INTERACTION & HMI DESIGN · VOICE AND CONVERSATIONAL UX

I design what the car says back.

More than eight years defining interaction logic and voice behaviour for safety-critical vehicle systems at **JLR**, **Maserati** and **Iveco**. I write the specifications engineers build from, settle the signal and safety constraints with systems engineering, and review what comes back before it ships.

MULTI-USER VOICE | CONVERSATIONAL & LLM ASSISTANTS | INSTRUMENT CLUSTERS | REQUIREMENTS & SPEC QA | DRIVER DISTRACTION COMPLIANCE

EXPERIENCE	PROGRAMMES	LANGUAGES	EDUCATION
8+ years automotive HMI	JLR · Maserati · Iveco	ES, IT, EN. Coordinating FR, DE, KO, JA	PhD, University of Bergamo

01 Voice and conversational

Four problems that only appear once a voice system has to share a cabin with more than one person, more than one feature, and more than one vehicle architecture.

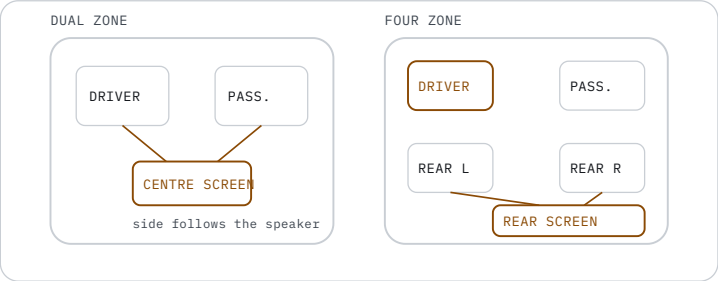
Two people speak at once. Who wins, and where do they see it?

Premium cabins are split into zones, two or four, each with its own voice and touch controls. When two occupants speak at the same time, the system has to have a defined behaviour or it fails unpredictably. There was no pattern to copy, so the model had to be built from first principles.

I defined the interaction priority logic: which signals establish who is speaking and from where, what the system does when that is ambiguous, and how it tells both occupants what it decided without confusing either of them.

Priority was only half of it. Where the answer appears has to follow cabin layout, not the priority rule:

- **Dual zone.** Feedback surfaces on the central screen, left or right according to who issued the command.
- **Four zone.** Rear occupants get feedback on the rear screen; the driver gets it on the driver display. Never on a screen the person who asked is not looking at.
- One production programme used an ambient light animation on the driver display as a lightweight confirmation that the command had registered, without taking the screen.



Feedback routing, abstracted. The rule is not “show it on the main screen”, it is “show it where the person who asked is already looking”.

OUTCOME

A reusable model for multi-user voice conflict: who has priority, and where the response is rendered. Applied across dual-zone and four-zone configurations.

MY ROLE

Defined priority logic, ambiguity fallback and feedback placement rules.

HARD PART

No prior art. Spatial context and speaker identity had to be inferred from available signals.

TRANSFERS TO

Any assistant with several concurrent users: shared devices, meeting rooms, multi-agent handoff.

How much screen a spoken command deserves

The native assistant covers media, phone, navigation, vehicle controls and climate. Treating those the same produces either constant interruption or silent uncertainty. I set the feedback strategy per command class, driven by one variable: how much the system already knows.

- **Climate.** Seat heating, ventilation, steering wheel heat and similar land in the status area already on screen. The climate screen never opens. No interruption at all.
- **Media, weather, navigation, calling a contact.** When a command matches more than one result, a carousel pick list offers the candidates. Same disambiguation principle as the multi-user case, applied to search results instead of speakers.
- **Ambient lighting.** “Change the ambient lighting” shows colour options. “Change the ambient lighting to red” executes with no screen at all, because nothing was ambiguous.

Two patterns follow from the same rule. **Call a contact:** the matches are surfaced, the driver picks, the call is placed; the system asks rather than guessing. **Send a message:** the transcription is shown back for confirmation or edit, and nothing is ever sent without that step.

Intervention scales with uncertainty, not with feature. The interface earns its interruption.

WHY IT MATTERS BEYOND THE CAR

Precision-scaled friction: ask only when genuinely uncertain, act immediately when not. The same principle governs whether an LLM assistant confirms, clarifies or simply does the thing.

MY ROLE

Defined per-class feedback strategy, pick-list template behaviour and confirmation rules.

HARD PART

Resisting a uniform pattern. Consistency here would have cost either attention or trust.

TRANSFERS TO

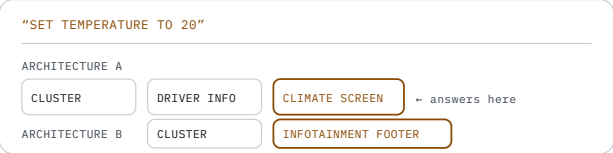
Assistant confirmation design, tool-call confirmation, voice commerce.

ALEXA CAR CONTROL

One command set, two screen architectures

Alexa controls climate only, so every specification starts by establishing what is actually voice-controllable. Distraction rules then push the answer toward audio, with visual feedback only where it is necessary.

The complication is hardware. Jaguar vehicles have a cluster, a driver information area and a dedicated climate screen. Range Rover and others have no equivalent climate screen at all. For every command, from max air conditioning to temperature sync and defrost, the specification had to say which surface answers, for driver and passenger, per vehicle.



OUTCOME

One voice command set mapped correctly onto different hardware, without the driver ever sensing the architecture underneath.

VOICE AS AN APP

Settings that teach, and examples that must actually work

Voice is also a settings surface, where the driver configures behaviour and finds out what is possible.

- **Things to try.** Curated example commands by category, from "find a coffee shop near me" to "set the driver temperature to 20 degrees".
- **Settings with stateful copy.** Custom wake word, default wake word, follow-up requests and AI voice each need their own on and off messaging, so the wording itself tells the driver which state they are in.

The requirement I am testing now: every example shown has to work, in every supported language. A discoverability feature that teaches a command which then fails is worse than none.

OUTCOME

A settings surface that teaches the system rather than only toggling it, and a verification requirement that keeps it honest across languages.

02 Multimodal HMI systems

Where the interesting work is not the screen but the state model behind it: different signals, different failure modes, one mental model for the driver.

Three mechanisms that had to feel like one feature

On estate-body vehicles the tailgate area splits into two menus, seat memory and “trunk and flaps”. Trunk, front trunk and charge flaps each run on different signals and each fail differently. To the driver they have to read as one system.

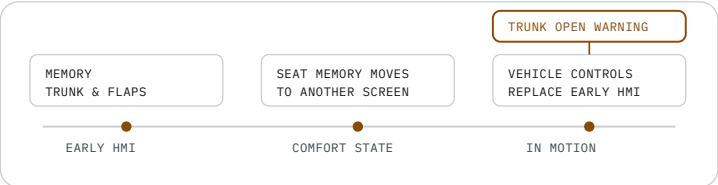
TRUNK The brief asked for a single button to open and close. The available signals could not reliably distinguish the two, so I specified two dedicated buttons. Clarity kept, within what the vehicle could actually detect.

FRONT TRUNK Opens from the system, but always closes by hand. The interaction has to make that asymmetry obvious instead of implying it behaves like the trunk.

SAFETY If the vehicle moves with the trunk open, the cluster warns and the relevant control is surfaced with the warning, so the driver closes it from there instead of hunting through a menu.

FLAPS Two flaps, one per side, with a hard rule that only one may be open at a time. Every valid state had to be legible and consistent with how trunk and front trunk already behaved.

VEHICLE STATE Seat memory exists only in early HMI, before the vehicle reaches its comfort state, then moves to another screen entirely. Trunk and flaps is early-HMI only too, and is replaced by vehicle controls once moving, with the in-motion warning still able to override.



Both menus exist only in early HMI. Seat memory then moves elsewhere, vehicle controls take the screen once moving, and the safety warning overrides all of it with its own control.

OUTCOME

One mental model holding across three mechanisms with different signals and different limits, rather than three features sharing a menu. Translating a hard constraint into a coherent interaction, instead of forcing an ideal that cannot be built.

MY ROLE

State model, button architecture, warning behaviour and menu visibility rules across vehicle states.

HARD PART

Signal reality overruled the original single-button brief. The redesign had to lose nothing in clarity.

TRANSFERS TO

Any product where one surface fronts several mechanisms with unequal capabilities.

One menu entry, two incompatible state machines

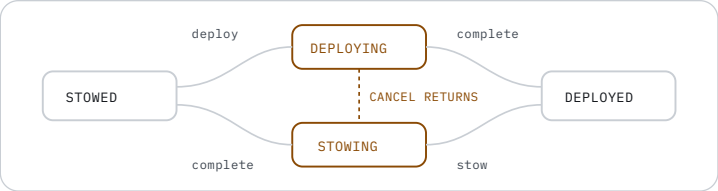
Depending on how the vehicle is fitted, “tow” means either a trailer, where the system can only run a check, or a powered tow bar that deploys and stows. The logic is not interchangeable, and the driver should never have to know which one they own.

TRAILER Start and stop during the check, with the relevant warnings raised while it runs. Deliberately minimal, because that is all the hardware supports.

TOW BAR Deploy, cancel mid-deploy and return to stowed with explicit feedback, stow, cancel mid-stow and return to deployed, from where stowing can be attempted again. Every transition needed feedback naming the direction of travel.

UNSAFE STATES Specific messaging rather than a generic warning, for example telling the driver the tow bar is not secure, to check the area, and not to tow.

COPY Every string defined in the specification. Nothing left as placeholder for engineering to invent.



Cancel is not an escape hatch, it is a direction change. The feedback has to say which way the mechanism is now moving.

OUTCOME

Two hardware-appropriate models behind what looks like a single feature, selected by what the vehicle is actually fitted with.

MY ROLE

Both state models, transition feedback, safety messaging and every user-facing string.

HARD PART

Mid-transition cancellation. Each cancel has a different destination and must be legible while moving.

TRANSFERS TO

Reversible long-running actions: uploads, deployments, physical automation.

APPS & NOTIFICATIONS

A notification system defined from first principles

One pattern had to carry a missed call, a safety warning and a routine update, without a bespoke rule for each new type added later. What I defined:

- **Density.** A clear visual difference between single-line and multi-line.
- **State.** Read against unread, with unread and expanded given more prominence.
- **Severity.** Distinct types and icons for red, amber and plain warnings, legible at a glance.
- **Grouping.** Related items, such as missed calls, collapse rather than stack.
- **Structure.** Variants with images and with accept or reject actions, each with its own sizing rules.
- **Interaction.** Scroll, expand, mark as read, and swipe to dismiss into history.

OUTCOME

A system flexible enough for the next notification type nobody has specified yet.

VARIABLE TRANSMISSION GLAZING

Two states that never revert on their own

The glass roof has two persistent states, opaque and clear. Neither times out and neither returns by itself. The state holds until the opposite option is selected, which makes the feedback question different from a momentary control: the driver is not being told that something happened, they are being told how the roof is now.

So the confirmation is a picture rather than a message. Choosing an option shows the roof from a front and overhead view, drawn in its new state, and that view stays true until the driver changes it.

OUTCOME

Feedback that doubles as a status display, which is what a latching state needs and a momentary one does not.

WIPERS

A menu the stalk opens, and arrows that know their own limits

The wiper menu is summoned by the stalk rather than from the screen, and offers four options. The middle area steps through automatic and intermittent levels with a pair of arrows. At the highest level the right arrow is disabled, at the lowest the left one is, so the control shows its own range instead of failing silently at the end of it.

The specification also settles the parts nobody notices until they are missing: what the cluster shows as feedback, how settings are reached from here, and how the driver gets back to the previous screen.

OUTCOME

A physical entry point, an on-screen menu that stops where the hardware stops, and a defined way back.

03 Standards, scale and transfer

Work that outlived its own programme: regulated symbols, cluster priority logic, and the shared library that let several teams stop rebuilding the same parts.

MASERATI CLUSTER

Telltales, priority logic and several layouts

Every telltale in the cluster enumerated, positioned and validated against the standards in force at the time.

Steering wheel controls change what the cluster shows, on a last-selected-wins rule, including interrupt and resume: change the layout during a call and the system allows it, then returns to the call view rather than losing its context.

Several layouts for different driving contexts, navigation-forward or an ADAS-forward two-dial view among them, each with its own content visibility rules over the same interaction logic.

OUTCOME

A cluster that changes what it shows with context and input, without ever reading as a different product.

JLR DESIGN SYSTEM

The shared library that replaced the blank file

Figma was new at JLR. Cluster specifications had been delivered as documents and slide decks, as they still were at Maserati. One cluster programme was running and a Jaguar programme was starting, both meant to share a visual and interaction language.

I collaborated on building a shared component library, including every telltale, cluster layout, and background, so that any designer could pull them into their own document instead of recreating them.

OUTCOME

A common foundation across cluster programmes. A process contribution as much as a visual one.

HOW I WORK

Specification is the deliverable

The artefact engineering builds from is a specification: interaction logic, every state, every string, every failure path. I take it through review with systems engineering, where signal availability and safety cases usually reshape the design, then review what returns before development handoff.

I also test what ships against the specification, which is how requirements like “every example command must work in every language” end up written down in the first place.

04 Skills

VOICE & CONVERSATIONAL

Native assistants and Alexa integration. Dialogue and prompt flows. Error and fallback handling. Multi-user voice priority. Multilingual evaluation in English, Italian and Spanish, coordinating French, German, Korean and Japanese.

INTERACTION & HMI

Interaction logic and state definition. Multimodal design across touch, voice, physical controls and cluster. Specification and requirements writing. Spec review and QA. Design to engineering handoff.

METHODS

User-centred design. Usability engineering. Requirements analysis. Competitive benchmarking. Ergonomics and human-computer interaction research.

COMPLIANCE & STANDARDS

NHTSA driver-distraction guidelines. Icon and telltale symbol standards. Safety-case driven warning design.

TOOLS

Figma, Jira, Polarion, Confluence.

CURRENT WORK

Designing conversational interactions for an LLM-based in-car assistant. The programme is confidential; the design questions are the ones above, asked of a system that generates its own language.

Terms used here

Telltale	The regulated warning symbols in an instrument cluster, with standardised shape, colour and meaning.
Cluster	The driver's own display behind the steering wheel, distinct from the central infotainment screen.
Early HMI	The interface state before the vehicle is fully woken into its comfort state, when a different set of screens is available.
Pick list	A short list of candidate results offered when a spoken command matches more than one thing.
Front trunk	Storage under the bonnet on electric platforms, which on these vehicles opens electrically but closes by hand.
Zone	A seating position with its own climate, voice and display controls, typically two or four per cabin.
Variable transmission glazing	A glass roof that switches between opaque and clear and stays in the state it was set to.
Stalk	The lever behind the steering wheel, used here to open the wiper menu on screen.

05 Work with me

CONTACT

ailed.labrada.sosa@gmail.com

+44 7928 483701
Birmingham, United Kingdom

LOOKING FOR

Permanent roles in voice and conversational design, HMI and interaction design, or human-robot interaction. Birmingham, the Midlands, London or remote within the UK.

AVAILABILITY

Available from 1 December 2026. UK Skilled Worker visa, currently sponsored, so a new employer needs to be a licensed sponsor.

Screens, model names and proprietary visuals from JLR, Maserati and Iveco are covered by non-disclosure agreements and are not reproduced here. These case studies describe design reasoning in the abstract. The process and the decisions are mine; the confidential deliverables are not shown.