

어플리케이션 서비스 제공자(ASP)를 이용한 텔레매틱스 서비스 제공 시스템

Telematics Service Providing System Using ASP: Application Service Provider

성경복, 김주완, 장병태

(ETRI 텔레매틱스 솔루션 연구팀, 연구원, 선임연구원, 책임연구원)

Key Words : 텔레매틱스, 어플리케이션 서비스 제공자

목 차

1. 서론
2. 관련연구
3. 어플리케이션 서비스 제공자(ASP)
4. 어플리케이션 서비스 제공자를 이용한 텔레매틱스 서비스 제공 시스템
5. 결론

I. 서론

모바일 컴퓨팅 및 무선통신 기술의 발전으로 인하여 차내 이동형 정보서비스를 지원하는 텔레매틱스에 대한 관심이 고조되고 있다. 텔레매틱스는 위치정보와 이동통신망을 이용해 운전자와 탑승자에게 교통안내, 긴급구난, 영화, 게임 등의 정보를 제공하는 종합적인 멀티미디어 정보서비스를 말한다[1].

현재 텔레매틱스 서비스는 대부분이 텔레매틱스 단말기에 서비스를 위한 소프트웨어가 설치되어 서비스를 제공하고 있다. 그러나 단말기에 소프트웨어가 설치되어 서비스가 제공되는 경우에는 소프트웨어의 수정 및 업데이트가 어렵고, 제공되는 서비스의 확장이 어렵다는 단점이 있다. 또한 소프트웨어 설치비용으로 인한 단말기의 초기 가격이 상승하는 문제도 있다.

일부 텔레매틱스 서비스는 텔레매틱스 단말기에 별도의 소프트웨어 설치 없이 인터넷 연결을 통해 서비스를 제공한다. 인터넷을 통한 텔레매틱스 서비스 제공 방식은 텔레매틱스 단말기에 소프트웨어를 설치하는 방식에서 나타나는 단점이 발생하지 않으나, 인터넷 단절시 서비스가 불가능하다는 점과 저속 무선 통신망을 이용하는 텔레매틱스 환경으로 인하여 서비스 실행 속도가 늦고, 과도한 통신비용이 발생하는 단점이 있다.

이러한 문제를 해결하기 위해, 본 논문에서는 어플리케이션 서비스 제공자(Application Service Provider: ASP)를 이용하여 소프트웨어를 제공하는 텔레매틱스 서비스 시스템을 설계하고 구현한다. 이는 소프트웨어를 처음부터 단말기에 설치하지 않고, 서비스를 사용하려는 시점에 소프트웨어를 다운받아 사용하는 방법을 이용하여 텔레매틱스 서비스를 제공하는 시스템이다.

이를 위해 1)어플리케이션 서비스 제공자를 이용하여 텔레매틱스 서비스를 제공하는 시스템을 설계한다. 이 때, 사용자가 쉽게 소프트웨어를 찾고 서비스 받을 수 있도록 텔레매틱스 포털 서버를 구성한다. 그리고 2)무선 환경을 고려하여 소프트웨어의 일부분만 다운받아 실행하는 방법을 제안한다. 소프트웨어를 일부만 다운 받는 것이 가능하도록 소프트웨어를 분할하는 방법에 대해 설명하고, 분할된 소프트웨어를 실행하기 위한 소프트웨어 가상 실행환경에 대해 설명한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련연구로서 텔레매틱스 서비스 제공 방법을 소개한다. 제 3장에서는 본 논문에서 텔레매틱스 서비스를 제공하는데 사용하려는 방식인 어플리케이션 서비스 제공자에 대해 소개하고, 제 4장에서는 본 논문에서 제안하는 어플리케이션 서비스 제공자를 이용한 텔레매틱스 서비스 제공 시스템에 대해 설명한다. 마지막으로 제 5장에서 결론을 내린다.

II. 관련연구

기존의 텔레매틱스 서비스는 대부분이 클라이언트에 소프트웨어가 설치되어 서비스가 제공되는 시스템이 주를 이루고 있으며, 그 외에 무선 통신 기술의 개발로 인하여 인터넷을 이용한 다양한 기능이 추가되고 있다[2].

Mercedes-Benz의 경우, TELEAID라는 이름으로 텔레매틱스 서비스를 제공하는데, 주로 긴급 전화, 전자지도, 차량간 통신 서비스를 제공한다[3]. General Motors(GM)는 OnStar라는 텔레매틱스 서비스를 통해서 도난 차량 추적, 응급 상황 서비스, 원격 진단 등의 안전-보안 서비스와 주식, 뉴스, 위치 기반 날씨, 교통 정보 등 충고 서비스를 하고 있다[4]. 이러한 서비스들은 텔레매틱스 단말기 제조시부터 해당 서비스를 위

한 소프트웨어를 텔레매틱스 단말기에 설치하여 제공되거나 콜센터(Call Center)를 이용해서 서비스 받는다. 이 경우, 서비스를 위한 소프트웨어가 기본적으로 설치되어 나오기 때문에 서비스의 확장이나 기존 서비스의 업그레이드가 쉽지 않고, 텔레매틱스 단말기 가격에 소프트웨어 가격이 포함되어 단말기 가격이 높아지는 단점이 있다.

Toyota의 텔레매틱스 서비스인 G-Book 서비스에서는 네비게이션 등의 항법 서비스, 뉴스, 일기예보 등의 정보서비스를 제공하고 이외에도 엔터테인먼트 및 보안-안전 서비스를 제공하고 있다[5]. Toyota는 이러한 서비스를 휴대 전화 접속 등을 통한 인터넷 서비스 형태로 제공한다. 인터넷을 통한 서비스 제공 형태는 서비스가 텔레매틱스 단말기에 설치되어 제공되는 방식의 단점인 서비스 확장 및 업그레이드 문제를 쉽게 해결하였다. 그러나 이처럼 인터넷을 통한 서비스 제공 방법은 인터넷 단절시 서비스가 제공되지 않는다는 점과 서비스의 실행 속도가 늦고, 과도한 통신비용이 발생한다는 단점이 있다.

본 논문에서는 어플리케이션 서비스 제공자 방식을 이용하여 텔레매틱스 서비스를 제공하는 방법을 제안한다. 소프트웨어가 단말에 설치되는 경우와 비교하여 어플리케이션 서비스 제공자 방식을 이용할 경우에, 소프트웨어의 유지 보수와 서비스의 확장이 쉽고, 단말기 가격이 저렴해지는 장점이 있다. 인터넷으로 텔레매틱스 서비스를 제공하는 방식과 비교하여서는, 인터넷 단절 시에도 지속적으로 서비스가 가능하고, 서비스 실행 속도가 빠르다는 장점이 있다.

III. 어플리케이션 서비스 제공자(ASP)

어플리케이션 서비스 제공자(Application Service Provider; ASP)란 네트워크를 통하여 원격으로 소프트웨어 기반의 서비스나 솔루션을 제공하는 서비스 또는 제공자이다[6]. 어플리케이션 서비스 제공자에게 서비스를 받는 사용자는 전체 소프트웨어를 구매하지 않고 일정 금액을 내고 해당 프로그램을 빌려 쓰게 된다.

인터넷 서비스 제공자(Internet Service Provider)가 단순히 통신망과 데이터 센터를 갖추고 서버 호스팅 등 인터넷 서비스를 제공하는 반면에 어플리케이션 서비스 제공자는 인터넷 서비스 제공자가 제공하는 서비스와 함께 인터넷을 통하여 소프트웨어를 제공한다. 어플리케이션 서비스 제공자는 여러 업체가 필요로 하는 응용업무시스템(Application)을 특정 서버에만 설치하고 고객으로 하여금 네트워크 접속을 통해 소프트웨어를 빌려 쓰도록 한다. 소비자 입장에서는 고가의 소프트웨어를 구매할 필요 없이 어플리케이션 서비스 제공자를 통해 소프트웨어를 제공받아 사용하면 된다. 소비자는 매달 또는 사용량만큼 비용을 지불하면 되므로 초기 투자비용을 절감할 수 있을 뿐 아니라 시스템 관리도 대폭 줄일 수 있다. 또한 사용량에 따라 월 일정액의 비용을 청구함으로써 고객은 비용을 예측하고 조절할 수 있어 합리적인 사업계획을 세울 수 있다.

소프트웨어를 직접 설치하여 사용하는 경우와 비교하여 어플리케이션 서비스 제공자를 통해 소프트웨어를 제공받아 사

용할 때의 장점은 다음과 같다. 먼저, 시스템을 위한 소프트웨어 초기 구입비용이 줄어든다. 고가의 소프트웨어를 구입하지 않고 빌려서 사용하므로, 빌려서 사용한 만큼의 비용만 지불하면 되기 때문이다. 둘째로, 소프트웨어의 유지 관리가 쉽다. 소프트웨어를 구매하여 사용할 경우, 사용자가 지속적으로 해당 소프트웨어를 유지 관리해야만 한다. 그러나 어플리케이션 서비스 제공자를 이용할 경우, 소프트웨어의 유지 관리를 어플리케이션 서비스 제공자가 대신하게 되므로 사용자 측면에서 소프트웨어의 관리가 쉽다. 관리자 측면에서는 배포되어있는 소프트웨어를 모두 관리해야 하는 것이 아니라 중앙 집중적인 소프트웨어 관리로 인해, 자신이 서비스하는 소프트웨어를 관리하면 되기 때문에 소프트웨어의 유지 관리가 쉬워진다. 셋째로, 서비스의 확장이 쉽다. 서비스를 위해 모든 시스템에 소프트웨어를 설치할 필요 없이, 소프트웨어를 관리하는 중앙 데이터 센터에 소프트웨어를 등록하면 원하는 사용자는 해당 서비스를 제공 받을 수 있다.

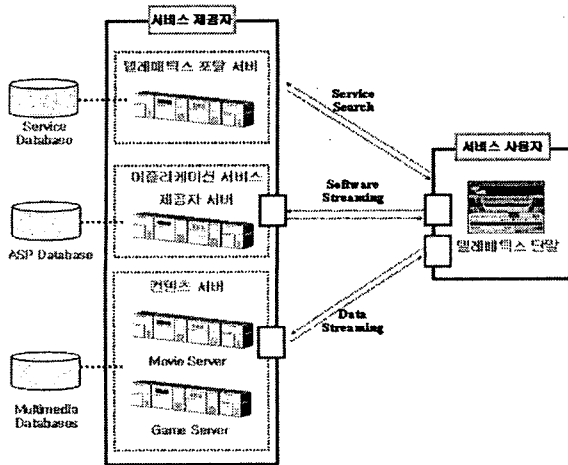
이와 같은 어플리케이션 서비스 제공자를 이용할 때의 장점은 텔레매틱스 서비스를 제공할 때에도 그대로 적용이 된다. 어플리케이션 서비스 제공자 방식으로 서비스를 제공하면, 먼저 소프트웨어의 설치가 필요 없기 때문에 사용자는 원하지 않는 소프트웨어를 위해 비용을 지불할 필요가 없고, 단말기 제공자는 소프트웨어 설치비용이 없어 단말기 가격을 낮출 수 있다. 또한 소프트웨어의 유지 보수가 쉬워 텔레매틱스 서비스의 안정성을 높일 수 있다. 그리고 서비스의 확장이 쉬워 사용자에게 다양한 서비스를 빠르게 제공할 수 있다.

IV. 어플리케이션 서비스 제공자를 이용한 텔레매틱스 서비스 제공 시스템

본 장에서는 1)어플리케이션 서비스 제공자를 이용하여 텔레매틱스 서비스를 제공하는 텔레매틱스 서비스 제공 시스템을 제안하고, 2)제안한 시스템 상에서 효율적인 소프트웨어 제공을 위해 필요한 소프트웨어 분할 과정에 대해 설명한다. 그리고 3)분할된 소프트웨어의 실행을 위한 소프트웨어 가상 실행환경에 대해 설명한다. 마지막으로 4)서비스 제공 방법에 대해 설명한다.

1. 시스템 아키텍처

본 논문에서 제안하는 텔레매틱스 서비스 제공 시스템은 <그림 1>과 같은 아키텍처를 가진다. 먼저 서비스를 제공하기 위한 시스템으로 사용자에게 어떤 서비스가 있는지 알리고, 서비스를 분류하고 서비스 검색을 지원하는 텔레매틱스 포털 서버가 존재한다. 그리고 어플리케이션 서비스 제공 방식으로 서비스를 제공하기 위한 어플리케이션 서비스 제공자 서버가 존재한다. 이들과 함께 실제로 서비스에 대한 콘텐츠를 제공하는 콘텐츠 서버가 존재한다. 사용자가 서비스를 받기 위한 시스템으로 텔레매틱스 단말기가 존재한다. 텔레매틱스 포털 서버, 어플리케이션 서비스 제공자 서버, 콘텐츠 서버는 여러 사업자가 독립적으로 운영할 수 있다.



<그림 1> 텔레매틱스 서비스 제공 플랫폼

텔레매틱스 포탈 서버는 텔레매틱스 서비스를 받기 위해 사용자가 최초로 접속하는 접속 지점(Access point)의 역할을 한다. 텔레매틱스 포탈 서버는 크게 서비스 관리, 사용자 관리, 과금 관리의 세가지 역할을 수행한다. 서비스 관리에는 서비스의 분류, 자주 사용되는 서비스들의 순위, 서비스 검색 등의 기능이 존재하고, 사용자 관리에는 사용자의 아이디, 패스워드 관리, 개인 환경 설정, 자주 사용하는 프로그램의 목록 등 개인화 관리 기능이 존재한다. 과금 관리에는 사용자가 사용한 서비스의 시간, 양 등의 기록을 이용하여 사용 비용을 관리하는 기능이 존재한다.

어플리케이션 서비스 제공자 서버는 사용자가 선택한 서비스를 어플리케이션 서비스 제공 방식으로 사용자에게 제공하는 역할을 한다. 사용자에게서 소프트웨어의 요청이 있으면, 어플리케이션 서비스 제공자 서버에서는 해당 소프트웨어가 구동될 수 있는 최소한의 소프트웨어 조각만을 사용자에게 전달한다. 이를 위해서는 먼저 소프트웨어를 일정 크기로 분할하는 과정인 소프트웨어 분할 과정을 수행한 후 어플리케이션 서비스 제공자 서버에 저장한다. 소프트웨어 분할 과정과 서비스 제공 과정에 대하여는 다음 절에서 자세히 설명한다.

컨텐츠 제공 서버는 실제적으로 소프트웨어 이외의 콘텐츠를 제공하는 서버이다. 예를 들어, 음악 서비스를 위해 음악 파일을 가지고 있는 음악 서버나 영화 서비스를 위한 영화 서버가 콘텐츠 제공 서버의 한 종류이다.

텔레매틱스 단말기는 사용자가 텔레매틱스 서비스를 받기 위하여 사용하는 텔레매틱스 서비스 기기이다. 텔레매틱스 단말기에는 서비스를 위해 텔레매틱스 포탈 클라이언트와 어플리케이션 서비스 제공 클라이언트가 설치되어야 한다. 텔레매틱스 포탈 클라이언트는 텔레매틱스 포탈 서버에 접속하여 서비스에 대한 정보를 받아오고, 사용자가 선택한 서비스를 어플리케이션 서비스 제공 클라이언트에게 알린다. 어플리케이션 서비스 제공 클라이언트는 어플리케이션 서비스 제공자 서버에 접속하여 소프트웨어 구동을 위한 최소한의 소프트웨어 블록을 받아오고, 받아온 블록을 이용하여 소프트웨어를 구동할 수 있는 가상 소프트웨어 실행환경을 구성하여 소프트웨어를 구동한다. 텔레매틱스 포탈 클라이언트와 어플리케이션 서비스 제공자 클라이언트는 텔레매틱스 단말기 제작시

부터 설치될 수 있으나, 다양한 텔레매틱스 포탈 서버 및 어플리케이션 서비스 제공 서버를 지원하기 위해서 실시간으로 다운로드 되어 설치된다.

위와 같은 아키텍처에서 서비스 제공은 다음과 같은 순서로 이루어진다. 먼저 어플리케이션 서비스 제공자는 서비스하고자 하는 소프트웨어를 분할하여 어플리케이션 서비스 제공 서버에 저장한다. 그 후 텔레매틱스 포탈 서버에 소프트웨어가 등록되었다는 것을 알린다. 컨텐츠 제공자도 역시 콘텐츠를 컨텐츠 서버에 저장하고 텔레매틱스 포탈 서버에 콘텐츠가 저장되었다는 것을 알린다. 텔레매틱스 포탈 서버는 자신에게 등록되어있는 소프트웨어와 콘텐츠를 사용자가 원하는 서비스 및 콘텐츠를 찾기 쉽도록 분류, 관리한다. 사용자는 텔레매틱스 포탈 클라이언트를 이용하여 텔레매틱스 포탈 서버에 접속한다. 포탈 서버에 접속한 후 서비스를 검색하여 자신이 원하는 서비스를 선택한다. 그러면 선택된 서비스에 대한 정보가 텔레매틱스 포탈 클라이언트로부터 어플리케이션 서비스 제공 클라이언트로 전달되고, 어플리케이션 서비스 제공 클라이언트는 어플리케이션 서비스 제공 서버에 접속하여 선택된 서비스에 맞는 소프트웨어의 최소 실행 블록을 전달받는다. 어플리케이션 서비스 제공 클라이언트는 소프트웨어 실행을 위한 가상환경을 구성하고, 전달받은 소프트웨어 블록을 이용하여 소프트웨어를 실행하여 사용자에게 서비스를 제공한다.

2. 소프트웨어 분할 전송

어플리케이션 서비스 제공자를 이용하여 텔레매틱스 서비스를 제공할 경우 가장 문제가 되는 것은 통신량이다. 텔레매틱스의 경우, 차량의 이동환경 때문에 저속의 무선 통신망을 이용하게 되는데, 이때 소프트웨어의 사이즈가 큰 대용량 소프트웨어를 어플리케이션 서비스 제공자 방식을 이용하여 서비스하게 되면 서비스가 실행되기까지 오랜 시간이 소요될 뿐만 아니라, 많은 통신량으로 인해 과도한 통신비용이 발생하게 된다.

이러한 문제를 해결하기 위해 본 논문에서는 소프트웨어를 분할하여 저장하였다가, 요청이 있을 경우 필요한 부분만 전송하는 소프트웨어 분할 전송 기법을 사용한다. 이는 소프트웨어의 모든 부분을 다운 받아서 실행하는 것이 아니라 일부 부분만 다운 받아서 소프트웨어를 실행하는 것을 의미한다. 이렇게 소프트웨어의 일부만 다운받아 실행할 때의 장점은 사용하지 않는 기능을 위해 소프트웨어를 모두 다운받지 않아도 된다는 것이다. 필요한 부분만 다운 받아 사용하기 때문에 다운로드 되는 소프트웨어의 사이즈가 매우 줄어든다. 이것은 사이즈가 큰 대용량 소프트웨어라도 구동시간이 오래 걸리지 않고, 통신비용이 줄어드는 장점이 있다. 이러한 특징은 텔레매틱스 환경과 같이 통신 속도가 저속이고, 비용이 비싼 무선 환경에서는 큰 장점이 된다. 소프트웨어의 필요한 부분만 다운받아 사용하는 방식의 단점은 일부만 다운 받은 후 소프트웨어를 구동하기 때문에, 다운 받지 않은 부분에 대한 기능 사용시 다운로드를 추가로 수행해야 하므로, 소프트웨어의 반응 시간이 느려진다는 것이다. 그러나 이러한 단점은 캐쉬(Cache)와 프리패치(Pre-fetch)를 이용하여 극복 할 수 있다.

3. 소프트웨어 가상 실행환경

분할된 소프트웨어의 일부 블록만을 이용하여 소프트웨어를 실행하기 위해서는 소프트웨어 가상 실행환경이 필요하다. 소프트웨어 가상 실행환경이란 소프트웨어의 일부만 존재하더라도 소프트웨어의 모든 부분이 있는 것처럼 실행환경을 만들어주는 것이다. 기본적인 운영체제에서는 소프트웨어의 모든 부분이 있어야 소프트웨어가 실행되기 때문에, 소프트웨어 가상 실행환경을 이용하여 소프트웨어의 모든 부분이 존재하는 것처럼 만들어 운영체제에서 소프트웨어를 실행할 수 있게 한다.

각 소프트웨어 마다 실행환경을 알아내는 과정은 다음과 같다. 먼저 소프트웨어를 설치하기 위한 단말기의 현재 상태를 알아낸다. 즉, 파일의 개수, 구성 및 레지스트리를 검색한다. 다음으로 소프트웨어를 설치한다. 마지막으로 소프트웨어 설치 이후에 단말기의 상태를 알아낸다. 그 후, 소프트웨어 설치 이전의 단말기 상태와 설치 이후의 단말기 상태를 분석하여 변경사항을 추적한다. 추적된 변경사항을 저장하여 해당 소프트웨어의 실행환경으로 기록한다. 기록된 실행환경을 이용하여 나중에 텔레매틱스 단말에서 소프트웨어 실행 시, 소프트웨어가 이미 설치된 것과 동일한 효과를 내기 위해 소프트웨어 가상 실행환경을 구축하여 소프트웨어를 실행하게 된다.

4. 서비스 제공

사용자가 어플리케이션 서비스 제공자를 이용하여 텔레매틱스 서비스를 제공받는 과정은 다음과 같다. 먼저 사용자는 자신이 원하는 소프트웨어를 찾기 위해 먼저 텔레매틱스 포털 서버에 접속한다. 이때는 범용의 브라우저를 이용하여 텔레매틱스 포털 서버에 접속한다. 텔레매틱스 포털 서버 접속한 후 해당 포털서버에 맞는 포털 클라이언트가 설치된다. 사용자는 텔레매틱스 포털 서버에서 검색 등을 이용하여 자신에게 맞는 서비스를 선택한다. 그러면 포털 클라이언트는 어떤 서비스가 선택되었고, 해당 서비스는 어플리케이션 서비스 제공 서버 중 어느 서버에서 제공되는지를 포털 클라이언트에게 알려준다. 그러면 포털 클라이언트는 해당 어플리케이션 서비스 제공 서버에 접속하여 어플리케이션 서비스 제공 클라이언트를 다운로드 하여 설치 한 후, 어플리케이션 서비스 제공 클라이언트는 포털 클라이언트로부터 어플리케이션 서비스 제공 서버의 정보와 사용자가 선택한 서비스에 대한 정보를 얻은 후, 소프트웨어 구동에 필요한 최소한의 블록만 가져와 소프트웨어 가상 실행환경을 이용하여 소프트웨어를 실행하게 된다. 소프트웨어 구동에 필요한 최소한의 블록만으로 소프트웨어를 실행하다가 추가적으로 블록이 필요하게 될 경우에는 다시 어플리케이션 서비스 제공 서버에 요청하여 필요한 블록을 받아온다. 어플리케이션 서비스 제공 서버는 각 어플리케이션 서비스 제공 클라이언트가 요구한 서비스의 종류와 데이터의 양을 기록하였다가, 과금 및 서비스 최적화에 사용하게 된다.

VI. 결론

본 논문에서는 어플리케이션 서비스 제공자를 이용하여 소프트웨어를 제공하는 텔레매틱스 서비스 시스템을 설계하였다. 이를 위해 1) 어플리케이션 서비스 제공자를 이용하여 텔레매틱스 서비스를 방법을 제안하였다. 사용자가 쉽게 소프트웨어를 찾고 서비스 받을 수 있도록 텔레매틱스 포털 서버를 구성하였다. 2) 무선 환경을 고려하여 소프트웨어의 일부만 다운로드 받아 실행하는 방법을 제안하였다. 소프트웨어를 일부만 다운로드 받는 것이 가능하도록 소프트웨어를 분할하는 방법에 대해 설명하고, 분할된 소프트웨어를 실행하기 위한 소프트웨어 가상 실행환경에 대해 설명하였다.

자동차가 이동 작업 환경을 급속히 필요로 함에 따라, 작업 환경 구성에 필요한 기본 응용 프로그램과 추가 응용 프로그램을 실시간 요구에 따라 온-디맨드(On-demand) 방법으로 공급할 수 있는 체계를 구성하는 것이 핵심적으로 필요하게 되었다. 본 논문에서 제안한 텔레매틱스 서비스 방식은 이러한 요구를 만족시키는 서비스 공급 방식을 제안하였다는 점에서 의미가 있다고 할 수 있다.

참고문헌

1. C.J.D. Senior, L. Legate, C. P. Thomson, "Telematics Systems From The service Perspective," IEE International Conference on Advanced Driver Assistance Systems, pp. 15-19, Sept. 2001.
2. Crofts, L.P., "In-car navigation as an aid to the business driver," IEE International Conference on Advanced Driver Assistance Systems, pp. 17-18, Sept. 2001.
3. URL: MERCEDES-BENZ REACHES DRIVERS VIA SATELLITE WITH ALL-NEW TeleAid SYSTEM Available at : <http://www.autoworld.com/news/Mercedes/TeleAid.htm>
4. URL: OnStar Corp. Available at: <http://www.onstar.com>
5. URL: G-Book, Toyota Motors Corp. Available at: <http://g-book.com>
6. Euan Davis, James Eibisch, Lars Schwaner, "ASP Industry Review and Forecase, 2001-2006," IDC, March 2002.