Chapter XXXVII
Asynchronous Communication Protocol for Multiple Transactions in Mobile Architecture for a Mobile Agent System

Anand Kuppuswami
Dialog Information Technology, Australia

ABSTRACT

Mobile agents and framework built on mobile agents have been the key research area for the past few years. The major impedances like latency factor, abrupt disconnection in service, and minimal processing power, were solved in the mobile agent paradigm. Also, with the advent of intelligent framework of mobile agents, mobile agents were empowered with decision making powers and were able to roam the network in search of the best service provider. This further increased the efficiency of the system and reduced the system outage time. Although the system projected itself as the ideal solution to the real-world problems, it could not be implemented in commercial applications. This is attributed to the lack of sessions in the mobile agent's environment. Predominantly in the mobile agents paradigm architecture was still the client-server architecture. In this chapter the framework has been extended to incorporate transaction capabilities to the mobile agents. This would enable them to perform a full transaction and complete a workflow. We present the scenario of Customer Relationship Management (CRM) where the framework could be put to use.

INTRODUCTION

There has been a tremendous growth in the usage of mobile technologies for the past few years. Previously the mobile technologies were seen as an extended form of Internet and Intranet applications, but with the number of mobile handsets crossing the billionth mark, mobile environment have become an independent area of work. They offer several advantages like dynamic connectivity, smaller gadgets and information processing power irrespective of place and time Gray (1996). Although they seem to be a promising

technology, their certain inherent qualities like low memory and latency factor restricts their usage to many real world applications (Jipping, 2002). Remote Procedure Call (RPC) provided by Java (Birrell, 1984), Network Command Language (NCL) (Meandzija, 1986), Remote Evaluation (REV) (Stamos, 1990) and SUPRA-RPC (Stoyenko, 1994) offered various techniques wherein the inherent short comings of mobile technologies could be bypassed. But all of the approaches lack a crucial feature: Coordination between various application nodes.

The novel approach of transportable programs (Gray, 1995; Cybenko, 1994) offers a promising solution for various issues raised. Transportable agents or Mobile agents, as they are called now, are autonomous programs that can migrate from one machine to another machine in the network. By migrating to the machine having the resource, the agents have the advantage of working on site where the resource is present and also use the processor's power. This eliminates all the middleware that is required for transporting the data to the client's site. Mobile agent's paradigm provides an effective solution to the problem of low latency, poor interface and bad network conditions (Gray, 1996). The middleware and the communication control mechanism form the major workload in client-server architecture and by eliminating them we can build a better working environment and increase the efficiency of the system. This is so because the code as well as the state of code in execution is migrated to another machine for resuming its execution. This also eliminates the interface required for service access. The fact that there is no need for permanent connection makes it very suitable to the mobile environment. The ad hoc client-server model is overridden by the peer-peer model which matures into grid computing, where the machines can act as client or server depending on the environment. The programmer is swayed away from traditional multi-tier architecture to grid computing(Lauvset, 2001). Majority of the mobile agents architecture present in the literature lacked the feature of intelligence embedded into the mobile agents. The system designed by Cabri and Kendall (Cabri, 2002; Kendall, 1998) and Anand (Anand 2005) had agents roaming autonomously and making intelligent decisions. They could navigate and collaborate with other agents with minimal human intervention. They also learn and adapt to the environment. A detailed review of various mobile agents present is presented in the work by Anand (Anand 2005)

The typical architecture of mobile agents service would be a client server one. In the typical Internet based applications, a new request would be generated and submitted to the server through the TCPIP and the server would respond to the request with a valid response. In both the channels data would be transported and requires the channel to be active till the end of response receipt. This would be fine applications requesting small amount of data and applications having very low latency factor. In some cases even though the amount of data transmitted would be low, their might be a huge number of requests to the server. There would be a load balancer which parses the request to the server with the minimum load. This typically reflects the commercial application of large multinational companies. The applications like Enterprise Resource Planning (ERP), Customer Relationship Management (CRM) are some applications which have a low data demand but very high number of requests involved. In such applications, the channel has to remain active till the response is received from the server for each request. This puts an unnecessary load on the network usage and increases the response time and average waiting period. Mobile Agent's technology seems to be the perfect answer to the issues raised.

Although the Mobile Agent's technology offers various advantages, and offers a promising solution, they still cannot be implemented as they do not have the concept of Sessions in them. Most of the commercial applications are based around some workflows which involves multiple requests and responses between the service requester and service consumer. If the nature of communication remains stateless as now, for each request and response a large amount of redundant data would be flowing through. Typical large applications would involve multiple servers which further increases the amount of redundant data. In case of heterogeneous application we would have to avail the service of Asynchronous Message Queues to perform the operation. We present a solution to the various issues raised with the proposal of introducing SessionState into the Mobile Agents framework and show how this could be implemented for CRM application.

SYSTEM ARCHITECTURE

The general architecture of the system is as shown in Figure 1 and the involved steps are described below. The client can be a PC connected to the network or a