A KPN based Model for Describing and Verifying the Interaction of Web Services

Li Bao[1], *Le Hao*[2]

1. School of Data Science and Application, Inner Mongolia University of Technology, Huhhot 010080, Inner Mongolia Autonomous Region, China;

2. Endocrine Department, Geriatric Research Center, Inner Mongolia People's Hospital, Huhhot 010080, Inner Mongolia Autonomous Region, China;

Abstract: Correct interaction between Web services is essential for successful Web service composition. This paper proposes a Web Service Interaction Model (IWSN) that aims to ensure correct interaction between Web services, improve the scalability of Web service composition, solve behavioral compatibility issues in the process of Web service interaction, and promote the application of service composition technology in related fields. The Kahn Process Network (KPN) supports parallel computing based on data streams and channels, and the proposed Web Service Interaction Model in this article is based on the KPN. The formal semantics of the IWSN model are based on process algebra Pi calculus, and the model's properties are discussed. Finally, an application case is used to demonstrate how the IWSN model can be applied to Web service composition and interaction.

Key Words: Web Service, Service Composition, Service Interaction, Process algebra

1. Introduction

As the modern economy has fully transitioned into modern service epoch, the modern service industry has emerged as a prominent driver for the sustainable development of the contemporary economic landscape. Web services are computing resources that are scalable and elastic, providing users with practical production-oriented needs. These resources are offered by cloud providers. However, it is important to note that a single web service has a limited capacity to provide sufficient computing resources. For instance, diabetic patients require information consultation on controlling postprandial blood sugar levels in order to slow the progression of the disease. Various factors such as insulin dosage, meal caloric intake, daily physical activity, and other common factors impact postprandial blood glucose. To predict and provide intervention for postprandial blood glucose, diabetic patients require various data. Stepping recording services, calorie intake recording services, and medical data analysis services represent some common software services used to collect relevant data. These services belong to different cloud platforms. However, it is clear that a single software service alone cannot meet the above requirements. A combination of blood sugar control consultation services, step recording services and data analysis services, is essential. These services together form a service composition that provides diabetic patients with information consultation on their blood sugar levels.

The technology of web service cross-cloud composition offers a novel solution to the aforementioned problems. This service involves dynamically adding or releasing the necessary resources as per the customers' requirements, merging web services from various providers, offering value-added services to users, and executing multifaceted functions. The primary goal of cross-cloud service portfolio is to proficiently select and coordinate services from various cloud platforms, closely integrate services across cloud platform limitations, and advance and innovate its functionalities. In doing so, these services can work in unison to provide users with more inclusive and better functions. The technology of cross-cloud service composition injects new momentum into the modern service industry.

The technology of web service composition (as referenced in) offers a means to connect all component services, presenting a fresh approach to resolving the aforementioned issues. A significant advantage of SOA is its ability to blend numerous interactive service components into composite services. Service designers can create (compose) business processes by merging existing service components to cater to business process necessities, eliminating or minimizing the need for expansion or modification.

The remaining sections of this paper are arranged in the following manner: Section 2 gives an insight into the relevant research work, Section 3 illustrates the Web service interaction model, Section 4 provides the behavioral semantics of the Web service interaction model, Section 5 presents a case study, and the final section concludes the paper and discusses future works.

2 Related Works

Web Service Composition is the process of combining component services into services with larger granularity based on business processes and rules. Service composition methods can be roughly divided into two categories: service composition driven by business process and service composition based on real-time task solution: one is service composition driven by business process, and the other is service composition based on real-time task solution. The business process-driven service composition method is evolved from the workflow-based approach, whose application has been relatively mature and there are not many new achievements in recent years. Whereas, the service composition method based on real-time task solving is the current research hotspot. Several representative new methods are introduced below.

Researchers investigated issues related to service reconfiguration and supply and demand convergence in cloud manufacturing in . They proposed an autonomous service composition and optimization selection method for cloud manufacturing. In order to improve service selection accuracy, a decision-making method based on fuzzy sets and volatility analysis was proposed. Meanwhile, presented a multi-level workflow orchestration framework comprising of five stages to solve the issue of finding and selecting suitable candidate services using semantic Web service discovery mechanisms. The framework uses functional attributes to describe the behavior of candidate services and can adapt to dynamic changes in user goals. Document proposed a hybrid Web service composition method focused on generating semantic input-output-based service compositions with optimal end-to-end QoS and minimal participation of services. To facilitate efficient service composition, Reference provides a visual REST service modeling framework that includes a visual service composition modeling tool and a service development platform. In addition, the heuristic test case generation method was used in literature to improve the quality of Web service composition test cases and identify the path that could cause the greatest probability of service composition failure. Also, in , the authors combine reinforcement learning with game theory and come up with a multi-agent reinforcement learning algorithm to address large-scale adaptive service composition problems. Finally, proposed a hybrid genetic algorithm that integrates genetic algorithms and fruit fly optimization algorithms with the aim of solving NP-difficult problems in the process of Web service composition.

3 Interactive Web Service Network: A Service Interaction Model

3.1 Kahn Process Network

The Kahn Process Network (KPN) is a data flow-based computing model that has advantages in achieving concurrent communication and execution of processes on a distributed structure. This feature makes the KPN model particularly suitable for modeling Web service interaction behavior. In the KPN model, concurrent processes are represented by nodes, and ordered data sequences are represented by arcs (also known as tokens). Processes communicate with each other through first-in, first-out channels, which are represented in KPN using data token queues. Starting a read operation on a FIFO channel requires the presence of at least one token in the communication channel. Because the size of the FIFO channel is infinite, write operations are non-blocking. The KPN model adopts asynchronous communication, and the output result of KPN calculation is independent of its execution order, i.e., the order in which concurrent processes are executed has no impact on the calculation result.

3.2 Web Service Interactive Network

In the IWSN model, Web services are represented as computing processes in KPN, while Web service composition is represented as a concurrent, autonomous network of processes. Communication between processes is achieved through first-in, first-out channels. Each process in the network exercises autonomous control over the message sequence, and its operations are executed concurrently with those of other processes. The behavior of a single process (service) is modeled based on its operations, while trace is used to model the interaction between processes. The structure of the IWSN model is illustrated in Figure 3.

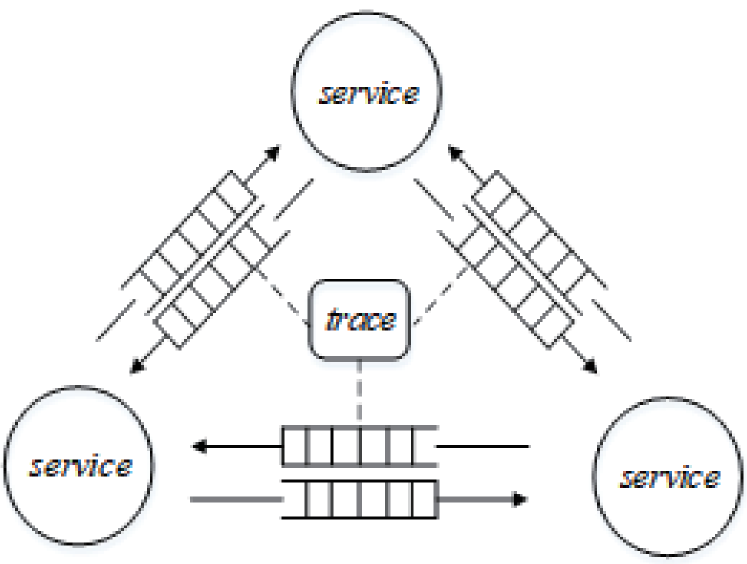


Fig.1 The Structure of IWSN

3.3 Formal Definition of Interactive Web Service Network

Since operations are the fundamental unit of Web service functionality, IWSN models Web service invocation as the process of invoking operations, represented by the set Operation. Symbol Processes are utilized in IWSN to model a series of operations executed sequentially by services, where an element in the set Process is a linear concatenation of operations. If the output message of one operation op1 is the input message of another operation op2, there is a corresponding relationship between op1 and op2. A binary relationship, Ro, is introduced in this case. The symbol T is used in IWSN to denote the interaction record of a process (Web service).

Definition 1 (Interactive Web Service Network). An Interactive Web Service Network is a Six-tuple <Service, Process, Operation, T, Message, Ro, >, where:

●Service is the set of Web Services;

●Process denotes the set of Processes;

●Operation denotes the set of Operations;

T is the set of operationsand partial ordering relations, in which the partial ordering relations define the execution order and structure of operations, and this set is used to document the execution process of the operations;

●Message denotes the messages that comes into and leaves from the processes;

●Ro Operation × Operation is a binary relation over the set Operation.

4 Behavior Semantics of Interactive Web Service Network

This section introduces the behavioral semantics of IWSN, which is based on process algebra Pi-calculus and its axiomatic operational semantics. Behavior semantics consists of two parts, semantic transformation domain and transformation function. Firstly, Pi-calculus is utilized as a semantic transformation domain, and then a transformation function definition and transformation algorithm are given to convert the semantics of the IWSN model into Pi-calculus expressions

4.1 Syntax of Pi-calculus

Let symbol N denotes a countable infinite set, in Pi-calculus processes are denoted by the elements P, Q, R(capital letters) … of N, actions performed by processes are denoted by the elements u, v, w, x, y, z …(Lowercase letters) of N. Expressions of Pi-calculus are constructed using the following gramma:

$$P::=0|\sum P_i|cx.P|(P_1|...|Pn)|_{[x=y]}P \quad (1)$$

A null process is denoted by number 0, which indicates that no action will be executed; symbol ∑ denotes nondeterministic choice, for example, expression R + P denotes execute one of the processes R or P, but not both; process cx.P denotes the sequential execution of processes, where P is a process, c is a channel name, x is a action name. The process cx.P read and perform an action from channel c and then proceed to execute process P; prcess P1|...|Pn denotes the parallel execution of processes P1, P2, ..., Pn ; expression [x=y]P behaves in same way as process P when the name of channel x equals the name of channel name y, otherwise process [x=y]P is a null process.

Table 1. An Semantic Transform Algorithm for IWSN

Algorithm 1. IWSN Translation

```
INPUT: IWSN Instance
OUTPUT: The translated Pi-calculus expressions
While ( |P| < O ) do
  for each j from 1 to O
    for each k from 1 to O
      if j != k
        Process[j, k] = Operaton (j, k)
      end if
    end for
  end for
 for each i from 1 to N
   action[i] = message[i]
     Pi = Pi ∪ {P};
     If Sig then {Sig-count2=Sig-count1;O1=O2}
   } end of for-loop
}end of while-loop
If(|P|=|S|) then Pi=Pi ∪ P;
Return establish_Pi
```

5 CASE STUDY

5.1 The Blood Glucose Information Consulting Service for Diabetes Patients

This section demonstrates the effectiveness and feasibility of the method proposed in this paper with regards to implementing the medical service architecture. We achieve this by using the IWSN model to model and verify the case of Web service composition, specifically the diabetes patient blood glucose information consulting service. Since our focus is on service interaction behavior, we will not be modeling the technical details of other aspects involved in service composition at this time. The blood glucose information consulting service for diabetes patients consists of three component services: the patient medical record database service, data analysis service, and blood glucose management consulting service. The business process is briefly described as follows: When receiving client service questions for blood glucose management consulting services, data is obtained through the patient medical record database service and sent to the data analysis service. After obtaining relevant data analysis results from the data analysis service, the customer service questions are answered based on the corresponding information. Figure 2 illustrates the structure of blood glucose information consultation service for diabetes patients. Pi-calculus and messages are utilized to formally model the component services' interaction. The explanation of the meaning of some messages is detailed in Table 2.

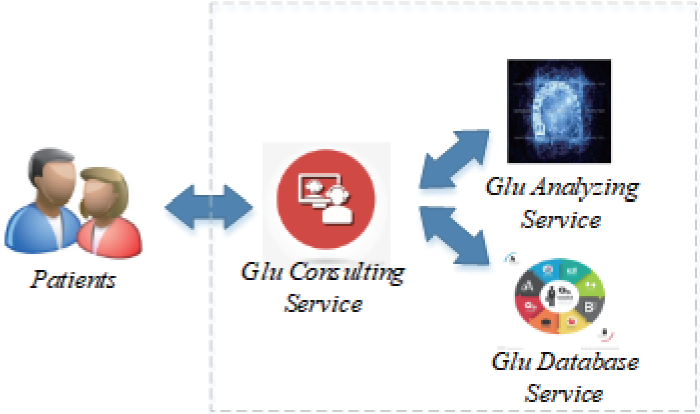


Fig.2 Blood Glucose Information Consulting Service for Diabetes Patients

Table 2. Messages and Their Meanings

Messages	Explicit Meanings
Glu_Info_Consult:	Message sent from *Patients Service* to the *Glu_Consulting_Service*, meaning *Patients Service* requests to receive the information on how to better reduce patients' postprandial blood glucose.
Req_Analyze	Message sent from the *Glu_Consulting_Service* to the *Glu_Analyzing_Service*, meaning the *Glu_Consulting_Service* needs the analysis result of the patients' blood glucose data that the *Glu_Analyzing_Service* will provides.
Req_Glu_Data	Message sent from the *Glu_Analyzing_Service* to the *Glu_Database_Service*, *meaning* the *Glu_Analyzing_Service* needs the patient's the recent blood glucose data and past medical history that the *Glu_Database_Service* will provide.
Glu_Data_Send	Message sent from the *Glu_Database_Service* to the *Glu_Analyzing_Service*, meaning the *Glu_Database_Service* transmits the related medical data requested by the Analyzing service.
Analysis_Glu_result	Message sent from the *Glu_Analyzing_Service* to the *Glu_Consulting_Service*. The *Glu_Analyzing_Service* transmits the analysis results of medical data to the *Glu_Consulting_Service*.
Glu_Info_Reply	Message sent from the *Glu_Consulting_Service* to the *Patientrs Service*. The *Glu_Consulting_Service* provide the *Patientrs Service* with information on how to control patients' blood glucose based on the results of data analysis by *Glu_Analyzing_Service*.

Based on the previously defined model semantics, the blood Glucose Information Consulting Service can be described in Pi-calculus as follows:

Patients_Consulting _Service = Glu_Info_Consult. !Req_Analyze. Analysis_result.Glu_Info_Reply

Glu_Analyzing_Service = Req_Glu_Data_Analyze. !Req_Glu_Data. Glu_Data_Send. !Glu_Analysis_result

Glu_Database_Service = Req_Glu_Data. !Glu_Data_Send

Composite_Service = (Patients_Consulting_Service | Glu_Analyzing_Service | Glu_Database_Service) / { Req_Analyze, Analysis_Glu_result, Glu_Data_Send}

5.2 Verification of the Services Interactive Behavior

We conduct reachability analysis for Blood Glucose Information Consulting Services for diabetes patients. Through the reachability analysis, we may find the incorrect system designs, such as deadlocks and the omission of key system functions. Computation Tree Logic (CTL) is used to verify whether the system has expected properties. CTL is a type of branching temporal logic, for which the temporal operators of formulas are composed of path operators and temporal operators.

Definition 2 (CTL Model). A CTL model is a triplet M=(S, R, Label), where S is a non-empty state set; RS×S is a transition relation of states; Label: S→2AP is a state labeling function.

Definition 3 (CTL Grammar). There are two types of path operators: A represents "applicable to all paths" and E represents "at least one or more paths exist". The temporal operators consist of X (time) operator, F (Future) operator, G (Global) operator, and U (Unit) operator. CTL expressions are constructed with the following grammar:

$$\varnothing::p|-p|p1\vee p2|EXp|E[p1\cup p2]|A[p1\cup p2] \quad (2)$$

where: the symbols and are logical operators, and the symbols EX, E, A, U are temporal operators. The logical operators True, False, , and are re deduced from and . The temporal operators EFp, AFp, EGp, AGp, and AXp are obtained by using the above operators and the following derivation rules: { EFp = E[truep], AFp = A[truep], EGp = AFp, AGp = EFp, AXp = EXp }

Expression (2) is a logical formula that defines a state that the system should never reach, of which the symbol should_reply is used to represent this state. A detailed explanation of the state should_reply is as follows:

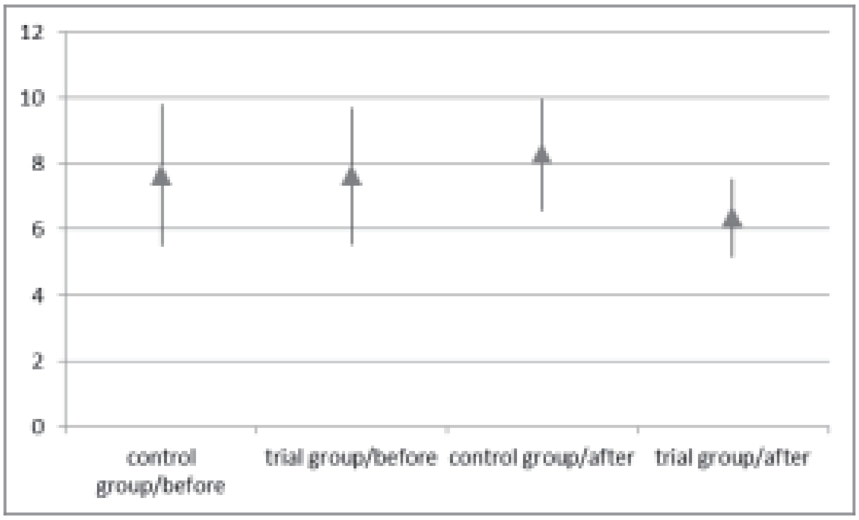
the value of Should _ Reply for the system is true if after the patient service has sent the request message, the Patient service will definitely receive a response message from the Glu Consulting Service after a limited number of internal execution steps.

prop Should_Reply=

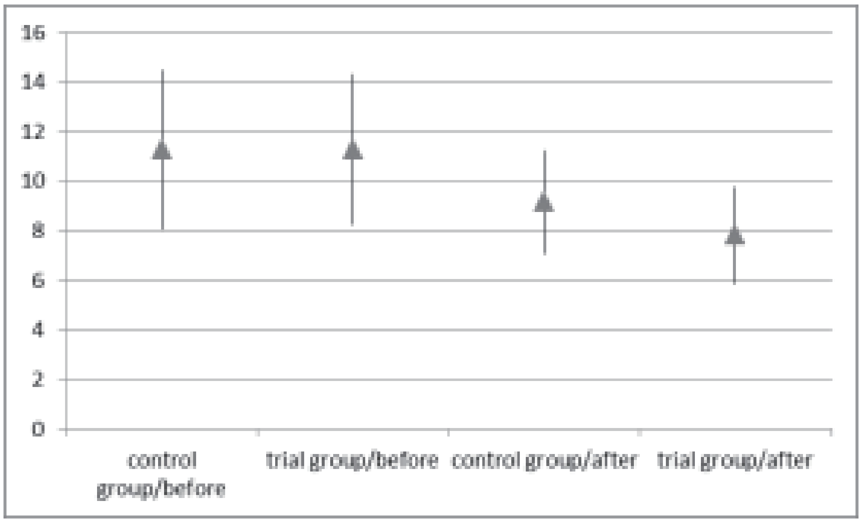
*min*X=(<*'Glu_Info_Reply*>tt∨<*'Glu_Info_Reply*>tt)∨<->x (3)

5.3 Verification of the Model's Effectiveness

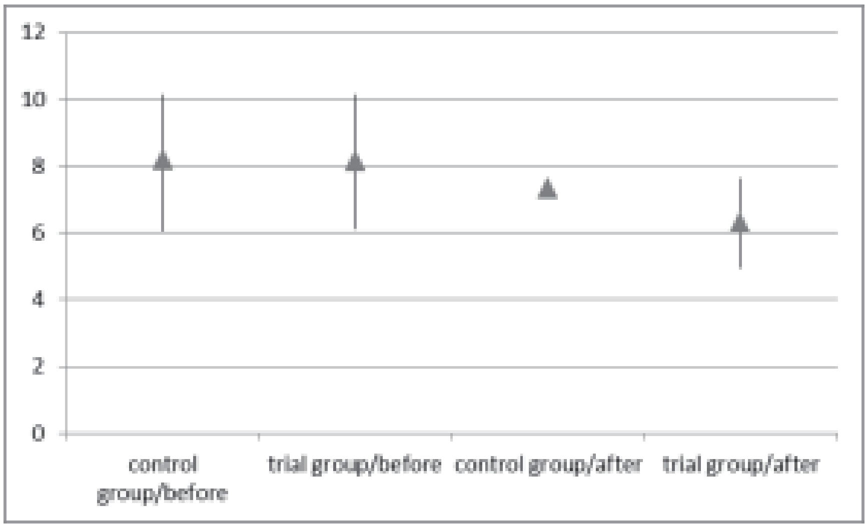
To verify the effectiveness of blood glucose information consulting services in aiding diabetic patients to control their blood sugar levels, we selected 66 type II diabetes patients admitted to the Elderly Disease Research Center of Inner Mongolia Autonomous Region People's Hospital from December 2022 to February 2023. The study included 33 patients in the control group and 33 patients in the experimental group who used the online-deployed blood glucose information consulting service. Data in Figure 3 shows that the experimental group patients who used the blood glucose information consulting service had significantly better improvements in fasting blood glucose (FBG), 2-hour postprandial blood glucose (2hPBG), and glycated hemoglobin (HbA1c) levels than the control group patients who received regular diabetes health education.



(A)Comparisons of FBG between Two Groups



(B)Comparisons of 2hPBG between Two Groups



(C)Comparisons of HBAlc between Two Groups

Fig. 3

6. Conclusion and Future Works

This paper proposes a Web service interaction model (IWSN) based on Kahn process network, which aims to solve the problem of correct interaction of Web services and at the same time ensure the behavior compatibility during the interaction process of Web services. On the basis of introducing the definition of Pi-calculus, the formal semantics of IWSN model is given, and the semantic transformation algorithm of IWSN model is given. In the case of vehicle maintenance and repair service, the IWSN model is used to describe the business process, and on the basis of giving a compatible definition of service behavior, it is verified whether the vehicle maintenance and repair service has the expected attributes by using branch time logic Computational Tree Logic (CTL). The experimental results of the case show that our proposed method is feasible and applicable.

To sum up, the KPN-based Web service interaction network has the characteristics of concurrent communication mechanism, composability, and executability, and is suitable for modeling service interaction in the process of Web service composition. Our future work is to realize the prototype system of Web service interaction network, and lay the foundation for the market promotion of related technologies.

Acknowledgement:

This work was supported by the Inner Mongolia Natural Science Foundation under the grant No. 2022MS06004, 2020MS06025; Inner Mongolia Key Technology Research Program Project(2019GG271); Central Guidance Local Science and Technology Development Fund Project(0222ZY0169); Fundamental Research Fund Project of Inner Mongolia Universities(JY20220227, JY20220249).

Reference:

[1] Meng Wei. Research on the Development Path and Countermeasures of an Open Service Economy [J] Economic Forum, 2017, 4: 124-125

[2] Varghese, Blesson, Buyya, et al. Next generation cloud computing: New trends and research directions[J]. Future Generations Computer Systems Fgcs, 2018, 79(3): 849-861.

[3] Canal C., Villari M.. Advances in Service-Oriented and Cloud Computing[M]. Springer International Publishing, 2015.

[4] PengWei Wang, ZhiJun Ding, ChangJun Jiang. Constrain-Aware Approach to Web Service Composition[J]. IEEE Transactions on Systems Man & Cybernetics Systems, 2014, 44(6): 770-784.

[5] A. L. Lemos, F. Daniel, B. Benatallah. Web service composition: a survey of techniques and tools[J]. ACM Computing Surveys (CSUR), 2015, 48(3): 1-41.

[6] Li C, Guan J, Liu T, et al. An autonomy-oriented method for service composition and optimal selection in cloud manufacturing[J]. International Journal of Advanced Manufacturing Technology, 2018, 96(3): 1-22.

[7] Arul U, Prakash S. Toward automatic web service composition based on multilevel workflow orchestration and semantic web service discovery[J]. International Journal of Business Information Systems, 2020, 34(1): 128-156.

[8] P. Rodriguez-Mier, M. Mucientes, M. Lama. Hybrid Optimization Algorithm for Large-Scale QoS-Aware Service Composition[J]. IEEE Transactions on Service Computing, 2017, 10(4): 547-559.

[9] Sun Zuhan, Li Ying, Lo Chi Ling, etc. . Design and implementation of a visual REST service composition framework[J]. Small microcomputer systems, 2017,38(1) : 10-14.

[10] Song Y, Gong Y. Web service composition on IoT reliability test based on cross entropy[J]. Computational Intelligence, 2020, 36(4): 1650-1662.

[11] H. Wang, X. Chen, Q. Wu, et al. Integrating reinforcement learning with multi-agent techniques for adaptive service composition[J]. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 2017, 12(2): 1-42.

[12] F. Seghir, A. Khababa. A hybrid approach using generic and fruit fly optimization algorithms for QoS-aware cloud service composition[J]. Journal of

Intelligent Manufacturing, 2017: 1-20.

[13] G. Kahn. The semantics of a simple language for parallel programming[J]. Information processing, 1974, 74: 471-475.

[14] Milner R, Communication and Mobile Systems: The Pi-Calculus[M], Cambridge University Press, 1999.

[15] E.M.Clarke, E.A.Emerson, and A.P.Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications[J]. ACM Transaction on ProgrammingLanguages and Systems,1986, 8(2): 244-263.