MIDDLEWARE AND NETWORKING SUPPORT FOR RE-CONFIGURABLE TERMINALS

DoCoMo Communications Laboratories Europe GmbH, Munich, Germany,
Christian Prehofer (Prehofer@docomolab-euro.com)
Lian Yao (Yao@docomolab-euro.com)
Katsuya Kawamura (Nick@docomolab-euro.com)
Bertrand Souville (Souville@docomolab-euro.com)

ABSTRACT

Software defined radio is one of the key technologies for systems beyond IMT-2000. One of the commercial applications of SDR is to realize seamless roaming between networks with different capabilities in terms of coverage, data rate and services. Network operators have to add new functionalities in their networks that manage and control the re-configuration of SDR terminals to optimize application requirements and network resources.
This paper will give a network operator view of the key requirements to be fulfilled for the design of a network infrastructure for SDR technology. It will also describe distributed network architecture to support the re-configuration of SDR terminals. In this architecture, the novel approach is to clearly separate the network functionalities into a middleware and networking part. The interactions between these components are illustrated by a detailed scenario.

1. INTRODUCTION

Software defined radio technology is one of the key technologies for systems beyond IMT-2000. The research on software defined radio began in the 80s driven by military needs to develop interoperable radios [1]. Within the current industry, the lack of harmonization and the multiplicity of wireless standards have led to the introduction of multi-mode, multi-band terminals to the market.

There is a vast amount of research with regard to architecture and enabling technologies for SDR terminals see e.g. [2]. Many new technologies for adaptable and re-configurable radio have been developed and are expected for the future. Software
re-configuration should cover all layers, including antennas, analog and digital signal processing, networking protocols as well as applications.

In this paper, we address re-configurability support for terminals from the network side. We develop the infrastructure for downloading new software modules in order to change the configuration of SDR terminals. Thus, re-configurability will permit to fix software bugs, improve quality or to introduce enhanced functionalities in SDR terminals.

There has been some work on network support functionalities and architectures needed for terminal re-configurability [3] [4], which shows that a flexible and scalable architecture for software update is quite involved. For instance, one has to support heterogeneous access networks with different capabilities and at the same time allocate the resources for a software update, e.g. for terminal mass upgrades.

In order to fulfill the requirements, we propose an architecture which is clearly structured into a middleware part and into a networking part. The middleware part is designed to be flexible and independent of access technologies, while the networking support is access network specific. This paper focuses on the work split between middleware and networking support, which has not been covered so far.

2. REQUIREMENTS FOR NETWORK SUPPORT FOR RE-CONFIGURATION

We assume that the network operator is in charge of software updates. From a network operator's point of view, the ability of the network to support re-configuration is essential, since there are some advantages of such network-centric re-configuration:

- It is possible to manage the terminal capabilities and available services from operator perspective.
- Operator can install new end-to-end protocols.
- Optimal usage of network resources and radio spectrum.

- Either network or terminal can trigger the re-configuration.

The motivation of network operators for SDR is to provide the best services to the user and also to ensure positive user experience. For this, the software update has to be controlled from the network side in order to optimize application requirements and network resources.

To realize this software provisioning with optimal support for different radio access technologies, network operators have to add new functionalities in their networks to manage and control the re-configuration of SDR terminals. Such approach has been investigated in several articles and projects [4] [5].

There are many different scenarios for terminal re-configuration. For instance, re-configuration can be categorized to two classes depending on the trigger:

- Terminal initiated, e.g. if the user or a software agent determines the need to re-configure.
- Network initiated, e.g. if the operator decides that the terminal shall be re-configured.

There is also a classification regarding the goal of terminal re-configuration:

- Re-configure to enable the hand-over to a different network, e.g. local hotspot LAN.
- Re-configuration to new SW or new mode in the same network, e.g. bug fix, functionalities enhancement.

Other classifications can be done according to the components to be updated or on how the software is deployed.

Although re-configuration support can be used in many different ways and different contexts, we can identify a number of key requirements from a network operator's point of view:

- Secure and reliable distribution of software in order to avoid failures and delays during re-configuration.

- Integration with user profiles, applications and accounting. Since the re-configuration may change the terminal capabilities, it is important to align this with the user profile and related functions like accounting.

- Scalability for a large number of terminals. Typically, distributed software updates support for terminals is needed.

- Support for heterogeneous network infrastructure and terminal diversity. Since current and future access networks have different characteristics, the infrastructure must be flexible and consider different access networks regarding QoS, mobility support etc.

Note that the re-configuration support has to be flexible and independent of the access network. On the other hand, it has to provide optimal support for download resources, mobility support (including hand-over) etc.

3. NETWORK ARCHITECTURE FOR TERMINAL RE-CONFIGURATION

In order to fulfill the requirements discussed above, a flexible and reliable architecture for SDR is needed. For this purpose, we introduce a middleware layer for SDR, which supports interfaces to the applications and other system parts, and appropriate functions on the networking layer.

A distributed functional architecture is depicted in Fig. 1. In particular, we assume distributed functionality in each access network. For middleware, the distribution is supported by most middleware platforms. This gives the scalability which is required to support terminal mass upgrades.

The networking part is split into the core network part and the access network part, which is specific for the access network technology.

The middleware part shall be independent of the access networks and provides flexible services in order to accommodate different re-configuration scenarios and different terminals.
It covers mainly re-configuration management, integration with user profiles, applications and accounting.

The networking support shall take care of the access network specific functionalities. This covers mainly resources for download (QoS) and mobility management (including also vertical hand-over to other networks and location information). In addition, the access network should provide some basic functionalities to control the success of the re-configuration in real time, if this affects networking services.

A detailed overview of the functions of the architecture is presented in the following sections.

4. MIDDLEWARE SUPPORT

In future wireless communications systems, we assume that a middleware layer will decouple the deployment of services from the underlying network infrastructure [6]. This middleware layer will hide the complexity of underlying network technologies and provide well defined interfaces to application service providers.

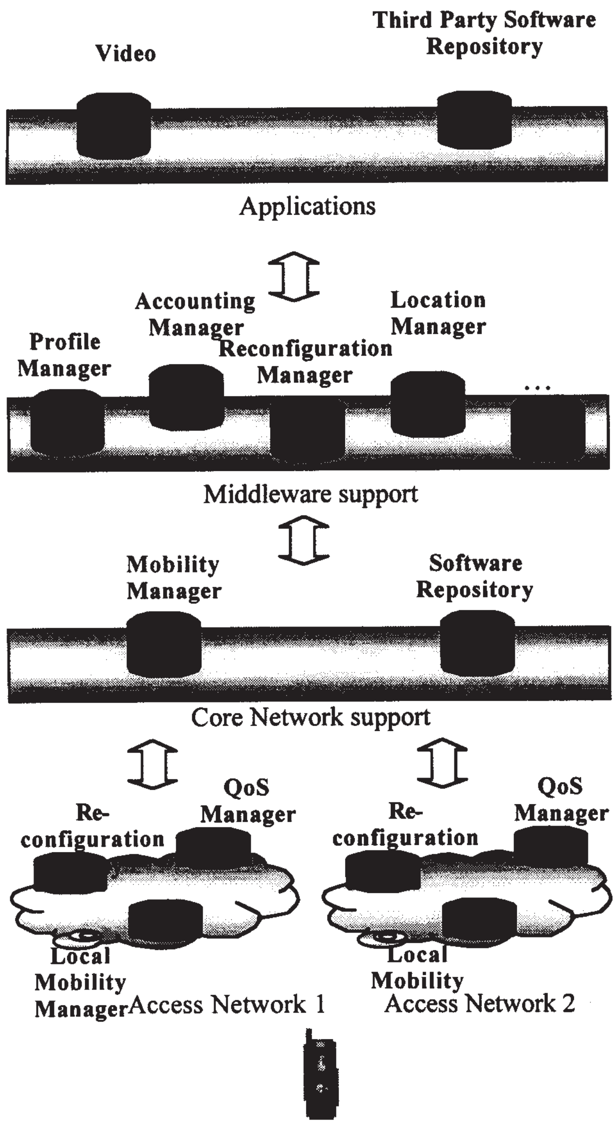

Fig.1 Distributed Architecture for Software Update

Middleware support to re-configurable terminals has already been identified as a research topic [7] [8].

The main design goals of the middleware support are the following:

- Abstraction from network topology / terminal diversity.
- Independent of access networks.
- Flexibility to support new kinds of services, irrespective of the evolution of network or terminal technologies.
- Easy support with applications and other system parts by the definition of well defined interfaces.
- Support to secure distribution of software.

The responsibilities of defined middleware components in the architecture are now explained.

Re-configuration Manager: It has to locate the required software components stored in software repositories in the different parts of the networks. It has to authenticate users, and software repositories. It has to select possible target access network technologies taking into account the user profile/preferences, the terminal capabilities, and the application profiles. It interacts with the Profile Manager, Location Manager, Accounting Manager and networking layer components.

Location Manager: It controls the mobility and location of users. With the location information of the user, available networks can be detected [9]. It interacts with the networking layer, responsible for providing the current location of the user.

Accounting Manager: Accounting is an important characteristic for network operators. The role of accounting is to compute the charges to be paid by users for specific services based on billing models for these services. Software updates of re-configurable terminals can be one of the services provided by network operators. The charges may be distributed between different operators and third party service providers. The middleware has to support such complex interactions between the different accounting systems owned by different network operators.

Profile Manager: The Profile Manager interacts with Re-configuration Manager to exchange information such as subscription data and application profiles.

5. NETWORKING SUPPORT

As stated in the previous sections, current and future access networks have different characteristics, thus a flexible and scalable architecture is necessary. Together with the functional components introduced before in the middleware layer, we introduce some functional components allocated in the networking layer to fulfill the requirements of the networking support for re-configuration.

The main design goals of the networking support are:

- Distributed functionality for scalability.
- Optimal re-configuration support considering the access network capabilities.
- Stable and robust network services.
- Resource and mobility aware, depending on access network characteristics (network specific).
- Real-time support.

In order to achieve these specific goals of networking support, the following components in the networking layer take the responsibilities of networking support.

Re-configuration Proxy: It is located in the access network and acts as a software repository cache. It collects and keeps

all the software that pass through it, when a terminal asks for software, the Re-configuration Proxy checks if this is stored in it. If so, it can check to see if this software is the newest version. Then it passes the requested software to the terminal. Otherwise it will contact the remote software repository based on the information from the Re-configuration Manager in the middleware layer.
Since every Re-configuration Proxy is located in the access network, the localization reduces the time of software updates and decreases the load of the wireless link.
In addition, the Re-configuration Proxy also has some control and management functionalities of the re-configuration. For instance, to manage fallback, i.e. when a failure of new re-configuration happens, the Re-configuration Proxy should be able to interact with the terminal if connectivity is possible (i.e not for lower layer re-configuration) and make it possible to keep the old configuration. It should also be able to inform the terminal that the new re-configuration is successful as well, when the new configuration functionality is correct.

Local Mobility Manager: It is in charge of the hand-over of the mobile terminal within the access network. It knows the location of the terminal in the network. Note that here the location of the terminal means the location of the terminal within the logical structure of the network, e.g. in the UMTS network, the logical structure is the cell and area composed of groups of cells [10].

QoS Manager: It takes care of the requested QoS parameters, and monitors the status of the software download to ensure the quality of download session [11].

All the above mentioned functional components are access network specific, which means that they have the same interfaces to interact with the middleware functional components, but they are adapted to different radio access technologies to take care of the download resources and mobility support. For instance, WLAN offers high data rates for data communication, but it does not intend to support frequent hand-over between different access points, and it has no QoS guarantee. In this case, only "soft" guarantee is possible for the QoS in the re-configuration process.

In order to support the hand-over between different network domains, besides the Local Mobility Manager allocated in the access network, there is also Mobility Manager functionality within the core network. This mobility functional component interacts with the Local Mobility Manager in every different access network, and communicates with Location Manager in the middleware. It also can support vertical hand-over, i.e. a hand-over between different radio access technologies.

6. INTERACTION SCENARIOS

In this section we describe an interaction scenario to obtain a better understanding of each component.

User Initiated Class

A user on the street is talking to his colleague in the office over GSM phone. He is asked to join the videoconference to explain his project. He finds a cafÈ, which offers wireless services. He uses a service to determine which networks to use on the spot. He sends application information (videoconference) to the network and his mobile terminal shows W-LAN as preferred network. His mobile terminal is currently available for GSM service. For a videoconference, he confirms W-LAN network as recommended from the network. Soon the mobile terminal shows that now W-LAN network is supported. He connects this mobile terminal to his PC and set up videoconference with his colleague.

In this scenario, the following interactions are assumed between mobile terminal and network, and between functions within the network.

(1) A user specifies the application which he wants to use to the mobile terminal
(2) The mobile terminal sends application information and terminal capability to the Re-configuration Manager
(3) Re-configuration Manager obtains subscription data, preference/priority data and application profile from the Profile Manager
(4) Re-configuration Manager requests available networks from Location Manager
(5) Location Manager and Mobility Manager interact and detect available networks
(6) Location Manager returns the available networks to Re-configuration Manager
(7) The available networks are checked according to the subscription data, preference/priority data, application profiles and terminal capability at Re-configuration Manager and preferable network is selected
(8) The selected preferable network is notified to the mobile terminal together with the related information (e.g. price difference, estimated re-configuration time, etc.)
(9) The mobile terminal displays the necessary information (e.g. necessity of re-configuration, price information etc.) and waits for user action
(10) User confirms the re-configuration of the mobile terminal
    ˚0 If the user does not like the recommended re-configuration, the mobile terminal sends the message to request other availabilities from the Re-configuration Manager
(11) If the selected network is not supported on the mobile terminal, it requests software download from Re-

configuration Proxy by sending necessary software information

(12) Re-configuration Proxy starts download

(12)' In case the requested software is not on the Re-configuration Proxy, the following process occurs.

- ˚0 The Proxy asks the location of the software to Re-configuration Manager.
- ˚0 The Manager looks up Software Repository and returns the information to the server
- ˚0 The Proxy gets the software from the Software Repository
- ˚0 The Proxy starts download

(13) When software download process starts, Re-configuration Proxy notifies to the Re-configuration Manager and the Manager notify to Accounting Manager for charging

(14) When download completes, the mobile terminal notifies to Re-configuration Proxy and the Proxy notifies to Re-configuration Manager, and the Manager notifies to Accounting Manager

(15) The mobile terminal verifies the downloaded software and performs the terminal re-configuration

(16) During the download process, QoS Manager ensure the quality of download session to support reliability

(17) The mobile terminal re-configures with the new software and registers to the network

(18) In case of failures, the mobile terminal or the network decide on fallback

<u>Network Initiated Class</u>

A user usually uses 3G network especially for video clipping services to check the news highlight. He is on business trip in a rural area where 3G network is not available but GSM is. When he is on the business trip, his mobile terminal is configured for GSM networks automatically. He receives news there and he read by text that available media on GSM. On the way to the city after his work, when he is reaching to 3G network service area his mobile terminal is automatically re-configured to 3G network. He received another news and now he can watch the news highlight with video contents.

In this scenario,
network detects the context of the mobile terminal automatically and initiates re-configuration process. In this case, re-configuration is preceded without user confirmation (instead of the user confirmation in the user initiation case, it is done by user preference/priority data).

7. CONCLUSION

We have presented a new architecture for managing terminal re-configurations which is clearly structured into a middleware part and into a networking part. The middleware part is designed to be flexible and independent of access technologies, while the networking support is access network specific. This structure is important to fulfill the requirements of flexibility with respect to different terminals and different scenarios, and at the same time to support different access networks efficiently. In order to support our new architecture, we have described the responsibilities of the architecture components and shown their cooperation in a detailed scenario.

8. ACKNOWLEDGEMENT

This work includes results from the IST project IST-1999-12070 TRUST. Ongoing work is undertaken in the framework of the IST project IST-2001-34091 SCOUT, which is partly funded by the European Union. The authors would like to acknowledge the contributions of their colleagues, although the views expressed are those of the authors and do not necessarily represent the project.

9. REFERENCES

[1] R.J. Lackey, and D.W. Upmal, "Speakeasy: The military software radio",
IEEE Communications Magazine, Volume 33, pp. 56-61, May 1995.

[2] W.Tuttlebee,
Software Defined Radio, Enabling Technologies, Wiley & Sons, 2002.

[3] M. Dillinger, R. Becher, "Decentralized software distribution for SDR terminals", *IEEE Wireless Communications*, Volume 9, pp. 20-25, April 2002.

[4] K. Moessner, S. Gultchev, R.Tafazolli, "Software defined radio reconfiguration management", *Personal, Indoor and Mobile Radio Communications*, Volume 1, pp. 91-95, Sept. 2001.

[5] N.J. Drew, and M. Dillinger, "Evolution toward reconfigurable user equipment", *IEEE Communications Magazine*, Volume 39, pp 158-164, Feb. 2001.

[6] H. Yumiba, K. Imai, M. Yabusaki, "IP-based IMT network platform", *IEEE Communications Magazine*, Volume 8, pp. 18-23, October 2001.

[7] A. Munro, "Mobile middleware for the reconfigurable software radio", *IEEE Communications Magazine*, Volume 38, pp. 152-161, August 2000.

[8] N. Jefferies, J. Irvine, A. Munro, K. Moessner, "Managing increasingly software dependent mobile systems", *3G Mobile Communication Technologies*, Second International Conference on (Conf. Publ. No. 477),
pp. 247-251, 2001.

[9] G. Clemo, T. Farnham, "The use of service discovery to support reconfigurable terminals",
Vehicular Technology Conference, pp. 2465–2469, 2001.

[10] H. Kaaranen et al, "*UMTS Networks – Architecture, Mobility and Services*", John Wiley & Sons Ltd., 2001.

[11] D. Bourse, M. Dillinger, T. Farnham, R. Navarro, N. Olaziregi, T. Wiebke, "SDR Equipment in Future Mobile Networks", *IST Summit*, pp. 189-193, 2002.