International Journal of Hybrid Information Technology
Vol.7, No.2 (2014), pp.321-330
http://dx.doi.org/10.14257/ijhit.2014.7.2.28

Self-Organization Mechanisms for Service Composition in Cloud Computing

Lirong Qiu

School of Information Engineering, Minzu University of China
Beijing, China

qiu_lirong@126.com

Abstract

Cloud computing is becoming an interesting alternative as a flexible and affordable on-demand environment for deploying custom applications in the form of services. In this work, a self-organizing system model based on dynamic relation network is proposed. In the model, autonomic element can self-adapted to the weight of the relationship under the guidance of the self-organizing policy. Based on this organization model, we present the service-oriented dynamic self-organizing algorithm in Cloud computing, which implement autonomic element self-organization through service finding, service composition, task implementation and service optimization.

Keywords: *Cloud Computing, Self-organizing, Autonomous Society, Service Composition*

1. Introduction

Nowadays, there has been an increasing interest in Cloud Computing research domain due to its importance as a widely used computing platform where many difference services are published and available in Cloud data centers [1]. Clouds are used through a service-oriented interface that implements the service-related paradigm to offer Cloud services on demand. Different kind of Cloud services have been published and made available for Cloud consumers, which are deployed as self-contained components. When there is no single service has the capability to satisfy a request, there should be a method to compose those single services to fulfill Cloud consumers' goal. This composition of services should be carried out in a dynamic and automated manner to promptly satisfy consumer requirements [2].

However, Cloud participants may have incomplete knowledge of Cloud resources and other participants, considering the basic feature of cloud computing system: distributed and constantly changing. Therefore, the problem of how to effectively and efficiently compose existing services has attracted a lot of research interests and is an important open problem [3].

In Cloud service composition, collaboration between brokers and service providers is essential to promptly satisfy incoming Cloud consumer requirements and application needs [4]. Therefore, Cloud service composition system needs to implement the self-organizing of autonomic units, and then through these functional autonomous autonomic units, by the means of establishing efficient relationship of coordination, interaction and cooperation, to accomplish complex control tasks or resolve complicated problems efficiently and quickly.

Self-organizing system not only regulates or adapts its behavior, but creates its own organization, thus can be used as basic components enabling more autonomous, adaptive, and flexible in service finding, service management, and service matching to Cloud services composition [5].

ISSN: 1738-9968 IJHIT

In this paper, we address a self-organizing system for Cloud service composition. This initiative ultimately aims to develop a system capable of making decisions on its own, using high-level policies self-management, to overcome the dynamic changing request of supply and demand in Cloud service, and to reduce the barrier that complexity poses to further management.

A self-organizing framework is composed of components interacting with each other, which have several advantages over traditional systems: robustness, flexibility, capability to function autonomously while demanding a minimum of supervision, and the spontaneous development of complex adaptations without need for detailed planning [6]. The actual operation of the self-organizing system is dictated by the logic, which is responsible for making the right decisions to serve its purpose, and influence by the observation of the operational context (based on the input).

Driven by such vision, a framework based on "self-organizing" autonomic components for Cloud service composition has been proposed in this paper. Section 3 and Section 4 study the self-organization mechanisms between components. An algorithm is proposed to dynamic service organization. Section 5 is the related work and Section 6 draw a conclusion and cited the future work.

2. Methodology

Cloud computing composition is in fact used as a service-oriented interface implementing the service paradigm to offer Cloud services. The service providing and using among autonomic units is the foundation of their effective cooperation, as well as Autonomic Computing and self-organizing. By means of service discovery, cooperation and combination, autonomic units can aggregate massive computing resources, and then form a newer or more powerful computing ability.

The self-organizing system consists of a large number of interacting individuals, which can cooperate and complete the mission. In multi-agent systems, the agents contact with others through an acquaintance network, which is a list of contacts that have been previously contracted by the agent or the agent is aware of their existence. Due to the distributed nature of multi-Cloud environments, acquaintance networks are assumed to be incomplete. In this section, the self-organizing mechanism is introduced as a relationship network.

In realistic society, when people are faced with a problem, the normal way to solve this problem is relying on their own ability firstly. If the problem is still unresolved, they will resort to acquaintances and hope they can work together to solve the problem. If the problem is still remaining, the acquaintances themselves will resort to their own acquaintances to ask for help. The process will be continuing until the problem is solved. In that way, all these acquaintance relationships involved in the problem solving process is called a Social Relationship Network. Humans' organization structure and organization operation offered a favorable reference to the self-organizing of autonomic-computing system.

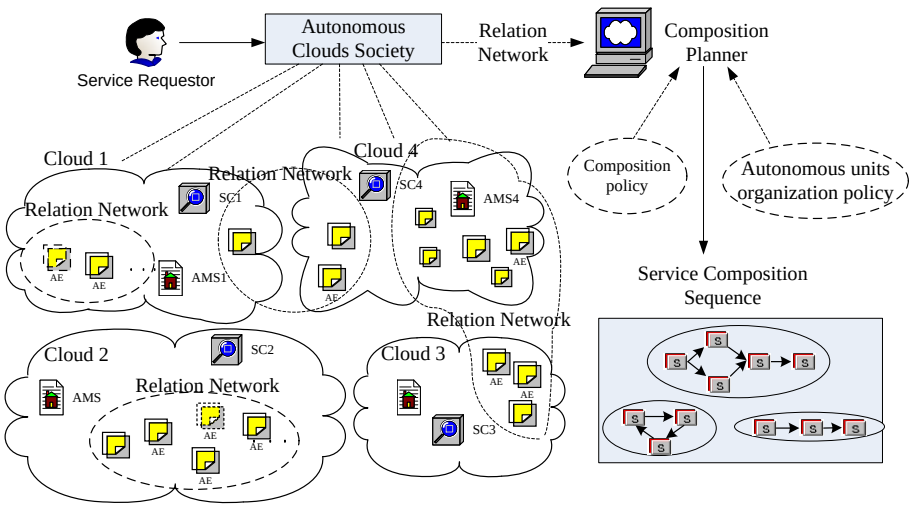

Figure 1. Self-Organization system for Cloud Service Composition

A self-organizing system model for service composition in Cloud computing are illustrated in Figure 1, which consists of services, autonomic units, autonomous Clouds society and Composition planner.

In line with strategy customized by users or business goal, autonomic unit should provide certain services when it incorporates into self-organizing system. However these certain services can be provided only in certain conditions and with certain assistance from other autonomic units, in other words, it requires services offered by other autonomic unit to complete the user customization strategy. Under this circumstance, according to the views of organizational behavior, an organization can be regarded as the distribution, combination and execution of tasks. Self-organizing system is a special kind of organizational behavior and cooperative computing method. The implementation of self-organizing in Cloud service composition system is mainly shows in four ways as follows:

(1) When autonomic unit found it facing the problem which is difficult to resolve alone, it would establish a cooperative relationship with other counterparts through the relationship network, thereby strengthening the ability of self-organizing to solve the problem;

(2) Coordinating and organizing multi autonomic units to build new services, promote service composition autonomously and improve the reusability and utilization of the Cloud service composition system;

(3) In the process of dynamic self-organizing, we should not only consider the demanded functions but also consider non-functional factors such as efficiency, performance and so on, thus to ensure acquiring a satisfying composition result to fulfill the user's requirements;

(4) Due to Cloud computing environment is an open and dynamic one, unexpected situation may occurred (such as autonomic unit leaves suddenly), and thus an accident handling mechanism should be considered.

3. Dynamic Self-Organizing Mechanism

3.1. Autonomic Unit

Autonomic Unit is a kind of software entity which has self-management. It can manage itself base on the ultimate target proposed by the designer. On one side due to the self-management feature, autonomic unit can do the configuration and management work which are owned by engineers before. On the other side, users can involve the organization and management of network application just by making high level strategy without thinking about the low level detailed implementation. This chapter gives an introduction about how to design autonomic Unit and the relationship network for self-organizing of autonomic Units. Following these methods autonomic unit can do logic decision, planning and movement and has good intelligence, expansibility, Reusability and applicability.

In this section, we formulate the relationship network between autonomic uits in self-organizing system firstly as follows:

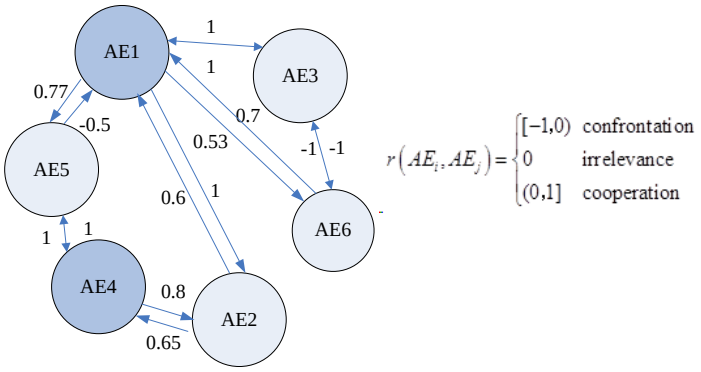

Figure 2. Relationship Network Sample

Definition 1: Relationship network can be expressed as a graph G=(V, E), in which $V = \{AE_1, AE_2, \cdots, AE_n\}$, $\forall AE_i \in V$ represents an entity in the relationship network; $E = \{R_1, R_2, \cdots, R_n\}$, in which $R_i = \{r_{ij} | i, j = 1, 2, \cdots, n\}$, and $r_{i,j} = (AE_i, AE_j)$ denotes the weight of the relation between $AE_i \in V$ and $AE_j \in V$. $\forall R_i \in E$ is the set of relationship of the network AEi.

Figure 2 shows a relationship network example: AE_1 has established cooperation relationships with many autonomic elements, such as AE_2, AE_4; AE_4 has established cooperation relationships with AE_2 and AE_5. The value on the lines between two autonomic elements represents their relation weight, known as credibility. The positive value measures the cooperation degree between the two corresponding autonomic elements.

The value of relation weight denotes the nature of the relationship between the nodes and the degree of the inter-influence. If the relationship is confrontation, we set the relation weight $r \in [-1,0)$; if it is cooperation, then we set the relation weight $r \in (0,1]$. If there is no confrontation or cooperation relationship between two autonomic elements, weight is 0.

The providing, execution and requesting service of an autonomic unit can be seen as a corresponding action (atomic action or complex action), and then the providing, execution and requesting of multiple autonomic units can be seen as multiple corresponding actions (atomic actions or complex actions).

Definition 2: autonomic unit *AE=< AID, Service, Addr，Acq，RN>*，denote identifying of the autonomic unit *AE*, belief knowledge base, service ability, policy, address, acquaintance list and the corresponding relation weight respectively, among them:

Service ={AS_1，AS_2，…，AS_m*}*，each of these elements is called an autonomic unit service AS_j，the active capacity of autonomous unit is the foundation of its service providing，each service AS_j can be expressed as a quadruple，AS_j =（*ID，Act，Prt*），in which，*ID* denote service identifier；*Act* is the description of service function，corresponding to certain autonomous action that is in the active capacity set of the autonomic unit；*Prt* is the set of property，include time, function, resource constraint and so on.

Acq={L_1，L_2，…，L_m*}*，each of the elements called an acquaintance record L_j，L_j =<*ID，Address，Type*>，respectively denote acquaintance identifier, address and acquaintance type. Type can be classified as Parent, Child, Equal and so on. The classification is according to actual demand, which described the explicit organization structure of an autonomic unit. The size of AE_i can be defined as the acquaintances number it has，marked as：|*Acq* (AE_i)|.

RN={ R_1，R_2，…，*R|Acq* (AE_i)| *}*，each of the elements ,means a trust relation R_j，R_j =< AE_i，AE_j，r>，in which r is the credibility between AE_i *and* AE_j. The credibility describe the dynamic cooperation relationship among autonomic units, autonomic unit cooperates with an acquaintance whom it shares a higher credibility with will more likely to success. In reverse, if the credibility is negative, the greater the absolute value of |r|, the confrontation between autonomic units is fiercer.

3.2. Autonomic Cloud Society

In self-organizing system for service composition in Cloud computing, we use service description language describing the service of autonomic units, for the sake of realizing efficient and quick service matching, combination, cooperation and so on. Autonomic unit can be divided into three types: service providing unit, service requesting unit and service composition broker unit (also called SC unit). A service providing unit can provide service; a service requesting unit requests a certain service from the service providing units, while a SC unit can help service requesting units locating the corresponding service providing units, which also has the functions of service registration, selection, consultation, combination and so on.

An autonomic unit can be either a service providing unit or a service requesting unit. The autonomic unit determines whether a given task can be fulfilled bases on its own service capability. If a complex task that need cooperation to fulfill, the autonomic unit will seek acquaintances in acquaintance list to establish corresponding cooperation in line with the relation weight R, and then form a virtual autonomic unit organization, as known as autonomous society.

Definition 3: autonomous Clouds society can be expressed as a quadruple: $ACS = \{AE_s, SC, AMS, RN\}$, AE_s is the finite set of autonomic units $AE_s = \{AE_1, AE_2, \ldots, AE_n\}$; SC is service agent element who has a special role in the autonomous computing organization with the function of service selection, consultation, combination and so on; CMS is a special unit—also called as registration element—who is responsible for the registration, revocation and deletion of all autonomic elements; *RN* is the relationship network setting all the autonomic elements contained, which can be expressed

as $\cup_{AE_i \in AE_s} RN(AE_I)$. We call the number of autonomic elements in the set AE_s as the size of the autonomous Clouds Society, expressed as $|AE_s|$.

Each autonomic unit can either send tasks to acquaintances or receive tasks from acquaintances or non-acquaintances. In order to complete certain tasks in a shorter time, an autonomic unit in the relationship network needs to interact with more acquaintances. The certain tasks in the relationship network would experience the process of transmitting, decomposing and execution, which will consume a large amount of communication resources. In extreme case, suppose Autonomic unit *AE*'s size *|Acq(AE)|)* = 1, the average costs of a task will reach *Cost(Task)* = n / *2*, that will bring tremendous communication expenditure to the system. Another extreme case is *|Acq(AE)|* = n – 1, which means every autonomic unit has a relationship with all the rest counterparts. In this case the average cost *Cost (AE，Task) =0,* but in other hand the space complexity as well as costs of acquaintance list maintaining will be very high. To a certain autonomic unit in Autonomic Computing system, we tend to achieve best average performance in interacting with other counterparts in system.

From both explicit organization relation and dynamic cooperation relation the two perspectives, relationship network model precisely described the complicated relation among autonomic units. Taking hierarchical structure as an example, every autonomic unit has one superior but can has multi subordinates, which form a tree structure as shown in Fig. 2, $Acq(AE_1)$={< AE_2，*Child*>，< AE_3，*Child*>，< AE_4，*Child*>}.

That means in initial condition, the superior and subordinate autonomic units in relationship network can establish complete cooperation relationship, so we can set the relation weight to 1. For example, if r(AE_1, AE_2)=1，r(AE_2, AE_1)=1，r(AE_1, AE_3)=1, and AE_2 needs cooperation from AE_3, the massage can only transmit through AE_1. The cooperation relationship between superior and subordinate may alter in actual operation process, for instance, if AE_3 always can't complete tasks it received, then AE_1 will alter the credibility relate to AE_3. Hence, autonomic unit need to alter relevant relation weight according to actual interaction situation, and we utilize the method introduced in [15].

4. Self-organizing Mechanism for Cloud Service Composition

Self-organizing system proceeds in an open, dynamic environment, so the autonomic units may in and out the autonomic Cloud environment freely. Autonomic units' behavior depends on the self-organizing mechanism of each autonomic unit. By means of acquaintances finding, service combination and cooperation in dynamic relationship network, autonomic units can complete complex tasks they can't handle along.

The autonomic unit self-organizing is a highly intricate process, when an autonomic unit found the problem is difficult to solving alone or cooperative solution is more efficient, it can authorize other autonomic units to manage the whole organizing process or manage by itself. To achieve organization goal, organization manager will search each autonomic unit who can undertake subtasks via the relationship network and choose acquaintances with high credibility to cooperate. Autonomic unit self-organizing mechanism is relate to factors include performance, benefits, quality, time and so on. To autonomic units, the main purpose for providing service is to get certain benefits, and service requesting units will cost a lot for seeking assistance, the whole process can be describe as（*AE*，*RN*，*G*，*U*，*SC*，*CMS*）:

Initial autonomic units set *AE={*AE_1, AE_2, …, AE_n*}*, relationship network *RN*=(*V*, *E*), composition goal *G*={g_1, g_2,…, g_n}, benefit distribution *U*={u_1, u_2, …, u_n}, service agent unit *SC* and Cloud manager unit *CMS*.

For all the unrealized subgoals g_1, g_2, …, g_n, *CMS* search autonomic units who can undertake the subgoal g_i in turn (the autonomic units in this time are not ones really undertake corresponding subgoal, they just match target functions with the subgoal. Autonomic units also consider the trust relations and benefit issue about involved undertaking units), and meanwhile sort these subgoals according to the subgoal relationships. If there are any deadlocks with subgoals, a message should be passed to the autonomic unit for corresponding changes.

If subgoal g_i can't be complete by one autonomic unit but can be done jointly by several autonomic units: If there are service series to complete the subgoal n_i SC database already, the return directly, Otherwise SC call service composition algorithm to organize multi autonomic units to build new service series, then return. If service compositing failed, then CMS directly notices the autonomic unit that the requesting can't be meets under present Autonomic Computing environment.

If subgoal g_i cannot be undertaken by any autonomic units, then *CMS* notice the requesting unit that the requesting can't be meets, and then return.

According to goal distribution and execution method, *CMS* determines the undertaker of each subgoals and the dividing of benefits, and then calls each service one by one in line with goal realizing sequence. For composite service, SC coordinates and organizes corresponding autonomic units to provide it, *CMS* is responsible to monitor execution and handle unexpected situation that may occur.

CMS evaluates the whole process according to task evaluation methods published in [15] and saves or modifies cooperation strategy, and updates trust relationships.

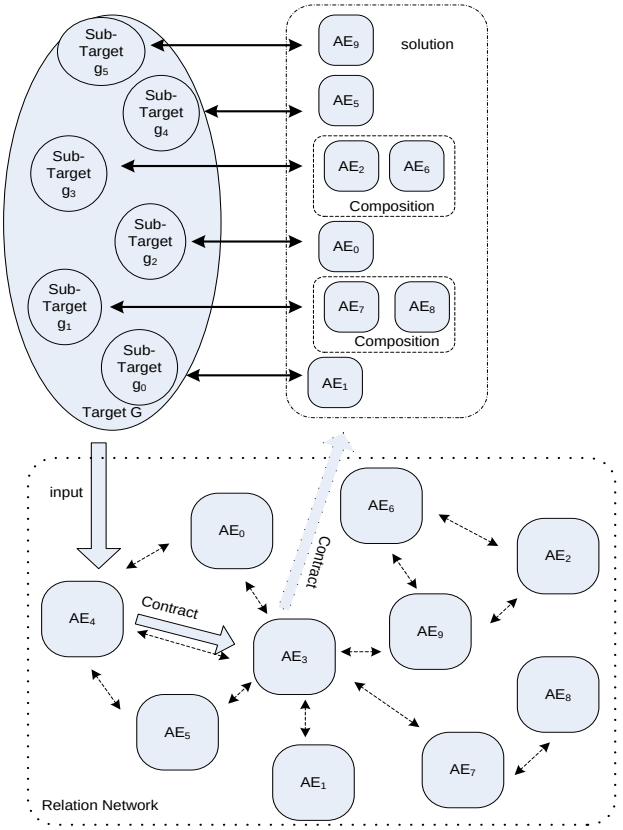

Figure 3. Self-Organizing Process Sample

Figure 3 shows a simple self-organizing process sample, the current service request has six sub goals $G=\{g_0, g_1, g_2, g_3, g_4, g_5\}$, with benefits distribution strategy $U=\{0.1, 0.1, 0.2, 0.1, 0.1, 0.4\}$. Suppose AE_6 is entrusted to organize the whole process via user preference, comparison of relation weight, ability matching and benefits consulting. Then *CMS* determined that the subgoals g_0, g_2, g_4, g_5 can be undertaken be AE_1, AE_0, AE_5 and AE_9 respectively, while g_1 must be completed jointly by AE_7 and AE_8 and g_3 should be completed jointly by AE_2 and AE_8.

Finally, on the basis of task completions of each autonomic unit to make corresponding evaluation, and then according to the evaluation all the autonomic units in the whole virtual organization would update or establish trust relations as well as alter or save cooperation strategy.

5. Related Work

In [9], the service composition is partitioned and agents are assigned to monitor and capture the progress of the composition. However, the adopted centralized perspective to handle errors causes that a single fault impedes the development of the entire composition.

A bio-inspired, self-organizing method to support deployment of services on virtual machines in a cloud infrastructure is proposed in [10], in order to meet the service level agreements and to minimize the number of required virtual machines. But there is lack of complete description of the architecture and algorithms of the proposed system.

The idea of applying self-organization systems (*e.g.* software agents) or managing Cloud services was first introduced in [7, 8], which is the earliest efforts in applying negotiation agents for self-organizing agent-based approach for dealing with Cloud service composition.

In [3], a framework of service composition in multi-Cloud base environment is presented, and three different Cloud combination methods are presented to reduce the costs of finding the minimum clouds. But this is a centralized method and need complete knowledge of all existing web services.

Our work was inspired by the work proposed in [4] and [2], which introduced a marriage between Clouds and Agents. In [4], an agent-based Cloud service composition is presented. The self-organizing agents using acquaintance networks and the contract net protocol to evolve and adapt Cloud service composition. In [2], Cloud computing models and architectures are discussed to implement high-performance complex systems and intelligent applications by using of Cloud systems and software agents.

As cited in [2] and [4], Cloud-enabled agents can coupling agents and large-scale dynamic distributed computing platforms bringing new properties of system, such as autonomy, pro-activity, negotiation and learning.

6. Conclusion and Future Work

Cloud computing is becoming an interesting alternative as a flexible and affordable on-demand environment for deploying custom applications in the form of services. In this work, a self-organizing system model based on dynamic relation network is proposed. In the model, autonomic element can self-adapted to the weight of the relationship under the guidance of the self-organizing policy. Based on this organization model, we present the service-oriented dynamic self-organizing algorithm in Cloud computing, which implement autonomic element self-organization through service finding, service composition, task implementation and service optimization.

Future work will include more precise and accurate simulations as well as a complete description of the architecture and algorithms of the system.

Acknowledgements

Our work is supported by the National nature science foundation of China (No. 61103161), the Program for New Century Excellent Talents in University (NCET-12-0579).

References

[1] M. Armbrust, *et al.*, "A View of Cloud Computing", Communications of the ACM, vol. 53, no. 4, **(2010)**, pp. 50-58.
[2] D. Talia, "Cloud Computing and Software Agents: Towards Cloud Intelligent Services", In proceedings of 12th Workshop on Objects and Agents, vol. 741, **(2011)**, pp. 2–6.
[3] G. Zou, Y. Chen, Y. Yang, R. Huang and Y. Xu, "AI Planning and Combinatorial Optimization for Web Service Composition in Cloud Computing", In proceedings of the International Conference on Cloud Computing and Virtualization, (**2010**).
[4] J. O. Gutierrez-Garcia and K. -M. Sim, "Self-Organizing Agents for Service Composition in Cloud Computing", In proceedings of 2nd IEEE International Conference on Cloud Computing Technology and Science, (**2010**).
[5] F. Heylighen and C. Gershenson, "The meaning of Self-organization in Computing. IEEE Intelligent Systems, vol. 18, no. 4, (**2003**), pp. 72-75.
[6] F. Heylighen, "The Science of Self-organization and Adaptivity", In: Knowledge Management, Organizational Intelligence and Learning, and Complexity, In: The Encyclopedia of Life Support Systems, EOLSS Publishers Co. Ltd., **(2003).**
[7] K. M. Sim, "Agent-based Cloud Commerce. In Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management, HongKong, (**2009**), pp. 395-406.
[8] K. M. Sim, "Towards Complex Negotiation for Cloud Economy", In proceedings of the 5th International Conference on Grid and Pervasive Computing, LNCS 6104, Taiwan, (**2010**), pp. 395-406.
[9] G. B. Chafle, S. Chandra, V. Mann and M. G. Nanda, "Decentralized orchestration of composite web services", In proceedings of the 13th international World Wide Web Conference on Alternate Track papers & Posters, ACM, NewYork, (**2004**), pp. 134-143.
[10] B. A. Caprarescu, *et al.*, "SOS Cloud: Self-Organization Services in the Cloud", In: Bio-Inspired Models of Network, Information, and Computing Systems, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 87, (**2012**), pp. 48-55.
[11] B. A. Caprarescu, "Robustness and scalability: a dual challenge for autonomic architectures", Proceedings of the Fourth European Conference on Software Architecture: Companion, (**2010**), pp. 22–26.
[12] F. Lin, "Research on Self-Organizing and Self-Optimizing in Autonomic Computing", Doctor Thesis of Chinese Academy of Sciences, (**2009**).
[13] H. Jamjoom, *et al.*, "Self-Organizing Network Services", Ann Arbor. University of Michigan, CSE-TR-407-99, (**1999**).
[14] C. Adam and R. Stadler, "A Middleware Design for Large-scale Clusters Offering Multiple Services", IEEE Transactions on Network and Service Management, vo. 3, no. 1, (**2006**), pp. 1-12.
[15] H. Tao, *et al.*, "A Multi-agent Negotiation Model based on Acquaintance Coalition and Extended Contract Net Protocol", Journal of Computer Research and Development, vol. 43, no. 7, (**2006**), pp. 1155-1160.

Author

Lirong Qiu

She received his M.Sc. in Computer Sciences (2004) and PhD in Information Sciences (2007) from Chinese Academy of Science. Now she is full professor of computer sciences at Information Engineering Department, Minzu University of China. Her current research interests include different aspects of natural language processing, artificial intelligence and distributed systems.

