

# SDR용 미들웨어 구조: SCA 코어 프레임워크

Middleware for SDR: SCA Core Framework

## 차세대 이동통신 특집

김홍숙 (H.S. Kim)	이동미들웨어연구팀 선임연구원
오상철 (S.C. Oh)	이동미들웨어연구팀 선임연구원
박남훈 (N.H. Park)	이동미들웨어연구팀 팀장
김진업 (J.U. Kim)	SDR연구팀 팀장

## 목 차

- .....
- I. 서론
  - II. SCA 프레임워크
  - III. 기지국 개발 적용 사례
  - IV. 결론

모든 유/무선 네트워크에서 제공하는 서비스를 언제, 어디에서나 단일 단말 시스템으로 접속할 수 있는 유비쿼터스 유무선 통신 시대의 도래가 가시화됨에 따라, 하나의 개방형 플랫폼상에서 소프트웨어적인 변경을 통하여 다양한 무선 접속 규격 및 서비스를 지원할 수 있는 SDR 기술의 중요성이 더욱 부각되고 있다. 본 고에서는 SDRF에서 정의한 SDR 소프트웨어 참조 모델에 기반하여 JTRS에 의해 개발한 SCA 규격 중 코어 프레임워크를 중심으로 구조 및 기능을 살펴보고, SDR 기반 이중모드 기지국 구현에서 SCA 규격 적용 사례를 소개한다. 마지막으로 SDR 기술의 실현 가능성과 관련된 기술적 고려 사항을 논의한다.

## I. 서론

이동통신 사용자의 서비스 요구사항은 전통적인 음성 서비스, 문자, 이미지 서비스에서 다양한 형태의 멀티미디어 데이터를 단말, 통신모드, 네트워크의 종류, 사용 장소에 구애없이 사용하는 방향으로 변화하고 있다. 이러한 요구 사항의 변화는 통신 사업자 및 단말기 제조업체에게 기존 통신 규격을 포함한 여러 통신 규격의 지원 및 새로운 표준 규격의 신속한 적용 또는 업그레이드에 적용할 새로운 기술 또는 패러다임의 변화를 요구하고 있다[1]. 이를 위해서는, 상이한 복수 개의 표준으로 구성된 핵심 네트워크들과 이들 네트워크 위에 구현된 많은 애플리케이션들로 이루어진 기존 데이터 인프라구조로의 통합이 선행되어야 한다.

기존의 이동통신 네트워크의 구성 요소들이 일부 기능을 변경할 수는 있지만 대부분은 기능면에서 변경 불가능한 정적인 장치로 구성되어 있기 때문에, 새로운 무선 규격 또는 서비스를 추가하기 위해서는 많은 하드웨어의 교체가 필요하다. 하나의 개방형 플랫폼 상에서 소프트웨어적인 변경을 통해 다중모드, 다중대역, 그리고 다중 기능을 가능하게 하는 SDR 기술은 이동 통신업체가 직면하고 있는 이러한 기술적 도전에 대한 중요한 해결책으로 부상하고 있다[2],[3]. 또한 아날로그 변조를 대신할 디지털 변조 기술 및 스마트 안테나의 출현 등의 기술 발전도 SDR 기술의 실용화에 일조하고 있다[4].

SDR 시스템의 기술 개발 촉진을 위하여 1996년 MMTS Forum이 만들어졌으며, 1998년 SDRF로 개칭되었다[5]. SDRF에서는 SDR 시스템을 기능적 관점에서 정의하는 노력을 통하여, 상위 수준 기능 모델인 SDR 소프트웨어 참조 모델을 제시하였으며, 미국방성 JTRS 프로젝트에서는 SDRF의 SDR 소프트웨어 참조 모델에 기반하여 컴포넌트 기반 프레임워크인 SCA를 설계하였고 SDRF에 의해 SDR 소프트웨어 표준으로 채택되었다[6].

본 고에서는 SDR 기반에서의 미들웨어 관점에서 SCA 코어 프레임워크(Core Framework, CF)의 구

조 및 기능을 기술하고, ETRI에서 개발한 RBS에서의 SCA 적용 사례 및 SDR 기술의 실현 가능성과 관련된 기술적인 고려사항에 대하여 논의한다.

## II. SCA 프레임워크

### 1. SDR 소프트웨어 구조

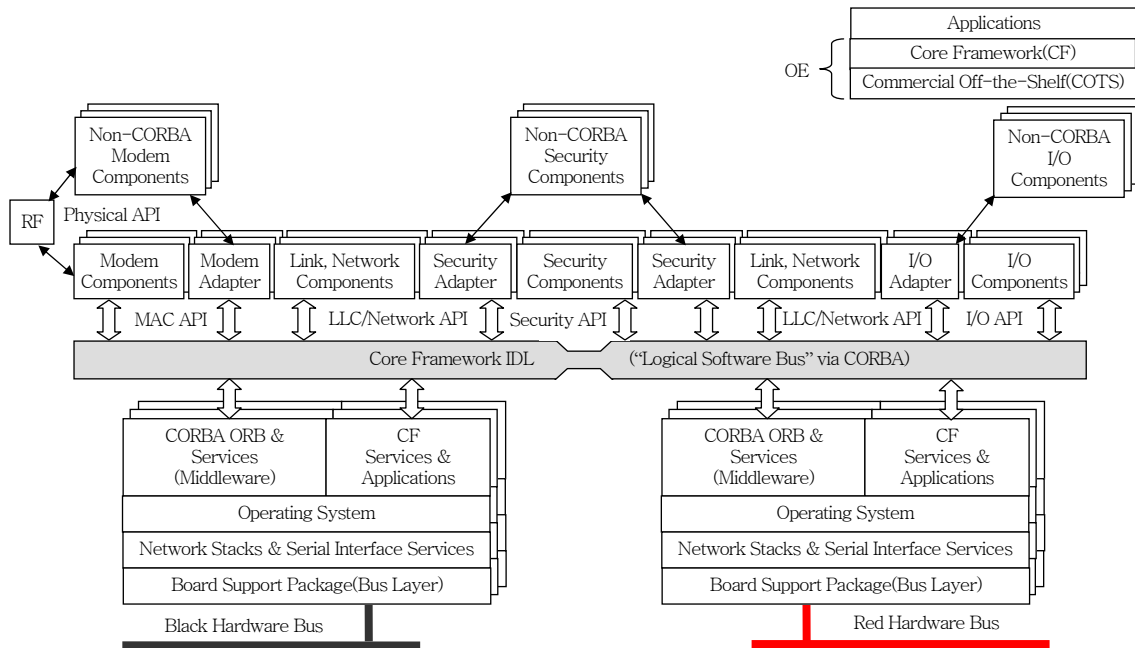
JTRS의 SCA는 SDR 소프트웨어 구조를 (그림 1)과 같은 계층적인 구조를 가정한다. SDR 소프트웨어 내의 계층은 크게 하부 운영 환경(OE) 계층과 운영 환경 계층에서 제공하는 서비스를 이용하여 구현되는 무선 애플리케이션(radio application) 계층으로 구성된다. 하부 운영계층은 다시 Core Framework 서비스, CORBA 미들웨어, POSIX 기반 운영체제로 이루어져 있다.

SCA는 애플리케이션 소프트웨어 컴포넌트들간 AP를 정의하기 위해 빌딩 블록 구조를 제공하며, 빌딩 블록 구조 적용은 컴포넌트 수준의 재사용을 용

#### ● 용 어 해 설 ●

SDR(Software Defined Radio): 무선 이동 통신 시스템에서 RF 영역을 포함한 대부분의 기능 블록을 프로그래밍이 가능한 고속의 processing element상에서 구현된 소프트웨어 블록을 사용함으로써, 하드웨어의 교체 없이 소프트웨어의 업그레이드, 재구성만을 통하여 다중 무선 접속 규격 및 서비스 기능 등을 실현하는 기술이다.

SDRF는 SDR 기술의 성숙 정도에 따라, i) SDR 기능이 전혀 없는 Hardware Radio, ii) 변조나 주파수 밴드 등의 시스템 특성 변경을 제외한 제어 기능을 소프트웨어로 구현하는 Software Controlled Radio, iii) 하드웨어 변경없이 소프트웨어만으로 광대역 동작을 제공하는 Software Defined Radio, iv) Software Defined Radio에서 D/A Conversion 이전 단계에 필요한 아날로그 증폭 또는 헤테로다인 믹싱까지도 제거한 Ideal Software Defined Radio, v) 개인용 장치에 쉽게 통합될 수 있도록 외부 안테나를 사용하지 않고 동작 주파수에 제한이 없는 소형 경량의 컴포넌트와 디지털 형태의 정보를 받아 신호를 변조하고 원하는 무선 waveform으로 송신하는 커넥터로 이루어지는 Ultimate Software Radio의 5단계로 분류하고 있다.



(그림 1) SCA 기반 소프트웨어 구조

이하에 하고, 개발자들이 waveform에 특별한 API 들을 정의하도록 유연성을 제공한다.

## 2. 코어 프레임워크 서비스

SCA 소프트웨어 구조에서 코어 프레임워크[6], [7]는 CORBA 기반으로 SDR 애플리케이션 프로그램의 컴포넌트 기반 컴퓨팅을 제공해주는 부분이다. CF는 소프트웨어 애플리케이션 설계자가 사용할 수 있는 하부 소프트웨어/하드웨어 계층에 대한 추상화(abstraction)를 제공하기 위한 개방형 애플리케이션 계층 인터페이스와 서비스들의 집합이며, (그림 2)와 같이 구성된다. 애플리케이션 프로그램 설계자들은 CF에서 제공하는 인터페이스를 사용하여 하나의 애플리케이션 프로그램을 구성하는 컴포넌트를 프로세싱 노드에 배치, 설치/제거, 시작/정지 등의 관리 작업을 수행할 수 있다.

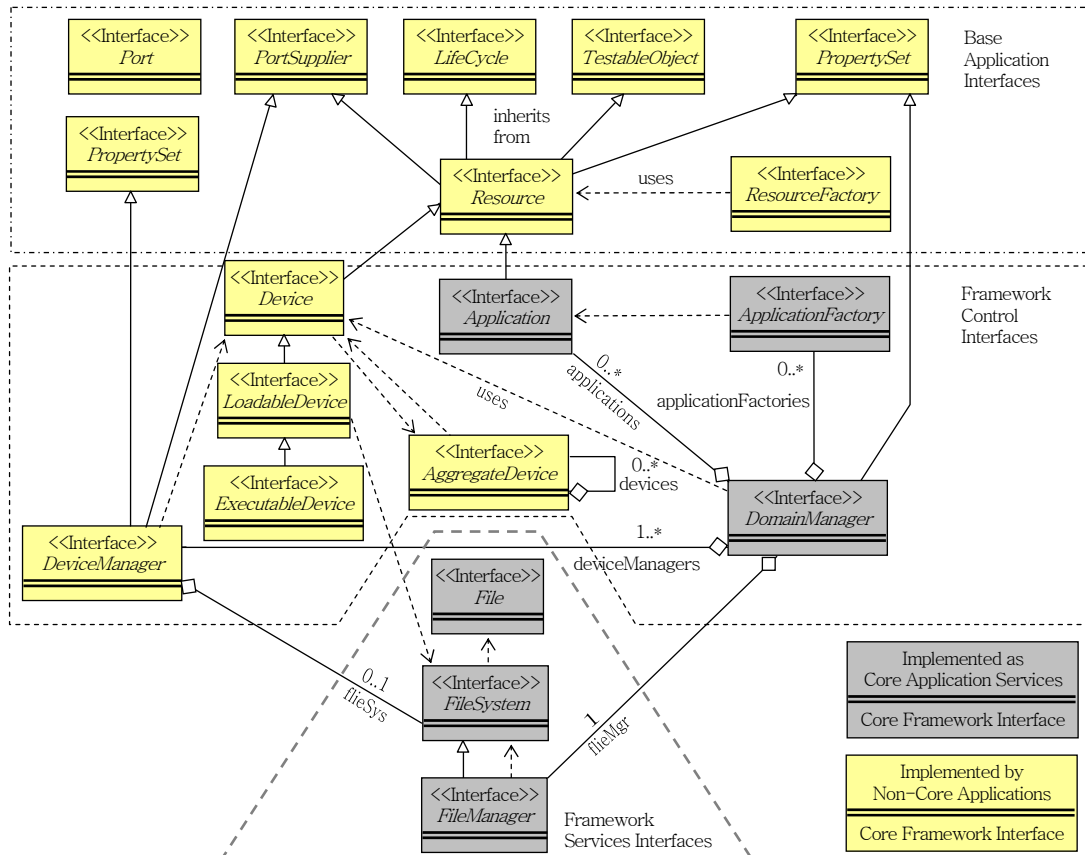
### 가. 기본 애플리케이션 인터페이스(Base Application Interfaces)

모든 애플리케이션 컴포넌트에서 구현해서 제공

해야 하는 인터페이스로 Port, LifeCycle, TestableObject, PropertySet, PortSupplier, ResourceFactory, Resource 인터페이스가 있다. Resource 인터페이스는 애플리케이션 컴포넌트 수준에서의 설치, 제거, 실행, 정지 메소드를 제공하며, Port 인터페이스는 애플리케이션 컴포넌트 수준에서의 통신 기능을 제공한다.

### 나. 프레임워크 제어 인터페이스(Framework Control Interfaces)

CF에서 제공되는 인터페이스로 시스템의 제어와 관련된 기능을 제공하는 Application, ApplicationFactory, DomainManager, Device, LoadableDevice, ExecutableDevice, AggregateDevice, DeviceManager 인터페이스로 구성되어 있다. 이 중 DomainManager는 애플리케이션 프로그램 수준에서의 설치, 제거 및 하드웨어 디바이스의 등록, 해지를 위한 인터페이스를 제공하며, Application 인터페이스는 애플리케이션 프로그램이 설치되는 순간 CF에 자동 생성되어 애플리케이션 프로그램과 구성



(그림 2) SCA 구조 중 Core Framework를 구성하는 인터페이스 서비스

리소스의 관리를 수행한다. Device 인터페이스는 하드웨어 장치에 필요한 기능을 제공한다.

다. 프레임워크 서비스 인터페이스(Framework Service Interfaces)

프레임워크 서비스 역시 CF에서 제공하는 인터페이스로 File, FileSystem, FileManager, Timer 인터페이스가 있다. 이중 FileManager 인터페이스는 물리적으로 다른 파일 시스템들을 논리적으로 하나의 파일 시스템처럼 다룰 수 있는 기능을 제공한다.

라. 도메인 프로파일(Domain Profile)

컴포넌트들을 배치하고 연결하여 애플리케이션 프로그램이 동적으로 구성될 수 있도록, 시스템 내의 하드웨어 장치 및 소프트웨어 컴포넌트 특성을

기술하는 XML 파일을 통칭하여 도메인 프로파일이라 한다. 하드웨어 및 소프트웨어 컴포넌트의 인터페이스, 기능적인 면에서의 능력, 논리적 위치, 상호 의존성 및 관련 파라미터 등의 내용이 포함된다.

도메인 프로파일들은 CF에 애플리케이션 프로그램, 애플리케이션 프로그램 구성 컴포넌트, 노드 정보, 노드를 구성하는 하드웨어 정보를 제공한다.

첫째, 소프트웨어 측면에서의 정보를 기술하는 소프트웨어 프로파일로는 하나의 애플리케이션 프로그램을 구성하는 애플리케이션 컴포넌트들과 이들간의 연결 관계를 기술하는 SAD, 각 컴포넌트의 실제 파일위치, 플랫폼, 실행 환경, 실행에 필요한 최소 메모리, CPU 요구량 등의 정보를 기술하는 SPD와 컴포넌트의 조정 가능한 옵션에 대한 정보를 기술하는 PRF가 있다. 특별히 애플리케이션 프로그램

램 자체가 컴포넌트로 구현된 경우에는 애플리케이션 프로그램이 사용/제공하는 CORBA 인터페이스를 기술하는 SCD가 있다.

둘째, 하드웨어 측면에서의 구성 정보를 제공하는 하드웨어 프로파일로는 전체 프로세싱 노드들의 형상 정보를 기술하는 DCD, 개별 노드 수준에서의 하드웨어 장치 정보(하드웨어 타입, 일련번호, 메모리 크기 등)를 기술하는 DPD와 디바이스의 조정 가능한 옵션을 기술하는 PRF가 있다. 각각의 프로세싱 노드들은 DCD에 지시된 내용에 따라 디바이스를 로딩하여 SCA CF를 부팅한다.

마지막으로, 애플리케이션 프로그램 수준에서의 설치, 제거 및 하드웨어 디바이스의 등록, 해지를 위한 최상위 인터페이스를 제공하는 DomainManager의 실행위치 및 사용되는 CF 인터페이스 또는 서비스 등을 기술하는 DMD가 있다.

### 3. CORBA 미들웨어 및 POSIX 운영체제

코어 프레임워크 서비스에서 사용하는 분산 컴포넌트 지원에 필요한 명명법(naming and typing), 오류 관리(error management), 서비스 중개 관리(service brokering management) 및 통신 링크 제어(communication link control) 등의 하드웨어, 소프트웨어, 운영체제, 네트워크 프로토콜, 프로그래밍 언어의 차이에서 오는 이기종 분산 컴퓨팅 관련 순수 미들웨어 기능은 OMG minimum CORBA를 사용한다[8].

SCA CF 및 minimum CORBA 미들웨어가 실행되는 하부의 운영체제는 그 위에서 실행되는 애플리

케이션 프로그램 또는 컴포넌트들이 무선 통신 프로토콜의 기능을 구현하는 경우, 주어진 종료시한 내에 실행을 완료하는 실시간 특징을 가지고 있으므로 실시간 운영체제(RTOS)이어야 한다. 이를 위해 SCA에서는 IEEE std 1003.13에 정의된 POSIX Realtime Application Support를 준수하는 운영체제를 사용하도록 명시하고 있다[9].

## III. 기지국 개발 적용 사례

휴대성으로 인해 크기가 제한되는 단말기에 비하여 기지국 시스템은 상대적으로 컴퓨팅 능력이나 하드웨어 리소스 사용 측면에서 제한이 적기 때문에 SDR 기술은 대개 기지국에 우선 적용되고 있다. 그러나, 기존 통신 모듈 중 많은 부분은 이미 DSP나 FPGA로 구현되어 있고, 범용 프로세서가 속도면에서 빨라지고는 있으나 일부 통신 모듈(특히, 물리 계층)에서 요구하는 속도를 소프트웨어만으로 구현하기에는 힘든 측면이 있다.

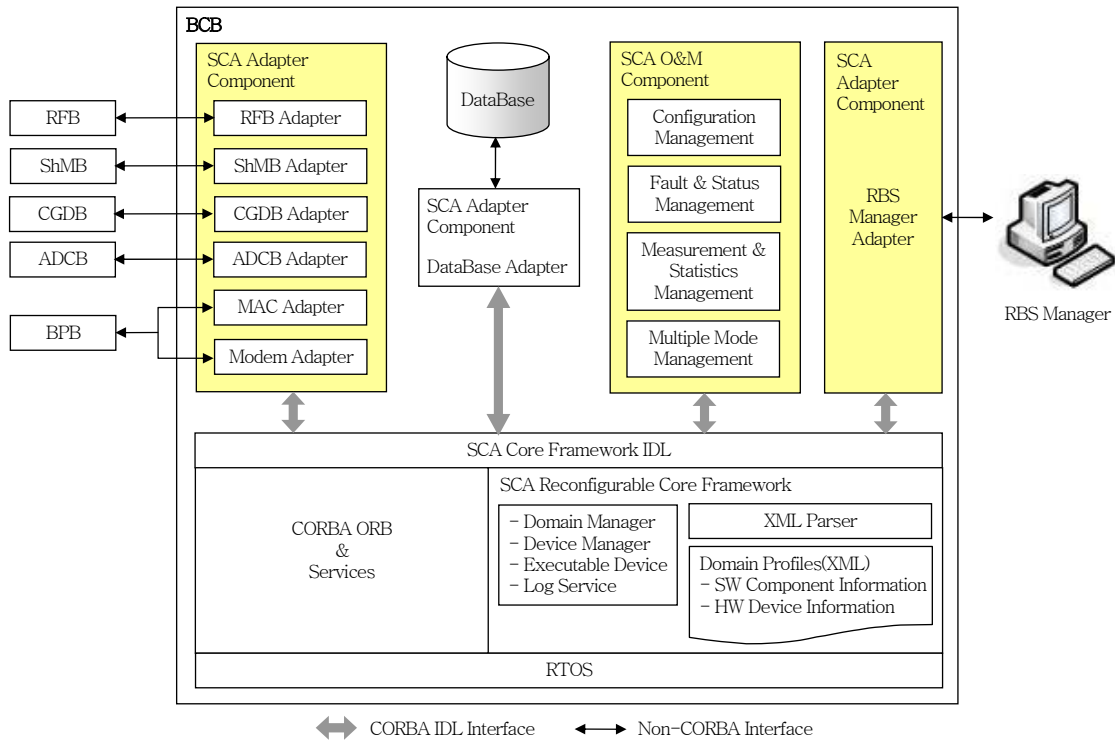
ETRI SDR연구팀에서는 2004년부터 SDR 기반의 RBS 연구 개발을 통하여 이중 모드 기지국을 구현한 바 있다[3]. RBS의 소프트웨어 구조는 (그림 3)과 같으며 미들웨어의 경우 SDR 포럼에서 표준으로 채택한 SCA 규격(버전 2.2)을 준수하였다. 또한, SCA 미들웨어 상위에서 동작하는 RBS용 애플리케이션 프로그램들은 상호간에 SCA Core Framework에서 제공하는 인터페이스에 있는 메소드를 사용하여 통신한다. 이것은 결국 CORBA ORB를 통해 데이터를 주고 받게 된다는 것을 의미한다.

RBS는 여러 종류의 하드웨어 보드들로 구성되어 있으며, 각 SCA 컴포넌트들은 구동시 XML로 기술된 DCD 파일에 있는 배치 정보에 따라 각각의 하드웨어 보드로 배치되어 실행되고, SAD 파일에 있는 입출력 연결 정보에 따라 상호간에 SCA 입출력 커넥션을 맺어 통신하게 된다.

RBS 안에 있는 SCA 미들웨어에는 하나의 Domain Manager와 하나의 Device Manager가 존재하며, CORBA의 Naming Service를 이용하여 등록/

#### ● 용 어 해 설 ●

**미들웨어(Middleware):** 분산컴퓨팅 환경에서 이질성(heterogeneity)—응용프로그램에서 사용하는 하부 네트워크, 운영체제, 또는 프로토콜 인터페이스 등에서의 차이점—을 가진 두 개의 프로그램 사이에서, 중개역할, 투명한 연결성(transparent connectivity) 등의 상호 운용성(interoperability)에 필요한 기능을 제공하는 소프트웨어 또는 소프트웨어 계층이다.



(그림 3) RBS 시스템 소프트웨어 구조

갱신/해제된다. Log Service의 경우 CORBA의 Log Service를 사용하지 않고 SCA 규격에서 별도로 정의한 가벼운 Log Service를 사용한다. 그 이유는 CORBA의 Log Service의 경우, 실제 구현에 적용시 너무 무겁고 제조업체별 상호연동에 문제점이 많아 사용하지 않았다. SCA 규격 제정 시에도 이러한 이유로 CORBA의 Log Service를 채택하지 않고 보다 가벼운 Log Service를 새로 정의하여 사용하였다.

RBS의 SCA 미들웨어는 메인 제어 보드인 BCB 보드에 실장되며, (그림 3)의 RBS manager와 연동하여 애플리케이션을 인스톨하고 애플리케이션 구동시 각 애플리케이션을 해당 하드웨어 보드에 배치하여 실행시킨다. 여기서 RBS manager는 RBS 전체의 제어를 담당하며 SCA non-CORBA 소프트웨어 형태로 구현되었다.

RBS 상에 실행되는 모든 애플리케이션 프로그램들은 하나의 jar 압축 파일 형태로 패키징되어 관리

되며 이 jar 파일 안에는 모든 실행 파일들과 XML 형태의 도메인 프로파일들이 저장되어 있다.

RBS 상에서 동작하는 애플리케이션 프로그램들은 기지국용 운용 및 관리 기능용 컴포넌트 소프트웨어(장애/상태/구성/통계/다중모드 제어)만이 순수 SCA CORBA 컴포넌트로 구현되어 있으며, 나머지 컴포넌트들은 SCA adapter를 이용하는 SCA non-CORBA 컴포넌트 형태로 구현되어 있다.

SCA adapter는 SCA CORBA 컴포넌트 인터페이스와 SCA non-CORBA 컴포넌트 인터페이스를 모두 갖고 있는 SCA 컴포넌트로서, SCA non-CORBA 컴포넌트를 abstraction 하는 기능을 포함하며, SCA non-CORBA 컴포넌트를 배치 및 실행시키고 상태 관리하는 역할을 담당한다.

SCA adapter를 사용한 이유는 기존에 이미 구현되어 있는 이동통신 소프트웨어의 재활용 측면도 있지만 현재의 RBS는 GPP, DSP, FPGA와 같은 다양한 형태의 프로세서를 복합적으로 사용하여 구현되

어 있고, 이들 형태의 프로세서를 모두 안정적으로 지원하는 CORBA 구현이 없기 때문에 SCA adapter를 사용하여 구현하였다.

그러나, 기술의 발전으로 다양한 형태의 프로세서를 안정적으로 지원할 수 있는 CORBA 제품이 출시된다면 모든 컴포넌트를 adapter를 사용하지 않고 순수 CORBA 컴포넌트로 구현할 수 있을 것이다. 그러나, 여전히 상충적인 면이 있는 성능과 이식성의 문제가 거론될 수 있으며 실시간 시스템인 이동통신 시스템 측면에서 향후 이에 대한 최적화 작업이 수반되어야 한다.

와 같이 프로세서 구조의 파이프라인화 또는 병렬 프로세서의 사용을 통해 물리계층의 통신 모듈에서 요구하는 실시간성을 만족시킬 수 있을 것으로 기대된다.

마지막으로, 기존 ASIC을 활용한 이동통신 시스템에 비해 비교적 전력 소모가 많은 SDR 시스템을 기지국과 같이 전력 소모를 고려하지 않아도 되는 시스템에 적용할 때는 문제가 없겠지만, 전력 소모에 민감한 이동단말을 적용할 경우에는 SCA 미들웨어 차원에서 전력 소모관리 기능을 반드시 고려하여야 할 것이다.

## IV. 결론

본 고에서는 SDRF에서 표준 소프트웨어 프레임워크로 사용하는 있는 SCA CF의 구조 및 기능을 살펴보고, SDR 기반 이중 모드 기지국 개발에서의 적용 사례에 대하여 기술하였다.

SDR 기술의 필요성에는 동의하지만, SDR 기술의 실현 가능성에 회의적인 연구자들의 의견은 소프트웨어만으로 통신 모듈, 특히 물리 계층에서 요구되는 고속, 실시간 처리가 가능하냐는 점으로 모아진다. SCA 구조에서 사용되는 CORBA 미들웨어는 이기종 분산 컴포넌트에 필요한 미들웨어 기능을 제공하는 데 따른 오버헤드로 인해 실시간 지원 측면에서 물리 계층과 같이 고속의 연산을 요구하는 곳에 사용하기에 부족한 것이 사실이지만, OMG의 minimum CORBA 표준 개발 및 최근 DSP상에 CORBA를 구현한 프리즘 테크놀로지[10] 등의 기술적 발전 사례를 본다면, 머지 않은 미래에 미들웨어에 수반되는 오버헤드가 통신 모듈의 실시간 처리에서 요구하는 범위 내로 줄어들 것으로 기대된다.

또한 범용 프로세서의 속도 발전에 따라 단말기에 장착된 프로세서들도 많은 부분의 통신 관련 모듈을 컴포넌트화하여 소프트웨어적으로 구현하는 것이 가능해질 것이다. 단일 프로세서의 속도 향상뿐만 아니라, [11]에서의 연구 결과에서 제안한 바

## 약어 정리

ASIC	Application Specific Integrated Circuit
BCB	Basestation Control Block
CF	Core Framework
CORBA	Common Object Request Broker Architecture
DCD	Device Configuration Descriptor
DMD	Domain Manager Descriptor
DPD	Device Package Descriptor
DSP	Digital Signal Processor
FPGA	Field Programmable Gate Array
GPP	General Purpose Processor
JTRS	Joint Tactical Radio System
MMITS	Modular Multifunction Information Transfer System
OE	Operating Environment
OMG	Object Management Group
POSIX	Portable Operating System Interface
PRF	Properties Descriptor File
RBS	Reconfigurable Base Station
RTOS	Real Time Operating System
SAD	Software Assembly Descriptor
SCA	Software Communications Architecture
SCD	Software Component Descriptor
SDR	Software Defined Radio
SDRF	SDR Forum
SPD	Software Package Descriptor
XML	eXtensible Markup Language

## 참 고 문 헌

- [1] Jeffrey H. Reed, *Software Radio: A Modern Approach to Radio Engineering*, Prentice Hall, 2002.
- [2] 김지연, 김진업, "차세대 이동통신시스템을 위한 SDR 기술," IT Standard Weekly, 2002. 5.
- [3] 김덕배, 김진업, "SDR 기지국 구조/기술," 대한전자공학회 학회지, 제33권 제2호, 2006. 2., pp.29-39.
- [4] Walter Tuttlebee et al., *Software Defined Radio: Origins, Drivers and International Perspectives*, John Wiley & Sons, 2002.
- [5] Software Defined Radio Forum. <http://www.sdr-forum.org>
- [6] JTRS, *Software Communications Architecture Specification V3.0*, Aug. 2004.
- [7] 김세화, 유종훈, 홍성수, "SDR(Software Defined Radio)의 소프트웨어 구조," 대한전자공학회 학회지, 제33권 제2호, 2006. 2., pp.17-28.
- [8] OMG, *Minimum CORBA Specification Version 1.0*, Aug. 2002.
- [9] ANSI/IEEE, *POSIX 1003.13: Standardized Application Environment Profile - POSIX Realtime Application Support(AEP)*, IEEE Std. 1003.13, 1998.
- [10] PrismTech, "Importance of Object Middleware on a Digital Signal Processor for SCA Type Architectures," Sep. 2004, <http://www.prismsystems.com>
- [11] T.K. Kim and S.C. Park, "Software Implementation and Performance Evaluation of WCDMA Physical Layer," *IEEE Trans. On Consumer Electronics*, Vol.47, No.4, Nov. 2001.