Reconfigurable Equipment Management: an Agent-Based Framework

Karim El-Khazen, Guillaume Vivier, Nicolas Lhuillier
Motorola Labs (CRM) – European Communications Research Lab
Parc des Algorithmes Saint-Aubin, 91193 Gif-sur-Yvette, France
{elkhazen,vivier,lhuillier}@crm.mot.com

Abstract:
Recent trends on wireless technologies as well as on software progress make the reconfiguration of wireless terminals not only an interesting feature but also an achievable one. This paper presents the motivation leading to the need of reconfigurable equipment management. A simple architecture of such equipment is given to allow the reconfiguration and its management. This architecture is based on the use of communicating agents, which provides a way for negotiating the appropriate configuration between the network and the terminal as well as easy context discovery or autonomous decision handling. Experiments based on a prototype shows that this agent-based approach is a good candidate for managing fully reconfigurable systems.

Key Words:
Reconfigurability, equipment management, agent

1. INTRODUCTION

The management of reconfigurable equipments is becoming a topic of great interest in the wireless industry as indicated by the following trends:

- New band allocated for mobile wireless services. A terminal can fully benefit from the new bands as long as it becomes agile.
- Novel application and air interface. Future applications, protocols, waveforms, cannot be anticipated now. Therefore, it is quite impossible to provision at the design time all the features that will be needed in the device terminal. Hence reconfigurable equipments will become able to download and run new functionality. Managing such equipment is of great interest.
- From military to public application of Software Definable Radio (SDR). While SDR was primarily devoted to military applications, it is nowadays moving to the public area, especially in the Beyond 3G trend.
- Beyond 3G trend. The evolution of telecommunication systems is going towards the convergence of different radio access technology towards an IP-based core network. To benefit from these heterogeneous radio systems, devices should be able to communicate through all of the radio access systems and to run all of the possible protocol stacks, applications etc.

This latter vision deserves to be further developed. Actually, telecommunication standard traditionally followed a 10-year regular life cycle. Now, the standards, whether they come from the telecommunication industry such as cellular phone systems or from the computer industry as the WLAN IEEE 802.11x standards, or from the broadcast one, converge towards the delivery of IP content. The B3G trend consists in an open, smooth and flexible integration of these different radio access technology standards to make them complementing instead of competing.

This trend assumes that future terminal will be multi-technology enable. Such ability could be provided efficiently by the reconfiguration of the terminal radio access layer parts, also known as Reconfigurable Radio, inherited from the SDR framework.

To handle the growing need of flexibility brought by the B3G vision and by the Reconfigurable Radio terminals, specific attention should be given to equipment management, a growing activity as it can be witnessed in several standardization bodies such as 3GPP, Telecom Management Forum, SyncML initiative, SDRForum which all followed different approaches.

This paper investigates another approach, amenable and open, based on the use of software conversational agents. Section 2 presents the equipment management aspects within the reconfigurability context, while Section 3 introduces and motivates the use of software agents. Then, the system architecture encompassing end-to-end and equipment architecture is presented in Section 4 and the agent platform JADE-LEAP in Section 5. Finally, lessons from the experimental testbed we developed are given in Section 6. Conclusions about agent-based reconfiguration management are drawn in the last section.

2. EQUIPMENT MANAGEMENT

Managing agile terminals is indeed a greater challenge than managing monolithic systems developed by a single vendor and communicating over proprietary interfaces and protocols. Management and reconfiguration of multi-technology (RATs) and multi-vendor systems, where software components are open and upgradeable, are of great interest to network operators and third-party service providers.

These service/network providers are looking at terminal management in order, for example, to improve their customer support, improve their network performance, and offer enhanced value-added services. The increase of device capabilities and the new expending services are raising the support costs in an exponential way. The advent of terminal management over the air will help the service/network providers.

With this perspective, complete management architecture is needed for such reconfiguration of terminals and network equipments.

3. SOFTWARE AGENT

Software agent technology encountered a renewal over the last decade, due to its foreseen suitability to the Internet ecosystem (e.g. search engines). However, the agent paradigm in itself is rather old and is sometimes dated from 1960. Because of its age, various approaches and uses of software agents have been made, so that it is now quite difficult to give a consensually agreed definition of an agent. However all agents share some common properties, which are of interest to management of reconfigurable equipments:

- Autonomy: agents are supposed to act and behave without the direct intervention of humans. Therefore, they must perform some control over their own actions and internal state;
- Sociability: agents form a community, which communicate and interact through semantically rich messages expressed in a structured language;
- Pro-activity: agents do not simply react in response to their environment; they are able to take initiatives when they consider to be appropriate;
- Adaptability: agents are able to learn on their experiences and improve their behavior accordingly.

Some elements suggest the evaluation of agent technology for terminal reconfiguration: Agent technology has today gained in maturity, as some commercial products include proprietary agents. Additionally, agents have recently moved from the Internet to the wireless device and can be executed on constrained mobile phones [1].

In this perspective, an overall management architecture was proposed, as described in next section.

4. SYSTEM ARCHITECTURE

The figure 1 depicts the overall system architecture, including a reconfigurable device. This device is built around various components. The terminal includes a reconfigurable modem that allows changing some radio parameters such as forward error correction coding rate, data rate, or even waveform type. Software radio modems [2] are typical examples of reconfigurable modems where the new configuration simply consists of changing the execution code. Such modems are appearing, see for instance [3] [4] or [5] but obviously, all the existing radio transceivers are not available today in full software version.

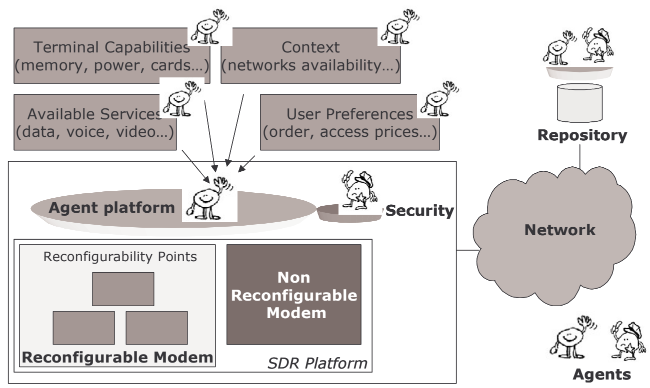

Figure 1: High level management architecture

An agent platform runs on top of the operating system. This platform provides the intelligence and the communication capability for triggering the appropriate configuration actions. It includes the following capabilities:

- Monitoring and discovery of the environment, in order to gather the information related to the available radio access technologies, the capability of the device, in terms of memory, processing power.
- Selection and negotiation of reconfiguration actions. Actually, based on its own context analysis, the device may want to change its own radio parameters but this choice may be invalidated by the radio operator, which has a larger view of the system and other priorities.

This latter capability yields to the need of having a distributed intelligence for managing the reconfiguration. Basically, the reconfiguration involves the equipment as well as supporting functions in the network. In our approach, we have chosen to have a limited network support since we originally wanted to contribute to the larger context of peer-to-peer services. The figure 1 shows a repository where all the known radio parameters and waveforms are stored, controlled by another agent.

This agent provides the basic security functions (download authorizations, subscription restriction information…). On the other hand, this agent could also prevent the user from downloading uncertified waveforms or even virus.

The monitoring operations described earlier could be performed in two ways, either retrieving the information directly or through the use of dedicated agents. For instance, an agent could be responsible for providing the most updated information on the terminal capabilities, and another one for providing the current environment characteristics in terms of networks availability…

The proposed management architecture could be used over any access technology, such as GSM/GPRS, WLAN and even Bluetooth or Ethernet…

5. JADE-LEAP AGENT PLATFORM

JADE (Java Agent DEveloper Framework) [6] is one of the major public implementations of the FIPA standard [8] for interoperable software agents.

JADE, developed by TILAB (formerly CSELT), aims at simplifying the development of multi-agent applications, by providing a runtime to execute and manage life-cycle of agents, as well as monitoring and debugging tools. While appearing as a single entity to the outside world, a JADE agent platform can actually be distributed over several hosts. JADE is implemented in version 1.2 of JAVA; hence agents can migrate or clone themselves to the different hosts of the platform, regardless of the OS. The communication architecture includes transparently to the agents an efficient intra-platform protocol based on RMI and a flexible inter-platform messaging relying on FIPA-compliant Message Transport Protocols (MTP) that are activated at run time. On the other hand, the conformance with the latest FIPA specifications ensures interoperability between agents at the syntax semantic levels.

As an outcome of the LEAP project (Lightweight Extensible Agent Platform, IST-1999-10211, [7]), the JADE platform has been enhanced with a set of libraries, which enable to deploy it over very constrained wireless devices, such as mobile phones, running Java 2 Micro Edition. The resulting platform, often refer to as JADE-LEAP, is the precursor of the second-generation FIPA compliant platforms. It represents a major technical challenge, as it brings the intelligence directly on the end-brings the intelligence directly onto the end-user device. JADE-LEAP benefits both from the development features of JADE and the lightweight properties of LEAP. It runs on a large variety of devices (computers, PDA, mobile phones, etc) and both the wired and wireless environments.

6. PRACTICAL EXPERIMENTATION

We implemented the architecture depicted in Figure 1 on classical Linux-based laptops running JADE-LEAP agent platform and demonstrated basic configuration action such as parameterizations of a WLAN networks, as described in Figure 2.

The intelligence in the terminal consists only in monitoring, filtering and managing the terminal capabilities (memory, power, cards…), the context (available networks…) and the available services (voice, video-conference…). Moreover, the user preferences rules are limited to a preference list, taking into account the requested services and the prices of the access. The repository in the network contains the WLAN connectivity information.

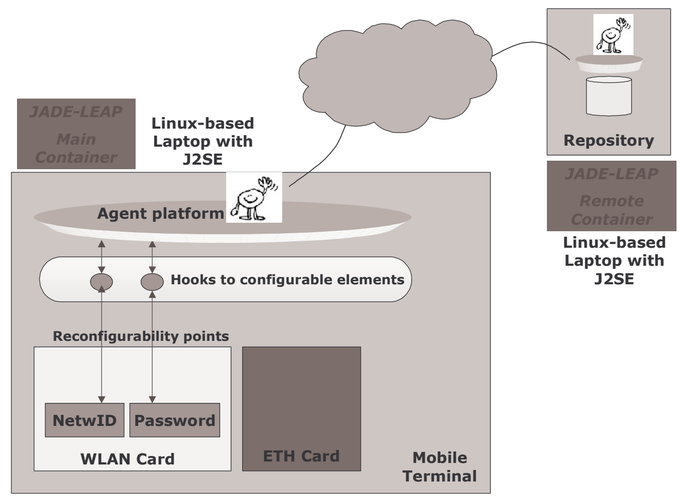

Figure 2: Demonstration Set-up

A JADE-LEAP Agent Platform (AP) is running on the terminal and contains the Terminal Agent (TA). This agent communicates with its corresponding Network Agent (NA) located in the repository. The NA belongs to the unique AP. Since both TA and NA belong to the same platform, they use the JICP protocol (Jade Internal Communication Protocol), the internal platform communications mechanism of JADE-LEAP.

Another solution could be to consider two different agent platforms running in the terminal and in the repository. The first one would contain the TA and the second platform would contain the NA. The difference resides in the communication protocol used between the two agents. The benefit of the proposed architecture with the use of only one agent platform resides in the use of the basic and efficient JICP protocol that reconciles the conflicting requirements of easy processing and reduced data transmission over wireless links.

Both TA and NA are running on Linux-based laptops with the standard version of Java (J2SE). The two agents communicate through either an Ethernet or a Bluetooth connection established between the devices. The mobile edition of Java (J2ME) could have been used instead of

J2SE. Moreover, the demonstration assumes that a WLAN network is available in the area, and the terminal has the possibility to use a WLAN PC Card.

By entering the PC Card in the laptop, the TA detects the WLAN capability since the terminal capabilities are monitored, but no connectivity information is available for the corresponding WLAN network, such as authorized user and password. The TA notifies the NA and requests the WLAN connectivity information. The NA analyses the request and responds to the TA with the needed information for the WLAN connectivity. Following the reception, the TA reconfigures its network configuration by launching a script that uses the received parameters. In this way, the terminal gets access to the WLAN network.

Though it does not include a real negotiation between yhe agents, this simple scenario demonstrates how the management architecture could be used, for instance, for getting access to the WLAN network. In the studied scenario, only parameters were retrieved from the repository in the network. However, pieces of code could be exchanged between the terminal and the network repository and executed locally. For instance, the TA could retrieve the necessary drivers for a new network card, updated waveform software code and install them on the terminal.

7. CONCLUSION

Mobile agent technology is the subject of a lot of attention in the last years, and is often considered as the next step after the client/server and object-oriented programming paradigms.

Even if reconfigurability has been for a long time a topic of interest with outcomes mostly limited to military applications, the convergence between heterogeneous radio systems provides a field of great application to configuration management of flexible equipments.

This developed setup depicts the promising future of the use of software agents in the convergence of the heterogeneous systems.

Future work might consist in addressing more complex scenarios of use, and developing the intelligence aspects as well as modelling the information exchanged between the agents.

REFERENCES

[1] G. Caire, N. Lhuillier, G. Rimassa, “A communication protocol for agents on handheld devices”, AAMAS Conference, Workshop on Ubiquitous Agents on embedded, wearable and mobile devices, Bologna, July 2002

[2] Joseph Mitola III “Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio”, Dissertation Thesis, Royal Institute of Technology (KTH), Sweden.

[3] J. Chapin, V. Bose, “The Vanu Software Radio System”, 2002 Software Defined Radio Technical Conference, San Diego, November 2002, http://www.vanu.com

[4] C. Bonnet, L. Gauthier, P.A. Humblet, R. Knopp, A. Menouni-Hayar, Y. Moret, A. Nordio, D. Nussbaum, M. Wetterwald, “An all-IP Software Radio Architecture under RTLinux”, Special Issue on Software Radio In “Annales des Telecommunications”, Volume 57 N°7-8, July-August 2002

[5] GNU Software Defined Radio, http://www.gnu.org/software/gnuradio

[6] JADE, http://sharon.cselt.it/projects/jade/

[7] LEAP, http://leap.crm-paris.com

[8] FIPA, http://www.fipa.org