

SDR 시스템의 운용소프트웨어 품질평가에 관한 연구

A Study of Quality Evaluation for SDR System Operating Software

김민수* · 이건준** · 하성재** · 조상영***†

Min-Soo Kim, Kun-Joon Lee, Sung-Jae Ha, and Sang-Young Cho[†]

*경남대학교 정보통신공학과, ** (주)엘아이씨티, ***한국의국어대학교 컴퓨터공학과

Dept. of Information & Communication Engineerng, Kyungnam University, ICT co., Ltd. and

[†] Dept. of Computer Engineering, Hankuk University of Foreign Studies

요 약

본 논문은 SDR(Software-Defined Radio) 송수신기의 운용 소프트웨어 품질평가를 위한 평가 척도를 제시하고 평가를 자동화할 수 있는 평가 소프트웨어 개발에 대해 설명한다. SDR 운용 소프트웨어는 하드웨어에 종속적인 동작을 하기 때문에 이에 맞는 평가 척도를 도출하기 위하여 동작 환경 및 운영 분야와 동작 방식에 근거한 품질평가 요구사항을 추출하였다. 도출된 평가 척도를 이용한 품질평가의 자동화를 위하여 평가 소프트웨어를 C#언어와 MySQL 데이터베이스를 이용하여 개발하였다. 평가 소프트웨어는 데이터베이스를 이용하여 평가 요소를 입력 또는 갱신할 수 있으며 스크립트를 이용한 자동 평가 기능이 있으며 평가 결과의 시각적 표현이 가능하도록 개발되었다. 도출된 평가 척도와 평가 소프트웨어를 사용하여 파악된 소프트웨어 품질 향상을 위한 개선점은 고품질 SDR 운용 소프트웨어 개발에 유용하게 사용되고 있다.

키워드 : 운용소프트웨어, 품질평가, Software-Defined Radio, 평가소프트웨어

Abstract

In this paper, we described evaluation software that can propose rating scale and evaluate automatically on SDR(Software-Defined Radio) transmitter system. To deduct suitable rating scale, we extracted requirements based on the operating environment, operating application and operating method because SDR operating software is subordinative to the hardware. And we implementation evaluation software to automatically quality evaluation using rating scale draw from the requirements. The implemented evaluation software has automatically evaluation functions using script, can update the evaluation factor and show visual evaluation result. The implemented evaluation software is used to useful the high quality SDR operating software development.

Key Words : Operating Software, Quality Evaluation, Software Defined Radio, Evaluation Software

1. 서 론

SDR(Software-Defined Radio)은 “무선 통신의 재구성이 가능한 시스템 구조를 가능하게 하는 소프트웨어와 하드웨어 기술의 집합체” 또는 “하드웨어(플랫폼) 변경 없이 소프트웨어 업그레이드만으로 다중 모드, 다중 밴드, 다기능

의 무선 시스템 구현을 가능하게 하는 기술”이라고 정의된다[1]. 기존의 라디오 통신 시스템은 하드웨어 부품으로 구현되었으나 SDR은 개인용 컴퓨터 또는 임베디드 시스템의 소프트웨어로 구현된다. 이상적인 SDR 송수신기는 안테나에 ADC (Analog-to-Digital Converter)와 DAC (Digital-to-Analog Converter)가 장착되고 변조와 복조는 소프트웨어에 의하여 수행된다. 이러한 설계는 소프트웨어로만 광범위한 라디오 주파수와 프로토콜을 사용할 수 있는 라디오 시스템을 구현할 수 있기 때문에 실시간으로 매우 다양한 라디오 프로토콜을 변화시키는 서비스를 제공하는 군사 또는 휴대폰 서비스에 매우 유용하다[2].

그러나 이상적인 SDR 송수신기는 구현상 아직 가능하지 않기 때문에 현재의 대부분 시스템은 아날로그 단계의 처리를 하는 재구성이 가능한 하드웨어를 사용한다[3]. [4]에서는 SDR을 위한 핵심 하드웨어 칩을 개발하고 전체 SDR 시스템의 기술을 확보하기 위한 연구를 수행하였으며 고속 통신을 위하여 재구성 가능한 하드웨어를 사용하여 변복조 모듈을 개발하였다. 동시에 전체 SDR 송수신기의 운용 및

접수일자: 2014년 9월 23일

심사(수정)일자: 2014년 11월 13일

게재확정일자: 2014년 11월 27일

† Corresponding author

본 논문은 산업통상자원부 기술혁신사업(No.10043806)의 지원을 받아 수행된 연구임

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

SDR 하드웨어를 제어하기 위한 SDR 운용 소프트웨어가 구현되었다. 다중 주파수, 다중 프로토콜, 다중 기능을 제공하기 위한 SDR 시스템에서의 SDR 운용 소프트웨어는 기능성뿐만 아니라 신뢰성과 효율성이 매우 중요한 요구사항이다. 또한 군사 목적으로 사용되는 SDR 시스템은 보안성이 중요한 요구사항이 된다. 이러한 SDR 운용 소프트웨어의 품질을 보장받기 위해서는 소프트웨어 품질평가 모델에 기반한 소프트웨어 품질평가가 필요하다. 현재 개발된 SDR 운용 소프트웨어의 주된 기능은 하드웨어 장치 제어와 SDR 송수신기의 다른 모듈과의 통신이다. SDR 운용 소프트웨어는 하드웨어와 밀접하게 연관되어 있으며 임베디드 소프트웨어 특성을 갖는다.

기존의 소프트웨어 품질평가는 주로 응용 프로그램 단계의 소프트웨어에 집중되어 왔으며 하드웨어 제어를 포함하는 펌웨어 단계의 소프트웨어 품질평가에 대한 연구는 많지 않다. 본 논문에서는 SDR 운용 소프트웨어를 위한 소프트웨어 품질평가 모델을 도출한다. 이를 위하여 품질평가를 위한 국제표준인 ISO/IEC 9126[5]과 품질평가 프로세스 표준인 ISO/IEC 14598[6]을 기반으로 SDR 운용 소프트웨어의 동작 환경 및 응용 분야에 적합한 품질 요구사항을 파악하고 이를 통한 품질 측정 및 척도를 제시한다. 제시된 소프트웨어 평가 척도를 기반으로 한 평가의 자동화를 위하여 평가 프로그램이 필요하다. 평가 프로그램은 소프트웨어 평가를 체계적으로 진행할 수 있도록 하며 평가 결과를 요약하여 제시함으로써 소프트웨어 개발 과정의 소프트웨어의 미비점 또는 개선점을 도출하여 소프트웨어 품질을 높일 수 있는 개선방향을 제시할 수 있어야 한다.

본 논문에서는 SDR 운용 소프트웨어를 위한 소프트웨어 평가 프로그램 개발에 대하여 기술한다.

본 논문의 구성은 다음과 같다. 2장에서는 특정 분야 또는 제품의 소프트웨어 품질평가에 대한 기존의 연구에 대해 소개하고 3장에서는 SDR 송수신기의 전체 구성을 소개하고 운용 소프트웨어의 동작에 대해 설명한다. 4장에서 SDR 운용 소프트웨어 동작 환경과 응용 분야로부터 소프트웨어 품질 요구사항을 추출하고 품질 요구사항에 상응하도록 도출된 소프트웨어 품질평가 척도에 대한 설명을 한다. 5장에서 소프트웨어 평가를 자동화하기 위한 소프트웨어 평가 프로그램 개발과 활용에 대하여 설명하고 마지막으로 6장에서 결론 및 향후 연구 과제를 제시한다

2. 관련연구

소프트웨어 품질평가 모델은 소프트웨어 품질평가 표준 모델을 기반으로 각 응용 분야 또는 소프트웨어 형태 또는 특성에 따라 다양하게 적용되어 개발되어 왔다. 소프트웨어 품질평가에 관한 국제표준인 ISO/IEC 9126에서 정의하고 있는 품질특성은 기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성이며 각각은 다시 세부적인 부특성으로 구성된다. 각 부특성의 품질은 내부 및 외부 척도에 의해 평가된다. 외부 척도는 실행 가능한 소프트웨어나 시스템을 시험, 운영 또는 관찰해 봄으로써 추출되는 측정치를 위해 사용되며 내부 척도는 설계나 코딩 도중에 실행할 수 없는 내부 속성을 측정하기 위하여 사용된다[5,6].

침입방지시스템의 품질평가 모델에서는 제품의 고유 품질 요구사항으로 보안성이 강조되기 때문에 보안성을 별도

의 품질특성으로 도출하고 보안성능을 효율성의 부특성으로 추가하였다[7]. [8]에서는 사용자 요구사항을 충족하지 못하거나 보안에 문제가 있는 상태로 출시되는 모바일 소프트웨어를 위한 품질측정 척도 설계를 다루었다. 이를 위해 ISO/IEC 25000 표준을 기반으로 AHP를 활용하여 주특성의 우선순위를 정하여 기능성, 성능효율성, 신뢰성, 보안성의 4가지 품질특성을 도출하고 15개의 부특성 및 척도를 개발하였다. 프레임워크 기반 웹 애플리케이션을 위한 품질 특성 연구에서는 사용 경험이 많은 사용자들을 대상으로 자주 발생하는 오류들에 대한 수집과 분석을 바탕으로 기본 조사를 수행하고 웹 애플리케이션의 특성을 고려하여 기능성, 사용성, 효율성, 유지보수성, 안정성을 주특성으로 선택하고 부특성 및 척도를 개발하였다[9]. 이와 같이 품질평가 모델은 각 응용 분야 또는 소프트웨어 특성에 따라 표준 모델을 기반으로 적합한 모델을 개발하고 있다.

SDR 운용 소프트웨어는 임베디드 시스템에서 운용되는 소프트웨어로서 하드웨어의 제어와 다른 모듈 간의 통신을 주 기능으로 하는 소프트웨어이다. 이와 유사한 소프트웨어 품질평가에 대한 연구로 산업용 임베디드 시스템의 특징인 목적의 한정, 실시간 처리, 대량 생산, 강한 내구성을 고려한 소프트웨어 품질 요구사항을 추출하고 평가모델을 개발한 연구가 있다[10]. 또한 전체 소프트웨어가 아닌 프로그램 모듈에 대한 품질평가 함수 산출에 관련된 연구도 있다[11].

3. SDR 시스템과 운용소프트웨어

3-1 SDR 시스템의 구성 및 동작

구현된 SDR 시스템은 AP(Application Processor), SMH (SDR Main HW), RFM(RF Main)으로 나누어진다. 그림 1은 SDR 시스템의 전체 구성을 보여주고 있다. AP는 SDR 운용 소프트웨어를 구동하며 운영 PC와 연결되어 필요한 명령을 전달받거나 동작 상태를 보고한다. 전체 SDR 시스템의 설정 및 제어를 담당하며 운영 PC로부터 받은 전송 데이터를 기저대역 처리를 하여 SMH로 전달하거나 SMH가 수신한 데이터를 받아서 처리하여 운영 PC에 전달한다.

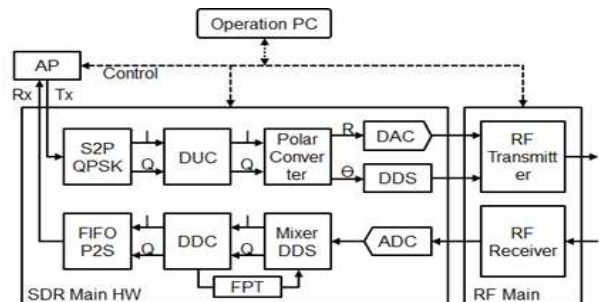


그림 1. SDR 시스템 전체 구성

Fig. 1. The block diagram of SDR system

RF 송신기는 다수의 송신 방식이 가능하도록 제작되었으며 자체 송신배열 제어 보드를 통하여 AP 그리고 운영 PC와 이더넷으로 연결되어 운영 PC에서는 RF 송신기의 동작을 감시할 수 있도록 되어 있다. AP는 RF 송신기와 수신기의 동작 설정을 할 수 있으며 SMH로 하여금 전송되는

RF 신호를 받아 처리할 수 있도록 제어한다. SMH는 FPGA와 하드웨어 부품으로 구성되며 송신부와 수신부로 구성된다. 송신부는 AP로부터 받은 전송데이터를 QPSK 모듈을 통하여 I/Q 신호로 변환시키고 디지털 상향변환기(DUC : Digital Up-Converter)를 이용하여 신호 성형을 하고 이를 폴라 변환기를 이용하여 진폭신호와 위상신호로 변환하고 최종적 RF 수신기에 전달할 신호로 변환한다. SMH 수신부는 RF 수신기의 아날로그 신호를 ADC를 통하여 디지털화 시킨다. DDS는 NCO의 역할을 하며 Mixer를 통하여 하향된 주파수는 디지털 하향변환기(DDC : Digital Down-Converter)를 통하여 다시 하향되며 FPT(Frequency & Phase Tracer)를 통하여 위상/주파수의 대안 정보 Mixer에 전달된다. 최종 I/Q 데이터는 송신과 반대로 다시 직렬화하여 송신 정보 데이터를 최종적으로 얻게 된다.

3-2 SDR 운용소프트웨어

SDR 운용소프트웨어는 AP에 탑재되는 소프트웨어로서 통신데이터를 SMH와 교환할 뿐만 아니라 SDR 시스템의 모든 하드웨어 모듈을 제어하는 기능이 가지고 있다. 그림 2는 SDR 운용소프트웨어가 동작하는 환경을 보여준다.

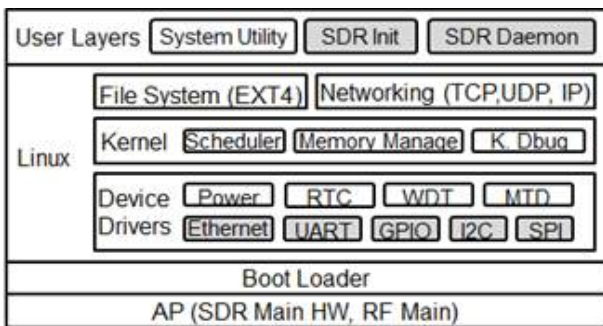


그림 2. SDR 운용소프트웨어 구조
Fig. 2. The configuration of SDR operation software

AP에 이식된 리눅스 상에서 동작하며 리눅스에는 외부 하드웨어 장치와의 통신을 위하여 장치 드라이버를 추가하였다. UART는 RF 송신기의 송신배열 제어기와 통신을 지원한다. 직렬 통신인 I2C, SPI, GPIO로 제어되는 장치들을 위한 드라이버도 추가하였다. AP 내의 SPI와 I2C 통신 제어기가 많지 않아서 GPIO를 사용하여 프로토콜을 소프트웨어적으로 구동하도록 드라이버가 구현되었다. Ethernet은 RF 수신기의 송신배열 제어보드의 제어와 운영 PC로부터의 제어 입력과 실제 통신데이터의 전송을 위하여 사용된다. SPI는 수신 제어보드를 제어하기 위하여 사용된다. 사용자 단계에서는 운영체제 부팅 후에 SDR 시스템 장치 하드웨어 초기화를 위한 SDR Init과 SDR 운용소프트웨어인 SDR Daemon 프로그램이 있다.

SDR 시스템에 전원이 들어오면 AP의 부트로더가 AP 하드웨어를 초기화하고 리눅스 커널을 적재하고 SDR Init을 구동한다. SDR Init은 SDR 시스템의 모든 장치들을 정해진 순서에 따라 초기화 하는 작업을 진행한다. 이를 위해 운용 소프트웨어는 비휘발성 메모리 영역에 모든 주변 장치들의 초기화에 필요한 데이터를 저장하도록 한다. 초기화 후의 SDR Daemon이 구동되어 SDR 시스템 동작을 수행한다.

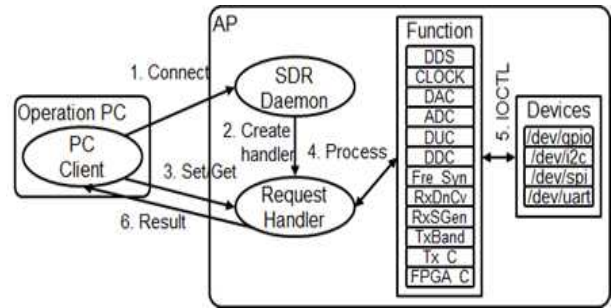


그림 3. SDR 운용소프트웨어 동작 방식
Fig. 3. The mechanism of SDR operation software

그림 3은 AP에 상주하면서 실제 작업을 처리하는 SDR Daemon의 동작 흐름을 보여준다. SDR Daemon은 운영 PC의 PC Client와 연동하여 동작한다. PC Client는 SDR Daemon에 연결되어 SDR 시스템의 하드웨어를 설정하거나 상태 값을 읽는 요청을 하고 전송할 데이터를 주고받는다. SDR Daemon은 PC Client의 요청 시에 처리 핸들러 쓰레드가 생성하고 핸들러는 수행에 필요한 기능을 호출한다. 호출된 기능들이 외부 장치 제어에 해당한다면 장치 드라이버를 통하여 외부 장치에 제어 명령을 전달하고 처리 결과를 반환한다. 외부 장치 제어 명령이 아닌 시스템 자체 처리 명령인 경우에는 내부 기능을 호출하여 처리한다.

PC Client 요청은 대부분 외부 하드웨어 장치 제어를 위한 레지스터의 설정 또는 하드웨어의 처리 상태를 파악하는 상태 레지스터의 값을 읽는 것이다. 하드웨어 제어는 장착되어 있는 다양한 하드웨어 장치의 설정이 필요하며 이러한 설정은 AP에서 직접 설정할 수 있도록 되어 있으며 FPGA를 이용하여 간접적으로 설정할 수도 있다. 소프트웨어적으로 동적인 전송 방식 변경이 빠르게 될 수 있도록 일련의 하드웨어 설정을 FPGA에서 하드웨어적으로 이루어지기도 한다. 동작 설정 후에는 PC Client의 통신 요청에 따라 통신이 이루어진다.

4. SDR 운용소프트웨어 품질평가 요구사항

4-1 SDR 운용소프트웨어 품질평가 요구사항

소프트웨어 품질평가 국제표준인 ISO/IEC 9126[5]은 소프트웨어 품질평가를 위한 6개의 주목성들과 주목성에 따른 부특성들을 제시하고 있다. 소프트웨어 품질평가를 위해서는 제시된 특성들 중에서 평가하고자 하는 소프트웨어 특성에 맞추어 정성적 또는 정량적 평가를 통하여 사용할 특성들을 선택한다. 또한 선택된 각 특성에 맞는 평가 척도를 고안하여 소프트웨어의 품질을 평가하고 이를 점수화하여 평가의 결과를 구한다.

SDR 운용소프트웨어 품질 평가 모델을 도출하기 위하여 소프트웨어의 사용환경 및 응용 분야, 소프트웨어의 형태 및 동작 특징을 기반으로 품질평가 요구사항을 정리하고 소프트웨어 품질평가 국제표준 ISO/IEC 9126을 기준으로 관련 주목성과 부특성을 선정하였으며 일반 소프트웨어에서 요구되는 일반 부특성 항목을 추가하였다. 선정된 부특성의 평가를 위한 척도는 실제 측정 가능한 항목들을 나열하여 가장 적절한 것들을 선택하여 구성하였다. SDR은 국제적 표준을 가지고 있지 않아서 부특성인 표준에 대한 준수성은

고려하지 않았다.

본 논문의 SDR 시스템의 운용소프트웨어 환경 및 응용 분야에 의한 품질평가 요구사항이 표 1에 정리되었다. 품질평가 요구사항에 대한 ISO/IEC 9126의 관련 주특성 및 부특성을 선정하기 위하여 요구사항에 해당하는 주특성과 부특성을 조사하여 나열하고 특성부분의 +는 필요함을 표시하고 -는 필요없음을 표시하며 0은 중간정도의 필요성을 표시하였다. 이를 통해 기능성, 신뢰성, 효율성이 주요 주특성으로 선정되었으며 부특성으로 보안성, 결합허용성, 회복성, 성숙성, 정확성, 시간반응성이 선정되었다.

표 1. 동작환경 및 응용분야에 의한 품질평가 요구사항
Table 1. The requirement of quality evaluation by operating environments and application fields

Contents	Quality Requirements	Sub-characteristic	Main-characteristic
Fields used for communication security is required	Operational and communication security capability is required	+Security	Functionality
No other operating PC connectivity equipment	Equipment linkage considered necessity is weakness	-Interoperability	Functionality
Use special purpose	Necessity of transplantation is weakness		-Portability
Operable in a movement/harsh environment	Error handling capacity of required by the environment	+Fault tolerance +Resiliency	Reliability
Operate using radio	Coping with errors in the data by the environment	+Maturity	Reliability
Real-time operation on using for communication	Satisfy the real-time requirements Compliance with specify processing time	+Accuracy +Time reactive	Functionality Efficiency

SDR 시스템은 소프트웨어의 제어에 의한 하드웨어 동작으로 구현되었기 때문에 소프트웨어의 평가가 하드웨어에 독립적이지 못하다. 또한 SDR 운용소프트웨어는 사용자 인터페이스를 제공하지 않으며 임베디드 소프트웨어의 특징을 가지고 있다. 이러한 특징을 반영한 품질평가 요구사항이 표 2에 정리되었으며 선정된 주특성과 부특성을 나열하고 있다.

표 2. SDR 운영소프트웨어 특성에 의한 품질평가 요구사항
Table 2. The requirement of quality evaluation by SDR operating software characteristics

Contents	Quality Requirements	Sub-characteristic	Main-characteristic
Hardware control and communication main functions	Operation check of each unit control Check the correct operation results Control whether or not completed within the required time	+Compatibility +Accuracy +Time reactive	Functionality Functionality Efficiency
In the form of hardware management software	Requirements of hardware monitoring functions The ability to easily identify the hardware state	+Compatibility +Interoperability	Functionality, Usability
The software operation is dependent on the hardware operation	Hardware and being interoperable Elements of the hardware performance evaluation A necessity of capacity of coping with error in hardware	+Interoperability +Time reactive +Maturity	Functionality, Efficiency Reliability
Operation at the request of the client	Degree of processing speed and throughput of requests	+Time reactive	Efficiency
Build-in embedded	Indirect use without the UI	0Resource	-Usability,

software	Sufficient resources to small, high-value products If the transplant make less. Concise HW interface is required	efficiency +Analyticity	Efficiency -Implantation Maintainability
----------	--	----------------------------	--

품질평가 요구사항 분석에 의하여 6개의 주특성 중에서 기능성, 신뢰성, 효율성이 SDR 운용 소프트웨어의 가장 중요한 특성이며 기능성에서는 적합성, 정확성, 보안성 부특성이 신뢰성에서는 성숙성, 결합허용성, 회복성 부특성이 효율성에서는 시간반응성 부특성이 중요하다. 다른 항목으로 사용성의 운용성, 유지보수성의 분석성 등이 필요하며 효율성의 자원효율성도 중요도는 떨어지지만 필요한 항목이다. 이 식성은 고려하지 않아도 되는 주특성이다. 이러한 요구사항 분석은 각 주특성 및 부특성에 대한 가능한 척도를 구하고 평가를 하고 난 뒤에 전체 소프트웨어 품질 평가 점수의 가중치로 활용하도록 한다.

4-2 기능성 및 신뢰성

기능성은 소프트웨어가 특정 조건에서 사용될 때, 명시적 혹은 묵시적으로 요구된 기능을 제공하는 능력을 말한다. 신뢰성은 주어진 조건에서 사용될 때, 정해진 수준의 성능을 유지하는 능력을 말한다.

표 3. 기능성 척도
Table 3. The functionality scale

sub characteristic	scale name	Purpose of the scale / Description
Suitability	Suitability of functional structure	Measure of suitable the functional structure in specified requirements X = A/B, A: Number of evaluation of the suitable function implementation in accordance with the requirements B: Total number of functions in the requirement
	Suitability of the module control	Measure of suitable hardware module control function from AP X = A/B, A: Number of set of the suitable register in accordance with the requirements B: Total number of register set in the requirement
Accuracy	Accuracy to requirement	differences between the actual and reasonable expected results acceptable X = A/T, A: Number of cases occurred over a reasonable acceptable range for the result T: Operation time
	Computational Accuracy	Measure the accuracy of calculation result with respect to the modules X = A/B, A: Number of correct calculations B: Total Number of attempt to calculation
		Precision
	Accuracy to module control	Measure the accuracy of hardware module control by AP X = A/B, A: Number of module is successful B: Total number of attempt to module control
Security	Protect the right of the administrator access to the system	Measure that administrator rights to operation software maintenance X = A/B, A: Number of access failure for software administrator B: Number of attempts to access an invalid administrator
		Encryption

	of the data	signal of the operation software X = A/B, A: Number of encrypted transfer from data/control signal B: Number of data transmission units
--	-------------	--

표 4. 신뢰성 척도

Table 4. The reliability scale

sub characteristic	scale name	Purpose of the scale / Description
Maturity	Failure density	Measure of a successful result to the specified attempt X = NFAI/SIZE, NFAI: Number of detected failure SIZE: Program size
	Fault density	Measure of the degree of success for unspecified various attempts X = NFAU/SIZE, NFAU: Number of detected fault SIZE: Program size
Fault tolerance	Breakdown avoidance	Measure how to avoid breakdown if a failure occurs X = 1-(A/B), A: Number of breakdowns B: Number of failures
	Incorrect operation avoidance	Measure how much avoid significant errors in the software operation X = A/B, A: Number of avoided critical and serious failure to test case of fault pattern B: Total number of test case of fault pattern
	Data endurance	Measure the data that is no errors in abnormal behaviour X = 1-(A/B), A: Number of failure the data retention B: Number of attempt to abnormal behaviour
Recoverability	mean system down time	Mean down time of system when a occurs system breakdown X = T/B, T: Total down time B: Number of observed system breakdown
	Auto recoverability	Does the system will automatically recover from situations that require recover X = T/B T: Number of successful recovery the system down in each case B: Number of system recovery is needed
	Restartability	Ease of system restart X = A/B, A: Number of times the user has to restart without difficulties successfully, B: Number of attempt to system restart
	Restorability	How data can be restored after an abnormal termination system X = A/B, A = Number of restoration successfully restore data B = Total number of attempt to data restoration

4-3 효율성 및 사용성

효율성은 주어진 조건에서 사용하는 자원의 양을 기준으로 적합한 성능을 제공하는 능력을 말한다. 사용성은 주어진 조건에서 사용될 경우, 사용자가 이해하고 배우고 사용하기 쉬운 소프트웨어 능력을 말한다.

표 5. 효율성 척도

Table 5. The effectiveness scale

sub characteristic	scale name	Purpose of the scale / Description
Time behaviour	Boot time	Time required until the run SDR daemon during boot or reboot
		X = Tf-Ts, Tf: Time to SDR daemon will be normal operation

Data transmission time		Ts: Time to request of boot or reboot
		Time to transmit data receive a response message from the SDR daemon X = Tf-Ts, Tf: Time of receiving the transmitted response message Ts: Time of transmitting data
CPU utilization		Measure the CPU utilization of SDR daemon in AP X = A/B, A: CPU utilization amount of the SDR operating software B: Total CPU usage in AP
		Measure the memory utilization of SDR daemon in AP X = A/B, A: memory utilization amount of the SDR operating software B: Total memory size of AP-BBPU
FPGA gate utilization		Measure the gate utilization for FPGA implementation X = A/B, A: Amount to actual use in FPGA B: Total amount used by the gate in FPGA
		Measure the frequency spectrum utilization in Software X = A/B, A: Amount to use of frequency spectrum B: Amount to the available frequency spectrum
Source code utilization		Measure the how much unwanted source code in SDR daemon X = A/B, A: Source code that uses SDR daemon B: Total number of lines of source code for the daemon

표 6. 사용성 척도

Table 6. The usability scale

sub characteristic	scale name	Purpose of the scale / Description
Learnability	Ease of function learning	How long does the user take to learn to use a function X = Avg(Ti), Ti: Mean time taken to learn to use a function correctly
	Ease of learning to perform a task in use	How long does the user take to learn how to perform the specified task efficiently X = Avg(Ti), Ti: Mean time until achieved to perform the specified task
Operability	Default value availability in use	Can user easily select parameter values during input X = A/B, A: Number of value that provides a default B: Number of value that user needs to set
	Operational consistency in use	Can user to use consistent function in software X = A/B, A: Number of assessment that is maintained consistent set of SDR device registers B: Total number of register set of the SDR device
	Erro correction	Can user easily correct error on task X = Tf-Ts, Tf: Time of completing correction of specified type errors of performed task Ts: Time of starting correction of specified type errors of performed task
	suitable operation time	Time of taken process specified task after completing the software training courses T = Time required for the user to operate the software to perform specified task
Provide for user defined functions		Can modify the interface or operation for convenience
		X = A / B, A: Number of successful the user defined B: Number of attempt to user defined

4-4 유지보수성 및 이식성

유지보수성은 요구사항 및 환경 변화에 따라 소프트웨어를 개선하거나 수정하고자 할 경우 변경될 수 있는 능력을 말한다. 이식성은 다른 동작, 하드웨어, 소프트웨어 환경으로 옮겨질 수 있는 능력을 말한다.

표 7. 유지보수성 척도

Table 7. The maintainability scale

sub characteristic	scale name	Purpose of the scale / Description
Analysability	Record the log data	Can see the log data about function of AP and hardware module $X = A/B$, A: Number of collecting log data about the operation corresponding to each function B: Total number of operation for functions
	Status monitoring to operation	Can user and maintainer monitor the operating state of the SDR Daemon $X = A/B$, A: Number of proper response to state information request B: Total Number of state information request
Changeability	Ease of parameter setting	Can the user or maintainer easily change parameters? $X = 1 - (A/B)$, A: Number of fails to change parameter B: Number of attempts to change parameter
	Ease of troubleshooting change	Can maintainer modify the program the solving problem in use $T = \text{Sum}(A/B)/N$, A: Time to modify the program B: Size of the modified program N: Total number of program modification request
Testability	Ability to perform the operation test	Can user or maintainer easily step by step check and operational test after maintenance $X = A/B$, A: Number of stop and restart by maintainer at check point step by step B: Total number of check point in steps
	Easy test	Can user and maintainer easily perform operational testing and determine whether the software is ready for operation or not? $X = \text{Sum}(T)/N$, T: Time spent to test to make sure whether reported failure was resolved or not N: Number of resolved failures

표 8. 이식성 척도

Table 8. The portability scale

sub characteristic	scale name	Purpose of the scale / Description
Adaptability	Operation software adaptability	Can user or maintainer easily adapt operation software to environment? $X = A/B$, A: Number of errors or failures do not occur, if user operate the software for operating system B: Total number of times adapted to the respective operating system
	Hardware adaptability	Can user or maintainer easily adapt hardware to environment? $X = A/B$, A: Number of errors or failures do not occur, if user operate the software for hardware B: Total number of times adapted to the respective hardware environment
Installability	Ease of installation	Can user of maintainer easily install software to operation environment? $X = 1 - (A/B)$, A: Number of cases in which user fails in set-up during set-up operation B: Total number of cases in which user attempt to setup during set-up operation
Replacability	Continued use of data	Can user or maintainer easily continue to use the same data after replacing this software to previous one? $X = A/B$, A: number of data which are used in other software to be replaced and are confirmed that they

		are able to be continuously used B: number of data which are used in other software to be replaced and planned to be continuously reusable
	Function inclusiveness	Can user or maintainer easily continue to use similar functions after replacing this software to previous one? $X = A/B$, A: number of functions which produce similar results as previously produced and where changes have not been required, B: number of tested functions which are similar to functions provided by another software to be replaced

5. SDR 운용소프트웨어 평가프로그램 및 측정

5.1 SDR 운용 소프트웨어 평가 프로그램 및 동작 환경

ISO9126 표준에 근거한 SDR운용 소프트웨어 품질평가 척도를 이용하여 품질평가를 하기 위한 소프트웨어를 개발하였다. Intel Core i5 2세대 CPU에 4GB 메인 메모리 PC를 사용하였으며 Microsoft VisualStudio 2010 개발 도구를 사용하여 C# 언어로 구현하였다. 그래픽 인터페이스를 위하여 Windows Form 4.0을 이용하였다. 품질평가 소프트웨어는 데이터베이스를 사용하여 평가하고자 하는 소프트웨어의 평가 요소들을 저장하고 이를 갱신할 수 있으며 사용하고자 하는 주특성과 부특성을 등록하고 각 평가 척도를 정의할 수 있도록 하였다. 데이터베이스로 MySQL Server 5.6을 사용하며 데이터베이스 드라이버로 ADO.Net driver for MYSQL 6.6.5.0을 사용하였다. 평가 척도에 따라 스크립트를 이용한 자동 평가와 평가자에 의한 수동 평가로 평가가 이루어진다. 그림 4는 평가 프로그램의 사용자 인터페이스를 보여준다



그림 4. SDR 운용소프트웨어 평가프로그램 인터페이스
Fig. 4. The interface of evaluation program for SDR operation software

SDR 운용 소프트웨어를 평가하기 위하여 SDR 송수신기의 AP와 평가 소프트웨어가 설치된 운영PC를 인터넷으로 연결한다. 자동 평가 항목에 대해서는 해당 평가에 필요한 스크립트 파일을 작성하여 로드하고, 수동 평가항목에 대해서는 평가자가 평가하여 결과를 평가 소프트웨어에 입력시킨다. 그림 5는 SDR 운용 소프트웨어 평가 환경을 도시하고 있다.

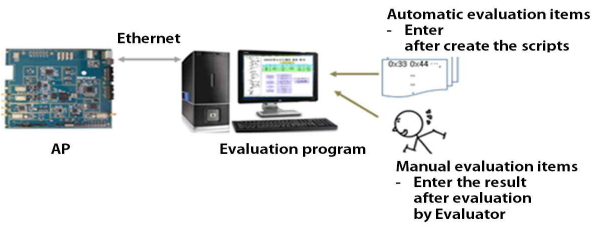


그림 5. SDR 운용소프트웨어 평가환경

Fig. 5. The evaluation configuration of SDR operation software

자동 평가의 경우에는 평가 소프트웨어에서 반복 시험 횟수를 입력하고 시험을 위한 스크립트를 지정하면 스크립트 내의 코드를 자동으로 AP에 반복 전송하면서 결과를 받아 해당 평가 값을 자동으로 입력한다. 평가 소프트웨어는 각 특성들의 평가 진행 상황과 결과가 수치로 표시되며 전체 평가 점수는 특성들의 중요도에 따른 가중치를 기반으로 계산된다.

5.2 SDR 운용 소프트웨어 평가 측정 및 활용

그림 6은4장에서 개발된 평가 척도를 적용하여 SDR 송수신기의 운용 소프트웨어 평가를 수행한 결과를 보여준다. 전체적인 평가 결과를 보여줄 수 있으며 각 주특성 별로 평가 결과를 수치적으로 보여주거나 방사형 그래프 또는 막대 그래프로 결과를 도시한다.

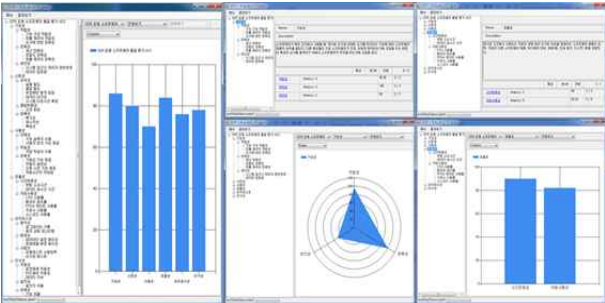


그림 6. SDR 운용소프트웨어 품질평가 결과 인터페이스

Fig. 6. The result interface of quality evaluation for SDR operation software

표 9. SDR운용 소프트웨어 품질평가 결과

Table 9. The result of quality evaluation for SDR operating software

Metrics Tables		Score(%)
Functionality	Suitability	97
	Accuracy	100
	Security	50
Reliability	Maturity	76
	Fault tolerance	40
	Recoverability	100
Usability	Learnability	60
	Operability	88
Efficiency	Time behaviour	90
	Resource utilisation	82

Maintainability	Analysability	90
	Changeability	80
	Testability	60
Portability	Adaptability	78
	Installability	100
	Replacability	58

표 9는 SDR운용 소프트웨어에 대한 품질평가 결과이다. 현재 SDR 송수신기의 개발이 진행되고 있으며 이와 함께 SDR 운용 소프트웨어에 대한 개발도 진행되고 있다. 소프트웨어 품질평가를 통하여 개발되는 SDR 운용 소프트웨어의 미비점 또는 개선점을 도출하고 있으며 이를 통해 운용 소프트웨어의 품질을 높이고 있다. 기능성의 부특성 보안성의 경우 50%의 점수를 보여준다. 이것은 시스템 접근시 관리자 권한 보호 100%와 데이터 암호화 0%의 평균 값이며 데이터 암호화를 통한 보안성을 강화해야 됨을 의미한다. 결합하용성의 40%의 점수는 시스템 평균다운 시간과 자동 복구성을 측정할 만한 프로그램 개발이 이루어지지 않았으며 재시작성과 복원성이 80%의 완전성을 가지고 있기에 측정되는 값이다. 이와 같이 낮은 측정치의 항목들을 확인하고 이를 개선하면서 최종적으로 개발 완료된 운용 소프트웨어의 품질을 측정할 예정이다. 품질 평가를 위하여 도출된 품질 척도에 기반한 평가 프로그램은 개발되는 소프트웨어에 대한 빠르고 체계적인 평가가 가능하도록 한다. 또한 소프트웨어 품질에 대한 평가 결과를 시각적으로 보여줌으로써 즉각적인 개선 사항 도출을 할 수 있으며 고품질의 운용 소프트웨어 개발에 유용하다

6. 결론 및 향후 연구

본 논문에서는 FPGA를 사용한 SDR 시스템의 운용소프트웨어 품질평가를 위한 품질평가 척도를 도출하였다. 기존의 소프트웨어 품질평가는 순수 소프트웨어의 품질 평가에 집중되어 왔으나 본 논문에서의 운용소프트웨어는 하드웨어 제어와 통신이 주목적인 소프트웨어로서 임베디드 소프트웨어의 특성을 갖는다. 소프트웨어의 동작이 하드웨어의 동작에 종속되는 부분이 많기 때문에 기존의 소프트웨어 척도를 보완하는 척도의 개발이 필요하였다. SDR 운용소프트웨어 품질평가 모델을 도출하기 위하여 소프트웨어의 사용 환경 및 응용 분야, 소프트웨어의 형태 및 동작 특징을 기반으로 품질평가 요구사항을 정리하였고 소프트웨어 품질평가 국제표준 ISO/IEC 9126을 기준으로 관련 주특성과 부특성을 선정하여 평가의 중요성이 높은 주특성과 부특성을 파악하였다. 여기에 일반 소프트웨어에서 요구되는 일반 부특성 항목을 추가하였고 선정된 부특성의 평가를 위한 척도는 실제 측정 가능한 항목들을 나열하여 하드웨어 제어에 기반한 척도를 선택하여 구성하였다.

도출된 평가 척도를 기반으로 빠르고 체계적인 평가가 진행될 수 있도록 평가 소프트웨어를 개발하였다. 평가 프로그램은 소프트웨어 평가를 체계적으로 진행할 수 있도록 하며 평가 결과를 요약하여 제시함으로써 소프트웨어 개발 과정의 소프트웨어의 미비점 또는 개선점을 도출하여 소프트웨어 품질을 높일 수 있는 개선방향을 제시하고 있다. 개발된 평가 소프트웨어는 평가에 필요한 항목을 데이터베이스를 이용하여 저장하고 갱신할 수 있으며 평가 자동화를 지원하고 있다. 평가 결과를 그래프의 형태로 표시하여 개

발자 또는 평가자가 소프트웨어의 품질을 시각적으로 인지할 수 있도록 하였다. ISO/IEC 9126을 기준으로 만든 평가 소프트웨어는 일반적인 구조를 가지고 있기 때문에 임의의 소프트웨어 평가에 사용할 수 있도록 되어 있다.

도출된 평가척도와 개발된 평가 소프트웨어를 사용하여 현재 개발되고 있는 SDR 운용소프트웨어에 대한 평가를 지속적으로 진행할 예정이며 최종 개발품에 대한 평가를 수행할 예정이다. 하드웨어와 연관되어 있는 소프트웨어 품질에 대한 품질평가 방법 및 척도 개발이 미흡한 실정이기 때문에 이에 대한 연구는 앞으로 매우 의미가 있는 연구가 될 것이다.

References

- [1] Y.J.Kim, J.M.Kim, B.Y.Lee, S.H.Cho, "SDR Technology", *The Journal of Korean Institute of Electromagnetic Engineering and Science*, vol.18, no.4, pp.3-13, Oct., 2007.
- [2] J. Mitola, Cognitive radio architecture evolution, *Proc. IEEE*, vol. 97, no. 4, pp.626-641, Apr. 2009.
- [3] Martin Palkovic, Praveen Raghavan, Min Li, Antoine Dejonghe, Liesbet Van der Perre, and Francky Catthoor, Future Software-Defined Radio Platforms and Mapping Flows, *IEEE Signal Processing Magazine*, March 2010, pp. 22-33.
- [4] M.S.Kim, K.J.Lee, S.J.Ha, "Design and Implementation of SW based polar converter for wireless communication system", *The Proceeding of the Korean Institute of Electromagnetic Engineering and Science*, vol.1, no.1, pp.258, 2013.
- [5] ISO/IEC 9126 TR, Software engineering-Product quality-Part 1, 2, 3, 4, 2005.
- [6] ISO/IEC 14598, *Information Technology-Software Product Evaluation-Part 1*, 2, 3, 4, 5, 6, 2005.
- [7] H.S.Yang, S.W.Kang, J.Y.Jin, "Quality Evaluate Model Development of Intrusion Prevention System", *Journal of Security Engineering*, vol.7, no.3, pp.225-241, June, 2010.
- [8] Y.M.Han, G.Y.Yun, S.C.Kim, J.M.Choi, H.Y.Yoo, "Design of Quality Measurement Metrics for Mobile Software". *Journal of Korea Computer Congress*, vol.39, no.1(B), pp.199-201, June, 2012.
- [9] Y.Y.Park, H.W.Byeon, "Efficient Development and Quality Evaluation of Online Library System using Framework", *Journal of Korea Multimedia Society*, vol. 13, no. 5, pp.627-640, May, 2010.
- [10] S.W.Lee, H.Y.Lee, H.S.Yang, "Development of Quality Test Module for industrial Embedded Software", *Journal of Korea Academia-Industrial cooperation Society*, vol.8, no.2, pp.297-308, April, 2007.
- [11] H.K.Kim, W.K.Choi, S.J.Lee, "A Study on the quality estimate function of the program module", *Journal of Korean institute of intelligent systems*, vol.12, no.1, pp.67-72, Feb., 2002.



김민수(Min-Soo Kim)

2009년 : 경남대학교 정보통신공학과 공학사

2011년 : 경남대학교 정보통신공학과 공학석사

2011년 ~ 현재 : 경남대학교 정보통신공학과 박사수료,

2013년 ~ 현재 : (주)엘아이씨티 주임연구원

관심분야 : SDR, System Quality Evaluation

Phone : +82-10-4560-2946

E-mail : mskim@lict.co.kr



이건준(Kun-Joon Lee)

1997년 : 원광대학교 반도체학과 이학사

1999년 : 광운대학교 전자공학과 공학석사

2012년 : 고려대학교 전자공학과 공학박사

2013년 ~ 현재 : (주)엘아이씨티 수석연구원

관심분야 : RF/Microwave System, Antenna

Phone : +82-10-7355-0328

E-mail : lkunj21@lict.co.kr



하성재(Sung-Jae Ha)

2001년 : 광운대학교 전자공학과 공학석사

2006년 : 광운대학교 전자공학과 공학박사

2005~2009년 : 삼성탈레스(주) 전문연구원

2011~2014년 8월 : (주)엘아이씨티 기술이사

2014년 8월 ~ 현재 : 한국폴리텍대학 신기술연수센터 조교수

관심분야 : SI, Embedded System

Phone : +82-10-4498-4432

E-mail : sungjae@lict.co.kr



조상영(Sang-Young Cho)

1988년 : 서울대학교 제어계측공학과 공학사

1990년 : 한국과학기술원 전기전자공학과 공학석사

1994년 : 한국과학기술원 전기전자공학과 공학박사

1997년 ~ 현재 : 한국외국어대학교 컴퓨터공학과 교수

관심분야 : Embedded System, Computer Architecture

Phone : +82-31-330-4298

E-mail : sycho@hufs.ac.kr

저 자 소 개