

WIRELESS TERMINAL MANAGEMENT ARCHITECTURES

R. State

The MADYNES Research Team, INRIA-LORIA, 615 Rue du Jardin Botanique, 54600 Villers-les-Nancy, France, Radu.State@loria.fr

Abstract: The advent of multi-technology networks offering ubiquitous services over advanced wireless network infrastructure demands an integrated management approach. We address in this paper an integrated management approach for wireless and mobile infrastructures by analysing the particularities of this environment and proposing a management architecture based on application based overlay networks. Our approach proposes the integration of a management agent within a service gateway as well as an additional notification support based either on a peer to peer infrastructure or on the session initiation protocol.

Key words: Terminal Management, OSGI, SyncML, Beyond 3G

1. INTRODUCTION

The advent of advanced wireless networks providing high quality multimedia services to mobile users creates an important market for value added services offered on top of these networks. Such services will be typically offered by service providers to end-users. Service providers will interact with different network operators in order to allow service deployment to be done efficiently over a large geographic area. One important problem that service providers are facing is concerned with the interaction with wireless network operators. The difficulty in this interaction lies mainly in service management capabilities offered to the service manager.

We address in this paper the management of future, next generation services. These services will be provided to the end-user seamlessly and in a technology and

infrastructure transparent way. Nowadays, mobile users are bound to network operators and have a limited choice of services to which they can subscribe. Most of these services are dependent on the network operators to provide them. In most cases, the management of new services is limited.

The challenge to provide ubiquitous services to users demands a new way to manage the inherent dynamics.

2. CHALLENGES FOR TERMINAL MANAGEMENT

Although, management architectures have been widely developed for the management of fixed infrastructure network elements, the advent of personalized wireless technology demands the extension of the management towards a new dimension. This new dimension is given by several factors: We have to manage low resource type of equipment needing minimal software overhead for their management.

The connection with these devices will be done wireless, over unreliable links across several management domains. Roaming from within foreign access networks demands access to management agents across one or several domains. Such a situation is rarely encountered in current network management, where the connectivity network to the agent is within the same administrative domain.

The managed entities will not be always online. Due to coverage factors, equipment battery power and its user behavior, management actions should be performed asynchronously.

The network connectivity towards the terminal will be done over heterogeneous links. While current practical management solutions assume an SNMP over IP communication, the range of wireless access protocols (WAP: Wireless Access Protocol), GPRS [1] (General Packet Radio Service, where incoming connections to a terminal are not possible), HTML or even SMS (Short Message Service) demands a versatile communication paradigm to work over these technologies. There are a large variety of terminal related technologies. These multiple technologies are due to the converged usage of multiple access networks envisaged in the beyond 3G proposals where multimode terminals have support for WLAN/GPRS/DVB or subsets of this set, and services are delivered based on the coverage, service accessibility or desired service levels, but also to a personalized service environment where multiple devices cooperate in order to provide a global service to a user.

On a higher level, service and application level management are more important, since in a first phase most new functionalities will be application related. Early attempts have already started with ringing tones that can be downloaded by users and next steps could go more further. For instance, in a car, new software for the injection control could be upgraded. Terminal management would allow easier service management and hence facilitate the introduction of new attractive services.

3. MANAGEMENT ARCHITECTURE

While current terminal software architectures can vary considerably and are mostly vendor specific, we consider a management architecture as described next (see Figure 1), as a common minimal requirement. The environment on which services are dynamically deployed is an Open Service Gateway Initiative [2] framework. This framework allows a service to deliver services to a local network. Services are implemented in Java and encapsulated in software packages called bundles. Although, the initial framework was designed with the target scenario of service providers providing services to residential gateways, the low resources required to run the framework make it a candidate to be use on a wide range of J2ME (Java Micro-edition) devices like portable phones, and to more powerful targets like for instance embedded computers in vehicles or homes.

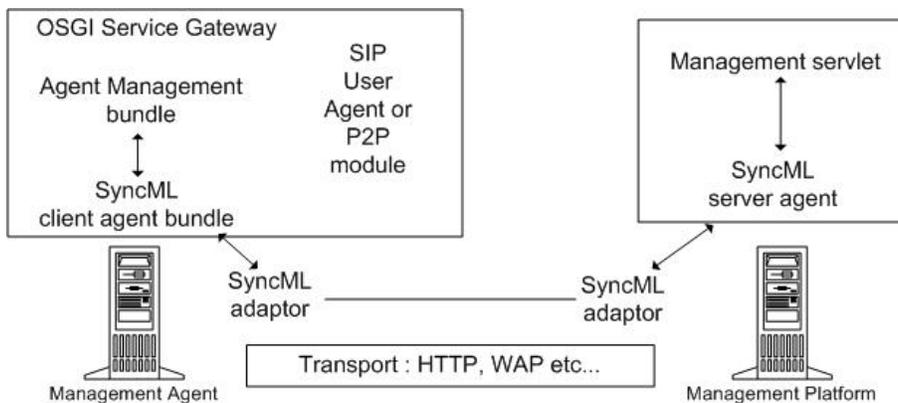


Figure 1. Management architecture

The framework uses a service registry, where bundles can locate needed services, and where each service can be associated to a set of properties. Events are generated whenever a service is registered, unregistered, started, stopped, or when some of its properties are changes. This mechanism allows building a management bundle responsible to manage the rest of the bundles relatively easy. Such a bundle relies on the framework to monitor the life cycles and changes in other bundles properties. It can be also used to change these properties. For our management purposes we require two special bundles to run in the framework. The first one is a management agent responsible for managing the OSGI framework, the second one is a SyncML client agent bundle, responsible for providing a communication stack to the management bundle.

The SyncML language is based on XML, and provides a synchronization protocol, as well as a device management protocol [3] and a set of device level standardized

managed objects. Several characteristics make it a good choice for a management protocol. Firstly it was conceived with small footprint devices in mind and transport over different protocols (IP, HTTP, WAP, Bluetooth, SMTP) is possible. Secondly, most terminals will have a SyncML engine for synchronizing e-mail boxes, calendars etc. Therefore, it is natural to reuse this engine for the management plane. Thirdly, a synchronization protocol is well adapted to communicate with equipment being offline relatively often. If SyncML was initially designed for the configuration of devices, its extension towards OSGI type of service management is relatively straightforward. From a management point of view we need an agent capable of coping with a dynamic MIB.

The communication between the management bundle and a management application comprises a regular datapath for the management information and an additional notification support. The main objective of this notification support is to allow the management application to contact the agent and notify the latter that it must be managed. This notification should be capable to deal with dynamic addresses of the agent and with firewalls blocking management traffic. Two potential solutions for such a notification support are proposed: the first one uses a peer-to-peer infrastructure and the second one is based on SIP.

The Agent Management Bundle is responsible for the management instrumentation of the service gateway. Its design is conceptually similar to the dynamic MIB agent proposed in [5]. This bundle is capable to receive a XML encoded MIB of a bundle, store it in its MIB and perform management operations based on this MIB. This approach is an extension to the standard usage of SyncML where only simple device management is proposed. The extension concerns the usage to service configuration. The agent is responsible to dynamically build the bundle MIB for bundles that are installed or configured otherwise than through the agent.

4. REFERENCES

- [1] C. Andersson. "GPRS and 3G wireless applications". Wiley, 2001 Publishing. 1999.
- [2] OSGI Service Platform Release 2. The Open Service Gateway Initiative www.osgi.org.
- [3] SyncML Device Management Protocol v1.1. SyncML forum. www.syncml.org.
- [4] H. Handley, H. Shulzerinne, E. Schooler and J. Rosenberg. "SIP: session initiation protocol". IETF (Proposed standard) 2543, March 1999.
- [5] A. John, K. Vandween, B. Sugla. "XNAMI –An extensible XML Paradigm for Network and Application Management Instrumentation". Proc 10 th. International workshop on Distributed Systems: Operations and Management DSOM'1999 Zurich, Switzerland