Capgemini

Test design for Infotainment

Increasing Test Coverage

Table of Contents

1.Introduction

Once considered novelties, In-Vehicle Infotainment (IVI) systems now have become de facto gadget in modern cars. It is no longer limited to high end vehicles. IVI has now become an important feature that influences the buyer.

OEMs depend largely on suppliers for developing and testing the IVI systems. Absence of appropriate and sophisticated test platforms and lack of best fit test designs pose challenges. In most cases, issues and defects are found when the IVI devices are tested in the vehicle, which dents the reputation of the brand. The software or the hardware must be modified and retested. The entire process disturbs the production schedule on calendar.

As the complexity of IVI systems continue to increase over the next few years, so will the importance of “Effective Testing” and faster turn-around time to fix issues.

2.Testing Technique in IVI Domain – The Need

Automotive Infotainment system is growing rapidly because of advancement in technology. More and more new technologies and devices are getting integrated in the system.

The automakers are now focusing on test solutions, which can mainly address the following:

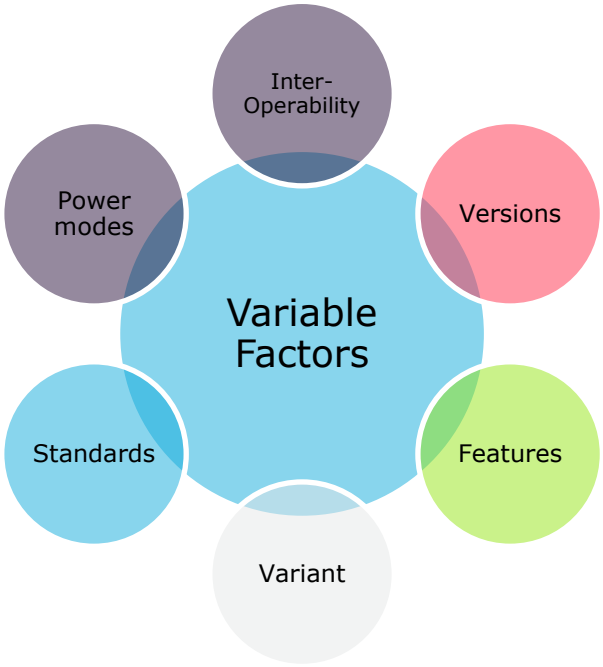
- Testing of latest and upcoming IVI devices and their seamless connectivity with other devices such as; smartphones, tablets, laptops and headsets
- Simulation and testing of real time scenarios; such as for GPS Navigation and Audio System
- Simulation and testing of safety scenarios to ensure minimum driver distraction
- Testing of IVI applications with other applications and devices, which share common multimedia devices, such as; the touch screen display is now more commonly used for Navigation, DVD video, Games, Internet, Rear View camera, Park Assist systems and Blind Spot Detection systems
- Security of personal data with proper authentication, authorization and encryption
- Testing of driver and passenger IVI systems with all possible scenarios
- Capability to test the system under different phases of the development cycle, before field and production tests

Keeping all the solutions in mind the most important role to be played is the Test designing and techniques used to cover all corners of the code and other variables for the IVI System.

3.Challenges in creating IVI test cases

The growing complexity of a typical IVI system introduces a lot of variable factors to be considered while planning Test cases. These variable factors plays are very critical as the IVI system provides the Human Machine interaction and hence having a user point of view becomes the key in finding and resolving issues. Below is the list of factors:

- Domains
- Features
- Variants
- Versions
- Standards
- Device Interoperability
- Power modes



All the above variables need to be covered along with our test designs and techniques for increasing the **"Test Effectiveness"** and **"Test Coverage"**

4.IVI Test Designing Techniques

Considering various factors and variables around IVI system, a robust test design technique is of utmost importance. Proposing "Matrix based approach" for the same. However, unlike traditional matrix based approach which is a common practice in Testing domain, here each Matrix to be designed considering the IVI testing aspect. Traditional way of writing test cases may not address the interdependency criteria and hence this approach is more effective and reliable.

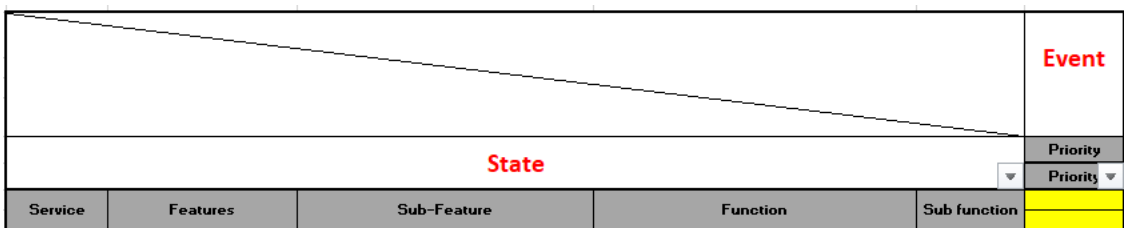
- Active Matrix: Feature Vs Feature
- Structure Matrix: Feature Vs Test Object
- Variant Matrix : Feature Vs Variant
- Cumulative Matrix : Continuous Operations
- Trouble Matrix : Negative Scenarios
- Start up Matrix : Feature Vs Start up conditions
- Device Matrix : Feature Vs Devices

4.1. Active Matrix Design

Active Matrix design is a unique test designing concept where the matrix is formed and arranged on “Feature Vs Feature” and “Domain Vs Domain”, keeping the entire SUT (system under test - Head Unit) “active” all the time.

This kind of test designing gives reliability and robustness to the system.

For any active system (many number of processes running) if a user invokes or provides any input to the system how would a system behave? The answer is in the Active matrix tests.



The diagram shows the concept used for this test design:

State: This is the current state of the System or the test steps which the user needs to follow

Event: The new test step which the user will introduce

Service: These are the services in the IVI System viz: Audio service, mode service, system service etc.

Features: These points to the IVI features viz Carplay, dial in, browsing, E-call etc.

Sub Function/Function: Another level of Features where an action is performed or the setting of current state of system.

Since this Active matrix would result in huge number of test cases to be generated, for ease of execution we can device a mechanism of priority assignment and grouping.

Both the State and event are assigned appropriate priority and in the common cell a combined priority is mentioned for eg.

State					Event	Faceplate and CCE Hardkey			
						CCE Back	CCE Clear	CCE Favorite HK	CCE Display off
					Priority	B	B	A	A
Service	Features	Sub-Feature	Function	Sub function					
	CD-MP3	Mute			A	AB	AB	AA	AA
		Unmute			B	BB	BB	BA	BA
		Volume Up			D	DB	DB	DA	DA
		Volume Down			D	DB	DB	DA	DA
		Seek			B	BB	BB	BA	BA
		Fast Seek			B	AB	AB	AA	AA
		Search	Current Track List		A	AB	AB	AA	AA
			Folder		C	CB	CB	CA	CA
			Select By cover		B	AB	AB	AA	AA
			Keyword	History	B	BB	BB	BA	BA
				For :	B	BB	BB	BA	BA
			Artist		C	CB	CB	CA	CA
			Albums		C	CB	CB	CA	CA
			Genre		C	CB	CB	CA	CA
			Track		C	CB	CB	CA	CA
			Year		D	DB	DB	DA	DA
			Composers		C	CB	CB	CA	CA
			Photos	Picture list	C	CB	CB	CA	CA
				Start slideshow	C	CB	CB	CA	CA
				Turn clockwise	C	CB	CB	CA	CA
				Turn anticlockwis	C	CB	CB	CA	CA
				Close picture viev	C	CB	CB	CA	CA
		Playback mode	Normal track sequence		B	BB	BB	BA	BA
			Random track list		B	BB	BB	BA	BA
			Random media		B	BB	BB	BA	BA
		Skip to time	CCE scroll clockwise		C	CB	CB	CA	CA
			CCE scroll anticlockwise		C	CB	CB	CA	CA
		Show track information			C	CB	CB	CA	CA
		TA			B	AB	AB	AA	AA
		Track number			D	DB	DB	DA	DA

For execution Planning the priorities like AA, AB, AC are given preference or grouped has high priority tests.

With experience it is seen that results of random testing are very good and hence the Tester actively starts thinking of the scenarios. But the challenge with random testing is that it is non-measurable and cannot be planned as required hence this approach of “Formal Random Testing” gives great results in terms of planning and execution. Thereby minimizing manual errors or skipping or missing some scenarios.

4.2. Structure Matrix Design

Tests which checks that HU (System Under Test) operates properly even if the objects are deleted or replaced, and the attributes of the objects is changed, from the perspective of the relationship between the test object and the other objects in the same environment.

This Matrix is designed considering **current state or event** of the Head Unit Versus **Test objects** to be changed in the same testing environment.

Test Objects	Sources											
	Radio	Navi	Tel	media								
				CD			USB			SD		
				Insert	Eject	Playing	Insert	Eject	Playing	Insert	Eject	Playing
Audio												
Mute												
Unmute												
RotaryVolume knob												
Networking												
Phone Connection HFP												

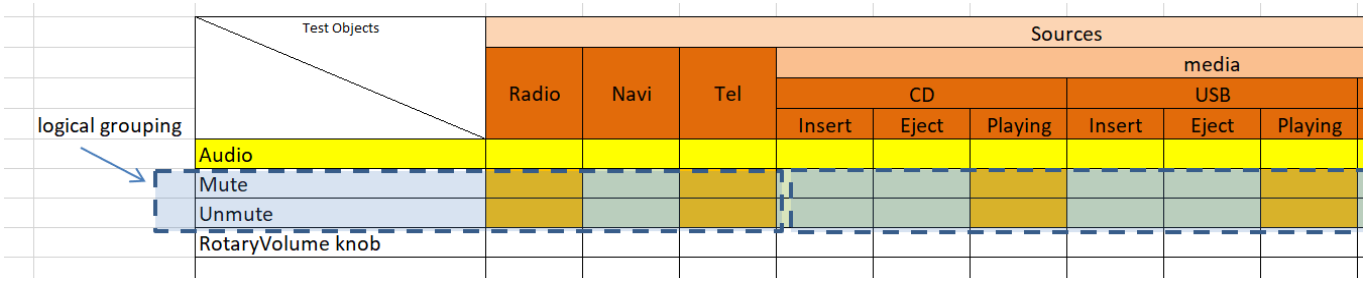
The above diagram shows the concept used for this test design below is an Example:

Test Objects: All the test objects for a test environment are listed in the x-axis of the Matrix viz. Hardkeys,Test media (CD,USB,SD etc.) insert, eject or play.

Features: Features are listed domain wise viz. Audio, Networking, Navigation etc.

This type of test design provides robustness to the overall structure of the functioning of the System Under Test.

Since this type of test design would generate large number test cases so a concept of **logical grouping** is introduced keeping in mind a set of test cases which can be executed at one go.



4.3. Feature Variant Matrix

These are the tests which checks the features of all variants in a program are intact and working as expected.

For a multi variant environment the software packaging plays a very important role. These different software packages might have changes in only a few components of the package with respect to variant and feature whereas keeping most of the package components common across these variants.

To Ensure those changed components functioning for a variant we can design this type of test approach named Feature-Variant Matrix.

Examples:

- NA (North America) region supports TA, SIRIUS, SXM, Regional FM etc
- EU (Europe) region supports DAB
- Baidu CarLife in China
- Language support for several other variants

As part of Integration testing, a separate test suite (Matrix) to be prepared where all feature listing is done as per variant.

		Variant 1	Variant 2	Variant 3	Variant 4	Variant 5	Variant 6
Serial -	**Variant / Feature**	**NA**	**EU**	**AU**	**ME**	**SAM**	**With Rear Entertainement NA**
1	AM	x	x	x	x	x	x
2	FM	x	x	x	x	x	x
3	FOTA	x	-	x	-	-	-
4	XM	x	-	-	-	-	x
5	Pandora	x	x	x	x	x	x
6	Stitcher	x	x	x	x	x	x
7	CD	x	x	x	x	x	x
8	DVD	-	-	-	-	-	x
9	USB	x	x	x	x	x	x
10	AUX	x	x	x	x	x	x
11	Bluetooth Audio	x	x	x	x	x	x
12	Energy	-	-	-	-	-	-
13	Navigation	x	x	x	x	x	x
14	Bluetooth	x	x	x	x	x	x
15	Climate	x	x	x	x	x	x
16	Tone	x	x	x	x	x	x

The above Matrix is prepared from feature summary documents and revised whenever a new requirement or feature is introduced.

This is the best suited technique for multi variant programs.

4.4. Cumulative Operations Matrix

Tests which checks that HU operates properly even if requested processing is performed repeatedly for a long time.

This testing technique involves validating that the software behaviour and response time, is consistent while performing a unique set of test in a cumulative manner.

Purpose: The main objective of Cumulative Operation Test is to ensure that the features and functionalities are performing as per the requirements and the response time is consistent, every time the test cases are executed.

CUMULATIVE Test			Build Info									
Domain	Test Scenarios		No of Iterations									
			1	2	3	4	5	6	7	8	9	10
Networking	Devices Connect Disconnect	Android OS										
		iPhone										
		USB while importing contacts										
		SD while importing contacts										
		Sim card from SAP module										
	Features	Auto Paging										
		HFP Call										
		Swap PP & SP										
		Import contacts to HU										
		Receive message one by one										

The diagram above shows a sample matrix where the Domains and operations are listed in rows and in columns we have listed number of times the operation is performed.

Sample Test Scenario: Verifying the response time of CD slot (of ECU) multiple times, to see the consistency in the behavior.

To perform this activity following steps are involved:

- Insert the CD in the CD slot and play the same.
- Note the response time, which CD player takes to play the music file.
- Repeat the above 2 steps multiple times (as to be mentioned in matrix) and note the different readings to check the consistency in the behavior.

During test planning one can plan as to how many times an operation needs to be performed.

This kind of tests improves the reliability and stability of the system and at the same time being close to real scenario.

4.5. Corresponding Trouble Matrix

Tests which checks whether it can respond to the obstacle which may be generated during the active running of the test environment.

In this type of matrix approach, we have divided the troubles into two different categories

- One, where a trouble can be a generic trouble pertaining to external environment conditions or generic in nature viz, power reset etc. Could be termed as "**Common Troubles**"
- Second, where the trouble could be caused in a specific domain or the action being performed viz, a scratched CD, corrupted USB. Could be termed as "**Domain Specific Troubles**"

Domains\Troubles	Scenarios	Common Troubles					
		User Data reset	HARD Reset	Battery Cycle	Sudden Voltage Drop*	Sudden Voltage Increase*	Voltage I
Networking Domain							
	TEL BASE Screen						
	Incoming CALL						
	OUTGoing CALL						
	CALL WAITING						
	CALL ACTIVE						
	Phone Book loading						
	Contact Import						
	User Data export						
	BT Pairing						
	Device Manager Screen						

The above diagram shows the design in terms of domains listed in rows and Common troubles listed in columns.

A mentioned scenario is performed by the tester and then the trouble is introduced.

Domains\Troubles	Troubles	Result	Observation
Networking Domain Scenarios	Make CALL with No SIM in device		
	Open Phone Book ,For a Device with No contacts		
	PhoneBook is downloading and Link Loss Occurs		
	Trying to Connect to a device which is "OUT of Range"		
	MAKE CALL to Invalid Number from PHBooK		
	MAKE CALL to Invalid Number ,Entered by HU HK		
	Device Switched OFF when ,establishing BT Connection		
	Play Media with a device Connected for HFP Profile only		
Media Domain Scenarios	Play BTA Music with No Media files in device		
	BTA Media Playing and Link Loss occurs		
	Play Media from Scratched CD		
	Play Media through , non GOOGLE Media player		
	PLAY Voice recorded File		
	PLAY unsupported Media format		
	PLAY Corrupted Track from ,CD,iPOD,USB,MTP		
	Verify Corrupted Track Metadata,Composer,Cover ART, Year, Artist ,Album		

In Domain Specific troubles all the troubles listed in rows and executed by the tester.

This type of test design approach improves reliability of the system in adverse conditions and yet being very close to real user scenarios thereby giving more confidence on the system.

4.6. Start up Matrix

Tests which checks operating correctly by the unstable state immediately after a system startup.

Domain	Scenarios	Pre-condition	Steps	Test ID	Startup Conditions				
					Cold Start (Wait till fan goes off)	Warm Start	User Off/On	Vehicle Reset	User Hard Reset
Audio	Factory settings validation	- Radio Startup	- Play any media device. - Change the Bass , Midtones ,Treble volume - Change Fader and Balance values - Perform Startup conditions - Verify that previously changed value are retained. Note: For User and Vehicle Reset , default values will be displayed i.e 0						
	Increase/Decrease volume	- Radio startup - Media Audio is playing (FM,USB,CD)	- Increase the volume with Rotary knob/MFL Key - Perform the Startup condition - During Startup , the display goes off and volume is not audible - After startup , verify that audio with increased volume is audible						

In this type of test approach all the scenarios are listed in rows (Features) and all kind of start up conditions viz. Cold Start, Hot Start, User on/off etc. are listed in the columns.

During Execution of such matrix a scenario is performed and then start up conditions are given in order to check instability in the System Under Test.

4.7. Device Feature Matrix

Tests which are performed for checking the interoperability of the 'N' number of devices, which are used with the IVI system

Typically, IVI systems of this generation and future generations largely depends on the secondary devices and their connectivity with the System. All these secondary devices come with different OS, filesystems and versions of OS differing with the make and model keeping this in mind providing support of the features with devices being used widely becomes important.

This testing technique involves validating the support of different kind of device and there model available in different markets. For e.g.: Asia market has different demand w.r.t Phone models, USB models SD Card brands, and many more devices.

Purpose: The main objective of Device Matrix Testing is to ensure the compatibility of the IVI System to support different brands and models of different devices.

Test Scenario: Verifying the compatibility of the Media Player with USB stick of different capacity and vendor's w.r.t. different song formats (e.g. m4a,mp3,wav, .aac, wma, etc.) using below mentioned device matrix:

ASIA Market	USB 3.0					USB 2.0				
	1 GB	2 GB	4 GB	16 GB	32 GB	1 GB	2 GB	4 GB	16 GB	32 GB
Transcend	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
SanDisk	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Sony	✘	✘	✘	✘	✘	✔	✔	✔	✔	✔
HP	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

5.Benefits of Matrix based Approach

- A System Under Test can be modeled with Features, Domains, Variants, Versions and restrictions.
- Matrix based test design lets you improve Quality and reduce efforts
- Ensures Test coverage at all interaction levels
- Provides Systematic test planning
- Running more test in lesser time
- Easy to review

6.Conclusion

Below is the summary of the Test design Techniques discussed in this document.

Test Design	Challenges Addressed	Description	Technique
Active Matrix	Features, Domains	Active Matrix is formed and arranged on Feature Vs Feature and Domain Vs Domain keeping the entire system under test (Head Unit) Active all the time.	Feature Vs Feature
Structure Matrix	Features, Test Objects	Test which checks that HU operates properly even if the objects are deleted or replaced, and the attributes of the objects is changed.	Feature Vs Test Objects
Variant Matrix	Variant, Features, Versions	These are the tests which checks the features of all variants in a program are intact and working as expected.	Feature Vs Variant
Cumulative Matrix	Features	Test which checks that HU operates properly even if requested processing is performed repeatedly for a long time.	Feature Vs Continuous Operations
Trouble Matrix	Features	Test which checks whether it can respond to the obstacle which may be generated	Features Vs Troubles
Start up Matrix	Power modes	Test which checks operating correctly by the unstable state immediately after a system startup.	Feature Vs Start up conditions
Device Matrix	Inter operability	Tests which are performed for checking the interoperability of the n number of devices which are used with the IVI system	Feature Vs Devices

Details of each few test designs are mentioned in the attached sample documents:

Active Matrix Design	**Active Test Matrix**
Structure Matrix Design	**Structure Matrix Test Design**
Cumulative Operations Matrix	**Cumulative Operations Matrix**
Feature Variant Matrix	**Feature Variant Matrix**
Corresponding Trouble Matrix	**Corresponding Trouble Tests**
Start up Matrix	**Start up Matrix**

About Capgemini

With more than 190,000 people, Capgemini is present in over 40 countries and celebrates its 50th Anniversary year in 2017. A global leader in consulting, technology and outsourcing services, the Group reported 2016 global revenues of EUR 12.5 billion. Together with its clients, Capgemini creates and delivers business, technology and digital solutions that fit their needs, enabling them to achieve innovation and competitiveness. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Learn more about us at www.Capgemini.com.

People matter, results count.