A π-calculus-based Complaint Model for a User-interactive Question Answering System

Yancheng Li, Hua Duan, Qingtian Zeng[+], Faming Lu
College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, 266590, China,
[+]corresponding author, E-mail: qtzeng@163.com

***Abstract*—User-interactive question answering systems are catching more and more attention, however users' complaints arise within it from such problems as unfair compensation. This paper focuses on a π-calculus-based formal model for the complaint process of a user-interactive question answering system. The formal model is verified by employing MWB tools so as to demonstrate the deadlock-free and other properties. The research results are of special significance to the comprehensive formal analysis of the user-interactive question answering systems.**

Keywords: User-interactive Question Answering System, π-calculus, Complaint Model, MWB tools

I. Background

With the ever accelerated development of the internet in terms of technology and applications, the information acquired from the search engines has become extremely diverse. What is more, due to the inadequacies made by search engines, such as excessive data in results need to be answered accurately, this problem is greatly exaggerated. It is worthy to note that the appearance of the question and answering system (QA)[1] has made up for these shortcomings effectively.

In the QA system, interacting with the users can improve the instantaneity and efficiency of the search results. In recent years, multi-user interactive QA has become a hot research topic. [2] defined semantic patterns for questions and answers in multi-user interactive QA systems. [3] applied a multi-user interactive QA system for personalized E-Learning.[4] proposed an approach to obtain the knowledge requirement of users using an interaction QA system.[5] studied the user reputation model for a user-interactive QA system. On the basis of our previous work[1-5], this paper focuses on the complaint model of the multi-user interactive QA system.

Figure 1 shows the framework of the multi-user interactive QA system[2-3]. The basic process of the system interacting with the user is described as follows: Users ask questions in the system and set certain rewards for their questions. Then, other users can answer these questions. Based on these questions the quizmaster would make an evaluation to distinguish the best answer and judge corresponding reward. When the user chooses the best answer from all the answers, he or she may not honor to have the corresponding reward or other respondents may think the best answer which the user chose is unreasonable. Once it happens, it will cause complaints from answerers to the quizmaster. As it is closely related to interests of each individual user, a correct complaints model is a must for the normal functioning of the system.

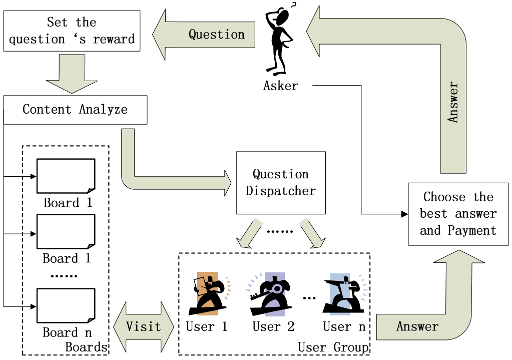


Figure 1. Framework for a User-interactive QA System.

II. Complaint Process Framework for a User-interactive Question Answering System

Users can complain through the complaint function the system provides. Figure 2 shows the framework of the complaining process and the paid interactive QA system consists of four user roles, including Replier (complainant), Questioner, ComplaintAdmin (complaints administrator) and arbiter.

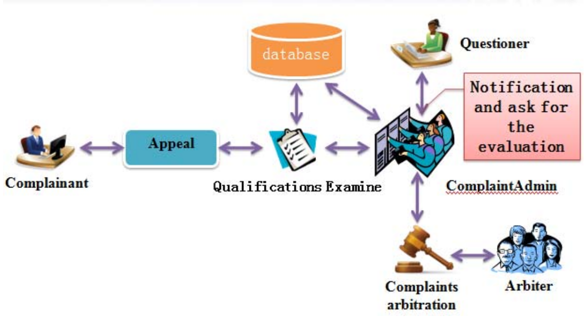


Figure 2. Complaint process framework for a User-interactive QA System[2-3].

A complete complaint process includes the following four steps:

Step1. Users submit complaints to the complaints administrator;

Step2. Complaints administrator audits the submitted complaints.

Step3. Complaint administrator asks the quizmaster for the data information which is related to the complaint, including questions, the best answer at present, complaint answers, the evaluation and interpretation from the quizmaster.

Step4. Complaints administrator deals with complaints and the easily judged complaints can be directly handled by the system pacificator. For the controversial complaints, the related data is sent to the complaint forum and set the limit time of the complaints arbitration, then vote arbitration by arbitrators.

If the complaint is successful, the quizmaster will pay the relevant remuneration for the complainant and then deduct the reputation value of the quizmaster; but if the complaint failed, the reputation value of the complainant would be deducted, and send the feedback information to the quizmaster and complainant.

III. FORMAL MODEL

π-calculus[6-7] is a mobile, interactive, basic and concurrent computing model unit for name and processes. Process interacted with each other by passing the name, while the system is constituted of a set of processes. By accepting the name, one process can dynamically interact with other processes, the process then creates a new channel, therefore the π-calculus can be used to describe the structure and change concurrent systems. π-calculus is suitable to describe the dynamic structure of the software system, and we can use the correlative analysis method of the π-calculus to analysis and verify the model so as to eliminate human errors in the modeling process. Since π-calculus is a formal theory foundation, in this paper, it is applied to model the paid interactive QA system complaints.

A. *The basic concept of multiple π-calculus*

Multiple π-calculus is an extension of the standard π-calculus, which allows one port to pass the message name-based vector. For example, output $\overline{a}<y_1,y_2...y_n>.P$, input $a<x_1,x_2...x_n>.Q$, if you pass only one name, it can be written as $\overline{a}y_1.P$. Although multiple π-calculus is adopted as the modeling tool, the name is with a lowercase letter and process with an uppercase letter.

Let *N* be an infinite set of names, the name on the set of names with lowercase letters such as *a*, *b*; and process identifier uppercase letters represented with *A*, *B*, *C*, etc; The process expressions uppercase letters represented by *P*, *Q*, *R*, etc. This paper describes only three forms for the definition of the process expressions:

1) Sum expression. $\sum_{i\in I} P_i = P_1 + P_2 + ... + P_n$, where *P* can only be chosen to perform one of the process P_i ;

2) Prefix expression. $\overline{y}x.P$,$y(x).P$,$\tau.P$, where $\overline{y}x$ is called negative prefix, $\overline{y}$ represents the output interface of the process and $\overline{y}x.P$ represents outputting the name x in the interface $\overline{y}$. The implementation process $P.y(x)$ is called positive prefix, where y represents the input interface of the process, $y(x).P$ represents input any name z in the interface y, then implementation process $P\{z/x\}$, among them, z/x represents replacing x with z. y and $\overline{y}$ are the dual name, and the link connection between them is called the channel y, τ is called dumb prefix, $\tau.P$ represents performing dumb action τ and then perform *P*. In general, τ is used to represent the internal action of the process which is not visible outside.

3) Parallel expressions. $P|Q$ represents the parallel execution of processes *P* and *Q*. Processes *P* and *Q* can be executed independently, but can also interact through the dual shared name.

B. *Complaint Model for a User-interactive Question Answering System*

In this section, a complaint model for a user-interactive QA system based on π-calculus is presented, which includes four characters, including the Questioner, Replier (complainant), ComplaintAdmin and Arbiter.

1) Repliers send complaints request message. If the complaint was rejected, then end; if the complaint is accepted, the replier will receive the final results of the complaints. According to π-calculus, repliers' behavior is described as follows:

$$Replier(appeal,reject,success,failed,wr1,wr2,wr3,wr4)= \overline{wr1}(appeal).(wr2(reject).0+wr3(success)+wr4(failed))$$

2) After receiving the requested data messages (*reqdata*), the questioner collates data(the data message which includes the raised issues, the best answer of the complaints, the answers of the complaint and evaluation and interpretation of the questions)and send it, and finally obtained the complaint results. π-calculus-based questioners' behavior is described as follows:

$$Questioner(reqdata,data,success,failed,wq1,wq2,wq3,wq4)=wq1(reqdata).\overline{wq2}(data).(wq3(success)+wq4(failed))$$

3) ComplaintAdmins receive complaints messages. According to the rules, if the complaint is admissible, send a reject message; if accepted, transmitting a data request message, and then receiving the data message. The ComplaintAdmins determine whether the complaint can be dealt immediately, if permitted, judge the complaint and sent the results of the complaints; if not permitted, send the data message back to the complaint forum. Then the results of the complaints is sent back to the ComplaintAdmin. According to π-calculus, the ComplaintAdmin behavior is described as follows:

$$ComplaintAdmin(appeal,reject,reqdata,data,success,failed,wr1,wr2,wr3,wr4,wq1,wq2,wq3,wq4,fw1,fw2,fw3)=wr1(appeal).(\overline{wr2}(reject)+\overline{wq1}(reqdata).wq2(data).(\overline{fw1}(data).(fw2(failed).(\overline{wq4}(failed)\ |\ \overline{wr4}(failed))+fw3(success).(\overline{wq3}(success)\ |\ \overline{wr3}(success)))+t.(fw2(failed).(\overline{wq4}(failed)|\overline{wr4}(failed))+fw3(success).(\overline{wq3}(success)|\overline{wr3}(success)))))$$

4) The arbiter receives the data message, and returns the results of the complaints after discussion. π-calculus -based arbiter behavior is described as follows:

$$Arbiter(data,failed,success,fw1,fw2,fw3)=fw1(data).(\overline{fw2}(failed)+\overline{fw3}(success))$$

5) Finally, we can get the π-calculus-based description of the entire complaint model:

Complaint=Questioner|Replier|ComplaintAdmin|Arbiter

The specific meaning of the model is: First of all, the Replier sends complaints request messages to the ComplaintAdmin, and then it reviews the complaints request. If the review failed, then it would send a reject message; otherwise, it would send a request data message to the Questioner. The Questioner who receives the request message, will return the data message to the ComplaintAdmin. If the ComplaintAdmin determines that the complaint can be dealt immediately, then directly determine the results and related processing, after that, returned the complaint results to the Questioner and the Replier; Otherwise, the data message is sent to the complaint forum, and the Arbiter determines the outcome of a complaint, then returns to the ComplaintAdmin, and it will return the complaint results to the Questioner and the Replier after the correlation processing.

6) By integrating the Questioner, Replier, ComplaintAdmin and Arbiter, we can get the overall model which is shown in Figure 3.

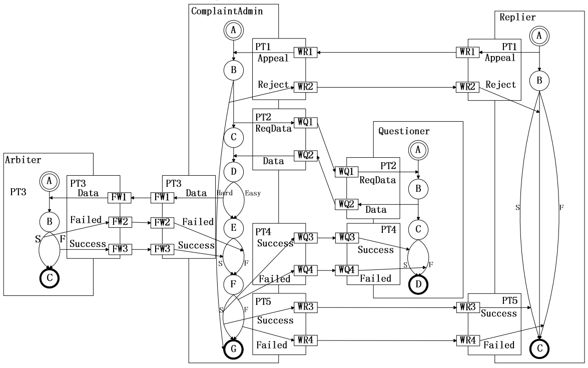


Figure 3. Complaints interaction model for a User-interactive QA System.

IV. MODEL VALIDATION

MWB is an automatic verification tool developed for π-calculus. Based on MWB, we can analyze and verify mobile concurrent systems which are described by multiple π-calculus or CCS, etc. In this paper, MWB based on the New Jersey SML language is adopted to verify the model and judge the correctness. Using MWB tool for algebraic expression parsing can find some basic errors of the process, such as the lack of synchronization, type errors and incomplete behavior, etc. It can also take advantage of the reasoning capabilities of the tool to exclude some of the most common mistakes. With deadlocks command, we can check whether there is process deadlock situation or not and have an overview of the process behavior and track deduction system behavior.

A. *The symbol conversion of the π-calculus to the MWB*

In order to facilitate the use of the MWB tools to verify the model, the π-calculus symbols are conversed into MWB symbols, and the conversion rules are shown in Table 1. At the same time, the four main types of the complaint model roles are made of agents.

The MWB symbolic description of the Questioner's behavior is:

Questioner(reqdata,data,success,failed,wq1,wq2,wq3,wq4)=wq1(reqdata).'wq2<data>.(wq3(success).0+wq4 (failed).0)

The MWB symbolic description of the Replier's behavior is:

Replier(*appeal*,*reject*,*success*,*failed*,*wr*1,*wr*2,*wr*3,*wr*4)=' *wr*1<*appeal*>.(*wr*2(*reject*).0+*wr*3(*success*).0+*wr*4(*failed*).0)

The MWB symbolic description of the ComplaintAdmin's behavior is:

ComplaintAdmin(*appeal*,*reject*,*reqdata*,*data*,*success*,*failed*, *wr*1,*wr*2,*wr*3,*wr*4,*wq*1,*wq*2,*wq*3,*wq*4,*fw*1,*fw*2,*fw*3)=*wr*1(*appeal*).('*wr*2<*reject*>.0+'*wq*1<*reqdata*>.*wq*2(*data*).('*fw*1<*data*>.(*fw*2(*failed*).('*wq*4<*failed*>.0|'*wr*4<*failed*>.0)+*fw*3(*success*).('*wq*3<*success*>.0|'*wr*3<*success*>.0))+*t*.(*fw*2(*failed*).('*wq*4<*failed*>.0|'*wr*4<*failed*>.0)+*fw*3(*success*).('*wq*3<*success*>.0|'*wr*3<*success*>.0))))

The MWB symbolic description of the Arbiter's behavior is:

Arbiter(*data*,*failed*,*success*,*fw*1,*fw*2,*fw*3)=*fw*1(*data*).('*fw*2<*failed*>.0+'*fw*3<*success*>.0)

The MWB symbolic description of the complaint model is:

Complaint=Questioner|Replier|ComplaintAdmin|Arbiter

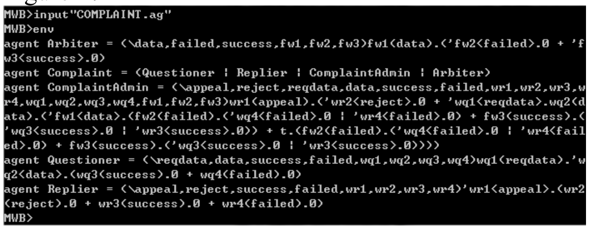
B. *A MWB-tools-based model validation*

TABLE I. THE CORRESPONDING TABLE OF THE SYMBOL CONVERSION OF THE Π-CALCULUS TO THE MWB

The Symbol of The π Calculus	The Symbol of The MWB	Explanation
v	^	Restrict
0	0	Null operation
$\bar{a}$	‘a	Output operation
τ	t	Internal operation
a(nlist)	a(nlist)	Input prefix
a<nlist>	‘a<nlist>	Output prefix

In this section, the verification for accuracy of the complaint model in the user-interactive QA system is introduced based on MWB tools. The specific operations are described as TABLE I.

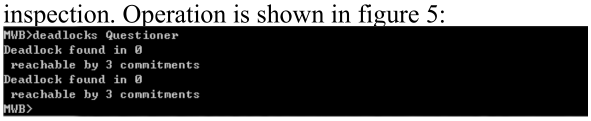
Open the MWB tool, and input the command: “COMPLAINT.ag”, and through the “env” command all defined agents are displayed. The operation is shown in Figure 4:



```
MWB>input"COMPLAINT.ag"
MWB>env
agent Arbiter = (\data,failed,success,fw1,fw2,fw3)fw1(data).('fw2<failed>.0 + 'f
w3<success>.0)
agent Complaint = (Questioner | Replier | ComplaintAdmin | Arbiter)
agent ComplaintAdmin = (\appeal,reject,reqdata,data,success,failed,wr1,wr2,wr3,w
r4,wq1,wq2,wq3,wq4,fw1,fw2,fw3)wr1(appeal).('wr2<reject>.0 + 'wq1<reqdata>.wq2(d
ata).('fw1<data>.(fw2(failed).('wq4<failed>.0 | 'wr4<failed>.0) + fw3(success).(
'wq3<success>.0 | 'wr3<success>.0)) + t.(fw2(failed).('wq4<failed>.0 | 'wr4<fail
ed>.0) + fw3(success).('wq3<success>.0 | 'wr3<success>.0))))
agent Questioner = (\reqdata,data,success,failed,wq1,wq2,wq3,wq4)wq1(reqdata).'w
q2<data>.(wq3(success).0 + wq4(failed).0)
agent Replier = (\appeal,reject,success,failed,wr1,wr2,wr3,wr4)'wr1<appeal>.(wr2
(reject).0 + wr3(success).0 + wr4(failed).0)
MWB>
```

Figure 4. Displays all defined agent.

Questioner agents, for example, by checking the deadlocks command to the death of the agent lock for inspection. Operation is shown in figure 5:

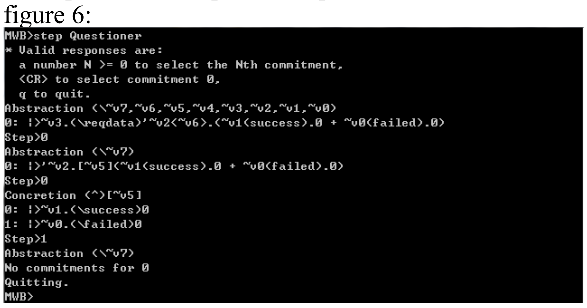


```
MWB>deadlocks Questioner
Deadlock found in 0
 reachable by 3 commitments
Deadlock found in 0
 reachable by 3 commitments
MWB>
```

Figure 5. Deadlock for the agent.

Each agent’s deadlock has been examined in the complaint model, and the results prove that each agent doesn’t exist deadlock.

Questioner agents, for example, by step command execute the step validation. Use step command simulation its implementation process, operation results are shown in figure 6:



```
MWB>step Questioner
* Valid responses are:
  a number N >= 0 to select the Nth commitment,
  <CR> to select commitment 0,
  q to quit.
Abstraction (\~v7,~v6,~v5,~v4,~v3,~v2,~v1,~v0)
0: |>~v3.(\reqdata)'~v2<~v6>.(~v1(success).0 + ~v0(failed).0)
Step>0
Abstraction (\~v7)
0: |>'~v2.[~v5](~v1(success).0 + ~v0(failed).0)
Step>0
Concretion (^)[~v5]
0: |>~v1.(\success)0
1: |>~v0.(\failed)0
Step>1
Abstraction (\~v7)
No commitments for 0
Quitting.
MWB>
```

Figure 6. Executive steps to verify for agent.

Each agent in the complaint model is verified according to the above steps, and the results show that each agent in the complaint model is able to execute correctly.

V. CONCLUSION

This paper presents the complaint model of a user-interactive QA system using π -calculus. This model is verified by employing MWB tool which is an automatic verification tool for π -calculus. Based on the π -calculus, system's integrity, individuality, and the interaction among users in a User-interactive QA system is described. Moreover, the correctness of the system is verified. On the basis of this work, it is still necessary to refine the credibility of evaluation in the complaint model and the arbitration process in the complaint forum. We can also try to verify and analyze the model with other formal models as well as comparing the advantages and disadvantages between them.

ACKNOWLEDGMENT

This paper is supported partly by the NSFC (61170079 and 61202152); Sci. & Tech. Development Fund of Shandong Province of China (2010GSF10811); Specialized Research Fund for the Doctoral Program of Higher Education of China (20103718110007) and Special Fund for Fast Sharing of Science Paper in Net Era by CSTD, Sci. & Tech. Development Fund of Qingdao(10-3-3-32-nsh), Excellent Young Scientist Foundation of Shandong Province of China (BS2010DX009); China Postdoctoral Science Foundation (2011M501155); and Natural Science Foundation for Distinguished Young Scholars of Shandong and SDUST (2010KYJQ101).

REFERENCES

[1] Li Liu, Qingtian Zeng: An Overview of Automatic Question and Answering System, Journal of Shandong University of Science and Technology,26 (4),73-76, October 2007.

[2] Tianyong Hao, Dawei Hu, Liu Wenyin, Qingtian Zeng, Semantic Patterns for User-Interactive Question Answering, Concurrency and Computation: Practice and Experience, v 20, n 7, May, 2008, p 783-799.

[3] Dawei Hu, Wei Chen, Qingtian Zeng, Tianyong Hao, Feng Min, Liu Wenyin, Using a User-interactive QA System for Personalized e-Learning, The International Journal of Distance Education Technologies, 6(3), 1-22, July-September 2008.

[4] Qingtian Zeng, Zhongying Zhao, Yongquan Liang. Course Ontology-based User's Knowledge Requirement Acquisition from Behaviors within E-Learning Systems, Computers & Education, Volume 53 , Issue 3 (November 2009) 809–818.

[5] Chen, Wei; Qingtian Zeng; Wenyin, Liu; Hao, Tianyong. A user reputation model for a user-interactive question answering system, Concurrency Computation Practice and Experience, v 19, n 15, October, 2007, p 2091-2103.

[6] http://www.cnham.com/unix/html/20/2005/0922/html

[7] Milner R, Parrow J, Walker D. A Calculus of Mobile Processes. Journal of Information and Computation, 1992; (1)

[8] Clarke E M, Wing JM. Formal methods: state of the art and future directions [J]. ACM Computing Surveys(S0360-0300), 1996, 28(4):626-643.

[9] Milner R. The Polyadic π-calculus: a Tutorial. Springer- verlag,1993