INFORMATION SOCIETY TECHNOLOGIES (IST) PROGRAMME

AIDE

IST-1-507674-IP

System Architecture, data flow protocol definition and design and AIDE specifications			
Deliverable No.		**D3.2.2**	
SubProject No.	**SP3**	SubProject Title	**Design and development of an Adaptive Integrated Driver-vehicle Interface**
Workpackage No.	**WP3.2**	Workpackage Title	**Specifications And System Architecture**
Activity No.	**A3.2.2; A3.2.3**	Activity Title	**System Architecture and data flow protocol**
Authors (per company, if more than one company provide it together)		**Holger Kussmann (Bosch), Holger Modler (Bosch), Andreas Engelsberg (Bosch), Angelos Amditis (ICCS), Luisa Andreone (CRF), Anastasia Bolovinou (ICCS), Christophe Couvreur (Nuance), Enrica Deregibus (CRF), Johan Engström (VTEC), Stephane Feron (PSA), Paul Kompfner (ERTICO), Paul Piamonte (VTEC), Andreas Pringle (VTEC), Patrick Robertson (Motorola), Johan Scholliers (VTT)**	
Status (D: draft, in progress, S: Submitted to EC, F: Final accepted by EC)		**S**	
File Name:		**AIDE D3.2.2 v1.5.doc**	
Project start date and duration		**01 March 2004, 48 Months**	

Executive Summary

This deliverable aims to describe the AIDE system architecture. This software architecture focuses on the logical structure of the software components and their communication. It is based on the requirements documented in D3.2.1 [Kus05] and the scenarios described in D3.1.2 [Amd04].

In addition to this the individual components are specified in terms of their tasks, responsibilities and dependencies which are necessary to provide adaptive functionalities for an integrated, in-vehicle HMI system.

To realize I/O management of interfering output events and adapt driver system interaction to the driver status and preferences and as well to the driving situation, the following main components were identified:

- A central intelligent communication assistant called ICA performing the management and adaptation functionality. It contains the rules determining the system behaviour or HMI strategy perceived by the user.
- A driver vehicle environment component called DVE monitoring the driver and the driving situation to derive condition information about the driver, the vehicle and the environment which is used by ICA and applications to adapt the output messages.
- The applications which provide specific functionalities to the user. They have to be extended to communicate with ICA and to provide additional adaptive functionalities.
- A gateway component and virtual applications to integrate nomad device functionalities and data into the in-vehicle HMI.

Since HMI strategies and system extend concerning I/O devices and applications differs between different vehicle manufacturers and between different car segments, modularity and flexibility are the most important requirements. This strongly affects the principle design decisions, the specification of the interfaces and the communication flow. For these reasons it is important to note that the AIDE architecture describes a logical structure and semantical communication independent from a concrete vehicle implementation, because the latter differ widely between the OEM's.

In order to fulfil the mentioned requirements ICA and applications are designed strongly independent from each other, i.e. ICA must not know the semantics of an application. Thus, an output message is not rendered or performed by ICA, but by the application itself. It only has to ask ICA in advance for permission and informs ICA about the output form and desired devices. This information is used by ICA together with the DVE data to determine the suited output strategy. After informing the applications about that decision they perform the output and access the I/O devices. Nevertheless, due to timing reasons the ICA controls the final I/O device access determining which device is perceivable for the driver.

This mechanism also holds for the virtual applications connected to the nomad devices. A virtual application (for example an mp3-player) is treated in the same way as ordinary installed in-vehicle applications. It is a software module installed in the vehicle accessing the in-vehicle I/O devices and differing to a normal application only in the fact that the basic functionality is located in the nomad device. For this example the mp3 data and the MPEG decoder software player run on the nomad device. The virtual application offers the HMI software to control the player and use the in-vehicle HMI. Also for the AIDE specific functionality the virtual applications do not differ to the normal ones – they are communicating in the same way with the ICA.

All interfaces are specified using generalized content to guarantee the modularity. For example, the ICA has to assign priorities to each output request from the applications. This is done using informing parameters send by the application to the ICA. These parameters objectively characterize the output message which has to be done in terms of their importance for the driver like "driving relevance", "safety criticality", etc. All parameters are defined unambiguously and do not include application specific aspects.

Thus, the communication flow between ICA and application remains simple and uses only 4 different messages:

- The Application Request Vector (ARV) to ask for permission to perform an output.
- The Reply Vector (RV) to inform the application about if and how to perform the output.
- The Channel Status Vector (CSV) to inform ICA when devices are freed again.
- The Request No More Valid Vector (RNV) to inform ICA when a postponed output is no more valid.

In addition to that there is one Driver Vehicle Environment Monitoring (abbreviated as DVEM) Vector and one Profile Management (abbreviated as PM) Vector to provide condition information and driver preferences which are used by ICA for all general valid adaptive functions and by applications for the application specific functions (see also 4.4.1.1).

The nomad device interface is clearly most complex but all necessary interface data are described. In this case the used mechanism is based on existing and standardized technology – Bluetooth (BT), but defining a new BT AIDE profile (see 5.4.1). In this case AIDE recommends a specific implementation, because there is no established implementation available.