DISTRIBUTED APPLICATIONS OVER WIRELESS NETWORKS: CASE WHITEBOARD

Ari Valtaoja  
Lappeenranta University of Technology  
P.O.Box 20, 53851 Lappeenranta, Finland  
ari.valtaoja@lut.fi

Juha Jääskeliinen  
Lappeenranta University of Technology  
P.O.Box 20, 53851 Lappeenranta, Finland  
juha.jaaskelainen@lut.fi

Sami Saalasti  
Lappeenranta University of Technology  
P.O.Box 20, 53851 Lappeenranta, Finland  
sami.saalasti@lut.fi

ABSTRACT
Ad-hoc networking is one of the fastest growing areas of information technology. Together with rapidly evolving handheld devices it has created a demand for running distributed applications also on wireless devices. CORBA, the most widely used middleware solution in fixed networks offers an open, platform-independent environment that enables easy development of distributed services. However, CORBA is not necessarily the most ideal solution to the wireless environment because of the assumptions it makes concerning the underlying network medium’s reliability. In this paper we describe these shortcomings and present how the effects of unreliable wireless links can be dismissed with solutions like Wireless CORBA and Implicit Connectivity Management Service (ICMS). We also present a multi-user, CORBA-enabled application called Whiteboard that, being aware of the wireless nature of the network, utilizes the ICMS service to react to the changing network conditions.

KEYWORDS
CORBA, Bluetooth, Ad-hoc, Distributed services

1. INTRODUCTION
The recent development of mobile communications and mobile devices has created a demand for running distributed applications on wireless devices. In the future, users will even expect to be able to use the same services in both stationary and mobile devices. One of these applications is Whiteboard, a collaborative drawing application. It allows a number of users in the same session to share a common drawing canvas.

Currently the most widely used distributed middleware solution is Common Object Request Broker Architecture (CORBA). However, the roots of this technology are deep in the wired networks so it isn’t perfectly suited for the mobile world. To overcome its shortages, the Object Management Group (OMG) has developed a new specification, “Wireless Access and Terminal Mobility in CORBA” that is intended to adapt CORBA into wireless media [1]. The resulting Wireless CORBA can be seen as an evolutionary model of the original specification in so that it tries to completely hide the existence of
wireless network from application developers. This has resulted in somewhat complex and heavy system.

The last solution discussed in this paper, Implicit Connectivity Management Service (ICMS) is a completely new Common Object Service (COS) that is built on the side of traditional CORBA [2]. It is meant to take the responsibility of connection state monitoring and network failure handling off from the application by providing these services itself. Unlike Wireless CORBA, ICMS doesn’t try to hide the existence of possibly unreliable network but to offer means to cope with the situation.

2. DISTRIBUTED SERVICE ARCHITECTURES

In the scope of this paper, distributed service architecture describes an environment where services are deployed into different kinds of wireless or partially wireless networks. In all solutions either CORBA or its extended version, Wireless CORBA along with possible additional component(s) is used. Possible underlying network media are e.g. Bluetooth, Wireless LAN and GSM network via GPRS. The idea is to provide access to services through all these wireless technologies and also enable roaming between them while still using the same service.

2.1 CORBA

CORBA is Object Management Group’s (OMG) open, vendor-independent architecture that enables creation of distributed applications (figure 1). Usage of standard protocols allows CORBA-based program from any vendor, on almost any computer, operating system, programming language, and network to interoperate with a CORBA-based program from another vendor [3]. Originally CORBA was intended for fixed networks where the links are reliable.

![Figure 1. The CORBA architecture](image)

The main strength of CORBA is that network layer is completely hidden from applications so all details related to argument marshalling, connection handling, flow control etc. are taken care by the Object Request Broker (ORB). At the same time however, the underlying network must be reliable because no connection failures are expected. This is also one of CORBA’s main weaknesses when it is used in wireless environment.
2.2 Wireless CORBA

Compared to traditional CORBA, Wireless CORBA has some additional components that enable handoff between network access points and even roaming between different network technologies (figure 2). Of these components, Terminal Bridge (TB) and Access Bridge (AB) form a tunnel that is used to carry GIOP packets over one-hop wireless link using a special GIOP Tunneling Protocol whereas Home Location Agent (HLA) keeps track of terminal’s current location.

![Wireless CORBA architecture](image)

Figure 2. The Wireless CORBA architecture

The whole underlying architecture is transparent for applications, so in theory existing CORBA application could be used as they are. In practice however, the situation is more complex; Wireless CORBA lacks methods to detect possible new access points in the case of connection drop and especially HLA makes the whole architecture quite heavy.

2.3 ICMS

ICMS is a service designed to monitor network activity and search for possible service access points on application’s behalf. It is located on a mobile device like a PDA or eventually a smartphone. Applications can order the ICMS to monitor the connection to some service of interest. ICMS will then inform the application about connection losses and newly founded instances of the used service (figure 3). This way, an application can continue to use the same ORB level logical connection while the underlying physical connection is changed. The physical connection can be made over any supported networking technology.

Although the ICMS is aware of network events like connection losses and new service access points in range it doesn’t make any decisions how this information is used. Instead the final responsibility belongs to the application while ICMS only provides information that the decisions like changing to another access point can be based on.
ICMS is not restricted to one single network medium. Instead, it can be used with different networks like Bluetooth, WLAN and GPRS. It also enables roaming between these technologies while still using the same service. This of course assumes that the same service is available through all supported networks. This way, it’s possible for an application to use the most suitable network based on different criteria like bandwidth, cost, battery usage etc.

2.4 Service Models

Using the abovementioned middleware solutions three main models of service distribution can be defined:

1. Traditional distributed model where pure CORBA without any extensions is used
2. Centralized service model that uses Wireless CORBA as the middleware solution
3. Ad-hoc based solution where services can be located also on mobile devices

2.4.1 Distributed Service Model

Distributed service model defines an architecture where terminal device connects to an access point and uses a service located there (figure 4). In order for CORBA to be usable, the GIOP protocol must be mapped to the underlying transport protocol that could be e.g. TCP/IP or in the case of Bluetooth, L2CAP.

The most fundamental problem of this architecture is that each access point should contain the same service for it to be usable. This is mandatory if the used network has several different technologies in use because CORBA allows only one transport
technology between the connection endpoints. So e.g. in a situation where some service is running in a fixed network using TCP/IP and the network also has Bluetooth access points, it’s impossible for a mobile terminal to use the service through the Bluetooth access points. This leads to a complex and impractical situation where the service is located in each access point and the access points synchronize the service between them if the terminal moves from an access point to another. One way to solve the problem is to run the service in one centralized location and leave nothing but a simple service specific forwarding interface into each access point. Now requests coming to the access points are forwarded to the real server and the same way back (figure 5).

![Figure 5. Distributed service with a central server](image)

Although this method solves the problem concerning transport protocols at some level it doesn’t help to the drawback that CORBA always assumes the underlying network to be reliable, a fact that wireless networks do not conform to. Also terminal’s roaming between different kinds of networks is not solved.

### 2.4.2 Centralized Service Model

Centralized service model differs from the previous one so that it uses Wireless CORBA instead of the traditional CORBA. This architecture, as presented in the figure 6, allows greater flexibility concerning transport protocols and network topology in general. This results directly from the fact that Wireless CORBA allows multiple transport protocols between the connection endpoints.

![Figure 6. Centralized service model utilizing Wireless CORBA](image)

Since this approach is based on Wireless CORBA, each access point has an Access Bridge running on it. This bridge transparently forwards traffic between the mobile terminal and the server, using any transport protocol that the Wireless CORBA in use supports. Compared to the previous model the benefits are obvious: there’s no need to run
the used service on each access point; instead, only one Access Bridge shared between all applications is enough. Secondly Wireless CORBA allows true, transparent roaming between the access points and even changing the used transport protocol is possible. Finally Wireless CORBA is more tolerant for network failures than the original specification.

2.4.3 Ad-hoc Service Model

Ad-hoc service model is implemented using normal CORBA in conjunction with the ICMS service presented in 2.3. It arises from the idea that as well as dedicated servers, also mobile terminals itself could offer some simple services for others to use. The resulting network topology (figure 7) is quite different from the earlier examples since it has almost nothing to do with traditional client-server architectures. Instead, this approach leads into peer-to-peer network.

Because the network doesn’t have any access points some device discovery mechanism is required in order to create it. Bluetooth’s device inquiry offers a way to create a network that has the desired topology. In addition Bluetooth has other characteristics like low battery consumption and relatively high bandwidth that make it suitable for peer-to-peer networking on handheld devices.

![Figure 7. Ad-hoc based service model](image)

This architecture is best suitable for situations where some simple service like Whiteboard is needed in a spontaneous manner. For example, participants of a business meeting might very well need to create an ad-hoc network in order to exchange information during the meeting. In these cases it would be impractical to use some server-based solution since it normally requires configuration of participating devices and no servers are always present. Using ICMS, most of these problems can be solved as described in the following sections.

3. CASE: WHITEBOARD

Whiteboard is collaborative, multi-user enabled drawing application implemented using both CORBA and the ICMS service. Its purpose was to function as a proof-of-concept demonstration to show that ad-hoc based service distribution model can be a viable alternative in certain use cases. Whiteboard offers a predefined set of tools and a common canvas shared between all session participants through a simple user interface (figure 8). Users are able to draw basic shapes like line and rectangle. Also editing as well as deleting
of existing shapes is possible. All shapes are described by a simple set of attributes: starting point, ending point and a color.

![Figure 8. Whiteboard user interface](image)

Because the service is distributed in ad-hoc manner there’s no centralized server to connect to. Instead, a user can connect to any device that is participating a Whiteboard session. Whenever user joins a session, his screen is automatically updated to reflect the current canvas. Now when the user draws a shape this shape is propagated to all participating clients in peer-to-peer manner; in other words, each client sends the newly drawn shape to all clients it is connected to, excluding the originating client.

When the user leaves the session his canvas stays intact so he can continue editing it at will but naturally the changes won’t be delivered to other clients. Rejoining the session is done automatically by the Whiteboard application with the help of ICMS as soon as the user enters the range again.

### 3.1 Implementation

Prototype implementations of both Whiteboard application and the ICMS service were developed for the Compaq iPaq running Linux OS. The used ORB was slightly extended version of MICO [4]. The extension allows Bluetooth to be used as a transport medium with MICO. High-level component structure of the application is in figure 9. The Graphical User Interface (GUI) part handles the user interface. GUI is implemented using the wxWindows framework [5]. Engine contains program logic while Communications part is responsible of data transmission and connection handling. Finally the ICMS is used for connection monitoring and service discovery.
The implemented Whiteboard application supported only Bluetooth because it was considered to be enough for testing purposes; in practice any network medium supported by both ORB and ICMS could be used as well. Outside of this paper’s scope some research was done to enable roaming between different networking technologies while using the Whiteboard application. Although a session where every participating client is connected to only one other client would be enough, the Whiteboard tries to build as extensive network as possible. So when a client receives a shape it tries to connect to the originating client. In the best case this would result in a session where every participant would have a connection to everyone else. This kind of topology would greatly increase network stability because any client could leave without the need for network to reorganize.

Whiteboard uses ICMS for connection monitoring and service discovery purposes. The ICMS is enabled by default although the user can disable it. When enabled it monitors all existing connections and informs the main application if a device goes out of range. When there are no existing connections the ICMS performs service discovery in order to find a viable, Whiteboard enabled device to connect to. When a suitable device is found reconnection takes place automatically without any user intervention. If the ICMS is disabled it’s still possible to use the Whiteboard as long as there are no network failures. However, if the connection fails there’s no way to reconnect until the ICMS is enabled again.

3.2 Results

The main result of the project was a working application capable of creating collaborative drawing sessions utilizing the ICMS service for connection handling and service discovery. For most of the time the implemented Whiteboard application worked flawlessly although some minor problems were encountered. By far the most fundamental problem was that on extremely rare occasions one session could become divided into two separate sessions (figure 10). This happened when other devices lost their connection to the master device of the Bluetooth piconet. When the network was reconfigured as a result of ICMS service discovery it sometimes created a network similar to the right part of the figure 10. However, this problem existed only because the used Bluetooth chips didn’t provide proper scatternet support.
The implemented solution seemed to scale well with the number of clients and no slowing down could be noted. Unfortunately Bluetooth restricted the maximum number of participating clients to eight because of lacking scatternet support. Because of this, no tests with a huge number of clients could be performed.

4. CONCLUSIONS

The developed application shows that for simple services ad-hoc based service distribution using peer-to-peer technique is a viable alternative. If the same service had implemented using Wireless CORBA additional components (TB, AB, HLA) including fixed network access point would have been required. On the other hand, existing applications can work with Wireless CORBA without any modifications while ICMS-based applications need some changes. Finally Wireless CORBA completely hides the underlying network from application which is not the case with the ICMS. However, in the unreliable world of wireless networks this is not necessarily a bad thing because with network awareness the user can be properly notified if needed.

REFERENCES