

# 메시지 기반의 HMI 아키텍처 적용에 관한 연구

최연준\*, 진성일\*\*

\*한국전자통신연구원

\*\*충남대학교 컴퓨터공학과

e-mail : [june@etri.re.kr](mailto:june@etri.re.kr)

## A Case Study : Message-Driven HMI Architecture

YeonJun Choi\*, SeongIl Jin\*\*

\*ETRI

\*\*Dept. of Computer Engineering, Choong-Nam University

### 요 약

차량용 텔레매틱스 시스템에서의 HMI(Human-Machine Interface)는 차량 환경, 단말기, 운영 체제 등에 맞게 특화되어야 한다. 본 논문에서는 특히 운전자와 미들웨어 특성에 특화된 차량 환경을 고려한 HMI 소프트웨어를 설계하고 이에 대한 적용 사례를 살펴 보고자 한다. 본 논문에서 제시된 HMI 소프트웨어는 국제적으로 임베디드 환경에서 널리 사용되는 운영 체제에 중립적이며 Java 기술을 적용하는 OSGi 프레임워크에 기반한다.

### 1. 서론

‘텔레매틱스’라는 용어는 통신(telecommunication)과 정보 공학(informatics)의 합성어에서 유래되었다. 즉, 통신을 이용한 정보에 주요한 관심을 두고 있으며 본래 차량만을 목적으로 한 것은 아니었으나, 국내에서는 주로 차량 내 단말기를 통하여 정보를 공유하기 위한 기술로 널리 인식되고 있다. 이는 국제적으로 크게 다르지는 않으며 텔레매틱스는 ITS (Intelligent Transportation System)의 일부로 중요시되고 있다. 텔레매틱스 기술은 차량 환경을 모바일 등의 통신 기술을 이용하여, 가정, 사무실 등과 연결하기 위한 유비쿼터스 컴퓨팅 기술로 점차 확대되어 나가고 있다. 차량 승객의 경우 자유로이 정보나 엔터테인먼트 서비스를 이용할 수 있다. 그러나 차량 환경의 특수성으로 인하여 특히 운전자를 위한 차량 내 텔레매틱스 시스템은 안전과 차량 기능을 고려할 때 여타의 시스템과는 다른 설계 특성을 가진다. 영화나 움직이는 동영상, 복잡한 수동 제어 장치는 자칫 차량의 운전자로 하여금 운전이 소홀하게 하는 결과를 낳게 되므로 운전자의 감각, 특히 시각을 통한 정보의 전달을 제한하며 손을 자유로이 쓸 수 없는 특성을 반영하여 설계해야 한다. 또한 운전을 정지하고 있는 동안 동일 기능이라도 서로 다른 인터페이스를 가질 수 있다. 차량 내 텔레매틱스 시스템을 위한 HMI 기술은 현재 한창 연구되고 있는 영역이다. 본 논문에서는, 임베디드 텔레매틱스 단말을 위한 HMI 설계를 제시하고 이에 대한 사례를 보여 주고자 한다.

### 2. HMI 요구사항

차량 내 텔레매틱스 시스템을 위한 HMI 요구 사항은

다음과 같다.

- 일관된 인터페이스 : 표준화된 인터페이스를 통하여 서비스나 기능에 관계 없이 일정한 인터페이스를 가져야 한다
- 관리성: 단말기가 제공하는 HMI 자원은 HMI 시스템에 의하여 통제되고 관리되며 제공되어야 한다
- 우선 순위 관리 : 안전에 기반하여 차량의 환경을 고려한 자원 분배의 우선 순위가 관리되어야 한다
- 서비스 등록 관리 : HMI 서비스와 이를 사용하는 어플리케이션은 등록되고 관리되어야 한다
- 선호도 관리 : 운전자의 다양한 HMI 자원에 대한 선호도가 관리되어야 한다
- 견고성 : 시스템은 견고한 특징을 가지고 HMI 자원과 어플리케이션의 돌발적인 사고에도 타 서비스에 영향을 미치지 않도록 설계되어야 한다.

### 3. HMI 아키텍처

본 논문에서 제안하는 시스템은 다음과 같은 OSGi 서비스를 기반으로 한다[1]

- Communication Service
- Positioning Service
- Persistent Storage Management Service

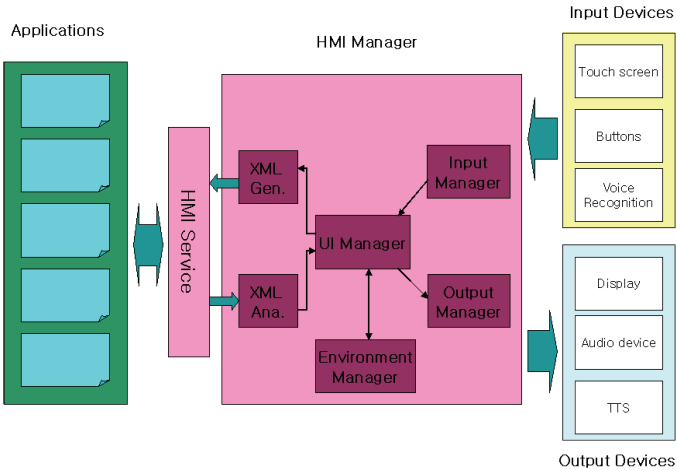
차량 환경 지원을 위하여 확장 AMI-C 서비스를 추가하였다[2]

- Application Management Service
- Vehicle Service Interface
- HMI Service

효율적인 차량과 연동된 정보 유지를 위하여 다음 서비스를 기반으로 한다.

- Vehicle Information Service
- Personal Information Service
- TSP Connection Service

요구 사항을 지키기 위하여 본 논문에서 제안하는 시스템은 (그림 1)과 같은 구조를 가진다.



(그림 1) HMI 아키텍처

- HMI service

모든 UI 는 HMI 서비스로 접근된다. UI 는 H/W device 뿐 아니라 value-added 된 S/W 유틸리티도 포함할 수 있다. HMI service 로 정의된 H/W 디바이스 및 S/W 컴포넌트는 HMI 자원으로 활용된다.

- XML Generator

모든 HMI 자원은 HMI 메시지를 이용하여 사용된다. 따라서 HMI 메시지 형태를 만들기 위하여 XML 메시지를 제공한다.

- XML Analyzer

XML 형태로 된 메시지를 분석하기 위한 구성 요소이다.

- UI Manager

입력 디바이스 및 출력 디바이스는 UI Manager 에 의하여 등록 및 사용 관리된다.

- Environment Manager

사용자의 선호도를 반영하고, 출력 디바이스의 특성을 설정 관리한다.

- Input Manager

입력 H/W 디바이스를 관리하고 실제 입력 데이터를 HMI 메시지로 바꾼다.

- Output Manager

출력 H/W 디바이스를 관리하고 HMI 메시지를 실제 출력 데이터로 바꾼다.

- HMI Manager

입출력 관리를 설정된 서비스 및 어플리케이션에 분배한다.

사용자 인터페이스를 제공하는 H/W 에 의존적이지 않도록 하기 위하여 (그림 2)와 같은 형태의 XML 인터페이스를 제공한다.

```
<!ELEMENT Header EMPTY>
<!ATTLIST Header
    Type (input|output) "output"
>
<!ELEMENT Body (Content)*>
<!ATTLIST Body
    Source CDATA #REQUIRED
    Dest CDATA #REQUIRED
    Action CDATA #IMPLIED
>
<!ELEMENT Content ANY>
```

(그림 2) HMI 메시지를 위한 DTD

4. 시스템 구현

본 논문에서 제안하는 시스템은 PXA 270 의 64M RAM 을 가진 H/W 에 서 구현되었으며 사용된 OS 는 Embedded Linux 이고, knopflerfish 상에서 시험하였다. 또한 본 시스템은 NF Sonata 에 연동하여 시험하였으며 (그림 3)과 같은 UI 를 가졌다.



(그림 3) 프로토타입 시스템의 UI

5. 결론

본 시스템에서 제안하는 HMI 소프트웨어 구조는 이벤트 및 메시지 구동 방식이며, H/W 디바이스 및 HMI 어플리케이션은 서로 의존적이지 않은 관계로 구현될 수 있었다.

참고문헌

[1] OSGi Services ,<http://www.osgi.org>  
 [2] AMI-C Requirements and Specification for HMI, AMI-C, <http://www.ami-c.org>, 2002  
 [3] Analysis and Design of Middleware Architecture for In-vehicle Telematics Applications, M.J. Kim, et al, ITS World Congress, 2005

```
<!ELEMENT HMIMessage (Header, Body)>
```