Wireless Architecture For Access to Remote Services (WiARS)

Amisha Thakkar
Dept. of Computer Science and Engineering
State University of New York at Buffalo
226 Bell Hall, Buffalo, New York 14260
(716) 645-3180
athakkar@cse.buffalo.edu

Bina Ramamurthy
Dept. of Computer Science and Engineering
State University of New York at Buffalo
226 Bell Hall, Buffalo, New York 14260
(716) 645-3180 ext. 108
bina@cse.buffalo.edu

ABSTRACT
Our research involves the creation of a plug and play environment where different devices and services can be made to function together effortlessly. Currently the connection of any entity to another entity, through a network, needs prior setup and configuration. We aim to design an architecture, which requires minimal setup processes or none, for easy access to remote services using the Wireless Application Protocol (WAP) and Jini.

Categories and Subject Descriptors
C.2.0 [**Computer-Communication Networks**]: General – *data communications,* C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design – *distributed networks, network communications, wireless communications,* C.2.3 [**Computer-Communication Networks**]: Network operations – *network management.*

General Terms
Management, Performance, Design, Experimentation, Languages.

Keywords
Wireless, WAP, Jini, Network, Devices, Services.

1. INTRODUCTION
Consider the scenario in which a commercial organization, which has its activities spread across several locations, wants to provide access to all its employees everywhere. An example would be a corporation that has employees on oilrigs and ships. To be able to provide ubiquitous, instantaneous access from anywhere is a very demanding goal. This goal would certainly help in better enterprise management but becomes a challenge when some of the locations are not easily accessible.

Another interesting scenario is the acquisition of data from unmanned monitoring stations located at inaccessible places. Collecting data from remote places would become very easy if beyond setting up the station no further human intervention were required.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
OOPSLA 2000, Companion Minneapolis, Minnesota
© Copyright ACM 2000 1-58113-307-3/00/10…$5.00.

2. CENTRAL IDEA
Our idea is to integrate WAP and Jini technologies to provide anytime-anywhere access. WAP provides the cross platform wireless environment and Jini provides the plug and play capability. Our design creates a component architecture in which any client can access any service object regardless of time or location. We envisage that the convergence of these two technologies will homogenize the environment yet use heterogeneous entities (devices or services). Simply put, it will mix and match devices, build standard interfaces for users to interact with. The interfaces form an abstraction such that users can effortlessly interact with the system even if the underlying implementation changes. The key interest is to provide the user with access to any device or service from anywhere without having to worry about any kind of configuration.

3. MOTIVATION
Currently when networking facilities are required, complex setup configuration becomes inevitable. For instance, a company whose employees may be working on an oilrig or ship would want to provide them with network access. They would have to setup a wireless system to the closest mainland point and from that point use wired connections to mainframes. Our design allows dynamic access configuration.

It is a challenging task to provide ubiquitous access from anywhere. This architecture provides for portable access, with no cumbersome devices to move around. WiARS is designed such that handheld devices, like personal desktop assistants or cell-phones, can be used. However, this does not restrict utilization of WiARS, with a regular desktop. A desktop can be used just as easily if desired.

Data is a very important component of resource planning. Even data consistency and coherency (that is freshness of data, and that everyone sees the same data - not different versions of the data) matters. So if the database were updated as soon as new data arrives or a change in existing data is tracked, resource

planning would be much more effective. It could be instrumental in administration and decision-making processes. For example, ships that are sent out for exploration of the seabed for minerals could update into the database almost instantly. This would improve the efficiency of schedule management. Consider our other example of an unmanned station, if it were to update data instantaneously predicting an earthquake, it would be very helpful. Our design looks at almost immediate bi-directional data transfer - greatly contributing to swift update, referral and analysis of data.

4. ENABLING TECHNOLOGIES

A large number of technologies are available which could be used but we have chosen WAP and Jini for the reasons explained below.

4.1 Wireless Application Protocol (WAP)

The Wireless Application Protocol is designed to internetwork across several bearers (wireless service providers, who may have differing technologies) and device types. Our objective of plug and play nature of the environment is supported by WAP's standardization (essentially framework-independence). Configuration would be needed if a specific bearer needed to be used or if information had to be conveyed across bearers.

The protocol also makes room for using various types of clients including handheld devices, even those which cannot run Java. WAP is designed to work with devices that have small CPUs, less memory. WAP also accommodates devices with small displays, low power and low bandwidth support. All these features make WAP an attractive option for ubiquitous access.

Also WAP supports a standard language, the Wireless Markup Language (WML), for content presentation. WML enables presentation of semantic information in a structured manner. WML is drawn from Extensible Markup Language. This works to our advantage in WiARS as content representation can be standardized across device types (wired or wireless).

4.2 Jini

The dynamic nature of the Jini architecture, where clients and services can join and leave, goes hand in hand with our vision of plug and play. Also the interface remains standard while the service implementation itself could change which implies reconfigurable service architecture.

The other interesting feature about Jini is that just about any entity can be a Jini service from a switch to a light bulb or even a database. So in our example even if the oilrig is on the Atlantic Ocean, the data from the rig could be updated to just any database-maybe to a database in America or to one in Europe. The client can select the database. Just about any entity could be modeled into a service or could be connected to. Services and clients could be chained if the service desired by the client cannot be reached directly.

4.3 Combining WAP and Jini

We draw on the object-oriented plug and play service capabilities of Jini. But at the same time avoid the using the messaging system of Jini, which would require a thin Java enabled client. The Java requisite for a handheld device could be a limitation. This is done away by WAP. We also make the most of the WAP standardization, so WiARS is not constrained by bearer restrictions.

5. WiARS ARCHITECTURE

The main components of the architecture are the WAP Enabled User Agent (WAEU), the WAP gateway, the Jini Lookup Service and other services.

The transactions in our model would progress as follows:

The WAEU requests a service. The request will be handled by the gateway that the WAEU currently subscribes to. The gateway would act as a Jini client on behalf of the WAEU and lookup a Jini Lookup Service. The Lookup Service returns a list of the currently available services of the type required. The WAP gateway then communicates the list to the WAEU. The WAEU replies to the gateway with the service selected. The gateway then, with the help of the Lookup Service, sets up a connection with the service. Beyond that point, the gateway just relays the messages the WAEU and the service functioning just as an interpreter.

This architecture forms a bi-directional data conveyance model with scope for message initiation from any entity.

6. SCOPE

Our interest in this area focuses on setting up the environment and testing the success of the functionalities incorporated. We aim to create a flexible architecture, which can be easily extended and integrated with other application schemes. Beyond constructing this environment we would also be looking at optimizing transaction latencies. Our research will explore the potential for wireless access to remote entities.

7. REFERENCES

[1] Steve Mann, Programming Applications with the Wireless Application Protocol: The Complete Developer's Guide, Wiley Computer Publishing, John Wiley &Sons, Inc. 1999.

[2] WAP Forum - http://www.wapforum.org.

[3] W. Keith Edwards, Core Jini, Prentice Hall, 1999.

[4] Steve Morgan, Jini to the Rescue, IEEE Spectrum, April 2000.