

Message Middleware In A Wireless Environment

by Peter Rysavy

Published April 15, 1996, Network Computing

Copyright Peter Rysavy and Network Computing

All rights reserved

You decide to take a vacation. You live in Seattle and you want to spend a week in New York City. If you were to call hotels and airlines directly, how many phone calls would it take to find the most convenient flights at the lowest price, and an affordable hotel located where you want to stay? How much time would all this take? A dozen calls? An hour of your time, including hold time? Now imagine calling your travel agent. In a couple of minutes you explain what you want and hang up. Five minutes later, your travel agent calls you back with your itinerary. The benefits that a travel agent brings to your personal travel are analogous to the benefits that message-oriented middleware can bring to wireless applications. Oracle Mobile Agents can be your wireless travel agent.

The Problem Wide area networks promise access to information from anywhere and the ability to work in new and more Wireless productive ways. But they suffer from being relatively expensive to use, compared with dial-up connections, and from being slow, compared with LANs. By slow, we mean both data throughput--typically in the range of 4,800 bps to 19,200 bps (most wireless networks publish rates that are raw throughput; because of overhead from forward error correction and other channel coding, users obtain net throughput that is typically half the quoted rate)--and network delays that increase when wireless modems have to retransmit data due to transmission errors. These network delays can range anywhere from one to six seconds round-trip time, depending on the network and conditions.

While a delay of several seconds may not seem too bad, many applications may send and receive dozens of low-level messages to conduct a single transaction. In our vacation analogy, think of these low-level messages as separate telephone calls you have to make to individual airlines and hotels. A Structured Query Language (SQL) database transaction that takes a fraction of a second to complete over a LAN can take a minute or more to complete over a wireless WAN. Some users may think their application took its own vacation. On top of the delay, users have to pay for all of this back-and-forth traffic.

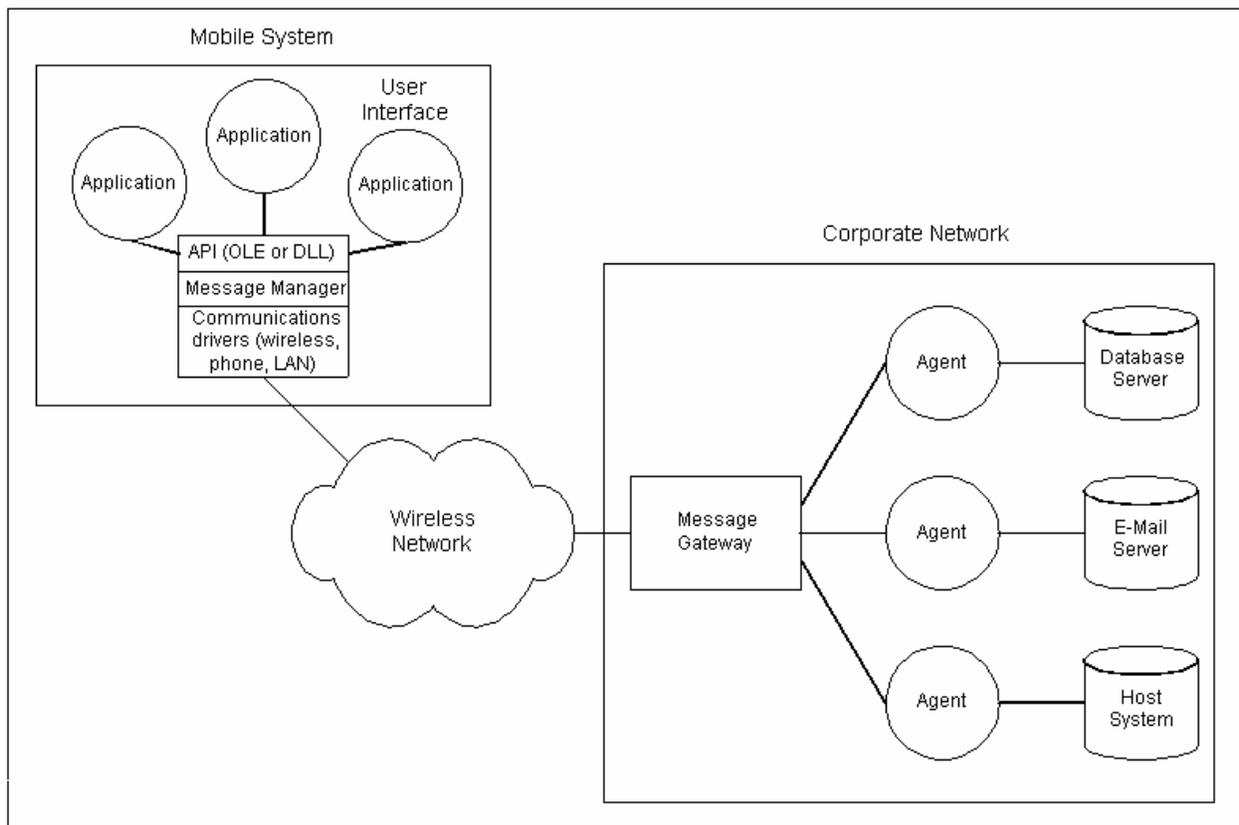
And there are other problems. In a LAN environment, a client application can regularly query servers to check for new information. But over a wireless link, it doesn't make sense for clients to poll servers for new information. It would be far better for the server to initiate messages of interest to clients. There must be a better way.

Enter Message Middleware Message middleware is an architecture where clients can send and receive higher-level Message messages--accomplishing an entire transaction or even a complete set of transactions with multiple servers. These more potent messages are the

equivalent of your call to the travel agent. Moreover, a number of companies have applied message middleware to wireless networks. Oracle's Mobile Agents is one particular example.

Oracle adds to a messaging system a robust communications transport suitable for wireless communications, a high-level application programming interface (API), and agents--programs located on the corporate LAN that act on behalf of mobile clients (travel agents, if you will).

Oracle Mobile Agents consists of a Message Manager on the mobile system, a Message Gateway on the corporate LAN and agents also on the corporate LAN. The application on the mobile system is written to an API that is specific to Oracle Mobile Agents. The servers on the corporate LAN are existing corporate servers, including database servers from Oracle and other companies, mail servers, fax servers, file servers, mainframe or midrange hosts and other back-end systems. See the figure to get a view of its components.



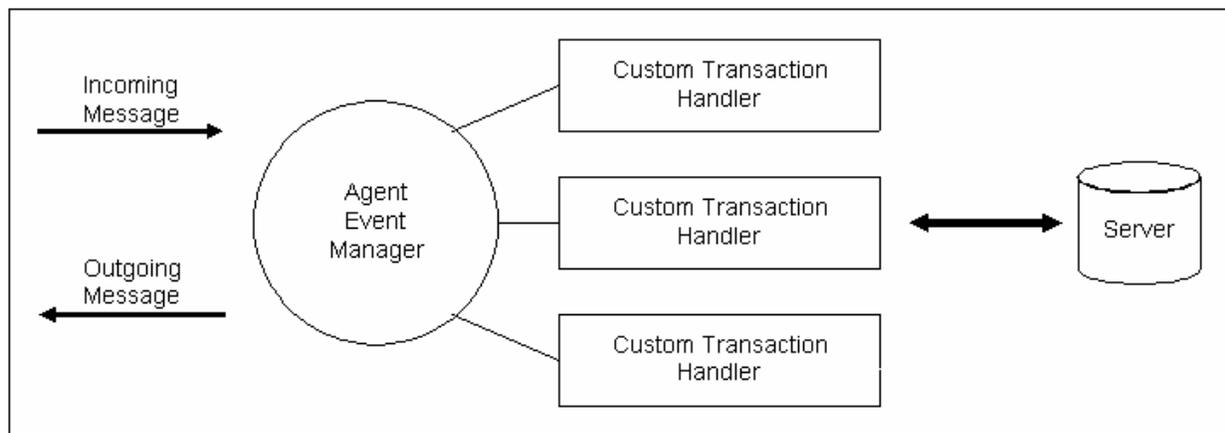
Oracle Mobile Agents: On the mobile system is the Message Manager, and on the corporate LAN are a Message Gateway and agents.

How It All Works The Message Manager on the mobile system sends and on receives messages to and from the Message Gateway on the corporate LAN. This messaging system uses communications drivers, which are part of Oracle Mobile Agents, to reliably communicate across both wireless networks, including CDPD, RAM and ARDIS, and dial-up networks. It can

queue messages when the destination is unavailable, buffering messages on hard disk to protect against power loss. Although messages are not encrypted in the current version, they are compressed.

The Message Gateway relays messages to the agents, keeps track of mobile users and authenticates them. Messages can either be application messages or system messages that configure and keep the system functioning smoothly. As a developer, you do not have to worry about any of the underlying intricacies of the network, including its communications protocols or events such as the user moving out of range from a base station. Oracle's communications drivers takes care of the details.

The agent consists of an event manager and custom-written transaction handlers, as depicted in "The Agent" figure below. The Agent Event Manager sends and receives messages from the Message Gateway and directs these to the custom Transaction Handlers, which are developed as part of an application that uses Oracle Mobile Agents. It is these custom handlers that conduct transactions on behalf of mobile clients. They are the pre-programmed actions of the "travel agent."



The Agent: The agent consists of an event manager and custom-written transaction handlers.

Unlike most client/server-based systems, where the server only passively responds to clients, the agent can initiate messages to mobile clients. Key to Oracle Mobile Agents is that existing servers, such as Oracle databases, do not have to change to accommodate the needs of mobile clients. It is the agent that accommodates the mobile clients' unique needs. You can install the Message Gateway on Windows NT or Unix, agents on Windows 3.x, Windows NT or Unix, and clients on Windows 3.x.

The user interface has two parts: the application and a status window. The user interface for the application is up to the programmer. A governing principle, however, is that Oracle Mobile Agents works best when its messages represent units of human work. An example might be a salesperson filling out an order for a customer who wants a number of different items. The

entire order form constitutes one unit of human work, and the entire order can be conveyed in one message. When the agent receives and processes the message, it will engage a destination server in one or more transactions. Each transaction may involve multiple SQL operations, perhaps one per individual part ordered.

In addition to the application user interface, Oracle Mobile Agents provides a small but handy status window that shows items such as connection status, strength of radio signal when using a wireless network and the number of incoming and outgoing messages.

How do you talk to your travel agent to specify that trip to New York City? The good news is that you can use English; the bad news is that you have to learn a new vocabulary. You can use familiar development tools, but you must use Oracle's application programming interface.

To create an Oracle Mobile Agents application, you develop your application both for the mobile client system and for the agent as a transaction handler. In other words, you have to decide what you are going to say to your travel agent, and then teach the travel agent what it should do when you make specific requests.

Fortunately, Oracle Mobile Agents makes it quite straightforward to develop applications. On the client side, you have a choice of using a Dynamic Linked Library (DLL) or Object Linking and Embedding (OLE) interface for accessing messaging services. This allows you to use a large number of tools to develop applications.

Oracle Mobile Agents facilitates your message creation by supporting predefined data types, such as integers, floating point values, arrays, and structures. Messages are not just bags of bits. Even if your Windows application writes various data types to a message, a Unix message handler will be able to easily make sense of the message contents. This is an important aid when developing across different platforms.

Plan, Plan, Plan First You need to keep in mind a number of points when developing your application using Oracle Mobile Agents. and foremost, you will have to plan very carefully how to distribute the functionality of the application between the mobile client and the agent. You should plan on most of your programming effort going into the custom transaction handler at the agent for a couple of reasons. One is that the transaction handler has to coordinate between both the mobile client and the back-end services. Another reason is that the custom transaction handler must conduct the actual transactions, which, depending on the application, may be complex.

One trick you might consider during development is to have the mobile client application access a local database installed on the mobile system rather than sending messages to the agent. Once you stabilize the mobile application, you can redirect the queries to Oracle's messaging API and bring up the rest of the system.

In designing your application, you will have to take into account the asynchronous nature of this environment. Unlike synchronous systems, such as today's client/server applications where you issue a call and wait for a response before proceeding to the next task, your mobile client, using asynchronous communications, can initiate additional transactions before receiving results from prior transactions. You will need to consider very carefully all the states that your application can enter.

Oracle Mobile Agents has a straightforward API that an experienced Visual Basic programmer or Oracle database programmer should master in less than a week. Implementing a simple test application that exercises all the components of Oracle Mobile Agents should only take two or three days of effort, especially if you use Oracle's sample applications as a starting point.

The Final Word So far, we have emphasized the benefits of solutions based on message middleware. But what are some of the problems? One difficulty is that this is a fundamentally new approach to computing. It has taken more than a decade for the industry to shift from enterprise systems based on hosts and terminals to client and server. Message middleware such as Oracle Mobile Agents builds on client/server, but goes one step beyond. Developers do not have much experience designing, installing and debugging systems based on this architecture, despite how powerful it may be. Such experience will only come with time.

Message middleware is a powerful--and in many cases the best--approach for developing wireless applications. Applications using technology such as Oracle Mobile Agents are reliable, efficient and minimize connection costs. One word of caution: Unless you have prior experience deploying mobile applications, you should consider a pilot project with a small number of users. But if you are planning a trip to the world of mobile computing, message middleware optimized for wireless communications is a route you may want to travel.