

# Middleware Design Framework for Mobile Computing

Imran Sarwar Bajwa

School of Computer Science,  
University of Birmingham,  
Birmingham, B15 2TT, UK

*i.s.bajwa@cs.bham.ac.uk*

**Abstract.** Mobile computing is one of the recent growing fields in the area of wireless networking. The recent standardization efforts accomplished in Web services, with their XML-based formats for registration/discovery, service description, and service access, respectively UDDI, WSDL, and SOAP, certainly represent an interesting first step towards open service composition, which MA supports for mobile computing are expected to integrate within their frameworks soon. A middle-ware that can work even if the network parameters are changed can be a better solution for successful mobile computing. A middle-ware is proposed for handling the entire existing problem in distributed environment. Middleware is about integration and interoperability of applications and services running on heterogeneous computing and communication devices. The services it provides - including identification, authentication, authorization, soft-switching, certification and security - are used in a vast range of global appliances and systems, from smart cards and wireless devices to mobile services and e-Commerce.

**Keywords:** Mobile Computing, Middleware Design

## 1. INTRODUCTION

Distributed operating system assumes homogeneous underlying hardware. In contrast, network operating system does not assume the underlying hardware to be homogeneous. So it is a goal of distributed system design to hide the heterogeneity of the underlying hardware from the applications while using network operating system. While hiding this heterogeneity there are specific issues that need to be considered and resolved. These issues include but not limited to transparency, openness and scalability. In order to design a solution for addressing these issues Research community developed several middle-wares.

Traditional middle-ware solutions are not designed to handle mobile users, their frequent temporary disconnections or the wide range of access devices they employ. Several middle-wares have been designed to provide solutions for such issues but most of these middle-wares are addressing problems and issues for a distributed system in a non mobile context. All these middle-wares try to address some of the issues of distribution. One approach is to extend these middle-wares to support the mobile systems as well but with the rapid development and increase of mobile

entities it has become the research need to design and develop entirely new middle-ware, addressing the specific issues of network operating system in the mobile context.

An important issue in distributed systems is transparency. Transparency is applied to the distributed system for multiple purposes. It is applied to hide location of data and other resources used by a client. It hides the differences in data representations and its accessing techniques from users. It keeps users unaware of migration of resources from one place to another. As in use resources can be relocated hence it's the responsibility of distributed system design to offer transparency for hiding the relocation details. Moreover replication strategies , concurrency , failure and persistence are the parameter that need to be consider while making efforts for transparency in distributed system design.

## **2. PROBLEM STATEMENT**

Applications targeting mobile devices introduce challenging problems to designers. Devices such as mobile telephones, personal digital assistants, and intelligent sensors face temporary and unannounced loss of network connectivity when they move; they discover other hosts during operation; they are likely to have limited resources, such as battery power, CPU speed and memory that require careful management; they are required to react to frequent changes in the environment, such as new location, availability of different devices and variability of network bandwidth.

Middle-ware technologies have the ability to offer a simplified view of the network and the environment to applications, addressing security, heterogeneity, distribution, and resource sharing as well as increasing the dependability of the applications by building applications on a solid, verified base. The aim of this proposal is to investigate how the use of middle-ware has influenced mobile computing, how traditional middle-ware can be employed in mobile computing scenarios, and which characteristics of mobile computing should influence the way middle-ware tailored to mobile setting is designed.

## **3. RELATED WORK**

Wireless networks allow mobile devices to communicate with one another, connect to the Internet or access network services. Advances in wireless networking technologies have provided solutions for local area (LAN) and wide area Network (WAN) coverage. WANs are infrastructure based (devices connect to and roam between fixed base stations) and cover larger areas ranging from whole cities to continents. This section introduces the key technologies currently in widespread use; for a more detailed survey of wireless networks see [1], [2], [3].

However, these are demonstrably inappropriate for the mobile domain. They have heavyweight implementations unsuited to memory-constrained devices [4]. Their operation across unpredictable wireless networks is poor [5][6][7]; for example remote object invocations fail CORBA [8], .NET [9], SOAP [10], DCOM [11], Enterprise Java Beans [12] and Java RMI [13] are examples of established middleware in the fixed network domain.

Mobile agents are now a popular technique to create distributed applications across the Internet and a large number of systems are now available to do this. Examples are AgentSpace [14], Aglets [15], Concordia [16] and Jumping Beans ([www.jumpingbeans.com](http://www.jumpingbeans.com)).

Furthermore, agents do not have to be Java-based, and other solutions are now emerging; the .NET framework [9] contains a Common Language Runtime (CLR) that agents are able to execute upon. For example,  $\mu$ Code is a Java agent solution that has been ported to .Net [17].

#### **4. RESEARCH METHODOLOGY**

Although distribution transparency is generally preferable for any distributed system, there are situations in which an attempt to blindly hide all distribution aspects from users is not always a good idea especially when we talk about a Mobile Computing Systems in which the clients & resources can be mobile. Main problems with middle-wares offering transparency in mobile distributed systems are following:

- They can be heavy weight application in term of code and data.
- Application can have better information required for decision but the applications lacks decision power. The decision power is embedded with middle-ware.

The wide spreading of mobile computing is changing the way to develop, deploy and expect to access Internet services. Nomadic users who disconnect from the network to re-connect to a new point of attachment after a time interval and mobile terminals that continuously roam in the network without suspending the on-going service sessions introduce new challenges in service design.

High heterogeneity, dynamicity and resource shortage/discontinuities typical of mobile computing environments stress the relevance of context-dependent services. However, the design, implementation and deployment of context-dependent mobile applications is significantly more complex than the development of traditional distributed services, thus risking to slow down this emergent service market. Therefore, it is needed for highly flexible and innovative middle-ware solutions, with full context awareness, to facilitate the development and runtime support of context-dependent mobile services.

The research work done in the last years in the area of mobile computing middle-ware has been emerged with the aim of putting side by side the different solutions emerged and of extracting the primary lessons learned. In this overview, I will adopt a specific perspective, i.e. to focus on the innovative middle-ware solutions that choose the Mobile Agent (MA) programming paradigm for their design and implementation. Following are the inhibitors in current middle-ware solutions:

- a. Heterogeneous service environments
- b. Unclear business and value chain roles of the various types of service providers
- c. Lack of service management and administration
- d. Lack of service authoring tools that can leverage common service environment foundations

To provide a smooth mobile service, these inhibitors should be resolved and a robust solution should be provided. To address these inhibitors, I will design the following services and tools.

- a. A service architecture that spans the telecommunication and IT worlds;
- b. A set of relevant enabling services to ease the creation of end-users services;
- c. A set of tools to enable the framework users to easily create services;

In the design of a conventional middle ware, various issues are involved. The proposed middle-ware design will address following core issues:

- a. Service Adaptation
- b. Context awareness
- c. Resource discovery
- d. Service state capture and transfer

Following is the brief account of these design issues of the proposed middle-ware for mobile computing.

#### **4.1. Service adaptation**

Service adaptation deals with effective reuse of components and reuse of web services require adaptation. Often there is no option to build your own version of a web service, hence must adapt existing service.

#### **4.2. Context-awareness:**

Context-awareness is also important in mobile computing. With increased user mobility, and with increased sensing and signal processing capabilities, there is a wider variety of context available to tailor program behavior.

#### **4.3. Resources discovery**

Required and available resources also need to maintain. The resources availability is important and a robust solution should be provided to provide a smooth mobile service.

#### **4.5. Service state capture and transfer**

Middle-ware also deals with capturing and transferring of service states for improved middle-ware actions.

### **5. CONCLUSION**

Dynamic service composition require accepted models and representation formats to describe the interface, the invocation syntax and the semantics associated with available service components. The recent standardization efforts accomplished in Web services, with their XML-based formats for registration/discovery, service description, and service access, respectively UDDI, WSDL, and SOAP, certainly represent an interesting first step towards open service composition, which MA (Mobile Agent) supports for mobile computing, are expected to integrate within their frameworks soon. A middle-ware that can work even if the network parameters are changed and it can be a better solution for successful mobile computing.

## REFERENCES

1. A. Friday, "Infrastructure Support for Adaptive Mobile Applications", Ph.D Thesis, Computing Department, Lancaster University, 1996.
2. Y. Lin and I. Chlamtac, "Wireless and Mobile Network Architectures", John Wiley & Sons, 2001
3. W. Stallings, "Wireless Communications and Networks", Prentice-Hall, 2002.
4. M. Roman, F. Kon and R. Campbell, "Reflective Middleware: From Your Desk to Your Hand", IEEE Distributed Systems Online, 2(5), August 2001.
5. M. Haahr, R. Cunningham and V. Cahill, "Towards a Generic Architecture for Mobile Object-Oriented Applications", Proceedings of SerP 2000: Workshop on Service Portability, San Francisco, December 2000.
6. M. Liljeberg, K. Raatikainen, M. Evans, S. Furnell, K. Maumon, E. Veldkamp, B. Wind and S. Trigila, "Using CORBA to Support Terminal Mobility". Proceedings of TINA '97, 1997.
7. S. Campadello, H. Helin, O. Koskimies and K. Raatikainen, "Wireless Java RMI", Proceedings of The 4th International Enterprise Distributed Object Computing Conference, pp. 114-123, Makuhari, Japan, September 2000
8. Object Management Group, "The common object request broker: Architecture and specification", Tech. Report. Version 2.0, July 1995.
9. Microsoft Corporation, "The .NET Framework", 2010. Available at: <http://www.microsoft.com/net/>, Accessed on Dec 2010.
10. D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Nielsen, S. Thatte and D. Winer, "Simple Object Access Protocol (SOAP) 1.1. Technical Report", <http://www.w3.org/TR/SOAP>, May 2000.
11. Microsoft Corporation. "Distributed Component Object Model Protocol-DCOM/1.0, draft", Available at :<http://www.microsoft.com/Com/resources/comdocs.asp>, November 1996.
12. R. Monson-Haefel, "Enterprise Java Beans", O'Reilly UK (2nd Edition), 2000.
13. Sun Microsystems Corporation, "Java RMI Specification", Available at: <ftp://ftp.javasoft.com/docs/jdk1.1/rmi-spec.pdf>, 1997.
14. A. Silva, M. Mira da Silva and J. Delgado, "Motivation and Requirements for the AgentSpace: A Framework for Developing Agent Programming Systems", Proceedings of the Fourth International Conference on Intelligence in Services and Networks, Cernobbio, Italy, 1997.
15. D. Lange and M. Oshima, "Programming and Deploying Java Mobile Agents with Aglets", Addison-Wesley, 1998.

16. D. Wong, N. Paciorek, T. Walsh, J. DiCeglie, M. Young and B. Peet, "Concordia: An Infrastructure for Collaborating Mobile Agents", Proceedings of the 1st International Workshop on Mobile Agents, Berlin, Germany, April 1997.
17. M. Delamaro and G. Picco, "Mobile Code in .NET: A Porting Experience", in the Proceedings of the 6th International Conference on Mobile Agents, Barcelona, Spain, October 2002.