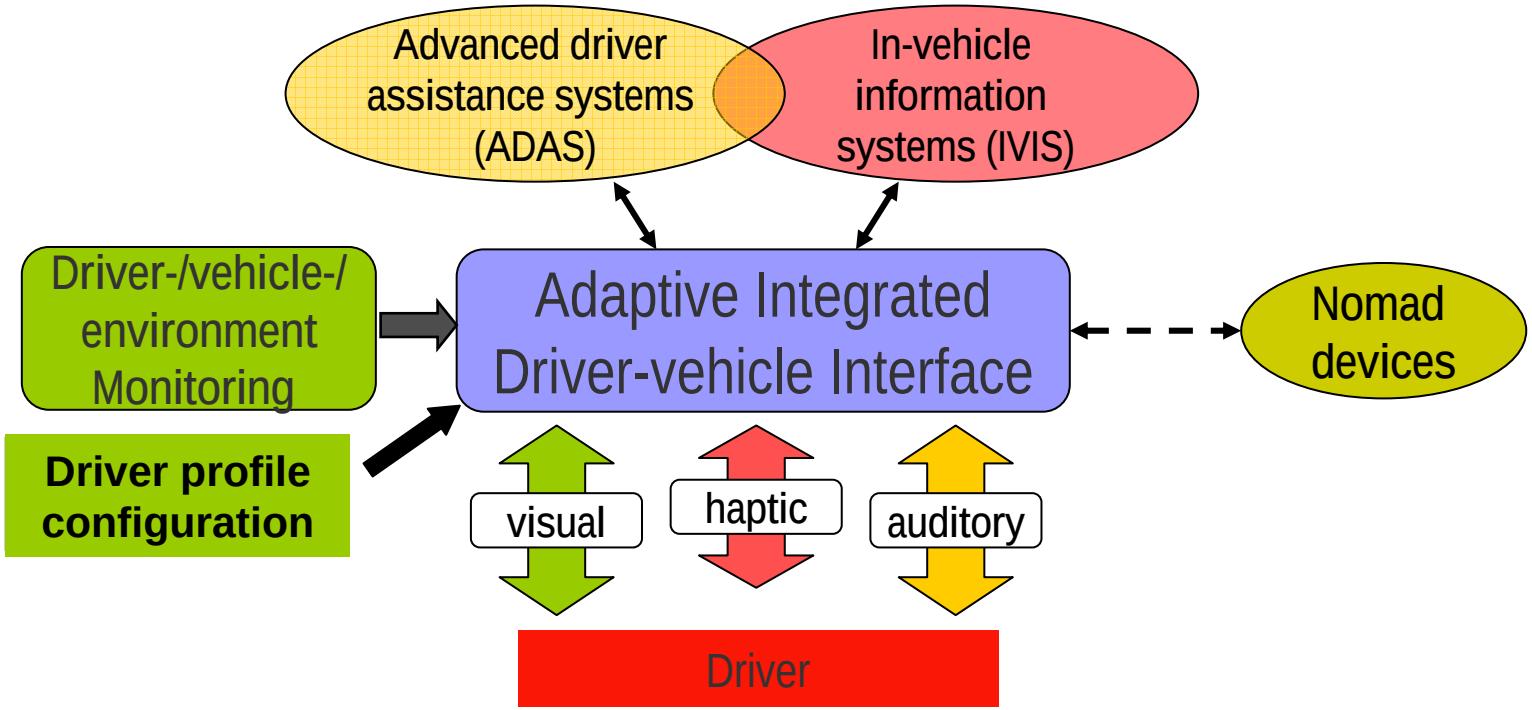
AIDE Use cases & Design Scenarios

Maria Romera Rué - SEAT

Objective of AIDE Scenarios

- To describe systems and functions to be integrated → desired primary functionality
- To verbalize the desired system's adaptivity in certain driver- system interactions under different Driver-Vehicle-Environment (DVE) conditions



Components of the AIDE scenarios

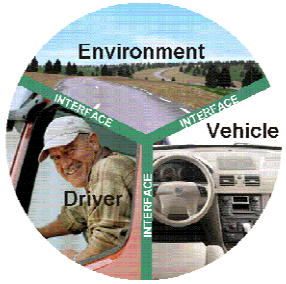
Interaction aspects

- IVIS / ADAS interaction
- Adaptivity
- HMI Personalization

Relevant terminology

- Actions
- DVE conditions
- Driver profile parameters

Scenarios' derivation

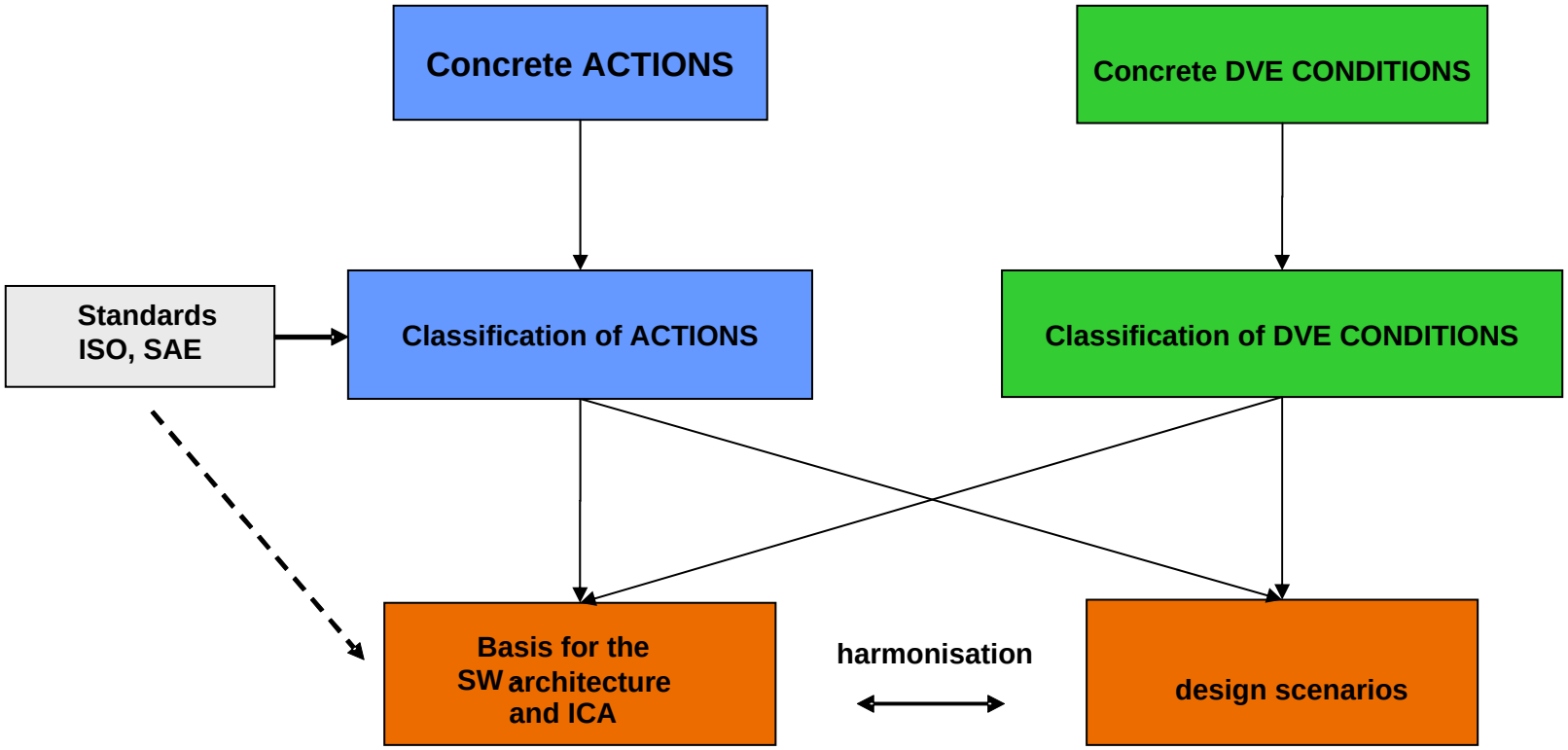
- Identification of possible conflicts
- Nomad devices use cases
- 

- Use cases for personalization

Action parameterization

DVE state parameterization

AIDE Scenarios design



Action categorization - Objective

The **objective** of the action categorization is to describe driver/application actions relevant for AIDE in a generalized format.

This has three purposes:

- To be the basis for the abstract representation.
- To enable generalized design scenario descriptions
- To make the AIDE independent of the function or system

Action categorization - Method

Define a set of basic parameters that can be used to characterize the actions:

AIDE action = f (initiator, duration, safety criticality, time criticality, real time, mandatory, driving relevance, preference)

Classes of actions:

- **Warnings**, which present very urgent information to the driver. Such an information is of highest priority for the driver and has to be presented in any case.
- **Dialogs** are initiated by the driver, then should be answered or followed by the system immediately. Nevertheless the first "warning" class is of higher priority and might even allow an interruption of a dialog.
- **Output messages** which comprises all output information not belonging to class 1 and 2.

Action categorization - Parameters

Parameter	Levels	Definition
Initiator	**User/ System**	Information about the initiator of the action.
Duration	**Transient/Sustained**	Duration of an action or information presentation
Safety Criticality	**None / Low / High**	Safety criticality of the output content for the safety of the driver or other traffic participants. The safety criticality is only related to the consequence if the information is not transferred to the driver. Thus, it characterizes mainly messages that indicate a current possible unsafe situation (vehicle malfunction or traffic-environment difficult conditions)
Time Criticality	**None / Low / High**	Time criticality with respect to a necessary action of the driver according to an information given by the system
Real Time	**Yes / No**	Signaling or having a real time communication with someone (- but not with the system).
Mandatory	**Yes / No**	Output information has to be presented due to law
Driving Relevance	**Yes / No**	Relevance of the action for the driving task. Distinguish between actions related to the driving task and actions related to secondary tasks.
Preference	**Yes / No**	The importance of this action is increased by the driver or others. That is only possible if the same action can take place as a non-preferred action like VIP versus normal phone call or urgent or normal traffic announcement

Action categorization - Classes

Classes	Definition (verbal description)	Set definitions
W (warning)	Warnings indicating high traffic/ environment risk	{safe criticality= HIGH, …*}
D (dialog)	Anything, which have to be activated by the driver and represents a real time interaction with the system or an other person (dialogs or inputs, real time activity)	{user initiated, …*}
OP_1 (output prio 1)	Mandatory messages or important info related to the instant driving task. This class can also include important message on second task applications (e.g. VIP phone call)	{time criticality= HIGH, safety criticality*,..*} OR {safe criticality = *, mandatory= yes, ..*} OR {real time= YES, user preference= YES}
OP_2 (output prio 2)	Output messages requiring an action in the near future or at least an increased attention - mainly for actions of the primary task (driving task)	{all the rest or those that don't belong in W, D, OP_1, OP_3}
OP_3 (output prio 3)	Sustained information about or for the primary task, not requiring an action in the near future. Also, messages or information for actions of the secondary tasks (not related to the driving task)	{system initiated, duration = sustained, driving relevance= YES, ...*}OR {system initiated, duration= transient, driving relevance= NO,...*} OR {System initiated, duration= transient, driving relevance= NO,...*}

Action categorization table

Action Parameter list	Initiator		Duration		Safety Critical			Time Critical			Real Time		Mandatory		Driving Relevant		Preference		PRIO
Action	System	User	Transient	Sustained	None	Low	High	None	Low	High	Yes	No	Yes	No	Yes	No	Yes	No	Priority class
Entering a telephone number		x	x		x			x				x		x		x		x	D
Frontal collision warning	x		x				x			x		x		x	x			x	W
Drowsiness alert	x		x			x				x		x		x	x			x	OP1
Parking distance information	x		x			x		x				x		x	x			x	OP2
MP3 Information	x			x	x			x				x		x		x		x	OP3

Classification of DVE conditions

ID	DVE parameter	Explanation	Value
DVE1	Driving availability (output of DAE)	driver's "level of availability" to receive and process information, according to the requirements of the current driving task (not available <=> high driving demand)	available not available intermediate state
DVE2	Distraction (output of CAA)	Cognitive load or shift of visual attention away from the road ahead, induced by an external event or a secondary task).	High Low intermediate stage
DVE3	Driver ability (output of DSD)	The physical ability of the driver to drive (fatigue, sleepiness, etc.)	High Low intermediate state
DVE4	Driver intent (joint output from TERA and CAA)	The driver's intention, e.g. for a lane change	Maneuvering non maneuvering (*intention only)
DVE5	Traffic and environmental risk (output from TERA)	The total level of risk concerning the environmental and the traffic conditions (Environment type, traffic density, environmental conditions etc.)	High Low intermediate stage

AIDE Design Scenarios

AIDE design scenario An application action or a combination of application actions is/are initiated/in progress in a single or complex DVE condition, DVEi.
Action/s = {…}
DVE Condition/s = {…}
Flow of events
Possible AIDE solutions(*) ...
Example : ...

AIDE Design Scenarios

AIDE design scenario 1.1: Conflict between concurrent warnings
Action/s = {W + W}, where W= warning indicating high traffic risk.
DVE Condition/s: DVE$_{1,3,4}$=LOW/NO, Traffic/environment Risk(DVE$_5$)=HIGH, Driver Distraction (DVE$_2$)=YES
Flow of events 1. W_1 is initiated 2. W_2 is initiated while W_1 is executing
Possible AIDE solution(s) 1. Allocate the default I/O so that interference between warnings is minimized 2. I necessary, re-direct W to a different modality 3. Let the more critical warning override the other
Example *Conflict scenario:* The driver is distracted and, as a result, swerves out of the lane. The lane departure warning system gives a warning. However, shortly after, the vehicle ahead brakes suddenly, triggering the forward collision warning. *Possible solution(s):* 1. Present the LDW in tactile model as default and the FCW in visual and auditive mode as default. 2. If the warnings interfere (e.g. if LDW is also auditory), the FCW could be given only visually. 3. Let the FCW interrupt LDW regardless of whether there is an I/O resource conflict.

AIDE Design Scenarios

AIDE design scenario 2.1: Conflict between an output message and a demanding driving situation
Action/s = {OP_i}, where i=1...3 OP_1= Mandatory messages or important info related to the instant driving task OP_2= Temporary info related to the driving task, requiring an action in the near future OP_3= Permanent status-info related to the driving task, not requiring an action in the near future OR output messages related to the secondary task
DVE Condition/s: Driving Demand (DVE_1)=HIGH, DVE_{2-5}=LOW/NO
Flow of events 1 1. DVE_1 turns from LOW to HIGH 2. OP_i is initiated 3. DVE_1 turns from HIGH to LOW
Possible AIDE solution(s) OP_i is delayed until DVE_1 turns from HIGH to LOW, if i>1 OP_i is presented if i<=1
Example: *Conflict scenario:* The driver enters a busy intersection. While negotiating the intersection, a phone call is received. *Possible solution:* Delay the phone call until after the intersection

AIDE Design Scenarios

AIDE design scenario 2.3: An important output message (OP1) is given while driver is tired and drives during night.
Action/s = {OP_1}, where OP_1= Receiving mandatory messages or important info related to the instant driving task
DVE Condition/s: Traffic/environment Risk(DVE_5)=HIGH, Driver Fatigue(DVE_3)=YES, $DVE_{1,2,4}$=LOW/NO
Flow of events 1 1. DVE_5 is HIGH and DVE_3 turns from LOW to HIGH 2. OP_i is initiated
Possible AIDE solution(s) OP_1 is presented in the most efficient way (choose/change modality) The intensity of OP_1 is enhanced
Example: *Conflict scenario:* The driver is tired and travels in a highway during night and under fog when a route guidance information “turn left immediately” is given from the navigation system *Possible solution:* Present the message in a vocal mode instead of a visual mode, if that was the default case (maybe with enhanced intensity)

AIDE Metafunctions

The AIDE meta-functions represent the responses of the AIDE system,

Possible solutions, to the conflict situations described by the AIDE design scenarios. The meta-functions could be regarded as "function that control other (more basic) functions", and are a key focus of in the AIDE project.

In the proposed AIDE functional architecture, the meta-functions are mainly implemented by the ICA (Interaction and Communication Manager).

Five general AIDE meta-functions have been identified:

- **HMI I/O management**
- **Action prioritization**
- **DVE-dependent action scheduling**
- **Modality selection**
- **Warning adaptation to DVE state**

AIDE Metafunctions

Meta function	Description
I/O resource management	Allocating I/O devices to application actions (maybe default standards by the demonstrators)
Action prioritization	According to ICA actions' conflict-settlement strategy. This is independent of the availability of I/O resources. Thus, even if I/O resources are available, an action, may be prevented from being executed concurrently with another action: 1. Actions-related conflict recognition 2. Actions scheduling: ▪ Include pauses (the system can add output pauses after messages) ▪ Terminate actions ▪ Interrupt actions
DVE-dependent I/O management and actions' scheduling	Debase non-critical actions in demanding situations: 1. DVE-related conflict recognition 2. Modality selection (the system can determine which modality an application should use in a specific DVE condition) 3. Modality alteration (the system can initiate a change in the used modality even for running actions in a specific DVE condition)
Modality dependent I/O management	Derive the possible modality which can be allocated according to ICA strategy (maybe default standards by the demonstrators)
Warning adaptation	Optimize critical warnings (and even OP_1) to driver state

SEAT

Centro Técnico

Maria.Romera@seat.es