Computing System Modeling for Flexible Radios

V. Marojevic, J. Salazar, I. Gomez, A. Gelonch
Universitat Politècnica de Catalunya
Dep. of Signal Theory and Communications, C/Jordi Girona 1-3, 08034 Barcelona, Spain
{marojevic|jose.salazar|ismagom|antoni}@tsc.upc.edu

***Abstract*—This paper presents a computing system modeling for flexible radios based on software-defined radio. It facilitates to modularly model and dynamically update computing resources and application characteristics, including computing requirements and service related features. We discuss the applicability of this modeling as the basis for trading off radio against computing resources (flexible radios).**

***Index terms*—cognitive radio, computing resources, flexible radios, modeling, software-defined radio (SDR)**

I. INTRODUCTION

The evolution of wireless communications indicates a trend towards flexible radio implementations. Beyond 3G systems (B3G) represent heterogeneous radio environments where different radio access technologies (RATs) coexist. Long term evolution (LTE) describes the improvements of these RATs and system architecture evolution (SAE) characterizes future network architectures. The steadily growing demand on personalized multimedia service has implication on the entire system design, from hardware platforms to software applications.

Software-defined radio (SDR) provides a framework for the software-reconfiguration of radio equipment (mobile terminals and network elements) to operate in different transmission and reception modes, or use different air interfaces, at different times [1]. This concept, though mostly applied to the baseband processing, is applicable to the entire radio system [2].

The SDR platform, a distributed computing platform of limited resource, needs to provide sufficient computing power to meet the real-time computing requirements of software-defined RAT implementations (SDR applications) and user applications. SDR and user applications should implicitly account for service and Quality of Service (QoS) constraints. Once mapped to the computing resources of an SDR platform, these applications define the platform's radio functionality and service presentation to the wireless user.

Cognitive radio has been introduced as the evolution of software-defined radio [3], although it does not necessarily require the SDR context [4] (Fig. 1). It introduces cognition and intelligence to radio communications: A cognitive radio system can autonomously make reconfiguration decisions to optimize the usage of radio and computing resources, increasing the system's capacity [5].

Flexible radios facilitate trading off radio for computing resources and vice versa. We consider it a concept on top of SDR and cognitive radio: Flexible radios facilitate trading off radio against computing resource, while operating in ambient-aware, intelligent, and software-reconfigurable systems (Fig. 1).

Fig. 2 shows the radio, hardware, and application environments and some of their principal characteristics. A suitable computing system modeling is necessary for being able to trade resources from one environment for those of another. This paper introduces such a modeling. It consists of the computing resource modeling, which describes the available computing resources (section II), and the application resource modeling, capturing the SDR and user application requirements and features (section III). We introduce general modeling templates that facilitate adding new computing and application models to account for additional resources as necessary. We discuss several relations between resources from different environments and indicate two general scenarios for flexible radios (section IV). In future work we will simulate these scenarios to quantify the possible advantages of flexible radios.

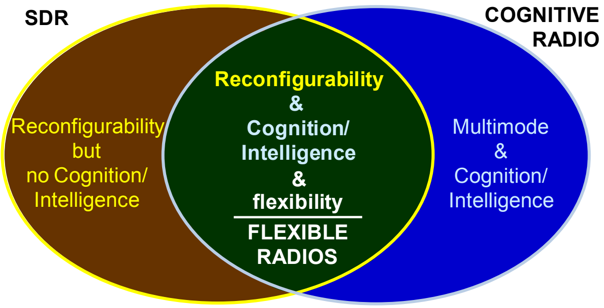
SDR
COGNITIVE RADIO
Reconfigurability but no Cognition/ Intelligence
Reconfigurability & Cognition/ Intelligence & flexibility
FLEXIBLE RADIOS
Multimode & Cognition/ Intelligence

Fig. 1 SDR, cognitive radio, and flexible radios.

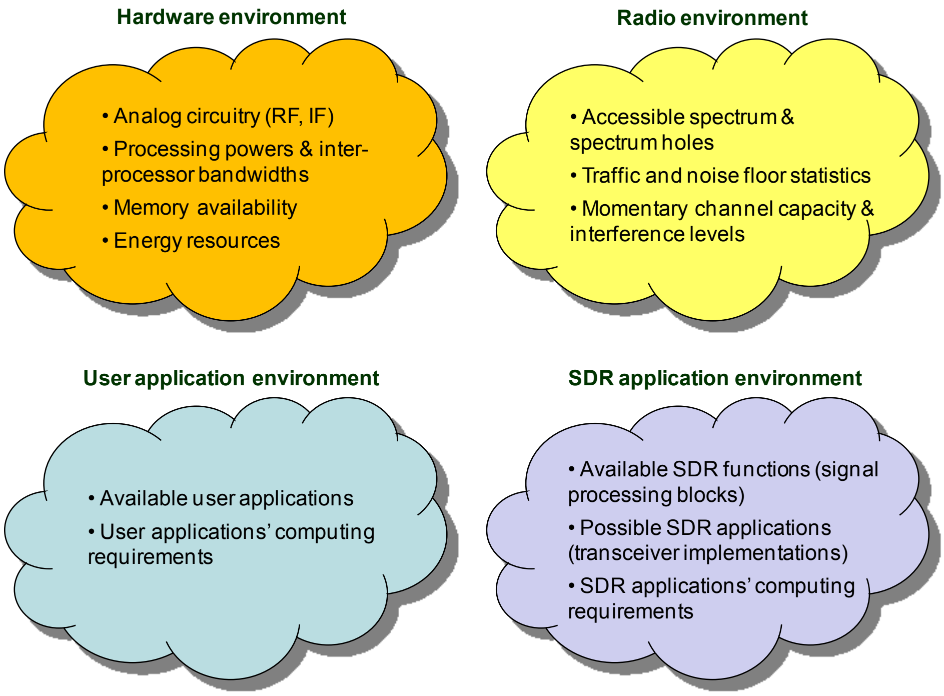
Hardware environment
• Analog circuitry (RF, IF)
• Processing powers & inter-processor bandwidths
• Memory availability
• Energy resources
Radio environment
• Accessible spectrum & spectrum holes
• Traffic and noise floor statistics
• Momentary channel capacity & interference levels
User application environment
• Available user applications
• User applications' computing requirements
SDR application environment
• Available SDR functions (signal processing blocks)
• Possible SDR applications (transceiver implementations)
• SDR applications' computing requirements

Fig. 2 Radio, hardware, and application environments and their principal characteristics.

II. Computing Resource Modeling

A. Computing Environment

An SDR platform, which characterizes either a mobile terminal or a network element, consists of general-purpose and software-reconfigurable processing entities, such as digital-signal processors (DSPs), microprocessors (µP) and field-programmable gate arrays (FPGAs). Fig. 3 shows an example SDR platform.

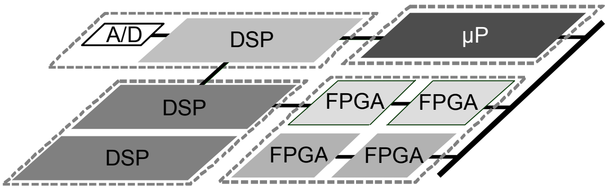

Fig. 3 Example SDR platform.

Any processing platform has a limited amount of computing resources. These include processing capacities, inter-processor communication bandwidths and energy resources. (Although network elements may have practically unlimited energy resources, optimizing their power consumption would increase their efficiency.) Management of computing resources is necessary for switching from one radio standard or service to another, or, in other words, for mapping another SDR or user application to a given SDR platform. An SDR platform is characterized by the available and occupied computing resources for radio and non-radio specific computation. The rest of this section presents the general computing resource model and several instances of it.

B. General Computing Resource Model

$\boldsymbol{R}^{(T,\,D)}$ (1) is a $U(T, D)$-by-$V(T, D)$ matrix of nonnegative real elements ($\mathbf{R}_{+}$) that models the available computing resources of type T for SDR platform D (T, D, $U(T, D)$, $V(T, D) \in \mathbf{N}$, where $\mathbf{N}$ stands for natural numbers).

Equation (1) is a template for modeling different types of computing resources and therefore without unit. In continuation we demonstrate its suitability for modeling several types of computing resources.

C. Computing Resource Model Instances

We identify 4 relevant computing resource types. These are the processing, bandwidth, memory, and power resources. We model them as instances of (1).

Equation (2) captures the processing powers of processors P_1, P_2, …, $P_{N(D)}$ of SDR platform D. Without loss of generality, we number processors in order of decreasing processing capacities; that is $(C_1)^D \geq (C_2)^D \geq (C_3)^D \geq \ldots \geq (C_{N(D)})^D$. SDR platforms generally consist of heterogeneous processing devices. Therefore we assume a hardware abstraction layer, such as PHAL [6], that provides a pseudo-homogeneous processing environment, where processing capacities can be defined in MOPS (million operations per second) [7].

Equation (3) describes a platform's bandwidth capacities in MBPS (mega-bits per second). $(B_{32})^D$, for instance, is the bandwidth capacity that is available for the directed data transfer from the local data memory of processor P_3 to the local data memory of processor P_2 of SDR platform D. We assume direct memory access (DMA) or pointer transfers, where processor internal bandwidths are sufficiently high to be modeled as infinite.

We propose to model a shared link as several logical links, which map to one physical link. $\boldsymbol{B^D}$ can then model an interconnection network that consist of dedicated and directional communication lines between the platform's devices as well as a platform with shared communication links. In other words, (3) can capture the bandwidths of dedicated or shared, uni- or bidirectional links.

An SDR platform may have a global memory to store the downloaded radio and user applications. The application mapping process then distributes the SDR and user applications' functions between the processors' local memories for their distributed processing. The memory availability for program and data can be modeled as shown in (4), where $(M_1)^D$ through $(M_{N(D)})^D$ represent the processors' local and $(M_{N(j)+1})^D$ the platform's global memory resources in mega-bytes (MB).

If we assume that the total energy of E^D milliwatt per second (mWPS) is shared between all the computing resources of SDR platform D, we can model it as $R^{(4,\ D)} = E^D$ mWPS (4). Equation (5), on the other hand, shows a distributed energy model, where $(E_x)^D$ is the amount of energy that is available at processor P_x of SDR platform D.

Note that this modeling is easily extensible just by defining additional model instances of $\boldsymbol{R^{(T,\ D)}}$. Also, matrices can be removed or simply ignored when the corresponding resource becomes unlimited for practical issues and does not require its management (any more).

III. Application Resource Modeling

A. Application Environment

SDR and user applications basically differ in their utility: The SDR application defines the radio functionality of an SDR platform and eventually the transmission mode for service delivery, whereas the user application facilitates the use of the service itself. In other words, an SDR application defines how data is transmitted and received over the wireless link (OSI layers 1–3), whereas the user application defines how the service is locally presented to the user (OSI layer 7).

$$\boldsymbol{R^{(T,\,D)}} = \begin{pmatrix} R_{11}^{(T,D)} & R_{12}^{(T,D)} & \cdots & R_{1,V(T,D)}^{(T,D)} \\ R_{21}^{(T,D)} & R_{22}^{(T,D)} & \cdots & R_{2,V(T,D)}^{(T,D)} \\ \vdots & \vdots & \ddots & \vdots \\ R_{U(T,D),1}^{(T,D)} & R_{U(T,D),2}^{(T,D)} & \cdots & R_{U(T,D),V(T,D)}^{(T,D)} \end{pmatrix} \tag{1}$$

$$\boldsymbol{R^{(1,\,D)}} = \boldsymbol{C^D} = [(C_1)^D, (C_2)^D, \ldots, (C_{N(D)})^D] \text{ MOPS} \tag{2}$$

$$\boldsymbol{R^{(2,\,D)}} = \boldsymbol{B^D} = \begin{pmatrix} \infty & (B_{12})^D & (B_{13})^D & \cdots & (B_{1,N(D)})^D \\ (B_{21})^D & \infty & (B_{23})^D & \cdots & (B_{2,N(D)})^D \\ (B_{31})^D & (B_{32})^D & \infty & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & (B_{N(D)-1,N(D)})^D \\ (B_{N(D),1})^D & (B_{N(D),2})^D & \cdots & (B_{N(D),N(D)-1})^D & \infty \end{pmatrix} \text{ MBPS} \tag{3}$$

$$\boldsymbol{R^{(3,\,D)}} = [(M_1)^D, (M_2)^D, \ldots, (M_{N(D)})^D, (M_{N(\mathrm{D})+1})^D\,] \text{ MB} \tag{4}$$

$$R^{(4,\,D)} = E^D \text{ mWPS} \tag{5}$$

$$\boldsymbol{R^{(5,\,D)}} = [(E_1)^D, (E_2)^D, \ldots, (E_{N(D)})^D\,] \text{ mWPS} \tag{6}$$

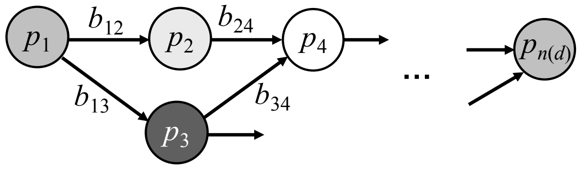

Fig. 4 Modular application.

Modular applications, consisting of independent processing blocks can be processed following the distributed computing concept and individually exchanged for other software implementations. This facilitates meeting real-time constraints and an easy distribution of new SDR application's functions or user application's processes.

Fig. 4 shows a modular application example. It describes a directed acyclic graph (DAG) that may model an SDR application consisting of the $n(d)$ SDR functions p_1, p_2, …, $p_{n(d)}$ (d, $n(d) \in \mathbf{N}$) [7]. Similarly, a user application, which is basically a general-purpose computing application, consists of several processes. Because of the equivalency between SDR functions and user application's processes, we treat both as processes. Then a user or SDR application d consists of $n(d)$ processes p_1, p_2, …, $p_{n(d)}$.

B. General Application Resource Model

Equation (7) provides the general application resource model. It is a $u(t, d)$-by-$v(t, d)$ matrix $\boldsymbol{r}^{(t,\, d)}$ that captures the characteristics of SDR or user application d (t, $u(t, d)$, $v(t, d) \in \mathbf{N}$; $\boldsymbol{r}^{(t,\, d)} \in \mathbf{R}_+^{u(t,\, d) \times v(t,\, d)}$). Parameter t indicates the application resource type, which may be a computing resource requirement or a service-related indicator.

C. Application Resource Model Instances

Equation (8) resumes the $n(d)$ processing requirements of application d. Matrix $\boldsymbol{r}^{(2,\, d)}$ gives the inter-process bandwidth demands, where $(b_{xy})^d$ corresponds to the minimum bandwidth demand for the data transfer from process p_x to process p_y of application d.

Equation (10) summarizes the memory demands, where $(m_1)^d$ through $(m_{n(d)})^d$ represent the processes' program and data memory requirements and $(m_{n(\mathrm{d})+1})^d$ the application's total memory demand.

According to the modeling of energy resources, either as (5) or (6), we model an application's energy requirement either on application (11) or process basis (12).

$$\boldsymbol{r}^{(t,\, d)} = \begin{pmatrix} r_{11}^{(t,d)} & r_{12}^{(t,d)} & \cdots & r_{1,v(t,d)}^{(t,d)} \\ r_{21}^{(t,d)} & r_{22}^{(t,d)} & \cdots & r_{2,v(t,d)}^{(t,d)} \\ \vdots & \vdots & \ddots & \vdots \\ r_{u(t,d),1}^{(t,d)} & r_{u(t,d),2}^{(t,d)} & \cdots & r_{u(t,d),v(t,d)}^{(t,d)} \end{pmatrix} \tag{7}$$

$$\boldsymbol{r}^{(1,\, d)} = \boldsymbol{c}^d = [(c_1)^d, (c_2)^d, \ldots, (c_{n(d)})^d] \text{ MOPS} \tag{8}$$

$$\boldsymbol{r}^{(2,\, d)} = \boldsymbol{b}^d = \begin{pmatrix} (b_{11})^d & (b_{12})^d & \cdots & (b_{1,n(d)})^d \\ (b_{21})^d & (b_{22})^d & \cdots & (b_{2,n(d)})^d \\ \vdots & \vdots & \ddots & \vdots \\ (b_{n(d),1})^d & (b_{n(d),2})^d & \cdots & (b_{n(d),n(d)})^d \end{pmatrix} \text{MBPS} \tag{9}$$

$$\boldsymbol{r}^{(3,\, d)} = [(m_1)^d, (m_2)^d, \ldots, (m_{n(d)})^d, (m_{n(\mathrm{d})+1})^d] \text{ MB} \tag{10}$$

$$r^{(4,\, d)} = e^d \text{ mWPS} \tag{11}$$

$$\boldsymbol{r}^{(5,\, d)} = [(e_1)^d, (e_2)^d, \ldots, (e_{n(d)})^d] \text{ mWPS} \tag{12}$$

$$\boldsymbol{r}^{(6,\, d)} = [(q_1)^d, (q_2)^d, \ldots, (q_s)^d] \tag{13}$$

The “right” radio and user applications are selected as a function of the service and quality of service (QoS) a user desires as well as the radio resource availability. The availability of radio resources is typically characterized in terms of spectrum availability as a function of the momentarily observed interference in a given frequency band and the tolerable interference in that band. Spectrum holes, for example, describe unused or sparsely used frequency bands between (heavily) used bands. To be able to choose another transmission band, it is necessary to know at which bands an SDR application can operate.

Element $(q_x)^d$ in (13) models the quality of service that application d can provide for service x. $\boldsymbol{r}^{(6,\,d)}$ thus contains the information about the service and QoS suitability of a given SDR or user application. An element in (13) would be a function of the theoretically achievable bit rate and BER, among others. For each application, we suggest to define a reference QoS value, such as $(q_{\text{“video”}})^{\text{”128kbps UMTS implementation I”}} = 1$. Since we do not scale $\boldsymbol{r}^{(6,\,d)}$, we can choose this reference freely, facilitating its definition and accounting for future QoS enhancements. The 128 kbps UMTS implementation II could, for instance, deliver a 20% higher quality video service than implementation I. Then, $(q_{\text{“video”}})^{\text{”128kbps UMTS implementation II”}} = 1.2$. $(q_{\text{“video”}})^d = 0$, on the other hand, means that application d does not support a video service at all.

Parameter s in (13) specifies the number of services the SDR or user application d can deliver ($s \in \mathbf{N}$). It will grow with each new service that application d supports, requiring an update of $\boldsymbol{r}^{(6,\,d)}$.

The computing and computing-related resource requirements of SDR applications are non-deterministic. The algorithms’ loop-dependencies and branches suggest a statistical modeling: Some algorithms’ actual computing complexities are a function of the non-predictive radio environment. The user application, furthermore, influences the SDR applications’ momentary computing complexity. For example, in periods where the user application does not receive any data from the user, the corresponding SDR applications (transmitter on one side, receiver on the other) have nothing to process and can be “turned off”, consuming very little computing resources. The elements of $\boldsymbol{r}^{(1,\,d)}$, $\boldsymbol{r}^{(2,\,d)}$, and so forth should therefore be considered as variables representing probability density functions with finite ranges. We will explore this statistical modeling in future work; further discussion is beyond the scope of this paper.

As more than one SDR and user application may deliver the requested service and QoS using the available radio resources, the selection should be a function of the available computing resources. This and other relations between the radio, computing and application environments are discussed in continuation.

IV. Discussion

A. Resource Environments and their Relations

Table I summarizes the main resources of the radio, hardware, and application environments, presents their principal optimization goals, and discusses some inter-environmental implications.

Within the radio environment, the common radio resource management (CRRM) tries to optimize the usage of radio resources across operators and frequencies to balance the system load and interference [8]. Advanced spectrum management techniques moreover assume that frequencies can be dynamically assigned to different RATs. This is called flexible or dynamic spectrum allocation (DSA) [9]. It provides the means for balancing the spectrum utilization and increasing the overall system capacity.

The limited computing resources need to be managed and should be optimally allocated, for example, to minimize the power consumption of SDR platforms. Following this objective could increase the battery life of hand-held devices or minimize the operational cost of network elements.

SDR and user applications deliver a service at a certain QoS to the wireless user. The high number of applications and even higher number of applications’ modules require their efficient management. Applications are selected as a function of the user’s preferences and desires, among others. The principal optimization criterion would then be to maximize the users’ satisfactions.

The different resources of the 3 environments and the corresponding objectives are related, as the 4th column of Table I indicates. For example, the applications cannot be chosen as a function of the service and QoS demand alone, but also as a function of the available computing resources. Flexible radios try to take advantage of these implications.

Table I Resources, optimization goals, and implications.

Environment	Resources	Optimize…	Relations (*implications*)
Radio	• Spectrum • Transmission power	…the joint usage of radio resources to maximize overall system capacity (minimize or distribute interference across RATs)	More wireless users => more interference => better receivers required (*application environment*) => more computing power needed (*computing environment*)
Hardware	• Analog circuitry (RF, IF): availability and capabilities • DSP modules: Topology, processing and bandwidth powers, memory capacity, etc. • Energy resources • Application mapping • …	…the usage of computing resources: 1. distribute computing loads to meet RAT-specific computing requirements (real-time processing and latency) 2. minimize power consumption (= minimize usage of computing resources) 3. …	1. Desired application may not be feasibly mappable to available computing resources (*radio & application environments*) 2. Less complex applications desired to minimize the power consumption (*radio & application environments*)
Application	• Functions • Applications • Computing Requirements • …	…the users' satisfactions: Choose optimal application for each user('s QoS request) under given radio and computing environments	Different SDR functions and applications may meet QoS demands. The one that provides the highest user satisfaction may not be feasible (*computing & radio environments*).

B. Flexible Radios

The flexible usage of radio, computing, and application resources describes flexible radios, which can trade off a resource from one environment for that of another, for example a radio resource for a computing resource.

If a frequency band or RAT is saturated, we can switch to another band or RAT that is less saturated. If, on the other hand, the computing resources allow for more processing, we can, for instance, implement better interference cancellation techniques at the receiver and possibly transmit at a lower power, generating less interference to other users. This is one way to trade radio for computing resources.

If low interference is observed at some frequency band, the radio could switch to that band and, if necessary, transmit at a higher power to deliver the desired service and QoS using less computing-complex transmitters and receivers. In this case the system would trade computing for radio resources.

The goal of flexible radios is thus to distribute the usage of resources across environments. The final parameters to optimize could be the user satisfaction and the operators' revenues [10]. These two parameters remain to be adequately modeled before introducing algorithms that fairly optimize them in heterogeneous radio, hardware, and application environments.

REFERENCES

[1] J. Mitola III, "The software radio architecture," *IEEE Commun. Mag.*, vol. 33, no. 5, pp. 26–38, May 1995.

[2] J. M. Pereira, "Beyond software radio, towards reconfigurability across the whole system and across networks," *Proc. IEEE VTC-F'99*, pp. 2815–2818.

[3] J. Mitola, "Cognitive radio: an integrated agent architecture for software defined radio," Ph.D. dissertation, Royal Institute of Tech-

nology (KTH), Stockholm, Sweden, May 2000.

[4] S. Haykin, “Cognitive radio: brain-empowered wireless communications,” *IEEE JSAC*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[5] V. Marojevic, X. Revés, A. Gelonch, “Cooperative Resource Management in Cognitive Radio,” *Proc. ICC 2007*, Glasgow, June 24–28, 2007, pp. 5939–5944.

[6] X. Reves, A. Gelonch, V. Marojevic, R. Ferrus, “Software radios: unifying the reconfiguration process over heterogeneous platforms,” *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 16, pp. 2626–2640, Sept. 2005.

[7] V. Marojevic, X. Revés, A. Gelonch, “Computing resource management for SDR platforms,” *Proc. PIMRC 2005*, Berlin, Sept. 11–14, 2005, pp. 685–689.

[8] J. Perez-Romero, *et al.*, “Common radio resource management: functional models and implementation requirements,” Proc. PIMRC 2005, Berlin, 11-14 Sept. 2005, pp. 2067–2071.

[9] P. Leaves, *et al.*, “Dynamic spectrum allocation in composite reconfigurable wireless networks,” *IEEE Commun. Mag.*, vol. 42, iss. 5, pp. 72–81, May 2004.

[10] L. Giupponi, *et al.*, “Towards balancing user satisfaction and operator revenue in beyond 3G cognitive networks,” *Proc. 15th IST Mobile and Wireless Summit*, Myconos, 4-8 June 2006.