

차량 환경에서의 HMI 에 관한 연구

최연준, 김선중, 권오천
한국전자통신연구원
e-mail : {june.kimsj, ockwon}@etri.re.kr

A Study of HMI on In-Vehicle Telematics System

YeonJun Choi', SunJung Kim
Telematics S/W Platform Team, Telematics Research Division
Electronics and Telecommunication Research Institute

요 약

차량 내에 도입되는 텔레매틱스 시스템을 설계할 때에는 차량의 특성이 반영되어야 한다. 차량 내의 사용자 인터페이스, 즉 HMI(Human-Machine Interface)는 외관에서 비롯되는 상업성 뿐 아니라 안전과 차량 기능에 있어서 더욱 중요하다. 본 논문에서는 차량 탑승자 중 특히 운전자를 위한 차량 환경을 고려하여 HMI 시스템을 설계하기 위한 방안을 제시한다. HMI 아키텍처는 OSG-i 프레임워크와 같은 국제 표준을 따른다.

1. 서론

텔레매틱스 기술은 모바일 환경과 차량 환경이 결합되어 더욱 지능적인 서비스를 제공할 수 있는 차세대 산업의 유망 주자이다. 운전자를 위한 차량 내 텔레매틱스 시스템을 설계하는 데에 있어서 안전과 차량 고유의 기능을 고려해야 한다는 점에서 일반적인 시스템 개발과 차이가 있다.

동승자(운전자를 제외한 탑승자)는 차량 내에서 영화, 웹사이트 서핑, 게임과 같이 시각을 사로잡는 서비스를 받을 수 있다. 또한 복잡한 다단계 메뉴를 보면서 원하는 서비스를 선택할 수 있다. 그러나 운전자에게 있어 운전 중 시각을 사로잡는 동작을 수행하는 것은 사고를 불러 올 위험이 있다. 따라서, 운전자는 동승자와는 다른 방식으로 메뉴에 접근하거나 정보를 획득할 수 있어야 한다.

차량 내 텔레매틱스 시스템을 위한 HMI 는 현재에도 지속적인 연구가 이루어지는 분야이다.

AMI-C¹는 OSG-i² 프레임워크에 기반한 HMI 에 특징적인 구조를 제안하였다[1]. 차량의 각 장치에 대한 연계 및 가변 UI 를 고려한 확장성 있는 구조인 반면, 성능에 제한이 있는 임베디드 시스템에 사용하기에는 복잡하다는 단점이 있다.

W3C 는 HMI 를 지원함에 있어 멀티모달 프레임워크를 제안한다. 멀티모달 프레임워크는 유연하고 풍부한 표현력을 제공하지만 임베디드 시스템에서 사용하기에는 무겁다[2, 3]. 멀티모달 프레임워크는 모바일에서 홈네트워크를 포괄하는 풀 스펙을 지원한다.

본 논문에서는, 임베디드 텔레매틱스 단말기를 위한 가벼운 형태의 HMI 구조를 제안한다.

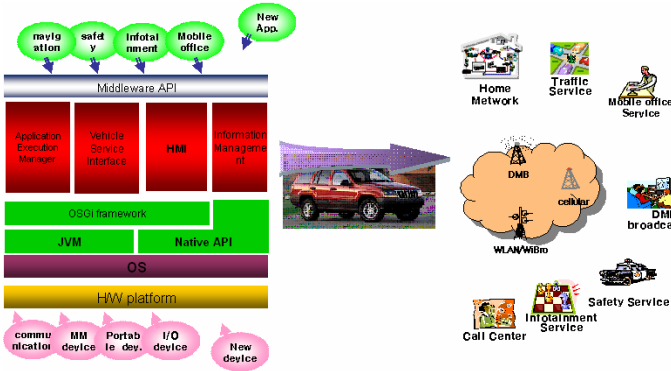
2. 분석

(1) 텔레매틱스 서비스 구조

텔레매틱스 기술은 DMB, CDMA, WLAN, DSRC 등의 다양한 통신 기술을 접목하여 차량 내에서 다양한 서비스를 제공한다. 또한 텔레매틱스 시스템은 이미 존재하는 서비스를 업그레이드하거나 외부의 서비스 콘텐츠와 연동하여 안전 서비스를 효율적으로 확장할 것이 요구된다. (그림 1)은 일반적인 텔레매틱스 구조와 차량 텔레매틱스 시스템의 도식이다.

¹ AMI-C : Automotive Multimedia Interface Collaboration

² OSG-i : Open Service Gateway Initiative



(그림 1) 텔레매틱스 서비스 구조

텔레매틱스 단말기에서 다양한 어플리케이션 서비스를 제공하기 위하여 텔레매틱스 미들웨어는 다음과 같은 서비스를 제공해야 한다[4].

- 통신 서비스(Communication Service) : CAN, MOST, CDMA, DSRC, Bluetooth 등과 같은 다양한 통신 방식에 대한 일관된 접근성을 제공한다.
- 위치 서비스(Positioning Service) : 일관된 방식으로 획득된 현재 위치 데이터를 제공한다.
- 영속 저장 서비스(Persistent Storage Service) : 어플리케이션과 서비스에서 사용하기 위한 저장소 정보에 대한 일관된 접근성을 제공한다.
- 차량 정보 관리 서비스(Vehicle Information Management Service) : 차량의 정적/동적 정보를 제공한다.
- TSP 연동 서비스 : TSP(Telematics Service Provider)와 단말 간 통신을 통한 서비스 프로토콜을 제공한다.
- 차량 어플리케이션 관리 서비스(Telematics Application Management Service) : 차량 환경에 적합한 어플리케이션 수행 특성과 우선순위를 관리한다.
- 차량 서비스 인터페이스(Vehicle Service Interface) : AMI-C에 기반하여 차량 종류에 무관하도록 각 장치 정보를 접근성을 제공한다.
- 개인 정보 관리 서비스(Personal Information Management Service) : 차량과 개인의 정보를 연동한 서비스이다.
- HMI 서비스 : 차량의 각종 I/O 장치에 대한 일관된 접근성을 제공한다.

(2) 요구사항 분석

차량 내 HMI 시스템은 안전과 HMI 자원에 대한 사용성을 위하여 다음과 같은 요구 조건을 가져야 한다.

1) 일관된 인터페이스

차내 HMI 시스템은 사용자에게 I/O 서비스를 접근하기 위한 일관된 인터페이스를 제공한다.

이는 차량과 단말의 I/O 디바이스에 대한 접근을 단순화하며 디바이스를 추가하거나 제거하는 데에 있어서 용이성을 제공한다.

2) 관리 기능

사용자와 어플리케이션이 사용하는 HMI 자원은 HMI 관리기를 통하여 접근해야 한다.

3) 우선 순위

차량 내 HMI 기능을 사용하는 어플리케이션은 우선 순위를 가지고 접근해야 한다.

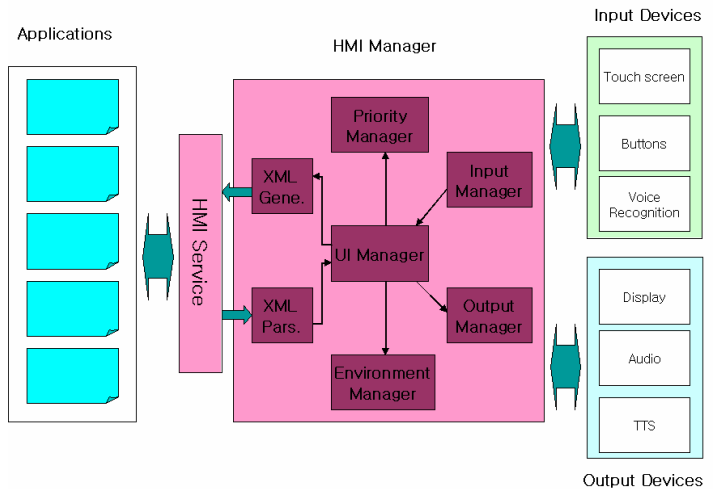
4) 등록 서비스

HMI 자원을 등록하거나 등록 제거하는 기능을 가진다.

3. 설계

(1) HMI 아키텍처

차량 내 HMI 관리기는 2 장에서 언급한 서비스와 요구 사항을 만족하기 위하여 (그림 2)와 같은 7 개의 서브시스템으로 구성된 아키텍처를 가진다.



(그림 2) HMI 관리기 구조

가정은 다음과 같다.

- 입력 장치
 - 터치 스크린, 외부 버튼과 음성 인식
- 출력 장치
 - 화면 디스플레이³, 오디오, 음성 합성⁴

³ 임베디드 시스템의 특성 상 계산량이 많은 화면 디스플레이에 대한 중앙 집중식 제어는 하지 않고 각 어플리케이션이 직접 담당함

⁴ 음성합성은 TTS(Text-to-speech)로 약칭하는 경우가 많다.

HMI 관리기 각 서브시스템의 기능은 다음과 같다.

- HMI Manager : HMI 서비스를 제공혹은 사용하기 위하여 HMI 자원 및 어플리케이션과 인터페이스한다.
- XML parser : XML 형태의 HMI 메시지를 파싱하는 역할을 한다.
- XML generator : HMI 데이터를 XML 형태의 HMI 메시지로 변환시킨다.
- UI manager : 입력 및 출력 기능을 관리한다.
- Priority manager : HMI 자원을 사용하는 어플리케이션의 우선 순위 비교 테이블을 관리한다.
- Environment manager : 입력 및 출력 디바이스 정보를 관리하며 등록 및 등록 제거를 관리한다.
- Input manager : 입력 디바이스를 관리하며 입력 데이터를 수집한다.
- Output manager : 출력 디바이스를 관리하며 어플리케이션이 출력하고자 하는 메시지를 출력시킨다.

(2) 알고리즘

HMI 기능의 흐름에는 크게 세 가지 알고리즘이 필요하다.

첫 번째, 입력 흐름은 입력이 발생하면 시작되어 어플리케이션 실행으로 종료된다.

두 번째, 출력 흐름은 어플리케이션이 HMI 자원에 대한 점유를 요청하면서 시작되어 HMI 자원의 사용으로 종료된다.

세 번째, 등록 및 관리 흐름은 HMI 자원의 등록 및 관리, 그리고 어플리케이션 우선 순위 관리에 대한 기능 흐름이다.

1) 입력 흐름

입력이 들어오면 입력 알고리즘은 다음과 같이 수행된다.

- ① 적절한 입력 디바이스가 입력 시그널을 감지한다.
- ② 입력 디바이스는 입력된 데이터를 Input manager 로 보낸다.
- ③ Input manager 는 데이터를 분석하고 이것을 HMI 메시지로 변환한다. 멀티모달 프레임워크와 달리 하나로 융합되는 여러 개의 입력 (multiple input)은 지원하지 않는다.
- ④ UI manager 는 HMI 메시지를 타겟인 목적지 (destination) 어플리케이션으로 보낸다. 이 경우, 목적지는 항상 어플리케이션 실행이다. 그러므로 어플리케이션 관리기는 해당 어플리케이션을 실행시키게 된다.

2) 출력 흐름

출력 흐름은 입력 흐름 과정보다 조금 더 복잡해진다.

- ① 어플리케이션이 HMI 출력을 원하면 XML generator 를 이용하여 HMI 출력 메시지를 작성한다.
- ② 어플리케이션이 HMI 관리기에 HMI 출력 자원에 대한 요청 메시지를 보낸다.
- ③ XML parser 가 요청 메시지를 분석한다.
- ④ UI manager 가 요청된 HMI 자원이 사용 가능한지 아닌지 검사한다.
- ⑤ 자원이 사용 가능하면, UI manager 는 어플리케이션에 요청된 HMI 자원을 할당한다.
- ⑥ 자원이 사용 중인 경우 Priority manager 는 자원을 사용 중인 어플리케이션과 자원을 요청한 어플리케이션의 우선 순위를 검사한다.
- ⑦ 자원을 사용 중인 어플리케이션의 우선 순위가 높은 경우 자원을 요청한 어플리케이션은 사용 중인 어플리케이션이 HMI 자원 사용을 마칠 때까지 기다린다.
- ⑧ 자원을 요청한 어플리케이션이 자원을 사용 중인 어플리케이션보다 우선 순위가 높은 경우 기존에 사용 중인 어플리케이션으로부터 자원을 해제하여 새로 요청한 어플리케이션에 할당한다.
- ⑨ 해당 출력 디바이스로 데이터 출력을 요청한다.

3) 등록 및 관리 흐름

HMI 자원 관리는 다음과 같은 흐름을 가진다.

- ① HMI 자원은 등록을 원할 경우 HMI manager 에 등록 요청 메시지를 보낸다.
- ② HMI manager 는 유일한 이름을 가지는 특정 HMI 자원으로 등록한다.
- ③ HMI 자원이 등록을 제거하고자 하는 경우 HMI manager 에 등록 제거 메시지를 보낼 수 있다.
- ④ HMI manager 는 필요한 경우 등록된 HMI 자원에 대한 등록을 제거할 수 있다.

어플리케이션 우선 순위 관리는 다음과 같은 흐름을 가진다.

- ① HMI manager 는 Run Level⁵에 기반한 어플리케이션 우선 순위 테이블을 관리한다.
- ② HMI 자원 요청이 들어오는 어플리케이션의 우선 순위와 현재 자원을 사용 중인 어플리케이션의 우선 순위는 우선 순위 테이블에 있는 어플리케이션 우선 순위를 기반으로 하여 비교된다.
- ③ 만약 두 어플리케이션의 우선 순위가 동일한 경우 자원을 사용 중인 어플리케이션의 우선 순위가 자원을 요청한 어플리케이션의 우선 순위보다 더 높은 것으로 간주된다.

⁵ AMI-C 의 어플리케이션 관리는 차량 상태에 따른 Run Level 을 이용한 어플리케이션 우선 순위 관리 기능을 제공한다.

4. 결론

본 논문에서 제안하는 시스템은 OSG-I 프레임워크에 기반하고, AMI-C 스펙에서 제시하는 어플리케이션 관리와 차량 서비스를 이용하도록 설계되었다.

현재 HMI 시스템에 대한 설계 단계를 끝내고 임베디드 보드 상에서 개발하는 단계에 와 있다.

향후 차량 내 텔레매틱스 시스템을 위한 HMI 관리기의 구현 및 적용이 필요하다.

참고문헌

- [1] AMI-C Requirements and Specification for HMI, AMI-C, <http://www.ami-c.org>, 2002
- [2] A Study on e-Learning in Vehicle Environment, Y.J. Choi, et al, WSEAS, 2004
- [3] EMMA, W3C, <http://www.w3c.org/emma>
- [4] Analysis and Design of Middleware Architecture for In-vehicle Telematics Applications, M.J. Kim, et al, ITS Congress, 2005
- [5] A Study of HMI on In-Vehicle Telematics System, Y.J. Choi, et al, AIC, 2005
- [6] An HMI Message Architecture of In-Vehicle Telematics System, Y.J. Choi, et al, WSEAS Transactions on Information Science and Applications, 2005
- [7] 텔레매틱스 소프트웨어 미들웨어 설계 방안, 최연준 외, 한국정보처리학회, 2004