Ubiquitous and Pervasive Application Design

M. Bakhouya
The George Washington University, Washington DC, USA

J. Gaber
Université de Technologie de Belfort-Montbéliard, France

INTRODUCTION

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 and they are now considered to be the foundation of other new technologies. A 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 resources and services, and a reduction of the need for the administration effort typical in client-server networks (Gaber, 2000, 2006). More precisely, in ubiquitous and pervasive computing, to be able to develop and implement applications, new ways and techniques for resource and service discovery and composition need to be developed. Service discovery is the process of locating which services are available to take part in a service composition. The service composition process so far concentrates on combining different available existing services as a result of the service discovery process. Most research to date in service discovery and composition is based on the traditional client/server interaction paradigm (CSP). This paradigm is impracticable in ubiquitous and pervasive environments and does not meet their related needs and requirements. Gaber (2000, 2006) has proposed two alternative paradigms to the traditional client/server interaction paradigm to design and implement ubiquitous and pervasive computing applications: the adaptive services/client paradigm (SCP) and the spontaneous service emergence paradigm (SEP).

Bio-inspired approaches are adequate to carry out these new paradigms for designing and implementing ubiquitous and pervasive applications (Gaber, 2000). Indeed, the adaptive servers/client paradigm, considered as the opposite of CSP, could be implemented via a self-adaptive and reactive middleware inspired by a biological system like the natural immune system. The service emergence paradigm could also be implemented by a natural system that involves self-organizing and emergence behaviors (Gaber, 2000).

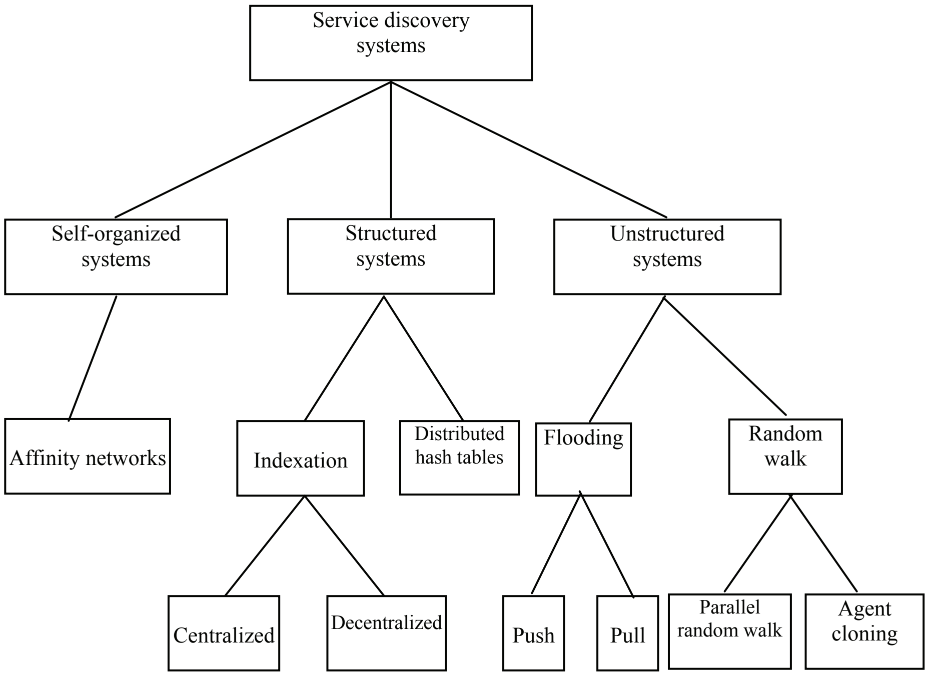
Recently, agent-based approaches, with self-adapting and self-organizing capabilities, have been proposed in Bakhouya (2005) and Bakhouya and Gaber (2001, 2006a, 2006b) to implement SCP and SEP respectively. More precisely, these approaches, inspired by the human immune system, provide scalable and adaptive service discovery and composition systems for ubiquitous and pervasive environments.

UBIQUITOUS COMPUTING

In ubiquitous computing (UC), the objective is to provide users the ability to access services and resources all the time and irrespective to their locations (Weiser, 1993). Service discovery and access systems can be classified into three categories as depicted in Figure 1: structured systems, unstructured systems, and self-organized systems. Structured systems can be classified also in indexation-based architectures and hashing-based architectures. In indexation-based architectures, there are two categories: centralized and decentralized systems. In centralized indexation-based systems, typical resource discovery architectures (Bettstetter & Renner, 2000), such as Jini (2001), consist of three entities: service providers that create and publish services, a broker that maintains a repository of published services to support their discovery, and services requesters that search the service broker's repository. Centralized approaches scale poorly and have a single point of failure. To overcome the scalability problem, decentralized approaches, such as m-SLP (Zhao, Schulzrinne, & Guttman, 2000) or Secure Service Discovery Service (Xu, Nahrstedt, & Wichadakul, 2001), traditionally have a hierarchical architecture consisting of multiple repositories that synchronize periodically. In hashing-based architectures (Wang & Li, 2003), proposed primary to file-sharing, distributed hash tables (DHTs) are used to assign files to specific nodes. This technique allows the implementation of direct search algorithm to efficiently locate files. However, hashing-based architectures require overlay networks between nodes that are generally hard to maintain.

In unstructured systems, the most typical localization mechanisms are flooding and random walk. There are two main flooding techniques: the push and the pull technique. In the first technique, the server advertises periodically its services across the network. The clients receive the service

Figure 1. Classification of service discovery systems according to their architectures and their operating modes



advertisement and cache the information. This information must have a time period associated with it, and must be flushed out from the cache when this time period expires. Hence, the user has a complete knowledge of the available services, and no request resolution process is required. In the pull technique, the client has no knowledge of services present in the network. In this case, a service request is broadcast to all neighbors within a certain radius with a TTL (time to live) tag (Wang & Li, 2003). A random walk is a stochastic process that evolves in the following manner (Gaber & Bakhouya, 2006b). A client sends its query message (i.e., a walker) to a randomly chosen neighbor. At each step, the query message is forwarded to a neighbor of its current location, and the process continues this way by taking random steps that are independent of all the previous ones until meeting the required service. Consequently, the random walk technique avoids message duplication inherent to the flooding mechanism (Wang & Li, 2003). More precisely, by using one walker, it cuts down the message overhead significantly. Nevertheless, the delay for a successful request resolution could be high. To decrease this delay, a requester could send k parallel query messages, and each query message takes its own random walk. However, it is difficult to determine a priori a suitable value for k. In other words, if this number k is big enough, the message traffic could increase considerably. An alternative approach to avoid this problem uses both random walks together with an adaptive cloning agent-based technique for service discovery (Gaber & Bakhouya, 2006b).

It should be noted that the fundamental aspect of these systems is the process of service discovery based on the traditional client to server paradigm. More precisely, 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. This is why the use of repositories is essential in these discovery systems. However, ubiquitous 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 currently has 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 numbers 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 (Gaber, 2000, 2006). More precisely, 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, the difficulty of repositories updating, or the network loading problem. 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.

An appropriate model was proposed originally by Gaber in Gaber (2000) as an alternative to the traditional client/