

Plug & Play 지원형 차량용 미들웨어

김대순*, 한영주**, 강동기*, 윤찬현*

*한국과학기술원 전기 및 전자공학과

**한국과학기술원 정보전자연구소

e-mail : {sundae21, y.han, dkkang, chyoun}@kaist.ac.kr

Design of Automotive Middleware for Supporting Plug & Play

Daesun Kim*, Youngjoo Han**, Dongki Kang*, Chan-Hyun Youn*

*Dept. of Electrical Engineering, KAIST

**Information and Electronics Research Institute, KAIST

요 약

AUTOSAR 는 차량용 시스템의 소프트웨어 인프라를 표준화하여 차량용 소프트웨어의 재사용성을 높이고자 개발된 차량용 소프트웨어 구조 표준안이다. 그러나, 기존 AUTOSAR 는 새로운 하드웨어의 추가 및 변경을 지원할 수 없기 때문에 하드웨어의 변경에 대해 유연하게 적용할 수 있는 구조 또는 기법이 필요하다. 따라서, 본 논문에서는 AUTOSAR 의 모델에 Workflow 를 도입하고, 동적인 네트워크를 구축하여 Plug & Play 기능을 지원하는 차량용 소프트웨어 구조를 제안한다.

1. 서론

자동차 산업에 있어서 전기전자부품과 소프트웨어의 비중이 점점 높아지고 있다. 이에 따라 차량용 소프트웨어 시스템 개발 단계에서의 편의를 제공하기 위해, 유럽의 자동차 제조사들과 협력업체들이 중심이 되어 차량용 소프트웨어 구조 표준안인 AUTOSAR 를 개발하였다. 하지만 AUTOSAR 를 통해 개발된 차량용 소프트웨어 시스템은, 차량에 한번 설치되고 나면 모듈 별로 업데이트를 하거나 교체하는 것이 불가능한, 다소 정적인 형태로 구현된다. 기계부품보다 현저히 빠른 전기전자부품과 소프트웨어의 개발 속도, 발전 속도와 더불어, 차량용 소프트웨어 시스템이 재설정 가능한 동적인 구조를 갖게 된다면 차량 구매 이후 모듈 별 업그레이드를 통해 운전자의 만족도를 향상시킬 수 있고, 차량의 수명을 연장시킬 수 있을 것이다. 따라서 본 논문에서는 Workflow 를 기반으로 하여 Plug & Play 기능을 가능하게 하는 차량용 소프트웨어 구조를 제안한다.

2. AUTOSAR

AUTOSAR 의 주된 목표는 차량용 시스템을 위한 소프트웨어 인프라를 표준화하는 것이다. 표준화를 통하여 소프트웨어의 재사용성을 높이면, 자동차 제조사의 입장에서는 개발비용을 절감하는 효과를 얻을 수 있기 때문이다. 이러한 목표를 달성하기 위해 AUTOSAR 표준은 다음과 같은 내용을 포함한다[1].

- 표준화된 API 를 통해 하드웨어와 소프트웨어 계층을 분리
- 소프트웨어 컴포넌트의 캡슐화
- 소프트웨어 컴포넌트의 데이터 형 정의

- 소프트웨어 모듈 식별

AUTOSAR 는 AUTOSAR 소프트웨어 계층, Runtime Environment(RTE), 기본 소프트웨어 계층으로 구성되며, 각각은 다음과 같은 역할을 한다[2][3].

- AUTOSAR 소프트웨어 계층: 특정 ECU 에 mapping 된 소프트웨어 컴포넌트들로 구성
- RTE: 하나의 ECU 안에서, 또는 ECU 와 ECU 사이에서 일어나는 데이터 교환을 관리, ECU 간에 동일한 인터페이스와 서비스를 제공하여 통신 환경 추상화
- 기본 소프트웨어 계층: 컴포넌트가 지정된 작업을 수행하는 데에 필요한 서비스를 제공

2.1. AUTOSAR 의 Plug & Play 지원 제약 사항

AUTOSAR 는 소프트웨어가 설치된 이후에 하드웨어 구성의 변경을 고려하지 않았다. 따라서 하드웨어 구성의 변화를 발견하기 위한 하드웨어나 소프트웨어가 존재하지 않는다. 또한, 새롭게 연결된 장치가 자신의 존재를 알릴 수 있는 방법도 없다. 따라서 운전자가 자신의 차에 새로운 전자부품을 장착하더라도, 자동차가 그 장치를 발견하고 인지할 수 없다. 또한, AUTOSAR 의 개발 방법론 Plug & Play(P&P)를 지원하는 데에 문제가 된다. AUTOSAR 의 개발 방법론을 따르면, 소프트웨어 컴포넌트(SWC; Software Component)와 시스템 제약 사항, ECU 자원에 대한 정보를 수집하는 단계가 전체 시스템의 수명 중 컴파일 단계 직전에 단 한 번 존재하기 때문이다. 따라서 구축된 AUTOSAR 시스템의 RTE 는 더 이상 추가, 변경될 여지가 없이 고정된 형태의 SWC 네트워크를 구성한다. 이러한 구성을 변경하기 위해서는 AUTOSAR 소프트

웨어의 전체 시스템을 다시 컴파일 하여야 한다.

3. Plug & Play 지원형 차량용 미들웨어 구조

3.1. Master/Slave ECU

본 논문에서 제안하는 미들웨어 구조에서 ECU 들은 Master ECU 와 Slave ECU 로 나뉘어진다. 차량에는 반드시 하나의 Master ECU 가 존재하여야 하며, ECU 들의 네트워크에서 중심점과 같은 역할을 맡는다. 나머지 ECU 들은 Slave ECU 로써, 각각 차량 기능의 일부분을 담당한다. 그림 1 은 Master ECU 와 Slave ECU 의 구조를 나타낸다.

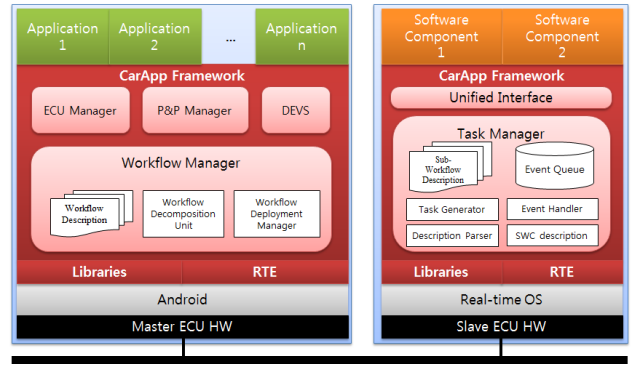
제안하는 미들웨어에서는 안전성의 문제를 고려하여, 차량 시스템의 하드웨어적인 재구성 및 재설정 과정은 운전자가 차량의 시동을 걸기 전에 이루어진다고 가정한다. 따라서 Master ECU 는 차량의 시동이 걸린 직후에, 모든 ECU 와 각각의 ECU 에 설치된 SWC 의 정보(ECU description, SWC description)를 수집하고, 수집된 정보를 기반으로 하드웨어 구성이나 SWC 배치 상태의 변화를 감지하며, 이후의 동작에 변경된 설정을 적용한다. ECU description 은 Master ECU 안에 있는 ECU manager 가 관리하며, SWC 가 재배치되었을 때 참조되고, ECU 들을 모니터링하거나 관리하는 데에 사용된다. SWC description 은, 역시 Master ECU 안에 있는 P&P manager 가 관리한다. P&P manager 는 SWC description 과 Car Application(Car App: SWC 들의 시퀀스로 이루어진 소프트웨어 시스템 단위)의 정보를 고려하여, Car App 이 동작할 수 있는 환경을 조성한다. Slave ECU 들은 각 파트의 실제 동작을 수행할 수 있는 SWC 들을 가진다.

3.2. Workflow Manager 와 Task Manager

Car App 은 구현하려는 시스템의 전체 동작을 정의하며, SWC 들의 선후 관계를 기반으로 한 시퀀스로 구성된다. 따라서 Car App 은 Workflow[4][5]의 형태로 표현이 가능하다. 따라서, 시스템 외부에서 Workflow Designer 를 통해 Car App 의 Workflow 를 작성하면, Workflow Designer 는 Workflow 를 XML 의 형태로 저장한다. 이 XML 파일을 Master ECU 에 있는 Workflow Manager 가 관리하고, ECU 정보들을 대입한 후 ECU 별로 필요한 정보들을 추려낸 Decomposed Workflow 를 생성하여 Slave ECU 들에 있는 Task Manager 로 전송한다. Task Manager 는 전송 받은 Decomposed Workflow 에 따라 같은 ECU 안에 있는 SWC 들을 동작시키고 결과 데이터를 전송한다.

3.3. Runtime Environment

RTE 의 주된 역할은 어플리케이션 소프트웨어들 또는 ECU 들간의 실시간 통신 환경을 제공하는 것이다. 각각의 SWC 는 ECU 의 네트워크 주소와 SWC 정보(ID)를 통해 구분할 수 있으며, 따라서 Master ECU 의 RTE 가 가지는 Master Mapping Table 은 Source 가 되는 SWC 의 ID 와 네트워크 주소를 Destination 이 되는 SWC 의 ID, 네트워크 주소와 엮어서 가지고 있게



(그림 1) Master ECU 와 Slave ECU 의 구조

된다. Mapping & Distribution Manager 는 이러한 Mapping Table 을 관리하는 모듈을 가리킨다. Slave ECU 가 Mapping Table 을 사용하는 경우 Master Mapping Table 을 모두 필요로 하지는 않기 때문에, 해당 ECU 가 가지고 있는 SWC 의 Destination SWC ID 와 네트워크 주소만을 가지는 경량화된 Slave Mapping Table 을 사용한다. Session Manager 는 ECU 들 사이의 연결(Connection 또는 Session)을 관리하는 모듈이다. ECU 들간의 연결을 계속 유지하고, 새로 추가되거나 제거되는 장치들과의 연결을 관리해야 하기 때문에 필요한 모듈이다.

4. 결론

자동차 산업에서 전기전자부품과 소프트웨어의 비중이 증가함에 따라, 유럽의 자동차 제조사들과 협력 업체들은 소프트웨어 재사용 성을 높여 개발비용을 절감하기 위해 AUTOSAR 라고 하는 차량용 소프트웨어 구조 표준안을 개발하였다. 하지만 AUTOSAR 를 통해 개발된 차량용 소프트웨어 시스템은 차량에 한번 설치되고 나면 모듈 별로 업데이트를 하거나 교체하기 위해 전체 시스템을 다시 컴파일 해야 하는 형태를 가지고 있다. 따라서 본 논문에서는 AUTOSAR 의 모델에 Workflow 를 도입하고, 동적인 네트워크를 구축하여 Plug & Play 기능을 지원하는 차량용 소프트웨어 구조와 동작 원리를 제안하였다. 제안한 Plug & Play 지원형 미들웨어는 구현하여 임베디드 시스템 레벨에서 가용성을 검증하고 있으며, 실제 차량용 네트워크 환경에서 안정성 및 가용성 등의 성능 이슈에 대한 고찰은 향후 연구 주제로 남아 있다.

Acknowledgement

이 논문은 2011 년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (2011-0018244).

참고문헌

[1] Georg Pelz, Peter Oehler, Eliane Fourgeau, and Christoph Grimm, "Automotive system design and AUTOSAR", Advances in Design and Specification Languages for SoCs, pp.293-305, 2005 Springer.

- [2] Harald Heinecke, et al., "AUTomotive Open System ARchitecture - an industry-wide initiative to manage the complexity of emerging automotive E/E-architectures," Convergence International Congress & Exposition On Transportation Electronics, 2004.
- [3] 유우석, 박지용, 유종훈, 김세화, 홍성수, "AUTOSAR 에 기반한 차량용 소프트웨어의 구조," 한국자동차공학회 워크숍 및 심포지엄 논문집 (전기전자, ITS 부문 심포지엄), pp. 60-65, Aug 2006.
- [4] Workflow Management Coalitition, <http://www.wfmc.org/>.
- [5] K. Belhajjame, G. Vargas-Solar, C. Collet, "A flexible workflow model for precess-oriented applications," Proceedings of the Second International Conference on Web Information Systems Engineering, 72-80, Dec. 2001.