

논문 2007-02-88

# 차량용 AUTOSAR 표준 플랫폼 지원을 위한 RTE 및 Generator 개발

(Development of RTE and Generator for Supporting AUTOSAR Standard Platform for Vehicle)

박세관\*, 조현철, 조성래, 류현기, 정우영

(Shi-Quan Piao, Hyun-Chul Jo, Sung-Rae Cho, Hyun-Ki Ryu, Woo-Young Jung)

**Abstract** : AUTOSAR is a partnership of automotive manufacturers and suppliers working together to develop and establish a de-facto open industry standard for automotive software architectures. In the AUTOSAR architecture, the runtime environment is at the heart of the AUTOSAR architecture to provides the infrastructure services that enable communication between software components and between software components and basic software components. The RTE generator is an automated generation tool for AUTOSAR specification based RTE according to the ECU configuration. It generates communication API function for the software components and the basic software components like OS and COM of the application. The availability and the accuracy of the RTE generator are important for the automobile software which is based on AUTOSAR architecture. this paper proposes an architectural design of the RTE generator for auto-generation of the AUTOSAR based RTE.

**Keywords** : Automotive system, Software architecture, AUTOSAR, RTE, XML parser

## 1. 서론

최근 자동차산업에서 자동차 전자장치 분야가 차지하는 비중이 날로 증가하고 있으며 그 기능도 점차 복잡화, 복잡화되고 있다. 자동차 전자장치에서 사용되는 소프트웨어의 구조 또한 날로 복잡해지고 양도 방대해지고 있다 [4]. 일반적으로 자동차 전자장치 기능의 복잡화로 전자장치 시스템 개발을 위하여 수많은 소프트웨어 및 하드웨어회사들이 참여하고 있다. 전자장치 시스템은 수많은 ECU(Electronic Control Unit)로 구성되고 개발된 소프트웨어 및 하드웨어는 최종 ECU로 통합되어 주어진 기능을 수행한다. 하지만 소프트웨어 개발에서

하드웨어 종속적인 속성으로 하여 시스템 통합에 어려움을 겪고 소프트웨어 이식성도 크게 떨어지고 있다.

따라서 소프트웨어의 하드웨어 종속적인 특성을 제거하여 소프트웨어의 개발을 자유롭게 하고 기존 개발된 소프트웨어의 재사용성을 증가시켜 소프트웨어 개발에 드는 비용과 노력을 절감할 필요성이 증대되었다. 자동차 전자장치에서 대두되는 이러한 요구사항을 만족시킬 수 있는 전장소프트웨어 아키텍처를 정의하고 표준화하기 위하여 만들어진 조직이 바로 AUTOSAR(AUTomotive Open System ARchitecture)이다 [1]. AUTOSAR는 자동차 OEM과 협력사 및 소프트웨어 개발업체를 중심으로 자동차 전자장치 구조의 사실상 표준을 만들고 발전시키기 위하여 만들어진 표준화조직이다.

AUTOSAR 아키텍처는 하드웨어, OS를 포함한 베이직 소프트웨어, RTE(RunTime Environment) 및 어플리케이션 등 네 개의 계층으로 나뉜다. 어플리케이션 계층은 여러 개의 소프트웨어 컴포넌트들로 구성되며 소프트웨어 컴포넌트의 특성 및 ECU

\* 교신저자(Corresponding Author)

논문접수 : 2007. 12. 07., 채택확정 : 2008. 02. 18.  
박세관, 조현철, 조성래, 류현기, 정우영 : 대구경북과학기술연구원

※ 본 논문은 대구경북과학기술연구원 기관고유과제 연구비에서 지원하여 연구하였음.

자원 등을 고려하여 소프트웨어 컴포넌트들을 여러 ECU에 적절히 매핑한다. RTE는 각 ECU에 할당된 소프트웨어 컴포넌트와 컴포넌트 사이 및 상위 소프트웨어 컴포넌트와 하위 베이직 소프트웨어 컴포넌트사이의 통신을 지원함으로써 매핑된 소프트웨어 컴포넌트가 ECU 하드웨어 및 베이직 소프트웨어 상에서 실행가능 하도록 한다 [2], 따라서 각 ECU내에서 RTE는 상위 소프트웨어 컴포넌트의 종류 및 개수, 컴포넌트 사이 통신 설정 및 상위 어플리케이션 기능수행에 필요한 하위 베이직 소프트웨어의 지원기능 등 시스템 설정에 따라 동적으로 구성되어진다.

RTE Generator은 상기와 같은 시스템 설정정보에 근거하여 자동으로 각 ECU내의 RTE를 동적으로 생성해주는 도구로 컴포넌트 기반의 AUTOSAR 아키텍처 구성을 위해 꼭 필요한 도구이다. RTE Generator은 OS와 관련된 AUTOSAR 소프트웨어 컴포넌트 API를 생성하고 어플리케이션 컴포넌트사이의 통신 및 어플리케이션 컴포넌트와 베이직 소프트웨어 모듈 사이의 통신 관리에 필요한 API를 생성하여 RTE기능을 지원하도록 한다. 따라서 RTE Generator에서 시스템 설정정보를 분석, 처리하여 정확한 시스템 설정 정보를 추출하고 설계한 소프트웨어가 정상적으로 동작할 수 있도록 RTE를 자동으로 생성할 수 있는지 여부는 AUTOSAR기반의 소프트웨어 설계개발 가능 여부를 결정짓는 중요한 요소이다. 본 문에서는 소프트웨어 아키텍처 설계과정에서 생성된 AUTOSAR프로젝트파일을 입력으로 하여 필요한 RTE를 생성해주는 RTE Generator을 설계하고 간단한 어플리케이션을 이용하여 일부 RTE모듈을 생성하여 RTE Generator의 정확성을 검증한다. 설계된 RTE Generator은 다음과 같은 특징을 가지고 있다. 1) 설계된 RTE Generator은 RTE 템플릿과 Generator 엔진 2원화 구조로 되어 있다. 이러한 구조는 Generator 엔진에 대한 수정없이 RTE 템플릿에 대한 수정만으로 변경된 AUTOSAR RTE 표준을 RTE생성에 반영할 수 있다는 장점이 있다. 2) RTE 템플릿은 RTE모듈별로 나뉘어 있으며 수요에 따라 필요한 RTE모듈만을 생성할 수 있으므로 RTE Generator의 효율을 향상시킨다. 3) Generator 엔진은 처리과정에서 전체 시스템 정보를 추출하므로 기타 시스템 정보를 필요로 하는 AUTOSAR도구에서 Generator 엔진정보를 사용가능하게 한다. 이는 차후 개발될 AUTOSAR 어플리케이션 개발도구 등 기타 AUTOSAR도구와의 상호연동을 용이하게 한다. 현

재 설계된 RTE Generator은 프로토타입의 개발수준으로 향후 지속적인 기능추가와 수정이 필요할 것으로 사료된다. 예를 들면 현재 목표 프로그램 언어는 표준 C언어이며 차후 C++에 대한 지원기능이 추가되어야 할 것이다. 또한 ECU 상태관리(State Management) 등 서비스계층의 기능에 대한 지원기능도 추가되어야 할 것이다.

서론에 이어 2절에서는 AUTOSAR표준에서 정의하고 있는 AUTOSAR RTE 및 RTE Generation 처리과정에 대해서 살펴보고 3절에서는 AUTOSAR 표준을 만족할 수 있는 RTE를 생성할 수 있도록 설계한 RTE Generator 구조에 대해서 설명한다. 4절에서는 본문의 RTE Generator검증을 위하여 간단한 어플리케이션을 설계하고 일부 RTE모듈을 생성하여 RTE Generator의 유효성을 검증한다. 마지막으로 5절에서는 결론을 맺는다.

## II. AUTOSAR RTE 및 RTE 생성방법

### 1. AUTOSAR 및 RTE 개요

자동차 전자기술의 발전으로 소프트웨어의 기능이 점차 복잡하게 되고 소프트웨어의 개발에 많은 시간과 인력이 필요하게 된다. 따라서 차량용 소프트웨어의 복잡한 구조를 획기적으로 감축하여 개발을 용이하게 하고 구조화된 아키텍처를 사용하여 소프트웨어의 재사용을 향상시킬 필요가 있게 된다. AUTOSAR는 차량용 소프트웨어의 규격과 환경을 표준화 하여 여러 회사가 공통적으로 사용할 수 있도록 정의한다. 궁극적으로는 확장성이 용이한 소프트웨어 실행 환경을 구축하고, 또한 재사용성을 가능케 하는 소프트웨어 플랫폼의 기능을 제공하는 데에 초점을 맞추고 있다.

이런 개념에 기반하여 다음 그림 1과 같은 AUTOSAR 소프트웨어 아키텍처가 제안되었다. AUTOSAR 소프트웨어 아키텍처는 크게 어플리케이션 계층, RTE, 베이직 소프트웨어 계층, 하드웨어 계층 등 네 개의 계층으로 나뉘어져 있다. 어플리케이션 계층은 자동차 전자장치에서 동작되어야 할 기능들이 구현되어 있는 계층으로 여러 개의 소프트웨어 컴포넌트들로 구성된다. 각각의 소프트웨어 컴포넌트는 소프트웨어의 실질적인 기능을 수행하는 다수의 Runnable Entity를 포함한다. 베이직 소프트웨어 계층은 OS를 포함한 서비스 모듈, 하드웨어에 독립적인 상위 어플리케이션 개발을 지원할

수 있는 ECU 추상화 계층, 상위 어플리케이션 소프트웨어에서 마이크로 제어기 레지스터에 직접적으로 액세스하는 것을 방지하는 MCAL(Micro Controller Abstraction Layer) 등 모듈을 포함한다. 베이직 소프트웨어 하단에는 ECU 하드웨어 계층이 존재한다.

RTE는 상위 어플리케이션 계층과 베이직 소프트웨어 계층사이에 존재하며 소프트웨어 설계단계에서 추상적으로 설계된 VFB(Virtual Functional Bus)의 기능을 ECU매핑 후 구현된 계층이다. 소프트웨어 설계단계에서 컴포넌트단위로 구현된 어플리케이션은 컴포넌트가 수행해야 할 기능만을 설계하고 통신기능과 하위 베이직 소프트웨어와의 상호연동은 VFB라는 추상적인 형태로만 설계된다. 소프트웨어 설계가 완료되면 컴포넌트사이의 통신종류 및 베이직 소프트웨어를 액세스하기 위한 API는 RTE생성단계를 거쳐 구현된다. RTE는 주로 소프트웨어 컴포넌트들 사이의 통신 서비스 및 소프트웨어 컴포넌트가 OS를 포함한 베이직 소프트웨어에 액세스할 수 있도록 하는 기능을 제공한다. 현 단계에서 AUTOSAR는 S/R(Sender-Receiver)통신방식과 C/S(Client-Server)통신방식을 지원한다. 각각의 통신모델은 Intra-task, Inter-task 및 Inter-ECU 세 가지 패턴으로 분류할 수 있다.

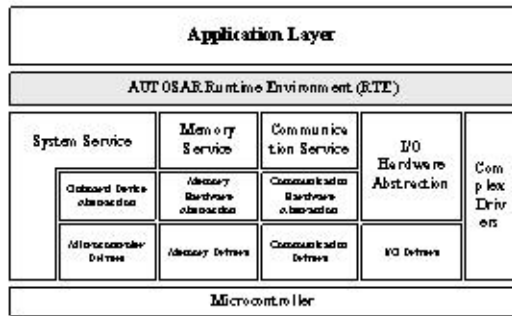


그림 1 AUTOSAR 소프트웨어 구조  
Fig. 1. AUTOSAR Software Architecture

Intra-task 패턴은 동일 OS에 할당된 Runnable Entity사이의 통신방식이며 Inter-task 패턴은 서로 다른 타스케에 할당된 Runnable Entity사이에서 발생하는 통신방식으로 구문스위칭(Context Switch)가 발생한다. 반대로, Inter-ECU 패턴은 서로 다른 ECU에 할당된 Runnable Entity사이에서 발생하는 통신으로 데이터는 통신버스를 통하여 전송된다. RTE는 상위 어플리케이션에서 설계된 통신방식에

따라 상기의 통신API를 생성하여 설계된 어플리케이션이 수행될 수 있도록 한다.

2. RTE 생성과정

RTE Generator은 기본적으로 RTE Contract 단계와 RTE Generation 두 단계를 거쳐 RTE를 생성한다. RTE Contract 단계에서는 AUTOSAR 인터페이스와 같은 중요한 컴포넌트 정보를 사용하여 컴포넌트 헤더파일을 생성한다. RTE Generation 단계에서는 컴포넌트와 관련된 모든 기타 정보를 이용하여 RTE를 생성한다.

2.1 RTE Contract 단계

이 단계는 상위 소프트웨어 컴포넌트 설정으로부터 소프트웨어 컴포넌트와 RTE사이의 인터페이스에 필요한 API를 추출하여 '컴포넌트 API'를 생성하는 과정이다. 이 단계에서 소프트웨어 '컴포넌트 API'를 추출하기 위하여 필요한 입력정보로는 SW컴포넌트 내부 행위 기술서(SW-Component Internal Behavior Description)가 있다. 이 기술서에는 소프트웨어 컴포넌트 타입과 포트정보 및 인터페이스 등과 실제 소프트웨어 컴포넌트 기능을 지원하는 Runnable Entity정보 등을 포함하고 있다. RTE Generator는 이러한 정보에 기반하여 '컴포넌트 API'를 생성한다. 실제 생성된 '컴포넌트 API'는 컴포넌트 헤더파일로 소프트웨어 컴포넌트에서 데이터를 읽거나 쓰는 등 ECU내부 통신 또는 ECU와 ECU사이 통신에 필요한 API를 정의하고 있다.

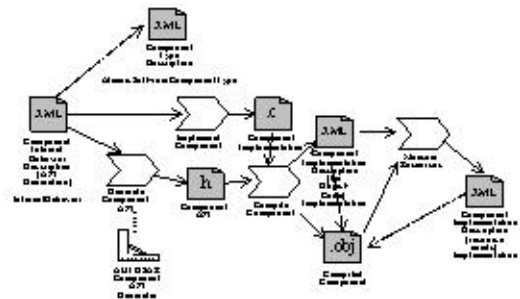


그림 2. RTE Contract 단계  
Fig. 2. RTE Contract Phase

소프트웨어 컴포넌트는 이 헤더파일을 포함하고 헤더파일에 있는 통신API를 호출하여 데이터를 주거나 받는 등 통신기능을 수행한다. 따라서 소프트웨어 컴포넌트에서는 실제 API가 어떻게 구현되었는지는 알 필요가 없다. 헤더파일에 정의된 '컴포넌트 API'는 RTE구현부분에서 구현되게 된다. 다음 그림 2는

RTE Contract 단계에서 어떤 입력정보를 이용하여 어떤 결과를 제공하는지를 보여주고 있다.

## 2.2 RTE Generation 단계

ECU설정이 모두 완료되면 RTE모듈을 생성할 수 있다. RTE는 컴포넌트모듈과 컴포넌트모듈 사이 통신뿐 만 아니라 컴포넌트 모듈과 하위 베이직 소프트웨어 모듈과의 통신기능도 제공하고 있다. 따라서 최종 RTE를 생성하기 전에 OS, COM등을 포함한 베이직 소프트웨어의 설정을 미리 알고 있어야 한다. ECU설정과정에서 이런 부분에 대한 설정이 이미 완료된 상태이므로 RTE Generator은 설정정보에 근거하여 RTE모듈을 생성할 수 있게 된다. 이 과정을 거치면 RTE는 상위 소프트웨어 컴포넌트에서 필요한 API를 제공할 뿐 아니라 하위 베이직 소프트웨어 모듈에서 필요한 API도 제공하게 된다. 상위 소프트웨어 컴포넌트는 RTE를 통하여 필요한 베이직 소프트웨어에 액세스할 수 있게 되고, 하위 베이직 소프트웨어도 RTE를 통하여 상위 소프트웨어 모듈에 액세스할 수 있다. 따라서 소프트웨어 컴포넌트, RTE 및 베이직 소프트웨어는 하나의 완벽한 ECU소프트웨어를 구성하여 ECU의 기능을 지원한다. 다른 그림 3은 RTE Generation 단계 과정을 보여준다.

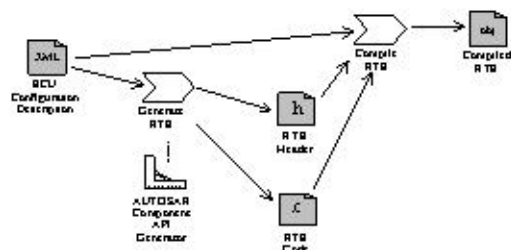


그림 3 RTE Generation 단계  
Fig 3 RTE Generation Phase

## 3. RTE 모듈

RTE Generation 단계를 거치면 RTE코드가 생성된다. 생성된 RTE모듈은 ECU설정에 따라 생성되는 생성코드와 라이브러리 코드 호출을 위한 표준화코드로 구성된다. 표준화코드는 필요에 따라서 다르게 구성할 수 있다. RTE헤더파일은 RTE.h이며 벤더모드, 공용어러 등을 정의한다. 수명주기(Lifecycle) 헤더파일은 RTE시작과 종료를 위한 Rte\_Start와 Rte\_Stop API를 정의한다. 어플리케이션 헤더파일은 RTE Contract 단계에서 설명한바와 같이 어플리케이션에서 호출될 RTE API를 정의한

다. AUTOSAR 타입 헤더파일은 RTE 데이터 타입을 정의한다. VFB 추적(Tracing) 헤더파일은 ECU 설정에서 설정된 VFB 추적 이벤트를 정의하며 RTE Generation 단계에서 생성된다. RTE 설정 헤더파일은 VFB 추적을 활성화 혹은 비활성화시키거나 RTE 훅(Hook) 정의 등을 포함하고 있다. RTE에서 정의한 RTE API는 Rte.c에서 구현된다. 이 파일에서는 소프트웨어 컴포넌트사이 통신API 컴포넌트 내의 Runnable Entity가 OS에 의해 액세스되어 실행되도록 하기 위한 TASK 바디(Task Body) 및 어플리케이션 호출에 사용되어질 각종 RTE이벤트를 생성한다. 어플리케이션에서 사용되어질 RTE API는 어플리케이션 헤더파일에서 정의되고 어플리케이션은 이 헤더파일을 포함하여 API를 호출한다. 각각의 RTE모듈사이 상관관계는 그림 4에 표시한 바와 같다.

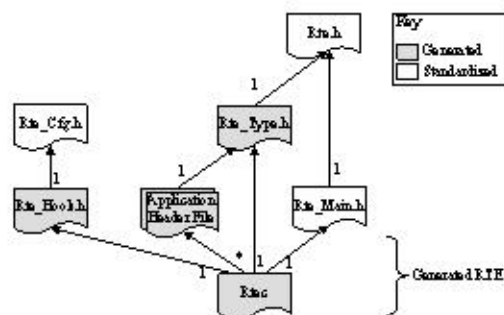


그림 4 RTE 모듈 상관관계  
Fig 4. Relationships between RTE Modules

## III. RTE Generator설계 및 구현

본 절에서는 앞에서 설명한 RTE생성방법을 참조하여 RTE를 생성할 수 있는 Generator을 설계한다. 생성된 RTE는 C소스코드형식으로 제공되며 어플리케이션 소스파일 및 베이직 소프트웨어 소스파일과 함께 수행가능한 바이너리 소스파일로 컴파일된다. AUTOSAR표준에서는 C/C++ 형식의 RTE파일을 지원하도록 하고 있으며 설계된 RTE Generator에서 생성하는 목표파일은 코드형식의 소스파일이다. RTE Generator은 크게 Generator 엔진, XML 파서, RTE 템플릿으로 구성된다. 여기서 Generator 엔진은 ECU설정을 분석하고 필요한 정보를 추출하며 설정에 근거하여 적당한 RTE 모

들을 생성하는 모듈이다. XML파서는 xml형식으로 저장된 ECU설정파일로부터 Generator 엔진에서 사용할 수 있는 설정정보를 추출해 내는 기능을 수행한다. RTE 템플릿은 Generator 엔진이 RTE모듈을 생성할 때 사용할 템플릿을 제공한다. Generator 엔진, XML 파서, RTE 템플릿사이의 상호관계는 다음 그림에 표시한 바와 같다.

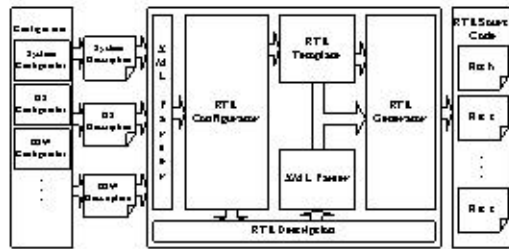


그림 5. RTE Generator 모듈 및 상관관계  
Fig. 5. RTE Generator Modules and Relationships

1. Generator 엔진

Generator 엔진은 초기화모듈, 템플릿 파서 모듈, 인쿠르드 파서 모듈, 데이터 파서 모듈로 구성된다. Generator 엔진 초기화 모듈은 주로 각종 글로벌 변수를 초기화하고 RTE생성에 필요한 출력경로 등 파라미터 값들을 정의하고 RTE소스코드를 저장할 디렉토리를 생성한다. 템플릿 파서 모듈은 초기화 모듈로부터 템플릿 파일명과 파일 경로 등 파라미터를 입력받아 템플릿 파일에 대한 처리를 실행한다. 이 과정에서 템플릿 소스파일에서 불필요한 주석 등을 지우고 헤더설명, 반복문 등을 처리해 주고 RTE모듈에 따라 템플릿 소스파일을 분할함으로써 RTE Generator 엔진에서 ECU 설정값을 입력하면 각각의 RTE모듈 소스가 생성될 수 있는 중간단계 템플릿 파일을 생성한다. 생성된 중간단계 템플릿 파일에는 인쿠르드 파일을 포함하고 있다. 생성된 RTE소스코드에 인쿠르드 파일을 포함시키기 위한 모듈로 인쿠르드 파서가 있으며 이 모듈은 중간단계 템플릿 파일에 포함된 인쿠르드 주석을 해석해주는 기능을 한다. 인쿠르드 파서를 거친 후 중간단계 템플릿 파일은 최종 소스코드와 같은 형태를 가지게 되며 XML 파서를 통하여 받은 ECU설정 파라미터를 넘겨받아 최종 소스코드를 생성한다. 데이터 파서 모듈은 최종 템플릿 파일에서 필요한 파라미터를 추적하여 ECU설정에 맞게 수정한다.

그림 6은 RTE Generator 엔진 구조도이다.

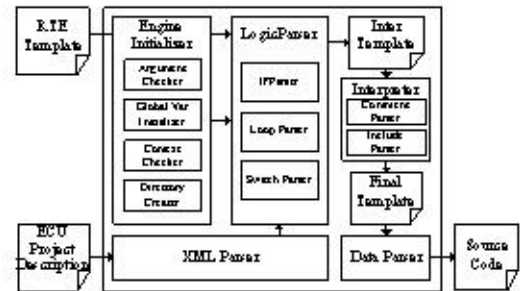


그림 6. RTE Generator 엔진  
Fig. 6. RTE Generator Engine

2. XML 파서

XML은 웹상에서 문서교환을 위하여 정의된 데이터 형식으로 W3C(World Wide Web Consortium)에서 정의한 표준이다. XML에서는 사전 정의된 태그만을 사용하는 것이 아니라 사용자가 필요에 따라 정의한 태그도 사용가능하다. ECU 프로젝트는 ECU설정에 관련된 내용을 XML형식으로 프로젝트 디렉토리 내에 저장한다. XML 파서는 XML문서에 대해 읽기/수정/조작이 가능한 XML문법 해석기이다. XML 파서를 이용하여 ECU 프로젝트 디렉토리 내의 XML파일을 분석하여 ECU설정관련 정보를 Generator 엔진에 제공한다. XML 파서에는 두 가지 종류가 있다. 하나는 트리형식에 기반한 XML 파서이고 다른 하나는 이벤트에 기반한 XML 파서이다. 트리형식에 기반한 XML 파서는 전체 XML문서에 대해 분석하고 각각의 트리노드로 접속하는 경로를 제공한다. 이벤트 기반의 XML 파서는 전체 XML문서를 여러 개의 이벤트 집합으로 간주하며 하나의 이벤트가 발생할 때마다 대응하는 함수를 호출하여 처리한다. 본 논문에서 사용하는 XML 파서는 다음과 같은 순서에 따라 동작한다. 우선 XML 파서 생성함수를 생성하여 XML 파서를 초기화하고 서로 다른 이벤트에 대한 핸들러 함수를 생성한다. 다음 XML 요소 핸들러를 이용하여 XML 문서에서 호출해야 할 함수를 지정해주고 XML 파서 함수를 이용하여 XML문서를 해석한다. 마지막으로 해석한 XML문서를 텍스트형식으로 출력하거나 에러가 발생하면 에러를 텍스트형식으로 출력해준다. 본 논문에 사용되는 XML 파서는 XML 문서에 대해 해석하고 그 결과값을 파라미터 형식으로 Generator 엔진으로 전달해주도록 한다.

3. RTE 템플릿

RTE 템플릿은 RTE에서 구현되어야 할 각종 기



능과 API를 템플릿형식으로 작성하여 RTE소스파일을 생성할 수 있도록 하는 파일이다. RTE는 기본적으로 RTE의 실행 및 종료를 지원하는 RTE 수명주기 모듈과 OS에 등록된 타스크를 대기(Waiting)/준비(Ready) 등 상태로 전환시키는 RTE 타스크 관리 모듈, 소프트웨어 컴포넌트들 및 소프트웨어 컴포넌트와 하위 베이직 소프트웨어 컴포넌트사이 S/R, C/S통신기능을 지원하는 RTE 통신관리모듈, AllIsrLevel, OSIsrLevel, TaskLevel등 서로 다른 종류의 독점영역을 지원해주는 RTE 영역관리모듈 및 RTE API에 대한 추적 기능을 지원해주는 RTE VFB 추적모듈로 구성된다. 그림 7은 상기의 RTE 기능을 지원할 수 있도록 설계된 RTE 템플릿의 구조도를 보여준다. 각 모듈의 상세기능은 다음과 같다.

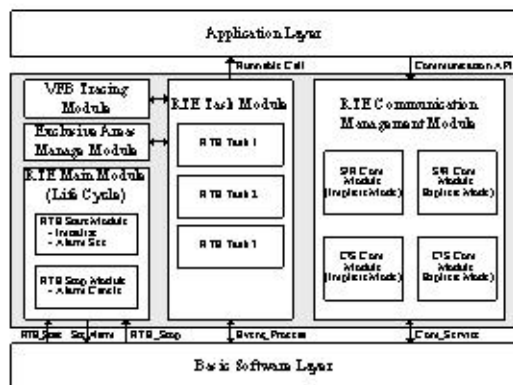


그림 7. RTE 템플릿 구조

Fig. 7. RTE Template Structure

### 3.1 RTE 수명주기 모듈

RTE 수명주기 모듈은 RTE를 실행 및 종료시키는 기능을 제공한다. 시스템 부팅 후 OS/서비스 모듈에서는 RTE 수명주기 모듈의 시작 기능을 호출하여 RTE를 실행시키며, 시스템을 종료하기 전 OS/서비스 모듈은 다시 RTE 수명주기 모듈의 종료기능을 호출하여 RTE를 종료시킨 후 시스템을 종료한다. RTE 수명주기 모듈은 RTE 개시 API와 RTE 퇴출 API로 구성되고, 기능은 아래와 같다.

- RTE\_Start: RTE를 초기화하고 필요한 자원을 할당한다.
- RTE\_Exit: RTE를 종료하고 시스템 및 통신 자원을 릴리즈 한다.

### 3.2 RTE 타스크 관리 모듈

RTE 타스크 관리 모듈은 OS에 등록된 타스크

를 대기/준비상태로 전환시켜 스케줄러에서 스케줄링이 가능하도록 하는 기능을 제공한다. 또한, RTE 내의 타스크를 등록하고 타스크내부의 이벤트를 처리하며 타스크를 스케줄링이 가능하도록 한다. 현재 RTE에서 지원되는 이벤트는 아래와 같다.

- 타임 이벤트
- 데이터 수신 이벤트
- C/S 이벤트

### 3.3 RTE 통신 관리 모듈

AUTOSAR는 S/R와 C/S 두 가지 종류의 통신 패턴을 지원한다. S/R통신 패턴에서는 Implicit와 Explicit통신방식을 지원한다. 이 모듈에서는 어플리케이션에서 정의된 통신방식을 판단하여 S/R Implicit통신과 S/R Explicit통신 API를 구현하여준다. C/S통신방식은 클라이언트 모듈에서 서버 모듈의 Runnable Entity를 호출하는 방식으로 구현한다.

### 3.4 RTE 영역 관리 모듈

이 모듈에서는 어플리케이션 Runnable Entity에서 진입(enter) 및 퇴출(exit)이 가능한 배타적 영역을 지정한다. 본 설계의 RTE 영역 관리 모듈에서는 AllIsrLevel, OSIsrLevel, TaskLevel 세 가지 종류의 배타적 영역을 제공한다.

### 3.5 RTE VFB 추적 모듈

RTE VFB는 VFB 레벨에서 컴포넌트 사이의 통신과 OS 및 COM 등에 대한 추적기능을 제공하는 RTE 특장적인 요소이다.

### 3.6 RTE 템플릿 파일

2장에서 설명한 구조별 RTE 모듈은 위에서 설명한 RTE의 모듈별 기능을 지원한다. RTE 템플릿은 기본적으로 RTE구조에 따라 각각의 템플릿을 생성하고 대응되는 RTE소스코드를 생성해준다. 표 1은 RTE Generator 지원을 위한 각각의 RTE 템플릿에 대해 설명한다. RTE의 주 기능은 Rte\_Template파일에서 구현되어진다. 이 템플릿 파일은 타스크 바디, 소프트웨어 컴포넌트에 필요한 통신 API 및 각종 RTE 이벤트 등을 생성해주는 파일이다. RteHead\_Template은 Rte.c에서 사용되어지게 될 각종 내부성명 등을 정의한다. RteComponent\_Template은 소프트웨어 컴포넌트에서 사용되어지게 될 RTE 통신 API를 정의하며 소프트웨어 컴포넌트에서 사용되어진다.

표 1 RTE 템플릿 파일 종류

Table 1. RTE Template File Types

Template file	Description
RteComponent_Template	RteComponent_Template은 소프트웨어 컴포넌트에 포함될 어플리케이션 헤더파일을 생성하는 템플릿
Rte_Template	RTE_Template은 RTE API의 실행기능을 구현한 Rte.c코드를 생성하는 템플릿 파일
RteHeader_Template	RTE.h헤더파일생성을 위한 템플릿으로 RTE에서 사용될 내부성명 등을 정의
RteMain_Template	Rte_Main.h파일을 생성해주는 템플릿으로 RTE 수명주기 API를 생성
RteCfg_Template	Rte_Cfg.h파일로 VPB 추적 여부 등 RTE 설정에 관련된 모든 생성
RteCbK_Template	Com Callback 등 RTE 추의함수를 정의한 Rte_Cbk.h를 생성
RteHook_Template	Rte_Hook.h을 생성하며 VPB 추적 API를 생성
RteType_Template	RTE에서 정의한 자체 데이터 타입 생성

4. RTE요구사항 및 지원

RTE의 정의 및 기능으로부터 RTE는 OS COM 등 베이직 소프트웨어 및 상위 어플리케이션 소프트웨어와의 상호작용에 대한 기능을 제공해야 할 수 있다. 다음 표 2는 AUTOSAR표준에서 정의한 주요 요구사항 및 본 설계에서 요구사항을 만족하기 위한 설계방법에 대해 설명하였다. 이 외에도 C소스코드에 대한 지원, VPB추적 지원 등에 대한 요구사항이 있으며 이는 C소스타일의 RTE템플릿, VPB모듈 등을 통하여 지원한다.

표 2 RTE 요구사항 및 지원

Table 2. RTE Requirements and Supporting

주요 요구사항	지원방법
OS지원	<ul style="list-style-type: none"> <li>- AUTOSAR OS의 액세스</li> <li>- TASK() API를 이용하여 OS API 호출</li> <li>- TASK() API를 이용하여 TASK() API를 이용하여 OS API 호출</li> <li>- 어플리케이션과 OS계층사이 디커플링</li> <li>- RTE에서 모든 OS API 처리</li> </ul>
COM지원	<ul style="list-style-type: none"> <li>- 표준화 COM프 로토콜 지원</li> <li>- 표준화 API를 이용하여 COM API 호출</li> </ul>
어플리케이션과 지원	<ul style="list-style-type: none"> <li>- Runnable Entity 활성화</li> <li>- RTE이벤트를 이용하여 Runnable Entity 호출</li> <li>- Inter Runnable Variable 지원</li> <li>- 데이터 공유체를 이용하여 다중 Runnable Entity의 동일 데이터 액세스 지원</li> <li>- 비타적 영역 지원</li> <li>- 비타적 영역 모두 설계</li> </ul>

IV. RTE 생성 및 검증

설계 및 개발한 RTE의 동작을 검증하기 위하여 응용프로그램을 설계하고 설계한 응용프로그램과 RTE Generator를 이용하여 RTE 일부 모듈을 생성하여 요구하는 RTE모듈이 생성되는지를 확인하여 RTE Generator의 유효성을 검증한다.

AUTOSAR 소프트웨어 개발을 위해서는 기능별로 다수의 설계 및 설정도구가 필요하다. 상위 어플리케이션의 개발을 위한 도구로는 벡터사의 Davinci Software Architect, Mathworks사의 Simulink, dSpace사의 SystemDesk 등 도구가 있다. 이 도구들을 이용하여 소프트웨어 아키텍처를 설계하고 필요한 소프트웨어 컴포넌트 및 컴포넌트 내의 Runnable Entity를 설계한다. 베이직 소프트웨어 설계도구로는 벡터사의 Oil Configurator 도구, GENy CAN 설정도구 등이 있다. 어플리케이션의 설계에 따라 베이직 소프트웨어를 설정하며 상위 어플리케이션과 베이직 소프트웨어 설정을 입력으로 RTE를 생성한다. 타겟보드를 선정한 후 타겟보드에 맞는 컴파일도구(예하면 Freescale사의 CodeWarrior for H(C)S) 12)를 이용하여 생성된 어플리케이션 소스, RTE 소스 및 베이직 소프트웨어 소스를 컴파일하면 타겟보드에 포팅가능한 이미지파일을 생성할 수 있다.

RTE생성을 위하여 다음과 같은 간단한 어플리케이션을 설계한다. 전체 어플리케이션은 2개의 ECU로 구성되며 ECU1은 1개의 소프트웨어 컴포넌트, ECU2은 2개의 소프트웨어 컴포넌트를 포함한다. 다음 그림은 설계된 어플리케이션의 설명도이다. InstrumentCluster ECU는 DisplayInfo 컴포넌트를 포함하고 TyrePressure ECU는 WarningAlgorithm과 SendValuePreparation 두 개의 컴포넌트를 포함한다. SensorValuePreparation 컴포넌트는 PhysicalValue포트를 이용하여 압력데이터와 온도데이터를 나타내는 PressureData와 TempData를 전송한다. WarningAlgorithm 컴포넌트는 이 데이터를 수신하여 Warning여부를 판단한 후 DisplayInfo에 Warning데이터를 전송한다. DisplayInfo컴포넌트는 수신한 Warning데이터를 나타내준다. ECU1과 ECU2의 어플리케이션 계층은 Davinci프로그램을 이용하여 설계한다. 어플리케이션 아키텍처 설계 완료 후 시스템 설정에 관련된 모든 정보는 xml형식으로 프로젝트 폴더에 저장된다.

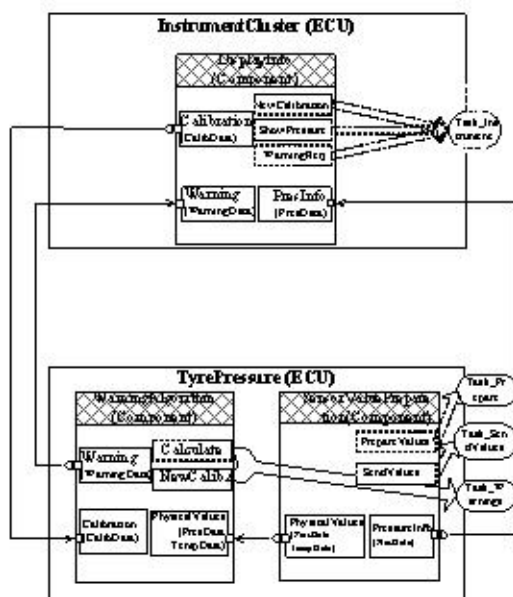


그림 8. 어플리케이션 구조도  
Fig. 8 Application Structure

PressureData와 PressureData를 판단한 후 리턴되는 WarningData를 전송하기 위하여 어플리케이션 설계 시 uint8형식의 PressureData 데이터 타입과 bool형식의 WarningData 두 가지 자체 데이터 타입을 정의하였다. 그림 9는 생성된 RTE 타입 소스이다. RTE Generator은 어플리케이션의 설정에 따라 RTE에 필요한 데이터 타입을 정상적으로 생성해줄 수 있다.

```

.....
r: Inorder according to order of AUTOSAR type and RTE structure (Generated by RTE Generator)
.....
RTEs: RTE_Type
WorkSpace: C:\Program Files\Samsung Electronics\AUTOSAR\Tool\Generator\Tool\Task
VehicleProject: Data\Autosar\1111
ECUProject: Instrument
Identifier: 661
ISAPrivator: 661
TaskImpl: r:\type\gen\ec
Generator: 142 5424 109117264
.....
r: available parameters:
RTE: RTE_Type, J
Package: RTE_Type, J
Package: RTE_Type, J
.....
* ECU specific data:
.....
open: block: CalibrationData;
open: block: PressureData;
open: block: WarningData;
.....

```

그림 9. 생성된 RTE 데이터 타입  
Fig. 9 Generated RTE Data Type

## V. 결론

본 논문에서는 AUTOSAR기반의 차량 전장 소프트웨어 개발에 있어서 필요한 RTE생성도구를 설계하고 구현하였다. AUTOSAR기반의 전장 소프트웨어 개발에 있어서 RTE는 설계된 상위 소프트웨어 컴포넌트와 설정된 하위 베이스 소프트웨어에 따라 동적으로 구성됨으로써 어플리케이션 계층, RTE계층, 베이스 소프트웨어 계층으로 구성된 전체 소프트웨어가 동작가능하게 한다. 본 논문에서 설계된 RTE Generator은 시스템 설정정보를 입력정보로 하면 상위 소프트웨어 설계정보, 하위 베이스 소프트웨어 설정정보를 추출하여 자동으로 RTE를 생성해주는 기능을 제공한다. 설계된 RTE Generator은 Generator 엔진과 C레벨의 RTE 템플릿 2원화 구조로 구성되어 AUTOSAR표준안 변경에 따른 표준반영이 쉬워지고 효율성과 타 AUTOSAR도구의 확장성이 강하다는 장점을 가지고 있다. 앞으로 미 구현된 RTE기능과 업데이트된 AUTOSAR RTE 표준을 반영하여 RTE Generator 기능을 지속적으로 향상시키고 RTE기타 모듈에 대한 검증작업도 지속되어야 할 것이다.

## 참고문헌

- [1] AUTOSAR, "Technical Overview", 2006.
- [2] AUTOSAR, "Specification of RTE Software", 2006.
- [3] AUTOSAR, "Methodology", 2006.
- [4] W.S.Yoo, J.Y.Park, J.H.Yoo, S.H.Kim and S.S.Hong, "AUTOSAR-based Software Architecture for Automotive Systems", KSAF 2006 Symposium, pp.60-65, 2006.
- [5] S.S.Hong, J.Y.Park and W.S.Yoo, "Technology Trends in Automotive OS and Middleware: OSEK and AUTOSAR", Journal of the Korean Society of Precision Engineering, Vol.21, 2006.
- [6] H. Bernd, K. Thorsten and K. Adreas, "Reuse of Software in Distributed Embedded Automotive Systems", Proceedings of the 4th ACM International Conference on Embedded Software, ACM Press, pp.203~210, 2004.
- [7] OSEK, "OSEK/VDX Operating System v2.2.3", 2005.



[8] AUTOSAR, "Specification of Communication v.2.0.1", 2006.

[9] AUTOSAR, "Requirement on Operating System v.2.0.1", 2006.

## 저 자 소 개

### 박세권

1999년 북경대학교 전자학과 학사. 2002년 영남대학교 정보통신공학과 석사. 2004년 영남대학교 정보통신공학과 박사수료. 현재, 대구경북과학기술연구원 연구원.



관심분야: 임베디드 소프트웨어, 차량용 소프트웨어 미들웨어.

Email: sqpiao@dgist.ac.kr

### 조현철

2005년 한양대학교 전자전기컴퓨터 학사. 2007년 서울대학교 신호처리 석사. 현재, 대구경북과학기술연구원 연구원.



관심분야: 무선통신시스템, 음성식별, 차량용 표준 플랫폼.

Email: hcjo@dgist.ac.kr

### 조성래

2000년 영남대학교 컴퓨터공학과 석사. 현재, 대구경북과학기술연구원 선임연구원.



관심분야: 차량용 임베디드 SW 플랫폼, 컴포넌트 기반 소프트웨어 개발방법론.

Email: srcho@dgist.ac.kr

### 류현기

2005년 경북대학교 대학원 컴퓨터공학 석사. 현재, 대구경북과학기술연구원 연구원 및 경북대학교 전기전자컴퓨터공학부 박사과정.



관심분야: 임베디드 소프트웨어, 네트워크 QoS, 멀티미디어

Email: hkryu@dgist.ac.kr

### 정우영

1984년 서울대학교 전기공학과 학사. 1986년 한국과학기술원 전기 및 전자학과 석사. 1991년 한국과학기술원 전기 및 전자학과 박사. 현재, 대구경북과학기술연구원 책임연구원.



관심분야: 차량용 통신 기술, 임베디드 시스템, 차량용 표준 플랫폼

Email: wyjung@dgist.ac.kr