

A Reinforcement Learning-based Approach for Spontaneous Service Emergence in Pervasive Computing Environments

M. Bakhouya

HPCL, George Washington University
Washington DC, USA
bakhouya@gwu.edu

Abstract: In this paper, a reinforcement learning approach based on the creation of affinity relationships between users having similar preferences and interests is presented. This approach suits pervasive environments to promote computer-based human interactions for spontaneous service emergence. By mimicking the human immune system behavior for the implementation of pervasive applications, services or user interests are represented by agents that can establish relationships based on affinities. Affinity corresponds to the adequacy with which two services could bind to share common interest attributes. These affinities are adjusted or reinforced by users' satisfaction regarding their interaction and dynamic network condition changes.

Keywords: Pervasive computing, Spontaneous service emergence paradigm, Propitient multi-agent systems, Reinforcement learning.

1. Introduction

Ubiquitous and Pervasive Computing (UPC) are new infrastructure networks with a goal to provide computing and communication services anytime and everywhere. In Ubiquitous Computing (UC), the objective is to provide users the ability to access services and resources all the time and irrespective to their location [1,2]. Since Pervasive Computing (PC) often considered the same as ubiquitous computing in the literature, Gaber in [2] provides a classification of these two emergent infrastructures and argued that PC is a related concept that can be distinguished from ubiquitous computing in terms of environment conditions [3]. According to this classification, the aim in UC is to provide any mobile device an access to available services in an existing network all the time and everywhere while the main objective in PC is to provide spontaneous services created on the fly by mobiles that interact by ad hoc connections. In other words, pervasive computing concerns the use of mobile computing technology to enhance people's interactions during unexpected contexts. Interactions between users are based on geographic proximity and context-awareness and can occur in a shared local such as in the office, in the elevator, at the grocery store, in the class, in the library, in the pool area, in the gym, etc. Users live close enough so they can interact and eventually see and meet each other in real life if they wish. Usually, interactions can occur when users find that they share intersects, hobbies, jobs, and

so on. However, this new environment enables to create new forms of computer-based human interactions and context-aware collaboration.

In 2000, Gaber [2] has point out that most of research works to date in service discovery and composition are based on the traditional Client-Server paradigm (CSP). He states that this paradigm is impracticable in ubiquitous and pervasive environments and does not meet their related needs and requirements that are adaptability to a dynamic environment. Therefore, Gaber [2,3] has proposed two alternative paradigms to the traditional client to server interaction paradigm (CSP) to design and implement Ubiquitous and Pervasive applications: the adaptive Services-to-Client Paradigm (SCP) and the Spontaneous Service Emergence Paradigm (SEP). The adaptive Services/Client Paradigm (SCP) can be considered as the opposite to CSP. Indeed, with the traditional CSP paradigm, it is the user who should initiate a request, should know a priori that the required service exists and should be able to provide the location of a server holding that service. With the alternative SCP paradigm, it is the service that comes to the user. In other words, in this paradigm, a decentralized and self-organizing middleware should be able to provide services to users according to their availability and the network status. As pointed out in [2], such a middleware can be inspired, for example, from a biological system like the natural immune system. More precisely, unlike the classical Client/Server approach, each user request is considered as an attack launched against the global network. An immune networking middleware reacts like the natural immune system against pathogens that have entered the body. It detects the infection (i.e., user request) and delivers a response to eliminate it (i.e., satisfy the user request). Recently, agent-based approach, with self-adapting and self-organizing capabilities, has been proposed in [3,8] to implement the SCP paradigm for UC applications. This approach, inspired by the human immune system, provides scalable and adaptive service discovery and composition systems for ubiquitous computing environments.

The second alternative SEP paradigm to the Client/Server one is more adequate for pervasive applications. It involves the concept of spontaneous emergence service composition that suits pervasive environments. More precisely, spontaneous services can be created on the fly and be provided by mobiles that interact by ad hoc connections [2,3].

The SEP could be also implemented by a natural system that involves self-organizing and emergence behaviors [2,3]. Natural and biological systems like the human immune system [10] has a set of organizing principles such as scalability, adaptability and availability that are useful for developing a networking model in highly dynamic and instable setting [2,7]. In this paper, an approach to implement the second paradigm is presented.

The rest of the paper is organized as follows. In section 2, we present a related work. Section 3 presents the proposed agent-based approach. Section 4 presents the results of the simulation. Conclusion is given in section 5.

2. Related Work

In ubiquitous and pervasive computing, service composition plays a fundamental role, and automation will be essential to improve speed and efficiency of users' responses. Service composition is a process that takes several component products or services, and hands them together to meet the needs of a given user. Generally, in UPC, automatic service composition requires dealing with four major research issues: service matching and selection, service coordination and management, scalability, and adaptiveness. The service matching and selection is the first step in creating any composite service and requires a service discovery system to locate the service components that provide functionalities to be placed in the new service. Service discovery systems should be able to find out all services conforming to a particular functionality, irrespective of its way of invocation. To achieve this requirement, semantic level reasoning of describing the functionality of service is required. This is why the Web service community has developed a number of languages to formally describe services in order to facilitate their discovery [14,15,16]. The discovery system should be also scalable across large-scale networks and adaptable to dynamic changes especially when services dynamically join and leave the network. To implement this process, most of service discovery systems like SLP [11] and SSDS [12] are centralized and require that information about service components must be stored in component directories that can be accessed at runtime. These centralized systems cannot meet the requirements of both scalability and adaptability simultaneously. Several decentralized systems are proposed to address these issues. A survey and a classification of service discovery systems proposed in the literature are presented in [8].

Service coordination and management is the second issue to be addressed in automatic service composition. More precisely, composition platforms must have one or some brokers that coordinate and manage the different services involved in the composition. The problem of coordination and management becomes difficult when the entities are distributed across the network and poses a scalability problem, especially when numerous users are concurrently making composite service requests. For example, if elementary services are distributed across the network, the use of centralized broker requires a lot of CPU cycles to process users' composition requests. In dynamic networks, where services are coming up and going down frequently, the service composition platform should be able to adapt its

composition by taking maximum advantage of the currently available services. This increases the composite service availability in a dynamically changing environment.

These important issues can be classified into two major directions in dynamic service composition research. The first direction addresses languages to specify and describe services including complex planning mechanisms that utilize these descriptions to generate composite services. In other words, this research direction is trying to define languages to specify services, invocation mechanisms and composite services. The second direction aims to develop architectures that enable scalable, fault tolerance, and adaptive service composition. A survey and classification of service composition approaches proposed in the literature is presented in [18].

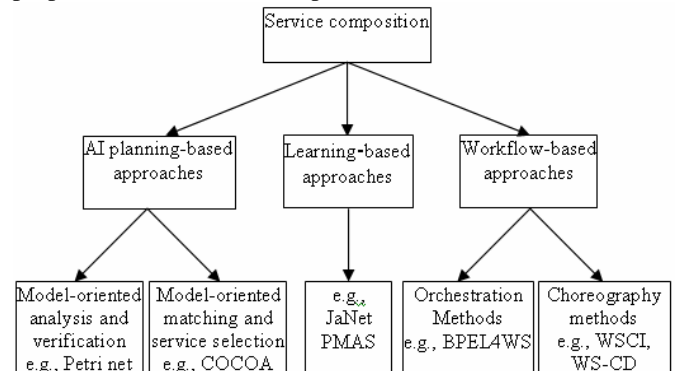


Fig.1. a classification of services composition approaches

Figure 1 presents a classification of service composition approaches into three categories: workflow-based approaches, AI planning-based approaches, and learning-based approaches [18]. Workflow-based approaches are classified into two categories: orchestration methods and choreography methods [21]. Orchestration methods use a central coordinator (i.e., broker) to describe, combine and invoke services. Several Web service flow specification languages like BPEL4WS (Business Processes Execution Languages for Web Services) have been proposed to compose Web services [23]. Unlike orchestration methods, choreography methods do not assume the exploitation of a central coordinator to control the order of the execution of the participating services. These methods define the conversation that should be undertaken by each participant. In other words, the composition is achieved via peer-to-peer interactions among the services. Therefore, choreography methods are more collaborative by allowing each involved party to describe its part in the interaction [22]. For example, Web Service Choreography Interface (WSCI) is an XML-based description language, proposed by BEA Systems, Intalio, SAP AG, and Sun Microsystems, for choreographing message flow between Web services.

AI planning-based approaches are also classified into two categories: models-oriented analysis and verification and models-oriented matching and service selection. Model-oriented analysis and verification are proposed mainly to translate service composition descriptions (e.g., Darpa Agent Markup Language for Service description, DAML-S [25]) to formal description using formal methods like Petri net [24]. The objective is to describe, simulate, and verify the behavior of the service composition at design time. However, they do

not deal either with the location of services nor the recognition or the selection of those that match to the requested service and create thus the suitable. Unlike the specification and the correctness verification of the composition process, the aim of models-based matching and service selection is how to identify candidate services, compose them, and verify how closely they match a request. Recently, an approach classified as model-based matching and service selection strategy is proposed in [5]. In this approach, called COCOA (CONversation-based service COMposition middleware), the DAML-S specification is complemented in one hand by an execution model that preserves the DAML-S semantics and on the other hand by an implemented computational architecture that enables and ensures dynamic, run-time semantic service discovery, interaction, and composition across the Web. This approach permits to select arbitrarily one among the resulting composite services as they all conform to the target user task. However, it not allows the selection of the most effective composite service among the eligible ones.

In general, most of research works to date in service composition are based on a broker that chooses the composite service after calculating all the candidates. Client/Server strategy based on a broker is not scalable due to the large amount of available services and the dynamic change of the environment. To set up self-adaptive service composition systems, learning mechanism could be used [2,3]. Itao et al. in [9] have proposed a platform, called Ja-Net, for service composition in ubiquitous and pervasive computing environments. Ja-Net architecture is motivated by the observation that the above desirable properties, such as scalability and adaptability, have already been realized in various biological systems like the bee colony. However, Ja-Net is not adaptive to pervasive computing environments. This is because, if a node fails or leaves the network, mobile agents, called cyber-entities, residing in it is destroyed and their state information is also removed. Consequently, agents could be being misinformed about the presence of some services at particular nodes. Thus, since agents' collaboration requires global information state maintenance, this platform does not address pervasive environments (e.g., mobile ad hoc networks) where nodes are mobiles and could be partially connected. An approach inspired by immune system with the intention to extend the Ja-Net approach for service discovery and composition for pervasive computing environments has been proposed in [4].

In this paper, an agent-based learning approach, with self-adapting and self-organizing capabilities to involve the concept of spontaneous emergence service for pervasive environments is presented. This approach offers the possibility to nodes to recognize, learn and create affinity relationships to their network neighborhoods having similar interests. More precisely, when two users come in a close physical proximity their respective interests enable their personal mobile nodes to interact and exchange information and access each other services. This ability to establish relationships among users' nodes can be used to facilitate and augment users' interactions without a priori knowledge about the presence of other users sharing common interests. For example, in real-time collaboration and face-to-face game

play, this approach offers the ability for users to find other gamers with similar interests and play a game together. In other words, a gamer can set up an ad-hoc game among users sharing similar interests. The game service emerges without a priori intention of users to request this service.

3. Learning-based approach

As stated by Gaber in [2, 3], the second alternative paradigm to the Client/Server one for service composition that suits pervasive environments should involve the concept of spontaneous emergence. This paradigm can be carried out by an inspired natural immune middleware that allows the emergence of ad hoc services on the fly according to dynamically changing context environments such as computing context and user context [2]. In this model, a service is represented by an agent. The service is characterized by the set of functionality that is provided, while the agent represents the behavior and activities of the service and its capability of performing tasks as a component of an ad hoc network (i.e., an ad hoc coalition). Like in the immune system, agents correspond to the immune system B-cells. Agents establish relationships based on affinities to form communities. Communities are groups of users who share a common purpose, interest, or goal. A community of agents corresponds to the idiotypic network in the human immune system [3]. The emergence and the behavior evolution of services depend on the agents' internal states, the agent-to-agent interactions and on the context sensed by the agents during their interactions [2]. More precisely, agents' interactions, i.e. B-cells, act like the idiotypic network of the immune system by stimulation/suppression chain to influence the emergence of an immune response against antigens i.e., the ad hoc context [4]. It should be noted that in immune system there are two strategies to create affinity networks: a proactive strategy and a reactive strategy. In the first strategy, affinity networks are emerged by the introduction and the stimulation of the antigen. However, in the second strategy the affinity networks are emerged without the introduction of the antigen but through stimulation/suppression chains between B-cells [10]. The first strategy has been applied in [4] for service discovery and composition in pervasive computing environments.

In this paper, the reactive strategy is applied for the service emergence without a priori intention of users to request these services; no prior planning on behalf of users. More precisely, agents together with their affinity relationships as a whole form a propitious multi-agent system (PMAS) [6]. A propitious system is a system with the ability to self-organize in order to adapt towards the most appropriate agent organization structures according to unpredictable changes in the environment and users' interests. An emergent behavior is delivered as a result of agents-to-agents and agent-to-environment interactions that adapt until the system reaches a most suitable affinity network [4]. In other words, a propitious multi-agent system implements the SEP for pervasive applications. To achieve this, agents that represent services can accomplish the following steps or phases. The first phase, called "building affinity relationships", allows for an agent to find appropriate agents to interact with. The second phase, called "learning affinity relationships", allows

an agent to learn from its satisfaction the appropriate established relationships. The third operation, called “dropping affinity relationships”, permits for an agent to drop its or their relationships with other agents. These phases allow agents to adapt to dynamic environment conditions through the establishment of affinity relationships to other agents. These phases are described in next sections.

3.1 Building affinity networks

During this phase, agents representing services search for other agents in order to construct affinity relationships. Affinity corresponds to the adequacy in which two similar services to bind. Each service is represented by an agent and each node can provide one or several services. To determine this affinity, services can be expressively described by a language description in order to obtain effective matches between their capabilities. We can use a semantic language, such as OWL or DAML-S [15] to determine the exact matching between their provided services. In the discovery affinity algorithm, semantic keywords are used for matching relevant keywords to each nearby agent. Hence, an agent will co-operate only with selected and suitable co-agents. Also, an individual agent can be a member in parallel of several affinity networks.

Let us consider two nearby agents, A_i and agent A_j . The agent A_i represents the service s_i described by d_i . The agent A_j represents the service s_j described by d_j . The semantic keywords matching technique used in the discovery affinity algorithm takes d_i and d_j as an input and produces an affinity value that corresponds to portion of matching between the service s_i and the service s_j based on their set of semantic keywords K_i and K_j respectively. More precisely, this value is calculated as the number of keywords that are in common between s_i and s_j as follows: $m_{ij}(t) = match(s_i, s_j)$, where the initial step t is equal to 0. $match(s_i, s_j)$ is the number of intersection set $|K_i \cap K_j|$ between the set K_i describing the service s_i and the set K_j describing the service s_j . In the discovery algorithm, if this value $m_{ij}(t)$ is greater than 0 a relationship will be established in two directions between the agent A_i and the agent A_j . This means that one at least of keywords should be similar. Therefore, initially $m_{ij}(t)$ and $m_{ji}(t)$ could be set to an equivalent value. For example, as depicted in figure 2, the agent A_1 provides a service s_1 described by three semantically keywords (k_1, k_2, k_3) establishes a relationship with the agent A_3 holding a service s_3 described by three semantically keywords (k_1, k_3, k_5) . The affinity values m_{13} and m_{31} are initialized by the value 2 that corresponds to the number of similar keywords (i.e., k_1, k_3) between the service s_1 and the service s_3 .

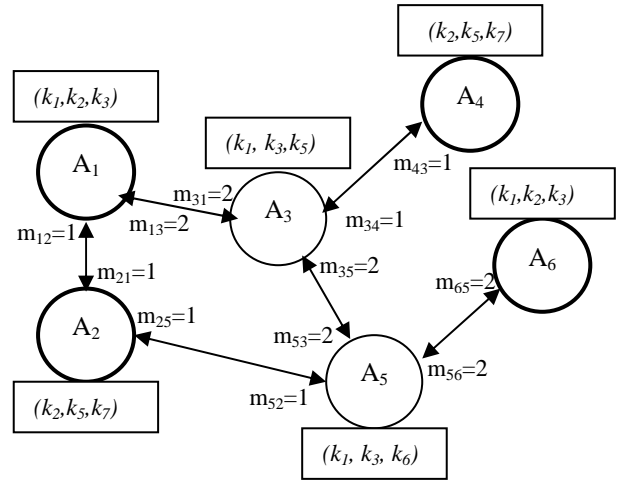


Fig. 2. An affinity network created between six agents based on keywords similarity. Two services are emerged, one between the agent A_1 and A_6 and other between the agent A_2 and the agent A_4

After building the affinity relationships, agents inform their users that some similar interests are found and they can start to learn the utility of relationships. Until now there are no emerged services, but after the agent A_i establishes an affinity relationship with an agent A_j , the agent A_j takes the interest of the agent A_i and tries to select the most suitable agent in his community. During this phase, the interest of agent A_i is considered as a request and the agent A_j forwards it to the members of the community seeking agents that are more relevant to this request. For example, as depicted in the figure 2, the agent A_1 has a service that the agent A_3 does not have. Thus, the agent A_3 in turn asked the agent A_5 holding similar service than A_3 . The agent A_5 asked the agent A_6 holding similar service than A_1 . After this phase, the agent A_1 and A_6 inform their users and start the global adjustment of the selected path as described in the next section.

Since agents have affinity relationships to other agents with at least one similar keyword, this guides some them to locate other ones that possess similar keywords. This behavior emerges through the establishment of the affinity networks.

3.2 Learning affinity Networks

In this phase, a reinforcement learning mechanism, presented in [4], is used to allow adjusting the affinity values according to users' satisfactions. More precisely, the affinity m_{ij} between an agent A_i and an A_j is a real variable that is adjusted or reinforced based on two levels of satisfaction. The first level, called local satisfaction level, is determined by services offered by neighboring agents. The second level, called global satisfaction, is determined by the user satisfaction (i.e. user context) with respect to the emerged service.

Let's consider u_i^l (resp. u_j^l) denotes the local satisfaction value of the agent A_i (resp. A_j) according to its potential cooperation with a neighboring agent A_j (resp. A_i). Using the local satisfaction value, the agent A_i adjusts, at a step t , the affinity value of the relationship with the agent A_j using the following equation:

$$m_{ij}(t) = m_{ij}(t-1) + \tau(u_i^l \cdot u_j^l - g(m_{ij}(t-1))) \quad (1)$$

where τ is a positive value between 0 and 1. The value u_i^l (resp. u_j^l) is determined as the ratio between the number of similar keywords over the total number of keywords described the service provided by A_i (resp. A_j). For example, if all keywords are similar, the satisfaction is set to 1. The affinity value m_{ij} is mapped to a value between 0 and 1 by using the logistic equation $g(m_{ij}(t)) = 1/1 + e^{-m_{ij}(t)}$. When a user is satisfied with an emerged service provided by other users, the value of a global satisfaction denoted by u^g is set to 1; otherwise the satisfaction value is set to ratio between the number of similar keywords, between the agent A_i and the last agent A_k in the path. This satisfaction value is then forwarded to all agents (i.e., participating agents) that interact with to provide the emergent service according to the following equation:

$$m_{ij}(t) = m_{ij}(t-1) + \tau(u^g - g(m_{ij}(t-1))) \quad (2)$$

To show the impact of local users' satisfaction on the strength of affinity relationships (Eq. 1), let us consider the link between two agents, A_i and A_j , inside a given community. At each step, for example, the agent A_i interacts directly with the agent A_j without intermediary agents. There are three possible cases. In the first case, A_j holds a similar service required to A_i . In the second case, there are some of keywords that are similar between services provided by A_i and A_j respectively. In this case, we assume that the satisfaction values, u_i^l and u_j^l , depend on how many keywords are similar. Figure 3 shows results of affinity adjustment of 50 satisfaction feedbacks, where the satisfaction values, u_i^l and u_j^l , are set to 1, 0.7, 0.1. In the case when u_i^l and u_j^l are set to 1, the satisfaction value m_{ij} increases quickly when it is near to 0. We can see also that when u_i^l and u_j^l are set to 0.7 or 0.1, the affinity value m_{ij} decreases quickly when it is near 0.

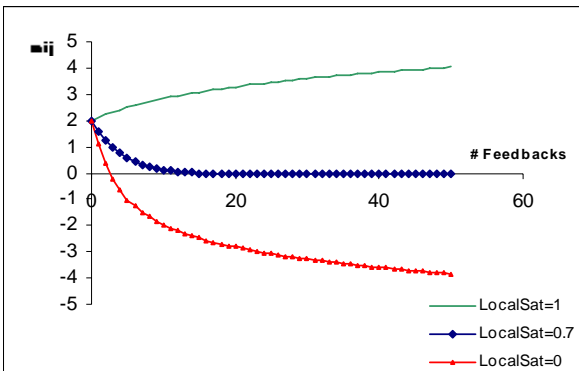


Fig. 3. the variation of the affinity value, m_{ij} , between A_i and A_j when their local satisfaction values are set to 1, 0.7, and 0

To show the impact of the global users' satisfaction on the strength of affinity relationships (Eq. 2), let us consider the link between A_i and A_j inside a given community. Similar to the local satisfaction, figure 4 shows results of affinity adjustment of 50 satisfaction feedbacks where the global satisfaction value u^g is 0, 1, or a random (between 0 and 1). With random value of the satisfaction, the affinity value m_{ij} increases and decreases in function of the value of u^g . For all this case the value of τ is set to 1 and the initial value of m_{ij} is equal to 2.

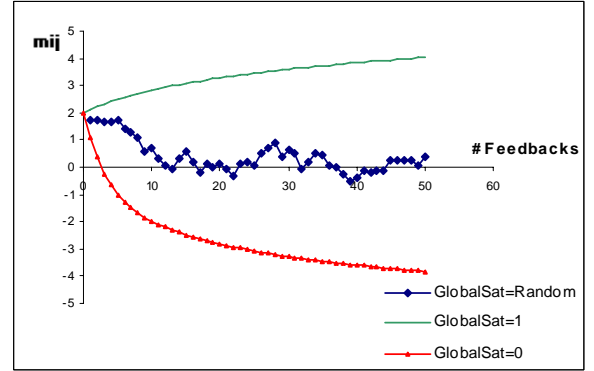


Fig. 4. the variation of the affinity value, m_{ij} , between A_i and A_j when the global satisfaction values are set to 0, 1, and random (0 or 1)

It is worth notice that this learning mechanism is similar to the immune system-based learning mechanism. More precisely, the immune response to eliminate foreign antigens is offered by the entire system in a collective manner. This immune response is called an emergent phenomenon as a result provided from the interaction and the cooperation between these entities that show particular behavior that the individual entities (i.e., B-cells) cannot achieve on their own behavior. In addition, the immune system entities are distributed throughout the body and interact frequently using their own lymphatic network and blood vessels without any central controller. The immune system entities work continually to learn and remember from past and adapt to environment changes. In other words, the immune memory greatly speeds up the response to the pathogen that has been previously encountered. When a new pathogen is encountered, the immune system mounts a primary response during which affinity maturation is used to learn the structure of this pathogen; the primary response can take some time to clear the infection. If the body is re-infected with a previously encountered pathogen, it will have an adapted subpopulation of B-cells to provide a very specific and rapid secondary response.

3.3 Dropping affinity relationships

Using the reinforcement learning mechanism agents can drop relationships to unsuitable agents based on their accumulated satisfaction values generated by emerged services. An affinity relationship between an agent A_i and an agent A_j is considered valid or useless at each step if $m_{ij}(t) \geq -match(s_i, s_j)$, otherwise, it is discarded and could

be removed from the affinity relationship set of the agent A_i if $m_{ij}(t) < -match(s_i, s_j)$. In other words, an agent A_i maintains its relationship to an agent A_j means that the service provided by the agent A_i is available and has similar interest with the agent A_j . In some time, the agent A_j has just some interest by the agent A_i but it has other affinity relationships to other agents having common interests with the agent A_i . In this case, the agent A_j plays just an intermediary role between the agent A_i and other agents sharing similar interests. The affinity relationships can be maintained because of the accumulated global satisfaction values.

Recall that the reinforcement learning mechanism allows agents to select suitable agents or dropping inadequate ones according to its affinity relationships values. The affinity network resulting from this behavior constitutes an organizational memory that permits to the system to adapt to the environment changes, similar to the adaptive memory of the immune system. More precisely, each relationship is characterized by a weight, the affinity value that summarizes history on how the relationship is performed in the past and is adjusted according to its utility. Thus, agents used affinity values to distinguish which of relationships to other agents are more likely to be useful for satisfying its interest. Acquiring new or dropping current agents through establishing or deleting affinity relationship allows the emergence of new services. In other words, the change of the organizational structure of the multi-agent system influences the emerged services but also the future organization structure. These organizational structures are created through interaction between agents according to their affinities (i.e., common interests). Consequently, several services emerge through dynamically adding and removing agents according to the dynamically changing environment.

4. Simulation Result

The agent-based approach presented in this paper is evaluated by a simulator implemented with NS2 [20]. A network of 100 nodes is generated randomly. Each node provides one service described by a randomly keywords form ten kinds of keywords. Figure 5 shows the percentage of agents' satisfaction with and without reinforcement learning. This result shows that without reinforcement learning, each agent has no priori knowledge, accumulated in the affinity values, of services provided by other agents. Consequently, each agent selects one neighboring agent randomly and its satisfaction depends of this random choose. In contrast, with reinforcement learning, at the beginning of the simulation, there is no knowledge about users having common interests. As more simulator time elapses and progress and using the affinity adjustments, agents select and interact with agents having the highest affinity value and a selected service will emerge leading to better satisfaction.

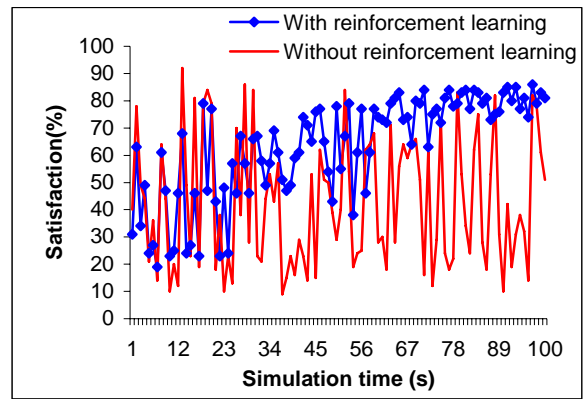


Fig.5. the users' satisfaction during the simulation time in the two cases: with and without the reinforcement learning mechanism

5. Conclusion

In this paper, an agent-based approach to carry out the spontaneous service emergence paradigm in pervasive computing environments is presented. Ad hoc emergent services are created spontaneously on the fly for a user or between groups of users in unpredictable manner; without a priori intention. The learning mechanism is used to adjust the affinity relationships between agents representing the services to cope with the user context and provoke or produce emergent services. In this approach, the mechanism of establishing affinity relationships is based on keywords similarities. Other mechanisms, based on ontologies, can be introduced to increase the rate at which the agents acquire the relationships that meet the desired and required interests. Current work will address the integration of context-awareness parameters in the decision process together with additional simulations.

References

- [1] M. Weiser. "Hot topics: Ubiquitous computing". IEEE Computer, 1996.
- [2] J. Gaber. "New paradigms for ubiquitous and pervasive computing". White paper (Research Report RR-09-00). Université de Technologies de Belfort-Montbéliard (UTBM), France, 2000.
- [3] J. Gaber. "New paradigms for ubiquitous and pervasive applications". Proc. of *First Workshop on Software Engineering Challenges for Ubiquitous Computing*. Lancaster, UK, 2006.
- [4] J. Gaber and M. Bakhouya. "An affinity-driven clustering approach for service discovery and composition for pervasive computing". In *ACS/IEEE International Conference on Pervasive Services ICPS'06*, pp. 277-280, ISBN: 1-4244-0237-9, 2006.
- [5] S. B. Mokhtar, N. Georgantas, V. Issarny. "COCOA: CONversation-based Service COMposition in Pervasive Computing Environments". In *Proceedings of the IEEE International Conference on Pervasive Services (ICPS'06)*, pp. 29-38, ISBN: 1-4244-0237-9, 2006.
- [6] M. Bakhouya, J. Gaber. "Self-organizing Approach for Emergent Multi-agent Structures". In *Workshop on Complexity through Development and Self-organizing*

- Representations* (CODESOAR'06) at GECCO'06. ACM Press, Seattle, Washington, USA, 2006.
- [7] J. Gaber. "Spontaneous Emergence Model for Pervasive Environments", In SUPE workshop, *IEEE GlobeCOM'2007*, to appear.
- [8] M. Bakhouya, J. Gaber. "Approaches for Ubiquitous computing". In *Wireless Ad hoc and Sensor Networks*, Edited by H. Labiod, Hermes Science, ISBN: 9781905209866, to appear 2008.
- [9] T. Itao, T. Nakamura, M. Matsuo, T. Suda, T. Aoyama. "Adaptive Creation of Network Applications in the Jack-in-the-Net Architecture", *IFIP Networking*, pp. 129-140, 2001.
- [10] N. Jerne. "Towards a network theory of the immune system". *Ann. Immunol.*, 125 (Inst. Pasteur) 125-373, 1974.
- [11] E. Guttman. "Service Location Protocol: Automatic Discovery of IP Network Services". *IEEE Internet Computing*, 3(4), 71-80, ISSN: 1089-7801, 1999.
- [12] D. Xu, K. Nahrstedt, D. Wichadakul. "Qos-aware discovery of wide-area distributed services". In *First IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid)*, Brisbane, Australia, pp. 92-99, 2001.
- [13] M. Beek, A. Bucchiarone, S. Gnesi. "A survey on service composition approaches: from industrial standards to formal methods". Technical Report, Bib. Code 2006-TR-15. Available at <http://dienst.isti.cnr.it/>
- [14] K. Sycara, M. Paolucci, A. Ankolekar, and A. Srinivasan. "Automated discovery, interaction and composition of semantic web services". *Journal of Web Semantics*, 1(1):27-46, 2003.
- [15] X. Su and J. Rao. "A Survey of Automated Web Service Composition Methods". In *Proceedings of First International Workshop on Semantic Web Services and Web Process Composition 2004*. Available at <http://www.cs.cmu.edu/~jinghai/>
- [16] C. Abela. "Semantic Web Services Composition". 2003. Available at citeseer.ist.psu.edu/735292.html.
- [17] Z. Maamar, Q. Z. Sheng and B. Benatallah. "On Composite Web Services Provisioning in an Environment of Fixed and Mobile Computing Resources", *Information Technology and Management*, Kluwer Academic Publishers, pp 251-270. Vol 5, No 3-4, 2004.
- [18] M. Bakhouya and J. Gaber. "Service Composition Approaches for Ubiquitous and Pervasive Computing Environments: a Survey". *Advances in Electronic Business*, Vol. 3, Ed. Soe-Tsyr Yuan, Idea Group Pub. 30 pages, (to appear, 2007).
- [19] D. Chakraborty, "Service composition in ad-hoc environments". Available: citeseer.nj.nec.com/529431.html, 1999.
- [20] The NS-2: Network Simulator 2, available at the Information Science Institute.
- [21] C. Peltz. "Web services orchestration and choreography". *Computer*, 36(10), 46-52, ISSN: 0018-9162, 2003.
- [22] M. Hu, "Web Services Composition, Partition, and Quality of Service in Distributed System Integration and Re-engineering". In *XML Conference*, Philadelphia, PA, IDEAlliance. http://www.idealliance.org/papers/dx_xml03/papers/05-05-04/05-05-04.pdf.
- [23] F. Curbera, Y. Goland, J. Klein, F. Leymann, D. Roller, S. Thatte, S. Weerawarana. "Business Process Execution Language for Web Services". Version 1.0, 2002. <http://www-106.ibm.com/developerworks/library/ws-bpel/>.
- [24] R. Hamadi, B. Benatallah. "A Petri-Net-Based Model for Web Service Composition". *Proc. 14th Australasian Database Conf. Database Technologies*, Vol. 17, pp. 191-200, ISSN:1445-1336, 2003.
- [25] S. Narayanan, M. Paolucci, T. Pyne, and K. Sycara. "DAML-S: Web Service Description for the Semantic Web". In the *proceeding of the First International Semantic Web Conference (ISWC'2002)*. Available at <http://www.cs.cmu.edu/~softagents/>

Author Bios

Mohamed Bakhouya received his PhD degree in 2005 in computer science from the University of Technology of Belfort-Montbéliard (UTBM), France. From september 2001 to august 2006 he was lecturer at the Computer Science Department of UTBM. He is currently a postdoctoral researcher at the High Performance Computing Laboratory at the George Washington University. His research interests include distributed and parallel algorithms, mobile computing, ubiquitous and pervasive computing, artificial intelligence, grid computing.