Zhu and Jin *Human-centric Computing and Information Sciences* 2012, **2**:15
http://www.hcis-journal.com/content/2/1/15

Human-centric Computing and Information Sciences
a SpringerOpen Journal

RESEARCH Open Access

An adaptively emerging mechanism for context-aware service selections regulated by feedback distributions

Yishui Zhu and Qun Jin*

* Correspondence: jin@waseda.jp
Graduate School of Human Sciences, Waseda University, 2-579-15 Mikajima, Tokorozawa-shi, Saitama, Japan

Abstract

Background: In the cloud computing environments, numerous ambient services may be created speedily and provided to a variety of users. In such a situation, people may be annoyed by how to make a proper and optimal selection quickly and economically.

Methods: In this study, we propose an Adaptively Emerging Mechanism (AEM) to reduce this selection burden with an interdisciplinary approach. AEM is applied and integrated into the Flowable Service Model (FSM), which has been proposed and developed in our previous study. We consider the user's feedback information is a pivotal factor for AEM, which contains the user's satisfaction degree after using the services. At the same time, we assume that these factors, such as the service cost, matching result precision, responding time, personal and social context information, etc., are essential parts of the optimizing process for the selection of ambient services.

Results and Conclusion: By analyzing the result of AEM simulation, we reveal that AEM can (1) substantially improve the selection process for LOW feedback users; (2) bring no negative effect on the selection process for MEDIUM or HIGH feedback users; and (3) enhance the rationality for services selection.

Keywords: Adaptively emerging mechanism, Cloud computing, Flowable service model, Service selection, User satisfaction

Introduction

Cloud computing brought great advantages in terms of its cost control, reliability, elasticity, etc. Services, as the elements in cloud computing, attracted more attentions in recent years. According to [1], the cloud computing's information processing capacities are quickly growing at clearly exponential rates. Consequently, the numerous ambient services can be highly expected. In the meanwhile, more and more people, including researchers, enterprise users, and general users, must acquire new capabilities to build and master services [2]. By this means, we think it becomes necessary to support people on how to make a proper selection of ambient services quickly, and how to make optimal selection of ambient services economically.

As we know, it is not enough to utilize and assess cloud computing just from the perspective of businesses/enterprises profitability or technical evolution. We have proposed a new framework of Flowable Service Model (FSM) in our previous work, and tried to gain a solid understanding of a novel concept from a human-centric view in this compelling new trend. Under the framework, users do not need to know which company or organization provides the services they are having, since the basic concept of FSM is to synthesize and mash-up numerous different services that are being run by different companies or organizations or other people. In this study, we newly propose and develop an Adaptively Emerging Mechanism for the selection of ambient services to be applied and integrated into this model.

The pivotal factor in AEM is the feedback of user satisfaction. We assume that the satisfaction information can be captured and obtained by some means, such as context-aware sensors in the process of users' responses. The user, who is very particular about the exploited services, often rejects the result which is provided by FSM. We consider this kind of particular user is in the LOW feedback users group. We take the other three user groups as MEDIUM, HIGH, RANDOM by their distribution of feedback values.

To arrange these similar but different numerous services, we set five dimensions for grouping services:

- service cost,
- matching result precision,
- service responding time,
- personal context information, and
- social context information.

The first three dimensions are relevant to the service provider. The last two dimensions depend on the user and the using situation. The dimensions for grouping services are not limited into these five ones. Further consideration of all-inclusive dimensions shall be more effective for grouping services.

The rest of this paper is organized as follows. Section 2 surveys related work. Section 3 describes the basic concept of our proposed FSM and three approaches to capture user's needs. In Section 4, we discuss AEM that includes Emerging Resource, Feedback Distribution and Selection. In Section 5, we analyze the results of AEM simulation through our evaluation measurement. Finally, in the last section, we summarize the contributions of our present work, and discuss future research plan.

Related work

Services selection

In [3], they introduced the Quality of Service (QoS)-based web service selection systems, for which inexperienced end users are not the focal point of the design. Most of the systems assume that users could formulate their QoS requirements easily and accurately by using the provided query languages. To emphasize more on the user-centered design of the service selection system, their study considered a more expressive and flexible way for non-expert users to define their QoS queries, together with the user support on formulating queries and understanding services in the registry. Their work greatly enhanced selection model.

We share the same concern stated in [4], that is, a key issue in service computing is selecting service providers with the best user desired quality. As multiple service providers may compete to offer the same functionality with different quality of service, the existing service selection approaches mostly rely on computing a predefined objective function. When multiple quality criteria are considered, users are required to express their preference over different quality attributes as numeric weights. They developed the so-called service skyline, and a set of service skyline computation techniques that return a set of most interested service providers. These providers are non-dominant in all user interested quality attributes. Our method is different from this study. We import the user's feedback value as an important factor to regulate the output service flow.

[5] discussed a human-centric integrated approach for web information search and sharing by incorporating the important user-centric elements, namely a user's individual context and 'social' factor, which are realized with collaborative contributions and co-evaluations, into web information search.

The reputation attribute of QoS is investigated and considered in [6]. It is very important for users to obtain reliable services in service selection. Their reputation measure has three phases (i.e., feedback checking, feedback adjustment, and malicious feedback detection) to enhance the accuracy. In this study, the feedback ratings are adjusted with different user feedback preferences by calculating the feedback similarity. And malicious feedback ratings are detected by adopting cumulative sum method. Their user survey focuses on feedback ratings between different users. In our study, we consider the distribution of feedback as an important factor.

[7] tried to deal with the dynamic web service selection problem. It aims to invoke web services at runtime so as to successfully orchestrate the services. They observed that both the composite and constituent services often constrain the sequences of invoking their operations. As a result, they assigned each state of execution an aggregated reliability to measure the probability that the given state will lead to a successful execution in the context where each web service may fail with a certain probability. In our study, to orchestrate a composite web service, we propose two strategies to select web services that are likely to successfully complete the execution of a given sequence of operations.

Service-oriented systems invoke a number of available web services. Users need to specify different preferences and constraints, and service selection can be performed dynamically at runtime. [8] introduces a new modeling approach to the web service selection problem that is particularly effective for large processes and when QoS constraints are severe. This model has three features: (1) the web service selection problem is formalized as a mixed integer linear programming problem, (2) the loops peeling is adopted in the optimization, and (3) the constraints posed by stateful web services are considered.

In [9], the authors proposed a methodology to rank the relevant services for a given request, by introducing objective measures based on dominance relationships defined among the services. And they investigated the methods for clustering the relevant services in a way that reveals and reflects the different trade-offs between the matched parameters.

Services composition

In [10], the authors mentioned an important topic – services composition. It is capable to recursively construct composite services, which are provided by different organizations and offer diverse functionalities. However, the selection for each user's needs was left as a big challenge. They addressed the issue of selecting and composing services not only according to their functional requirements but also to their transactional properties and QoS characteristics. Their selection algorithm that satisfies the user's preferences is expressed as the weights based on the QoS criteria and as the risk levels that define semantically the transactional requirements.

Another research group presented a fully decentralized service composition framework called SpiderNet in [11]. It provides the statistical multi-constrained QoS assurances and load balancing for service composition. One more research has been done based on the study of relationships of SOs' states and services in [12]. The researchers provide an algorithm to compose SOs' services, which can avoid generating the identified unqualified composite services.

Miscellaneous selection approaches

In the research work presented in [13], the discovery of web service is improved by social networks. When the web services are provided in or through social networks, they can identify their peers, (1) with which they'd like to work; (2) those that can replace them in the case of failure; (3) those that compete against them for selection. In the six steps for social web services development, they decomposed the profile into five categories: preconditions, inputs, outputs, effects, and QoS. And they established the degree of similarity (DS) which is split into three clusters between the web services. The three clusters are weak-similarity cluster, average-similarity cluster, and strong-similarity cluster. We refer to this social management mechanism as grouping services.

For data clustering, the clustering stability methods on model selection techniques have been investigated in [14]. They focus on the behavior of clustering stability using the k-means clustering. They gave an exact characterization of the distribution to which suitably scaled measures of instability converge have been defined, based on a sample drawn from any distribution in a satisfying mild regularity conditions. They showed that the clustering stability does not 'break down' even for arbitrarily large samples, at least for the k-means framework. And they identified the factors which eventually determine the behavior of clustering stability. This research helps us to make a choice of appropriate services grouping and service selection.

Work given in [15] is about prototype selection and [16] is concerning about view selection. Prototype selection is an important task for classifiers since through this selection process the time for classification or training could be reduced. The authors proposed a new fast prototype selection method for large datasets, based on clustering, which selects border prototypes and some interior prototypes in [15]. It reported that the experimental results showed the performance of the method and compared accuracy and runtimes against other prototype selection methods. They automated a process by evaluating the quality of a view, captured by every single camera, for which a human producer manually selects the best view in [16]. They regard human actions as three-dimensional shapes induced by their silhouettes in the space-time volume. Resting on the features which are evaluated based on the features of the space-time shape, two view quality approaches have been proposed. Their experiments showed that the proposed view selections could provide intuitive results which match common conventions. In our study, we also consider human factors to regulate a selection process.

In [17], the authors present an efficient diversity preserving selection (DPS) technique for multi-objective evolutionary algorithms (MEAs). It aims to preserve the diversity of non-dominated solutions to the problems with scaled objectives. Their core mechanism selects a group (of individuals) that is statistically furthest from the worst group, instead of just concentrating on the best individuals, as in the truncation selection. Their experiments demonstrated that DPS significantly improved the diversity of non-dominated solutions for badly-scaling problems, while at the same time it exhibits an acceptable proximity performance.

In [18], the service-oriented software engineering (SOSE) methodologies have been proposed and practiced. Service-oriented computing can be expected to effectively deliver software services in a dynamic environment. Most of SOSE methodologies share common features but are presented for different purposes, ranging from project management to system modernization, and from business analysis to technical solutions development. It is very difficult for a company to decide which methodology would fit best for its specific needs. [18] introduced a feature analysis approach and devised a framework for comparing the existing SOA methodologies.

Both of the research works given by [19] and [20] are about web searching. [19] introduced approaches used by web search tools to interact with users. They examined interactive interfaces that use textual queries, tag-focused navigation, hyperlink navigation, visual features, etc., with respect to different kinds of information resources. The implementation of a visual-interface based on the concepts of query token network and the WordNet ontology has been described. The work presented in [20] assumed that the top ranked pages returned are relevant to the user's query. They made the search process more convenient for users. They found the most important synonyms and hypernyms for the terms of the user query, utilizing Japanese WordNet. And by combining the aforementioned terms together, a new expanded query is then submitted to the search engine. This web searching process is easier for beginners.

Context-aware flowable service model

In our previous study, we have proposed and developed a new framework of Flowable Service Model (FSM), which aimed to provide a flow of services that is served to users seamlessly in the cloud computing environments. In this section, firstly, we briefly overview FSM, and then discuss how to deal with context information and capture user needs under the framework of FSM.

Flowable service model

A flowable service is a logical stream that organizes and provides circumjacent services in such a way that they are perceived by individuals as those naturally embedded in their surrounding environments. A flow of service is a metaphor for a subconsciously controlled navigation that guides the user through fulfillment of a flowable service process that fits the user's context and situation and runs smoothly with unbroken continuity in an unobtrusive and supportive way. An FSM puts a large emphasis on knowing the users for the purpose of intuitively providing the required services and, thus, increasing the level of satisfaction of the user (For more details, refer to [21-23]).

To introduce the model, we first show its two important characteristics: flowability and constitutional similarity. Then, we discuss its fundamental components.

Flow ability is a characteristic of a service that effectively combines diverse services to maximally satisfy a user's needs and provides the perception, to the user, of using one service through all of the user's daily activities. This characteristic ensures that the user is provided with the services he/she needs regardless of the activity that the user is engaged in. Furthermore, it ensures hiding all of the connections of service discovery and combination, thus introducing a flow of services that is interpreted by the user as one integral service. Hence, although the flow ability must be realized within certain technology solutions, this property is tightly related to how a user interprets, and perceives a set of services—therefore, a close consideration of the user is essential to realizing it.

Another characteristic of the FSM is constitutional similarity. A constitutionally similar object is exactly or approximately similar to a part of itself, in its constitution. There are three constituents in the proposed model: service, user and mediator. These three elements compose one mesh through mediators, and the whole system is composed of a number of such meshes. No matter how many meshes it includes, the picture of the whole system does not change. The mesh consists of User, Mediator and Service. Therefore, we can say it is "roughly" constitutionally similar. In addition, because the proposed model conducts close consideration of a user, the resources that a service provides are not limited to the hardware and machine-provided resources, and can include human-provided service resources as well.

Context-aware

FSM is context-aware. In this study, we classified contexts into three categories:

Human context is about human self. There are physiological contexts and mental contexts. The physiological contexts, such as heart beat, body temperature, blood pressure and so on, can be checked by simple instruments, and the result values represent whether a user's body (i.e., health) is in good or poor condition. The mental contexts, for instance, pleasure, anger, sorrow, joy, etc., can reflect a user's innermost status. It is difficult to capture them by instruments directly. But the mental status can be illustrated by the emotion expression on one's face or the gesture or some body language. We think the expression on face or gesture – the action of hand or finger or the movement of the body, e.g., nodding, can be tracked by chips and transformed into signals to be processed. Both the physiological contexts and mental contexts are important elements for FSM based systems.

Nature context includes the WHEN (time context) and WHERE (space context) information. Timestamp is a format of absolute time that records the trigger time of an event or the end point of a change. The relative time is used as a stopwatch. Space contexts, such as humidity, temperature, location obtained by a GPS, or some physical information for a certain space, are also important. For a vehicle, if it is moving, the velocity is one of the space contexts.

Cultural context represents the group attributes of human relationship information. Group contexts, such as one's occupation and organization information, may have some effect on the communication between a user and the people around him/her. For example, a visually-impaired person communicates in his/her own way, while the general people cannot understand the meaning well unless they learn the communication methods or body signs of the visually-impaired people. For a bigger human group, Social contexts could not be ignored. The nationality, tradition, belief, etc. are a few examples in this category. As an automatic ordering service in a restaurant, the menu provided to muslin in Islam world is different from the general menu.

Capturing user needs

User needs are important input for our model. And we think when a user can not accurately express their needs, or do not realize how to express her/his own needs, it does not represent she/he has no needs at that time. We consider three approaches to capture a user's needs: Clear requirement, Ambiguous requirement, and Implicit requirement.

Clear requirement It can be delivered by voice and/or characters. And methods such as semantic technology may support interaction with others in the system.

Ambiguous requirement It is a kind of those inexpressible by voice or characters, but the context-awareness technology can translate that inexpressible information into a captured representation.

Implicit requirement It may be obtained from a user's logs and the similar users' requirement, which are in one group. It is depended on the log model and the group user model.

Adaptively emerging mechanism

Basic idea

When we search for information or services, the search engine lists a long result set. Artificial checks are needed to filter the entries in the result set one by one. But mostly, the filtered information has some certain similarities, which can be processed automatically. In this study, we newly propose and develop adaptively Emerging Mechanism (AEM), which aims to help users save their time and cost in a certain degree.

The basic idea of AEM is to dynamically group the ambient services based on a variety of available context information, and thus adaptively select the candidate components that constitute the current flowable service based on the feedback information of user satisfaction and the cost of a service. AEM works in the service selection process, which matches the query with the available resources and adaptively adjusts the result based on the feedback information. The main process is the same as a usual selection method which only focuses on matching the query conditions with the available resources. In addition, AEM adds a complement process which can dynamically adjust the result by the user's satisfaction feedback.

Emerging resources

There are many useful resources that can be used and included in the provision of services. Users get satisfied by a service mashing-up or synthesizing these resources. In order to find a best or better selection for every user, we utilize the brain working model presented in [24], and extend it to build the proposed mediator's model.

From the perspective of users, Context Capturer [21] emulates the Human Information Processing (HIP) for consciousness as shown in [24]. Based on this, a user's needs are inferred and provided to the mediator. Note that here we do not try to give the complete solution for needs inference, but indicate an abstract method for it to show that needs have to be extracted and passed to the mediator. In order to infer the user's needs, the method can be substituted by any other one capable to do so. Emerging Resources are extended from this kind of "resources". It is a complex of Agents and Services. One agent can be integrated with one or plural services into an emerging resource. It not only has the features of agents: autonomous, reactive, proactive, but also has the features of services, for example, performing business processes more efficiently and effectively.

As shown in Figure 1, the Emerging Resources Space is shown in the right side box. There are many agents, such as A1, A2, A3, etc.. One agent can compound with one or more services. In the left side box, the User Model Space constitutes an individual user model, a group user model, or a machine user model, which in turn compounds with agents respectively. Between the spaces, two types of dialogue exist. One is searching for required services, and the other is providing proper services. These dialogues are established between agents with the language of agents, such as ACL (Agent Communication Language).

Feedback distributions

Feedback may be obtained or captured by sensors, such as an eye-tracking device, or other devices, such as camera. We define the feedback value scope is between 0 and 100. The user satisfactory level may be set according to the real application situation. In this simulation experiment, the satisfactory level is set to be 50. That means the user is dissatisfactory if the feedback value is lower than 50.

In our simulation, we created a feedback generator which can produce four kinds of feedback distributions shown as follows.

1) RANDOM: in the Uniform distribution. As shown in Figure 2 (a), the feedback generator produces 1000 feedbacks randomly within the assumed value scope.
2) LOW: in the Gaussian distribution with the standard deviation equal to 15 and the average value equal to 25. The result is shown in Figure 2 (b).
3) MEDIUM: the same Gaussian distribution with the standard deviation equal to 15 and the average value equal to 50. The result is shown in Figure 2 (c).
4) HIGH: the same Gaussian distribution with the standard deviation equal to 15 and the average value equal to 75. The result is shown in Figure 2 (d).

The four feedback distributions represent four types of users. RANDOM means the user accepts a flowable service with a good satisfactory level randomly. HIGH means

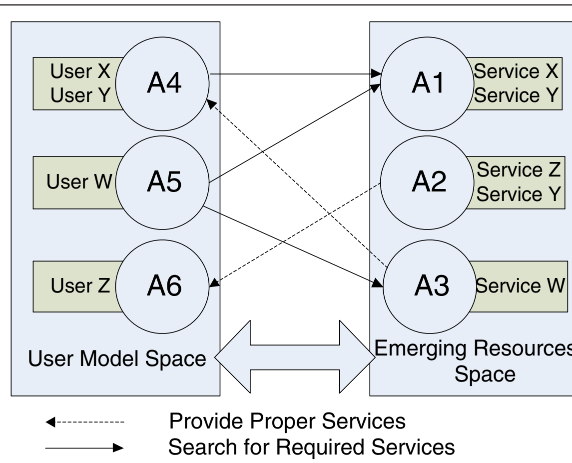

Figure 1 **Emerging resources space and user model space.**

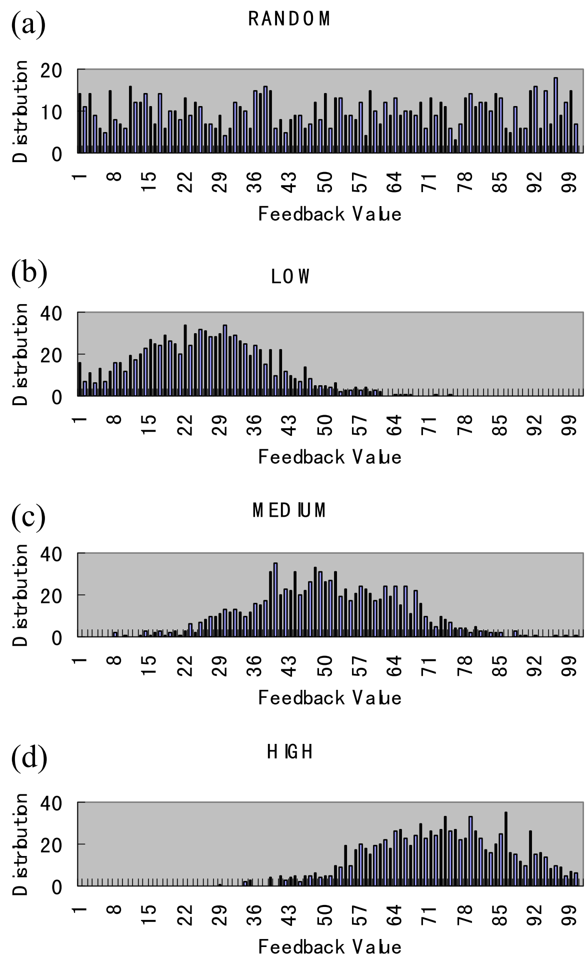

Figure 2 Feedback distributions.

the user can accept a service with satisfaction easily, while MEDIUM means the user can accept a service with satisfaction averagely. On the other hand, LOW means the user can accept a service with satisfaction difficultly. We have conducted the simulation of AEM considering these four types of users.

Regulated selection

In order to regulate the selection results by the feedback value, we dynamically group those similar but different services based on a set of five dimensions. Services are grouped according to the service cost, matching result precision, responding time, and personal and social context information with a weight of different values.

$$\text{Dimensions} = \{\text{d1}, \text{d2}, \text{d3}, \text{d4}, \text{d5}\} \tag{1}$$

$$\text{Weight} = \{\text{w1}, \text{w2}, \text{w3}, \text{w4}, \text{w5}\} \tag{2}$$

$$\text{Grouping} = |\text{Dimensions}| * |\text{Weights}| \tag{3}$$

These are the essential parts of the optimizing process for regulated selection of ambient services.

Service cost represents the price for every service usage. The service provider independently determines the price of a service based on the laws of the market. For cost-saving users, this dimension gives a higher weight for selection.

Matching result precision means a scale of precision, just as a number 1.00 is more accurate than 1. In a certain use case, users need to consider not only the value but also the result precision. For precision-priority users, this dimension works better.

Service responding time is considered as a period of time that starts at a new query and ends at a given result. For time-saving users, this dimension takes an important role for the selecting process. This is dynamic. The same service could have different responding time per usage.

Personal context information is considered as a user priority. By personality, every user has a special selecting pattern. A certain brand, or a certain expressive style or a famous service provider, all of these factors can be decisive ones.

Social context information is given from a social perspective. If there is not enough personal context information, the social context can make a supplement by assuming that the users in a same social group have a similar selection tendency.

Services can be grouped by Grouping value applied with k-means or other clustering methods.

An example for this regulated selection process is shown in Figure 3 to illustrate the basic idea of AEM. There are four users, from U1 to U4, following to the Random, Low, Medium, High feedback types separately. The right side table of Figure 3 is a result set, which are ranked by the similarity of a user's needs and the services. User U3 wants to cut down the expenses for service usage, and his personal context information shows that he tends to make an economy choice. The services are classified by the dimension – service cost. These services that are well-matched by cost are in a same service group, SA, Group A in yellow line, SB, Group B in blue line, or SC, Group C in red line. If U3 rejected S1 in Group A, it can be inferred that both S2 and S4 in Group A, in yellow line, could NOT satisfy U3. Then S3 is emerged and recommended to U3 while skipping S2.

The description is given as follows.

S, is a services set, which contains all available ambient services.

user	Satisfaction (Feedback)
U1	high
U2	medium
U3	low

Service	Rank	Service group
S1	98%	SA
S2	96%	SA
S3	92%	SB
S4	85%	SA
S5	80%	SB
S6	70%	SC

Figure 3 An example to illustrate AEM.

$\bar{S}_{(needs)}$, is a sub services set, which contains the services matched with users' needs, and $\bar{S}_{(needs)} \subset S$.

$Q_{(similarity,needs)}$, is a queue which ranks the element services in $\bar{S}_{(needs)}$ by the similarity. The similarity is decided by services and user's needs. In the traditional method, services are provided to a user one by one according to this ranking. The ranking is static.

$\bar{Q}_{(grouping,similarity,needs)}$, added the grouping information of services. And AEM can regulate the ranking based on the user feedback information. The ranking here is dynamic and adjustable.

As shown in Figure 4, Step 1, 2 and 4 compose a usual selection process (matching query and outputting result). But in AEM, we further add Steps 3 and 5. In Step 3, the preliminary results are dynamically grouped by Dimensions and weights, which have been discussed above. In Step 5, the user's satisfaction feedback is used to adjust the results. It skips the result(s) which are in the same group that the user has rejected.

Analyzing the simulation results of AEM

We design the AEM simulation by using four types of Feedback distributions discussed in the previous section. The number of services is assumed 400 in this simulation. Total services are counted to 400. At the same time, the AEM applied services are counted to 100 included in the total services. The grouping scope is considered into multiple groups from 2 to 11. The simulation interface is shown in Figure 5.

Step1: Initialization

  Needs ← user's needs

  S ← all available services

Step2: Selection & Ranking

  $\bar{S}_{(needs)}$=Match(Needs, S);

  $Q_{(similarity,needs)}$=Rank($\bar{S}_{(needs)}$ by the Similarity);

Step3: Grouping Services

  Dimension=GenerateDimension($\bar{S}_{(needs)}$);

  for each Service in $Q_{(similarity,needs)}$

    $\bar{Q}_{(grouping,similarity,needs)}$=Grouping(Dimension);

  end for

Step4: Outputting

  Provide service s to user

Step5: Getting Feedback

  if Feedback == REJECT then

    Regulate($\bar{Q}_{(grouping,similarity,needs)}$, s.Dimension);

    goto Step4

  end if

Figure 4 AEM algorithm.

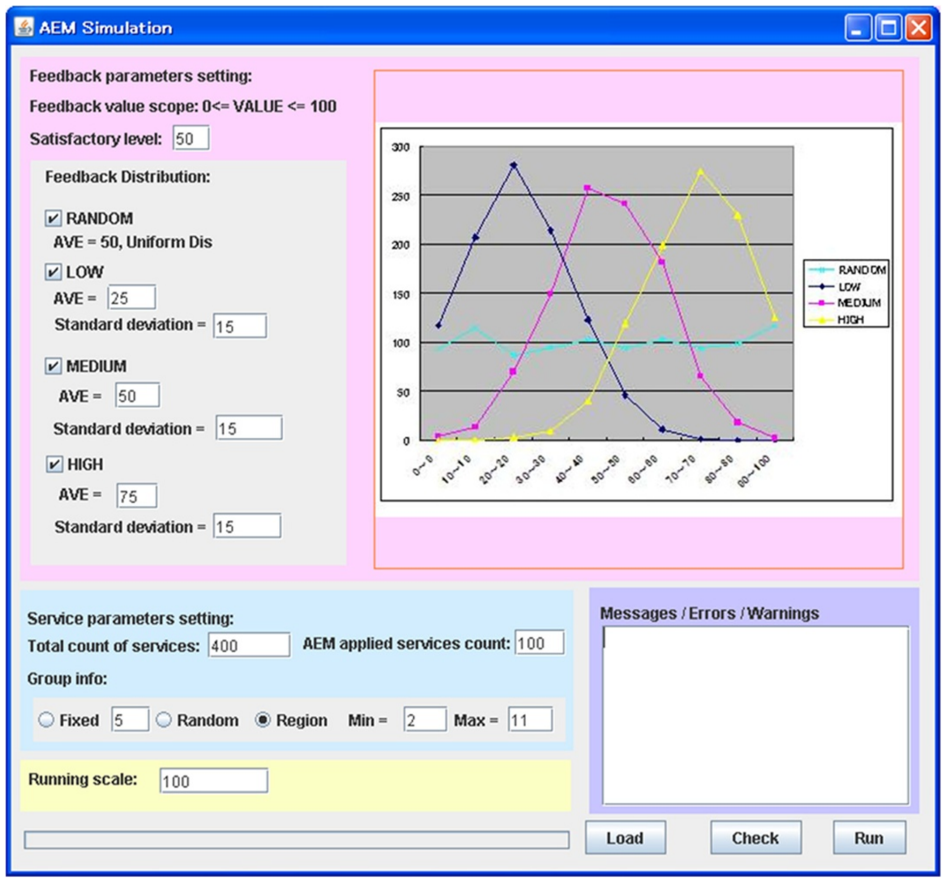

Figure 5 AEM Simulation interface.

We considered a measurement evaluation for AEM, shown in Eq. (4).

$$Saving = \sum_{i=1}^{MAXTOTAL}\left(\frac{TimesCount(i)}{TOTALTIMES} * \sum_{j=0}^{MAXSKIP}\left(\frac{TimesCount(j)}{TimesCount(i)} * \frac{Skipsteps}{Totalsteps}\right)\right) \tag{4}$$

where Saving stands for the whole evaluation. How many skipped steps can be counted by TimesCount() function. During the running of the simulation, the service group number is changed gradually. The more the skip steps there are, the higher percentage of Skipsteps/Totalstep can be. The value of Saving is a relative evaluation. Although the real cost of selection cannot be represented by Saving, which is based on the skipped steps, the selection processing time/step can be decreased greatly.

We use the notation of Step as the service number which starts to count when a service is provided and ends when a user accepts one service. Total steps represents it. Skipsteps is a skipped service number applied in AEM. The simulation result is shown in Figure 6. In the different grouping case, the AEM saves higher percentage in LOW feedback users than in others. In Figure 6, the horizontal axis is grouping information. The vertical axis is saving value with a scale of 0 to 1. Figure 6 (a) shows the 400 services divided into 5 groups, 6 groups, 7 groups and 8 groups separately, while Figure 6 (b) shows the 400 services divided into 2 groups to 11 groups separately.

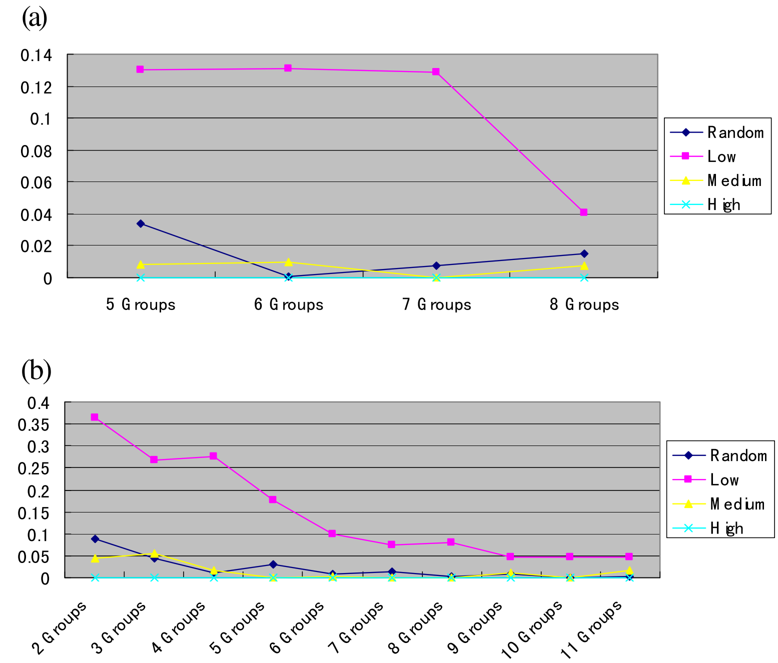

Figure 6 AEM Simulation results.

From this AEM simulation result, some of interesting findings can be described as follows.

(1) AEM can substantially improve selection process for LOW feedback users;
(2) AEM does not bring negative effect on the selection process for MEDIUM or HIGH feedback users;
(3) AEM can enhance the rationality for services selection.

Conclusion

In this study, we have proposed and developed an Adaptively Emerging Mechanism (AEM) to reduce the burden of services selection in the cloud computing environments, and thus decrease the cost of service provision and increase the user satisfaction as well. AEM has been applied and integrated into the Flowable Service Model (FSM), which has been proposed in our previous study. The major feature of AEM is that it can adaptively select the candidate components that constitute the current flowable service based on the feedback information of user satisfaction and the cost of a service, by an integrated mechanism that can dynamically group the available ambient services based on a variety of available context information. In addition to the user's feedback information such as the user's satisfaction degree, these factors, such as the service cost, matching precision, responding time, personal and social context information, have also been taken into account in the optimizing process for the selection of ambient services.

In this paper, after briefly introduce the previously proposed FSM model and how to make it context-aware, we have described and discussed the basic concept of AEM, related algorithm, and how AEM works with the FSM-based service system. We have designed and implemented a practical scenario-based simulation experiment, in which four types of feedback distributions and five service grouping dimensions have been considered. AEM can be expected to provide a regulated selection for ambient services, and it can be applied in different applications, such as service selection, personalized information recommendation.

As for our future work, we will refine AEM by introducing more measures that take into account of a variety of contexts and reflect the real-world situations. We will develop algorithms that realize AEM with complex and composite factors of human, nature and cultural aspects. Based on these, we will design and implement a context-aware service system that is based on FSM and integrates AEM in the cloud computing environments, and evaluate the performance of the system in order to demonstrate the effectiveness and usefulness of the context-aware FSM and AEM.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
The described approach was developed through discussions collectively by the both authors. Also, YZ has implemented software prototypes and drafted the manuscript, which was critically revised by QJ. All authors read and approved the final manuscript.

Received: 5 May 2012 Accepted: 13 August 2012
Published: 21 September 2012

References
1. Hilbert M, López P (2011) The world's technological capacity to store, communicate, and compute information. Science 332(6025):60–65
2. Iab F (2005) Service-oriented science. Science 308(5723):814–817
3. Mobedpour D, Ding C (2011) User-centered design of a QoS-based web service selection system. Service Oriented Computing and Applications :1–11
4. Yu Q, Bouguettaya A (2012) Multi-attribute optimization in service selection. World Wide Web 15(1):1–31
5. Shtykh R, Jin Q (2011) A human-centric integrated approach to web information search and sharing. Human-Centric Computing and Information Sciences 1(1):1–37
6. Wang S, Sun Q, Zou H, Yang F (2011) Reputation measure approach of web service for service selection. IET, Software 5(5):466–473
7. Hwang S-Y, Lim E-P, Lee C-H, Chen C-H (2008) Dynamic web service selection for reliable web service composition. IEEE Transactions on Services Computing 1(2):104–116
8. Ardagna D, Pernici B (2007) Adaptive service composition in flexible processes. IEEE Transactions on Software Engineering 33(6):369–384
9. Skoutas D, Sacharidis D, Simitsis A, Sellis T (2010) Ranking and clustering web services using multicriteria dominance relationships. IEEE Transactions on Services Computing 3(3):163–177
10. El Hadad J, Manouvrier M, Rukoz M (2010) TQoS: Transactional and QoS-aware selection algorithm for automatic web service composition. IEEE Transactions on Services Computing 3(1):73–85
11. Xiaohui G, Nahrstedt K (2006) Distributed multimedia service composition with statistical QoS assurances. IEEE Transactions on Multimedia 8(1):141–151
12. Feng Y, Cao J, Sun Y, Wu W, Chen C, Ma J (2010) Reliable and efficient service composition based on smart objects' state information. Journal of Ambient Intelligence and Humanized Computing 1(3):147–161
13. Maamar Z, Santos P, Wives L, Badr Y, Faci N, de Oliveira JPM (2011) Using social networks for web services discovery. IEEE Internet Computing 15(4):48–54
14. Shamir O, Tishby N (2010) Stability and model selection in k-means clustering. Machine Learning 80(2-3):213–243
15. Arturo O-LJ, Ariel C-OJ, Trinidad José Francisco M (2010) A new fast prototype selection method based on clustering. Pattern Analysis and Applications 13(2):131–141
16. Rudoy D, Zelnik-Manor L (2012) Viewpoint selection for human actions. International Journal of Computer Vision 97(3):243–254
17. Ahn C, Ramakrishna R (2010) A diversity preserving selection in multiobjective evolutionary algorithms. Applied Intelligence 32(3):231–248
18. Gu Q, Lago P (2011) Guiding the selection of service-oriented software engineering methodologies. Service Oriented Computing and Applications 5(4):203–223
19. Pyshkin E, Kuznetsov A (2010) Approaches for web search user interfaces: How to improve the search quality for various types of information. Journal of Convergence 1(1):1–8
20. Klyuev V, Yokoyama A (2010) Web query expansion: A strategy utilizing Japanese WorldNet. Journal of Convergence 1(1):23–28
21. Zhu Y, Shtykh R, Jin Q, Ma J (2009) A flowable service model for seamless integration of services. In: International Conference on Advances in Computer Science and Engineering 2009, Phuket, Thailand, pp 199–204

22. Zhu Y, Shtykh RY, Qun J (2010a) Harnessing user contexts to enable flowable services model. In: The 3rd International Conference on Human-centric Computing, Cebu, Philippines, pp 165–170
23. Zhu Y, Shtykh RY, Qun J (2010b) Provision of flowable services in cloud computing environments. In: The 5th International Conference on Future Information Technology, Busan, Korea, pp 1–6
24. Minsky M (2007) The emotion machine: Commonsense thinking, artificial intelligence, and the future of the human mind. Simon and Schuster

doi:10.1186/2192-1962-2-15
Cite this article as: Zhu and Jin: **An adaptively emerging mechanism for context-aware service selections regulated by feedback distributions.** *Human-centric Computing and Information Sciences* 2012 **2**:15.