



Technology Solutions for In-Vehicle Infotainment: A White Paper

Abstract

The principal objective of this document is to elaborate on the creation of customizable In-Vehicle Infotainment solutions.

Executive Summary

The characteristic of the automobile buyer has undergone a rapid change in the past few years. What was once an industry that was dominated by comfort, good steering and fuel efficiency, now has a new variable that influences the buyer. Introducing In-Vehicle Infotainment [or **IVI**]. Entertainment ideas of the past, like a simple DVD player on the dashboard, have now evolved into a complex, almost tablet-like, touch screen-based system with multiple arrays of infotainment options.

In-Vehicle Infotainment is a term that defines the entertainment and informative applications that are made available in automobiles. Display screens in cars allowing access to maps, GPS, media players, electronic dashboards, engine diagnostics, internet and social networking, and integration with handheld devices, are the new generation of infotainment in automobile development today.

The need for In-Vehicle Infotainment has seen a consistent growth of expectation since the dawn of the GPS. Basic navigational options, like maps and GPS, have now spawned a barrage of ideas for infotainment solutions. The current generation of In-Vehicle Infotainment can give you accurate engine diagnostics, in the event of a car breakdown, and can direct you to the nearest service station to report the problem. Things like remote ignition give you the option of warming up the car's engine before you actually get into the car. Furthermore, current ideas eventually morph into more useful solutions, like remote ignition has morphed into anti-theft systems, in which your automobile sends you a message in the event of a break-in and allows you to track your car's movements.

Going by the rate of development in IVI, the future of the automobile industry lies with the intelligent car. It would come as no surprise, in the not-so-distant future, when your car plays a role very similar to a personal assistant [imagine KITT from the *Knight Rider* series]. Fantasy eventually meets reality. A couple of years from now, it would be wrong not to expect your car to give you suggestions about preparing for the weather, send e-mails, schedule meetings, give you reminders, etc.

A budding field in IVI is the electronic dashboard, more commonly known as "Instrument Clusters". The level of detail in LCD speedometers and tachometers has dumbfounded even the analog-purists. Problems of a not-so-smooth display that takes away the "feel" of acceleration have long gone. Digital meters are identical to analog meters when it comes to look-and-feel. The future of Instrument Cluster-integration with handheld devices will give parents of teenagers a sense of security. Imagine being able to regulate the speed of car sitting in your own home, while your kids are out on a drive!

The Birth of In-Vehicle Infotainment Solutions

The first application of Electronic Instrument Clusters started appearing during the 1970's. They included not only a speedometer with a digital readout, but also a trip computer that displayed factors like the outdoor temperature, travel direction, fuel economy and distance-to-empty [DTE].



During the initial stages of the introduction and use of digital meters they were criticized mainly for the following reasons:

- They were hard to see in the strong sunlight or other bright light
- They took away the sense of continuous acceleration experience that is provided by an analog speedometer.
- They were expensive to repair in the event of a malfunction

Manufacturers introduced a combination of analog and digital meters, with an additional option of a trip computer that functioned independent of the speedometer.

The new millennium saw a rise in digital meters being incorporated into Hybrid vehicles, especially by Toyota and Honda. A hybrid version of analog and digital dashboards was installed in cars, and drivers started relying on digital speedometers more than the analog variants.

Vehicle manufacturers like Mercedes and Lexus also introduced dashboards which contained instrument gauges shown on LCD Displays instead of analog meters. The advancements in Instrument Cluster technology had improved viewability, supporting high resolutions in bright sunlight and improved usability at night. Maintenance and failure rate had also dropped over the years with the technology to support it.

There has been substantial research and development in Heads-Up Display [HUD] for the automotive industry. HUDs have been used in fighter aircraft, and similar technology is being used to incorporate it into automobiles.

With high-quality graphics processors [that even supports 3D graphics] emerging in the embedded market, digital Instrument Clusters are the dashboard of the new age.



Graphical representation of a HUD on a car windscreen

Customized Instrument Clusters

Electronic Instrument Clusters, when developed for automobiles, until now have been an “exclusive to a brand of automobile” affair. They have been developed from the ground up for individual client projects, and recreating them based on another client’s requirement would mean starting from scratch.

In any scenario involving new technology, a short time to market can be a deciding factor. The driving idea behind customizable instrument clusters is to tackle this very issue. Client-exclusive design and development of the same IVI features will incur higher investment and higher time to market since the codes used are not necessarily portable and reusable.



A customized Instrument Cluster using the Embitel toolkit

The usage of digital instrument clusters by automotive manufacturers can reduce investment by using a common hardware platform across models, and customizing the instrument clusters based on the model/variant of the car. Additionally, software technology for instrument clusters will prove to be a benefit to the industry by allowing customizations to achieve a look-and-feel that is unique and OEM-specific.

Customization of the IVI also allows the solutions to be extended to multiple parameters in an automobile. The screens can be customized to display the parameters received over a Controller Area Network [CAN]. These parameters can be related to fuel readings, engine diagnostics and temperature, seat belt alarms, and so on.

Design Challenges

The following are some of the challenges faced in development of IVI solutions.

Fast Boot-up

The issue of boot-up time can be a major decision maker when it comes to IVI. The average boot-up time for IVI on MeeGo was approximately 15 seconds. Imagine switching off your engine at a signal and attempting to restart when the signal turns green. An almost-zero delay is preferred. The instrument cluster must be functional in less than one second.

Embedded operating systems' boot-up time has always been a concern. Real-Time Operating Systems [RTOS] are capable of fast boot-up, but a full-fledged Linux OS like MeeGo needs customization to reduce the boot-up time. There are solutions which run a full-fledged Linux OS along with an RTOS which is capable of a fast boot-up. Other solutions which tackle this constraint [that is, reducing the boot-up time to less than a second for MeeGo] are available in the industry.

Efficient CPU Usage

Instrument Clusters should be able to acquire information from the various sensors on the automobile and render the data immediately on the display without any delay.. High-quality graphics require a high amount of CPU usage and processing. Multi-processing is vital in such systems and requires hardware support. Multi-threaded applications require an advanced CPU with multi-core architecture that can handle IVI applications efficiently.

Display Management

Multiple displays in IVI prove to be a better option for displaying information of several types. For example, the front display shows the instrument cluster, while a movie plays on the left-rear display, and music plays on the right-rear display.

The ability to run numerous applications on a central device and handle multiple displays requires lot of processing, and an efficient graphics processor capable of handling high amounts of rendering. Many IVI-customized boards provide outstanding graphics processing and drivers capable of such operations.



An IVI-integrated car representation

Additional IVI Solutions - Customized Applications

Video Player

A video player helps play movies and other video content that are either on a local playlist or stored on a removable device. It is very useful to have a multiple display configuration which permits selection and playing of any video content available locally on the selected display.



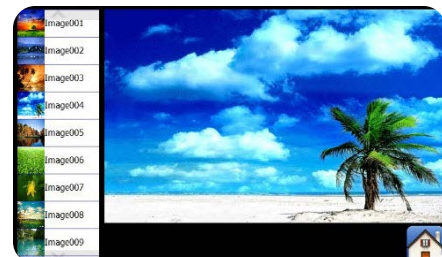
Audio Player



A music player helps you localize and select music using the display panel. It allows the user to organize playlists by genre, artist, etc.

Image Viewer

An Image Viewer allows you to view images/photos on full screen. The user can choose images from a list of thumbnails.



Note: All the above apps have been created using Embitel's IVI toolkit which runs on the MeeGo IVI platform.

Design Considerations in IVI

IVI solutions, still being in a nascent phase, face considerable drawbacks in mass production.

Listed below are some of the key design considerations.

Time to Market

Every automotive manufacturer's concern in developing an instrument cluster is to be able to provide a solution that matches the specifications of the car, and meets the expectation of the driver.

Car manufacturers that have migrated to the use of digital displays can benefit from the technology innovations available in both IVI hardware and software. These solutions facilitate the customization of instrument clusters for various models of cars without causing delays in the designing of customized dashboards.

Development Costs

Developing a customized solution from scratch, involves a lot of research and development. Additionally, a suitable hardware and software platform needs to be chosen and validated before the solution is implemented. The initial lead time is normally very high. The investment in development is considerably high owing to the fact that resources and time is invested in the creation of base components. Using existing solutions that can be reused and customized with minimal efforts is a method to reduce costs.

Reusability

While developing custom solutions, reusability is given very less consideration. If base components are reusable, then porting it to new designs, platforms and subsequent revisions become possible.

Reusability was not considered as a key aspect in software design primarily because hardware and midlevel software platforms did not follow any specific standards, and varied mostly due to technological differences and cost factors. Aiming to make all base components reusable reduces cost and time-to-market of solutions for OEMs.

Extensibility

The high rate of advancements in technology makes it important to develop solutions that can be extensible. The advancements/upgrades, while adding value, must be capable of being incorporated into existing products and solutions. Constraints in hardware/software may not always allow this to be possible. Creating an entirely new system just to add minor technological improvements is not a workable solution. Solutions must always be extensible for improvement and must be able to include new features and technologies.

Configurability

Digital displays provide a good intuitive user interface by providing touch-based control. With the advancements in IVI, the amount of information available [instrument clusters, engine diagnostics, fuel calculators, etc.] is a lot more than what a single screen can display. Allowing the manufacturer or the driver to customize these displays adds very high value to the solution.

Our Development Approach

Embitel's R&D team, after in-depth research into IVI and its best practices, has come up with valid pointers to benefit the development of IVI-based solutions.

Develop a Reusable Solution

With Embitel's solutions, reusable doesn't mean "recycling". All solutions are individually crafted to each client's requirement. What it does mean is one solution can be reapplied in a number of chosen hardware platforms.

Consider this: When an IVI application like a media player is created to run on the display of an automotive manufacturer, the chances of the same application being compatible with another brand of display hardware are very slim.

For the above mentioned scenario to become possible, the solution should have some kind of a common ground, based on which the application will be able to run on both automotive manufacturers' systems. A universal solution to this predicament is what the GENIVI Alliance addresses. The GENIVI initiative has created a neutral platform for IVI applications to operate.

Embitel looks to raise the bar a little higher by offering reusability in the Human-Machine Interface [HMI]. Taking the example of instrument clusters, the advantages of allowing customizability to any parameter at any given time is a useful enhancement to the IVI domain.

The advancements of open source software technology enable the creation of re-usable components. This will allow rapid creation and launch of diverse IVI applications, based on specific requirements, and will result in a reduced time to market for automotive manufacturers.

Develop an Extensible Solution

The rate of technological advancements in IVI is so high that solutions that are in use today could be redundant in the coming months. The integration and utilization of new technology is a vital benchmark to be mindful of while developing new solutions.

Consider the example of internet access in a vehicle. The need for entertainment and social networking is fast becoming a necessity with the driver population. A software solution that supports the integration of new technology and concepts is the way to go.

While "extensible" is still the keyword, it would be wrong to assume that software solutions can be made to adapt to any new technology. What it implies is that there is a scope for improvement, and there will be features that allow expansion of the existing systems.

Develop a Configurable Solution

Manipulation of solutions to configure them to a different specification is of utmost importance. If an automotive manufacturer wants to reconfigure their series of instrument clusters to match the specifications of a newer variant of the automobile, the solution needs to be configurable. For example, if a speedometer on an old variant of the automobile has a maximum speed of 140 kilometers-per-hour, and the new variant has a maximum speed of 220 kilometers-per-hour, reconfiguring the solution is going to require a lesser investment of time and money.

A networked car is capable of displaying a large amount of information, and providing customization to the display will be great features to implement in the IVI segment. Software technology permits creating such solutions since data is available at any given point. The solution just needs to be innovative in the way customizability is developed.

Hardware Considerations

The following points are an indicator of the hardware requirements for IVI.

Requirements

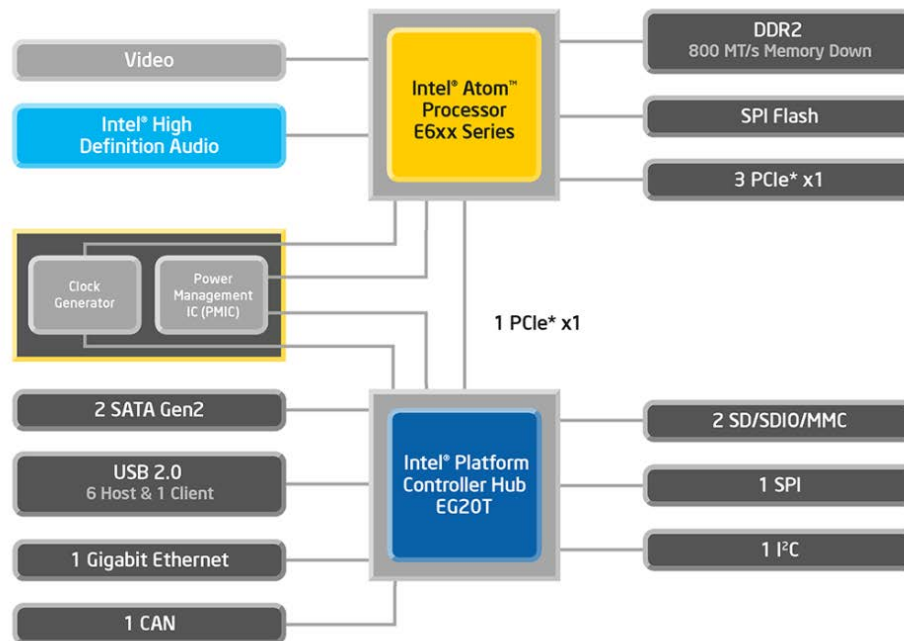
- Processor of at least one gigahertz processing speed
- A dedicated Graphics Controller to support navigation, 2D and 3D displays
- A minimum of two channels of audio support
- Possibility for multiple high resolution Graphics Display Unit [GPU]
- Support for multiple types of media devices [iAP/ USB/ MTP/ SD/ CD-ROM/ CD-DA]
- CAN bus to interface within the vehicle network
- Media-Oriented Systems Transport (MOST) for high speed video/audio signal transfer
- Bluetooth, WiFi, and Ethernet connectivity
- UART, USB connectors for various consumer electronics devices
- Low power consumption
- Should be able to support fast boot-up
- Should be able to match automotive standards for In-Vehicle applications
- The hardware should support at least one of the following operating systems: Linux, Android, Ubuntu, MeeGo, MontaVista

Range of Leading Automotive-Specific Hardware Platforms

- Renesas R-Car H1 (Marzen)
- Freescale i.mx6q Sabre-light
- Freescale i.mx53 Sabre
- Freescale i.mx53 QSB
- Intel Atom Crossville

Why Intel Atom?

Why Embitel decided on the Intel Atom Crossville for IVI solutions.



- Fast boot/start up, high performance, power management and security options built in
- Comprehensive networking and internet support
- Multi-channel multimedia capabilities for audio and video requirements
- Integration for consumer devices via Bluetooth or APIs
- A hardware platform customized for IVI applications
- Support to run multi-threaded applications for high-end HMI

Software Considerations

Keeping in mind the keywords “multi-platform”, “reusable” and “customizable”, our R&D team narrowed down to the best possible software options for the development of the IVI solutions. These options have been incorporated into the project based on their versatility.

Advantages of Open Source Solutions

Open Source software solutions offer significant benefits when compared to typical commercial products. There is more emphasis on quality rather number of features available. There are three major benefits to using open source technology:

1. A reduced dependency on software vendors; and the ease of integration of a wide range of technology variants
2. Lower costs of operation
3. A very large open source developer community to support in fixing issues and enhancing solutions, reducing downtime and time-to-market.

With closed source, or proprietary technology solutions, overcoming simple bugs in the software will require new releases. On the other hand, with open source, the code can be altered and the bug fixed; much like an automobile engineer who can open the bonnet of a car and make necessary fixes.



MeeGo

MeeGo is a Linux-based open source operating system, developed by Intel and Nokia, mainly created to target the mobile and consumer electronic market. It has been chosen by the GENIVI Alliance as the standard operating system for IVI. The adaptability of MeeGo to IVI is why MeeGo has been favored over other operating systems for the Customizable Instrument Clusters project.

The Linux-based MeeGo platform is intended to power multiple types of devices including smartphones, netbooks, tablets, connected TVs, mobile computers and in-car entertainment systems.

MeeGo IVI is a Free and Open-Source Software (FOSS) OS distribution integrating the MeeGo Core for embedded systems, leveraging and extending functionality known from mobile computers and handsets into vehicles.

MeeGo will offer a range of online and computing experiences with rich graphics, multitasking and multimedia capabilities. And not just on devices with Intel chips - it is envisaged as an open platform for "multiple processor architectures".

The User Interface experience [UX] are designed and created for various platforms using QT, GTK+, Clutter etc.

The GENIVI Alliance, a consortium of several car makers and their industry partners, driving the broad adoption of an In-Vehicle Infotainment open-source development platform chose MeeGo as the platform of choice working with the Linux Foundation. This opened the door for MeeGo IVI to become a GENIVI-compliant platform.

The MeeGo IVI 1.2.0 UI is primarily based on a collection of QML based applications designed for the MeeGo Tablet. It has an IVI specific taskbar app launcher that adds support for scroll wheel and voice command inputs.

The IVI Platform provides the necessary framework required to develop and run IVI specific applications.

Tizen

Since the announcement by the Linux Foundation of the redirecting of efforts from MeeGo to Tizen, the IVI industry has been shifting gears to adapt. Tizen is also a Linux-based open source operating system that has been customized for IVI.

Tizen is positioned to capture the MeeGo community and advance to the next level of development. Tizen adds native support for HTML5. Though the support for Qt has not been announced in the base code, it is expected to be ported sometime in the near future. Additionally, Qt was meant to run on any Linux-based operating system.

The Tizen IVI solution will enable modern portable applications provide rich Internet and multimedia experiences to consumers while traveling.

The Tizen project resides within the Linux Foundation and is governed by a Technical Steering Group.

Qt

Qt is an open source application framework that can be used to write advanced applications and UIs. These applications and UIs can be operated across desktop and embedded operating systems. Qt's platform-independent UI framework can be used for HMI creation.

Qt lives up to its reputation among developers of GUI as an advanced toolkit allowing the creation of advanced cross-platform applications. It was initially developed by Trolltech [now taken over by Nokia]. Qt is Free and Open Source Software.

It is written completely in the C++ programming language. It features a Meta-Object Compiler which facilitates macros and enables cross-compilation, supporting many compilers.

The introduction of QML in Qt [Qt Quick] brought about a big change in the way applications were designed and implemented in C++. The designer could work easily using QML and the developer could write complex logic in C++, and also manipulate all elements exposed by QML.

Using Qt, you can write web-enabled applications once, and deploy them across desktop, mobile and embedded operating systems without rewriting the source code. Widgets can be created, extended and customized as plugins, required for various purposes. Since it has a write-once run-everywhere policy, it becomes very useful to create applications that run on different types of hardware platforms, varying from PC to embedded devices, without having to rewrite any piece of the code.

Qt also integrated WebKit which provides a web browser that enables the use of web content in Qt Applications. It also provides a Script Engine which has support to execute JavaScript and interact using C++ code.

The next major release of Qt is version 5 which is expected in August 2012.

The major reasons for us to consider Qt are its ease of use to develop re-usable, extensible and configurable solutions that are essential for IVI. The UX [User Experience] of MeeGo uses Qt and contains all libraries that support and complement application development in Qt.



Middleware Considerations

The two possible options for the middleware platform are the GENIVI middleware and Embitel's custom-built middleware for enhanced performance.

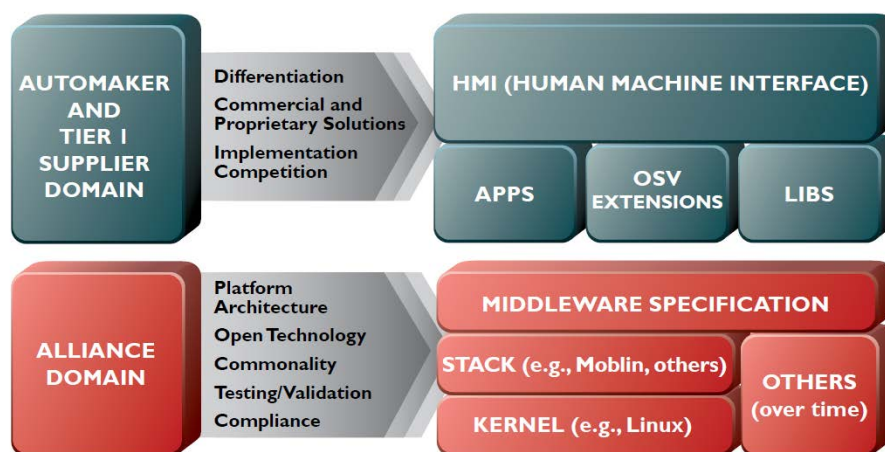
GENIVI Alliance

GENIVI® is a non-profit industry alliance committed to driving the broad adoption of an In-Vehicle Infotainment (IVI) open-source development platform. The alliance aims to align requirements, deliver reference implementations, offer certification programs, and foster a vibrant open-source IVI community.

GENIVI Middleware

- Reusable, open-source platform consisting of Standardized Linux-based core services, middleware, and open application layer interfaces
- Supports the Automotive domain by using the Open Source community to extend the platform
- The GENIVI IVI platform does not address the highly competitive areas such as user interfaces and logic that define the end-user experience
- Only GENIVI members get access to the development activities, updates, and manuals required for GENIVI compliant software development
- GENIVI chose MeeGo as their base OS platform to showcase its capabilities

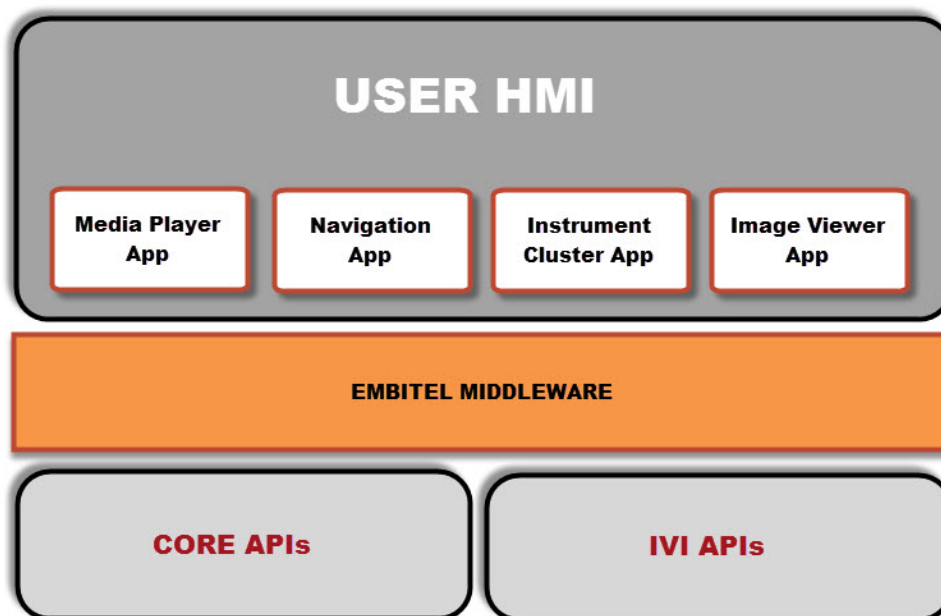
GENIVI Platform Vision



Embitel Middleware

At Embitel, the Core MeeGo APIs are accessed from the application layer. Certain interfaces such as CAN, LIN, MOST, and Embitel's architecture would directly access these interfaces to read and write data on the Bus.

Embitel's middleware consists of API's to access the MeeGo APIs, to access the hardware which directly interacts with hardware interfaces like CAN, LIN or MOST.



In addition, it would provide the mechanism to send or receive information from the HMI layer. This is achieved using the concept of D-Bus.

D-Bus is a system for *interprocess communication* (IPC). D-Bus works by sending messages between processes. Each message has a *header*, including *fields*, and a *body*, including *arguments*. The header is used as the routing information for the message, and the body as the payload. Header fields might include the sender Bus name, destination Bus name, method or signal name, and so forth.

Embitel's Middleware currently supports the exchange of information for the Instrument Cluster module. This layer collects the signal values via CAN APIs and over D-Bus, and interfaces with the HMI display.

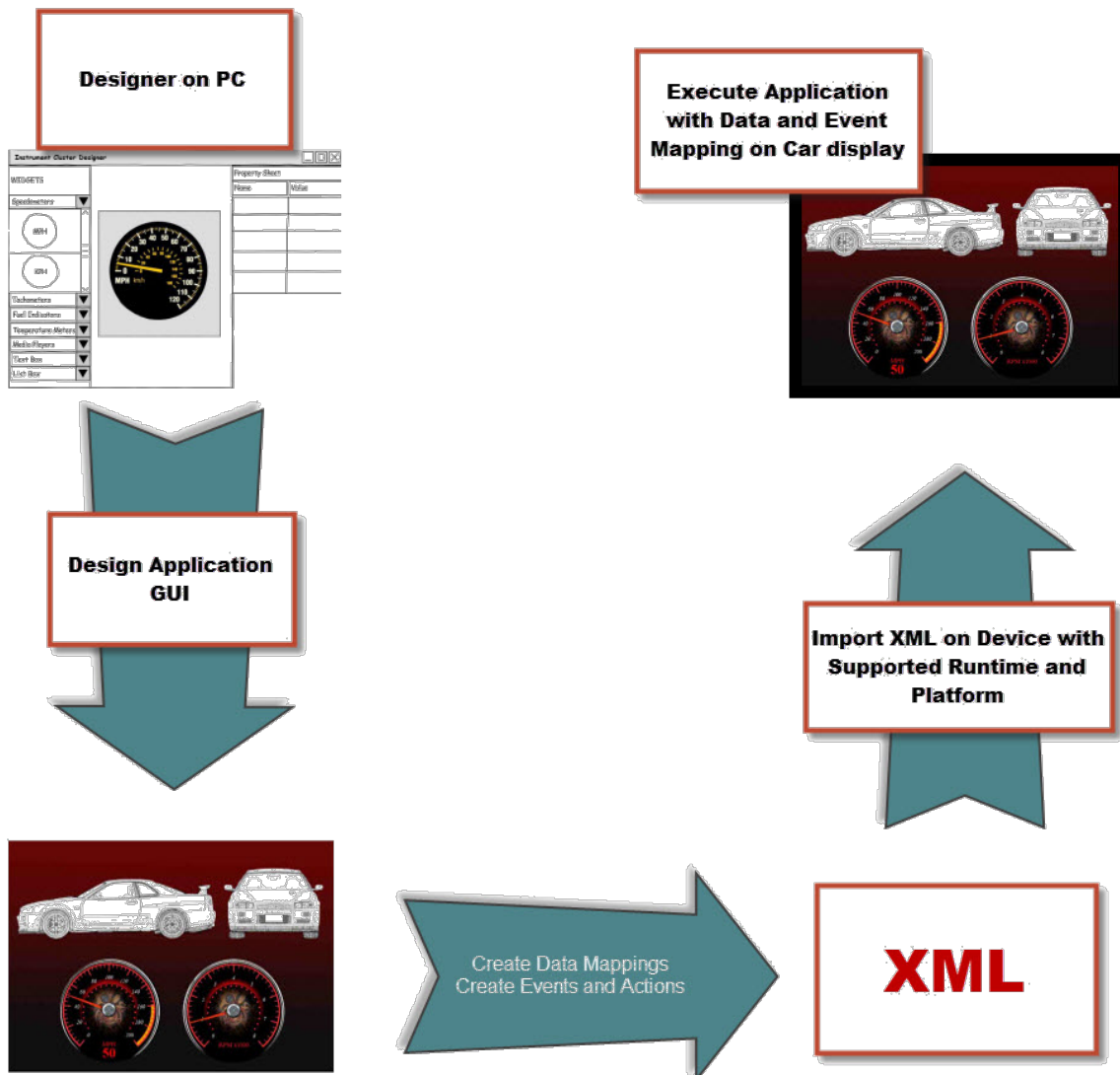
This would be expanded to include Media [supporting Audio/Video playback], GPS Navigation, and image viewer in the near future.

An Integrated Designer Component by Embitel

The designer module is database of customizable widgets. These widgets are developed and modeled on industrial-designed automobile dashboard instruments. This database of widgets is maintained in the form of a GUI toolkit. The designer module is built using Qt and can run on Windows and Linux.

Customizing

The flexibility of the Designer module allows for rapid customization of the Instrument Clusters. The overview of the methodology used is illustrated in the diagram below.



A Brief Overview of Embitel History

Embitel has extensive experience in Embedded Automotive to develop real time complex solutions for our global customers. Our engineers have prior experience of working with end OEMs being involved in all phases of the development lifecycle starting from requirements up to system testing on the target. We leverage on this experience together with good automotive system knowledge to provide end to end product solutions to our clients.

Our team has the expertise to understand complex automotive requirements and undertake projects in the areas of ECU software, AUTOSAR, Diagnostics, in-vehicle networking, Modeling and Simulation, and specialized last mile services.

Embitel's Automotive Embedded History

- Engine Management Functionalities
- Powertrain Systems
- Body Control Modules
- Vehicle Comfort & Safety Functions
- Infotainment and Telematics

Embitel's In-Vehicle Infotainment Systems

- Telematics Systems - Portable Navigation Device, GPS, Dead Reckoning, Internet, WAP, e-mail, SMS, Bluetooth
- Optical and Electrical Vehicle BUS systems (MOST, CAN, and FLEXRAY)
- OpenGL-based and Graphics Device Interface (GDI) to develop 2D or 3D-based graphical HMIs
- Safety: eCall, In-Band Modems for Text-Over-Voice, Embedded and NOMAD solutions
- DSP Codecs – WMA, AAC, WAV
- Strong knowledge base in Automotive Electronics
- Existing Automotive clients include few OEMs and Tier-1 Suppliers in Europe
- Existing clients showing interest in IVI
- HMI customization

Glossary

AAC - Advanced Audio Coding; is a standardized, compression and encoding scheme for digital audio

Actions – sensor outputs or user actions

Android – Linux-based operating system for mobile devices

API – Application Programming Interface; is a specification intended to be used as an interface by software components to communicate with each other.

AUTOSAR - Automotive Open System Architecture; an open and standardized automotive software architecture

Bluetooth - is an open wireless technology standard for exchanging data over short distances

boot-up - is the initial set of operations that a computer system performs when electrical power is switched on

C++ - A general purpose programming language

CAN – [Or CAN bus], Controller Area Network; is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other within a vehicle

CD-DA – Compact Disc Digital Audio, an audio recording format

CD-ROM – Compact Disc Read-Only Memory; a compact disc that contains data

Clutter - An open source graphics library for creating hardware-accelerated user interfaces

Codec - A device or computer program capable of encoding or decoding a digital data stream or signal

Data mapping – the process of creating data element mappings [or functions] between two distinct data models

D-Bus – Desktop Bus; an inter-process communication open-source system for software applications to communicate with one another

Dead Reckoning – Or Deduced Reckoning, is the process of calculating one's current position by using a previously determined position

Distance-to-empty – Estimated distance the vehicle can travel before it runs out of fuel; displayed on the instrument cluster

DSP - Digital Signal Processing; the mathematical manipulation of an information signal to modify or improve it in some way

DVD – Digital Versatile Disc, or Digital Video Disc; an optical storage format

eCall - a European initiative intended to bring rapid assistance to motorists involved in a collision anywhere in the European Union

ECU – Engine Control Unit; a type of electronic control unit that controls a series of actuators on an internal combustion engine to ensure the optimum running

Ethernet - A family of computer networking technologies for local area networks

Events - Software message indicating that something has happened, such as a keystroke or mouse click

FlexRay - is an automotive network communications protocol developed by the FlexRay Consortium to govern on-board automotive computing

FOSS – Free and Open Source Software; a software that is both free software and open source.

GDI - Graphics Device Interface; a Microsoft Windows application programming interface

GENIVI - A non-profit industry alliance committed to driving the broad adoption of an In-Vehicle Infotainment open-source development platform.

GPS – Global Positioning System; is a space-based satellite navigation system that provides location and time information

GPU – Graphics Processing Unit; a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the building of images

GTK+ - GIMP Toolkit; a cross-platform widget toolkit for creating graphical user interfaces

GUI – Graphical User Interface; a type of user interface that allows users to interact with electronic devices using images rather than text commands

handheld devices – A small, mobile computing device

HMI – Human-Machine Interface; the space where interaction between humans and machines occurs

HTML5 - A markup language for structuring and presenting content for the World Wide Web

HUD – Head-Up Display; a transparent display that presents data without requiring users to look away from their usual viewpoints

Hybrid vehicles - A vehicle that uses two or more distinct power sources to move the vehicle

IPC - Inter-Process Communication; a set of methods for the exchange of data among multiple threads in one or more processes

JavaScript - A prototype-based scripting language

Kernel - A bridge between applications and the actual data processing done at the hardware level

LCD – Liquid Crystal Display; a flat panel, electronic visual or video display, which uses the light modulating properties of liquid crystals

LIN – Local Interconnect Network; a serial network protocol used for communication between components in vehicles

MOST – Media Oriented Systems Transport; a high-speed multimedia network technology optimized by the automotive industry

MTP – Media Transfer Protocol; a devised set of custom extensions to the Picture Transfer Protocol

Netbook - a category of small, lightweight, inexpensive computers

OEM - Original Equipment Manufacturer; manufactures products or components that are purchased by another company and retailed under that purchasing company's brand name

OpenGL – Open Graphics Library; a standard specification defining a cross-language, multi-platform API

Plug-ins - A set of software components that adds specific abilities to a larger software application

Porting - The process of adapting software so that an executable program can be created for a computing environment that is different from the one for which it was originally designed

Powertrain - the group of components that generate power and deliver it to the road surface in an automobile

QML – Qt Modeling Language; a JavaScript-based, declarative language for designing user interface-centric applications

Real-Time Operating System - An operating system (OS) intended to serve real-time application requests

remote ignition - An electronic lock that controls access to a vehicle without using a traditional mechanical key

Runtime - A software component designed to support the execution of computer programs

SD – Standard-Definition; a television system that uses a resolution that is not considered to be either enhanced-definition television (EDTV) or high-definition television (HDTV).

Smartphones - A mobile phone built on a mobile computing platform, with advanced computing ability and connectivity

SMS - Short Message Service; a form of text messaging communication on phones and mobile phones

Tablet - A mobile computer, larger than a mobile phone or personal digital assistant, integrated into a flat touch screen and primarily operated by touching the screen

Telematics - Any integrated use of telecommunications and informatics

time-to-market - The length of time it takes from a product being conceived until its being available for sale.

Toolkit – Or Widget Toolkit; a set of widgets for use in designing applications with graphical user interfaces

trip computer - An onboard computer device fitted to cars which can generally record distance travelled, average speed, average fuel consumption, and display real time fuel consumption information

UART - Universal Asynchronous Receiver/Transmitter; a piece of computer hardware that translates data between parallel and serial forms

USB – Universal Serial Bus; an industry standard developed in the mid-1990s that defines the cables, connectors and communications protocols used in a bus for connection, communication and power supply between computers and electronic devices

UX – Or User Experience Design; a broad term used to explain all aspects of a person's experience with the system including the interface, graphics, industrial design, physical interaction, and the manual

WAP – Wireless Application Protocol; a technical standard for accessing information over a mobile wireless network

WAV - Waveform Audio File Format; a Microsoft and IBM audio file format standard for storing an audio bitstream on PCs

WebKit - a layout engine designed to allow web browsers to render web pages

WiFi - a popular technology that allows an electronic device to exchange data wirelessly (using radio waves) over a computer network, including high-speed Internet connections

WMA - Windows Media Audio; an audio data compression technology developed by Microsoft

XML - Extensible Markup Language; a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

References

- www.genivi.org
- <https://meego.com>
- <https://www.tizen.org>
- qt.nokia.com
- http://www.intel.com/p/en_US/embedded/applications/in-vehicle-infotainment/