

# Development of an Interface Prototype for Automotive Infotainment System

Lorena Espinoza- Bermúdez<sup>1</sup>, Alexa G Marmolejo-Alvarado<sup>1</sup>, José Ma. Celaya-Padilla<sup>1</sup>, Huizilopoztli Luna-García<sup>1</sup>, Jorge I Galván-Tejada<sup>1</sup>, Carlos E Galván-Tejada<sup>1</sup>, Hamurabi Gamboa-Rosales<sup>1</sup>, Jose G Arceo-Olague<sup>1</sup>, Arturo Moreno-Báez<sup>1</sup>

<sup>1</sup> Centro de Investigación e Innovación Automotriz de México (CIAM), Universidad Autónoma de Zacatecas, Jardín Juárez 147, Centro Histórico, CP 98000, Zacatecas, México  
{lorebe01, alexagma83}@gmail.com  
{jose.celaya, hlugar, gatejo, ericgalvan, hamurabigr, arceojg, morenob20}@uaz.edu.mx

**Abstract.** The design of infotainment interfaces is increasingly important and necessary for the development of technology in the automotive industry, since it allows the user to have an interaction with the vehicle so that the functions of the vehicle can be effectively controlled and that there is a feedback to the driver, thus allowing him to know if the action taken was correct and how to follow the operation. This work introduces the development of a prototype user interface for automotive systems that allows easy manipulation of its elements as well as active and useful feedback to the user.

**Keywords:** User Interface, Infotainment Systems, EB-GUIDE

## 1 Introduction

The rise of the automotive industry in Mexico has taken place for 25 years [1], today it is a dynamic growth industry within the global framework [2]. This industry has, as a basis for success, several important aspects to consider, for example, opportunity niches for both industry and topical research, including, being a protagonist in Latin America of development of technologies for this industry, reducing the number of cars imported, among others. Thus, the most important challenges are considered to lead to scientific research and the development of new technologies for this industry.

Within automotive research there are a number of challenges to improving automobiles, including, performance, safety, comfort and interaction between the user, passengers and information and entertainment systems or also known as Infotainment Systems [3]. Regarding convenience, the user increasingly requires tools to help him obtain information, control systems and have connectivity to the outside world. Infotainment systems can be described as a combination of systems used to provide entertainment and information to the driver and passengers through

audio/video interfaces, with control elements such as touch screens, buttons, voice commands and among others [4].

Another important aspect to consider in infotainment system design is usability, according to Jakob Nielsen, usability [5] is described as, a quality attribute that evaluates the ease of use of user interfaces. The word "usability" also refers to methods to improve ease of use during the design process. Similarly, interaction is a key aspect to consider in the design process, for example, today's infotainment systems allow interaction through voice commands that allows you to write or receive messages, make calls, access internet content from both social networks and informational content, weather or vehicular traffic conditions, to name a few [6,7].

However, one of the most commonly used functions in these systems is the playback and management of digital media as a form of entertainment (e.g. Music). Derived from this, the project in question is focused on one of the most basic but most used functions of this type of systems which is the handling of audio through an interface that offers enough visual feedback, without being struck for the driver and companions.

In-car audio systems have evolved rapidly and are becoming more user-friendly. As we already know, graphical interfaces are not new to the automotive industry, being an important part of user interaction today. Such is the case of the Volkswagen Jetta, its 2018 model had a 5" screen, while the 2020 model has an 8" screen and also had the removal of the cluster of instruments to incorporate a central screen, thus, aiming for this research to explore and develop more friendly report systems to provide greater comfort and safety in the use of them.

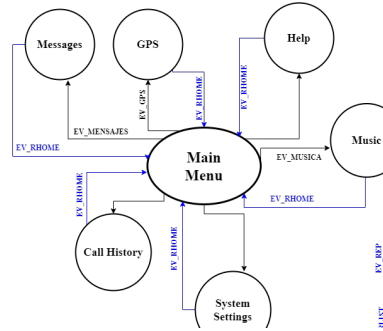
Latest car releases have included very good quality audio players, creating a very comfortable and safe environment for the driver. Some of these features are the buttons included in the steering wheel to have better control of the audio system without having to remove your hands from the steering wheel thus avoiding possible accidents related to lack of attention on the part of the driver.

Because the automotive industry is the largest developer of infotainment systems, there are limited open source options that can be used by the community for research. However, an alliance of several companies is under development, whose purpose is to be able to develop these systems in open source, Linux Automotive [8]. This allows developers, individuals or companies to have the same development platform and in turn, this makes it possible to reduce operating costs and allows the research community to enter these projects.

## 2 Materials and methods

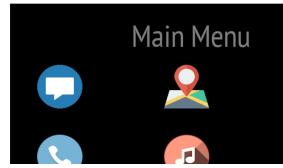
The EB GUIDE software by Elektrobit [9] was used to carry out this prototype (This software was used because the Universidad Autonoma de Zacatecas / CIIAM has a collaboration agreement with the EB-GUIDE company), this software is especially useful for the development of infotainment systems). This platform allows the elaboration, design and modeling of graphical interfaces of vehicle screens, among other applications. Elektrobit is a visionary global provider of integrated and connected software products and services for the automotive industry, which has worked with automotive companies such as Audi, BMW, Ford, Volkswagen Group among others. It works through a software framework that allows the creation of solutions in an augmented reality, all in real time, in addition in its "community" version it is possible to use it freely for non-profit [10]. This software works using state machines, where each state is an action performed by the software. The development of this prototype was done in EB guide version 6.7.1.

On the home screen, a menu was implemented showing different task options that the driver can perform. The options to be displayed were thought of to satisfy the main needs that usually occur when the user is driving and because of this the driver cannot pay much attention to his cell phone thus facilitating the option to solve the task without getting distracted from the steering wheel [11]. Fig. 1, outlines a schematic showing the functionality of the prototype.



**Fig. 1.** Schema of the program performed.

The designed menu contains 6 icons with different tasks that the user can select while driving. All these options appear at the same time on the main screen thinking of avoiding any kind of distraction that may occur. In the menu you will find the messaging icons, call history, GPS, system settings, system help and music as shown in Fig. 2. We selected these functions and icons because they are the most common in current infotainment systems, it is not the purpose of this article to delve into this aspect.



**Fig 2.** Main menu designed on the EB GUIDE platform.

## 2.1 Program code

This program contains pre-set functionalities that are subsequently inherited, changed and/or manipulated to achieve the desired functionality. This type of object-oriented approach significantly reduced the number of lines of code required for programming. In EB GUIDE objects are inserted into templates.

The general properties that display all templates are described in Table 1.

**Table 1.** Description of the general properties manipulated in templates.

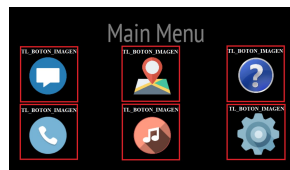
Property	Description	
height	Height of the object.	
image	An image that is assigned to the object.	
pivotX	Reference point on the X axis either to scale or rotate the object.	
pivotY	Reference point on the Y axis either to scale or rotate the object.	
scaleMode	original size	Scales the image assigned to the object so that it maintains its original size.
	fit to size	Scales the image assigned to the object so that it fits a certain size, regardless of whether it is distorted or not.
	keep aspect ratio	Scales the image assigned to the object so that it fits a certain size but without distorting the appearance of the object.
scalingX	Scales the object on the X axis from other properties of the object, such as those that are code-based.	
scalingY	Scales the object on the Y axis from other properties of the object, such as those that are code-based.	

touchPressed	Dictates what happens while you have a tactile stimulus on the object. It is controlled by lines of code, from which you can control other properties of the object.
touchShortReleased	Dictates what happens when a touch stimulus is stopped on the object. It is controlled by lines of code, from which you can control other properties of the object.
width	Width of the object.
x	The X position where the object is located.
y	The Y position where the object is located.

---

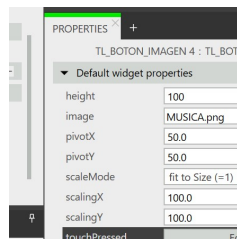
Each template contains particular properties that describe corresponding actions.

**Template *TL\_BOTON\_IMAGEN*.** This template was used in most interface development because it was developed to function as a button where, when selected, you could perform a default action. On the home screen, 6 templates of this type were placed with different images (Fig. 3), where each image is the representative icon of the window that will open if you select the button.



**Fig. 3.** Main menu with 6 templates.

This Template has the following properties (see Fig. 4):



**Fig. 4.** EB GUIDE *TL\_BOTON\_IMAGEN* Properties.

As described in Table 1, the properties of this template can be modified as many times as necessary in each of the copies generated, its general properties are seen in Fig. 4.

To visually know that a button was selected, a resolution change of the selected button was programmed to allow a visual resizing. The programming was edited in the *touchPressed* property and later in the *TouchShortReleased*. The scaling function property is used specifically to scale the size of the image and visually make the "click" that the user orders, thus achieving the resizing only while selecting the button.

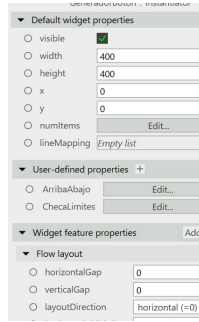
This is how the TL\_BOTON\_IMAGEN template was created. In the development of the interface was indispensable due to it was used in most of the development as it was to place the 6 icons of the main screen, the return button appears in each window, the option to move up or down the playlists, a button to play a song randomly and in the window where it simulates the playback of a song the template appears on the play icon, pause, advance song and turn the volume up or down. In all these cases, the template was used, changing only the image type and size of it.

**Template *TL\_ListaM*.** The "TL\_ListaM" template was used in two different interface windows. The performed function was to list the different types of data. The first window where it was implemented was to display playlist options and the second to display the songs contained in that list (see Fig. 5).



**Fig. 5.** TL\_ListaM implemented.

This container adds the size, element, and location (x,y) properties to the global interface so that they can be manipulated on any copies that are generated from the template. In addition, an event was inserted which allows access to another window if selected any item in the list. The particular properties used in the development of this template are seen in Fig. 6.



**Fig. 6.** EB GUIDE "Button Generator" properties.

This instance contains some general properties described in Table 1. The particular properties are described in Table 2:

**Table. 2.** Description of particular properties of the template TL\_ListaM.

	Event	Description
<b>User-defined properties</b>	ArribaAbajo	Allows sliding up or down the list as the case may be.
	ChecaLimites	Check the list limit. After the limit is complete, a slippage of the list is no longer presented.
<b>Flow layout</b>	layoutDirection	Positions the address of the list. The design has a horizontal direction.

For the ArribaAbajo event to be functional, the following code was created:

```
function(v:arg0::bool)
{
    match_event v:event = ev:Abajo in
    {
        v:this.yOffset -=30
    }
    match_event v:event = ev:Arriba in
    {
        v:this.yOffset +=30
    }
}
```

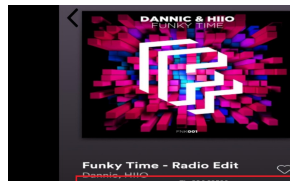
Meanwhile the event "ChecaLimites" code is:

```

function(v:arg0::bool)
{
    let v:MaxDesplazamiento = length v:this->^.Elen
v:this->^.height in
{
    if(v:this.yOffset>0)
        v:this.yOffset = 0
    if (v:this.yOffset< -v:MaxDesplazamiento)
        v:this.yOffset = -v:MaxDesplazamiento
}
// todo: auto generated return value. please adapt

```

**Template *TL\_PROGRESO*.** The development of this template was to be able to simulate the playback of a song, where it actually is an image that changes its horizontal size within a certain time, said template was only used in the playback window (see Fig. 7).



**Fig. 7.** Template “TL\_PROGRESO”.

A container was created where two conditional<sup>1</sup> properties named "ACTUALIZA\_ANCHO" and "ANCHO\_PAPA" were applied. These properties allow to modify the width of the container using conditions by simulating a run or playback of a song. At the same height of this container appears a label called "PERCENTAGE", this label contains an event called "AVANCE" and is the event that modifies its size compared to the characteristics of the "ANCHO\_PAPA" property. To make the job functional it was necessary to insert the following code:

```

function(v:arg0::bool)
{
    let v:mi_ancho=f:nearbyint((f:int2float(v:this->^->PORCEN
100.0)*f:int2float(v:this.ANCHO_PAPA)) in
{
    v:this.width = v:mi_ancho
    // Para revisar límites
    if (v:this->^->PORCENTAJE.AVANCE <0)
        v:this.width = 0
    if (v:this->^->PORCENTAJE.AVANCE >100)
        v:this.width = v:this.ANCHO_PAPA
    v:this->^->PORCENTAJE.text = f:format_!
true,f:int2float( v:this->^->PORCENTAJE.AVANCE)) + "%"
    // todo: auto generated return value. please adapt

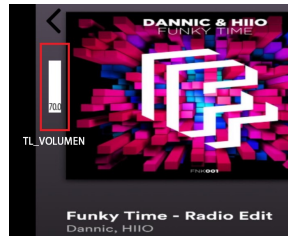
```

---

<sup>1</sup> A conditional script is a property that applies only if it carries a condition, that condition is described by code.



**Template *TL\_VOLUMEN*.** It works in a way that when you select the volume up or down button within the playback window, an image appears pointing in the percentage of the volume at which the playing song is being listened to. This template was developed much like the *TL\_PROGRESO* template, its internal architecture (hierarchy) is the same, the difference between these both is the bar, in this template it resizes vertically and is initially an invisible image that becomes visible only when the up or down button is selected. The code used was similar to the above described, the difference being the variables that are called because they are different.



**Fig. 8.** *TL\_VOLUMEN*.

Fig. 8 shows an image where you can see the template *TL\_VOLUMEN* which is the vertical white bar in the upper corner of the same figure.

### 3 Results and conclusions

This article briefly describes the process that was carried out for the development of an interface for an Infotainment System using the EB GUIDE software of the company Elektrobit. In the main window were used six templates called *TL\_BOTON\_IMAGEN* (Fig. 9 part 1), as buttons for the different options in the menu. The background images and triggering events were modified, as described in Fig. 1.

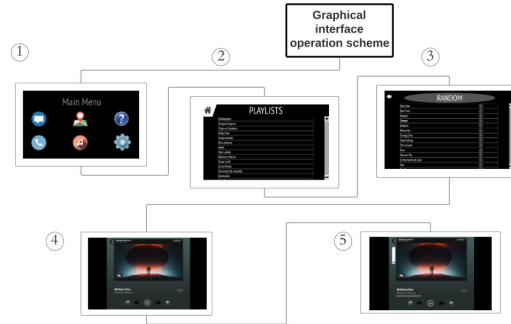
Design and development sequences are displayed specifically for the music icon. When you select the music button, an event triggers the next window which is a playlist, for this to happen, the template *TL\_ListaM* was used, besides adding two *TL\_BOTON\_IMAGEN* templates to be able to move up or down the list (Fig. 9 part 2). Choosing the playlist links directly to the list of songs where, like the previous case, the *TL\_ListaM* template and two *TL\_BOTON\_IMAGEN* templates were used to be able to see the list completely.

By moving up or down the list (Fig. 9 part 3) and choosing the song, there is a template *TL\_PROGRESO* on the play button and thanks to this template, on (Fig. 9 part 4) the simulation of the playback of said song can be seen. Besides this template, an additional one that shows the percentage of volume during music playback

TL\_VOLUMEN (Fig. 9 Part 5) and this also contains buttons to turn the volume up or down TL\_BOTON\_IMAGEN.

Icons with the image of an arrow or cottage appear in each window. It is actually a TL\_BOTON\_IMAGEN template that allows you to return to the previous window until you reach the main menu.

It is important to mention that this is a work in progress and at the moment, the progress of the prototype development is not available in any repository.



**Fig. 9** Schema and representation of the state machine, designed for the infotainment system.

As can be seen in Fig.3, it was possible to obtain an interface that is ideally user-friendly and allows an interaction with the various info-entertainment systems of the vehicle which allows the possibility of being used as a basis to develop a complete interface and thus evaluate its usability so that, in the future, it can determine the advantages or disadvantages that may have and thus have the possibility to develop such graphical interfaces and be able to apply it to vehicles.

## 4 Future work

The use of cell phones while driving it is a leading cause of accidents in Mexico [12] it is thought that the design of an info-entertainment interface could help to decrease the use of the cell phone and thus contribute to the decrease of accidents. On the other hand, the use of the interface itself was not a distraction to the driver, a minimalist design concept was taken which will subsequently undergo a usability assessment, aiming to achieve the standards used in the automotive industry.

Because of the obvious generational breach that exists today [13] it is easier for users under the age of 30 to make use of technology in any form, in the development of the designed info-entertainment interface digital immigrants could be taken into account. By studying and evaluating the usability of the interface, you

could get feedback as to what needs to be changed so that anyone, regardless of their age or skill in technology, can make correct and effective use of it.

The prototype design is still in the development stage, since at the end of the interface it is intended to undergo an evaluation to determine if it meets the appropriate standards of a car and to know if it is suitable for use

## References

1. Unger, K. The automotive industry: Technological change and sourcing from Mexico. *North American Review of Economics and Finance*, 2(2), (1991) 109-128.
2. Rodríguez, . Retos para el sector automotriz hacia 2020. marzo 01, 2020, de Alto Nivel Sitio web: [https://www.altonivel.com.mx/empresas/negocios/46748-retos-y-oportunidades-para-el-sector-automotriz/\(2012\)](https://www.altonivel.com.mx/empresas/negocios/46748-retos-y-oportunidades-para-el-sector-automotriz/(2012))
3. Powers, W. F., & Nicastrì, P. R. Automotive vehicle control challenges in the 21st century. *Control engineering practice*, 8(6), (2000) 605-618.
4. Saxena, A. Everything You Need to Know About In-Vehicle Infotainment Systems. marzo 01, 2020, de eInfochips Sitio web: <http://www.einfochips.com/blog/everything-you-need-to-know-about-in-vehicle-infotainment-system> (2018)
5. Nielsen, J. Usability 101: Introduction to Usability. Nielsen Norman Group. Retrieved 26 July 2020, from <https://www.nngroup.com/articles/usability-101-introduction-to-usability/> (2012)
6. Noso, K., Kishi, N., & Futami, T. U.S. Patent No. 4,558,459. Washington, DC: U.S. Patent and Trademark Office. (1985)
7. Hassenzahl, M., Laschke, M., Eckoldt, K., Lenz, E., & Schumann, J. “It’s More Fun to Commute”—An Example of Using Automotive Interaction Design to Promote Well-Being in Cars. In *Automotive User Interfaces* (pp. 95-120). Springer, Cham .(2017)
8. Automotive Grade Linux. The Linux Foundation Projects. Retrieved 23 July 2020, from <https://www.automotivelinux.org/>. (2016)
9. EB Elektrobit EB Guide Studio. [Online]. Available: <https://www.elektrobit.com/> (accedido el 2 de Abril, 2020). (2020, Abril)
10. *About (EB)*. (2020). Elektrobit. <https://www.elektrobit.com/about/>
11. Garzon, J. (2014). Android Auto: la propuesta de Google para llevar Android a los carros. Retrieved 3 September 2020, from <https://www.cnet.com/es/analisis/google-android-auto-primer-vistazo/>
12. Alavez, M. Celular, el rey de los accidentes de auto. *El Sol de México*. (2019)
13. Prensky, M. Digital natives, digital immigrants. *On the horizon*, 9(5).(2001)