system structure is shown in Figure 3.

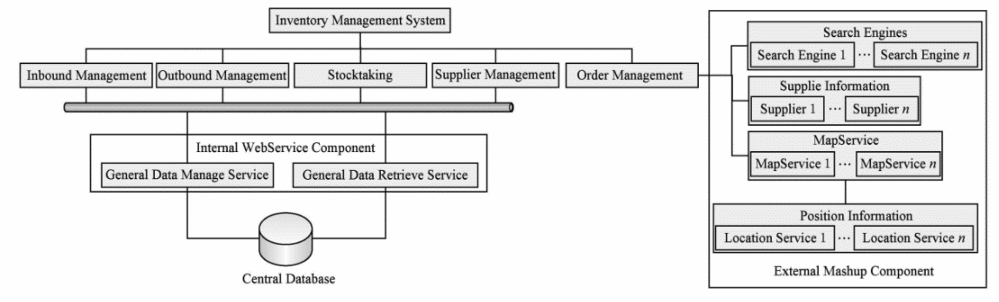

Figure 3 Inventory Management System Structure

The overall system can be divided into three parts: the business logic module, the system internal components and the external service components. In this paper, the internal components of the system adopt the form of the network services and interact with the business logic module through the standard WebService interface. The external service components adopt the combination of the mashup applications provided by the third party. The mashup application is an emerging web application that uses the Ajax technology to obtain new functions by integrating the data and functionality provided by the third party in the network, so as to construct the personalized Web applications.

5.1. System Structural analysis

The structural analysis is carried out on each module of the software system in a bottom-up manner so as to obtain the structural dependency diagram of the system. For the local modules, human and warehouse management, outbound management, inventory status inquiry, supplier management and so on in the system, as the same development environment is adopted and they are integrated for the application together, the interrelation is relatively close. In addition, there is no interface compatibility issue. Therefore, they can be treated as the same entity during the structural analysis. The internal components of the system can be adopted as the separate service entity. Since they are developed independently, they are fully compatible with the other modules, and therefore they do not involve the issue of structural dependency, but just exist as a part of the calculation of the system performance. Finally, the overall system structure dependency diagram can be obtained as shown in Figure 4 as the following.

In Figure 4, Goog1e map and Baidu map only support the SOAP protocol, while E-city only supports the REST protocol. Location service 1 and location service 2 only support the SOAP protocol, while location service 3 only supports the REST protocol. In this way, when the Goog1e map and Baidu map are used for the map service, the positioning service 1 and the positioning service 2 can be used as the basic services. However, when the E-city is used, the basic services can only use the positioning services 3. The subsequent work is to calculate the attributes of various aspects of services after the combination so that the optimal combination can be used.

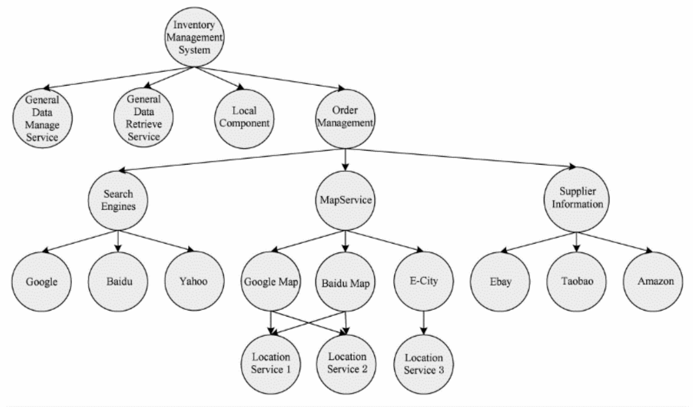

Figure 4 Structure Dependency Diagram of Inventory Management System

5.2. System Performance Calculation

A bottom-up approach is adopted to calculate the performance of the system layer by layer. Firstly, the nodes at the lowest layer are calculated. The performance of a single node is obtained by summarizing its prior index and making correction after analyzing the index. That is, the prior probability is calculated and corrected through the factual evidence. The entity location service 1 is taken as an example. And the obtained prior indexes are functionality 0.98, reliability 0.96, ease of use 0.96, efficiency 0.95, maintainability 0.92, portability 0.90, and reusability 0.94. Through the analysis of its development and management status, the factual evidence is obtained to correct the prior probability and establish a conditional probability table as shown in Table 1.

Table 1 shows the differences in the performance brought about by the different conditions of satisfaction to each condition in the software development, that is, the probability of the software performance modeling analysis under different conditions. The items with the value of 1 in the table indicates that the condition is met, and the items with the value of 0 indicate that the condition is not met. For example: The first entry of data indicates that the performance of the software can be guaranteed in the case of proper testing, standardized programming, detailed documentation, skilled employees and process management. This table summarizes various types of software development processes, which can also be gradually modified in accordance with the specific evolution process of the software in the software utilization. Due to the limited space, the table is not listed in full, but only a few representative data are listed.

Table 1 Performance Modeling Analysis Index Condition Probability Table

Quality test	Normal pogramming	Detailed document	Skilled stalf	Process management	dependability
1	1	1	1	1	1
1	1	1	0	1	1
0	0	0	1	1	0
1	1	1	1	0	0
1	1	1	1	0	1
0	1	1	0	1	0
1	0	0	1	1	0

The performance of the entity location service 1 is calculated in accordance with equation (3). And the formula is as the following:

$$P(D|M)=\frac{P(M_e|D)P(M_p)}{P(M_e)} \tag{5}$$

The left side of the equation is the node performance after the correction; and the right side of the equation $P(M_e)$ is the joint distribution probability that the analysis index meets the given condition; $P(M_e \quad D)$ stands for the joint distribution probability of each analysis index that meets the given condition under the condition of the software performance modeling and analysis; and $P(M_p)$ stands for the comprehensive performance value of each prior index. Since the prior indexes are independent of each other, $P(M_p)=\coprod_i P(M_{pi})$, in which M_{pi} stands for the six kinds of prior indexes listed above. Herein calculation is carried out in accordance with the analysis of the indexes that meet different conditions separately. And in the subsequent result analysis, the results of each calculation will be discussed in detail.

Case 1. The above five kinds of analysis indexes are met. In accordance with the analytical index conditional probability table, the joint probability of the five analytical indexes are calculated to obtain $P(M_e)=0.33$. Under the software performance modeling and analysis conditions, the joint probability of the five analytical indexes $P(M_e\ D)=0.24$ is met, and the data are introduced into the equation (5) to obtain the following:

$$P(D\,|\,M)=\{0.33\times(0.98\times0.96\times0.96\times0.95\times0.92\times0.90\times0.94)/0.24\}=0.918$$

Case 2. The above five kinds of analysis indexes meet the good testing, standard programming, detailed documentation and process management. In accordance with the analytical index conditional probability table, the joint probability of the five analytical indexes are calculated to obtain $P(M_e)=0.39$. Under the software performance modeling and analysis conditions, the joint probability of the five analytical indexes $P(M_e\ D)=0.31$ is met. And the data are introduced into the equation (5) to obtain the following:

$$P(D\,|\,M)=\{0.39\times(0.98\times0.96\times0.96\times0.95\times0.92\times0.90\times0.94)/0.31\}=0.84$$

Case 3. The above five kinds of analysis indexes meet good testing, standardized programming and skilled employees. In accordance with the analytical index conditional probability table, the joint probability of the five analytical indexes are calculated to obtain $P(M_e)=0.43$. Under the software performance modeling and analysis conditions, the joint probability of the five analytical indexes $P(M_e\ D)=0.37$ is met. And the data are introduced into the equation (5) to obtain the following:

$$P(D\,|\,M)=\{0.43\times(0.98\times0.96\times0.96\times0.95\times0.92\times0.90\times0.94)/0.37\}=0.776$$

Case 4. None of the above five kinds of analysis indexes are met. In accordance with the analytical index conditional probability table, the joint probability of the five analytical indexes are calculated to obtain $P(M_e)=0.09$. Under the software performance modeling and analysis conditions, the joint probability of the five analytical indexes $P(M_e\ D)=0.13$ is met. And the data are introduced into the equation (5) to obtain the following:

$$P(D\,|\,M)=\{0.09\times(0.98\times0.96\times0.96\times0.95\times0.92\times0.90\times0.94)/0.13\}=0.462$$

After completing the calculation of all the basic entities in accordance with the above method, calculation of the performance of the intermediate node is carried out. When calculating the performance of intermediate nodes, in addition to considering the entity performance of the node itself, it should be corrected through the performance of its base nodes.

Taking the entity Goog1e map as an example, the performance of the node is calculated to be 0.94, and the performance of its sub node is 0.918 and 0.928, respectively. The corrected entity performance is as the following

$$P(D\,|\,M)\times P(D\,|\,D')=0.94\times0.928=0.87$$

In which, *D'* stands for the relatively high performance value of the location service 1 and the location service 2 for its base node, that is, *P (D D')* stands for the performance of the current computation node when the base node performance is modeled and analyzed. The value of *P (D D')* before the system is put into operation can be set as a certain initial value, for example, *P (D')*. After the system is running for a period of time, the statistics of the evidence data can be carried out in accordance with the specific operating condition to conduct the correction.

In addition, when a node can use a number of base nodes at the same time, it usually adopts the redundant backup method to establish connections with multiple base nodes at the same time. When one node fails, it immediately switches to the standby node, which can greatly enhance the performance of the entire system. At this point, the performance of the node is corrected by using the probability that all base nodes fail as the following:

$$P(D|M)\times\left(1-\coprod_i\left(1-P(D|D_i)\right)\right)=0.94\times\left(1-(1-0.918)\times(1-0.928)\right)=0.934$$

In accordance with this method, the structural dependency diagram is calculated from bottom up and finally obtain the overall system performance of 0.909. When the selected entities are different, or when the analysis indexes that they meet are different, the results obtained from the calculation of the overall system performance are not the same.

5.3. Results Analysis

The above calculation process can be analyzed as the following: (1) In the process of calculating the performance of the single node, when the circumstances in which the nodes meet the analysis index are different, the probabilities of the joint distribution calculated in accordance with the conditional probability table are also different. More analysis indexes that the node meets indicate that the design, development and management process of the node entity is more compliant with the standard specification, the software quality level is higher, and the performance of the entity is also higher, which is in full compliance with the idea of the software quality assurance theory. And the calculation result has also reflected this point correctly. For the entities with exactly the same priori indexes, in accordance with the circumstances that the analysis indexes meet, the resulting performance after the correction is also different. That is to say, the entity with higher degree of satisfaction to the analysis index will have higher performance value after the correction. On the contrary, the entity with lower degree of satisfaction to the analysis index will also have lower performance value after the correction. (2) The effect of correcting the performance between nodes: The performance of the intermediate node is affected not only by the prior index of the node itself and the analytical index calculation, but also by the performance of its basis nodes. This has reflected the mutual relationship between the entities of the Internet ware under the practical conditions as well as the integrated performance of a group of collaborative entities. Through the method of performing redundancy backup for the nodes, the possibility of the entity suspending the service can be reduced, and the performance of the entity can be improved, which is in full accordance with the theoretical common sense for the design and development of the Internet ware system. (3) Objectivity of the final performance value: The final performance value integrates a number of evaluations on different aspects of the entity performance into a unique result and make the correction accordingly. In this way, it can clearly reflect the performance of the entity of the Internet ware. In addition, it does not rely solely on the data provided by the service provider but can be objectively analyzed by the user on the basis of the actual situation of the service entity.

6. Conclusions

In this paper, through the analysis of the structure of the Internet ware, a kind of analysis system for the internet ware performance modeling based on the fusion of process group component model is established. The model based on the fusion of the process component adopts the bottom-up, layer-by-level analysis and calculation method to analyze the various component entities of the Internet ware, the overall performance indexes of the system and multiple aspects of performance indexes and form a unified performance result. In addition, the objective data are used to make correction on the performance result. It can be seen from the previous experiments that the model put forward in this paper can calculate the performance of the Internet ware system and provide support for the user's choice of entity.

References

[1] Kong, J., White, C. A., Krylov, A. I., Sherrill, D., Adamson, R. D., & Furlani, T. R., et al. (2015). Q‐chem 2.0: a high‐performance ab initio electronic structure program package. Journal of Computational Chemistry, 21(16), 1532-1548.

[2] Hasan, F. A., Xiao, P., Singh, R. K., & Webley, P. A. (2013). Zeolite monoliths with hierarchical designed pore network structure: synthesis and performance. Chemical Engineering Journal, 223(3), 48-58.

[3] Nan, N., & Kumar, S. (2013). Joint effect of team structure and software architecture in open source software development. IEEE Transactions on Engineering Management, 60(3), 592-603.

[4] Czibula, G., & Czibula, I. G. (2014). Software systems performance improvement by intelligent data structures customization. Information Sciences, 274(8), 249-260.

[5] Müller, H. P., Gorges, M., Grön, G., Kassubek, J., Landwehrmeyer, G. B., & Süßmuth, S. D., et al. (2016). Motor network structure and function are associated with motor performance in huntington's disease. Journal of Neurology, 263(3), 539-549.

[6] Mu, J. (2014). Networking capability, network structure, and new product development performance. IEEE Transactions on Engineering Management, 61(4), 599-609.

[7] Bai, N., Xiang, K., Zhou, W., Lu, H., Zhao, X., & Chen, H. (2016). Lifepo4/carbon nanowires with 3d nano-network structure as potential high performance cathode for lithium ion batteries. Electrochimica Acta, 191, 23-28.

[8] Jiang, N., Shelley, J. D., Doyle, S., & Smith, R. (2014). Heat exchanger network retrofit with a fixed network structure. Applied Energy, 127(6), 25-33.

[9] Aldecoa, R., & Marín, I. (2014). Surpriseme: an integrated tool for network community structure characterization using surprise maximization. Bioinformatics, 30(7), 1041-1042.

[10] Yang, J., & Ma, J. (2016). A structure optimization framework for feed-forward neural networks using sparse representation. Knowledge-Based Systems, 109, 61-70.