An affinity-driven clustering approach for service discovery and composition for pervasive computing

J. Gaber and M. Bakhouya

Laboratoire Systèmes et Transports (SeT)
Université de Technologie de Belfort-Montbéliard (UTBM)
90010 Belfort, France

Abstract**—**Pervasive computing is a new paradigm with a goal to provide computing and communication services all the time and everywhere. In this paper, a service emergence model for the implementation of pervasive computing applications is presented. In this model, ad hoc or composite services are represented by an organization or group of autonomous agents. Agents establish relationships based on affinities. Affinity corresponds to the adequacy with which two services could bind to create a composed service or to point out a similar service. These affinities are adjusted or reinforced by user satisfaction regarding the provided service and dynamic network condition changes. Simulations of this proposed service emergence model with NS2 are also presented.

Index Terms**—**Pervasive computing, mobile ad hoc network, service discovery and composition, affinity networks, reinforcement learning.

I. INTRODUCTION

The recent evolution of networks connectivity from wired connection to wireless to mobile access together with their crossing has engendered their widespread use with new network-computing challenges. Networks infrastructure are not only continuously growing but their usage is also changing and they are now considered to be the foundation of other new technologies. Since the idea of telematics that was first conceived by Nora, Simon, Minc and Alain in [2] (via [3]) as the "increasing interconnection between computers and telecommunications which blend pictures, sounds and memories and transform the pattern of our culture", new kinds of network-computing paradigm, enabled today by the rapid evolution of communication infrastructures have become possible. Ubiquitous Computing (UC) is an emergent paradigm that deals with providing globally available services in a network by giving users the ability to access services anytime and irrespective to their location. Proposed originally by Mark Weiser in [4], Ubiquitous Computing vision is presented as the opposite of the virtual reality viewpoint "computer in the world" instead of “world in computer”. Pervasive Computing (PC), often considered as the same as ubiquitous computing in the literature, is another concept that can be distinguished from ubiquitous computing in terms of environment conditions. New paradigms that are more appropriate to Ubiquitous and Pervasive environments are presented in [5] as alternatives to the traditional Client/server paradigm. The issue in UC is to provide a mobile device an access to available services in an existing network all the time and everywhere while the main issue in PC is to provide emergent services constructed on the fly by mobiles that interact by ad hoc connections [5]. More precisely, different mobile devises could be connected together by mobile ad hoc networks and they could be available for a limited time. We do not assume that these devices (PDA, mobile phone, etc) have a connection to a fixed network infrastructure. These devices could be heterogeneous and would provide multiples services. Devices can communicate each other in their vicinity to compose their services. Service composition refers to the mechanisms of composing simple services in order to provides complex services. A service is defined as a software component, which accepts appropriate inputs and produces an output. In service composition, several important issues need to be addressed:1) service discovery that permit to know how to locate the constituent services [13,14]. When the user submits its request, eventually with QoS requirements, a service discovery system will process it according to the network status at that instant and finding more than one suitable service, and in order to satisfy the QoS constraints.; 2) service coordination is a processes that permit to link services together to form a composite service after discover them; 3) Service maintenance is a mechanism that detect the service leaving the network and find an alternative service to continue the composite service.

In this work, we address the service discovery and composition in pervasive computing especially when the services are mobile and connected in ad hoc mode. Within this framework, a satisfaction-driven clustering approach inspired by the immune system concept for adaptive service discovery and composition in dynamic and mobile ad hoc networks is presented, as an instance of a propitient system [25]. The immune system 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 [5,13,14].

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

II. Related Work

Recent work on the service composition issue can be found in [1]. Service composition protocols proposed so far concentrate in combining different available existing services. All protocols such as eFlow system [7] are based on a centralized broker which manages the service composition process [3, 4]. The drawback is that if a huge number of users attempt to access variety and increasing number of services distributed over the network, the broker becomes quickly a bottleneck. In mobile ad hoc network where the devises are connected together by ad hoc networks, we cannot assume the presence of centralized and fixed broker [1].

Itao and al. in [6] have defined a platform called Ja-Net for service emergence in large-scale network. A service is implemented by a collection of distributed agents, called cyber-entities. Ja-Net achieves built-in capabilities to create/emerge services adaptively according to user preferences. In this work, we intend to extend Ja-Net approach in the context of mobile ad hoc networks. More precisely, a composed service emergence is based on both the user satisfaction and on the dynamic network condition changes due to ad hoc interactions.

Service discovery is the process of locating which services are available to take part in a service composition. A useful taxonomy of service discovery approach is defined in [14]. Generally, we can classify service discovery into categories: structured and unstructured architectures. Structured architectures can be classified also in indexation-based architectures and hashing-based architectures. In indexation-based architectures, typical resource discovery architectures [15,18], such as Jini [22] and SLP [16] or m-SLP [19], consists of three entities: service providers that create and publish services, services brokers that maintain a repository of published services to support their discovery, and services requesters that search the service broker's repositories. Repositories have traditionally a hierarchical architecture consisting of multiple repositories that synchronize periodically [8]. In a pervasive environment, indexation-based architectures cannot meet the requirements of both scalability and adaptability simultaneously. The way in which they have typically been constructed is often very inflexible due to the risk of bottlenecks and the difficulty of repositories updating.

Hashing-based architectures [10] like Chord [9] are proposed primly to file-sharing and use Distributed Hash Tables (DHTs) to assign responsibility for each file to specific nodes. This structure permits to implement a direct search algorithm to efficiently locate files. Hashing-based architecture use overlay network (i.e., DHT) between nodes that are generally hard to maintain. In particular, nodes join/leave operations could incur huge overheads [11]. In contrast, unstructured architectures (Gnutella [12]) allow the node to enter and leave the systems without overheads. In unstructured systems, the most typical localization method is flooding, where the request is broadcast to all neighbors within a certain radius with TTL mechanism (TTL for Time To Live) [10]. More precisely, in order to find a file on the network, search queries were flooded to peer nodes. Each query had an attached time to live to control the number of hops that a query can be propagated. Each node that passed a query to its peer would decrement the TTL. When the TTL reached 0, the query was no longer forwarded. However, it is not possible to guarantee the success or failure of a query. To overcome this disadvantages, the mechanism of dynamic TTL or expanding ring is proposed in [11]. Principe of expanding ring is as follows: a peer starts a flood with small TTL, and waits to see if the search is successful. According to [10, 11], expanding ring guaranties that if the file is presents in the network it will be found compared with regular flooding with a fixed TTL. However, expanding ring mechanism solves the TTL selection problem, but does not address the message duplication issue inherent in flooding that can generate large loads on the network [10, 11]. Random walk-based search mechanism, is a well-known technique that can avoid this problem. Using one walker, it cuts down the message overhead significantly. However, it increase delay of successful searches [10, 11]. To decrease the delay, a requesting peer sends k query messages, and each query message takes its own random walk. However, it is difficult to determine the number of walks and when this number is big enough, the message traffic increases significantly [10]. The replication mechanisms, such as cache some objects in the reverse path of queries, are proposed in [10] in order to reduce the lookup length and decrease the message traffic. However, in dynamic and distributed setting, it is difficult to maintaining the coherence of duplicated objects.

Other service discovery in pervasive environment, such as DEAPspace [21], UPnP [23] and Konark [17], are proposed and are based on unstructured architectures [20]. In these service discovery systems, service provider does not distribute their service descriptions onto other nodes in the network, but leave them stored on their own device. This leads to a reactive service trading: service requests have to be forwarded to all members of the network where they are compared with the stored descriptions. However, these broadcasting mechanisms are not suited for pervasive environment, either, due to their heavy consumption of bandwidth and energy, which are not unlimitedly available on mobile devices [20].

In this paper, self-organizing technique based of affinity relationships establishment is shown to be the appropriate approach to provide scalable and adaptive service discovery and composition in pervasive environment.

III. Agent-Based Approach

In the proposed approach, to address scalability, nodes can establish relationships between them based on their affinity. To address adaptability issue, affinity relationships between nodes are dynamic; the affinity values can be adjusted at run-time to cope with changes in the environment. To achieve this, we consider the following operations :

- *Joining* operation: permits for an node to find the "appropriate" node to interact with.
- *Leaving* operation: permits for an node to drop autonomously their relationships.

• *Learning* operation: permits to nodes to adapts to dynamic environment conditions through the emergent behavior and affinity relationships to other nodes.

These operations represent an essential and challenging issues in self-organizing nodes that operate in dynamic and open environments. In this proposed approach, these operations permit to carry out the proposed service composition approach that consists of two phases: *building and leaving affinity networks* and *network affinities adjustments*. These two phases of the service composition approach will be described in the next section.

A. Building and leaving affinity networks

To build affinity networks, nodes establish affinity relationships between them based on their provided services. Service can be an elementary service or a composite service. The elementary service is a component that can be interfaced by a web service and is invoked in a composition process. A composite service denotes the list of elementary services requested by the user requests. Each service is represented by an agent Sagent. Sagent establishes a relationship which each other in order to construct the affinity network between them. The affinity network is not an entity; it refers to the composite service. Affinity corresponds to the adequacy which two services to bind. In other words, affinity is a value that corresponds to the adequacy with which two services are similar or complementary. Adequacy could be implemented based on keywords or objects in common describing a capabilities provided by 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 any particular language description to describe services and permit to *Sagents* to compare different services descriptions and to decide how are similar or complementary. For example, services could be interfaced by web services. Web services are applications that permit to describe software components; in particular they define identification and accessing methods that enable the use of these components (i.e., the elementary services). An example of a more sophisticated service description that can be used is WSDL (Web Services Description Language) [24]. For example, a service is described by the following information: identifier, identifier of provider, resource requirement for its execution [24]. More precisely, similar to description presented in [24], there are four main parts to describe a service: 1) the attributes, 2) the Capsule, 3) Constraints and requirements, and 4) Set of relevant semantic keywords. Attributes include the characteristic of the service such as operations that can be invoked and their respective input and output parameters. The capsule gives information about the service localization, the invocation protocol and the port number. The constraints and requirements give information about the required resources needed to execute the service. The semantic keywords are used by a discovery affinity algorithm for matching relevant keywords to each nearby Sagents.

To establish an affinity network, let consider that $D(S_i)$ is a description of the service offered by an Sagent S_i that want to create an affinity relationship with a nearby Sagents S_j. Let us consider also a function $MATCH(D(S_i),D(S_j))$ that return an affinity measure m_{ij} which indicates if the service description of S_i matches with the service description of the agent S_j. For example, m_{ij} can be calculated as the ratio of keywords that are in common between S_i and S_j. If this affinity value is above a certain threshold σ, agent S_i creates an affinity relationship with the agent S_j and S_j creates an affinity relationship with S_i. Initially, m_{ij} and m_{ji} could be set to an equivalent value. Moreover, a reinforcement learning mechanism is used to allow dynamic changes of the affinity values according to user requests. An affinity relationship between S_i and S_j is considered valid if $m_{ij} \geq \sigma$, otherwise, it is discarded and could be removed from the affinity relationship set of S_i (resp. S_j) if $m_{ij} \leq \sigma$ (resp. $m_{ji} \leq \sigma$).

B. Network affinities adjustments

The affinity m_{ij} between a Sagent S_i and a Sagent S_j is a real variable that is adjusted or reinforced based on two level of satisfaction. We refer to the first level by a local satisfaction level and it is described by services offered by neighboring Sagents and resources needed to run services (i.e. computing context). The second level is a global satisfaction described by the user satisfaction (i.e. user context) with respect to end composite service provided. More precisely, let u_i^ℓ (resp. u_j^ℓ) denotes the *local satisfaction* value of the agent S_i (resp. S_j) according to its potential cooperation with a neighboring agent S_j (resp. S_i). If the S_i is satisfied then u_i^ℓ is set to 1 and 0 otherwise. Moreover, the agent adjusts the affinity of relationships using the following equation:

$$\Delta m_{ij}(t+1)=\tau(u_i^\ell.u_j^\ell-g(m_{ij}(t)))$$

where τ is a positive value between 0 and 1. The value of affinity is mapped to a value between 0 and 1 by using the logistic equation $g(m_{ij})=1/1+e^{-m_{ij}}$. Within this equation, the affinity value m_{ij} increases quickly when it is near 0 and satisfaction is equal to 1. Also, the affinity value m_{ij} decreases quickly when the satisfaction is equal to 0. When a user is satisfied with a global provided service, the value of a global satisfaction denoted by u^g is set to 1 (positive satisfaction) otherwise the value of satisfaction is set to 0 (negative satisfaction). This satisfaction is forwarded then to the agents that it interacts with to provide a global composite service according to the following equation:

$$\Delta m_{ij}(t)=\tau(u^{g}-g(m_{ij}(t)))$$

IV. SIMULATION RESULTS

The proposed discovery and composition approach is evaluated by a simulator implemented with NS2 [26]. A network of 100 nodes is generated randomly. Each node provides one service of ten kinds of elementary services that is described by a single of keyword. The simulation abstracts any considerations about networking issues such as bandwidth constraints and time processing. Figure 1 shows the average number of time to discover and compose a service in the case of resource discovery and composition with and without creation of affinity networks. This result shows that without creation of affinity relationships, each node has no knowledge of services provided by other nodes. Consequently, service discovery and composition performs poorly. In contrast, with creation of affinity relationship, at the beginning of the simulation, there are no relationships, and service discovery and composition performs poorly. As more simulator time elapses, nodes create many affinity relationships and adjust them leading to improve performance in service discovery and composition.

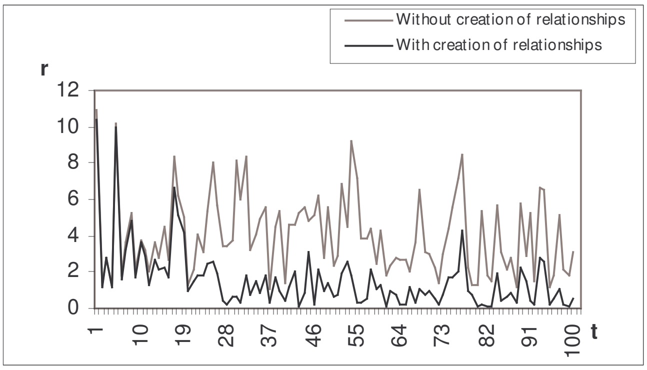

Fig. 1. Average number of time (r) to service discovery and composition with and without creation of affinity relationships during simulation time (t).

V. CONCLUSION

In this paper, a decentralized approach for service discovery and composition for pervasive environment is presented. In this approach, the mechanism of establishing affinity relationships is very simple. Other mechanisms can be introduced to increase the rate at which the nodes acquire the relationships that meet the desired and required services. Current work will address the formal specification issue of the proposed approach, the integration of context-awareness parameters in the equations described above together with additional simulations with ns2.

VI. REFERENCES

[1] D. Chakraborty, "Service composition in ad-hoc environments". Available: citeseer.nj.nec.com/529431.html, 1999.

[2] S. Nora and A. Minc. "The computerization of society", *A report to the president of france*. Transleted by the Massachusetts Institute of Technology. Cambridge, MA:MIT Press, 1980.

[3] M. Perron, "Coaching a society of robots in accomplishing joint tasks". *Crossroads Student ACM Magazine*, Issue 9.1:19–27, Fall, 2002.

[4] M. Weiser, "Hot topics: Ubiquitous computing". *IEEE Computer*, October 1993.

[5] J. Gaber, "New paradigms for ubiquitous and pervasive computing", *Internal research report*, Université de Technologies de Belfort-Montbéliard (UTBM), Septembre 2000."

[6] T. Itao, T. Nakamura, M. Matsuo, T. Suda and T. Aoyama, "Service Emergence based on Relationship among Self-Organizing Entities", *IEEE Symposium on Applications and the Internet*, Available: citeseer.ist.psu.edu/itao02service.html, 2002.

[7] F. Casati, S. Ilnicki, L. Jin, V. Krishnamoorthy and M. Shan, "Adaptive and dynamic service composition in efow". *Technical Report*, HPL-200039, Software Technology Laboratory, Palo Alto, CA, march 2000.

[8] D. Xu, K. Nahrstedt and D. Wichadakul, "Qos-aware discovery of wide-area distributed services". *In First IEEE/ACM CCGrid*, 2001.

[9] I. Stoicay, R. Morrisz, D. Liben-Nowellz, D. R. Kargerz, M. Frans Kaashoekz, F. Dabekz and H. Balakrishnanz, "Chord : A scalable peer-to-peer lookup protocol for internet applications". Available: http ://www.pdos.csail.mit.edu/papers/ton :chord/paperton.pdf, 2001.

[10] C. Wang, B. Li, "Peer-to-peer overlay networks : A survey". Available: http ://comp.uark.edu/ cgwang/Papers/TR-P2P.pdf, 2003.

[11] Q. Lv, P. Cao, E. Cohen, K. Li, S. Shenker, "Search and replication in unstructured peer-to-peer networks", *in Proceedings of 16th ACM ICS'02*, New York, USA, June 2002.

[12] *Gnutella*, Available: http://www.gnutella.com/

[13] M. Bakhouya and J.Gaber, "Approches de mise en œuvre de l'ubiquité numérique", Réseaux mobiles ad hoc et réseaux de capteurs sans fil (Traité IC2, série Réseaux et télécommunications), Eds. H.Labiod, Hermes Science, p. 129-163, ISBN 2-7462-1292-7, 2006.

[14] M. Bakhouya, "Self-adaptive approach based on mobile agent and inspired by human immune system for service discovery in large scale networks", *PhD Thesis*, Université de Technologies de Belfort-Montbéliard, 2005.

[15] M. Robert, "Discovery and Its Discontents : Discovery Protocols for Ubiquitous Computing", *UIUCDCS-R-99-2132*, March 25 2000.

[16] C. Perkins, "Service Location Protocol", *ACTS Mobile Networking Summit/ MMITS Software Radio Workshop*, Rhodes, Greece, June 1998.

[17] S. HelalL, N. Desai, V. Verma and C. Le, "Konark-A Service Discovery and Delivery Protocol for Ad-hoc Networks", *Proceedings of the* WCNC, 2003.

[18] C. Bettstetter and C. Renner, "A Comparison of Service Discovery Protocols and Implementation of the Service Location Protocol", *Proceedings of Sixth EUNICE Open European Summer School*, September 2000.

[19] W. Zhao, H. Schulzrinne and E. Guttman, "mSLP-Mesh-enhanced Service Location Protocol", *ICCCN* 2000, Internet Draft draft-zhao-slp-da-interaction-07.txt.

[20] C. Cho and D. Lee, "Survey of service discovery architectures for mobile ad hoc networks", Term paper, *Mobile Computing*, CEN 5531, Department University of Florida

[21] M. Nidd, "Service discovery in DEAPspace", *IEEE Personal Communications*, pp. 39-45, 2001.

[22] Sun Microsystems, "Jini Technology Core Platform Specification", Available: http://www.sun.com/jini/specs.

[23] The *Universal Plug-n-Play Forum*, Available: http://www.upnp.org.

[24] 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.

[25] M. Bakhouya and J. Gaber, "Self-organizing approach for emergent multi-agent structures", *CODESOAR workshop at GECCO'06*, July 8-12 2006, Washington, USA, ACM Press.

[26] The *NS-2: Network Simulator 2*, available at the Information Science Institute, web site: http://www.isi.edu/nsnam/ns/.