

AMI-C 기반의 텔레매틱스 단말 미들웨어

및 서비스 배포 메커니즘 설계 및 구현

박용규^{0*}, 김민영^{**}, 이종후^{*}, 곽남영^{*}
(주)윈아이티 기술연구소^{*}, 충남대학교^{**}

smile⁰@winitinc.com, jellygom@home.cnu.ac.kr, jlee@winitinc.com, nykwak@winitinc.com

A Design and Implementation of Distribution Mechanism of Telematics Middleware and Service based on AMI-C

YongKyu Park^{0*}, MinYoung Kim^{**}, JongHu Lee^{*}, NamYoung Kwak^{*}
WinIT Coporate^{*}, ChungNam Univ.^{**}

요 약

텔레매틱스 서비스는 크게 주변 지역 관심 정보, 날씨, 근접 정비소, 현금 자동인출기 등의 정보서비스 분야와 교통정보 서비스, 네비게이션 서비스, 응급서비스, 고장 서비스 및 차량수리 및 차량문제에 관한 진단서비스를 중심으로 서비스가 개발되고 있다. 혹은 두 가지 분야가 복합적으로 개발되고 있다. 또한 텔레매틱스 시장은 여러 가지 기술들이 독립적으로 분할되어 다양한 제품으로 정보를 제공하기 보다는 여러 정보 및 기능 등이 텔레매틱스 시스템으로 집중 및 통합되는 형태가 요구된다. 이에 본 논문에서는 차량 장치 정보를 열람하고 제어하기 위한 단말 미들웨어 모델을 제시/구현하고, 단말 미들웨어를 이용하는 텔레매틱스 서비스 배포를 위한 배포 모델을 제시하고자 한다.

1. 서 론

텔레매틱스(Telematics)란 원격통신(Telecommunication)과 정보과학(Informatics)과 결합된 것으로 차량, 항공, 선박 등 운송장비에 내장된 컴퓨터와 무선통신기술, 위성항법장치, 인터넷에서 문자신호와 음성신호를 바꾸는 기술 등에 의해 정보를 주고받을 수 있는 무선데이터 서비스를 말한다. 이 중에서 자동차 텔레매틱스는 가장 각광받는 분야라고 할 수 있다[1].

텔레매틱스 기술은 기본적으로 차량과 센터를 연결해주는 이동통신 기술, 차량의 진단 및 제어를 위한 자동차 제어 기술, 차량의 위치추적 및 실시간 교통정보 제공을 위한 GPS 및 네비게이션 기술, 차량 정보를 수신하고 연결할 수 있는 단말기 기술, 차량안전을 위한 상호커뮤니케이션을 위한 음성인식기술, 이러한 기술을 운영하기 위한 소프트웨어 플랫폼 등 다양한 첨단 기술들을 활용하고 있다[2].

단말 플랫폼 기술로는 Car PC, 임베디드 SW, 차량 내 기기제어, 음성인식 및 합성 기술 등이 연구되고 있는데 컴퓨팅 모듈은 모바일 CPU, 윈도우 CE, 음성인식을 위한 메모리 내장형태의 장치로 구성되어 있다. 현재 텔레매틱스 단말을 위한 운영체제에 대한 논의가 진행중인 가운데 표준으로 대두되고 있는 것이 마이크로소프트사의 Windows CE for Automotive이다. Windows CE for Automotive는 개방형 플랫폼을 지향하고 있어서 컴퓨팅 플랫폼의 선택, 하드웨어 주변장치와 소프트웨어 컴포넌

트의 선택에 있어 유연성을 제공한다. 같은 맥락에서 Sun은 Java Automotive를 개발하고 있다.

자동차 텔레매틱스의 기능은 크게 자동차에서 부품들의 성능을 관리하는 기능과 연결된 망을 통해 정보를 제공하는 기능 등 관리와 정보기능을 담당한다. 최근 텔레매틱스 서비스가 시장에 주목을 받으면서 국내 기업들도 텔레매틱스 관련 기능을 구현하기 위해 다양한 서비스 개발에 몰두하고 있으며, AMI-C(Automotive Multimedia Interface Collaboration)에서는 이러한 장치 관리 및 정보기능을 위한 핵심 기술을 표준화하여 제공하고 있다.

하지만 자동차 부품들은 생명과 밀접한 관계가 있는 장치요소이므로 해당 장치에 대한 제어 및 정보 제공을 사람, 제어 장치, 소프트웨어 등과 같은 제 3의 서비스 이용자에게 제공하는 것은 매우 위험한 상황이다.

그러므로 자동차 구성요소에 대한 제어 및 정보 접근을 제공하는 기반기술 뿐만 아니라 이를 이용하는 올바른 응용 서비스들을 구분하고 설치하도록 지원하는 단말 미들웨어의 중요성이 대두되고 있다. 이에 본 논문에서는 텔레매틱스 서비스를 위한 단말 미들웨어 모델을 살펴보고 자동차 부품들의 성능을 관리/제어하고 연결된 망을 통해 정보를 제공하도록 표준화한 AMI-C를 이용한 OSGi 기반의 텔레매틱스 단말 미들웨어 개발을 위한 고려사항과 응용 서비스를 구분하고 배포하기 위한 기본 요구 기술을 제안하고자 한다.

본 논문을 위해 2장에서는 AMI-C와 OSGi(Open Service Gateway Initiative)의 표준화 동향을 다루고, 3

장에서는 AMI-C 및 OSGi 기반의 텔레매틱스 단말 미들웨어 모델을 다룰 것이다. 4장에서는 3장에서 제시한 단말 미들웨어에 대한 구현 사례를 다루고 5장에서는 단말 미들웨어를 이용하는 텔레매틱스 서비스 배포를 위한 기본 모델을 제안한다.

2. 표준화 동향

현재 텔레매틱스 관련 국제 표준화 활동은 공식 표준화 기구보다는 사실 표준화 기구에서 사실표준 개발이 활발히 진행되고 있는데, 자동차업체 및 관련 소프트웨어 업체가 주도적으로 활동하는 사실 표준화 기구인 AMI-C와 OSGi 표준화 포럼이 대표적이다. 또한 전통적인 ITS의 일환으로 텔레매틱스 관련 표준화 활동을 추진 중인 국제 표준화 기구인 ISO 산하 TC204와 유럽지역의 국가들이 참여하는 지역 표준화 포럼으로 유럽의 ERTICO에서 진행 중이다.

2.1. OSGi 표준화 동향

1998년 Ericsson, Sun 및 IBM이 주축이 되어 시작된 OSGi는 인터넷 기반의 다양한 서비스 제공자, 네트워크 운영자, 장치와 게이트웨이 제작자간의 일관성 있는 서비스 접속을 통한 상호 운용성 확보를 추구한다. 또한, 집, 차량, 모바일 그리고 기타 다른 환경의 네트워크상에서도 관리되는 다양한 서비스들이 전달 가능한 프레임워크 표준을 개발하여 왔다. 2000년 5월 게이트웨이 스펙 1.0이 개발된 이후 2003년 3월 Release 3가 발표되었다. OSGi 서비스 플랫폼은 서비스 제공자, 장치 개발자, 그리고 다른 업체들이 빌딩, 가정, 휴대폰, 차량 그리고 다른 운용 환경 하에 풍부한 서비스를 원격 또는 동적으로 전달, 통합, 관리하도록 해주는 공통 플랫폼을 제공하는 개방형 소프트웨어 표준이다. OSGi Expert Group은 OSGi 플랫폼 규격을 개발하기 위한 실무 개발자 모임으로 CPEG, VEG, 그리고 MEG의 세 개 그룹으로 구성되어 있다. 텔레매틱스 관련하여서는 VEG에서 해당 규격을 개발중이고 BMW, Siemens VDO, IBM, AMI-C 등에서 주로 참여하여 활동하고 있다. 현재 OSGi 플랫폼 R4에는 4개의 차량관련 스펙이 포함된 것을 볼 수 있다. CPEG은 OSGi 플랫폼의 아키텍처에 관한 표준화를 진행하며 VEG와 MEG에서 제기된 표준안 사항을 공동으로 검토하고 이를 플랫폼에 채택할지의 여부를 결정한다 [3][5].

2.2. AMI-C 표준화 동향

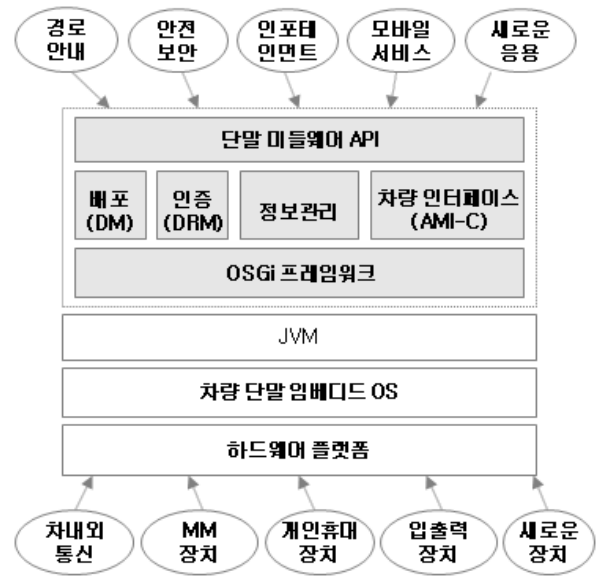
AMI-C는 세계 주력 자동차업체를 중심으로 구성된 사실 표준화 기구로서 Vehicle Communication Networks에 대한 공동된 차량용 멀티미디어 및 텔레매틱스 인터페이스 표준을 개발하는 단체이다. 주요 자동차 업체를 중심으로 결성된 AMI-C에는 Fiat, Ford Motor Co., General Motors Corp., Honda, Nissan, Toyota 등의 업체가 참여하고 있다. AMI-C에서는 release로 정의된

일련의 소프트웨어 규격을 개발하고 있는데 이는 기본적으로 소프트웨어 개발자들이 어떤 종류의 차량 하드웨어 및 플랫폼상에서도 동작하는 응용을 개발하도록 지원하는 동일한 APIs를 정의한다. 2001년 1월 Release 1을 발표한 뒤, 2003년 Release 2가 발표되었다. Release 2는 차량 내에서 인터페이스 사양개발에 초점이 맞추어졌으며, 향후 Release 3는 차량으로부터의 통신에 집중하여 사양을 개발할 예정이다 [4][5].

3. 텔레매틱스 단말 미들웨어 모델

AMI-C는 OSGi 기반의 차량 정보 관리 및 장치 제어에 대한 인터페이스 표준을 제시하고 있다. 이에 본 연구에서의 텔레매틱스 단말 미들웨어 모델은 OSGi 프레임워크를 기반으로 AMI-C 인터페이스 및 정보관리 구성요소가 하드웨어 플랫폼을 통한 차량 장치들에 대한 연동을 담당하고 인증 및 배포 구성요소를 통해 단말 미들웨어를 이용하고자 하는 어플리케이션들에 대한 검증 및 설치를 지원하도록 구성한다.

텔레매틱스 어플리케이션은 이러한 구성요소들을 이용할 수 있도록 정의한 단말 미들웨어 API를 통해 해당 기능들을 이용할 수 있다. (그림 1)은 본 논문에서의 텔레매틱스 단말 미들웨어에 대한 모델을 보여준다.



(그림 1) 텔레매틱스 단말 미들웨어 구조도

차량 인터페이스 및 정보관리는 차량을 구성하는 각종 장치(예:타이어, 엔진 등)에 접근하여 각종 상태(예:속도계, 타이어 상태, 오일 상태, 엔진 상태 등) 정보를 직/간접적으로 수집/제어하도록 지원하는 표준 인터페이스이다.

인증은 텔레매틱스 서버와 연동하여 텔레매틱스 단말로 단말 미들웨어를 이용한 단말 서비스를 제공할 수 있도록 서비스 인증 및 라이선스 발급 관리 기능을 제공한다. 차량 인터페이스는 인터페이스가 호출될 경우 인증 모듈에 호출자에 대한 라이선스를 요청하고 검증을 수행한 후 기능을 제공한다.

배포는 단말 미들웨어의 지속적인 업데이트 및 단말 어플리케이션의 배포를 위한 텔레매틱스 서버와의 연동 및 소프트웨어 관리를 지원한다.

4. 텔레매틱스 단말 미들웨어 구현

본 논문에서는 (그림 1)에서 제시한 모델의 구성요소들을 OSGi 번들로 구현하였다. 실제 AMI-C에서는 관련 인터페이스를 OSGi 번들로 정의하여 배포하여 업체에서 필요한 인터페이스에 대해 구현하도록 제안하고 있다. 특히, 차량에 대한 실제 장치를 보유하고 있지 않기 때문에 차량 장치에 대한 에뮬레이터를 구현하여 적용하였다[3].

4.1. 차량 에뮬레이터의 구현

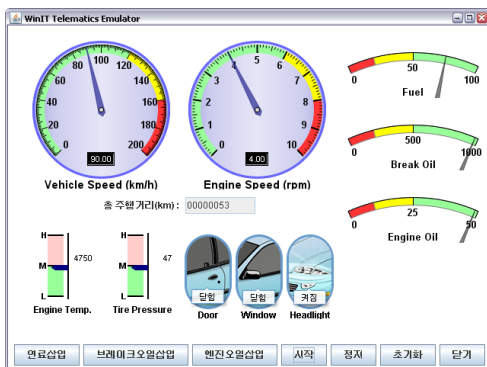
차량 에뮬레이터가 가져야할 중요한 특성은 실제 차량이 제공하고자 하는 상태 정보와 제어를 수용할 수 있도록 구현해야한다는 점이다. 특히, 모든 차량 장치들은 AMI-C 네트워크라는 차량 네트워크를 통해 각 장치에 대한 정보를 주고받게 되는데, 이러한 송/수신 정보를 공통 메시지 집합(Common Message Set)이라고 한다[6].

이러한 공통 메시지 집합 중 본 과제와 관련된 차량 정보에 대한 부분은 다음 [표 1]과 같이 정의하여 16bit 문자열로 출력되도록 구현하였다.

[표 2] 공통 메시지 형식 정의

구분	종류
메시지 생성 형식	Message Type + Message Class + Object Property + Operands
메시지 타입 (Message Type)	INQUIRE, REPORT, SET, COMMAND, CONFIRM, WARNING
메시지 클래스 (Message Class)	Management, Core, Body Module, Powertrain, Vehicle Diagnostics, Amplifier, General Player, Disk Media, Tape Media, Tuner, General Phone, Advanced Phone, Text Display
개체 속성 (Object Property)	Door, Window, Headlights, VehicleSpeed, Fuel, TirePressure, CoolantTemperature, EngineCoolantLevel, EngineRpm, BrakeFluidLevelStatus, EngineOilLevel

(그림 2)는 차량 에뮬레이터를 나타낸다. 그래픽 컨트롤을 이용하여 해당 장치의 CMS 값을 실시간으로 확인할 수 있도록 구성하였다.



(그림 2) 차량 에뮬레이터 구현 모습

4.2. 차량 인터페이스 구현

차량 인터페이스는 텔레매틱스 단말에 설치되는 다양한 텔레매틱스 응용 어플리케이션들이 필요한 서비스를 제공하기 위해 필요한 차량 구성 장치들에 대한 상태를 파악하고 제어할 수 있도록 지원하도록 구현하였다. 차량 인터페이스는 해당 차량 장치들에 대한 실제 상태값을 얻어오거나 설정하기 어렵기 때문에 앞서 구현한 차량 에뮬레이터에 접속하여 필요한 사항을 만족할 수 있도록 지원한다.

[표 2]는 본 과제에서 구현한 차량 인터페이스에 대한 리스트이다. 해당 인터페이스는 AMI-C에서 표준으로 제시하고 있으며 해당 인터페이스에 대한 구현부를 OSGi 번들로 구현하였다[7][8].

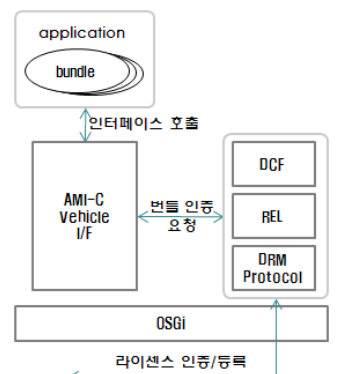
[표 3] 차량 인터페이스 정의

분류	인터페이스명
BrakeFluidLevel	public float getBrakeFluidLevel()
EngineCoolant	public float getEngineCoolantTemp()
	public float getEngineCoolantLevel()
EngineOil	public float getEngineOilLevel()
EngineSpeed	public int getEngineSpeed()
FuelLevel	public float getFuelLevel()
TirePressure	public float getTirePressure()
VehicleSpeed	public float getVehicleSpeed()
OdometerEvent	public float getDifferentialOdometerDistance()
Door	public void lockDoor(boolean type)
	public boolean isDoorLocked();
Window	public void setWindowClosed(boolean type)
	isWindowClosed()
Headlight	public org.ami_c.vehicle.types.HeadlightStatusType getHeadlights();
	public void setHeadlights(org.ami_c.vehicle.types.HeadlightStatusType state);

4.3. 인증 구현

인증은 단말 어플리케이션이 차량 인터페이스를 호출하고자 하는 경우, 차량 인터페이스는 호출 어플리케이션에 대한 검증을 수행한다.

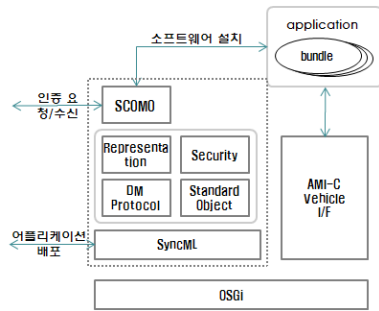
서비스 배포자로부터 다운로드된 서비스에 대한 관리객체(RO:Rights Object)를 수행하도록 권리 발급자(RI: Rights Issuer)로부터 다운로드 받는다. 어플리케이션은 유일한 개인키/공개키 쌍과 인증서를 갖고 있어 권리 발급자로부터 인증받을 수 있다. 서비스에 대한 인증이 완료되면 차량 인터페이스는 해당 어플리케이션에 차량 장치에 대한 접근 및 제어를 위한 권한(Rights)을 발급하고 서비스를 제공한다. (그림 3)은 차량 인터페이스의 어플리케이션에 대한 인증을 수행하는 소프트웨어 구조를 보여준다.



(그림 3) 인증 구조도

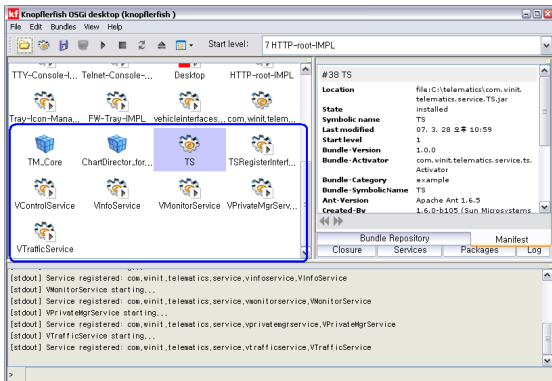
4.4. 배포 구현

배포는 텔레매틱스 단말 내에서 이루어지는 단말 미들웨어 및 소프트웨어에 대한 다운로드, 설치, 설치 후 바이너리의 삭제 등의 소프트웨어 업데이트를 수행한다. 단말 미들웨어 및 소프트웨어에 대한 인증은 '4.3. 인증'에서 정의한 인증 모듈을 사용한다. (그림 4)는 단말 미들웨어 및 어플리케이션 배포를 위한 소프트웨어 구조를 보여준다.



(그림 4) 배포 구조도

SCOMO 모듈은 텔레매틱스 단말기 상에서 소프트웨어 컴포넌트의 설치, 업그레이드, 삭제 등 일련의 소프트웨어 컴포넌트 관리를 위한 관리 오브젝트(MO: Management Object)를 정의하고 이 규격을 통해 어플리케이션에 대한 설치를 수행한다. (그림 5)는 OSGi 프레임워크에 단말 미들웨어 및 서비스 번들이 설치된 후의 모습을 보여준다.



(그림 5) 단말 미들웨어/서비스 번들 배포 예

4.5. 단말 서비스 구현

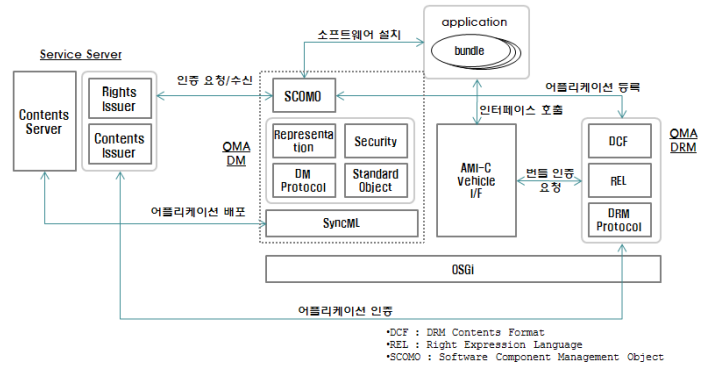
다음 (그림 6)은 앞에서 다룬 단말 미들웨어를 이용하여 차량 서비스를 제공하는 단말 서비스의 구현 예를 보여준다.



(그림 6) 단말 서비스 구현 예

5. 텔레매틱스 소프트웨어 배포를 위한 기본 모델

텔레매틱스 단말에 대한 미들웨어 및 서비스를 배포하기 위해서는 앞에서 정의한 인증 컴포넌트와 해당 소프트웨어를 배포하기 위한 배포 서버가 필요하다. (그림 6)은 텔레매틱스 서비스를 위해 필요한 단말 소프트웨어 배포를 위한 소프트웨어 구조도를 보여준다.



(그림 7) 텔레매틱스 단말 미들웨어/서비스 배포를 위한 소프트웨어 구조도

서비스 서버는 단말로부터의 권리(Rights) 관리 요청 처리를 위한 권리 발급자(Rights Issuer), 소프트웨어 배포 요청 처리를 위한 콘텐츠 발급자(Contents Issuer), 해당 소프트웨어를 저장하고 배포하는 콘텐츠 서버(Contents Server)로 구성된다.

단말 미들웨어가 텔레매틱스 서비스 배포에 필요한 서비스 검색을 위한 기본 요구사항은 다음 [표 3]과 같다 [8].

[표 4] 텔레매틱스 소프트웨어 배포를 위한 기본 요구사항

- 시스템에 서비스가 설치될 때 사용될 수 있는 신원확인 활성 서비스인 서비스 검색 절차를 지원해야 한다.
- 서비스는 이름과 버전, 서비스의 형태, 서비스를 제공하는 장치와 같은 정보를 관리자에 제공해야 한다.
- 서비스 검색 절차는 단말 미들웨어가 초기화될 때 서비스를 사용할 수 있도록 제정되어야 한다.
- 서비스 검색 절차는 서비스의 새로운 서비스가 추가 되었거나 지워졌을 때 사용 가능한 리스트를 갱신해야 한다.
- 단말 미들웨어는 차량 장치에 대한 검색을 위해 위해서는 이전의 서비스 검색을 완료 해야만 한다.
- 만약 서비스 등록이 실패한다면, 단말미들웨어는 서비스 레지스트리로부터 사용할 수 없는 서비스를 발견하고 삭제해야 한다.
- 컴포넌트는 서비스의 기능이 복구 되었을 때 레지스트리의 서비스를 복구한다.

7. 결론 및 향후 연구 방향

최근의 정보통신 기술의 발전으로 집이나 사무실에서 이용되는 컴퓨팅, 인터넷, 통신 어플리케이션들은 점차 모바일 환경으로 변화되어 가고 있다. 특히, 차량 이동중 여러 가지 업무, 정보이용, 엔터테인먼트 등 이용자들의

요구를 수용하기 위해서, 향후 자동차는 모바일 IT 솔루션의 중요한 부분으로 빠르게 성장할 것으로 전망되며, 이러한 시장변화의 중심에 텔레매틱스가 자리잡고 있다.

텔레매틱스 시장은 여러 가지 기술들이 독립적으로 분할되어 다양한 제품으로 정보를 제공하기 보다는 여러 정보 및 기능 등이 텔레매틱스 시스템으로 집중 및 통합되는 형태가 요구된다. 현재 텔레매틱스 시장은 기회와 도전이 양존해 있으며, 특히, 이동통신 및 무선컴퓨팅 시장과의 관계와 같이 다른 관련 시장들과의 시너지 효과를 내거나 경쟁관계에 있다.

본 논문에서는 텔레매틱스 서비스를 제공하는 단말의 구조 및 그 특성과 정보를 살펴보고, 텔레매틱스 소프트웨어에 대한 배포 모델에 대한 구조 및 정보를 살펴봄으로써, 차량 장치를 수용하고 다양한 텔레매틱스 서비스를 제공할 수 있는 텔레매틱스 단말 미들웨어를 구현하였다. 또한 이를 바탕으로 향후 텔레매틱스 단말 미들웨어를 탑재한 텔레매틱스 서비스 개발을 위한 요소 기술을 위해 구현하였다.

향후 연구 방향으로는 OSGi 기반의 단말 미들웨어를 WIPi, HSDPA와 같은 단말 플랫폼 기반에서의 확장과 복합 단말 환경에서의 확장을 위한 기본 요구 기술을 정의할 수 있다.

8. 참고문헌

- [1] 전자부품연구원 전자정보센터, “텔레매틱스(Telematics) 시장 동향”, Nov.2003.
- [2] 임광선, “텔레매틱스 기술 및 산업 동향”, ETRI 정보화 기술연구소 저널.
- [3] <http://www.osgi.org>.
- [4] <http://www.ami-c.org>
- [5] 김경호, “텔레매틱스 표준화 로드맵 및 표준개발방향,” TTA 텔레매틱스 표준기술 세미나, 2004. 9.
- [6] AMI-C 2002, "AMI-C Common Message Set v1.01".
- [7] AMI-C 5001, "AMI-C software API specifications-core API's v1.00".
- [8] AMI-C 1003, "AMI-C Release 2 architectural overview v1.00".