

# AUTOSAR 기반 차량용 소프트웨어의 컴포넌트 모델링 도구

박인수<sup>†</sup> · 이정선<sup>\*\*</sup> · 조성래<sup>\*\*\*</sup> · 정우영<sup>\*\*\*\*</sup> · 이우진<sup>\*\*\*\*\*</sup>

## 요약

최근 자동차 산업에서는 하드웨어 부품에 대한 연구뿐만 아니라 하드웨어를 제어하는 임베디드 소프트웨어에 대한 연구가 점차 증가하고 있다. 기존 임베디드 소프트웨어 개발은 자동차 제작사별로 독자적으로 이루어져 소프트웨어 재사용성, 상호운용성 등의 문제점이 대두되었다. AUTOSAR는 이러한 문제점을 해결하기 위해 제시된 컴포넌트 기반의 자동차 소프트웨어 아키텍처 표준이다. AUTOSAR에서는 다양한 다이어그램들을 작성하고 이들간의 연관성을 유지시켜야 하므로 도구 지원이 요구된다. 현재 몇몇 외국기업에서 고가의 AUTOSAR 지원 개발도구의 프로토타입을 선보이고 있다. 이 연구에서는 이클립스 GMF 기술을 이용하여 AUTOSAR 3.0 표준을 충실히 따르면서 기존 도구들의 단점을 보완하는 차량용 소프트웨어의 컴포넌트 모델링 도구를 개발한다. 개발된 도구는 크게 그래픽 기반의 컴포넌트 모델링 도구와 네트워크 토폴로지 도구로 구성된다. 이러한 도구들은 직접 구현된 것이 아니라 개발 업체에 맞게 손쉽게 이클립스 GMF로 생성할 수 있으므로 개발업체 맞춤형 및 국제표준 발전에 쉽게 발맞출 수 있는 장점이 있다.

키워드 : 차량용 소프트웨어, 컴포넌트 모델링 도구, 컴포넌트 다이어그램, 네트워크 토폴로지 다이어그램, AUTOSAR, GMF

## A Component Modeling Tool based on AUTOSAR for Automotive Software

In Su Park<sup>†</sup> · Jung Sun Lee<sup>\*\*</sup> · Sung Rae Cho<sup>\*\*\*</sup> · Woo Young Jung<sup>\*\*\*\*</sup> · Woo Jin Lee<sup>\*\*\*\*\*</sup>

## ABSTRACT

Recently, in automotive industry, there have been many researches related with hardware components and embedded software which controls hardware components. Since most of embedded software is tightly dependent on car manufacturers, there were some problems in reusability and interoperability of automotive software. In order to solve these problems, AUTOSAR standardized the component-based software architecture of automotive software. In AUTOSAR, several modeling diagrams should be described and their dependencies are also checked. Currently, a few company developed the prototypes of tools supporting AUTOSAR. In this paper, a component modeling tool based on AUTOSAR 3.0 is developed for enhancing the usability of existing tools using Eclipse GMF. The tool is composed of a graphical component modeling tool and a graphical network topology tool. Since these tools are generated based on GMF without hard coding, it is relatively easy to customize the tools for adopting company's needs and easy to follow the improvement of the standard and development environments.

Keywords : Automotive Software, Component Modeling Tool, Component Diagram, Network Topology Diagram, AUTOSAR, GMF

## 1. 서론

최근 자동차 분야에서 혁신을 이루고 있는 기술의 90%가 전자 장비와 임베디드 소프트웨어에 관련된 분야이다[1]. 일부 고급 차량의 내부에는 70개 이상의 ECU(Electronic Control Unit)가 들어가고 각 ECU들은 10가지 이상의 다른 버스 시스템으로 연결된다[2]. 그리고 향후 고급 기종의 자동차에는 1G 정도의 소프트웨어 코드가 포함될 것으로 예상된다[3]. 하지만 기존 자동차 소프트웨어 모듈들은 자동차 제작사별

※ 본 연구는 교육과학기술부의대구경북과학기술원 기관고유사업의 일환으로 수행하였음[2008IT2, 차량용 임베디드 시스템 기술 개발]  
† 정 회 원 : 삼성전자 무선사업부 사원  
\*\* 정 회 원 : LG CNS 사원  
\*\*\* 정 회 원 : 대구경북과학기술원 선임연구원  
\*\*\*\* 정 회 원 : 대구경북과학기술원 책임연구원  
\*\*\*\*\* 정 회 원 : 경북대학교 IT대학 컴퓨터학부 부교수  
논문접수 : 2010년 3월 23일  
수정일 : 1차 2010년 6월 8일, 2차 2010년 7월 2일  
심사완료 : 2010년 7월 7일

로 독자적으로 개발이 이루어지고 있어 소프트웨어 재사용성 및 상호운용성, 생산성 저하 등의 문제점이 대두되었다. 이러한 문제점들을 해결하기 위한 노력의 일환으로 공통 자동차 소프트웨어 아키텍처인 AUTOSAR(AUTomotive Open System ARchitecture)[4]를 지정하고 이를 국제 표준으로 채택하였다.

AUTOSAR는 컴포넌트 기반 개발에 기반을 둔 모델 기반 개발방법론[5]으로 먼저 논리적인 관점에서 시스템의 컴포넌트들을 식별하고 컴포넌트간의 관계를 다이어그램으로 표현한다. 개념적인 관점에서의 컴포넌트 개발이 완성되면 ECU와 네트워크로 구성된 실제 시스템 아키텍처에 매핑한다. ECU간의 연결은 토폴로지 네트워크를 이용하여 네트워크 특성 등을 기술하고 논리적 컴포넌트들은 각각의 ECU로 매핑된다. AUTOSAR의 모델 기반 개발 방법은 다양한 형태의 다이어그램들을 작성하고 이들간의 매핑 등을 제공하여야 하므로 이러한 작업을 효율적으로 수행하기 위해서는 AUTOSAR 표준을 따르는 모델링 도구들의 지원이 필수적이다. 현재 AUTOSAR에 연관된 연구로는 AUTOSAR 플랫폼에 관한 연구[6-8], AUTOSAR 지원 도구 연구[9-11], AUTOSAR의 요소기술 개발[12]로 나눌 수 있다. AUTOSAR 지원 도구로는 dSPACE사의 SystemDesk[9], Vector사의 DaVinci[10], Mentor Graphics사의 Vehicle System Architect[11] 등이 있다. 이러한 도구들은 현재 프로토타입 수준으로 직접 프로그래밍 언어로 구현되어 있으며 공개소스 방식으로 구현되어 있지 않아서 사용자 측면의 도구 맞춤화가 불가능하며 AUTOSAR 표준을 그대로 따르고 있는지 여부도 불확실하여 타도구와의 상호운용성도 떨어지는 단점이 있다.

이 논문에서는 이클립스 GMF(Graphical Modeling Framework) 개발 방법[13]을 이용하여 AUTOSAR 표준을 따르는 차량용 컴포넌트 소프트웨어 모델링 도구를 제안한다. 이 모델링 도구는 AUTOSAR 표준의 기본 개념 중에서 응용 소프트웨어 컴포넌트를 모델링하는 단계와 모델링한 컴포넌트를 각 ECU의 RTE(Runtime Environment) 상황과 네트워크 토폴로지를 고려하여 ECU로 매핑하는 단계를 지원한다. 즉, GUI 기반의 컴포넌트 모델링 도구, 네트워크 토폴로지 다이어그램 생성도구와 ECU와 컴포넌트 매핑 기능 등을 지원한다. 이클립스 GMF를 이용하여 다이어그램 편집기를 개발하므로 우선 컴포넌트 다이어그램과 네트워크 토폴로지 다이어그램의 메타 모델을 정의하고 이를 기반으로 GMF 프레임워크를 이용하여 다이어그램 편집기를 생성한다. 그리고 GMF로 생성되지 않는 부분들은 직접 프로그래밍하여 다이어그램 편집기와 연계하여 동작하게 한다. 이 논문에서 제공하는 AUTOSAR 지원 도구들은 구현언어로 직접 구현되지 않고 AUTOSAR 표준에 제공되는 메타모델을 기반으로 GMF를 이용하여 다양한 다이어그램 편집기들을 생성하는 방법을 이용한다. 이러한 생성 방법에서는 도구의 아이콘 모양, 레이아웃, 노드의 모양 크기 등을 맞춤화할 수 있기 때문에, 사용자의 요청에 따라 새롭게 생성하여 제공할 수 있는 장점이 있다. 그리고 AUTOSAR 표준을 그대로 따르

고 있으므로 표준을 따르는 도구간의 상호운용성도 만족한다.

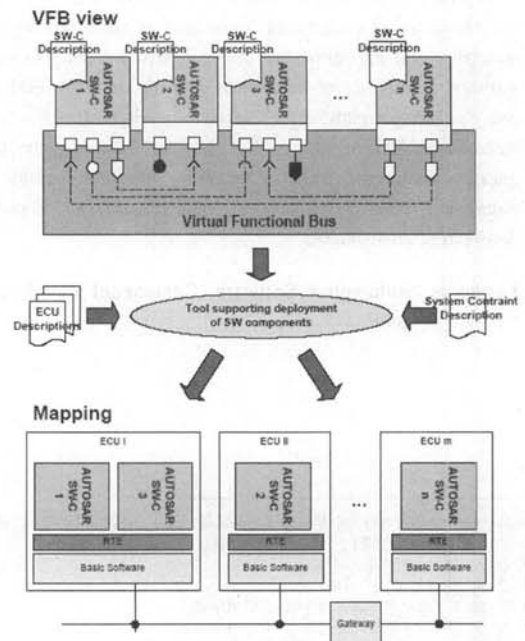
논문의 구성은 다음과 같다. 먼저 제 2 절에서 AUTOSAR에 대한 개요와 대표적인 상용화 도구인 dSPACE의 SystemDesk, Vector사의 DaVinci, Mentor Graphics사의 Vehicle System Architect에 대해 기술한다. 제 3 절에서 차량용 소프트웨어 컴포넌트 모델링 도구에 대한 메타 모델을 작성하고 전체 모델링 도구의 설계를 제공한다. 제 4 절에서 이클립스 GMF를 이용하여 컴포넌트 다이어그램 편집기와 네트워크 토폴로지 편집기를 생성하며 다양한 속성값을 설정하는 속성 창을 개발하여 생성된 다이어그램 편집기에 연동한다. 제 5 절에서 간단한 응용시스템을 대상으로 개발된 차량용 소프트웨어 모델링 도구의 유용성과 사용자 GUI 맞춤화 가능성을 언급한다. 제 6 절에서 결론을 맺고 향후 연구에 대해 기술한다.

## 2. 연구배경

### 2.1 AUTOSAR의 개요

지금까지 자동차 소프트웨어 모듈들은 자동차 제작사별로 독자적으로 개발이 이루어지고 있어 소프트웨어 재사용성, 생산성 저하 등의 문제점이 대두되었다. 이러한 문제점들을 해결하기 위한 노력의 일환으로 AUTOSAR라는 공통 소프트웨어 아키텍처를 국제 표준으로 채택하였다. AUTOSAR는 자동차 제작사와 부품업체 톨 제작업체가 협조하여 2004년부터 표준화 작업을 시작하였고 2007년 말 Release 3.0을 공개하였으며 이를 바탕으로 자동차 ECU를 개발하는 것이 가능하게 되었다.

(그림 1)은 AUTOSAR 표준의 기본 개념을 보여준다. 먼



(그림 1) AUTOSAR의 차량 컴포넌트 소프트웨어 개발 프로세스

〈표 1〉 AUTOSAR 지원 개발 도구

도구명	개발사	구현언어	상호운용성	사용자 맞춤화
SystemDesk	dSpace	Unknown	XML Import/Export	Not Supported
Davinci Tool Suite	Vector	Unknown	XML Import/Export	Not Supported
Vehicle System Architect	Mentor Graphics	Unknown	Unknown	Not Supported
Our Approach	Non-commercial	Open Source (Eclipse, Java)	Standard Meta Model	User Customization

저 VFB(Virtual Functional Bus) 관점에서 소프트웨어 컴포넌트들을 정의하고 컴포넌트의 인터페이스 등을 SW-C 명세에 기술하고 이들 소프트웨어 컴포넌트 간의 상호 연관성을 나타낸다. 이후, ECU 명세와 시스템 제약사항 명세를 참조하고 SW 개발 도구를 활용하여 SW 컴포넌트를 구현한다. 그 다음 단계는 구현된 컴포넌트들을 ECU에 배치하는 단계로 각 ECU의 RTE 상황을 고려하여 SW 컴포넌트들을 맞춤화 과정을 거친 다음 ECU로 매핑하여 배치한다.

## 2.2 AUTOSAR 지원 도구

현재 AUTOSAR를 지원하는 개발 도구로는 dSPACE의 SystemDesk[9], Vector의 Davinci Tool Suite[10], Mentor Graphics사의 Vehicle System Architect[11]가 있다. dSPACE에서 개발된 SystemDesk는 복잡한 분산 환경에서 자동차용 전자 시스템의 설계, 구현, 통합을 위한 아키텍처 기반의 시스템 디자인 도구이다. SystemDesk는 AUTOSAR 표준에 따른 소프트웨어 컴포넌트를 이용하여 소프트웨어 아키텍처를 모델링한다. 각 소프트웨어 컴포넌트는 시스템을 기능 단위로 나누어 독립적으로 개발될 수 있으며, 소프트웨어 재사용을 지원한다. 개발자가 소프트웨어 아키텍처를 설계할 시에 하드웨어 토폴로지를 고려하지 않도록 한다. 전체 시스템은 소프트웨어 아키텍처, 하드웨어 토폴로지, 네트워크 커뮤니케이션, 소프트웨어 컴포넌트의 하드웨어 매핑 단계로 각각 분리되어 모델링된다.

Vector사에서 개발된 DaVinci는 AUTOSAR 표준을 기반으로 하는 디자인 도구이다. DaVinci를 이용해서 차량용 분산 시스템을 설계할 수 있고, ECU Application을 개발할 수 있다. DaVinci는 차량의 분산되어 있는 기능을 디자인하고 소프트웨어 컴포넌트를 정의한다. 그리고 네트워크로 연결되어 있는 ECU들의 코드를 통합한다. DaVinci Design 부분은 DaVinci System Architect, DaVinci Network Designer 도구로 구성되어 있다. 이 두 도구는 동일한 데이터 모델을 기초로 동작하며, 독립적으로 사용할 수 있다. DaVinci System Architect는 분산된 차량용 시스템의 소프트웨어 아키텍처를 기술할 수 있다. 그리고 DaVinci Network Designer는 CAN(Controller Area Network), LIN(Local Interconnect Network), FlexRay 등의 버스 시스템을 이용하여 분산된 ECU 시스템들 간의 네트워크 아키텍처를 설계할 수 있다.

Mentor Graphics사의 Vehicle System Architect는 기능적인 관점을 기술하는 Vehicle Function Editor, 분산 네트워크의 실제 아키텍처를 기술하는 Vehicle Topology Editor,

기능을 실제 ECU에 매핑하는 Vehicle Function Mapper, ECU간의 CAN, LIN, FlexRay 클러스터 통신을 관리하는 Network Cluster Builder 등의 단위 도구들로 구성되어 있다.

AUTOSAR에서는 기능적 관점의 컴포넌트 기술, ECU 중심의 네트워크구조 설계, 기능 컴포넌트의 매핑, 코드자동 생성 등의 여러 개발 단계를 정의하고 있다. 각각의 단계를 효율적으로 지원하기 위해서는 CASE 도구의 지원이 필요하며 이러한 도구 간에는 상호운용성이 보장되어야 한다 [14]. 그리고 기본적으로 CASE 도구 사용자들은 개인적인 GUI 선호도가 다양하므로 도구의 맞춤화 기능이 요구된다. <표 1>는 기존 AUTOSAR 지원도구들을 구현언어, 상호운용성, 사용자 맞춤화 측면에서 비교한 것이다. 기존도구들을 상용으로 만들어져 있어 구현언어가 알려져 있지 않으며 단위 도구간의 상호운용성과 사용자 맞춤화 부분이 취약한 것으로 파악된다. 이 연구에서는 AUTOSAR 표준에 따라 도구간의 상호운용성을 보장하며 공개 소스 기반의 개발하여 사용자 맞춤화가 가능한 도구를 개발하고자 한다.

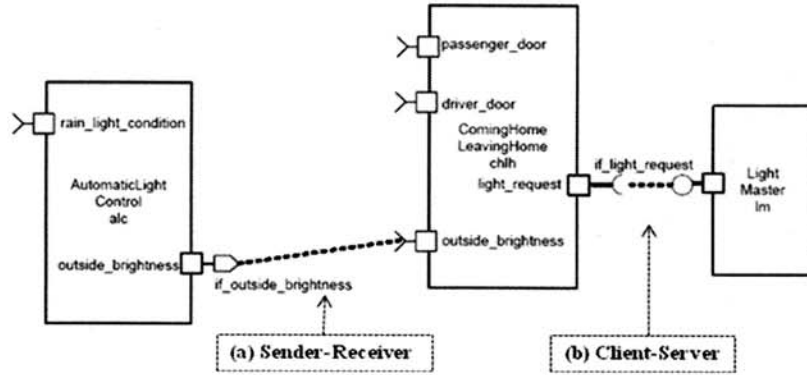
## 3. 차량용 소프트웨어 컴포넌트 모델링 도구의 설계

다이어그램 편집기를 생성하기 위해서는 먼저 다이어그램의 편집기의 메타 모델이 명확히 정의되어야 한다. 이 절에서는 AUTOSAR 3.0 표준 명세를 분석하여 소프트웨어 컴포넌트 다이어그램 편집기와 네트워크 토폴로지 편집기에 대한 메타 모델을 정의하고 소프트웨어 컴포넌트 모델링 도구를 설계한다.

### 3.1 컴포넌트 다이어그램 편집기의 메타 모델 정의

AUTOSAR의 컴포넌트 다이어그램은 시스템의 서브시스템을 컴포넌트로 나타내고 컴포넌트 간의 연계관계를 표현한다. AUTOSAR 소프트웨어 컴포넌트는 포트(port), 인터페이스(interface) 등으로 명확하게 표준 인터페이스를 표현한다. 포트는 컴포넌트의 인터페이스를 연결하는 연결점으로 PPort와 RPort가 있다. PPort는 제공 인터페이스와 연결되어 있으며 RPort는 요청 인터페이스와 연결되어 있다. 두 컴포넌트 간의 상호작용은 PPort에 연결된 제공 인터페이스와 RPort에 연결된 요청 인터페이스를 연결하여 나타낸다. (그림 2)는 AUTOSAR 컴포넌트 다이어그램의 예를 보여준다

AUTOSAR의 통신 패턴으로는 송신자-수신자(sender-receiver) 패턴과 클라이언트-서버 패턴이 있다. (그림 2)-(a)에 나타난 바와 같이 송신자-수신자 방식은 송신자가 데이

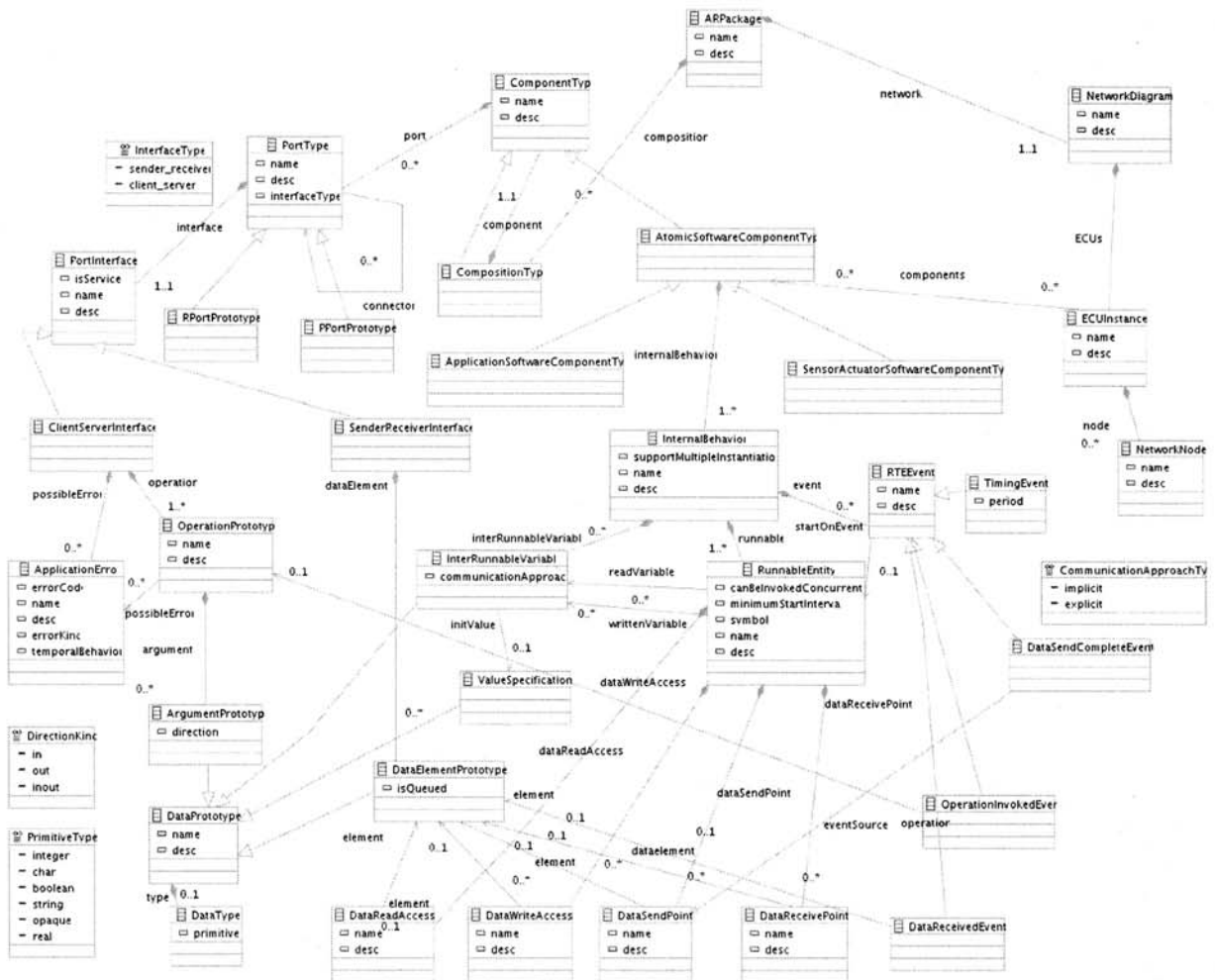


(그림 2) AUTOSAR 컴포넌트 다이어그램의 예

터 수신자에게 전송하는 방식으로 비동기적 데이터 전송 기능을 제공한다. (그림 2)-(b)에 나타난 바와 같이 클라이언트-서버 패턴은 분산 시스템에서 널리 사용하는 메시지 전송방식으로 서버는 서비스를 제공하고 클라이언트는 제공된 서비스를 이용한다.

소프트웨어 컴포넌트 다이어그램 편집기의 메타 모델은 크게 세 부분으로 나눌 수 있다. 편집기에 컴포넌트와 포트

등과 같이 노드 형식으로 그려지는 부분, 그려지지 않지만 속성 다이얼로그로 표시되는 인터페이스 내용 부분과 내부 행위 부분으로 구성된다. (그림 3)은 컴포넌트 다이어그램 편집기의 주요 구성요소들간의 관계를 표현한 메타 모델의 클래스 다이어그램을 보여주고 있다. 컴포넌트 다이어그램에 나타나는 여러 컴포넌트 타입, 연관관계 타입, 포트, 여러 인터페이스 타입, 컴포넌트 내부 행위, 그리고 기본적인



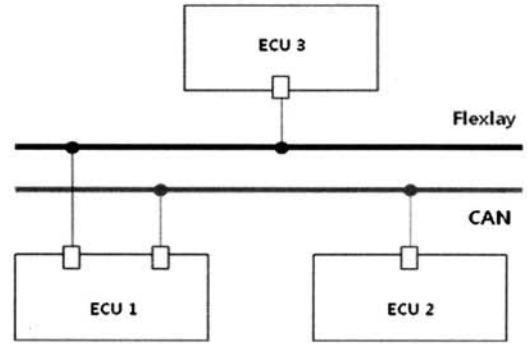
(그림 3) 컴포넌트 다이어그램 편집기의 메타모델클래스 다이어그램

자료형, 오퍼레이션, 형식인자 등의 구성요소의 관계를 나타낸다. 이러한 메타모델은 이클립스 GMF를 이용한 다이어그램 편집기의 입력으로 활용된다.

3.2 네트워크 토폴로지 편집기의 메타 모델 정의

네트워크 토폴로지 다이어그램은 ECU와 네트워크 버스의 연결 관계를 표현한다. (그림 4)는 AUTOSAR 표준에서 기술하고 있는 네트워크 토폴로지에 대한 표기법을 나타낸다. ECU는 실제로 차량용 시스템에 들어가는 전자 장비이다. 네트워크 토폴로지 다이어그램에서 ECU는 하나의 컴포넌트 형태로 표기되며, 실제로 다이어그램에 표기되지 않지만 내부적으로 매핑되는 컴포넌트에 대한 정보를 가진다. 또한 내부적으로 네트워크 버스를 통해 주고받는 TX, RX 메시지에 대한 정보를 가진다. 네트워크 버스는 CAN, LIN, FLEXRAY 등으로 구분된다. 네트워크 버스는 하나의 선으로 표기되며 내부적으로 버스를 통하여 전송되는 네트워크 메시지를 정의한다.

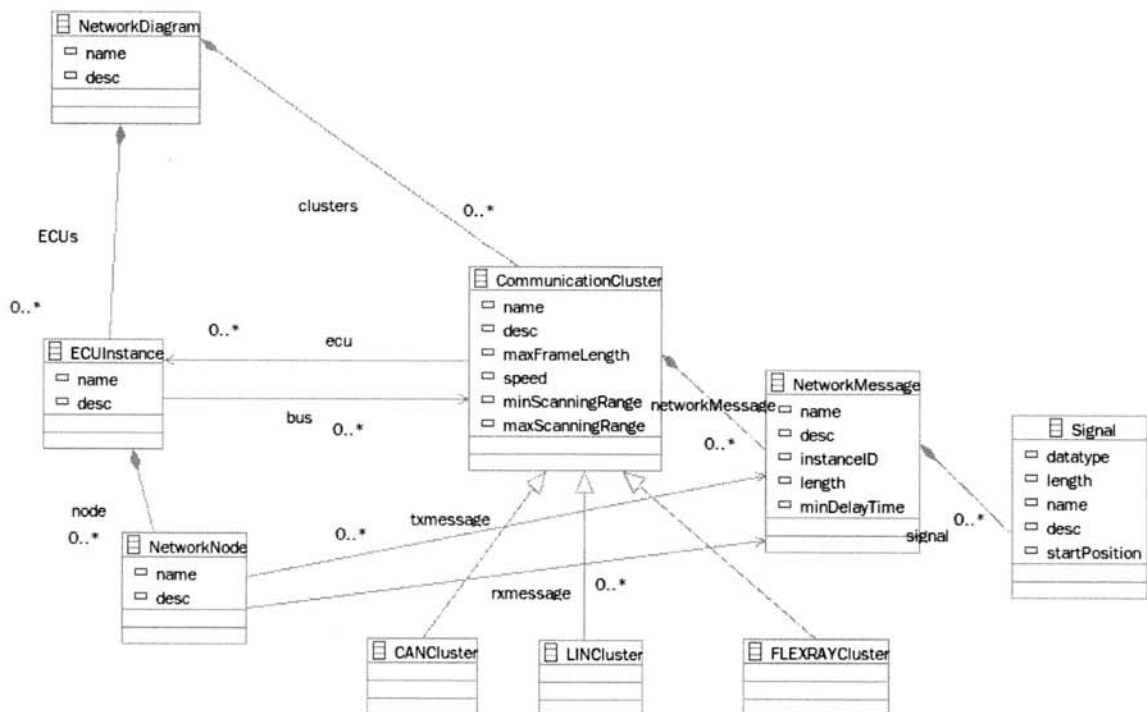
네트워크 토폴로지 다이어그램 편집기의 메타 모델은 크게 두 부분으로 나눌 수 있다. 편집기에 ECU와 통신클러스터 등의 노드로 그려지는 부분과 노드로 그려지지 않지만 속성창으로 표시되는 네트워크 속성 부분이다. 네트워크 속성으로 정의되는 네트워크 메시지와 네트워크 노드는 각각 통신 클러스터와 ECU에 포함된다. 네트워크 시그널은 네트워크 메시지에 포함되어 실제 전달되는 신호이다. (그림 5)는 네트워크 토폴로지의 핵심 구성요소인 ECU, 네트워크 노드, 통신 클러스터, 네트워크 메시지, 시그널 등의 연관 관계를 클래스 다이어그램으로 보여주고 있다.



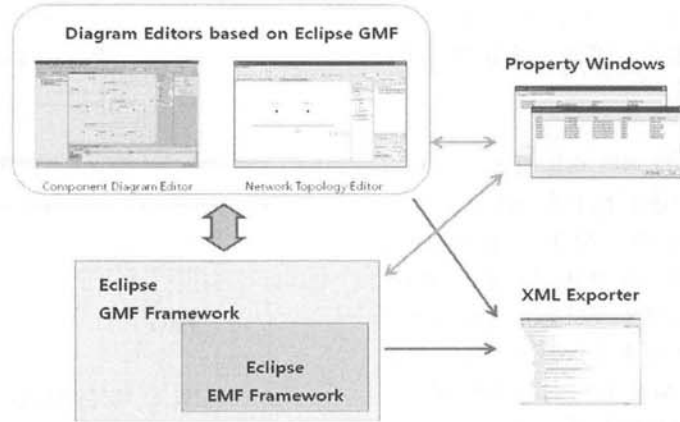
(그림 4) AUTOSAR 네트워크 토폴로지의 예

3.3 이클립스 GMF 기반의 컴포넌트 모델링 도구의 설계

차량용 소프트웨어 컴포넌트 모델링 도구는 크게 컴포넌트 다이어그램 편집기, 네트워크 토폴로지 편집기, 모델정보의 입력하는 속성 편집기로 나뉘어볼 수 있다. 컴포넌트 다이어그램 편집기와 네트워크 토폴로지 편집기는 앞서 정의한 메타모델을 이용하여 이클립스 GMF 프레임워크[13]를 이용하여 개략적인 다이어그램 편집기를 생성한다. 하지만 다양한 컴포넌트 및 네트워크의 속성들을 편집하는 속성 편집기는 속성 창으로 구현되므로 자바 언어로 구현된다. 컴포넌트와 네트워크의 속성들은 이클립스 GMF 프레임워크에서 관리되므로 이러한 자료들을 접근하기 위해서는 연계 모듈을 개발하여야 한다. 그리고 모델링 결과를 외부 도구와 효율적으로 연동하기 위해 XML Exporter 기능을 가진다. (그림 6)은 차량용 소프트웨어 컴포넌트모델링 도구의 구조를 보여준다.



(그림 5) 네트워크 토폴로지 다이어그램 편집기의 메타모델 클래스 다이어그램



(그림 6) 차량용 소프트웨어의 컴포넌트 모델링 도구의 구조

#### 4. 컴포넌트 모델링 도구의 구현

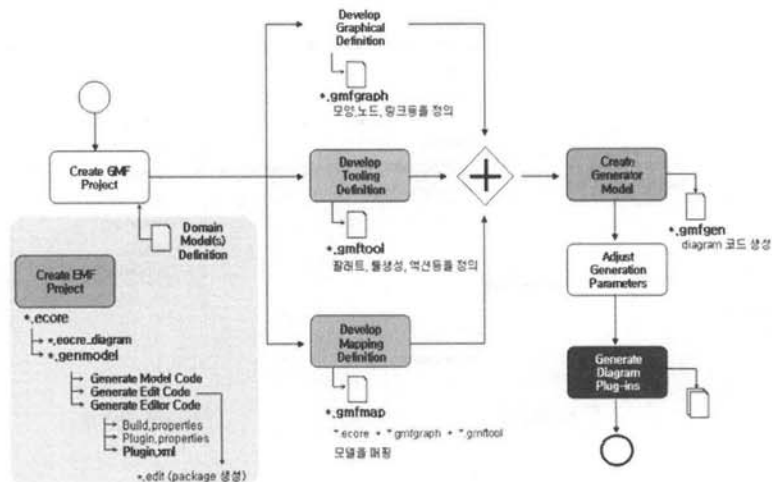
##### 4.1 이클립스 GMF 기반의 다이어그램 편집기 생성

이클립스는 자바 기반의 확장 가능한 개발 플랫폼이며 오픈 소스이다. 그 자체로는 프레임워크이며 플러그인 컴포넌트에서 개발 환경을 구현하는 서비스 세트이다. 이클립스에서는 다양한 플러그인들을 지원하고 있으며, 사용자가 원하는 플러그인들을 조합하여 사용할 수 있다. 이클립스의 GMF (Graphical Modeling Framework)[13]는 EMF(Eclipse Modeling Framework)[15]와 GEF(Graphical Editing Framework)[16]를 기반으로 한다. EMF는 모델 정보를 표현하기 위한 프레임워크이고 GEF는 다이어그램 편집기 제작을 위한 프레임워크이다. (그림 7)은 GMF를 이용한 다이어그램 편집기 개발 방법의 진행 과정을 보여준다.

GMF를 이용하여 모델링 도구를 구현하기 위해서는 먼저 EMF를 이용하여 구현하고자 하는 도메인의 메타모델을 정의해야 한다. 앞 절에서 정의한 컴포넌트 다이어그램 편집기와 네트워크 토폴로지 편집기의 메타모델을 Ecore 편집기를 이용하여 메타모델을 입력한 후 EMF 모델인 genmodel 파

일을 생성한다. Genmodel 파일에서 Generate Model Code, Generate Edit Code, Generate Editor Code 메뉴를 이용해 EMF 모델 플러그인을 자동 생성한다.

EMF 모델 플러그인이 생성되면 도메인 메타모델과 EMF 모델을 이용해 GMF 정의 작업을 한다. GMF 정의 과정은 그래픽 정의(Graphical Definition), 도구화 정의(Tooling Definition), 매핑 정의(Mapping Definition)의 세 단계로 나누어진다. 그래픽 정의는 다이어그램 편집기에 그려질 노드와 링크에 대한 정의이다. 여기서 정의된 노드와 링크는 매핑 단계에서 도메인 모델과 매핑된다. 노드와 링크의 모양에 대해서는 별도로 그림 갤러리에서 정의된다. 도구화 정의는 다이어그램 편집기에 노드와 링크를 그리기 위한 팔레트를 정의하는 부분이다. 사용자는 버튼, 메뉴, 툴바 등의 도구로 구성되는 팔레트를 이용하여 다이어그램 편집기에 노드와 링크를 추가할 수 있다. 매핑 정의는 도메인 모델, 그래픽 정의, 도구화 정의 간의 관계를 설정한다. 그래픽 정의에서 정의된 노드와 링크, 그리고 도구화 정의에서 정의된 도구, 도메인 모델의 실제 구성요소를 실제로 다이어그램 편집기에서 표현될 노드와 링크를 기준으로 매핑한다.



(그림 7) 이클립스 GMF 플러그인 개발 과정

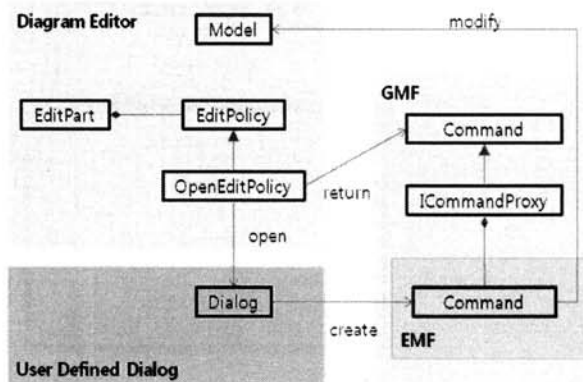
앞의 세 단계가 완료되면 매핑 정의에서 "Create generator model" 메뉴를 이용하여 GMF 코드 생성 모델인 gmfgen 파일을 생성한다. gmfgen 파일에는 앞서 정의한 도메인 모델부터 EMF 모델, GMF의 세가지 정의 등 모든 내용이 포함된다. 또한 생성될 플러그 인에 대한 정보가 포함되어 있다. gmfgen 파일에서 다이어그램 코드 생성 메뉴를 이용하여 컴포넌트 다이어그램 편집기와 네트워크 토폴로지 편집기를 생성한다.

#### 4.2 GMF 다이어그램 편집기와 연동 모듈 개발

이클립스 GMF를 이용하여 생성된 설계 도구는 실제로 다이어그램을 그리는 역할을 수행한다. 그 외에 속성을 설정하는 부분은 이클립스에서 제공하는 속성 창으로 제공되지만 일반 사용자가 사용하기에는 쉽지 않다. 그렇기 때문에 이클립스 GMF를 통해 생성된 소스 코드를 수정 및 확장하여 사용자가 쉽게 속성을 설정할 수 있는 다이얼로그 창을 만들어준다. 생성된 다이어그램 편집기에 속성 창을 추가하기 위해서는 먼저 생성된 소스 코드에 대한 분석이 이루어져야 한다.

이클립스 GMF는 기본적으로 MVC(Model, View, Controller) 모델을 따른다. 사용자가 뷰(view)를 통해 입력한 내용을 제어기(controller)가 받아 모델(model)을 변경한다. (그림 8)은 다이어그램 편집기에서 사용자 입력을 받아 모델이 변경되는 과정에서 나타나는 클래스들의 관계를 보여준다. 먼저 다이어그램 노드를 나타내는 EditPart에 사용자 행동에 반응할 EditPolicy를 설정한다. 그 후에 사용자 입력을 받게 되면 해당 EditPolicy가 실행된다. EditPolicy에서는 다이얼로그를 생성하고, 다이얼로그에서는 사용자 입력에 따라 모델을 변경할 EMF 명령어들을 생성한다. 이 EMF 명령어들을 EditPolicy가 반환하는 GMF 명령어에 담게 된다. 반환된 GMF 명령어는 나중에 모델을 변경할 수 있는 쓰레드에서 실행된다. 이때 GMF 명령어 안에 포함된 EMF 명령어가 실행되어 모델이 변경된다.

GMF로 생성된 편집기는 사용자 입력에 대한 반응을 처리하기 위해서 전략 패턴(strategy pattern)을 이용한다. 이렇게 함으로써 같은 사용자 입력에 대하여 다이어그램 노드 별로 각기 다른 행동을 정의할 수 있다. 이 패턴은 GMF로



(그림 8) EMF 모델정보 변경에 연관된 클래스간의 관계

생성된 코드에서 policy로 불리며 그 행동에 따라 여러 가지 종류가 있는데, 그 중에서 프로젝트에 사용된 추상 클래스 OpenEditPolicy는 다이어그램 노드를 더블클릭 했을 때의 행동을 정의한다. 이 클래스를 상속받아 구현하고, 다이어그램 노드에 그 인스턴스를 등록하면 그 노드를 더블 클릭할 때마다 이 구현 클래스의 인스턴스가 작동한다. 아래의 코드는 OpenEditPolicy를 구현한 OpenEditorEditPolicy 클래스의 인스턴스를 등록하는 코드의 일부를 보여준다.

```

public class ComponentEditPart extends ShapeNodeEditPart {
    .....
    protected void createDefaultEditPolicies() {
        .....
        OpenEditorEditPolicy policy = new OpenEditorEditPolicy();
        this.installEditPolicy(EditPolicyRoles.OPEN_ROLE, policy);
    }
}

```

이렇게 등록된 정책(policy)은 그 종류에 따라서 추상 클래스가 정의하는 핸들러의 이름이 조금씩 다르다. OpenEditPolicy의 경우에는 getOpenCommand() 메소드를 핸들러로 구현하도록 정의하고 있다. policy의 핸들러 메소드들은 반환형으로 GMF가 정의하는 명령어를 반환한다. 여기에서 사용된 패턴은 명령어 패턴(command pattern)이다. 그러므로 이 메소드의 반환값에는 메소드의 실행 결과로 모델에 적용되어야 할 사항들이 담겨 있다. 이 프로젝트에서는 노드의 속성을 정의하기 위해서 다이얼로그를 열고, 수정되어야 할 사항들을 EMF가 정의하는 명령어 형태로 만들어 나중에 GMF 명령어에서 실행되도록 매크로 명령어 형태로 구현했다. 이에 해당하는 코드는 아래와 같다.

```

public class OpenEditorEditPolicy extends OpenEditPolicy {
    LinkedList<org.eclipse.emf.common.command.Command> cmdList;
    .....
    protected Command getOpenCommand(Request request) {
        .....
        dialog = new ComponentDialog(..., cmdList);
        dialog.open();
        return new ICommandProxy(
            new AbstractTransactionalCommand(editingDomain, "", null) {
                protected CommandResult doExecuteWithResult(IProgressMonitor
                    monitor, IAdaptable info)throws ExecutionException {
                    for(int i = 0 ; i < cmdList.size() ; i++)
                        cmdList.get(i).execute();
                    return CommandResult.newOKCommandResult();
                }
            }
        );
    }
}

```

GMF로 생성된 다이어그램의 모델을 변경하기 위해서는 EMF의 명령어를 사용해야 한다. 왜냐하면 다이얼로그를 생성하고 실행하는 쓰레드는 모델을 변경하기 위한 쓰기 권한을 가지고 있지 않기 때문이다. 그래서 명령어를 만들어 두었다가 나중에 쓰기 권한을 가진 쓰레드로 명령어의 실행을 위임하는 방법을 사용한다. EMF는 모델을 변경하기 위해서 SetCommand, AddCommand, RemoveCommand 등 다양한

명령어를 정의하고 있다. 이 명령어들은 사용방법이 모두 같다. 먼저 static 메소드인 create() 메소드를 이용해 생성하고, 생성된 인스턴스는 execute() 메소드로 실행할 수 있다. 아래의 코드는 컴포넌트 노드의 이름 속성을 “새 이름”으로 변경하는 EMF 명령어를 생성하는 예이다.

```
Command cmd = SetCommand.create(editingDomain, component,
    DiagramPackage.eINSTANCE.getComponentPrototype_Name(),
    "새 이름");
```

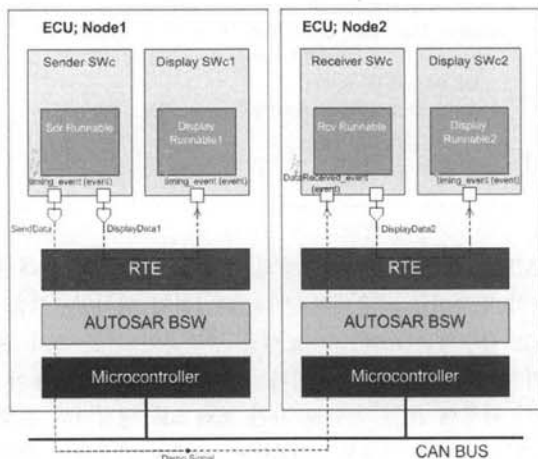
여기에서 editingDomain은 해당 노드를 가리키는 EditPart의 getEditingDomain() 메소드를 이용해 구할 수 있다. 형식인자인component는 변경하려는 대상을 의미한다. 세 번째 형식인자는 GMF가 생성한 모델의 어떤 속성을 변경하려는지를 명시한다. 마지막 형식인자는 모델을 변경할 때 사용될 데이터이다. 이렇게 만들어진 EMF 명령어는 바로 실행되지 않고, GMF 명령어에 담아서 쓰기 권한을 가진 쓰레드에서 실행된다.

### 5. 도구의 적용사례 및 맞춤형

#### 5.1 응용 시스템 적용사례

도구의 기능을 검증하기 위해 (그림 9)와 같은 간단한 응용시스템을 구성하였다. 응용시스템의 구성환경은 두 개의 ECU가 CAN 버스를 통해서 연결되어 있고, ECU 별로 두 개의 소프트웨어 컴포넌트가 정의되어 있다. Node 1의 Sender SWc는 1000ms 주기로 로컬 데이터의 값을 하나 증가하여 이 값을 CAN 버스를 통해서 시그널로 전송하며, Node 2의 Receiver SWc는 CAN 인터럽트 핸들러 및 RTE를 통해서 데이터를 수신하고 로컬 데이터의 값을 갱신한다. Display SWc1,2는 1ms 단위로 도트매트릭스 LED에 현재의 로컬 데이터 값을 표시하는 기능을 수행한다.

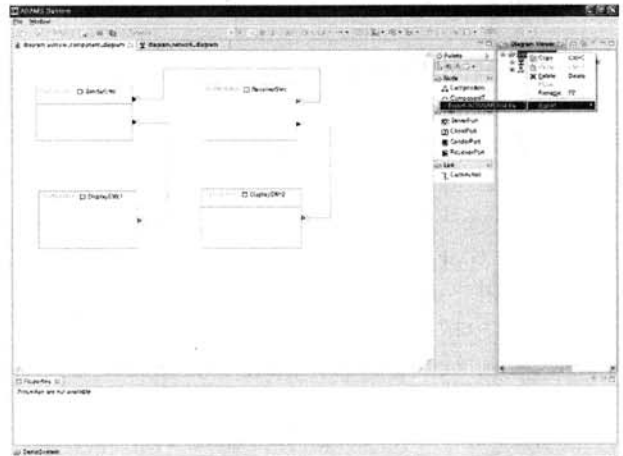
(그림 10)은 소프트웨어 컴포넌트 모델링 도구를 이용하여 응용시스템의 네 개의 컴포넌트인 Sender SWC, Display



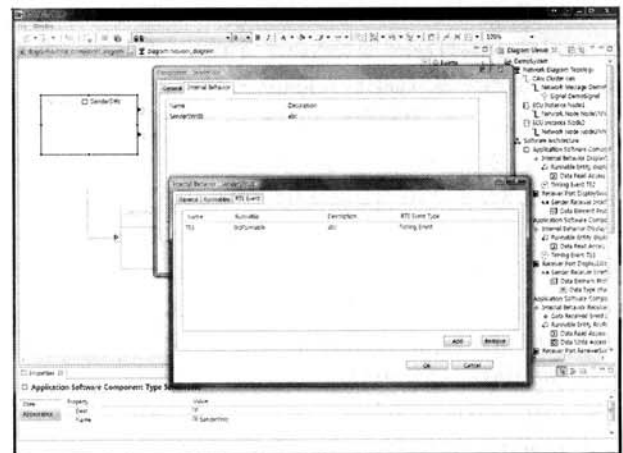
(그림 9) 응용시스템 구성도

SWc1, Receiver SWc, Display SWc2간의 연관관계를 나타내고 있다.

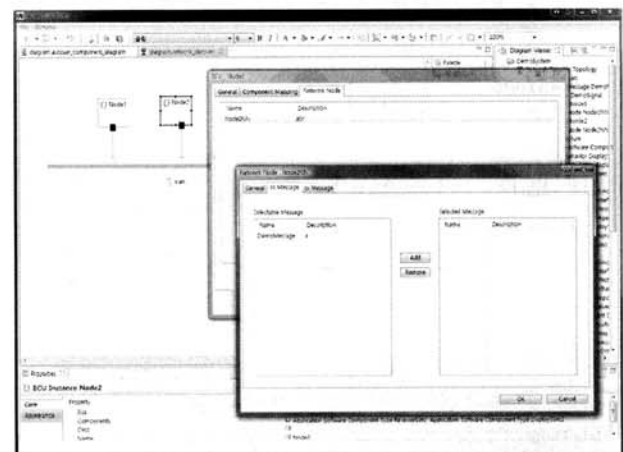
(그림 11)은 컴포넌트 다이어그램의 각 컴포넌트에 대해 RTE Event 및 Internal Behavior 속성을 입력하는 과정을 보여준다. (그림 12)는 CAN 통신으로 연결된 ECU Node1



(그림 10) 계층적 컴포넌트 다이어그램 생성



(그림 11) 컴포넌트 다이어그램에서 컴포넌트의 속성 입력



(그림 12) 네트워크 토폴로지 다이어그램 및 네트워크 속성 입력

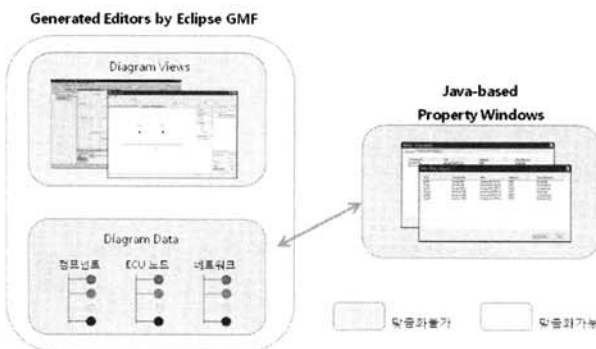


과 ECU Node2의 네트워크 토폴로지를 보여준다. 그리고 속성 편집기를 이용하여 각 네트워크 노드의 메시지를 정의하는 과정을 보여준다.

## 5.2 도구의 GUI 맞춤화

이 논문에서 개발된 AUTOSAR지원 도구는 공개소스 기반으로 이클립스와 자바 언어로 구현되어 있다. (그림 13)은 도구의 개략적인 구조로 이클립스 GMF 프레임워크에 의해 생성된 부분과 자바로 직접 구현된 속성창 부분을 나타내고 있다. 이클립스 GMF를 이용하여 생성된 다이어그램 편집기 부분은 맞춤화 가능하며, 반면 속성창과 다이어그램 데이터 부분은 변경이 불가능하다.

(그림 8)에서 설명한 바와 같이 이클립스 GMF를 이용하여 생성된 다이어그램 편집기 부분을 생성한다. 다이어그램 편집기 생성은 다이어그램의 메타모델만 있으면 항상 동일하게 생성할 수 있으며 다이어그램 편집기의 GUI 부분은 생성 과정에서 사용자의 취향에 따라 맞춤화 가능하다. 예를들어, 편집기의 메뉴, 메뉴 아이콘, 다이어그램 노드 모양, 노드 배치 등을 생성 과정에서 변경할 수 있다.



(그림 13) 개발된 도구의 맞춤화 영역

## 6. 결론 및 향후연구

AUTOSAR는 자동차 소프트웨어 아키텍처에 대한 국제 표준으로 차량용 소프트웨어를 개발하는 방법론을 제시하고 있다. AUTOSAR의 컴포넌트 기반 개발방법에서는 다양한 다이어그램을 작성하여야 하며 다이어그램간의 일관성을 유지하기 위해서는 도구의 지원이 요구된다. 이 연구에서는 이클립스 GMF 기술을 이용하여 AUTOSAR 3.0 표준을 충실히 따르면서 기존 도구들의 단점을 보완하는 차량용 소프트웨어의 컴포넌트 모델링 도구인 그래픽 기반의 컴포넌트 모델링 도구와 네트워크 토폴로지 도구를 개발하였다. 그리고 이러한 모델링 편집기의 속성을 입력할 수 있는 다양한 속성 창을 개발하여 다이어그램 편집기와 연계하였다. 이러한 도구들은 이클립스 GMF로 생성되어 개발 업체에 맞게 손쉽게 맞춤화할 수 있으며 국제 표준 및 개발 환경 발전에 쉽게 발맞출 수 있는 장점이 있다.

AUTOSAR 표준에 명시되어 있는 소프트웨어 컴포넌트와 네트워크 토폴로지를 좀 더 명확히 모델링 하기 위해서는 각 모델의 속성, 즉 데이터 타입, 오퍼레이션, 네트워크 메시지, 시그널 등을 입력할 수 있는 추가적인 모듈이 필요하다. 또한 AUTOSAR 개발 방법론의 다음 단계인 ECU 플랫폼 설정 단계에 대한 지원을 위해 AUTOSAR XML 기술 파일로 모델링된 소프트웨어 정보를 추출하는 기능이 요구된다. 향후 소프트웨어 컴포넌트 모델과 네트워크 토폴로지 모델에 속성을 추가하는 모듈과 AUTOSAR XML 스키마 정의를 참조하여 소프트웨어 모델링 정보를 XML 기술 파일로 추출하는 기능을 구현할 예정이다.

## 참고 문헌

- [1] Hardung Bernd, Kolzow Thorsten, and Kruger Adnreas, "Reuse of Software in Distributed Embedded Automotive Systems," Proceedings of the 4th ACM International Conference on Embedded Software, pp.203-210, 2004.
- [2] P. Hansen. "New S-Class Mercedes: Pioneering Electronics," The Hansen Report on Automotive Electronics, Vol.18, No.8, pp.1-2, Oct., 2005.
- [3] A. Pretschner, M. Broy, I.H. Kruger, T. Stauner, "Software Engineering for Automotive Systems: A Roadmap," Future of Software Engineering, May, 2007.
- [4] AUTOSAR, Automotive Open System Architecture, Release 3.1, <http://www.autosar.org>
- [5] O. Niggemann, J. Stroop, "Models for Model's Sake : Why Explicit System Models are also an End to themselves," 30<sup>th</sup> international Conference on Software Engineering, 2008.
- [6] H. Li, P. Lu, M. Yao, N. Li, "SmartSAR: A Component-based Hierarchy Software Platform for Automotive Electronics," International Conference on Embedded Software and Systems, 2009.
- [7] D. Schreiner, M. Schordan, K. M. Goschka, "Component Based Middleware-Synthesis for AUTOSAR Basic Software," IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, 2009.
- [8] H. C. Jo, S. Piao, S. R. Cho, W. Y. Jung, "RTE Template Structure for AUTOSAR based Embedded Software Platform," IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, 2008.
- [9] dSpace, "SystemDesk for developing complex system architectures," [http://www.dspaceinc.com/ww/en/inc/home/products/sw/system\\_architecture\\_software/systemdesk.cfm](http://www.dspaceinc.com/ww/en/inc/home/products/sw/system_architecture_software/systemdesk.cfm)
- [10] Vector, Network Designer, [http://www.vector.com/vi\\_networkdesigner\\_en.html](http://www.vector.com/vi_networkdesigner_en.html)
- [11] A. Rajnak, A. Kumar, "Computer-aided Architecture Design & Optimized Implementation of Distributed Automotive EE Systems," 44<sup>th</sup> ACM/IEEE Design Automation Conference,

2007.

- [12] 주홍택, 정설영, 이명숙, "AUTOSAR 기반의 FlexRay 모듈 개발," 정보처리학회논문지C, Vol.16, No.6, 2009.
- [13] Eclipse, Eclipse Graphical Modeling Framework, <http://www.eclipse.org/modeling/gmf>
- [14] U. Honekamp, "The Autosar XML Schema and its Relevance for Autosar Tools," IEEE Software, Vol.26, Issue4, July/August, 2009.
- [15] Eclipse, Eclipse Modeling Framework Project, <http://www.eclipse.org/modeling/emf>
- [16] Eclipse, Graphical Editing Framework, <http://www.eclipse.org/gef>



### 박인수

e-mail : ispark82@gmail.com  
 2008년 경북대학교 전자전기컴퓨터학부(학사)  
 2010년 경북대학교 전자전기컴퓨터학부(공학석사)  
 2010년~현 재 삼성전자 무선사업부 사원  
 관심분야: 임베디드 소프트웨어 모델링, 소프트웨어 테스트, 차량용 운영체제



### 이정선

e-mail : rubyindark@naver.com  
 2008년 경북대학교 전자전기컴퓨터학부(학사)  
 2010년 경북대학교 전자전기컴퓨터학부(공학석사)  
 2010년~현 재 LG CNS 사원  
 관심분야: 임베디드 실시간 시스템 모델링 및 분석, 모델 기반 개발 방법론 등



### 조성래

e-mail : srcho@dgist.ac.kr  
 1998년 경일대학교 컴퓨터공학과(학사)  
 2000년 영남대학교 컴퓨터공학과(공학석사)  
 2000년~2003년 (주)아라기술 전임연구원  
 2005년~현 재 대구경북과학기술원 선임연구원

관심분야: 임베디드 실시간 운영체제, AUTOSAR, 임베디드 S/W 검증 기술 등



### 정우영

e-mail : wyjung@dgist.ac.kr  
 1984년 서울대학교 전기공학과(학사)  
 1986년 한국과학기술원 전기및전자공학과(공학석사)  
 1991년 한국과학기술원 전기및전자공학과(공학박사)

1991년~1999년 삼성전자 수석연구원  
 1999년~2003년 ㈜코어세스 연구소장  
 2004년~2004년 송실대학교 객원교수  
 2004년~현 재 대구경북과학기술원 책임연구원  
 관심분야: Wireless sensor network, Mobile Ad-hoc network, Vehicle Area Network, 임베디드 시스템 모델링



### 이우진

e-mail : woojin@knu.ac.kr  
 1992년 경북대학교 컴퓨터학과(학사)  
 1994년 한국과학기술원 전산학과(공학석사)  
 1999년 한국과학기술원 전산학과(공학박사)  
 1999년~2002년 한국전자통신연구원 S/W 공학연구부 선임연구원

2002년~현 재 경북대학교 IT대학 컴퓨터학부 부교수  
 관심분야: 임베디드 시스템 모델링 및 분석, Formal methods, CBD, Requirements Engineering, Petri nets 등