IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 9, November 2014.
www.ijiset.com
ISSN 2348 – 7968

Online Service and Information Providing

Mrs. Angayarkanni. A[1], Thamizhselvan. S[2], Sundar. S[3]

[1]Assistant professor,epartment of Master of Computer Application,
Sri Manakula vinayagar Engineering College Madagadipet, Puducherry, India.

[2] PG Student, Department of Master of Computer Application,
Sri Manakula vinayagar Engineering College Madagadipet, Puducherry, India.

[3] PG Student, Department of Master of Computer Application,
Sri Manakula vinayagar Engineering College Madagadipet, Puducherry, India.

Abstract:

The concept processing the client request based on trustworthy services implemented in an online web commerce environment. The services can be selected by using heuristic algorithm which reduces the fake services and saves time consumption. The user can post the request on his own desire and it will be processed securely.

Web services:

Web service is a method of communication that allows two software systems to exchange this data over the internet. The software system that requests data is called a service requester, whereas the software system that would process the request and provide the data is called a service provider.

Kinds of services:

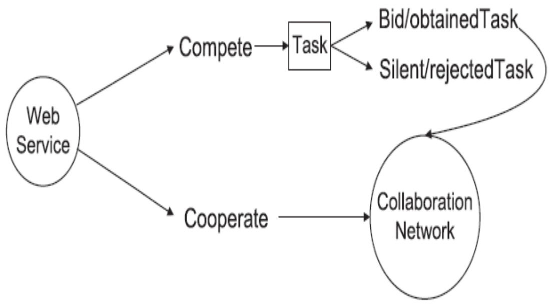
Services that reside inside a processing which aggregates a number of functionally similar or complementary services as a group. Within the same community, each service might have a network consisting of some other services that might get involved in a cooperative work (e.g. composition and substitution). As services are also competing, particularly when they provide similar functionalities, each one of them aims to maximize its individual income by adopting a given strategy.

(1) Master Service is the manager and representative of the community of services. Among other functionalities, the master agent is responsible for allocating the tasks to services within the community. After the task being accomplished, and based on the delivered quality of service, the master rewards or penalizes the associated service agent by updating its reputation. The master is equipped with a task allocation mechanism aiming to increase service user's satisfaction
(2) Users generate tasks with specified QoS. In our proposed system, tasks are continuously being generated and user satisfaction is abstracted since we focus on services' interactive strategies.

Service Selection and Providing:

Analyzing services acting behaviors in such a context in terms of deciding which strategy to choose is an open and challenging problem. In web most of the services are fake, untrustworthy and malicious. These services will produce enormous problems to the user while processing. More precisely, we define and analyze the request first and process by filtering the all untrustworthy services. And our main contribution in this paper is to address this problem using the competitive and cooperative strategies
(1) Competitive strategy where the service claims that it can accomplish a task, and

therefore can take the responsibility over the service consumer satisfaction.
(2) Cooperative strategy where the service does not take the responsibility to accomplish the task and only cooperates.



After these strategies requests are processed by choosing trustworthy services on the basis of the rating and ranking system. If the rank will be high then it will be possible to post at top of the service posting.

Existing System

In this existing system service selection strategy is to be addressed this problem using the game-theoretic best response technique. It is simply an alternative perspective with which to view the process of problem-solving. It is a tool, which, like all others, is best used by those who reflect on their own practice as a mechanism for improvement. In which best response is the strategy that produces the most favorable outcome for a player, taking other players'. The accuracy of the game based theory is not efficient. Through this mechanism services are selected and processed

Proposed System

We propose a Framework based on T-HEU (heuristic algorithm) to resume the ongoing process and to monitor the ongoing services. It starts with filtering all untrustworthy component services, because, the concept of trust in social relationships is the basis of human interactions and business transactions.
* Trust solves the problem of interaction, information sharing, and collaboration between two entities without any prior mutually shared experiences.
* In a service-oriented environment, trust is a key factor in determining the reliability of Web services.
* A number of efficient trust evaluation models have been proposed recently to solve the service selection problem. Thus, trust provides us with a promising starting point in the solution of Web service processing.

Previous approaches have only considered trust or reputation as a global QoS requirement proposed by users, but these approaches are incapable of avoiding the selection of malicious services in the process of service composition. That is, most of the selected component services have good reputations, but a few services with poor reputation that may have also been selected, so the aggregated value of the global quality or trust can still satisfy the users' requirements, even though untrustworthy services have been included in the optimal solution. Malicious services may subvert the system by delivering fake files or viruses. So the T-HEU judges if the trustworthiness of all the candidate services in each task is lower than the threshold means these services are not considered in the selection process.

Implementation:

As an optimization problem, the computation time for QoS-aware service composition grows exponentially with size. This is not reasonably suitable for a composition system that requires making real time decisions. Heuristic algorithms are useful in finding a feasible solution within a reasonable time.

Concepts

The heuristic algorithm T-HEU developed for QoS-aware service composition is based on the concepts enumerated below. It starts with filtering all untrustworthy component services, because malicious services may subvert the system by delivering fake files or viruses. At first, the T-HEU judges if the trustworthiness of all the candidate component services in each task is lower than the threshold; if yes, these component services are not considered in the

$$a_{ij} = \frac{Q_{ij} \cdot C}{|C|}$$

Selection process where aij represents the aggregate resource consumption for a component service. Each candidate component service for each task represents a point in two-dimensional space. The utility (F(sij)) represents the y co-ordinate and the transformed QoS value (aij) represents the x co-ordinate.

After establishing the qualified candidate component services for each task, a solution is sought from each task, where the service with the smallest utility is selected. If this is not feasible, T-HEU tries an iterative procedure by replacing one selected service with another to find a feasible solution. A feasible solution is an optimal solution if it maximizes the objective function.

Feasible Solution:

Let Sk be an instantiation of S, in which a component service is selected for task Ti if qk(Sk) ≤ Ck,¥ qk .€Q, Ck .€ C, (i.e. all the global QoS constraints are satisfied), then a vector R = (s1j; s2j; . . . ; snj) is a feasible solution, where

s1j, s2j, . . . snj are the selected services in task Ti. _ T-HEU then uses iterative improvement of the solution through the exchange of selected services. In each exchange, the status of the two services (one selected and the other not-selected) in a group are swapped.

In lines 19 to 24, the algorithm finds a service with higher utility, that satisfies three conditions:

(1) it decreases the feasibility factor of attribute α,

(2) it does not increase the feasibility factor of any previously infeasible attribute, and

(3) it does not make any previously feasible attribute infeasible.

Algorithm T – HUE ():

```
1:  FOR each task T_i
2:   FOR each service s_ij in T_i
3:    IF tr(s_ij) < θ
4:      remove s_ij from task T_i
5:    ENDIF
6:   ENDFOR
7:  ENDFOR
8:  FOR each task T_i
9:   FOR each service s_ij in T_i
10:    transform Q_ij to single dimension by applying Eqs. (12)
11:    T_i := construct_convex_hull()
12:   ENDFOR
13:  ENDFOR
14:  ℜ:= pick of services having the smallest utility for each task
15:  Q_α(ℜ) = Σ_{i=1}^{n} q_{αℜ_i}
16:  IF f_α ⩽ 1
17:    goto line #33.
18:  ENDIF
19:  ELSE consider a service s_ij that maximizes Δa_ij
20:    let ℜ' = ℜ|s_ij such that Q_α(ℜ') < Q_α(ℜ)
21:    IF k ≠ α and C_k(ℜ) ⩽ Q_k(ℜ) THEN Q_k(ℜ') < C_k(ℜ')
22:    ENDIF
23:    IF k ≠ α and Q_k(ℜ) > C_k(ℜ) THEN Q_k(ℜ') < Q_k(ℜ)
24:    ENDIF
25:   IF s_ij is found, set ℜ' = ℜ|s_ij and goto line #16
26:   ENDIF
27:   End the algorithm with "No solution found"
28:  find feasible upgrade service s_ij that maximizes Δa_ij
29:  IF Δa_ij > 0, set ℜ = (ℜ|s_(ij)) and goto line #28
30:  ELSE find feasible upgrade service s_(ij) that maximizes Δp_ij
31:    IF Δp_ij > 0, set ℜ = (ℜ|s_(ij)) and goto line #28
32:    ENDIF
33:  ENDIF
34:  find an upgrade service s_(ij) that maximizes Δp'_ij, set ℜ'_c = (ℜ_c|s_(ij))
35:  find an downgrade service s_(ij) that maximizes Δp''_ij
36:  IF Δp''_ij < 0, set ℜ = (ℜ'_c|s_(ij))
37:    IF ℜ is feasible, goto step line #28.
38:    ENDIF
39:    ELSE goto line #34
40:  ENDIF
41:  RETURN ℜ
```

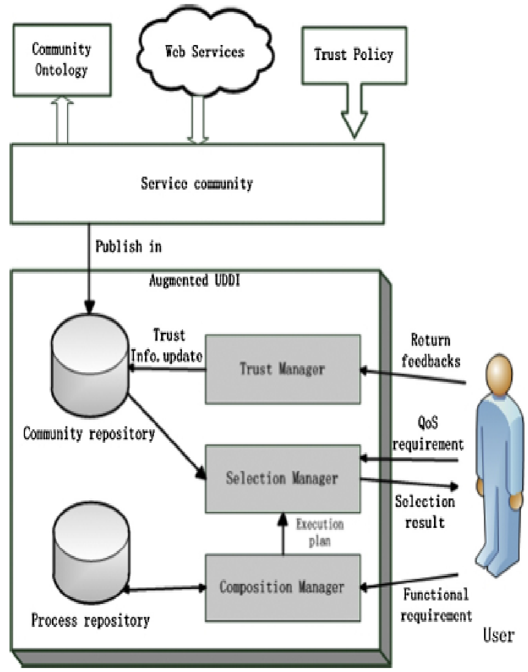
ISSN 2348 – 7968

Therefore, another important factor for the success of heuristics is to ensure that the selected services are trustworthy. We consider an optimal solution as unsuccessful if the trustworthiness of any component service used in the solution is below 0.5; otherwise, it is considered successful. We execute each heuristic 100 times and check if the solution is successful each time (if yes the number of successful solutions increases by 1). The success rate (state) is obtained using the following metric

$$\text{State} = \frac{\text{\#the number of successful times}}{\text{\#the total number of selection times}}$$

Experimental Results:

In this section, we provide an empirical analysis over the theoretical results regarding the characteristics of intelligent service agents hosted in different process of services. In the implemented system, we simulate the behaviors of service consumers as request generators, service agents as service providers, and master agent. In this section, the objective is to investigate the effectiveness of the proposed strategic system on selection of services' overall budget and also the average quality and quantity of tasks performed by the community of services, which directly affects user satisfaction. To verify these objectives, we study the overall performance of the community hosting the reasoning-empowered services compared to the ones hosting stochastic and purely competitive services. By processing services, we mean services that adopt at each moment competitive or cooperative strategies equally, but in a random way.



Conclusion

Due to the great diversity of services and of service providers, the qualities and levels of trustworthiness of available services vary enormously. A service, as a component service, may have a very low trust value even though the whole composite service it belongs to retains a high trust value. Therefore, developing a usable system to support the service users that distinguishes a trustworthy service from an untrustworthy one is critical. In this paper, we propose an efficient and trustworthy approach that achieves these goals.

(1) Competitive strategy where the service claims that it can accomplish a task, and therefore can take the responsibility over the service consumer satisfaction;
(2) Cooperative strategy where the service does not take the responsibility to accomplish the task and only cooperates with competitive peers.
Our proposed model advances the state-of-the-art in cooperative systems by enabling

ISSN 2348 – 7968

ranking and ratings to effectively choose their interacting strategies that lead to optimal outcomes. As future work, we plan to emphasize the service consumer role in the proposed model to obtain more accurate results when consumers post their service satisfaction feedback. Adding users feedback as a system parameter in our algorithm can improve the decision making process, by considering, not only the system and services perspective, but also the users perspective.

References

[1]Web services composition: A decade's overview
[2]Quan Z. Sheng a,⇑, Xiaoqiang Qiao b,*, Athanasios V. Vasilakos c, Claudia Szabo a, Scott Bourne a, Xiaofei Xu
[3]An efficient and reliable approach for quality-of-service-aware service composition
[4]Jun Li b,a, Xiao-Lin Zheng a,⇑, Song-Tao Chen a, William-Wei Song c, De-ren Chen
[5]A Fast Heuristic Algorithm for the Composite Web Service Selection
[6]Rong Wang, Chi-Hung Chi, Jianming Deng Reliable Web Services Selection Using a Heuristic Algorithm Hongjie Shen Key Lab. of Embedded Syst. & Service Comput., Tongji Univ., Shanghai, China
Zhijun Ding ; Hongzhong Chen ,Tao Yu, Yue Zhang, and Kwei-Jay Lin, "Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints", ACM Transactions on the Web. Vol. 1, 2007, pp: 1-23.
San-Yih Hwang, Ee-Peng Lim, Chien-Hsiang Lee, et al "Dynamic Web Service Selection for Reliable Web Service Composition", IEEE TRANSACTIONS ON SERVICES COMPUTING. VOL. 1, 2008.