New paradigms for ubiquitous and pervasive applications

J.Gaber
Laboratoire Systèmes et Transports
Université de Technlologie de Belfort-Montbéliard
90010 Belfort cedex, France
Tel : +33 (0)3 84 58 32 52
gaber@utbm.fr

The recent evolution of network connectivity from wired connection to wireless to mobile access together with their crossing has engendered their widespread use with new network-computing challenges. More precisely, network infrastructures are not only continuously growing but their usage is also changing. They are now considered to be the foundation of other new technologies. Related research area concerns ubiquitous and pervasive computing systems and their applications. The design and development of ubiquitous and pervasive applications require new operational models that will permit an efficient use of use resources and services and a reduction of the need for the administration effort typical in client-server networks.

More precisely, to be able to develop and implement ubiquitous and pervasive applications, new ways and techniques for resource and service discovery and composition need to be developed. Indeed, most of research works to date are based on the traditional Client-Server paradigm. This paradigm is impracticable in ubiquitous and pervasive environments and does no meet their related needs and requirements.

Ubiquitous Computing (UC) deals with providing globally available services in a network by giving users the ability to access services and resources all the time and irrespective to their location [Wei93]. An appropriate model has been proposed in [Gab00] as an alternative to the traditional Client/Server paradigm. The fundamental aspect of this model is the process of service discovery and composition. In the traditional *Client to Server* paradigm, it is the user who should initiates 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. However, ubiquitous and pervasive environments have the potential ability to integrate a continuously increasing number of services and resources that can be nomadic mobiles and partially connected. A user can be mobile or partially connected and its ability to use and access services will no longer be limited to those that she/he has currently at hand or those statically located on a set of hosts known a priori. Therefore, the ability to maintain, allocate and access a variety of continuously increasing number of heterogeneous resources and services distributed over a mixed network (i.e., wired, wireless, and mobile network) is difficult to achieve with the traditional Client-Server approaches. More precisely, most Client-Server approaches are based on hierarchical architectures with centralized

repositories used to locate and to access required services. These architectures cannot meet the requirements of 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. This is particularly true for the cases where some services could be disconnected from the network and new ones may join it at any time. The alternative paradigm can be viewed as opposed to the Client/Server model and is denoted as *Server to Client* paradigm (Server/Client). In this model, 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 [Gab00], such a middleware can be inspired from biological systems like the natural immune system (IS). The immune system has a set of organizing principles such as scalability, adaptability and availability that are useful for developing a distributed networking model in highly a dynamic and instable setting. The immune-based approach operates as follows: unlike the classical Client/Server approach,, each user request is considered as an attack launched against the global network. The immune networking middleware reacts like an 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). This immune approach can therefore be considered as the opposed approach to the Client-Server one.

Pervasive Computing (PC) often considered the same as ubiquitous computing in the literature, is a related concept that can be distinguished from ubiquitous computing in terms of environment conditions. We can consider that 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 emergent services constructed on the fly by mobiles that interact by ad hoc connections [Gab00]. The alternative paradigm to the Client/Server one involves the concept of emergence and is called the *Service Emergence* paradigm. This paradigm can be carried out also 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. In this model, ad hoc or composite services are represented by an organization or group of autonomous agents. Agents correspond to the immune system B-cells. Agents establish relationships based on affinities to form groups or communities of agents in order to provide composite services. A community of agents corresponds to the Jerne idiotypic network (in the immune system, B-cells are interconnected by affinity networks) [Jer74].

References

[Wei93] M. Weiser, “Hot topics: Ubiquitous computing”. *IEEE Computer*, October 1993.

[Jer74] N. Jerne, “Towards a network theory of the immune system”. Ann. Immunol. (Inst. Pasteur) 125C373, 1974

[Gab00] J. Gaber, “New paradigms for ubiquitous and pervasive computing”, *research report RR-09*, Université de Technologies de Belfort-Montbéliard (UTBM), Septembre 2000.