HMI Architecture and Bluetooth Phonebook Design in Car Infotainment

Archana Bhat

***Abstract*— Mobile devices are today a crucial part in many people's lives. Whether its accessing emails, streaming music or managing contacts - such scenarios are now expected to be fully accessible and functional in automotive environments. The automotive industry is a steadily growing market for Bluetooth technology, with Bluetooth enabled hands-free calling systems now included as standard equipment on millions of new cars and trucks. Safety concerns and new hands-free driving laws spurred the explosion in hands-free calling systems. Any driver doesn't want to fiddle with his phone while driving, so you need some way to get the names and numbers of your contacts from the phone into your car's infotainment system where they can be safely browsed or used in voice commands. Phone Book Access Profile (PBAP) gives your car's hands-free system access to your phone's address book for syncing. To access the phone book, HMI (Human Machine Interface) is used as the car display in the head unit. This paper explains about the architecture of HMI used in car Infotainment and downloading the phone book stored in the mobile devices to the car radio through Bluetooth (BT) connection. It briefly explains the flow of information in the downloading process from the BT system side to HMI.**

***Index Terms*— BT (Bluetooth), FU (Functional Unit), HMI (Human Machine Interface), PBAP (Phone Book Access Profile).**

I. INTRODUCTION

The automotive industry is involved in the design, development, manufacture, marketing, and selling of motor vehicle. As one of the largest and most diversified Tier 1 supplier of automotive parts, Delphi provides the vehicle manufacturer customers with global, single-point sourcing capability and systems tailored to meet their specific needs. The way drivers interact with their cars is evolving, driven by changes in a number of related domains. Automotive HMIs are influenced by global trends in several key areas like personal mobile devices, consumer electronics, car technology, demography, economy and legislation. Every new generation of Smartphone's becomes more powerful, enabling users to perform increasingly more complex interactions in such areas as entertainment, shopping, social networking or gaming. The consumer technology market which has changed and advanced most in the past decade is the mobile phone sector. Thanks to game-changing user experiences delivered through touch screens the market has rapidly evolved, giving rise to tablets, phablets and the like. Along with advances in chip, display and web based technologies; mobile devices have become more and more affordable, achieving unprecedented mass adoption.

Archana Bhat, Information Science and Engineering Department, National Institute of Engineering, Mysore, India. Currently working as summer Intern in Delphi Automotive Systems, Technical Center India (TCI), Bangalore, India.

It seems only natural that they have found their way into the cars we drive today, replacing physical buttons and knobs with pixels. In process automation industry operator interface is usually a Human Machine Interface Unit (HMI). HMI plays a significant role in creating a friendly visual environment between the user and the technology. It is considered to be the window to the automation control system. Controlling through finger touch has replaced the use of hammers and manual switches enormously. HMI is one of the features present in car radio which plays vital role apart from Audio. HMI is the only visible thing while the rest of the system simply resides behind or below the HMI almost unseen or unrecognized by the user. HMI is scattered across all modules of the radio such as Media, Tuner, Diagnostics, Bluetooth, etc.

Since drivers want to use their Smartphone's also while driving, automotive HMI has to allow seamless integration of various brought-in personal devices based on various software and hardware standards. It should enable safe completion of non-automotive tasks that users usually perform on their mobile devices, such as accessing the applications running in the mobile devices, being able to switch between various media sources, placing a call from radio and many more. One such task is accessing the phonebook in our mobile device from head unit through BT.

Bluetooth is a collection of dozens of profiles, which are basically communication modules that define how a particular feature operates or how the paired devices communicate with each other. One of the profiles is Phone Book Access Profile (PBAP). The PBAP defines the protocols and procedures that shall be used by devices for the retrieval of phone book objects. It is based on a Client-Server interaction model where the Client device pulls phone book objects from the Server device.

II. HMI ARCHITECTURE IN CAR INFOTAINMENT

With the need for safety, connectivity and advances in technology the automotive industry is moving towards the radios with bigger and brighter displays which unlike the traditional radios with a monochrome display that we generally fit in our car. These could be Touch-screen or hard key press operated or a combination of both.
In-vehicle Infotainment is flourishing which provide audio/visual entertainment, as well as automotive navigation systems with several modules coming together to deliver the user with almost anything he might want to access when he is in car like the traditional Tuner sources - FM, AM, XM, DAB, Pandora,Media Sources like CD/DVD, USB Audio, Picture, Video, SD Card support, iPod, AUX, Pair a phone and listen to Music and access the Phone book and make or accept calls,

configure the Radio as you wish, HVAC controls and so much more like Navigation and Smart Device Link(SDL).

In a nutshell, For all these to be able to be seen, experienced and appreciated by the end user, HMI (Human Machine Interface) plays the vital role apart from audio. We use a tool to design the screens/views of which some are static and some others are dynamic and some more are a mix of both.The user inputs directly on the screen (In Touch-screen devices) and the input is recognized by the HMI and passed on to the below layers with an appropriate ID for each event. This event is then handled by the corresponding FU (Functional Layer) and it formats it as needed by the System layer (where the actual processing of the data is done) before passing further.

Once the System is done with the processing as needed, it would publish the appropriate result and the same is received by the middle layer (FU) which again decodes and aligns it as expected by the HMI layer. The HMI displays the updated data to the user. The HMI and FU communicate using a display protocol. Fig 1 is the block diagram of the same.

The HMI development tool has editor, database and engine. The editor is used to create the entire HMI layout and HMI logic in an XML database. It is also used to specify the functional interfaces to the applications that are part of the system. It is built upon a concept called Database Driven Human Machine Interface. It is also used to specify the functional interfaces to the applications that are part of the system. These applications are called Functional Units. An HMI is created and verified in editor without having to write any software. The HMI is stored in the database and downloaded to the target environment in a binary format for improved efficiency. The HMI database contains all of the HMI logic and appearance. Engine executes the HMI in run-time and communicates with the applications using the Open Display Interface protocol.

The software needed for supporting this protocol can be automatically generated for the applications from the Editor. HMI Database (DDH) configures the behavior of the Engine for example what to display from which FU (Functional Unit), which actions to take on user input what to do on incoming events, the graphic design of screens and how to traverse between screens [4].

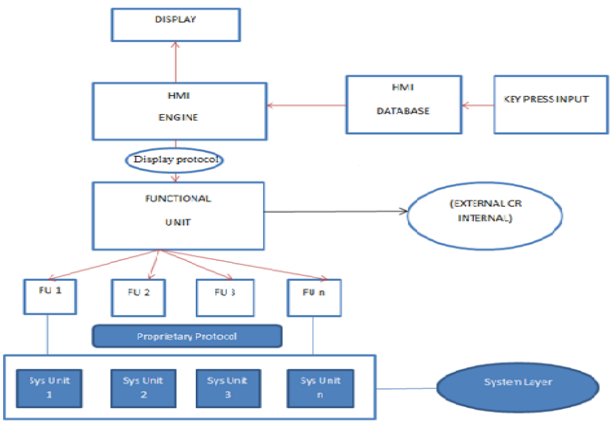

Fig 1: Block diagram of HMI architecture

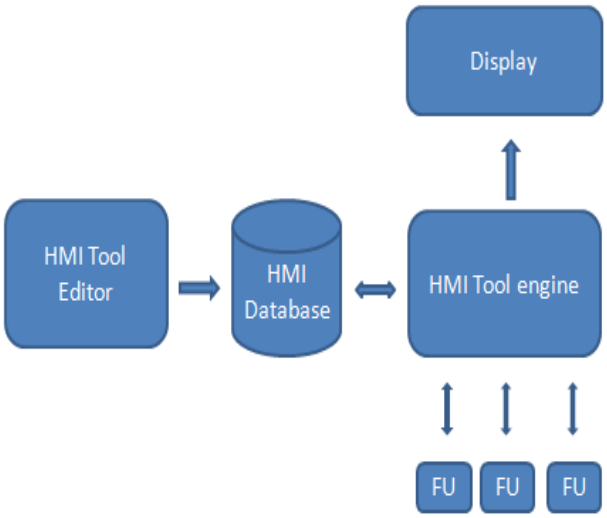

Fig 2: HMI development tool architecture

III. BLUETOOTH PHONE HMI

One of the several modules is Bluetooth used for communication. The radio provides a HMI with which users perform Bluetooth connectivity tasks, such as pair a mobile device with car head unit (HU), dial a number and make a call from HU itself rather than using mobiles. The user can select devices button and pair up to five devices. When the user selects contacts button a page transition is done to contacts list page where the mobile contacts can be seen in the radio. Similarly when the user selects call list button the call history from mobile is seen in the radio. The user can browse the call list and contacts list and place an outgoing call from the radio.

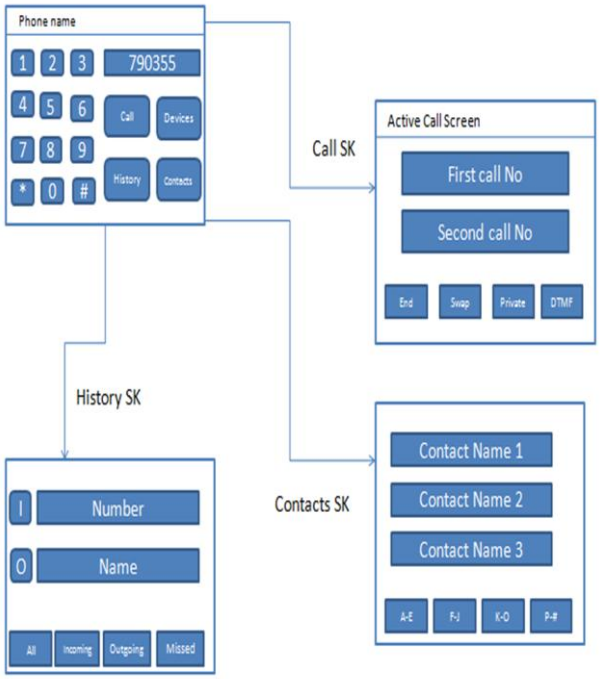

Fig 3: Example diagram of Bluetooth phone HMI

IV. BLUETOOTH SYSTEM ARCHITECTURE

The wide range of possible Bluetooth applications means that there are many Bluetooth software layers. The lower layers (Radio Baseband, Link Controller, and Link Manager) are very similar to the over-air transmissions. They can provide voice connections and a single data pipe between two Bluetooth devices. There are two types of stack: Lower layer stack and upper layer stack. Lower layer stack is the BT controller which consists of radio, baseband and Link controller. This will be residing on the BT chip which is given by CSR or Broadcom companies. The upper layer stack has protocol layers composed of protocols such as Logical Link Control and Adaptation Protocol (L2CAP), Radio Frequency COMM port(RFCOMM) and Service Discovery Protocol(SDP).Profile Layers consists of various profiles which are basically communication modules that define how a particular feature operates or how the paired devices communicate with each other like Hands free profile(HFP),Phone Book Access Profile(PBAP), Advanced Audio Distribution Profile (A2DP), Message Access Profile(MAP),etc. The lower layer stack is the BT stack.

It contains all necessary Bluetooth profiles and a majority of applications needed to deliver full Bluetooth functionality. Communication through the lower and upper layer stack takes place by Host Controller Interface (HCI). The Host Controller Interface is not a software layer, but a transport and communications protocol that aids interoperability between different manufacturer's solutions. It can be Universal Serial Bus [USB], RS232, or a simple Universal Asynchronous Receive Transmit [UART].

BT blocks are various modules like BT connection manager which deals with connect disconnect operations, Phonebook storage manager and Phonebook database which deals with operations related to phone book access and storage and BT gateway is for communication between the BT upper layer stack and HMI. The BT blocks communicate with HMI and the system side layers through an abstraction layer (AL).This AL facilitates the implementation of operating system-independent applications, thus improving their portability. It also provides support for paradigms such as communicating processes/state machines and publish/subscribe. In this approach, the system is composed of independent, concurrent components (hereafter referred to as applications or processes), containing one or more threads which communicate by means of message passing. HMI adapter is the Functional Unit which communicates with HMI and BT blocks. It mainly deals with how the HMI should look and respond according to the events received by the BT blocks and other lower layers.

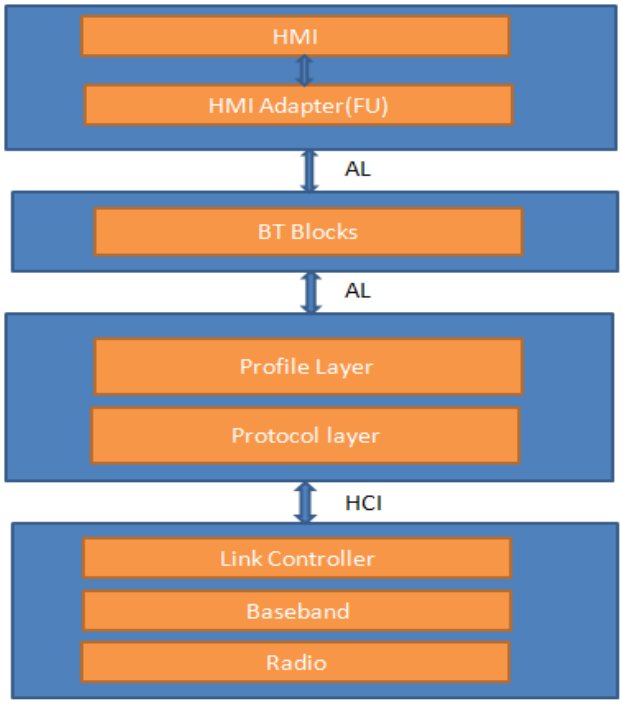

Fig4: Bluetooth system architecture

V. PHONE BOOK ACCESS PROFILE (PBAP)

The Phone Book Access Profile (PBAP) defines the protocols and procedures that shall be used by devices for the retrieval of phone book objects. It is based on a Client-Server interaction model where the Client device pulls phone book objects from the Server device. This profile is especially tailored for the Hands-Free usage case (i.e., implemented in combination with the "Hands-Free Profile" or the "SIM Access Profile"). It provides numerous capabilities that allow for advanced handling of phone book objects, as needed in the car environment. However that this profile only allows for the consultation of phone book objects (read-only). It is not possible to alter the content of the original phone book object (read/write)[3].

The following roles are defined for this profile:
Phone Book Server Equipment (PSE) – This is the device that contains the source phone book objects.
Phone Book Client Equipment (PCE) – This is the device that retrieves phone book objects from the Server Equipment.

For the Hands-Free use case, a typical configuration would be a mobile phone as PSE and a Hands-Free car kit as PCE.

A. User Requirements and Scenarios

The following are some of the main scenarios that are covered by this profile:

- The PCE to access the list of phone book entries stored in the PSE
- The PCE to download one or several phone book entries from the PSE
- The PCE to access the call histories stored in the PSE

B. Profile Fundamentals

The Phone Book Client Equipment may be able to use the services of the Phone Book Server Equipment only after a successful creation of a secure connection. Before a Phone Book Client Equipment may use the services of Phone Book Server Equipment for the first time, the two devices shall bond. Initialization includes exchanging security initialization messages, creation of link keys, encryption, and service discovery. Either the PSE or PCE may initiate bonding. As a minimum the PSE shall support Inquiry in order to initiate bonding. Both PSE and PCE shall support Inquiry Scan Mode in order to accept bonding.

Phone Book Download Feature:

This feature is used to download the entire content of a phone book object. It is especially designed for devices with relatively large storage capacities that are typically attempting to download and locally store the entire phone book. This feature is very basic.

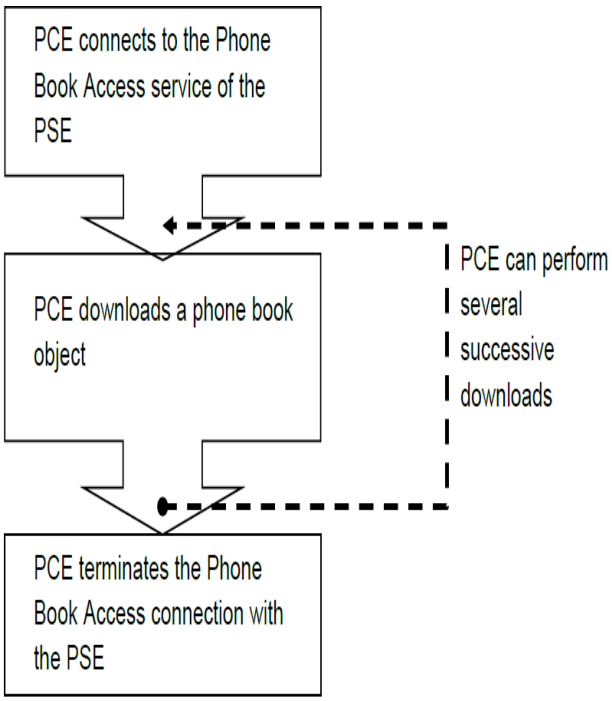

Fig 5: Phone Book Download Feature

VI. Phone Book Download Process

Initially HMI will start the pairing process by placing the radio in discoverable mode. It will send a pairing request to the BT stack. The stack will send the information to BT chip residing on the car radio processor through UART. This information is passed over the air in 2.4 GHz frequency band which is received by the antenna of the phone tuned for this frequency. The BT chip in the phone decodes and starts the pairing process from its side. This event is sent to the stack which in turn generates pins for authentication. It can be legacy pairing or Simple secured pairing (SSM). Once both master and slave devices generate pins and after successful authentication the connection will be established between them.

After successful connection, the master and the slave device will exchange the information regarding the profiles which each other support. The basic profile supported is HFP profile. Fig 6 explains the PB download process. The BT stack receives the HFP connection event from the phone which is broadcasted to all the modules. On reception of this event the phonebook storage manager module will send PBAP_Connect event to connect PBAP services to radio. Once the PBAP profile is connected, the PB storage manager will send Phonebook Download request to the stack to download phonebook with parameters as number of contacts to be downloaded, offset and from which store to be downloaded is sent as HCI command message passed through the UART to the BT chip on the processor of car radio. The store can be either local PB, SIM PB, All call lists, Incoming, outgoing, missed call list. This is received by the Antenna of the phone and then processed by the BT chip in the phone. In the reverse direction, after the phone has processed the command it will send HCI event message which is received by the stack. The stack in turn sends Phonebook Download Complete Event to the PB storage manager. On reception of this the storage manager gets the vcard one by one, parses it and stores it in PB database. After parsing and storing contacts from first store there is a download request from the next store sent by PB storage manager module to BT stack. Once the contacts are ready the PB storage manager will ask whether all the contacts are downloaded to PB database. If the response event is yes, then the PB storage manager will send PBAP_disconnect event to stack to disconnect PBAP profile. This is done specifically for PBAP profile which gives the advantage of less power consumption. Accordingly, HMI screen is updated. It enables the contacts and call list button in the Phone display screen so that the user can browse it. Figures below shows the sequence diagram of how the events are passes from the stack layer to the HMI.

Figure 7 shows the sequence diagram of a use case where the phone is connected to the radio and there is an incoming call. When there is an incoming call, call event is received by the stack, this event is passed to the HMI which shows the incoming call popup. After the call is accepted or ended the corresponding call list has to be updated.

When there is a call end event the HMI gets updated, the same event is sent to PB storage manager which in turn connects the PBAP profiles which was previously disconnected. The storage manager requests the stack for download of the incoming call list and all call list. After it receives the call list download event from the stack, it parses each vcard and stores it in the database. After completion of the successful storage of all the call lists the PB database will update the HMI saying new contact list is updated. The HMI will get the updated list and display it.

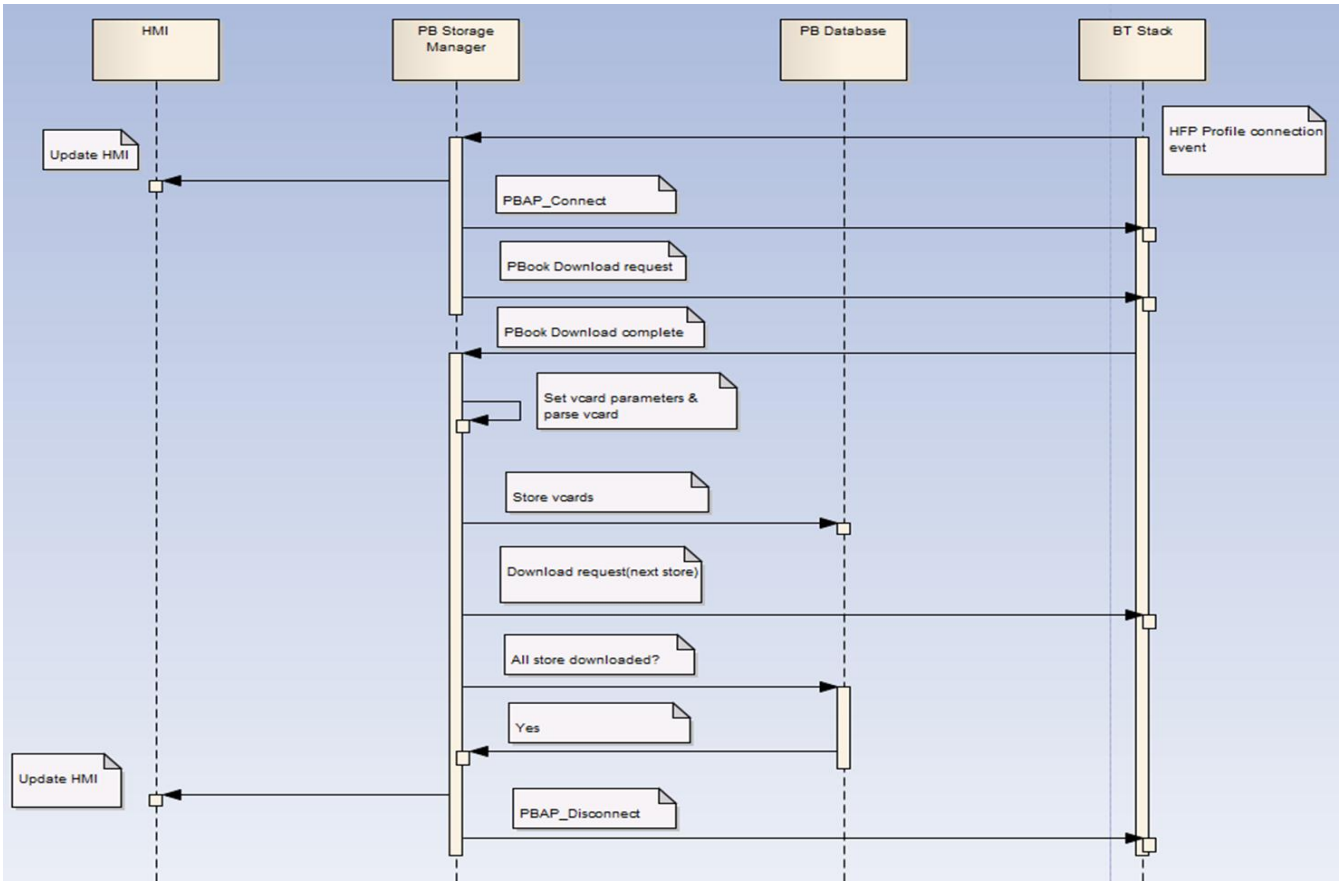

Fig 6: PB download process

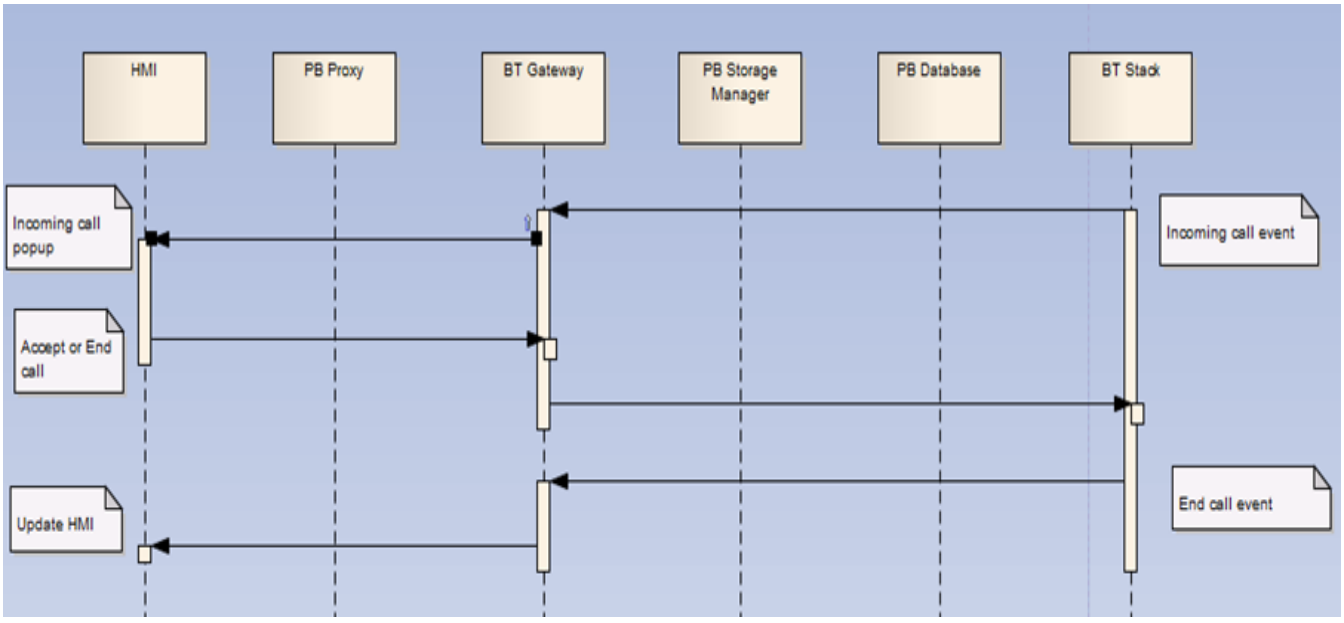

Fig 7: Incoming call use case

VII. CONCLUSION

The proposed system is implemented in one of the advanced car radios. HMI plays a very important role in infotainment design since it is the foreground of the system and the rest lies behind it being invisible to the user. The proposed phonebook download implementation gives the user advantages of ease of control while driving, easy way of placing calls, ease of access to contact and call list while driving.

ACKNOWLEDGMENTS

The successful completion of any task would be incomplete without the mention of people who made it possible. I would

like to thank Delphi for giving me the opportunity to publish this paper.

I express my sincere gratitude to Mr. Krishna Prasad, Director of Delphi TCI and Krishnaswamy Jambunathan, Chief Engineer, Infotainment, Delphi TCI for providing all the facilities and support for successful completion of this paper.

I sincerely thank Mr. Munusamy, Mr. Milton Crasto, Engineering Managers, Delphi TCI for their valuable guidance, support and encouragement.

I am indebted to Mr. Madan Kumar Reddy, Mr. Kamalakanta Barik, Technical Leaders, Delphi TCI, and the entire Delphi team for their constant supervision and for providing all technical and professional inputs.

I sincerely thank my Internal Guide Mrs. S. Kuzhalvaimozhi, Associate Professor, Department of Information Science & Engineering, National Institute of Engineering, and Mysore for her support and valuable guidance.

REFERENCES

[1] Prof. Akram Hossain,"HMI Design: An Analysis of a Good Display for Seamless Integration between User Understanding and Automatic Controls", Purdue University, Calumet, American Society for Engineering Education, 2012.
[2] Sergio Damiani, Enrica Deregibus & Luisa Andreone, "Driver-vehicle interfaces and interaction: where are they going?" Received: 19 September 2008 /Accepted: 13 April 2009 /Published online: 13 May 2009 # European Conference. Of Transport Research Institutes (ECTRI) 2009.
[3] David Kammer, Gordon McNutt, Brian Senese, Jennifer Bray, "Bluetooth Application Developers Guide".
[4] http://mecel.se/products/mecel-populus
[5] Robin Heydon, "Overview of Profiles and Interoperability", Cambridge Silicon Radio.
[6] Suresh Marisetty, Durgesh Srivastava, Joel Andrew Hoffmann, "An Architecture for In-Vehicle Infotainment Systems", January 29, 2010.
[7] Telephony and Car Working Group "Phone Book Access Profile", Revision V12, 2013-Nov-05.