The ALOE Project

Ismael Gomez, Vuk Marojevic, Antoni Gelonch

Universitat Politècnica de Catalunya

June 2010

Contents

This document provides information about the ALOE Project. ALOE (abstraction layer and operating environment) is an open-source SDR framework with cognitive computing resource management capabilities. We present the motivation for developing such a framework and introduce the ALOE concepts, architecture, and cognitive functionalities. We conclude with the current research issues and provide a list of further readings.

The ALOE computing resource management (FlexCRM) and other related projects—FlexWaves, aloeUI, and ALOEedu—are described in separate documents. Go to the FlexNets web site (http://flexnets.upc.edu/trac) for more details.

1 Motivation

The software-defined radio (SDR) concept provides more flexibility to wireless communications. It defines transceiver processing chains in software executed on general-purpose hardware. This hardware can, thus, be reconfigured to run different transceiver processing chains (SDR applications or waveforms). Dynamic waveform reconfigurations, switching the waveform during an active user session are also envisaged. This requires software and hardware support and a common framework for the development and deployment of waveforms, in particular.

While defining a common framework for developing and deploying SDR applications it is important to eliminate any platform (hardware and supporting software) dependency. Radio applications are built through a set of precedence-constrained modules. (These modules may be called *objects* or *components*.) Each module represents a more or less complex signal processing block that acquires information from preceding modules in the processing chain and delivers the processed information to the following modules.

2 The ALOE Concept

ALOE supports partial or total reconfigurations of waveforms while facilitating their deployment on heterogeneous and distributed hardware resources. The main attributes and functionalities of ALOE are:

- Flexibility – be able to trade implementation efficiency against flexibility. An efficient implementation makes best use of the available computing resources (low resource overhead), whereas a flexible solution allows for dynamic reconfigurations at the cost of some resource overhead.
- Execution control – coordinate execution across the entire distributed computing system.
- Abstractions – hide platform details and heterogeneity from radio applications, enabling portability.
- Data packet oriented messaging – packet-oriented instead of processor or device-specific communication mechanisms.
- Parameter control – runtime signal (parameter or variable) management.
- Resource monitoring – computing system/environment awareness.
- Computing resource management – efficiently manage the distributed and limited computing resources.

3 The ALOE Layers

ALOE assumes that object interfaces are unknown at design time. This enables dynamically composing and recomposing processing chains at execution time while integrating the objects that assemble the desired waveform.

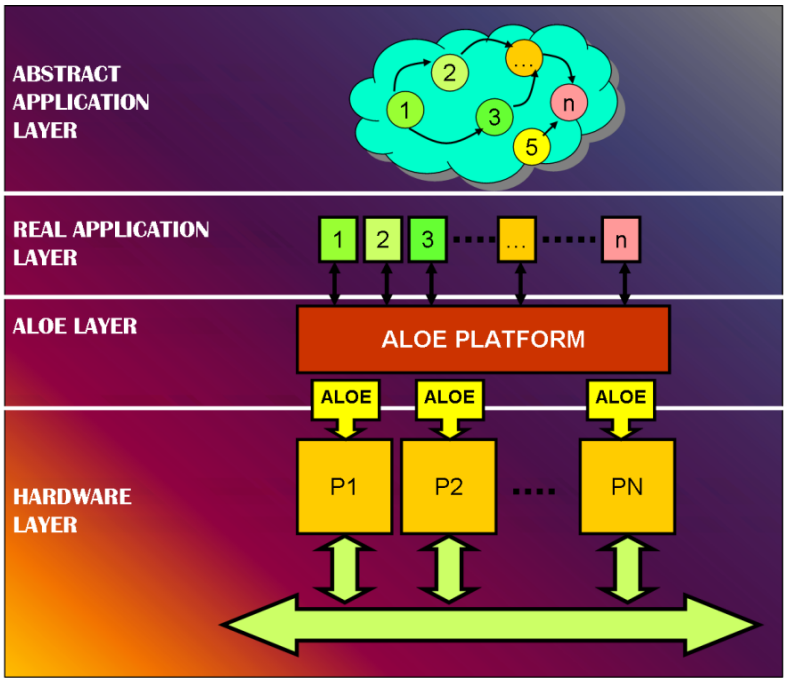

Figure 1. ALOE layers.

Figure 1 shows the ALOE layers. The hardware layer typically consists of several processors or processing elements (PEs), in general, and their physically interconnections. The ALOE Layer abstracts the hardware platform, providing a homogeneous execution environment, the ALOE platform, to applications. The abstract application layer models a waveform by means of the task graph. It abstracts the waveform modules, providing information about the encapsulated signal processing tasks (modules, components, or objects), their precedence constraints and data flow requirements. The real application layer uses the services provided by the ALOE layer for assembling the desired waveform and distributing its components among the available hardware resources.

4 The ALOE Architecture

Figure 2 shows the relation between the different ALOE components and libraries. The application software (here represented by a single module) uses the ALOE services to interact with its environment. These services are accessible as function calls; the ALOE software library contains their implementation. The basic operations provided by the software library may require profound platform or hardware management. The ALOE hardware library makes these issues transparent to the software library. It takes advantage of the available hardware services and operating system tools, if present.

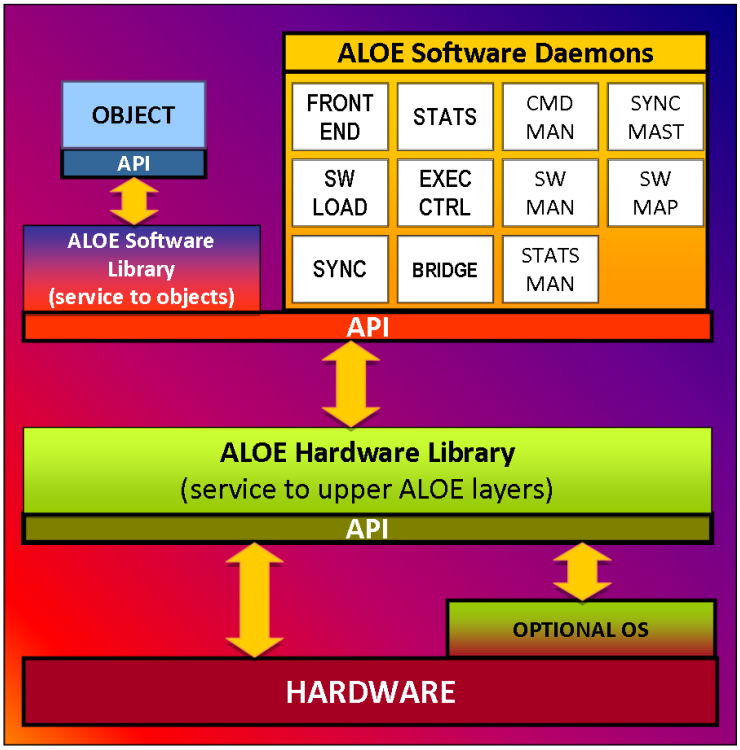

Figure 2. ALOE architecture.

The ALOE software components (ALOE Software Daemons) are accessed though the ALOE software library and perform several tasks for successfully running a waveform on distributed computing resources. The implementation of these components is platform independent and, hence, directly portable to other platforms (if the hardware library is available on these). A short description of the software daemons and their functionalities follows, where MAN is an acronym for manager.

- CMD MAN: Provides a central access to ALOE for higher level control applications, for instance, including GUIs, text-based commands, software development tools, etc.
- HW MAN: Automates the computing resource management for a dynamic allocation and reallocation of computing resources.
- SW MAN: Administrates the application and component repositories.
- STATS MAN: Provides the initialization parameters of application modules and monitors the evolution of application variables.
- BRIDGE: Acts as a link for data transfers between connected PEs.
- SYNC MAST: Provides the time reference for all PEs.
- FRONT-END: Routes the ALOE control packets among the daemons and gathers the hardware status information.
- SW LOAD: Assigns local resources to modules and their data interfaces.
- EXEC CTRL: Ensures that every software module is correctly running under the given quality of service (QoS) constraints (real-time computing resource requirements).
- STATS: Captures and modifies module variables and parameters.
- SYNC: Synchronizes the local time with the remote time reference.

5 Cognitive Functionalities

A cognitive radio is a radio that features a set of tools or procedures for detecting the user's communication needs and for providing radio (and computing) resources that are most appropriate to satisfy these needs.

ALOE provides several cognitive functionalities. We divide them in two groups:

1. Cognitive computing resource management,
2. Cognitive application and execution environment.

Figure 3 illustrates a two-dimensional space, where the vertical axis represents the intelligence level and the horizontal axis the hardware (left) and software (right) spaces.

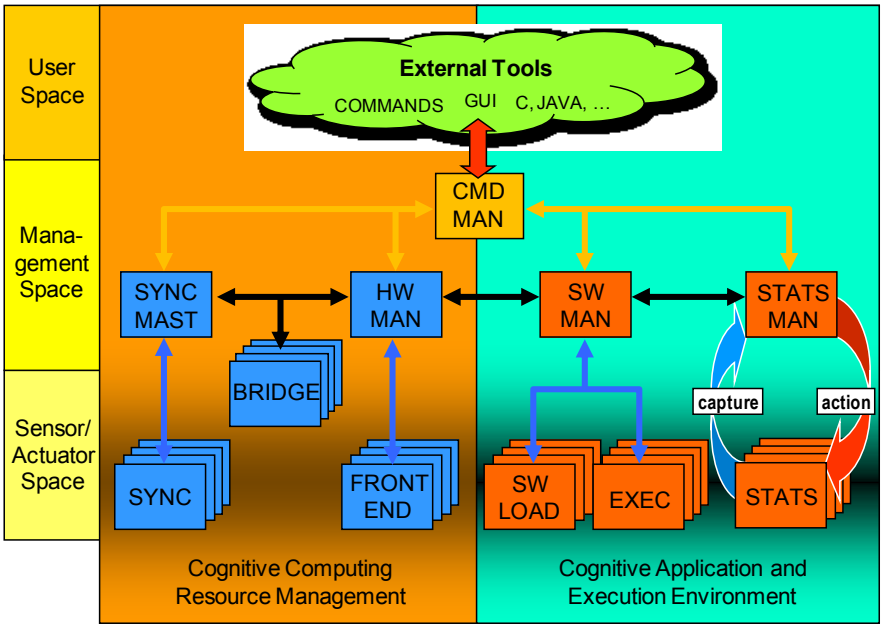

Figure 3. Cognitive functionalities.

5.1 Cognitive Computing Resource Management

The system automatically gains and continuously updates the information about its computing resource status and architecture. This includes:

- Plug-and-play network discovery (plugged/unplugged processors at runtime),
- Computing resources management (time, area, power, …),
- Processor-internal parameter states (consumed power, remaining battery, …).

5.2 Cognitive Application and Execution Environment

The cognitive application and execution environment manages waveform variables, including

- Real Time execution supervision,
- Waveform and component repository management.

5.3 User, Management, and Sensor/ Actuator Spaces

All cognitive functions are executed by software daemons. They are grouped in two types depending on their "intelligence" level:

- **Manager daemons:** These are intelligent elements that do not directly access environmental variables or parameters but make decisions as a function of their values and predefined methods.

- **Sensor/ actuator daemons:** These are unintelligent elements that provide direct access to waveform and system variables and parameters. The interaction is bidirectional, allowing the capture and modification of variable values.

The unintelligent entities at the bottom of Figure 3 are not allowed to directly communicate with one another. The information they gather is reported to its immediate manager, which controls their actions. This separation is very useful for clarifying and understanding the functionalities of the system interactions. The manager daemons are able to communicate with other manager daemons enabling a common management approach. A higher-level intelligent entity, the **CMD MAN daemon**, serves as centralized interaction gate to the ALOE framework.

6 Current Research

We are currently porting ALOE to DSPs and FPGAs. Limited ALOE versions for TI DSPs and Xilinx Virtex 5 FPGAs are already available. Future research will address multiprocessor-systems on chip (MP-SoCs). We also work on improving our ALOE tools for easing the development and deployment of waveforms. The ALOEedu contains educational material for experiencing ALOE and its tools. Visit the FlexNets web site for updates.

Further Readings

The following documents contain further information about the ALOE framework.

[1] X. Reves, A. Gelonch, V. Marojevic, R. Ferrus, “Software radios: unifying the reconfiguration process over heterogeneous platforms,” *EURASIP J. Applied Signal Processing*, vol. 2005, no. 16, pp. 2626-2640, Sept. 2005.
Online available: http://174.129.244.7/journals/asp/2005/158343.abs.html

[2] I. Gomez, "A Software Framework for Software Radio", Ms.C thesis, Universitat Politècnica de Catalunya, 2008.
Online available: http://flexnets.upc.edu/trac/wiki/Publications