ARTEMIS	**HoliDes** **Holi**stic Human Factors **Des**ign of Adaptive Cooperative Human-Machine Systems	HoliDes

D5.6 – Techniques and Tools for Empirical Analysis Vs2.0 incl. Handbooks and Requirements Analysis Update

Project Number:	332933
Classification:	Public
Work Package(s):	WP5
Milestone:	M5
Document Version:	V1.0
Issue Date:	September 12, 2016
Document Timescale:	Project Start Date: October 1, 2013
Start of the Document:	Month 31
Final version due:	Month 36
Deliverable Overview:	**Main document**: Techniques and Tools for Empirical Analysis Vs2.0 incl. Handbooks (Public)
Compiled by:	David Käthner - DLR Klas Ihme - DLR
Authors:	Morten Larsen (AWI) Adam Herout (BUT) David Käthner (DLR) Klas Ihme (DLR) Juliane Bürglen (HFC) Zdenek Moravek (HON) Paul Kaufholz (PHI) Flavia de Simone (SNV) Marieke Thurlings (TWT) Iris Cohen (TWT) Marco Botta (UTO)
Reviewers:	Ignacio Gonzalez Fernandez (ATO) Sonia Bilbao Arechabala (TEC) Sara Sillaurren (TEC)

Technical Approval:	Jens Gärtner, Airbus Group Innovations
Issue Authorisation:	Sebastian Feuerstack, OFF

DISTRIBUTION LIST		
Copy type[1]	Company and Location	Recipient
T	HoliDes Consortium	all HoliDes Partners
E	ECSEL JU	Patrick Vandenberghe

[1] Copy types: E=Email, C=Controlled copy (paper), D=electronic copy on Disk or other medium, T=Team site (AjaXplorer)

RECORD OF REVISION		
Date (DD.MM.JJ)	**Status Description**	**Author**
01.06.2016	Initial Version	David Käthner
07.07.2016	Added Feedback after WP5 conference call	David Käthner
30.08.2016	Added all contributions received so far	David Käthner
01.09.2016	Added the last contributions, a few workflows are still missing.	David Käthner
02.09.2016	Finalisation for the first draft version.	David Käthner
09.09.2016	Integration of the reviewer comments from TEC	David Käthner
12.09.2016	Integration of the reviewer comments from ATO	David Käthner
12.09.2016	Added textual description for one WP9-workflow	Fabio Tango / David Käthner
15.09.2016	Added BUT contributions (workflow and figure caption), requirements updates for DLR-MTTs, changed figure captions for TWT.	David Käthner
15.09.2016	Fixed captions for workflow figures.	David Käthner
16.09.2016	Added description for 5.2 workflow description	David Käthner
26.09.2016	Edited TWT contribution for public version	Iris Cohen

1 Table of Contents

2 List of Figures

3 List of Abbreviations

ADAS	advanced driver assistance system
AdCoS	Adaptive Cooperative Human-Machine System
CPM-GOMS	Cognitive Perceptual Motor-Goals, Operators, Methods, Selection rules
FG	focus group
HF-RTP	Human Factors - Reference Technology Platform
HMI	Human-machine interface *or* Human-machine interaction
MTT	Method, Technique and Tool
OSLC	Open Services for Lifecycle Collaboration
UML	Unified Modelling Language
WP	Work Package

4 Introduction

The main objective of WP5 within HoliDes was to develop methods, techniques and tools (MTTs) for the empirical analysis of Adaptive Cooperative Human Machine Systems (AdCoS). This last deliverable thus presents the final state (version 2.0) of the developed MTTs and shows their integration into the AdCoS development process and the Human Factors Reference Technology Platform (HF-RTP). WP5 was designed to receive requirements from the application work packages 6-9 and from the HF-RTP definition in WP1. WP5 partners worked in close cooperation with the experts from WP2 (Modelling), WP3 (Adaptation) and WP4 (model-based Human Factor Analysis).

In total, 14 human factors MTTs with different applications in the design, development and operation of AdCoS' have been engineered, adapted or extended in this work package. Of these, four can be considered as non-life cycle MTTs and will be integrated into cars or aircrafts to detect and determine operator states. One innovative example of these is the cognitive distraction classifier developed by TWT that has been integrated into the IAS AdCoS in the automated overtaking use case. In addition, WP5 produced ten life-cycle MTTs that aid the design and development process of AdCoS at certain stages. These include quantitative and qualitative empirical methods (e.g. Theatre Technique by DLR or SNV's focus groups), task analysis techniques and tools (e.g. the HF-TA by HFC) or tools for capturing and structuring the human-centred design process (e.g. HF Filer by AWI). Use cases from each of the four domains (Healthcare, WP6; Aeronautics, WP7; Control Room, WP8; Automotive, WP9) have been served by WP5, ensuring that the developed Human Factors MTTs will be applied and exploited in a wide range of applications.

In the course of the project, WP5 had many contacts and interactions with the other scientific WPs 2-4. In particular, the task and human models created in WP2 were informed by the task analysis techniques and empirical methods created in WP5. Workshops between the work packages helped sharpening the concepts commonly used, and create a common vision for task and human modelling. Empirical data from WP5 also helped to create algorithms and methods developed in WP3. In turn the non-life-cycle methods of WP5 can be utilized to create adaptive, context and user aware functionalities. Finally, for verification and validation the model-based evaluation MTTs of WP4 were complemented by the empirical evaluation techniques created in WP5. Examples for this collaboration effort can be found in the workflows and accompanying sections in 5.3 and 5.7.

The main challenges of WP5 were the integration of the MTTs into the HF-RTP and the demonstration of the collaboration with the partners from WP2-4. The former challenge comes from the fact that many of the MTTs of WP5, especially the empirical methods and task analysis approaches need manual and cognitive input from a human factors expert to exploit their full potential, which renders an OLSC compliance at all processing steps demanding. Thus, it was decided to show the usage of the MTTs within workflows for AdCoS design and development (see section 5). These workflows comprehensively show at which point of the process the MTTs should be utilized, thus aiding the exact tailoring of the HF-RTP.

A second measure to tackle the OLSC compliance was taken by AWI in designing the HF Filer (see sections 5.1 and 6.2), a tool that allows to document information from the complete human-centred design process and save these pieces of information in an OLSC-compliant format. The latter challenge arose as, in the beginning, the interplay of the developments of WP2-4 with WP5's MTTs was only apparent when assuming an application- or use-cased based approach. However, this was overcome as the workflows also make interactions between developments of different WPs apparent and therefore demonstrate how the results of WP2-4 and WP5 act together.

The structure of this deliverable D5.6 is as follows: It will first elaborate on the AdCoS workflows to delineate how the MTTs from WP5 interplay with each other and support developments from the other work packages to achieve the AdCoS realization (see section 5). In the following section, the versions 2.0 of the 14 developed MTTs are provided. These contain detailed descriptions on the functionality and application, an evaluation with respect to the requirements, and feedback from the AdCoS owners regarding the usefulness and usability of the MTTs (see section 6).

As a concluding note, as WP5's final deliverable, D5.6 is conceptualised as a stand-alone document, in order to be understood by recipients without having to confer to other documents more than necessary. This implies some overlap with previous deliverables when this was unavoidable to give a complete description of an MTT. References to other deliverables were made in cases of them containing more details of AdCoS or MTTs.

5 Workflows

This section is intended to display the workflows between WP5 and other work packages to develop the AdCoS' from WP6-9. It is intended to give a quick overview over the actual process of working together, especially regarding the interaction between WP2-5. MTTs from other work packages are included in these workflows inasmuch they interacted directly with WP5 MTTs. For instance, a number of activities in WP5 were concerned with collecting data for models developed in other work packages, which had to be specifically geared towards producing the required empirical output for those modelling efforts.

These workflows are not meant to explain the entire workflow of the AdCoS-development, rather to show coherently how the different research efforts in HoliDes, especially in WP5, all contributed towards the realization of each AdCoS' addressed in WP5.

5.1 WP6: Various AdCoS (PHI), HF-Filer (AWI), Means-ends-modelling (AWI), HF-TA (HFC), U-DAT (PHI)

Four MTTs from WP5, i.e. U-DAT, Means-ends Modelling, HF-task analysis and HF-filer, have been used in different stages of the general usability process for AdCoS in WP6 (Health). This allows for a more systematic concept validation and capturing of usability results. For the graphical representation see Figure 1.

To support the usability for an AdCoS design a number of usability activities have to be performed:

1. Perform a **task analysis** to identify user needs and tasks
2. **Design** a user interface for the identified requirements
3. Perform **empirical studies** to validate the UI design
4. **Report** on all above activities

Ad 1: Task Analysis

HF-TA and Means-end Modelling are tools to capture user needs and tasks based on system and domain knowledge. Also, the relations between the different entities in the task analysis are captured using these tools. The outcome of these tools is stored in a commonly agreed XML format for the

Task Map Model, called TMM 1.0[2]. The TMM 1.0 format has taken into account the theoretical task modelling from WP2, but adjusted and simplified it to suite the practical needs to link the MTTs in a real AdCoS development process. The output from the task analysis tools in the form of a TMM1.0 based file is used and extended in subsequent usability activities.

Ad 2: Design

Designing the user interface for an AdCoS is done based on the user needs and tasks identified in the task analysis. This is a creative design process where not only the task analysis is taken into account but also, UI Guidelines, style guidance, standards, etc.

[2] Details of the file format are discussed in deliverable D6.8, with further overview given in section 6.2.

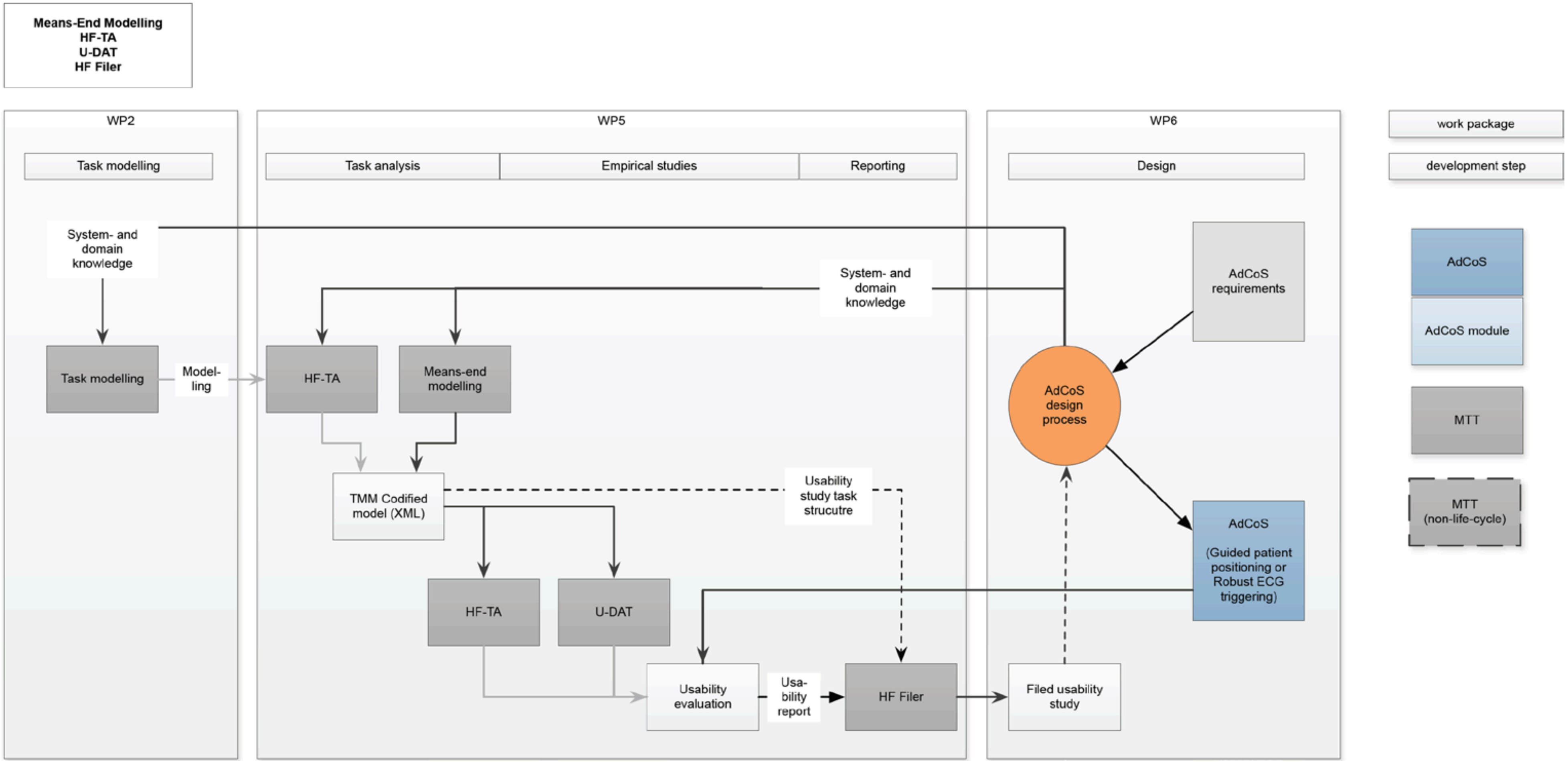


Figure 1: Usability workflow for a set of WP5 MTTs in the context of WP6 (Health).

Ad 3: Empirical Studies

Eventually, the UI design should fulfil the needs identified in the task analysis. To validate the UI design against the user needs empirical user studies are performed.

HF-TA and U_DAT have modules meant for performing structured user studies. This structured way of performing user studies is also based on the output of the task analysis. Therefore, these MTTs have interfaces built in to read the TMM-based files. The user test results can eventually be captured in an extended version of TMM. Instead of an informal usability report, which today is a common way to document study results, you might export a TMM file containing the user study results in a structured way.

Ad 4: Reporting

In the Health domain the task analysis and empirical user study results are formal documents that need to be auditable.

HF-Filer is a tool that can help archiving the study results in a structured way. Like the structure of a usability study is based on the information from the task analysis in TMM format, the structure that HF-Filer needs to store study results is the same. Hence, also HF-Filer can read a TMM-based file to create an internal structure that eventually can be filled with corresponding data from the exported information from the empirical studies. As such the usability study results have been filed.

5.2 WP6: Safe Parallel Transmit Scanning AdCoS (UMC), HF Filer (AWI), Focus Group (SNV)

The workflow for the Safe Parallel Transmit Scanning had the aim to develop and evaluate an AdCoS able to adapt to the level of expertise of the operator and to communicate to the operator the status of the system and the required actions. In particular, these activities implied a collaboration between WP6 and WP5 focused on a specific part of the AdCoS, the HMI. Potential end-users have been included into the designt and evaluation processes of the HMI by means of a methodology developed in social sciences to collect qualitative data, the focus group. More details about the methodology have been inserted in section 6.12. The data collected during the focus group sessions have been analysed by means of a content analysis and the main results have been reported in HF filer, described in section 6.2. In this way annotations about HF issues have been taken into account during the development process.

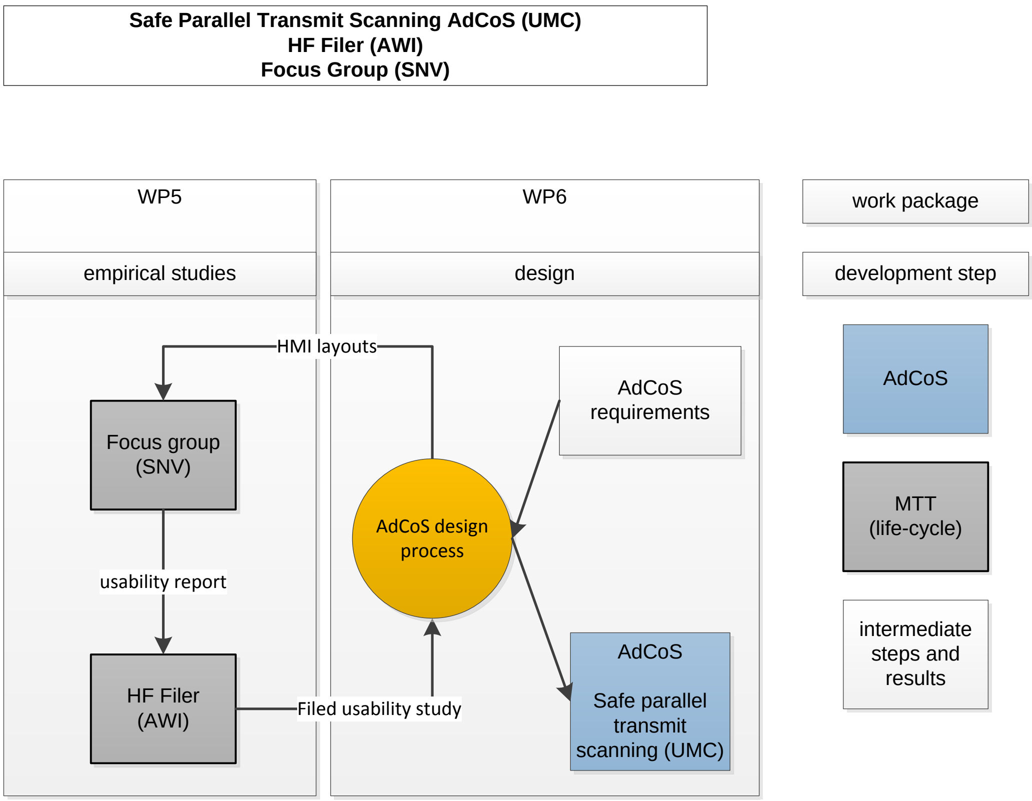


Figure 2: Workflow between AWI, SNV and UMC for the Safe Parallel Transmit Scanning AdCoS.

5.3 WP7: DiVA (HON), Operator state detection from hand-gestures (BUT), Detection of operator's head orientation (BUT)

The Operator State Detection-MTT was designed based on requirements and continuous interaction with/from WP7, mainly requirement WP7_HON_RTP_REQ78 "Create a tool/methodology that is able to classify an action of agent (human, machine) being either appropriate or erroneous. It is assumed that the tool has a task/procedure model with all supported alternate actions for a given situation." Since the MTT is specific in its input data (high-bandwidth video stream from a camera displaced in the operator's vicinity), its inputs are virtually impossible to be modelled by the standard means. This WP5 MTT took advantage of model-based tools for SW development and runtime error occurrence. The human factors modelling methodology is based on the principles defined within WP2.

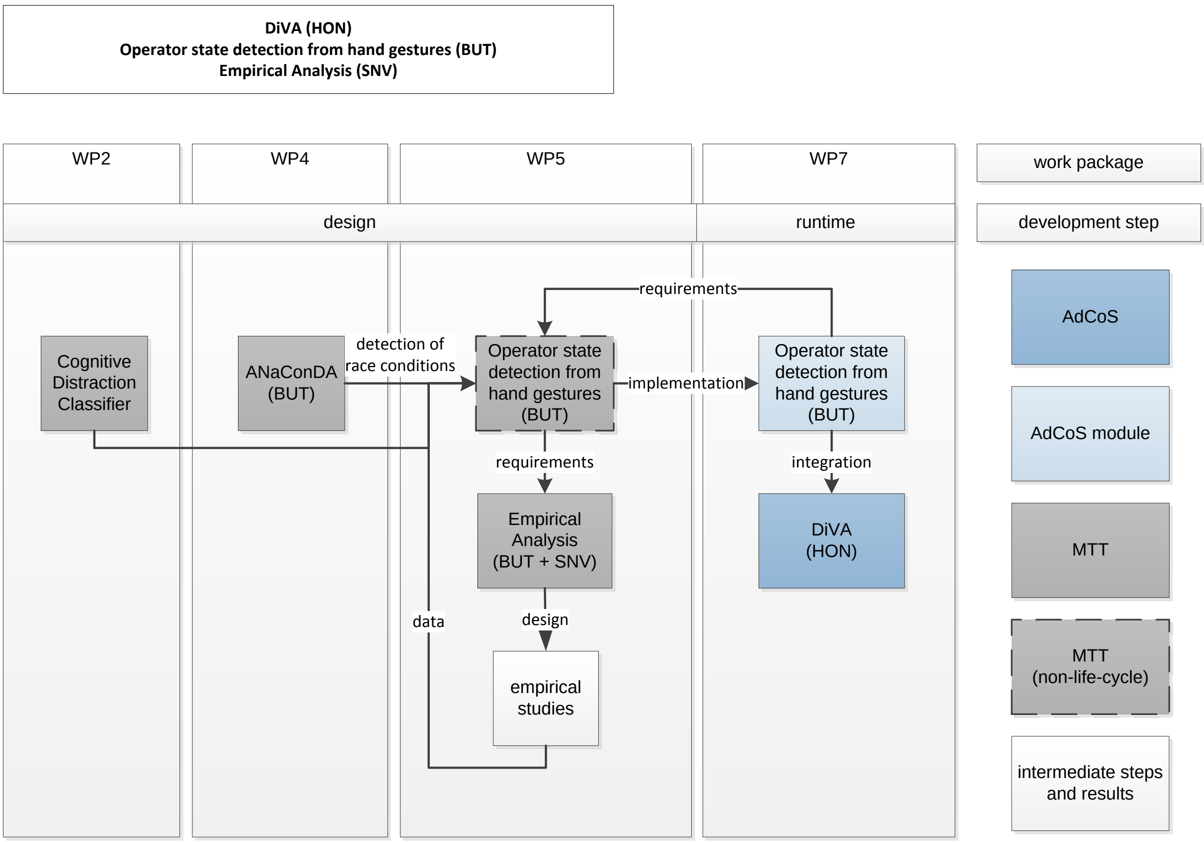


Figure 3: Workflow 1 between BUT, HON and SNV for the DiVA-AdCoS.

The workflow presented in Figure 4 around the Detection of operator's head orientation-MTT reflects the requirements defined within WP7, mainly WP7_HON_RTP_REQ82 "Compare benefits and disadvantages of using either head-mounted or cockpit mounted eye-tracker in highly unstable environment (cockpit, car). Define best practices/constraint when either of the two is more relevant.", WP7_HON_RTP_REQ83 "Investigate strategies of using eye-tracker when the subject needs to - turn head in wide range of angles, - may wear sunglasses or headsets, - undergoes sudden changes in illumination, - may need to change seat, - needs to be monitored for a long period of time.", and WP7_HON_AER_REQ90 "The AdCoS should monitor the head orientation of the pilot to detect if (s)he checked the mode of the aircraft to detect "missed event" situations." Since the MTT is specific in its input data (high-bandwidth video stream from a camera displaced in the operator's vicinity), its inputs are virtually impossible to be modelled by the standard means. This WP5 MTT took advantage of model-based tools for software development and runtime error occurrence. The human factors modelling methodology is based on the principles defined within WP2. The adaptability is supported by Adaptation MTTs as developed/defined in WP3.

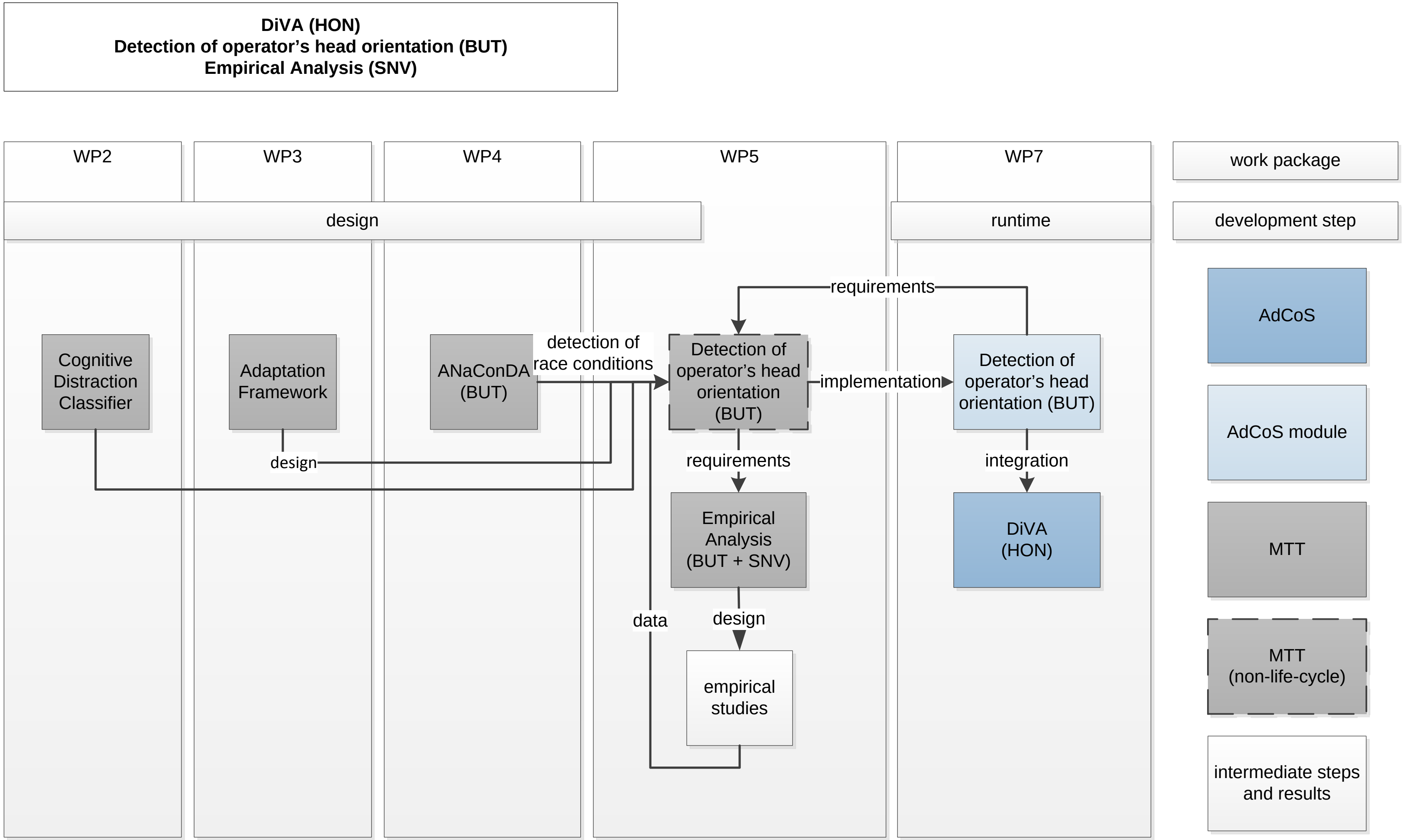


Figure 4: Workflow 2 between BUT, HON and SNV for the DiVA-AdCoS.

5.4 WP8: Border Security AdCoS (Airbus-DE), HF Filer (AWI), Focus Group (SNV)

The workflow for the Border Security AdCoS implied the utilization of an empirical methodology to evaluate the usability of the system considering experts judgements. Experts opinion has been collected by means of the Focus Group methodology (see section 6.12). The discussions addressed during the Focus Group have been registered, transcribed and analysed by means of the content analysis methodology. The report has been inserted in HF filer (see section 6.2). By means of this tool HF annotations could be considered in the development of the system.

Border Security AdCoS (Airbus-DE)
HF Filer (AWI)
Focus Group (SNV)

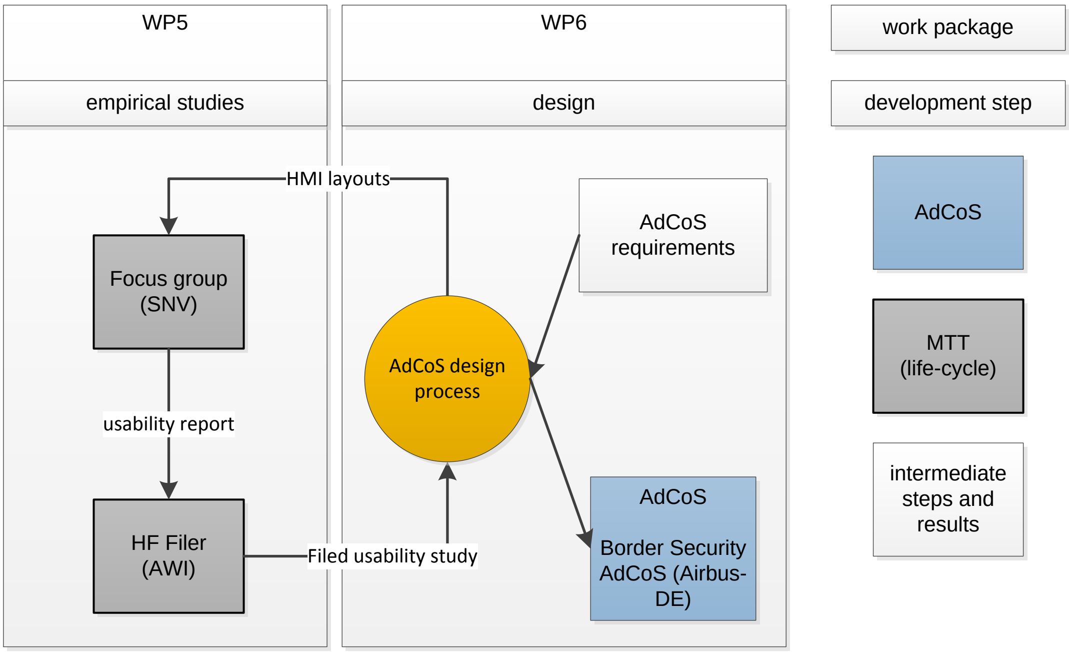


Figure 5: Figure 6: Workflow between Airbus-DE, AWI and SNV for the Border Security-AdCoS.

5.5 WP9: Adapted Automation AdCoS (IAS), CPM-GOMS (DLR), Methods and techniques for driver adaptive parameterization (DLR), Theatre Technique (DLR)

The workflow for the Adapted Automation AdCoS focused on developing two pieces of functionality: Individually preferred automated driving styles, and a fluent handover-of-control. For both developments, MTTs from WP5 were used, in collaboration with WP3.

The first functionality encompasses the adaptation of an automated vehicle's driving style to individual preferences of drivers. It is explained in more detail in section 6.6. To explore possible driving styles of an automation, those styles from human drivers were investigated first in a driving simulator experiment. After extracting prototypical driving styles, those prototypes were rated in a second experiment by human subjects. The data from these experiments was also used to extend the functionality of the CONFORM tool from WP3, to allow for automatic driving style classification in a real vehicle. Based on the personal driving styles and the prototype rating, a prediction was possible to adaptively choose between different driving styles in the simulator and the real vehicle. This functionality was then handed over to the Adapted Automation AdCoS.

Regarding the second functionality, handover-of-control between an automated vehicle and human drivers remains an open and critical issue, due to up to now unknown behaviours of automated cars in border conditions. Thus, a new concept to design such handovers was created. To develop the concept, the Theatre Technique was employed (see section 6.8) to explore design alternatives. This design session delivered as an outcome a state chart which serves as the basis for the implementation of the fluent handover-of-control. The CPM-GOMS task analysis was then used to analyse situations in which difficulties may arise during the transition, due to workload or lack of situation awareness of the driver.

Adapted Automation AdCoS (IAS) **CPM-GOMS (DLR)** **Methods and techniques for the driver adaptive parameterization of a highly automated driving system (DLR)** **Theatre Technique (DLR)**

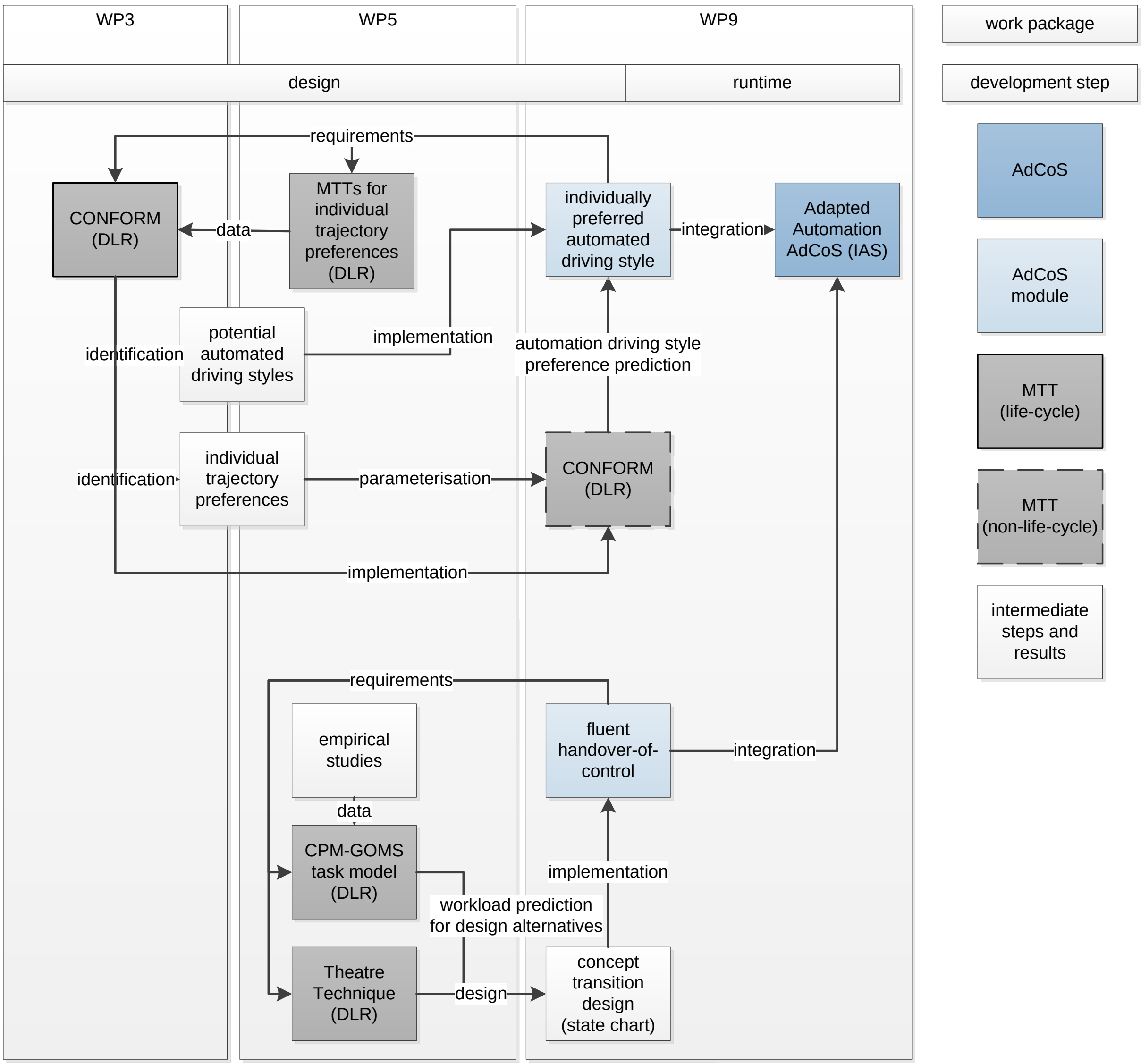


Figure 7: Workflow between IAS and DLR for the WP9 Adapted Automation AdCoS

5.6 WP9: Adapted Automation AdCoS (IAS), Adapted HMI AdCoS (TAK), Cognitive Distraction-Classifier (TWT)

TWT's non-lifecycle MTT, the Cognitive Distraction Classifier, comes from the need to detect cognitive distraction in the automotive industry. The AdCoS module "Detection of cognitive distraction" provided requirements as input for several WPs in which TWT contributed with different intermediate steps and results. An Applied Cognitive Model was established and refined into a Computational Model. In WP5, TWT conducted experimental studies in order to provide validation for the established human distraction model. TWT's activities spread over several WPs all of which contributed to the CDC that was finally implemented into AdCoS in WP9. The CDC has been integrated into the Adapted Automated AdCoS (IAS), with the possibility of its integration into the Adapted HMI AdCoS (TAK) having been positively evaluated.

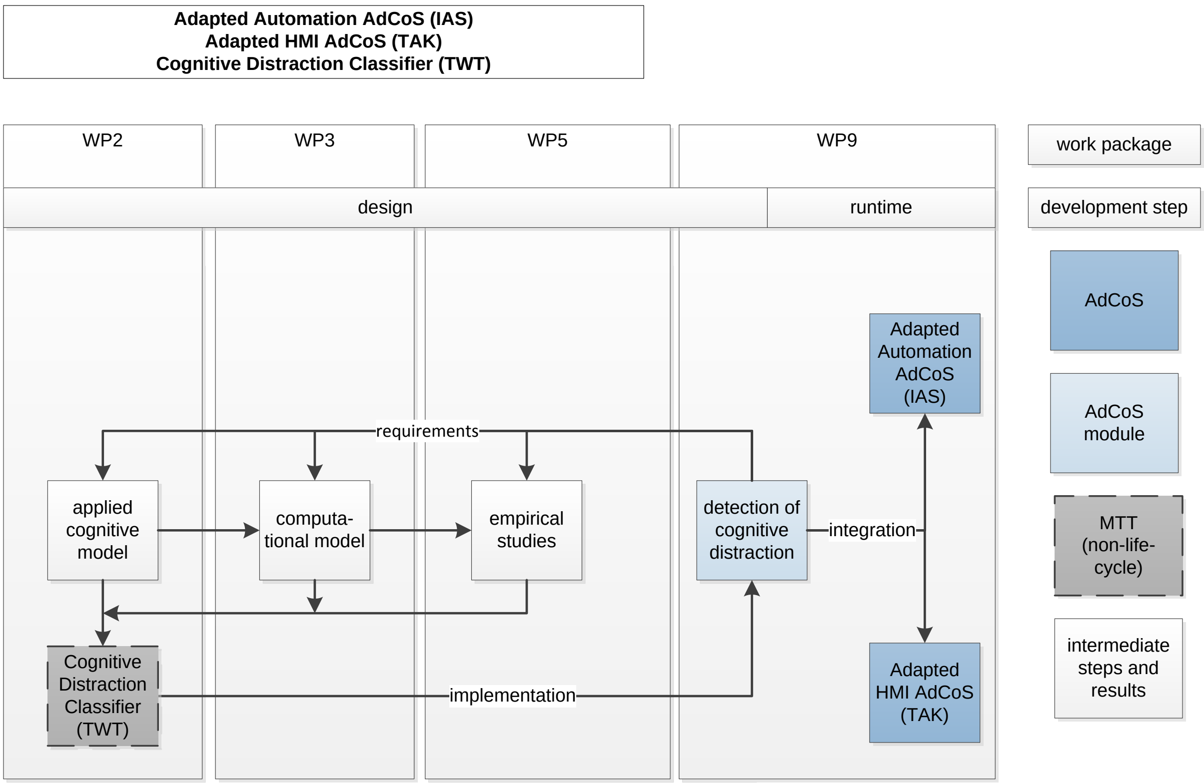


Figure 8: Workflow between TWT, IAS and TAK for the WP9 Adapted Automation AdCoS.

5.7 WP9: Lane-Change Assistant (CRF), SuRT (DLR), Empirical Analysis (SNV), Visual Distraction-Classifier (UTO)

The workflow for the Adapted Assistance AdCoS focused on the methodology to create distraction conditions for the user and on developing a visual distraction classifier (VDC). For both developments, MTTs from WP5 were used, in collaboration with WP4.

The first aspect is related to the adaptation of SuRT tool into RT-Maps software framework, which has been integrated both in the real vehicle and in the driving simulator. In particular, SuRT has been used to:

- to collect data about driver's distraction, then used to develop the related MTT (visual distraction classifier)
- in the evaluation phase, to create the "distractor" (i.e. the source of distraction) and thus to assess the adaptivity of the AdCoS.

More details can be found in section 6.7.

Regarding the second aspect, VDC is a non-lifecycle MTT, developed by UTO and CRF, able to classify the driver visual distraction, by using a supervised Machine Learning (ML) approach. The Visual Distraction Classifier comes from the need to detect the distraction in the automotive industry. This module receives as inputs the vehicle dynamic and environmental data and provides a binary output (Distracted = 1; Not Distracted = 0) to the HMI blocks and to the Co-pilot module (see Figure 9), which is responsible to take this information into account for the AdCoS strategies. This model was then checked using the GreatSPN-MTT from WP4.

An applied model was established and refined into a computational model, integrating it in the WP9 demonstrator, specifically in the Adaptive Assistance AdCoS, with its integration into the distributed HMI, as designed by REL.

More details can be found in sections 6.14.

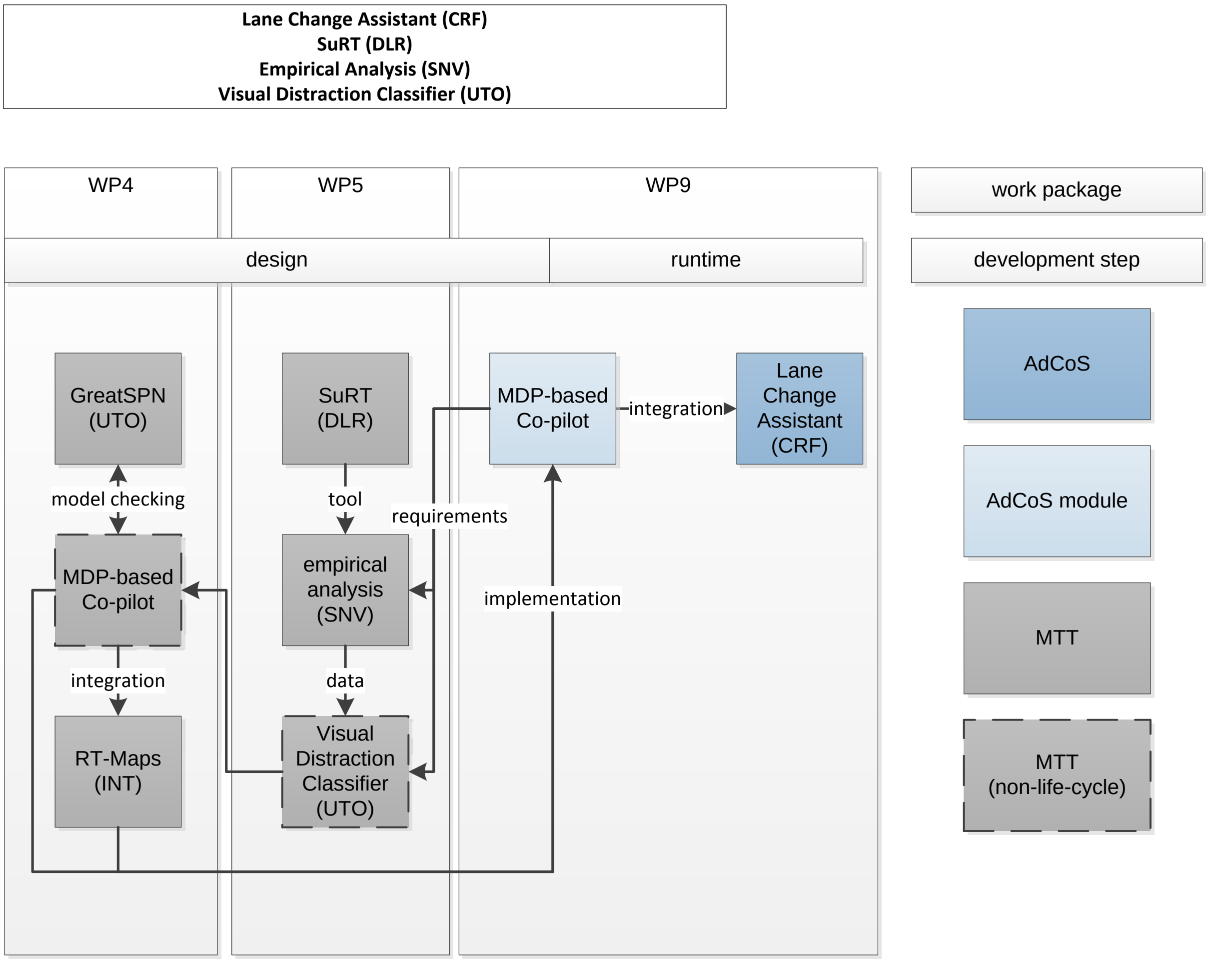


Figure 9: Workflow between CRF, UTO, SNV and DLR for the WP9 Lane Change Assistant-AdCoS.

6 MTTs Developed in WP5: Final Versions

6.1 Means-ends modelling of AdCoS (AWI)

6.1.1 Overview

Means-ends modelling of AdCoS can be used at different stages of the design and development process. In the context of HoliDes, it is used in the evaluation or validation stage, where it can be used to produce task models at different levels of abstraction to structure the analysis work[3].

This MTT allows the high-level task analysis work typically done in a conceptual phase to be exported directly to a new common file format that allows creating detailed usability test templates in an integrated human factors tool chain. This tool chain makes the design, development and certification process more efficient. HoliDes has allowed the definition of industrial development use cases and to refine and focus the analysis method on users operating computerised equipment.

6.1.2 Functionality

The method uses the functional modelling from a means-ends perspective to help analyse the abstractions (user needs and goals) behind a sequence of actions (user steps). By breaking these down into subsequently finer details and more concrete user actions, the method allows mapping user actions (or omissions of actions) onto safety related goal states and helps to categorise user error for feedback to system designers.

6.1.3 Overview of the analysis method

In order to guide the analysis, the functionality of the AdCoS is classified according to a system that is described in detail in deliverable 5.4 with extensions in deliverable 5.5.

[3] In the design stage, it can be used to identify functional blocks of the design and provide guidance for the HMI design and the structure of the supporting AdCoS. This aspect has been discussed briefly in deliverable 5.4 and will not be further discussed here.

In short, the method operates on three different levels, namely *Why*, *What* and *How*.

To make the analysis easier to map to the physical world (controlled entity and the AdCoS itself), we add *objectives*, as specialised, verifiable sub goals. Similarly, the "how" part is split into actions and physical equipment. This brings us to the final classification used in this method:

1. Goals (Why)
2. Objectives (Why)
3. Functions (What)
4. Behaviour (How)
5. Structure (How)

It should be noted that in the tailoring process in WP6, an extra layer has been added to express the "Who" part of the actions, in the form of a user level.

The MTT requires that one or more purposes can be associated with the operator's tasks, in such a way that goals at a higher level of abstraction can be identified. The more specific sub goals (the objectives) need to have criteria associated, and the modeller will need to have access to these criteria, or be able to determine them independently. They are dependent on the AdCoS, the use case and the test scenario.

In most cases, the data will be observations (or log data from the AdCoS) recorded in a format that can be used to verify if a given goal has been achieved.

As a part of the modelling, the method also allows for colour coding of types of goals and objectives. This is reported in deliverable 5.5.

6.1.4 Exporting task models to a common file format

To produce a document with the task model, the model must be encoded in a file format, and to ensure easy use in other applications, this should be a simple format.

To that end, the Task Map Model format previously explained in section 5.1, was also used during the tailoring process of the HF-RTP in the development of the Guided patient positioning and VCG triggering AdCoS. The export file format is based on XML, and the corresponding schema is available in annex I of D6.8. For this report, just the objects and properties are listed below.

Property	identifier	Type	Attr./ elem.		Constraints
System map (XML root element)					
ID	ID	String	Attribute	required	xs:id
Name	name	String	Element	[1..1]	1 to 40 chars
Task map					
ID	ID	String	Attribute	required	xs:id
Name	name	String	Element	[1..1]	1 to 40 chars
Context of use	contextOfUse	String	Element	[1..1]	Free text
User					
ID	ID	String	Attribute	required	xs:id
Role name	name	String	Element	[1..1]	1 to 40 chars
Characteristics		String	Element		Free text
Need					
ID	ID	String	Attribute	required	xs:id
Name	name	String	Element	[1..1]	1 to 40 chars
Need	need	String	Element	[1..1]	1 to 256 chars
Description	description	String	Element		Free text
Category	category	String	Element	[0..1]	1 to 40 chars
Relation					
ID	ID	String	Attribute	required	xs:id
Type	type	String	Attribute		Enumeration: include, specialization, optional
Parent	parent	String	Element	[1..1]	xs:id
Child	child	String	Element	[1..1]	xs:id
Execution Order	executionOrder	Number	Element	[0..1]	>= 0
Task					
ID	ID	String	Attribute	required	xs:id
Name	name	String	Element	[1..1]	1 to 40 chars

Property	identifier	Type	Attr./ elem.		Constraints
User goal	userGoal	String	Element	[1..1]	1 to 80 chars
Description	description	String	Element		Free text
Step					
ID	ID	String	Attribute	required	xs:id
Name	name	String	Element	[1..1]	1 to 40 chars
Description	description	String	Element		Free text

To create and edit the model, an exporter has been developed in the form of a commercial off-the-shelf application (OmniGraffle).

A screenshot to show the editor used to create a small example model is shown in Figure 10 and Figure 11 below.

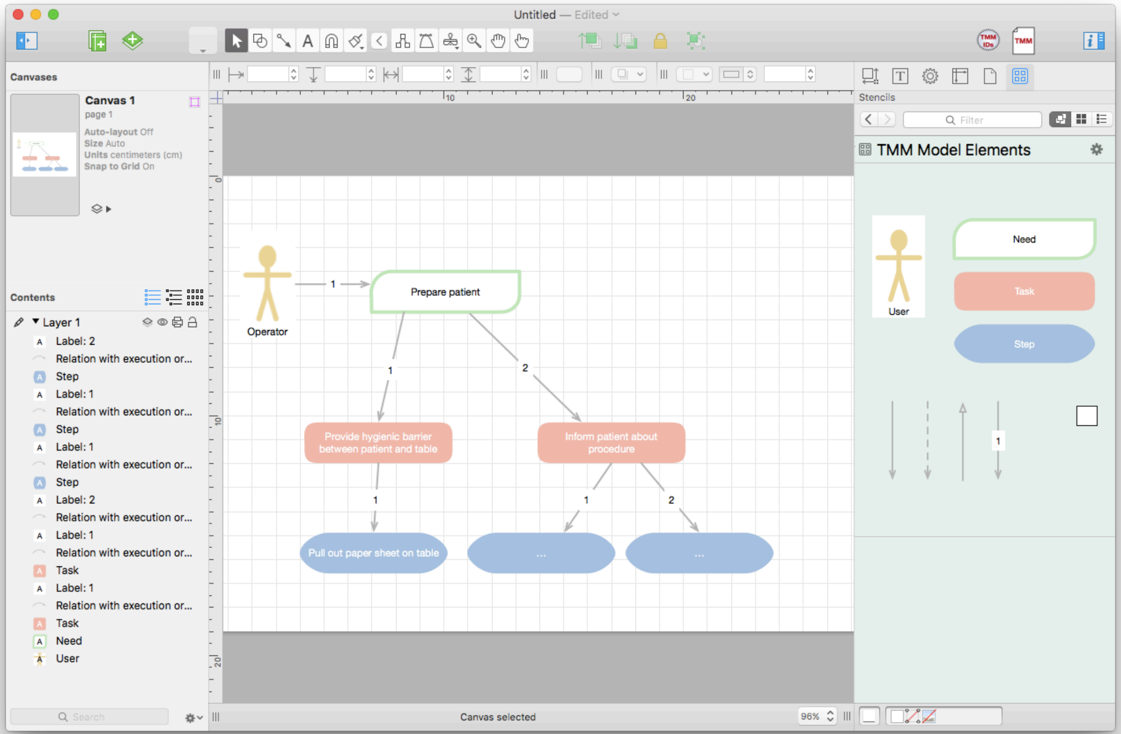


Figure 10: Screenshot of the Means-End exporter, with a library of elements on the right.

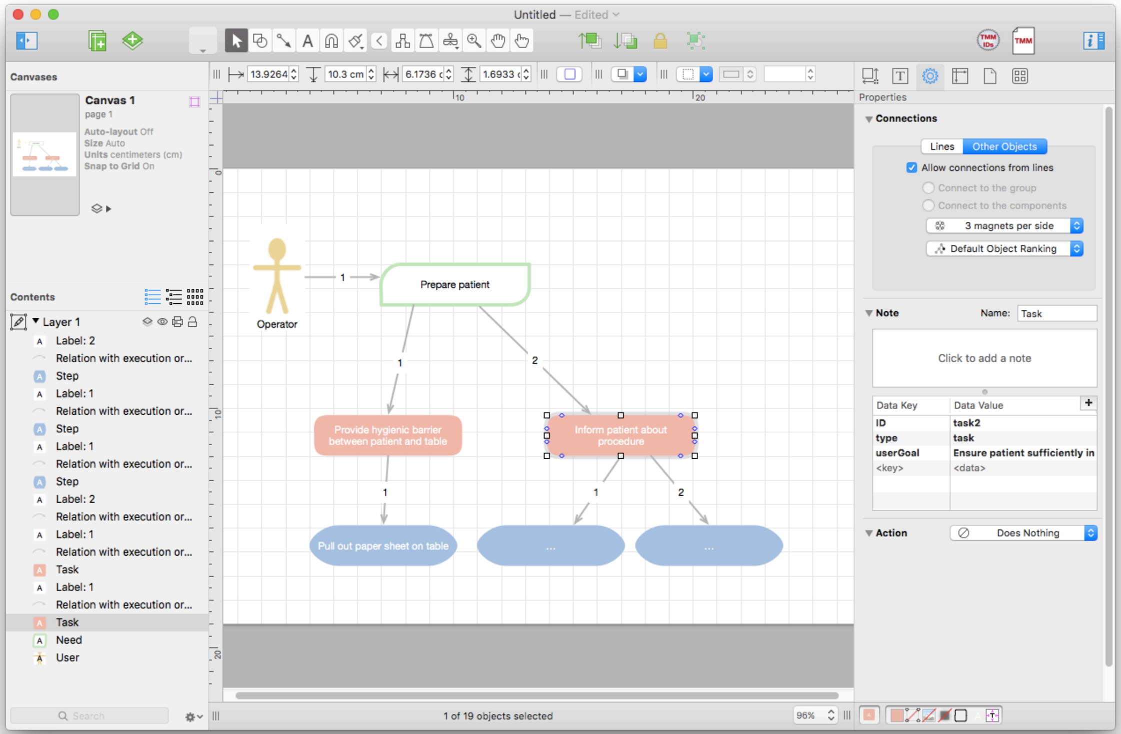


Figure 11: Means-End exporter, with pane to edit properties on the right.

The exporting functionality used to create the TMM file is accessed by clicking on the TMM icon in the toolbar seen in the screenshots.

6.1.5 Evaluation

Requirement:	**Compare recorded operator behaviour with official procedures**
addressed requirement(s)	
ID:	WP6_HEA_EBA_REQ_01
Ver:	0.2
Description:	Match recordings of operator behaviour to procedures to allow comparison between official definitions of procedures and actual operator behaviour.
Validation & Verification	
Method:	Qualitative evaluation to determine if the technique allows comparing a sequence of user actions (also simulated) with an official definition.
Metric:	User satisfaction
Success:	Tool has been used in an AdCoS development process and multiple layers of information have been provided. See feedback.

6.1.6 Feedback from AdCoS owners

Feedback on Means-End Modelling from Philips:

Means-end Modelling has been used for AdCoS ‘Guided Patient Positioning’ in WP6 – Health domain. It is first of all a methodology although also tooling has been provided to create a means-end modelling based analysis.

Pros:

- Addresses multiple layers in a task analysis: from goals and objectives to functions behaviour and actions.
- Visualizes different viewpoints like safety and cleanability.
- A creation tool has been added (OmniGraffle-based)
- The creation tool supports export to TMM 1.0 (XML file)

Improvement points:

- Add contextual information in the model
- Improved tooling to support this methodology

6.2 HF Filer (AWI)

6.2.1 Overview

HF Filer provides the following main functionalities:

- Record the results of human factors evaluation activities
- Make the evaluation data accessible from the HF-RTP
- Provide traceability of human factors data

These three points cover some very important elements in the process:

Evaluations – be they formal or informal – represent a structured way to gather feedback on a system design from a human factors standpoint.

Accessibility to evaluation data allows human factors data to be included into the system engineering workflow at the same level as other data (i.e., more technical development information) used in the management of the process. This makes it easier to ensure that human factors issues are handled at an early stage, instead of waiting until the first working prototypes have been produced and presented to stakeholders.

Traceability and versioning of the data makes it possible to link a design decision back to a piece of feedback (an item in an evaluation report). This is a requirement for certification according to some software development and certification standards, in general it will help answer the "why"-questions that are sometimes necessary to understand certain functions of complex systems.

In this description, the generic term “evaluation” is used, which is deliberately generic to cover terminologies used in different industries. Certain certification standards linked to specific industries require validation or verification of functionality, and HF Filer can store such data as well, provided that they can be expressed in a textual format.

HF Filer is intended to solve the problem of integrating HF results into the technical design and development workflow based on traditional software development tool chains. The aim of this is to make the HF considerations, for HF Filer expressed as evaluations, first class citizens in the workflow, together with other life cycle artefacts, such as requirements, issue reports, test results, etc.

HF Filer is needed in the HF-RTP for several AdCoS in HoliDes to capture HF experts' evaluations of especially usability in order to enable linking into systems to track development process and provide traceability for the certification process.

In order to develop HF Filer, HoliDes has provided input for the role in the design and development process as well as the analysis of the requirements for traceability of evaluations for certification, as identified in the deliverables D1.6 and D1.7.

6.2.2 Functionality

6.2.2.1 Connections to other tools

Through links within the HF-RTP instance, various tools for management of product features, user interface design or software architecture can refer to feedback stored in HF Filer to help users understand the reasoning behind a given design decision. Figure 12 illustrates this for a simple case.

HF Filer's support for textual reporting is important for non-formalised evaluation methods, which often rely on textual descriptions of observations. Without the support of OSLC and the HF-RTP (i.e. before HoliDes), the results of these methods would usually be reported in a normal text document format (such as Microsoft Word, PDF, etc.), lacking the structure required to track single issues in the RTP and link them into other tools.

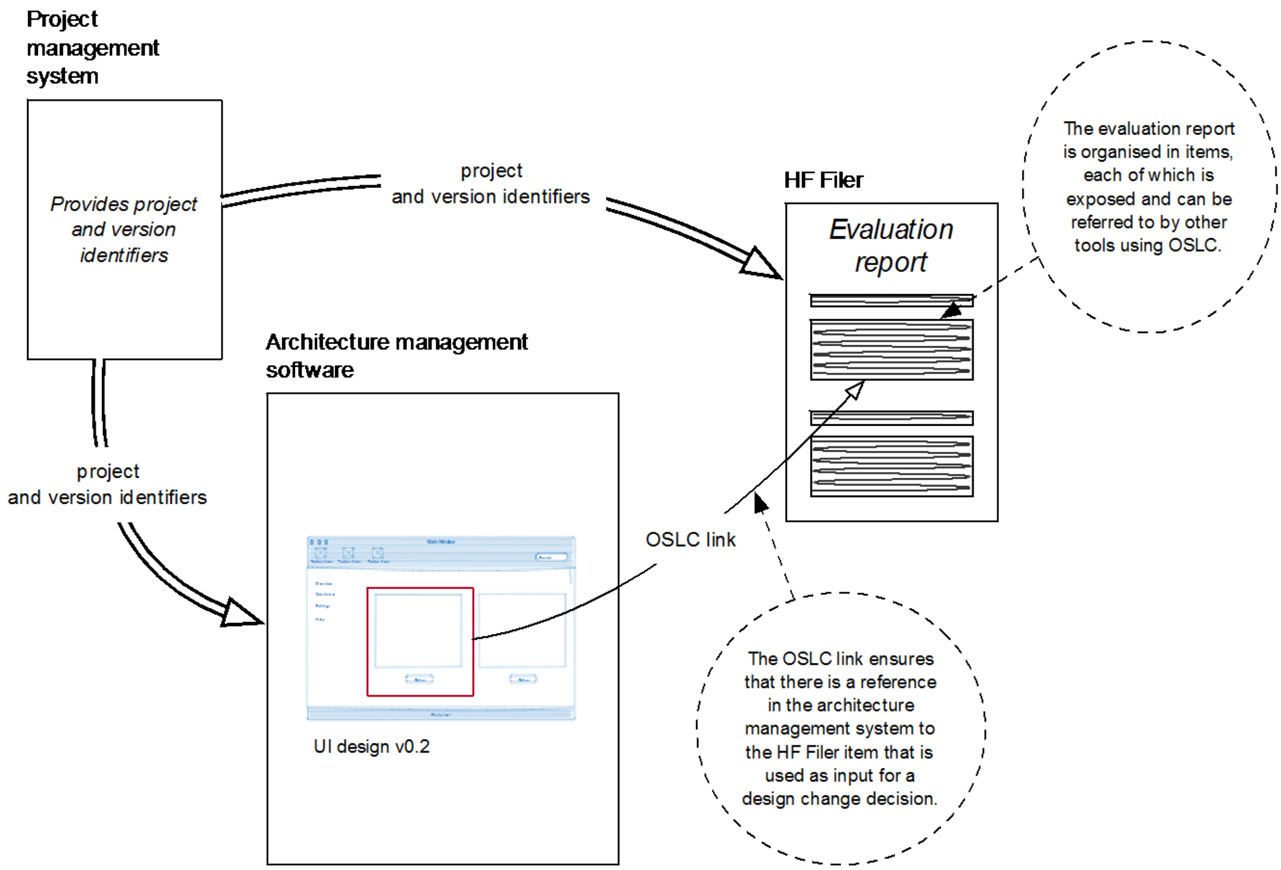


Figure 12: A small RTP instance with HF Filer providing evaluation results.

6.2.2.2 AdCoS Use-Cases

An AdCoS development use case may help understand the intended use of HF Filer.

As stated in the introduction, the main role of HF Filer is to file evaluation reports and make them available for other tools in the HF-RTP, so that the information contained in the reports can be used in a tool chain with project management, software design and similar tools.

To fulfil its role, HF Filer provides two main interfaces:

- A user interface for human factors experts to create evaluation plans and enter reports
- An OSLC interface to let other tools access evaluation reports

A simplified workflow to illustrate the intended use of HF Filer is shown in Figure 13. A point of note is that the actual design and re-design decisions are made by the designer (or other people taking responsibility for this role), based on requirements, evaluations recorded in HF Filer and of course other input from stakeholders. HF Filer does not provide any specific support for design modification; it merely provides the data so it can be used in the process.

The design process leading to version 0.1 and the design revision leading to version 0.2 in the example shown in Figure 13, typically consists of a series of steps, such as sketching, wire framing, production of mock-ups, etc.

HF Filer is *not* intended to directly support this design process, which typically is conducted using dedicated UI design tools, and any connections between evaluation reports and design tasks and decisions must be organised and supported by tools optimised for this in the HF-RTP. Examples of such tools are requirement management tools, issue trackers and product management systems.

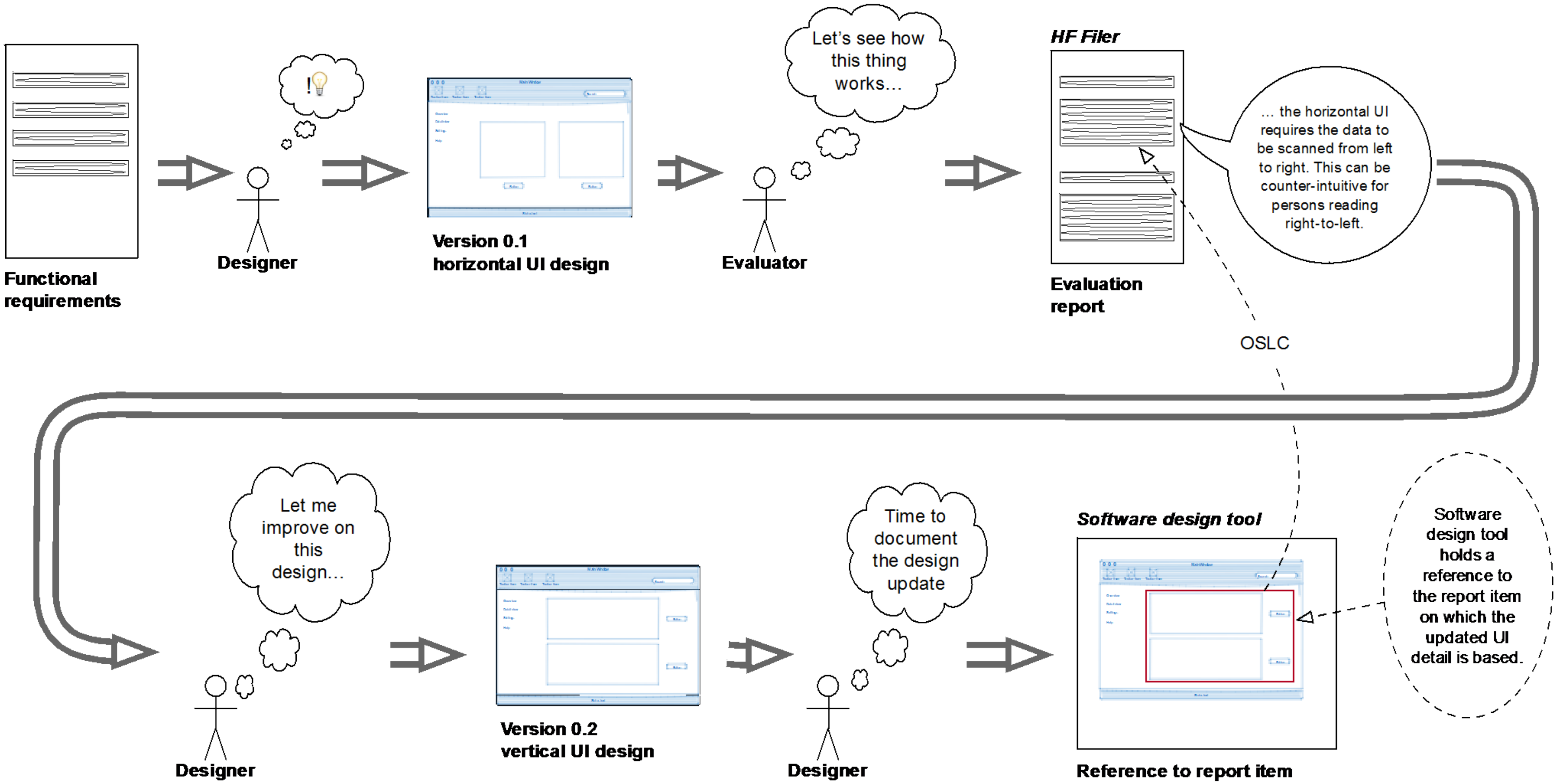


Figure 13: A simple workflow example using HF Filer to record design feedback

6.2.2.3 Information handled by HF Filer

HF Filer handles the evaluation plans and reports as discussed above. The more precise structure of this information is discussed in this section.

By design, HF Filer is a simple tool that can be thought of as a database program to record a list of points to evaluate in a design along with the result of this evaluation for each point in the list. To understand the data format, it helps to look at the purpose of the tool. To re-iterate, the tool aims to do the following:

- Record evaluation plans and reports, possibly based on non-formalised human factors methods.
- Make the evaluation data available for other tools in the RTP.
- Allow traceability and versioning of the evaluation plans and reports.

A real-life design and development workflow will involve revisions of the design of the system or parts of it, and in order to support this, HF Filer supports the evaluation of multiple versions of a design. Since a new version of a design in most cases will just be a modification of the original version, a plan to evaluate a new version of a design could be cloned and modified from the evaluation plan created for the original design. This will avoid duplication of work on the plan for the parts of the design that do not change.

In order to achieve this, a simple organisation of the information has been devised, so that HF Filer allows the following for a given development project:

- The creation of an arbitrary number of evaluation plans
- Each evaluation plan consists of a series of distinct items, each of which can be seen as a specific aspect of the system to evaluate. To the tool, this is just a piece of text, and the semantics are left to the person responsible for the evaluation.
- For each new version of the design, the plan can be cloned and modified if the changes to the design (or new insight into the user's expected operation of the system) require it.
- For each evaluation plan, and for each version of the plan, an evaluation report can be created. It will consist of an evaluation result, in the form of a free text string, for each item in the plan. As for the evaluation plan, a report can be cloned from a previous version and modified as needed in order to avoid having to type old results in again for items that have not changed and for which the evaluation has not changed, either.
- A distinct version identifier distinguishes the different versions of the design under evaluation. This identifier is a string, which by convention is formatted as a major.minor version number, such as 0.1, 0.2, 0.5, 1.0, 1.1, etc. Being a string, it can be prefixed with

letter characters, for instance a "v": v0.1, v0.2, v0.5, v1.0, v1.1, etc.

As the list above shows, the only data in this system that are composed of not free-text input from the human factors experts are the version identifiers used to distinguish the different versions.

6.2.2.4 Version and project identifiers

For the HF Filer to be useful (whether used in an RTP or not), it will have to follow a system for version numbering employed in a consistent way across all tools used in the process, as well as for the documentation produced along the way. In the current version, the task of ensuring this is left to the project manager and the tool does not provide any support to verify the consistency and correctness of the version identifiers.

Being a web-based tool (or intranet-based for organisations with that preference), it is also designed to contain data for more than one project simultaneously. A project identifier is used to keep the data for different projects separate from each other, following normal practice for software development tools. In a fashion similar to how version identifiers are handled, this project identifier is not automatically controlled for correctness or consistency with respect to its use in other tools; the project manager or other users have to ensure that.

The informal approach to the project and version identifiers means that the tool can be implemented and used as a simple stand-alone system. It can store its evaluation plans and reports independently from other tools, and still allow those tools to access the data through the public API. The project and version identifier values are entered into HF Filer by hand, and the project management system (if the project uses one) used to set and distribute these values to the other tools can in reality be paper-based (or kept in a spreadsheet), with the user of the tool making sure they are correct and consistent between the various tools using them.

Figure 14 illustrates how a project identifier can be created and stored, for instance as Post-It notes on a whiteboard in a meeting room. Future versions of HF Filer may support receiving a list of approved project and version identifiers from a central project management system, but in the current version such a mechanism is not supported.

Figure 15 illustrates in a simplified form how these data are entered into the current version of HF Filer as simple text input. As discussed above, there is no specific check of correctness of consistency with the use in other tools in the RTP, meaning that the user in the current version of HF Filer can, of course, also create new identifiers by mistake.

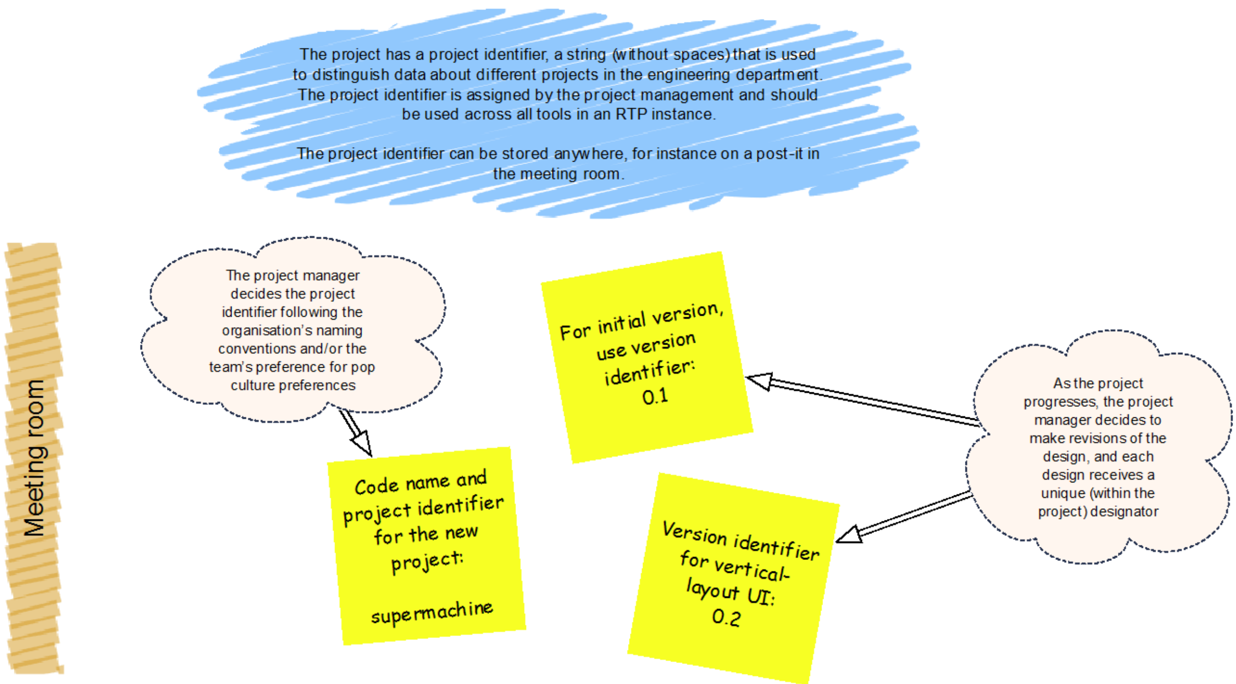


Figure 14: How project and version identifiers can be chosen and stored outside of the tools in the HF-RTP

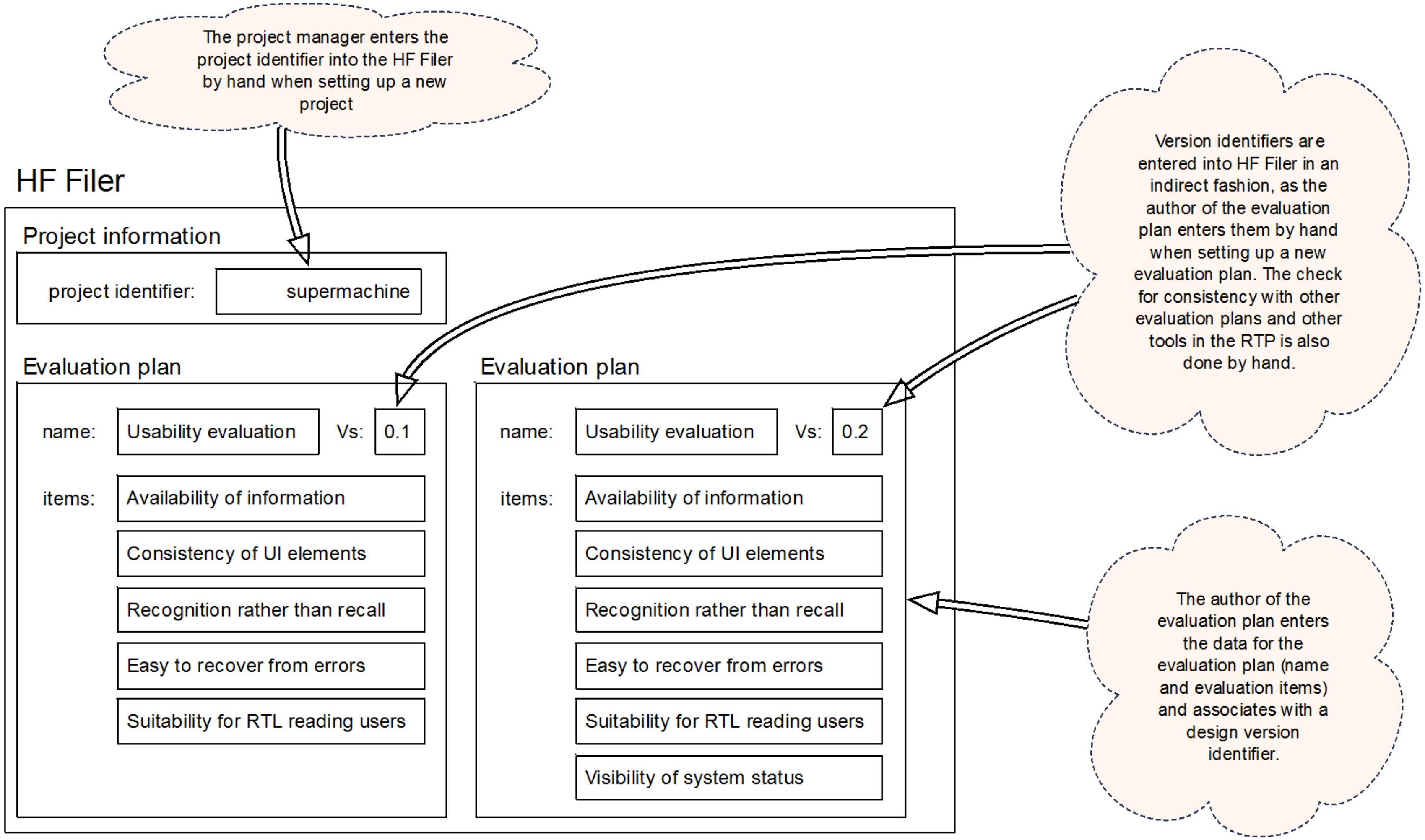


Figure 15: How the external project information (project and version identifiers) is stored in the HF Filer

6.2.2.5 Evaluation plans and reports

Evaluation plans and reports have a simple structure as seen from the perspective of the user and the tools that consume the data produced by HF Filer. In the discussion of the API and interchange of data it is important to note that the cloning of plans and reports discussed earlier only affects the data input through the UI and the internal storage in HF Filer.

The APIs are read-only, and therefore do not allow to store any data or modify any existing data in HF Filer's database. Other tools using the data should not be concerned with how an evaluation plan or report has been created in the database, and data exchanged over the APIs (OSLC and REST-like web interface) do not carry any meta data about the origin of a given piece of information, i.e., whether it is the result of a cloned report or not.

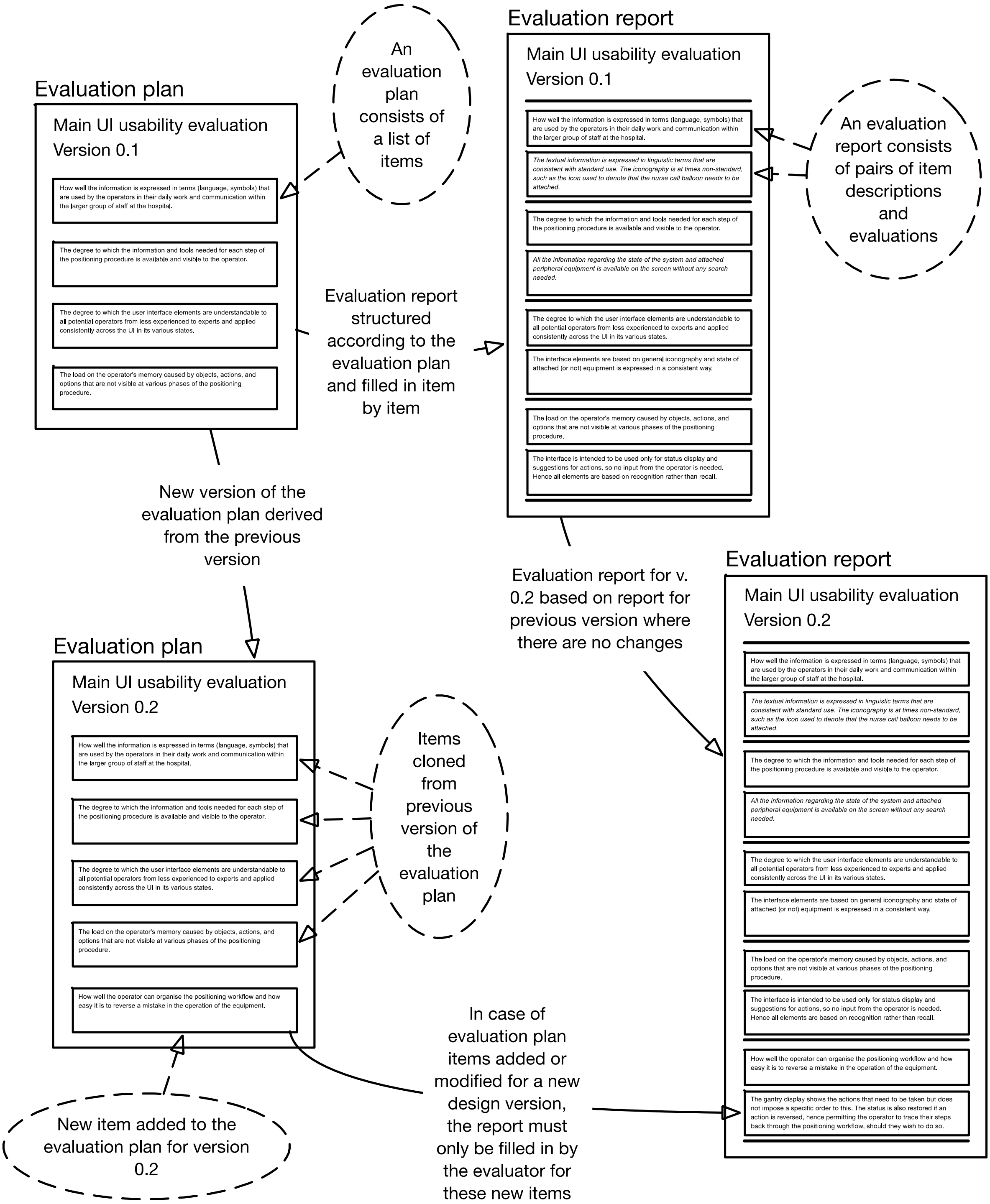


Figure 16: Structure of the evaluation plans and reports from a user perspective (as opposed to HF Filer's internal organisation of the data)

6.2.2.6 HF Filer in the HF-RTP

HF Filer is intended for use in the HF-RTP, with a role as a producer of data. Therefore, this section will only cover the output from the tool, and not go into details of the internal data structures.

All input to the tool is coming from the users (the people entering the evaluation report) through the graphical (web-based) user interface.

6.2.2.6.1 Data structures

HF Filer allows the filing of human factors evaluations for several development projects simultaneously, and for each project several evaluation reports can be created and filed. This work is all undertaken in the web-based user interface (UI) of the tool itself.

Exposure to the HF-RTP through OSLC is read-only, as a means to integrate the evaluation results into the workflow, reporting, etc. There is currently no means to file evaluation results in the tool through OSLC.

HF Filer is organised around simple data structures in two main strands as illustrated in Figure 16 – the evaluation plan and the associated report. It uses a project identifier, which must be shared by other tools in the RTP instance and a version string to identify the various versions of the design in the design and test cycle.

Internally, the tool allows items of the evaluation plan to be inherited from one version to the next, but this is information is not exposed to the OSLC interface. A query for an evaluation plan or report for a given version will produce the relevant information, without distinction of the version of the plan or report that first gave rise to the data.

These considerations lead to a structure of the data exchanged between HF Filer and other tools in the RTP as shown in Figure 17.

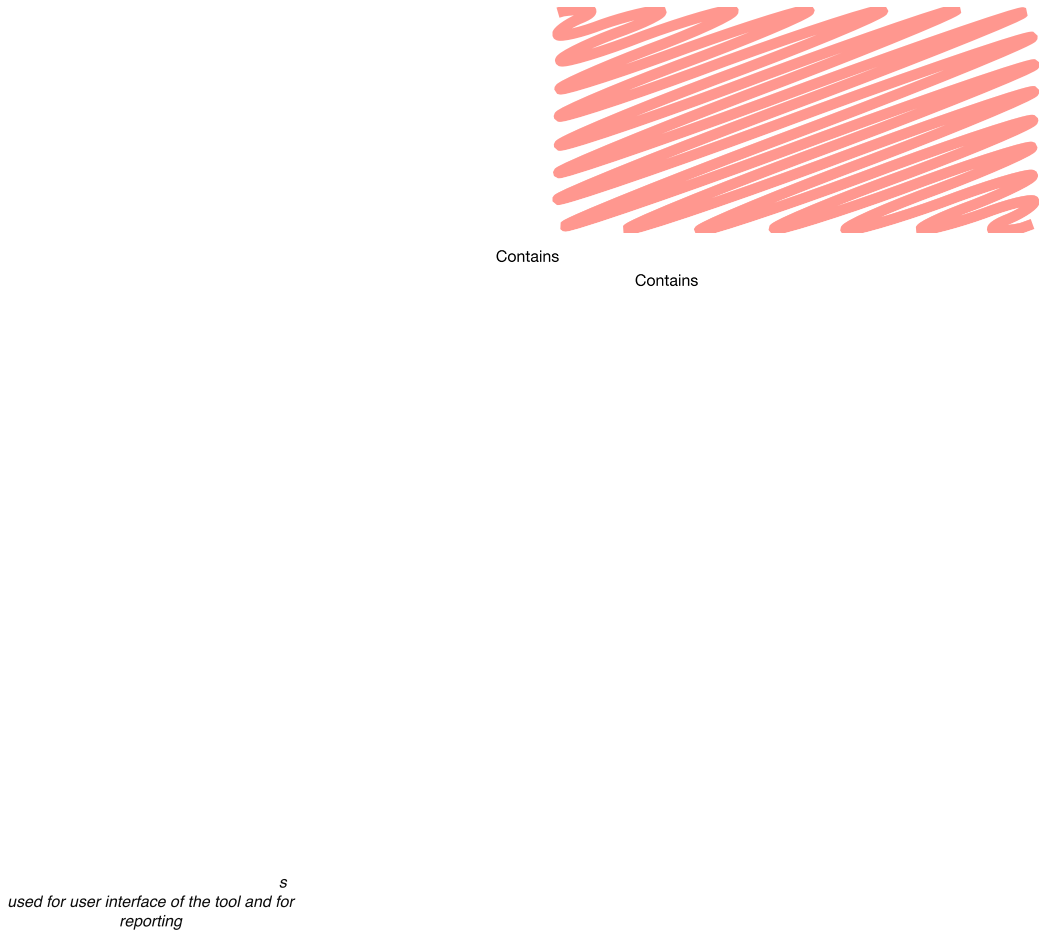


Figure 17: Annotated diagram of the data structure as seen from the HF-RTP.

6.2.2.6.2 Use with the Common Meta Model

When using the Common Meta Model to structure the data in the data exchange between two tools, this structure needs to be specified more formally. To this end, a UML class diagram of the data structure has been devised as can be seen in Figure 18.

The discussion above about the project and version identifiers and how they are used in HF Filer also applies here.

As listed in the table in section 6.1.4, the need for a project identifier to support AdCoS development is tracked in the requirement WP1_HFRTP_REQ32 ("*The HF-RTP shall provide means for transporting project identifiers to distinguish each AdCoS design and development project to be used by all tools in the same RTP instance*"). Similarly, the need for a version identifier is tracked in the requirement WP1_HFRTP_REQ33 ("*The HF-RTP shall provide means for transporting unique version identifiers to distinguish each design and development cycle of the same project with the same RTP instance. The identifier must be available to all tools in the RTP instance that require it*").

Therefore, a project identifier and a version identifier have been added to the class diagram in Figure 18.

As for the general OSLC discussion above, the Common Meta Model does not need to reflect the underlying data structure with its support for cloned plans and reports, and hence only contains the data directly needed to transfer the contents of evaluation plans and reports.

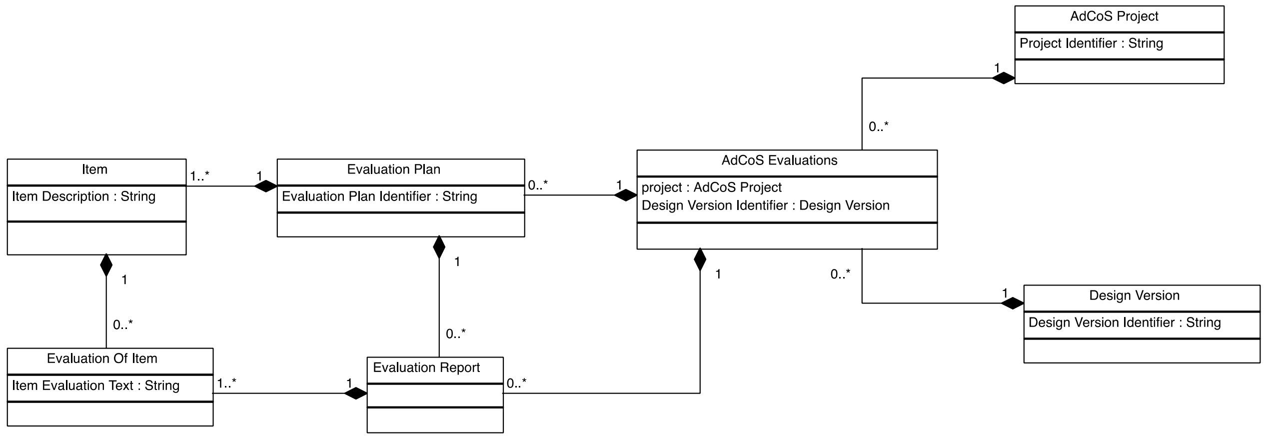


Figure 18: UML diagram of the data structure in line with the Common Meta Model

The diagram shown in Figure 18 only concerns the data needed to transfer evaluation data out of HF Filer. The integration into the Common Meta Model is not considered here.

6.2.2.6.3 Use with a REST-like API and querying interface

When querying the tool, a hierarchical query structure can be used, beginning from the top level, down to the items.

For the evaluation plan, the sequence of identifiers that can be used to access the data in HF Filer (i.e., the "empty form", without any evaluation data filled in) is the following:

```
[AdCoS project identifier]
  >> [Evaluation plan identifier]
    >> "plan"
      >> [Design version identifier]
        >> [Item identifier]
          >> [Item description]
```

As an example, consider a project with the following data:

Identifiers to read HF Filter data		
Data path element	**Identifier**	**Comment**
Project identifier	projectx	Project name
Evaluation plan identifier	data_input_evaluation	Human-readable identifier
Data type identifier	plan *or* report	Fixed strings
Design version identifier	0.1	Version string
Item identifier	i001	Machine-generated identifier
Item description	description	Fixed string

`Item evaluation`	`evaluation`	`Fixed string`

With a REST-like interface to the tool, a query string to access the description of the first item of the Data Input Evaluation plan would look like this:

```
https://hffiler/projectx/data_input_evaluation/plan/0.1/i001/description
```

For the evaluation report, the sequence of identifiers that can be used to access the data in HF Filer (i.e., the "form filled in" with the textual evaluations) is the following:

```
[AdCoS project identifier]
  >> [Evaluation plan identifier]
    >> "report"
      >> [Design version identifier]
        >> [Item identifier]
          >> [Item evaluation]
```

With a REST-like interface to the tool, a query string to access the evaluation of the first item of the plan would look like as follows, in line with the previously shown query string for the plan:

```
https://hffiler/projectx/data_input_evaluation/report/0.1/i001/evaluation
```

6.2.3 Evaluation

Requirement:	**Storage of evaluation plans**
ID:	WP1_HFRTP_REQ34
Ver:	2
Description:	The HF-RTP shall provide MTTs for an efficient storage of itemised evaluation plans. The evaluation plans must be

	incrementally modifiable for different design versions and refer to a unique AdCoS project.
Validation & Verification	
Method:	Evaluation to determine if the tool stores evaluation plans, divided into single items.
Metric:	Checklist
Success:	OK

Requirement:	**Storage of textual evaluation reports**
ID:	WP1_HFRTP_REQ35
Ver:	2
Description:	The HF-RTP shall provide MTTs for an efficient storage of textual evaluation reports based on itemised evaluation plans. The evaluation reports must be traceable to design versions.
Validation & Verification	
Method:	Evaluation to determine if the tool stores evaluation reports with version numbers.
Metric:	Checklist

Success:	OK

Requirement:	**Define procedures**
ID:	WP6_AWI_HEA_REQ14
Ver:	0.2
Description:	The HF-RTP must support descriptions of procedures for human factors verification, training and simulation.
Validation & Verification	
Method:	Evaluation to determine if the tool allows storing descriptions of procedures in an instance of the HF-RTP.
Metric:	Checklist
Success:	OK

Requirement:	**Reporting on experiment results**
ID:	WP6_UMC_HEA_REQ04
Ver:	0.1

Description:	The MTT should be able to collect the subjective and objective information of the operator (comments, suggestions, number of attempts to get to the end, etc.)
Validation & Verification	
Method:	Evaluation to determine if the tool stores subjective information about the operator.
Metric:	Checklist
Success:	OK

Requirement:	**Reporting on experiment results**
ID:	WP7_HON_AER_REQ88
Ver:	0.1
Description:	The MTT should be able to collect the subjective and objective information of the experiments.
Validation & Verification	
Method:	Evaluation to determine if the tool stores the subjective and objective information of experiments.
Metric:	Checklist

Success:	OK

Requirement:	**Reporting on experiment results**
ID:	WP8_ADS_CTR_REQ28
Ver:	0.1
Description:	The MTT should facilitate the collection of the experiment data.
Validation & Verification	
Method:	Evaluation to determine if the tool collects experimental data.
Metric:	Checklist
Success:	OK

Requirement:	**Reporting on experiment results**
ID:	WP9_CRF_AUT_REQ31
Ver:	0.1

Description:	The MTT should be able to collect the design of the experiment as well as the subjective and objective information of the driver (comments, suggestions, number of attempts to complete the task, etc.).
Validation & Verification	
Method:	Evaluation to determine if the tool stores design of experiment and subjective and objective information.
Metric:	Checklist
Success:	OK

6.2.4 Feedback from AdCoS owners

6.2.4.1 HF Filer Feedback from Airbus

Overall, Airbus found the HF Filer to be a very satisfactory tool. The user interface is very simple to understand and possess a shallow learning curve. The ability to create evaluation plans and evaluation items was easy to do and required no formal training. The online web browser basis on which the tool is based made integration into the Airbus design process a quick procedure. Typically Installation of software in high security firms such as Airbus can be problematic due and not limited to:

- The need of formal approval,
- Administration rights for installation
- Firewalls and Proxy issues

With the HF Filer, none of these issues were encountered.

Furthermore, the OSLC support offered by the tool meant that integration into the Airbus tool chain was very quick requiring only a small effort to create a client add-on for the modelling tool Enterprise Architect.

There were only one or two bugs found in the application during trials where evaluation plans were displayed via OSLC for all projects, not just the one in question. This bug was quickly remedied with no impact on Airbus business units.

To enhance the usefulness of the tool, Airbus would like to make the following points.

- The ability to delete Evaluation Plans through the GUI would be useful.
- The ability to delete Evaluation Items through the GUI would be useful.
- More information on the Available project evaluations screen would be useful since currently there is only the name. A description, date and author column would increase the usability of the application.

6.2.4.2 HF Filer Feedback from Philips

HF Filer has been used for AdCoS 'Guided Patient Positioning' in WP6 – Health domain.

Pros:

- Provides a structured way of filing user test results
- Can read pre-defined structure from TMM 1.0 (XML file)
- Web-based, hence easily accessible

Improvement points:

- Limited functionality
- May profit form interfacing to other (industry-standard) document archiving systems

6.3 Operator state detection from implicit hand gestures (BUT/HON)

6.3.1 Overview

In our solution, we aim at implicit gestures performed in aeronautic cockpit. In particular, we study pilot's implicit gestures connected with controlling of selected important cockpit elements: Yoke, Touch screen, Navigation control panel, Throttle lever, Electronic flight bag. We define three levels of interaction with a particular element: Full interaction, Touch-and-Go and Unfinished.

For the purpose of implicit gesture recognition, we proposed methods for detection of transitions between phases of implicit gestures described in previous deliverable. This section describes this solution and its usage in relevant AdCoS.

6.3.2 Functionality

The most important parts of all implicit gestures are the moments of transition between the phases of the gestures. They are key points defining temporal span of the interactions – moments when the pilot started and stopped interacting with the cockpit element or, in the case of unfinished interaction, the moments when the pilot's hand stopped and started to return.

We suppose that each implicit gesture consists of sequence of pilot's poses in time. Therefore, we estimate his pose in each frame and from trajectories of the poses in time we estimate these key moments of implicit gestures. The simple solution based on hand detection in Region of Interest (ROI) would not work in case of unfinished interaction. Also, when the pilot's hand leaves the ROI during full interaction, it does not have to mean that the interaction is finished.

For the purpose of human pose estimation, we tried to use three state-of-the-art methods: Pose Machine (PM), Joint Regressors with Pictorial structure (PS) and human pose estimation by deep neural networks (DNN). We recognize 10 most important joints of upper body: right and left hand, right and left wrist, right and left elbow, right and left shoulder, shoulder centre and head. Results achieved by Pose Machine (PM) were the best as it is shown in section 6.2.3.

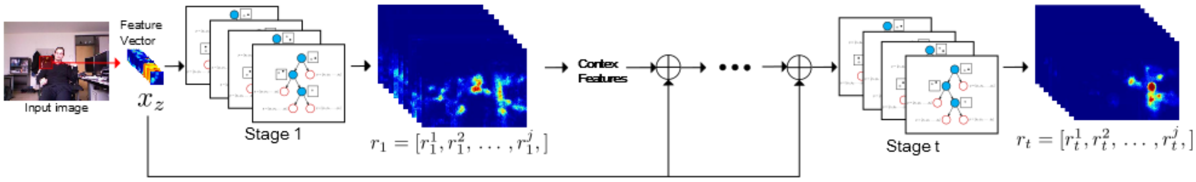


Figure 19: Scheme of the Pose Machine principle.

The basic idea of the Pose machine is presented in Figure 19. Pose machine consists of multiple stages, where multiclass random forest is used as model of each stage t. This multiclass random forest recognizes j classes where $j = 10$ joints of upper body.

Input to the stage $t + 1$ are context features computed from responses $r_t = r_t^1, \ldots, r_t^j$ of stage t together with the original patch features x_z computed for image patch at position z.

This context information from previous stages helps to learn relationships among body parts more precisely and leads to improved results in pose estimation accuracy. There are two types of context features. The first type of context features is patch of responses r_t at z position added as new feature channels to the original set x_z. The second type of context features are offset vectors from z position to the K highest peaks in each joint probability map.

However, estimation of human pose by Pose Machine is not always correct. There are a lot of cases where Pose machine fails. Example of such as wrong estimations is shown in Figure 2. For compensation of this problem, we propose system, whose scheme of principle is shown in Figure 3. This system does not take in account during recognition process only trajectories of the best localized pose joints, but it employs a Random Forest to learn what trajectories of what local maxima in probability maps are important for recognition of recognized implicit gestures. At first this system estimates human upper pose (HPE) by Pose Machine for each of n chosen frames $f, f + k, \ldots, f + (n - 1)k, f + nk$ where k is a step used to select frames from video sequence. It results in joint probability maps $r_i = r_i^1, \ldots, r_i^j$ for each frame i, where $i \in \{1, \ldots, n\}$ and j is index of body part joint. In the second step, each probability map r_i^j is processed by max-pooling operator (MAX POOL). Having an joint probability map r_i^j, this operator computes local maximum value $m\_r_i^j(x) = \max_{p \in \Omega(x)} r_i^j(p)$ for each location p, where $\Omega(x)$ is a local surroundings of position x, e.g. a 13×13 pixels block. Such a max-response image $m\_r_i^j$ is subsampled to achieve resulting max-pooled joint probability map s_i^j, e.g. 7×7 times. Resulting max-pooled probability maps s_i of all frames are fused into a single feature vector $x = [s_1, \dots, s_n]$ by concatenation. In the last stage of proposed system, random forest uses this concatenated feature vector as input for recognition of implicit gesture.

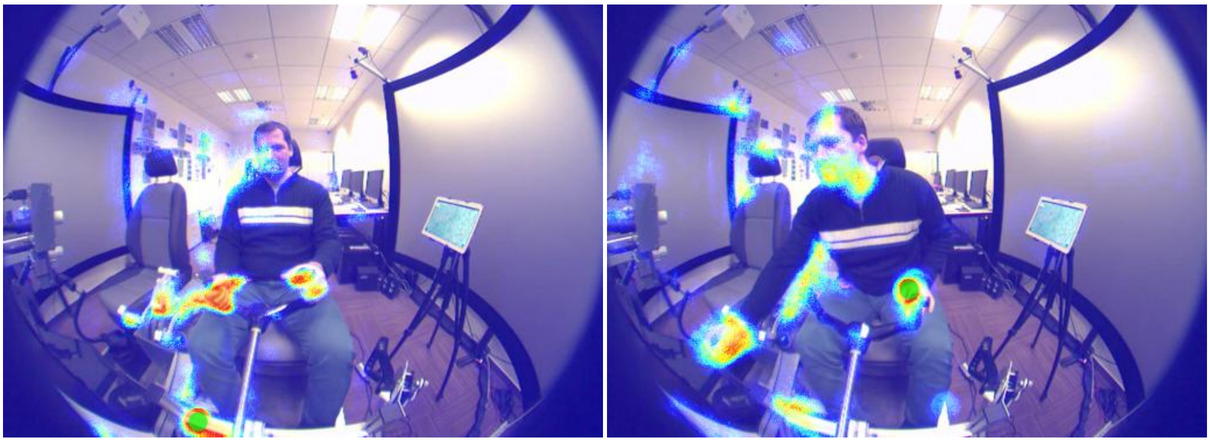


Figure 20: Frames with incorrectly estimated human poses. Global maximum is marked by green colour. Left: right hand. Right: right wrist.

A binary random forest is trained for each transition of each implicit gesture. Set of positive training samples contains sequences of frames with the transition frame at selected position. Set of negative training samples is extracted from other parts of videos.

We used two types of split rules. The first split rule $s_k^i(q) < \tau$ compares a value at location q in the chosen pooled map of joint i of frame k against a given threshold.

The second split rule $s_k^i(q) - s_l^i(q) < \tau$ compares two values from the same joint map i at the same position q of different frames k and l.

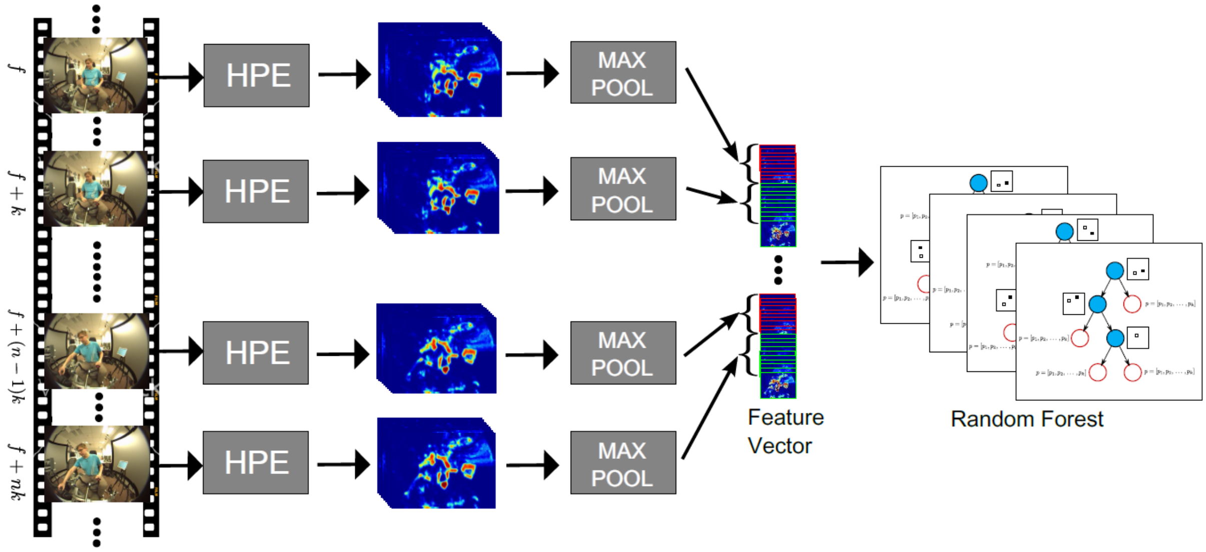


Figure 21: Scheme of proposed approach for implicit gesture recognition.

6.3.3 Evaluation

For the training purpose of Human Pose estimation system, a new dataset of human poses was created. Because pilots sit in the cockpit during controlling of aircraft, the dataset contains annotated poses of seated people. For reduction of the time needed to annotate poses manually, we used a process for semi-automatic annotation based on Kinect. Initial pose is done automatically by Kinect and the final pose is corrected manually in a short time. Each pose annotation includes 10 upper joints of body, which are defined by means of the Kinect SDK. Resulting dataset consists of 6213 pose frames of 24 people where only one seated person is presented in each frame. Resolution of frames is 640 x 480 pixels. Recording of this dataset was done in two environments: office and cockpit simulator. Frame poses used in dataset also contains various lighting conditions (dark, backlit by a window, constant office lightning, etc.) and various backgrounds (from white wall to complex background). As it is mentioned above, we experimented with three methods for human pose estimation. Figure 22 shows results achieved by these methods for this described dataset of human poses. Results show that the best result is achieved by Pose Machine.

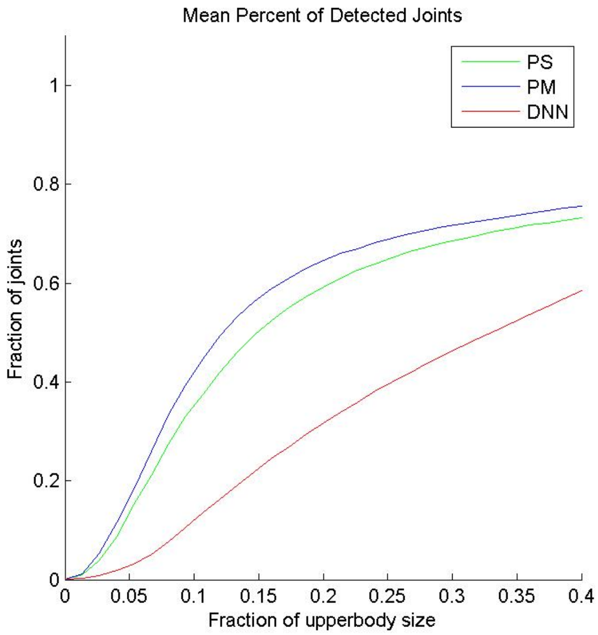


Figure 22: Results achieved for human pose estimation by Joint Regressors with Pictorial structure (PS), Pose Machine (PM) and deep neural networks (DNN).

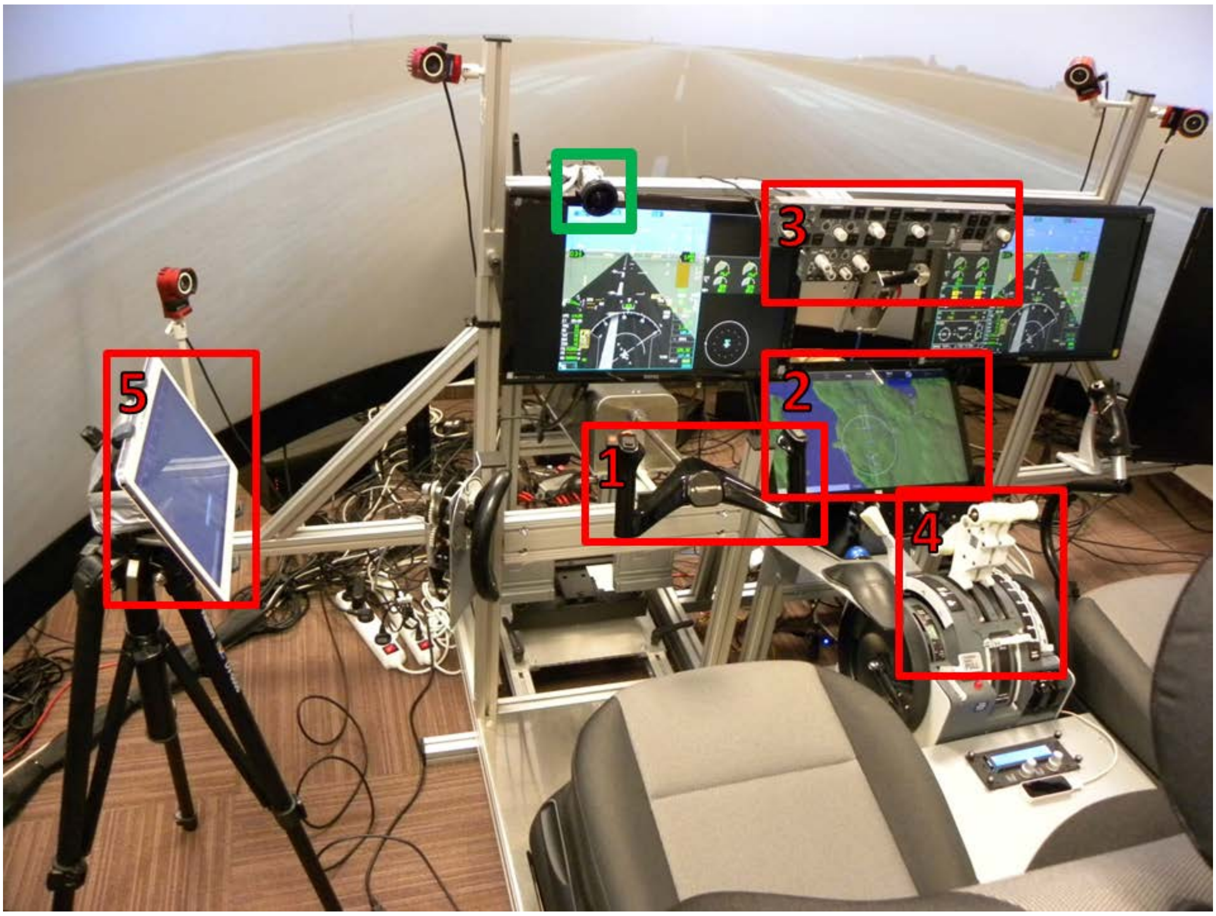


Figure 23: Image of the cockpit simulator and its important cockpit elements marked by red rectangles.

The elements of the cockpit simulator in Figure 23 (as marked by the red rectangles) are:

1. yoke
2. touch screen
3. navigation control panel
4. throttle lever
5. electronic flight bag.

The position of RGB camera is marked by the green rectangle.

Figure 24: Several examples of frames from implicit gestures dataset.

The experiments with implicit gesture recognition were performed on implicit gesture dataset. The recording of the implicit gesture dataset was performed in the cockpit simulator shown in Figure 23. Green rectangle marks recording camera located in front of pilot. Red rectangles in this Figure mark 5 important cockpit elements. There are 12 implicit gestures related to these elements:

	Full int.	Touch-and-go	Unfinished int.
Yoke	✓		
Touch screen	✓	✓	✓
Navigation control panel	✓	✓	✓
Throttle lever	✓		✓
Electronic flight bag)	✓	✓	✓

The implicit gesture dataset consists of videos where these 12 implicit gestures are annotated, but we do not distinguish active/passive second phase in full interaction. The final dataset contains 59 video sequences 708 implicit gestures. Examples of frames from these videos are present in Figure 24. These video sequences have 1080x780 pixels resolution and 35 fps. The dataset is divided into training set composed of 24 video sequences recorded with 8 different people and testing set of two people with 6 video sequences.

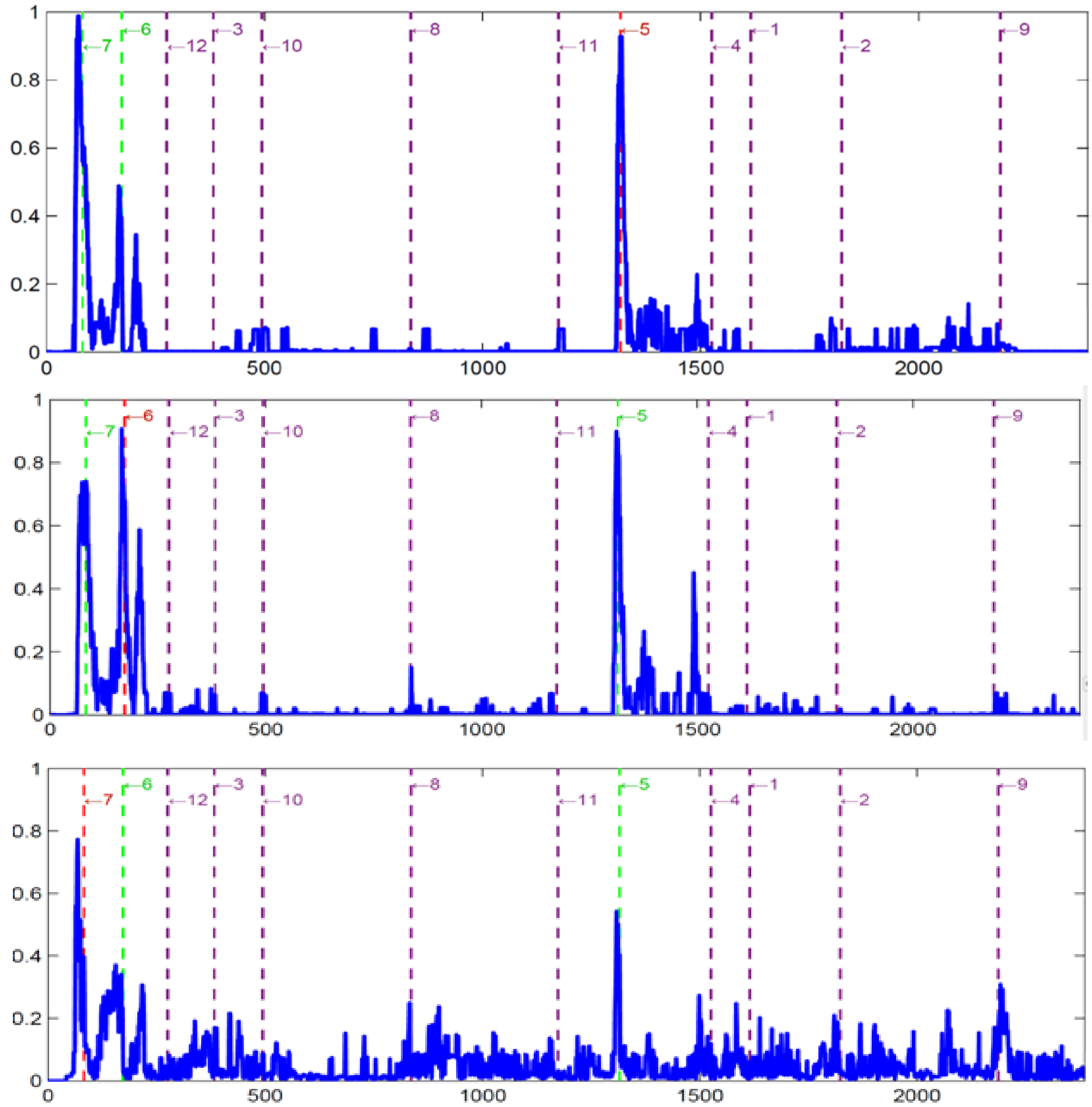


Figure 25: Response of the estimator for the recognition of implicit gestures connected with navigation control panel in time (horizontal axis is time - frame number).

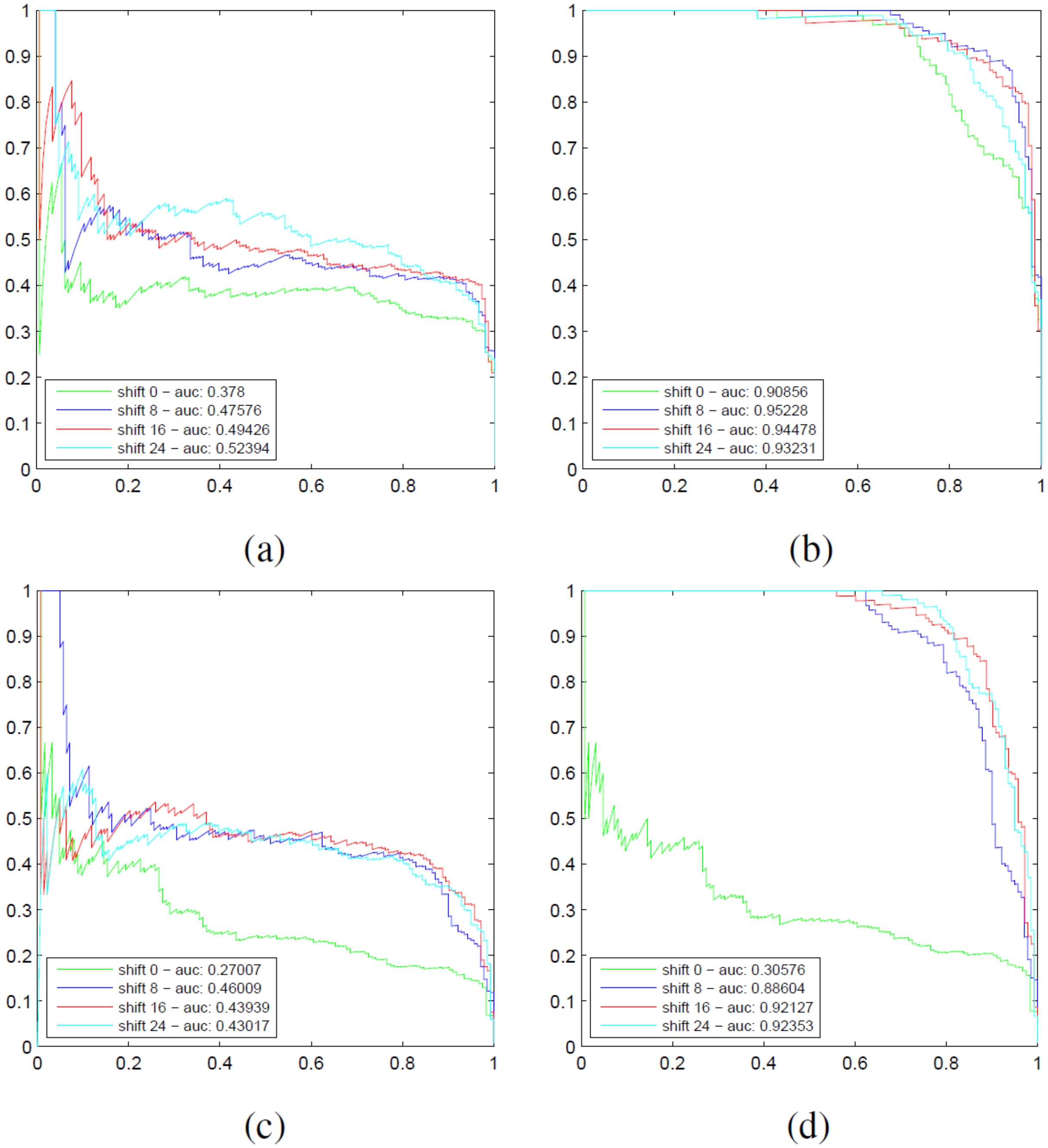


Figure 26: Precision recall curves for transition recognition between phases of implicit gestures for various frame shifts of the moment of transition between implicit gesture phases.

In our experiments, random forests of Pose Machine consists of 15 trees and uses these features: a normalized grey-scale version of the image; the Lab colour space where each colour channel is processed by a min and a max filtration with 5×5 filter size; HOG with 9 bins using a 5×5 cell and soft binning, where each bin of HOG is processed by a max filter. Offsets in context features are computed to $K = 3$ highest peaks of probability maps. The best results were achieved when 3 stages in Pose machine were used.

In our system for implicit gesture recognition, we experimented with different sizes of temporal windows. The best size of the temporal window was $n = 20$ frames (with $k = 0$) for the transition between 1^{st} and 2^{nd} phase of implicit gestures and $n = 25$ frames (with $k = 0$) for the transition between 2^{nd} and 3^{rd} phase of implicit gestures. Max-pooling operation subsampled the probability maps to 47×65 pixels. For each implicit gesture, we trained one binary random forest with 15 trees. For the training of each tree, $1\,000$ positive and $4\,000$ negative frame vectors were used. To increase the number of training samples, we randomly shifted the transition annotations of implicit gestures in time.

The detection performance of our system for implicit gesture recognition is presented for one video in Figure 25. This Figure shows our system's ability to estimate the transition between 1^{st} and 2^{nd} phase of implicit gestures. It consists of three graphs, each one representing one level of interaction with navigation control panel. These graphs present typical results of our system. It shows that the system responses with high probability not only of the correct level of interaction, but to all levels. However, the system produces low probabilities for other parts of the video where other implicit gestures are present. This is also confirmed for all implicit gestures in Figure 26, where Figure 26a (transition between 1^{st} and 2^{nd} phase) and Figure 26c (transition between 2^{nd} and 3^{rd} phase) show results of experiments, where transition moments of implicit gestures related to the same cockpit element were counted as negatives. Figures Figure 26b (transition between 1^{st} and 2^{nd} phase) and Figure 26d (transition between 2^{nd} and 3^{rd} phase) show results of experiments, where transition moments of implicit gestures connected to the same element of cockpit were not counted as negatives, they were ignored.

The interchanging of the levels of interaction for a gesture can be expected, as the 1^{st} as well as 3^{rd} phase of all levels of interaction look almost the same. However, the system's ability to differentiate between different element cockpit gestures is very reliable and stable. Furthermore, we are able to recognize also unfinished gestures even when the hand doesn't reach the given element at all, as shown in Figure 25.

Figure 26 also shows results achieved for various frame delay tolerance in recognition of transition between phases. The best results are achieved for 8 frames $(22.86\,ms)$ delay for transition between the 1^{st} and the 2^{nd} implicit gestures phase and 24 frames $(68.57\,ms)$ delay for transition between the second and the third phase of implicit gestures. However, these results also show very similar detection accuracies for smaller delays. We conclude that gestures can be detected very accurately in time - start of the second phase at single frame precision (auc: 0.909) and end of the second phase with $22.86\,ms$ (auc: 0.886).

Requirement:	**Evaluation of agent action**
ID:	WP7_HON_RTP_REQ78
Ver:	1.0
Description:	Create a tool/methodology that is able to classify an action of agent (human, machine) being either appropriate or erroneous. It is assumed that the tool has a task/procedure model with all supported alternate actions for a given situation.
Validation & Verification	
Method:	Quantitative, by success rate of individual implicit gestures recognition
Metric:	Success rate, False positive rate, False negative rate

Success:	95% Success rate
Comment:	N/A

Requirement:	**Prioritized destinations**
ID:	WP7_HON_AER_REQ89
Ver:	1.0
Description:	The AdCoS should provide the pilot with a list of prioritized destinations according to the mental state of the pilot as well as existing criteria (e.g. fuel calculations, range predictions, weather, airport cost, etc.).
Validation & Verification	
Method:	true/false
Metric:	The prioritized list of destinations must be provided
Success:	output = true
Comment:	N/A

6.3.4 Feedback from AdCoS owners

Implicit hand gestures are correlated with the cognitive state of the human operator. Given the three types of gestures,

- Full interaction
- Touch-and-go
- Unfinished

and their dynamic parameters such as duration, frequency and focus, the implicit hand gestures can be used to assess fatigue, e.g. longer and less precise interaction, distraction, e.g. more frequent touch-and-go/unfinished interaction, or workload of the operator, e.g. fast interaction.

The applicability of the method depends only on quality of hands' recognition in a video data that is unobtrusively recorded in variable illumination. This makes the method suitable for pilots in the aircraft cockpit as it does not influence them in their tasks.

Based on evaluation of the method in HoliDes project, the method fulfils requirements on accuracy of hand recognition even in cockpit environment.

6.4 Detection of operator's head orientation (BUT/HON)

6.4.1 Overview

Orienting attention towards new locations is normally accompanied by reorientation of the head direction. When interacting with an AdCoS, the operator may orient towards other locations in the work environment (e.g. a navigation system in a car) which can indicate distraction from the main task. Thus, automatically detecting these head movements provides valuable information about the operator's current focus of attention and possible distraction. Videos of the operator's head (pilot, driver, etc.) during task accomplishment are recorded and computer vision techniques are used to enable automated analysis of the video sequences.

Deriving knowledge about the human operator is valuable in the system validation phase. Despite the limited detection ability of a video recording, the tool can provide valuable information related to operator's visual focus. The applicability of such approach in design phase and real-time use of the tool was evaluated in comparison with traditional methods (eye-tracking, questionnaires).

In addition, this tool was used in real time to detect the likelihood of missing significant information in the environment. Based on the head direction, the elements in the (aeronautic and-or automotive) cockpit can be identified as being or not being in the primary focus. If an element with important information does not get in primary field of view, it is considered as missed and the system should adapt to regain attention.

6.4.2 Functionality

We aim to estimate the head pose in time as yaw, pitch and roll. Our system for head pose estimation is based on Random forests. In each node, we use split function $x(k_1) - x(k_2) < \tau$, where k_1 and k_2 are chosen positions in the feature vector x. In our experiments, we used several different features: pixel intensities, channel features and Gabor features. The best results were achieved with Gabor features.

During the training process, the split rule parameters (k_1 , k_2 and τ) were selected by minimizing expression

$$\frac{\sum_{x_l \in Left}(x_l - \mu_l)^2 + \sum_{x_r \in Right}(x_r - \mu_r)^2}{norm(\mu_l - \mu_r)},$$

where *Left* is the set of yaw, pitch and roll values of the left branch and μ_l is their mean. Similarly for the set Right.

The leaf nodes of the trees contain average values of yaw, pitch and roll of the samples which arrived to the particular leaf. During the testing process, output values of yaw, pitch and roll are computed by averaging over results from all trees.

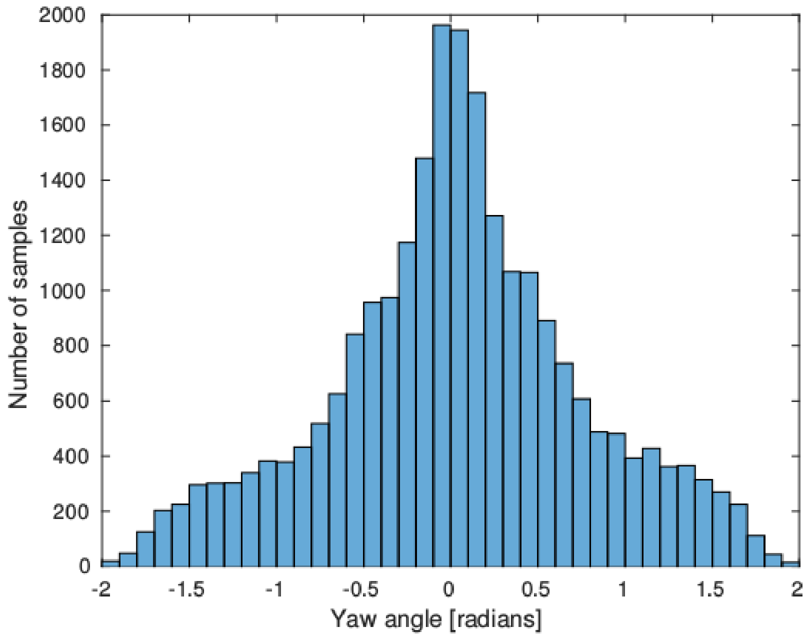


Figure 27: Distribution of the yaw angle of the samples in the AFLW dataset.

Given the distribution of the yaw angle in the AFLW dataset, presented in Figure 27: Distribution of the yaw angle of the samples in the AFLW dataset.

, we propose two other approaches which should improve the results, especially for the near-profile images where the number of training samples is lower.

Our first approach was to increase the number of near-profile training samples for each tree. The training subset for each tree was not chosen randomly from the whole set, but the set was divided into bins with specific range of yaw values and the same number (or less if there was not enough samples in the bin) of random samples was selected from each bin. The range size of each bin was 0.1 radians.

Our second solution uses the fact that the pose estimation needs to be preceded by head localization. We assumed the use of two detectors: frontal, right-profile detector. The left-facing head images can be mirrored. This provides us with very coarse head pose and gives us the opportunity to use a more specific random forest trained on limited range of samples corresponding to the coarse head pose. We propose to use two different random forests (*F* for near-frontal poses, *P* for near-profile poses) that would follow a successful detection by corresponding face detector. The training samples for each random forest were chosen based on their yaw value (|yaw| < 1.2 radians for near-frontal, |yaw| > 0.5 radians for profile faces).

6.4.3 Evaluation

6.4.3.1 Head Pose Estimation Accuracy

For training and testing of the head pose estimator, we used the Annotated Facial Landmarks in the Wild (AFLW) dataset [1]. This dataset contains approximately 24,000 face images obtained from Flickr with annotated facial landmarks and a head pose. This head pose annotation was estimated by fitting the facial landmarks on a head model and is present in the form of yaw, pitch and roll. We were most interested in the yaw (left to right) angle of the head, as it usually varies most. We chose 5,000 images from this dataset for testing, while the rest (approximately 19,000 faces) was used to train the Random Forests.

We tested our three proposed approaches on the testing subset of the AFLW dataset and the results are present in Figure 23 (Y axis in the graph represents the fraction of images and the X axis represents the yaw error. The blue line shows the result for approach where one random forest was trained fully randomly. The red line shows the result for approach where a random forest was trained on samples uniformly distributed over the range of yaw angle. The best results, represented by the yellow line, were achieved by approach where two random forests F (near-frontal poses) and P (near-profile poses) were used.). The results further show that selecting the training samples more uniformly from the range of yaw increases the precision of the estimates. Furthermore, the best results were achieved by evaluating the head images on one of two Random Forests, based on their coarse head pose. Instead of the head location estimated by face detector, which would discard the faces with the difficult images that cannot be detected by regular face detector, we used the annotated head positions in the dataset. To choose the images for evaluation on the correct Random Forest, we divided the dataset based on the annotated yaw angles into two overlapping subsets roughly corresponding to the range of angles from corresponding face detectors.

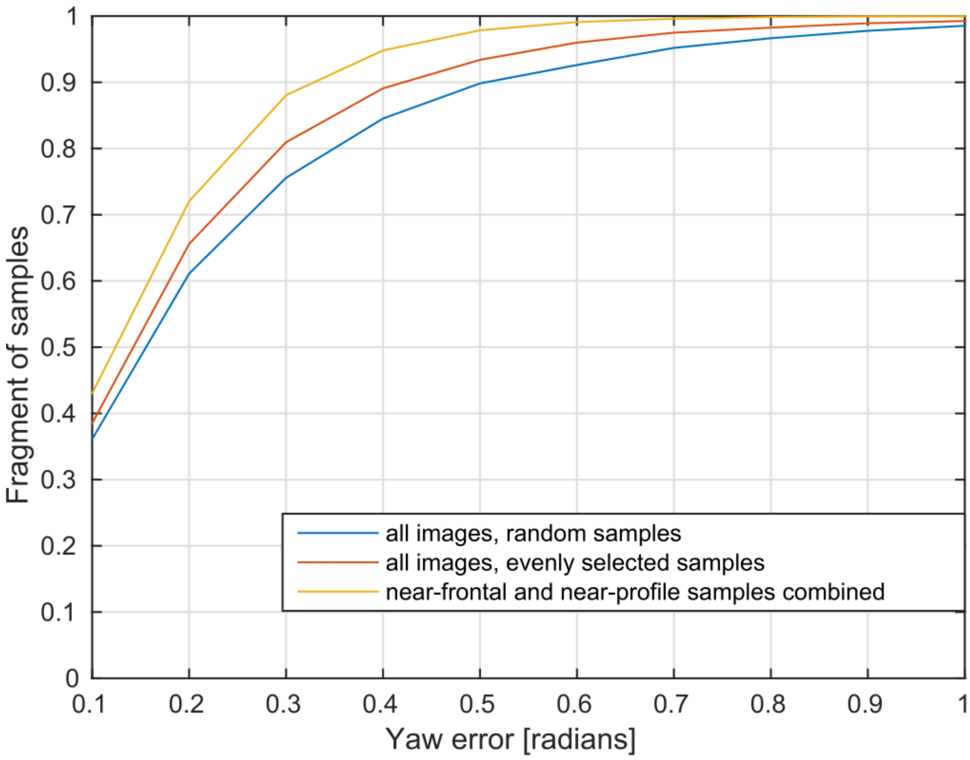


Figure 28: Cumulative histograms representing the error of our three approaches for head pose estimation on AFLW dataset.

Our method computes the head poses with average error of 0.15 radians (8.59°) and 83% of the estimated poses were within 15° distance from the ground truth. Related work by Wibowo et al. [2] reported the average error on the AFLW dataset to be 11.87° while 75.5% of samples met the 15° error threshold. Their results were evaluated only on the faces detected by viola-Jones detector. The method of Sundararajan and Woodard [3] estimated 58.05% of images with error bellow 15°.

To further estimate the precision of the estimates, we created video dataset in simplified aeronautic cockpit simulator. The head pose annotations were created automatically by the OptiTrack system. The pilot had TrackClipPRO attached to his head which emitted infrared light, and five Flex 3 motion tracking cameras were used to estimate ground truth head pose from the infrared markers in real time. The collected dataset consists of several videos captured by a regular video camera with 1920 x 1080 resolution and 50 fps. These videos captured different people seated in front of camera in pilot's seat. During recording, the people were instructed to look around the cockpit to capture various head poses (yaw, pitch and roll).

We tested the proposed algorithm on this video dataset and compared the results to the ground truth from the OptiTrack system. We processed each frame separately. At first, we used cascade face detectors to localize the head. We used frontal and profile detectors and we also mirrored and slightly rotated the frames to maximize the chance of successful detection. The order in which the detectors are triggered is based on the last few successful detections. Once a head is localized, the head image is resized, Gabor features are extracted and used as an input to the corresponding Random Forest. To limit noise and also to approximate the head pose in frames where the head detectors failed produce any results, we optionally use Kalman filter. The estimated yaw angle (without the application of Kalman filter) and ground truth data for one video are presented in Figure 29. The X axis represents frame number while the yaw angle in degrees is on the Y axis. Some errors (e.g. those in frames between 1,000 and 1,500) are due to false detections of the head. This occurs most often in frames where the test subject heavily rotates her head in the roll or pitch angle.

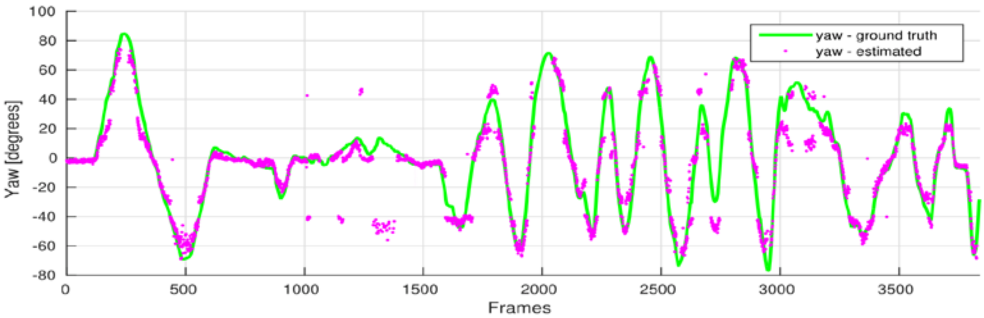


Figure 29: The graph shows the yaw estimation during time for one video example from our OptiTrack dataset. The green line represents the ground truth data from the OptiTrack system and the magenta dots represent the result achieved from our system.

We summarize the results as median errors in the table below. The first row presents the median error for all frames, including those where the face detector failed and the pose was approximated by Kalman filter. In the following two rows are separate results for frames detected by frontal and profile face detector, respectively. For comparison, we included also results where all the estimates were acquired from one Random Forest trained on all samples. 73.4% of frames had error below 15 degrees. If we take into account only the samples where face was successfully detected, the percentage of samples increases to 84%.

	Yaw [radians]	**Pitch [radians]**	**Roll [radians]**
Overall pose error (2RF)	6.10	13.45	3.20
Near-frontal pose error (2RF)	3.08	12.70	2.47
Near-profile pose error (2RF)	8.74	12.54	4.27
Overall pose error (1RF)	7.17	13.42	3.16

Near-frontal pose error (1RF)	3.16	12.62	2.38
Near-profile pose error (1RF)	9.57	13.51	4.43

6.4.3.2 Missed Event Detection Reliability

One of the main reasons for estimating the head pose of a pilot is to determine the area of his visual attention – where did he look? and, more importantly, where did he not look? To simulate this situation, we created another video dataset. Each video was approximately 1 minute long and included one sitting person with a tablet to his right side. The person was instructed to look at the tablet once in a while (5 – 10 times per each session).

Through manual annotation we provided each frame with information whether the person is/is not looking at the display. Then, we compared this information with the results from our system (computed from estimated head pose and known location of the display). The miss rate and false positive rate is given in Figure 30. For example, for angles dot product threshold 0.74, the miss rate was 4.89 % and while the false positive rate was 7.20 %. If we try to limit the annotation error by removing 3 frames at each annotation borderline, the miss rate decreases to 4.76 % and the false positive rate to 5.95 %.

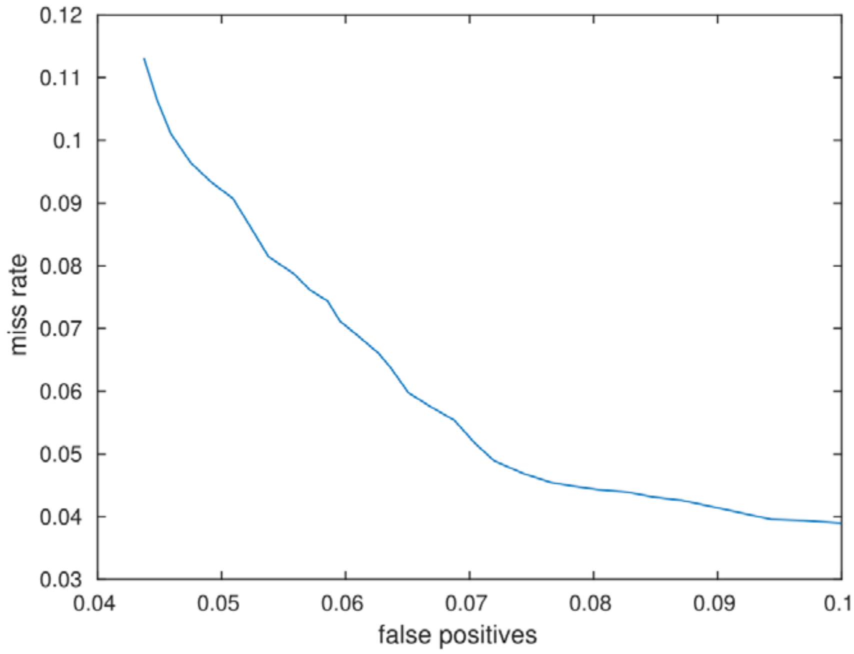


Figure 30: The graph represents correlation between miss rate and false positives rate based on distance from reference vector.

Requirement:	**Eye-tracker strategy**
ID:	WP7_HON_RTP_REQ82
Ver:	1.0
Description:	Compare benefits and disadvantages of using either head-mounted or cockpit mounted eye-tracker in highly unstable environment (cockpit, car). Define best practices/constraint when either of the two is more relevant.
Validation & Verification	

Method:	Weighted costs/benefits of different eye-tracker strategies
Metric:	Weighted costs/benefits
Success:	one technology wins by 10% or more
Comment:	N/A

Requirement:	**Eye-tracker operability**
ID:	WP7_HON_RTP_REQ83
Ver:	1.0
Description:	Investigate strategies of using eye-tracker when the subject needs to - turn head in wide range of angles, - may wear sunglasses or headsets, - undergoes sudden changes in illumination, - may need to change seat, - needs to be monitored for a long period of time.
Validation & Verification	
Method:	Expert judgement
Metric:	outcome of expert judgement
Success:	mode of operation verified and feasible

Comment:	N/A

Requirement:	**Mode confusion**
ID:	WP7_HON_AER_REQ90
Ver:	1.0
Description:	The AdCoS should monitor the head orientation of the pilot to detect if (s)he checked the mode of the aircraft to detect “missed event” situations.
Validation & Verification	
Method:	Quantitative, by success rate of detection of defined behaviour
Metric:	Success rate, False positive rate, False negative rate
Success:	85% Success rate or more
Comment:	N/A

6.4.4 Feedback from AdCoS owners

Operator’s head orientation can be exploited for recognition of inappropriate behaviour in the cockpit – the detection of missed event. This use case reacts to the fact that when a person looks in a certain direction, he/she does not necessarily perceive an object in that direction, but certainly the person cannot perceive object out of that direction. Such object or information can be therefore considered as missed and the information of a missed event can be used to mitigate a potentially hazardous situation.

The detection of head orientation depends only on quality of head recognition in video data that is unobtrusively recorded in variable illumination. This makes the method suitable for pilots in the aircraft cockpit as it does not influence them in their tasks. Based on evaluation of the method in HoliDes project, the method fulfils requirements on accuracy of head recognition in cockpit environment and with respect to angular resolution so that it can be used to differentiate the orientation towards various cockpit displays.

The detection of head orientation was implemented in the Diversion assistant AdCoS to prevent pilots using EFB device when operationally inappropriate. The safety related adaptation of Diversion assistant evaluates head orientation in context of information/situation in the cockpit, i.e. there are two triggers of adaptation.

The system has three parts, the face/gaze detection module, situation assessment module and the mitigation module, see Figure 31. Face and gaze detection is responsible detecting where pilot's head is oriented, i.e. what parts of cockpit are out of his attention.

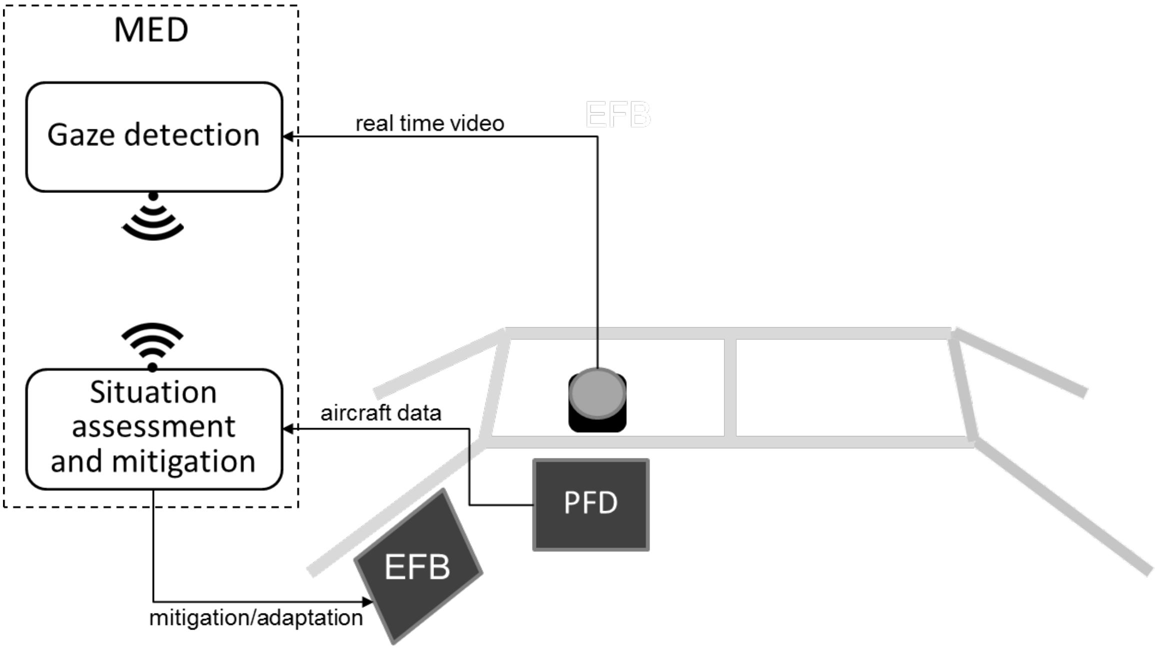


Figure 31: Architecture of Missed Event Detector

Situation assessment monitors the situation in the cockpit using the aircraft data and procedure models. If the situation assessment module recognizes a conflict between where pilot's attention is needed and where it is directed to, it uses the mitigation module to trigger adaptation. This type pilot state is easily comprehensible to the pilot.

The mitigation module is responsible for adaptation of the system in order to alert the pilot about an event that requires his immediate attention. The alert contains information why it was invoked and where pilot's attention is required.

The validation test case focused on enforcing a correct behaviour during an onset of 'sterile cockpit' conditions. Under these conditions, the pilot is required to monitor primary flight display. The test case is especially relevant to Diversion assistant, which is an Electronic Flight bag (EFB) application and EFB is not allowed to be used in 'sterile cockpit' conditions. Three scenarios illustrate the use and applicability of Missed Event Detection

1. Pilot does not use EFB and already monitors Primary Flight Display (PFD) when ‘sterile cockpit’ is set. In such situation system should not perform any action
2. Pilot uses EFB, the system informs him about the conditions and the pilot in response turns to PFD
3. Pilot uses EFB, the system informs him about the conditions, but the pilot ignores the information. Then the system escalates the alert.

As ‘sterile cockpit’ conditions, the tasks of changing the flight level were selected. In that case, the pilot is required to monitor PFD 1000ft before reaching the required altitude. As distraction task, the pilot was required to check availability of airports for diversion using the Diversion assistant application installed on his EFB mounted at the EFB rack aside from the pilot. The scenario was modified and repeated to cover all various situations mentioned in 1-3 above.

After the scenarios were flown, pilots were asked to provide expert opinion on the system, the mitigation alert, and applicability for solving a risky situation.

The performance of the system was assessed as

- Head orientation detection – in all measured configurations, the system correctly recognized when the pilot changed focus from the primary display to the EFB. This result exploits the rather large angular separation of EFB and PFD.
- The delay between movements of the head from one display to another was within 70ms which allows for real-time video processing. The video processing was fluent.

Pilots considered the system easy to understand and acceptable in terms of real use and also regulations. The only negative feedback was with respect to the escalation strategy when the pilot ignores or does not react to the alert. The pilot view is that no system should force the pilot to do something as a system has not complete context awareness. Pilots recommended looking for a less intrusive escalation strategy.

6.5 CPM-GOMS Task Analysis of a lane change for manual and automated driving (DLR)

6.5.1 Overview

To design and evaluate functionality for automated driving which involves the human user, a deep understanding of the driving task itself is urgently needed. Human models, especially cognitive models, can give fine grained insight into specific situations, but are very costly to build and require a large scale effort to achieve a usable outcome. We therefore decided for a task based approach to driver modelling, focusing on constraints of possible driver actions.

We adapted and extended an existing task analysis approach, CPM-GOMS, to achieve this goal. CPM-GOMS (Cognitive, Procedural, Motor - Goals, Operators, Methods and Selection Rules [1], [2]) describes human machine-interaction as a set of operators applied by the user to achieve specific goals. A transfer to the domain of automotive driving required an entirely new set of operators, which have not been described in the academic literature before. It further required new ways to model goals in the driving context, as well as an explicit consideration of the situation surrounding the vehicle in a concrete scenario. With our extended CPM-GOMS for driving, we therefore go far beyond what is currently available for the modelling of the driving task, as well as interactions between the driver and driver assistance and human machine-interfaces.

CPM-GOMS is well suited to describe existing data, such as recordings from vehicles, video data, or eye-tracking data. Further, it is also possible to model situations that have not occurred yet. Modelling existing data abstracts from data of a high granularity towards more high-level representations of driver behaviour. It thus aids the insight into relations between variables, the creation of new hypotheses, and a comprehensive description of data on a high level. Modelling new situations is an indispensable tool to predict workload or loss of situation awareness when operating advanced driver assistance systems.

Within HoliDes, the MTT was developed regarding the Adaptive Automation AdCoS of IAS, specifically for the handover-of-control functionality (see also 5.5). To achieve a seamless transition between the automation stages (from automatic towards manual control), it is crucial to understand the changes in the driving task when automating certain aspects of the driving task. To this end, task modelling with CPM-GOMS was applied to understand consequences of different design alternatives of the handover-of-control.

6.5.2 Functionality

CPM-GOMS can be constructed as a model for a specific human-machine interaction based on empirical data, or to model new, unseen situations. To this end, operators are created, which serve to transfer a current state into a goal state. Such a goal state could be to change lanes, for which a sub goal may be to activate the indicator. To do this, a driver may look at the indicator to find the lever's position (perceptual layer), prepare the hand movement on a cognitive level, and finally move the hand to the indicator. It must then be pressed, which leads to its activation. The clacking sound provides a means for verification that the operation has been successful.

Constructing such a model gives insight into the time course of an action, abstracts fine-granular data such as arm or head movements on a higher level, and provides an approach to connect low-level driving data such as accelerations to high-level driver actions such as eye movements.

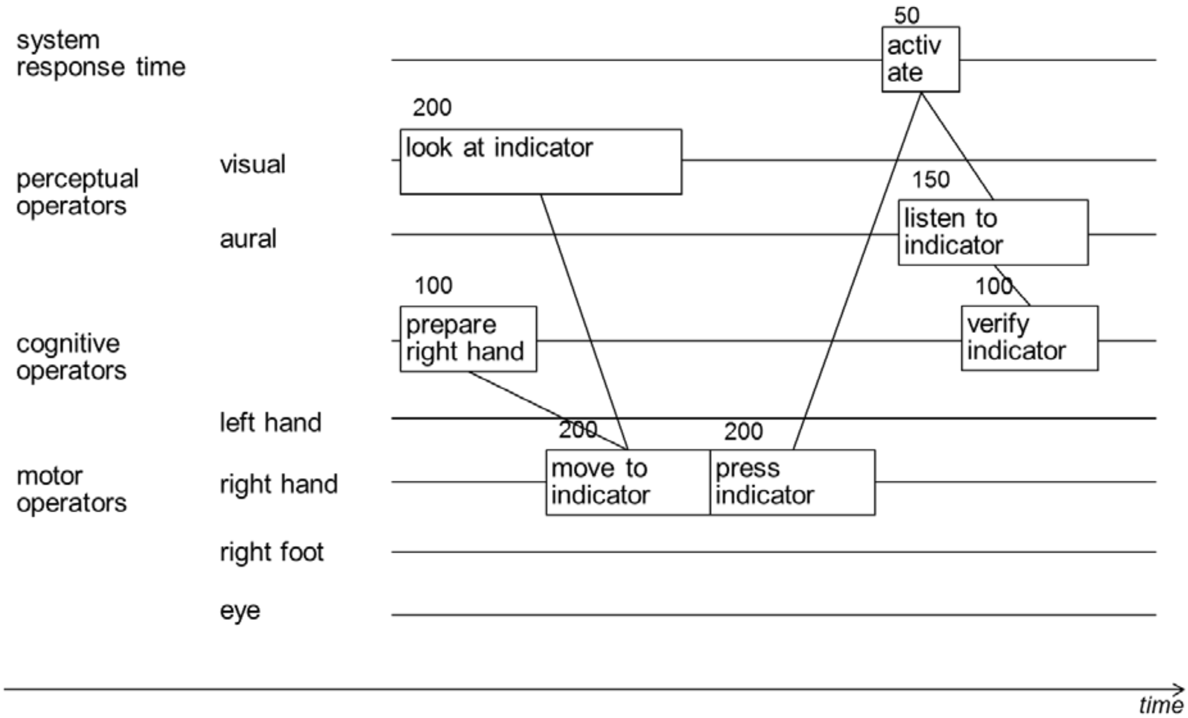


Figure 32: CPM-GOMS example for activating the indicator in a car.

6.5.2.1 Abstraction hierarchy

CPM-GOMS assumes the application of operators towards elements of the system with which the human user is interacting. This has been described so far only within a human-computer-interaction context, but not for the automotive domain. To model the relevant system elements, we constructed an abstraction hierarchy, of which an excerpt is shown in Figure 33. More details about the method can be found in [3]. The purpose of the hierarchy is to explicitly model the parts and pieces that define the driving task, and thus pose constraints for actions a driver or even and automation can take. For instance, at a given moment, it is not absolutely necessary to overtake a slower car in front, but it is absolutely necessary to avoid a collision with this vehicle.

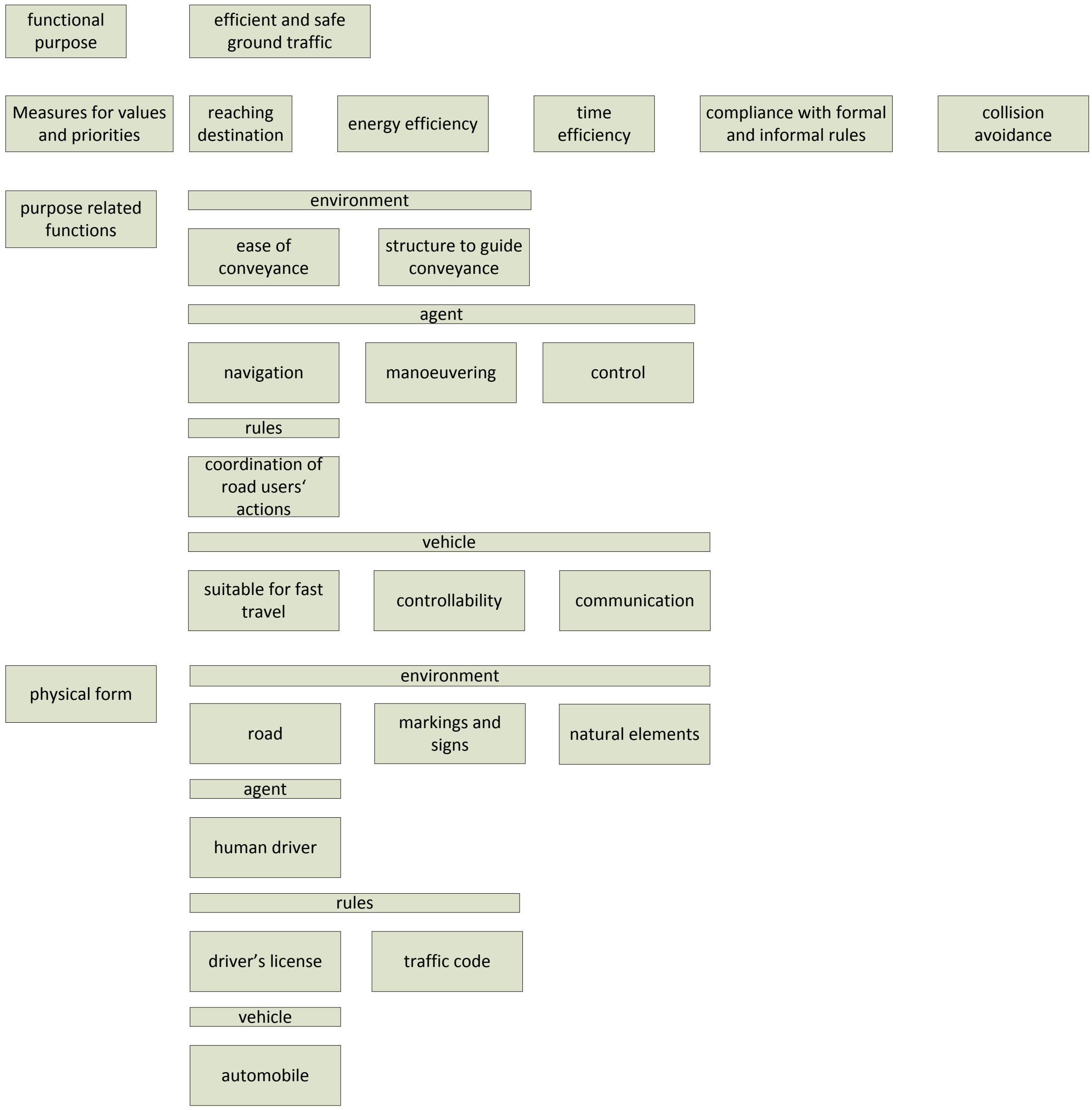


Figure 33: Abstraction hierarchy of ground based traffic on motorways.

6.5.2.2 Construction of the extended CPM-GOMS methodology

To extend and adapt the CPM-GOMS approach, we conducted two empirical studies: Within the first study, data from real driving with a small number of subjects was annotated and analysed. Based on this information and the abstraction hierarchy, we created motor, perceptual and cognitive operators. An example template for motor operators is shown below:

level	resource	operator	system unit (example)
motor	left foot	rest	clutch, floor
		press	clutch
	right foot	rest	throttle, brake, floor
		press	throttle, brake
	left hand	rest	indicator lever
		steer	steering wheel
		change mode	indicator lever
		move	to:from:
	right hand	rest	gear stick, wiper lever
		steer	steering wheel
		change mode	gear, wiper
		move	to:from
	head	point at	left mirror, right mirror, wind shield
		move	to:from:
	eyes	gaze at	dashboard, left mirror, …
perceptual	visual	observe	vehicle, platoon, object in environment
		discover	vehicle, platoon, object in environment
		gauge	vehicle, platoon, object in environment
cognitive		decide	
		judge	
		verify	
		detect signal	
		automatic procedure	

Based on insights from these data, our second study in DLR's driving simulator with 22 subjects was conceptualised and conducted. The controlled environment allowed for developing methods, based on verbal protocols (think aloud) and eye tracking data, to model drivers' goals during lane changes. Additionally, this experiment was used to automatically infer the motor operators with a custom made set of sensors.

Thus, in total our adapted CPM-GOMS method for driving comprises sensors for automatically deriving motor operators and methodologies for acquiring perceptual operators from eye tracking data and cognitive operators using a think aloud technique. With these possibilities at hand, a method for fine-grained task models of human driving can be obtained. In the future, this method will be refined and extensively evaluated. Also, a transfer of the extended CPM-GOMS to other domains such as aeronautics or control rooms appears possible.

6.5.3 Evaluation

Requirement:	**(Cognitive Task) Task Analysis**
ID:	WP1_HFRTP_REQ40_v5
Ver:	1.0
Description:	The HF-RTP shall provide MTTs to elicit information regarding objective task characteristics, cognitive task demands associated with a task, and human factors issues addressed by a task, which is performed by a user in interaction with a technical system.
Validation & Verification	
Method:	Expert judgement
Metric:	Outcome of expert judgement
Success:	yes

Requirement:	**Handover of control only when driver possesses sufficient degree of situation awareness**
ID:	WP5_DLR_AUT_REQ1
Ver:	1.0
Description:	Handover-of-control between car and driver (and vice versa) should only occur when the driver is capable of handling the information, i.e. handover of control leading to human errors and human behaviour leading to automation mode unawareness should be avoided.
Validation & Verification	
Method:	Expert judgement
Metric:	Outcome of expert judgement
Success:	yes

6.5.4 Feedback from AdCoS owners

The following feedback came from the AdCoS owner (IAS) regarding the GOMS modelling task analysis technique:

- "The task analysis technique provides us with the possibility to model the effect that new handover-of-control-concepts have on the driver."
- "With this technique it is now possible to exactly describe how design alternatives affect the workload of the driver on different description levels."
- "The technique supports the decision which design alternative to choose."

6.6 Methods and techniques for the driver adaptive parameterization of a highly automated driving system (DLR)

6.6.1 Overview

A set of driving simulator studies has been conducted in order to tune the CONFORM tool (from WP3) to realize driver-adaptive automation styles. The CONFORM is used in the IAS AdCoS to adapt the driving style of the automation to preference of the driver based on his/her previous way of driving. The experiments that were run in the dynamic driving simulator of the DLR (see Figure 34) were used to provide the data base for the machine learning approach implemented in the CONFORM tool.

Figure 34: DLR dynamic driving simulator that was used for the experiments.

6.6.2 Functionality

A pre-study and an evaluation study were run which are extensively described in D9.9 (see Figure 7 in this deliverable for the workflow). In the pre-study, 42 participants drove on a 2-lane highway with a fixed speed of 120 km/h and had to overtake slower cars (100 km/h) either without or with traffic on the left lane (these cars were going at 140 or 160 km/h). Driving parameters were recorded and used as input for CONFORM to initially estimate whether determining driving styles from these data is possible and to find similar and different driving styles for each participant. Then, an evaluation study with 35 participants (a subsample of the pre-study) was conducted. Here, participants were automatically driven in their own driving style via a replay, a driving style determined by CONFORM (similar driving style), and a driving style determined by CONFORM to be different than their own and had to rate which was their preferred style. The results of the studies with respect to the IAS AdCoS are described in D9.9.

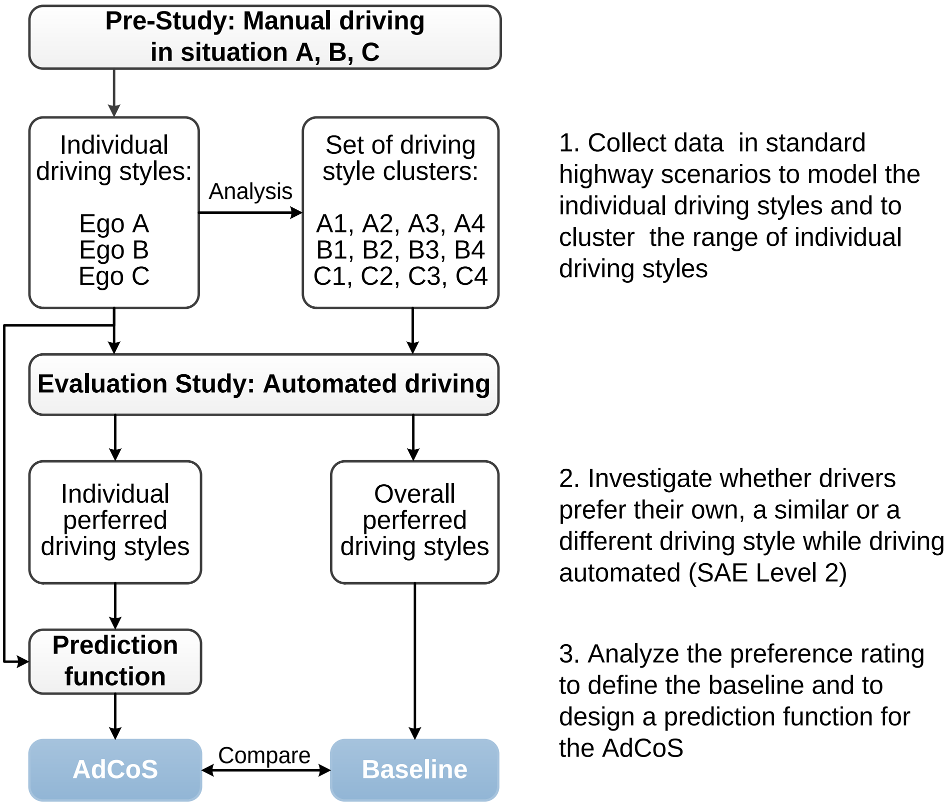


Figure 35: Flow of how the experimental studies were used to tune and evaluate CONFORM (also see D9.9).

6.6.3 Evaluation

Requirement:	**Learning of individual driving behaviour**
ID:	WP9_DLR_AUT_REQ01
Ver:	1.0
Description:	After several manual driven overtaking manoeuvres the driver model has learnt the natural driving behaviour of the driver.
Validation & Verification	
Method:	Calculation of a confidence value (range between zero and one) based on the observations
Metric:	Confidence value
Success:	> .9

Requirement:	**Online learning**
ID:	WP9_DLR_AUT_REQ02
Ver:	1.0
Description:	The driver model shall improve stepwise over several overtaking manoeuvres its current knowledge of the driver by considering inputs by the driver (steering angle, brake pedal position, throttle position) while driving manually. The driver model then updates its manoeuvre preferences.
Validation & Verification	
Method:	Expert judgement
Metric:	outcome of expert judgement
Success:	yes

6.6.4 Feedback from AdCoS owners

The following feedbacks came from the AdCoS owner (IAS) regarding this MTT:

- "The experimental data helped a lot to improve the automated driving styles we can offer to the driver."
- "Based on experimental data gathered and the tool Conform from WP3, we were able to tune the driving style of the automation to fit the preference of the drivers and increase their user experience."

6.7 Surrogate Reference Task (SuRT) for inducing driver distraction (DLR)

The final version of the SuRT can be found in D5.4, so that most of the information described here can already be found there.

6.7.1 Overview

Studying driver distraction and designing assistance systems to counteract requires research tools that can be utilized to reliably induce distraction in simulator and real traffic studies. Here we present a touch screen version of the Surrogate Task (SuRT) as a tool that can be applied in the automotive context and other domains as a secondary task to induce visual and manual distraction. The SuRT is specified in the ISO/TS 14198.

6.7.2 Functionality

The SuRT is fully functional and has been used in the use case of TAK and CRF.

The SuRT is a distracting secondary task consisting of visually and manually demanding parts. Participants are presented with a set of stimuli on a touch screen (e.g. a tablet or a smart phone) which can be mounted on the right side of the steering wheel in reach of the driver's right arm. The stimuli on the screen consist of a target in a set of distractors. Whenever the target stimulus is presented, the participant is asked to touch the half of the screen containing the target as fast as possible with the right arm. In this way visual (looking at the screen) and manual (touching the screen) distraction are induced. With respect to metrics, participants' performance in this secondary task can be evaluated in terms of error rate (percentage/amount of incorrect responses) and reaction times (time between target presentation and response.

The appearance of target and distractors can be set with the parameters shape (circle, square, cross), colour (RGB), radius and line width. Another parameter that needs specification is the time between two tasks, i.e. the latency of presentation of a new target after a participant's response. Moreover, the background colour of the two halves can be specified. Figure 36 shows an example appearance of the SuRT with a large pink circle as target and small violet circles as distractors. In this configuration the participant would have to touch the left half of the screen.

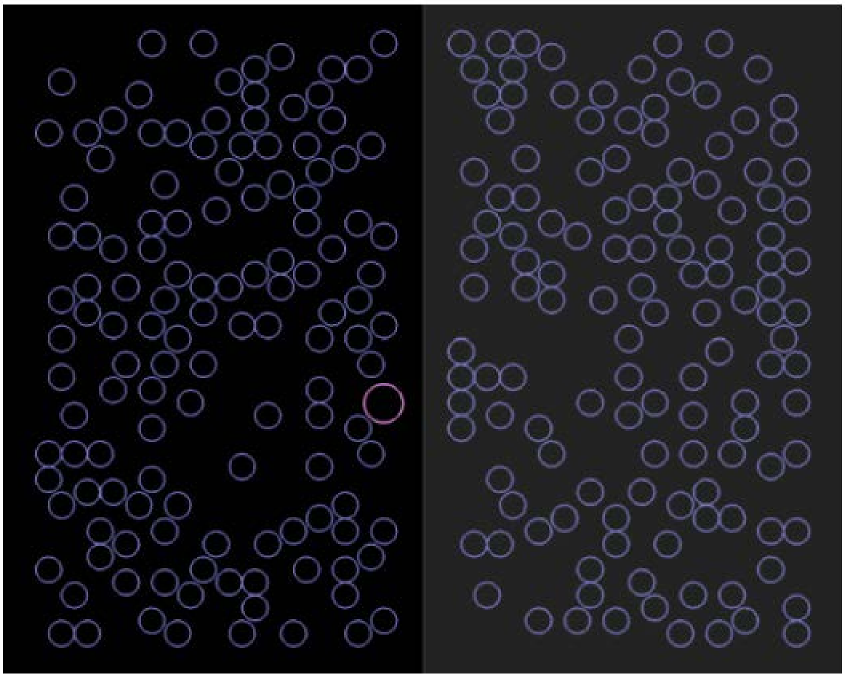


Figure 36: Example of SuRT appearance.

6.7.3 Evaluation

Requirement:	**Tool to induce driver distraction**
ID:	WP9_CRF_AUT_REQ32
Ver:	0.1
Description:	The MTT should involve real drivers to collect information for the creation of the distraction model
Validation & Verification	
Method:	Expert judgment: Does the SuRT offer to run studies with real participants on distraction?
Metric:	Yes/No
Success:	Yes, studies with real participants have been conducted by TAK and CRF with the SuRT. SuRT worked well to induce distraction.

Requirement:	**Data for distraction model is derived in validated distraction paradigm**
ID:	WP5_DLR_AUT_REQ3
Ver:	1.0
Description:	The data that is used to train the classifier for the distraction model of the AdCoS was collected using a validated and well-established secondary task for inducing distraction.
Validation & Verification	
Method:	comparison with norms
Metric:	Distraction task is described in a norm
Success:	Yes, ISO/TS 14198.

Requirement:	**Data for distraction model is derived in real traffic.**
ID:	WP5_DLR_AUT_REQ4
Ver:	1.0
Description:	The data that is used to train the classifier for the distraction model of the AdCoS is collected in real traffic and not in the simulator.
Validation & Verification	
Method:	setup in real car
Metric:	Yes / no
Success:	Yes, the setup in the real car has worked for the CRF use case.

6.7.4 Feedback from AdCoS owners

The following feedbacks came from the AdCoS owner regarding the SuRT:

Feedback from TAK (Adaptive HMI use case):

- "For the development of the adaptive HMI AdCoS, the SuRT was used in two experiments in a driving simulator with a total of 80 participants. The conditions with SuRT varied between the two experiments. It was used in all four conditions in the first experiment and in three out of five conditions in the second experiment."
- "The SuRT allows the easy implementation of a large variety of conditions (target and non-target colour, size, timing, etc.) via configuration file."

- "Regarding the communication with participants the SuRT was easy to explain and was understood by all participants without problem. Interacting with the SuRT resulted in the expected level of distraction and thus fulfilled the requirements."
- "The data output saved by the SuRT also fulfilled the requirements and could easily be summarized in any form needed."
- "During the experiments the SuRT never produced any technical problems and thus proved to be absolutely reliable."
- "The integration in RTMaps and in the simulation tool SILAB was easily achieved. However, although precompiled libraries are available, the source code is not. This makes a full integration in own programs difficult and requires that the SuRT runs as a standalone window in the background. Thus, a full availability of the source code would be appreciated."

Feedback from CRF (Adaptive Assistance AdCoS):

- "In the Adaptive Assistance AdCoS, the SuRT was used in two experiments on CRF real-car and in REL driving simulator, in order to induce visual distraction in the users, with a twofold goal:
    - to collect data about driver's distraction, then used to develop the related MTT (visual distraction classifier)
    - in the evaluation phase, to create the distractor (i.e. the source of distraction) and thus to assess the adaptivity of the AdCoS"
- "The total number of participants (considering both the experiments) was of around 60 people. The conditions with SuRT varied between the two experiments, it was used in one condition in the first experiment on the road and in two out of five conditions in the second experiment in the simulator."
- "For the rest, we found the same results of TAK that is the SuRT allows the easy implementation of a large variety of conditions (target and non-target colour, size, timing, etc.) via configuration file. Moreover, the SuRT was easy to explain and was understood by all participants without problem. Interacting with the SuRT resulted in the expected level of distraction and thus fulfilled the requirements."
- "The data output saved by the SuRT also fulfilled the requirements and could easily be summarized in any form needed (especially in MATLAB format, used to develop the classifier)."

- “During the experiments, only very few times the SuRT produced technical problems (block and crash of the related program in RT-Maps) and thus proved to be very reliable.”
- The integration in RT-Maps and in the driving simulation tool of REL (always by means of RT-Maps) was FULLY achieved. As TAK pointed out, although precompiled libraries are available, the source code is not. This makes a full integration in different programs difficult and requires that the SuRT runs as a standalone window in the background (and this can create some run-time problems in certain cases). Thus, a full availability of the source code would be appreciated.”

6.8 Theatre Technique (DLR)

6.8.1 Overview

The Theatre technique is a driving simulator containing two seat boxes with coupled steering wheels and pedals (throttle and brake) (see Figure 37). With this, two drivers can control the same simulated vehicle, so that one of them can mimic the functionality of the automation (the “confederate”), while the other can experience the interaction with the automation. This possibility to mimic the automation has the major advantage that innovative functionalities can be tested and experienced early in the product design process with human participants, but without the need to implement software prototypes. In HoliDes, the technique was specifically tailored to the needs of the overtaking manoeuvre use case of WP9 (UC 9.2) of IAS, however, the basic set-up for the Theatre Technique is available at the DLR facilities in Braunschweig, DE, and can in principle be used to mimic any automation and assistance functionalities.

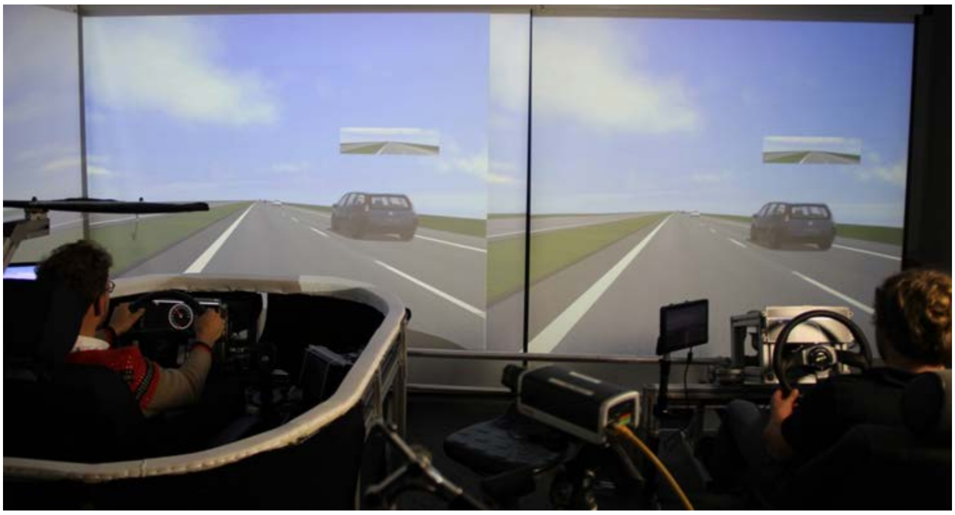


Figure 37: Set-up of driving simulator with Theatre Technique at the DLR facilities in Braunschweig, DE. The left driver is the participant, while the right one is the confederate mimicking the behaviour of the automation.

6.8.2 Functionality

Particularly, the Theatre Technique was used in a two day design session together with IAS in November 2015 with the goal to develop a concept transition design for the hand-over-of-control between driver and automation in various situations on a two lane highway of the IAS overtaking manoeuvre use case.

As preparation, relevant driving simulator scenarios for the considered overtaking manoeuvre needed to be formulated and implemented in the traffic simulation Virtual Test Drive. Moreover, different concepts for the handover of control between driver and automation were formulated. These were then trained by the confederate until he was able to reliably mimic the automation in the scenarios.

Then the participants of the design session tested and experienced the scenarios with different versions of the automation, dismissed inconvenient alternatives and discussed or refined the better ones. Finally, one alternative was selected that has been proposed for the IAS automated vehicle demonstrator (see Figure 38).

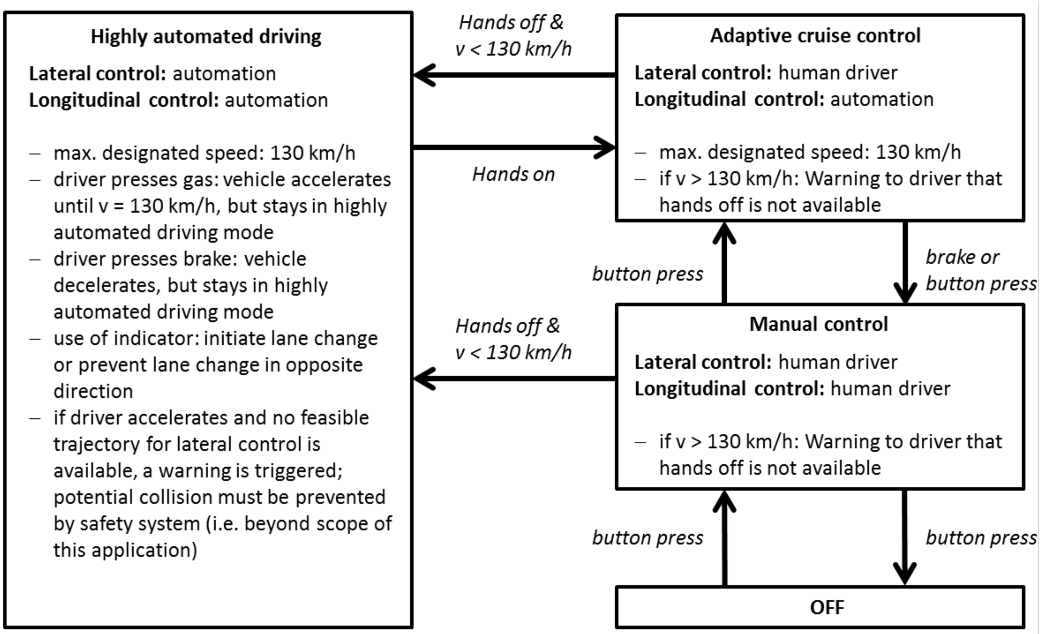


Figure 38: Result of the design session using Theatre Technique is a concept transition design for the handover of control between automated and manual driving. Note: v = speed.

6.8.3 Evaluation

The Theatre Technique has been used to address the requirements mentioned below. However, as it is a design tool, the evaluation is not final, but at a very early stage of the design process.

Requirement:	**Handover of control only when driver possesses sufficient degree of situation awareness**
ID:	WP5_DLR_AUT_REQ2
Ver:	1.0
Description:	Handover-of-control from car to driver should only occur when the driver has enough situation awareness to take over; thus car/HMI has to provide driver with the information.
Validation & Verification	
Method:	Expert judgement
Metric:	Outcome of expert judgement
Success:	Yes. The concept for the handover of control that has been developed in the design session seems to fulfil this requirement; however, here we are still at a very early stage in the design process. So, with the Theatre Technique a final validation of the requirement is not possible as the handover of control strategies are not yet implemented.

Requirement:	**Veh: Long. Control Driver Override**
ID:	WP9_IAS_AUT_REQ06
Ver:	2.1.2
Description:	The driver shall be able to override the automatic longitudinal control at any time. In case the driver applies the brake, the automated system shall turn off for the duration
Validation & Verification	
Method:	Expert judgement
Metric:	Outcome of expert judgement
Success:	Yes. The concept for the handover of control that has been developed in the design session seems to fulfil this requirement; however, here we are still at a very early stage in the design process. So, with the Theatre Technique a final validation of the requirement is not possible as the handover of control strategies are not yet implemented.

Requirement:	**Veh. Lateral Control Driver Override**
ID:	WP9_IAS_AUT_REQ10
Ver:	2.2.2
Description:	The automatic action of the automated system shall not be interrupted in case the driver operates the steering wheel manually, but taken into account by the automated system.
Validation & Verification	
Method:	Expert judgement
Metric:	Outcome of expert judgement
Success:	Yes. The concept for the handover of control that has been developed in the design session seems to fulfil this requirement; however, here we are still at a very early stage in the design process. So, with the Theatre Technique a final validation of the requirement is not possible as the handover of control strategies are not yet implemented.

6.8.4 Feedback from AdCoS-owners

The following feedback came from the AdCoS owner (IAS) regarding the Theatre Technique:

- “The possibility to try out different design alternatives with little technical effort was very helpful to select among different design alternatives“

- “Experiencing how different handover-of-control-concepts felt for the driver was very important for us engineers to better understand the needs of the driver and realize the human factors impact of our decisions”
- “The final concept that we generated in the design session was very sophisticated especially given the little effort it takes to mimic and test it with the Theatre Technique”

6.9 HF-TA: Human Factors Task Analysis (HFC)

6.9.1 Overview

HF-TA [4] is both a method and a tool: as a method, it details the steps of a comprehensive task analysis procedure. As a software tool, it supports this procedure in several ways to make it easier and more efficient.

The development was driven by the need to make task analysis easier to handle, and especially to combine data gathering with further analysis in one tool. As it is able to integrate human factors data from task analysis into system development processes, its purpose served the overall HoliDes goals.

Together with other MTTs, a tool chain was implemented for the workflow in the AdCoS system development in WP6 (see also section 5.1). Data interchange was done by using a common xml format (TMM, task map model, see section 6.2.2) that implemented parts of the HoliDes Common Meta Model. As of August 2016, data can be imported from this format.

The procedure of HF-TA is inspired by mainly Cognitive Task Analysis methods, especially ACTA (Applied Cognitive Task Analysis, [5]). It includes several steps, each of which is supported by and represented in the software. Collecting data (from documents, observations, and interviews), exportable visualisation of the collected input, HF expert analyses, and task modelling are included in one platform.

6.9.2 Functionality

In the HF-TA tool, the 7 steps of the Human Factors Task Analysis procedure are implemented as tabs in the GUI. The task models resulting from these steps are represented both in table and graphical (“task tree”) format. Both representations can be exported (as a *.csv file or image, respectively). Hierarchical, logical and chronological relations between tasks can both be displayed or hidden in the graphical task tree.

Hierarchical task analysis uses “plans” to model execution orders and relations of subtasks with each other. In HF-TA these are implemented with connection nodes which can be nested, such that most real task structures can be modelled.

For automation purposes, it is important to keep track of action and cognition components that are required to perform subtasks using a particular system. It can be annotated in the tool which modalities are used: perception (visual, auditory), higher cognition, or action (hands, speech).

Screenshots are available in deliverable D5.5, Annex II.

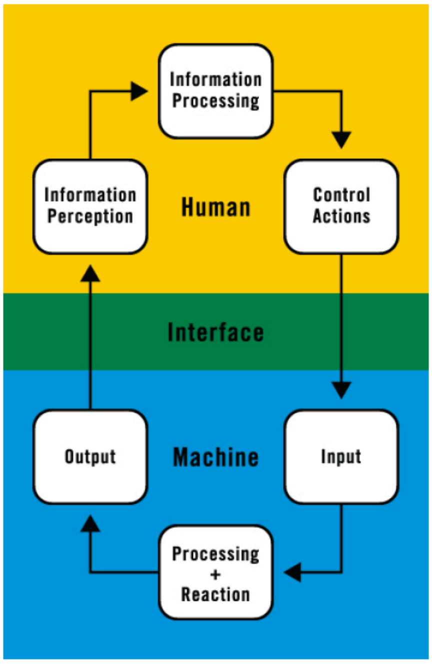
The seven steps of HF-TA:

- **Theoretical HTA** (hierarchical task analysis): Collecting preliminary knowledge (documentation and training materials) of how a task should be done in the view of the system`s designers. In the tool, the hierarchical structure of subtasks can be entered into a table and viewed as a task tree (graphical representation). The two views are synchronised.
- **Observation**: The theoretical HTA tree is compared to an actual user session by annotating a task path and taking notes on differences to the theoretical view.
- **Card Sorting**: A user`s perception of the task structure can be derived by conducting the Card Sorting usability method directly on-screen within the tool.
- **Interview**: The theoretical HTA is compared with a Subject Matter Expert's (SME) perception of (sub)task demands. A set of predefined (but extendable) questions guide the interviewer and answers can be entered directly for every subtask. For example, it can be of interest which subtask requires which skill.
- **Top Level Interview**: Also an interview with guidance through standardized questions, but concerning the overall task, the subject matter expert`s bigger picture of it, as well as expert knowledge and strategies.
- **Human Error Template** (HET, for human error identification): The HET checklist is used by an HF expert to classify and evaluate potential errors concerning each subtask.
- **Task Map Layering:** Automatic generation of time dependencies; time layers ("slices") indicate possible sequences of the subtasks and simultaneities. This can be used e.g. for scenario generation for user tests after the theoretical analysis has been accomplished.

6.9.3 HF-TA: Evaluation

Requirement:	**Task analysis data gathering**
ID:	WP6_HEA_ixR_UC01_3D_acquisition_REQ7
Ver:	0.1
Description:	The tooling supports task analysis procedure and collection of data from: • document analysis • observations • expert interviews • capturing of users` mental models • error analysis The tool combines these in one platform such that they can be further analysed together.
Validation & Verification	
Method:	Success is measured by the ability of the tool to gather data in the specified procedures of task analysis in one or more use cases.
Metric:	completeness of checklist above
Success:	complete (yes)
Comment:	N/A

Requirement:	**Task Modelling**
ID:	WP6_HEA_ixR_UC01_3D_acquisition_REQ6
Ver:	0.1
Description:	The tooling should be able to model the work flow of the tasks with indication of their dependencies and order of execution. It should also capture details of each step of the interaction, like interfaces, the controls and displays involved.
Validation & Verification	
Method:	Check: Ability of the tool to model the abovementioned details for all important steps under specified circumstances (scenario).
Metric:	Successfully modelled sequence of interaction in a given scenario.
Success:	Yes
Comment:	N/A

Requirement:	**Perception, Cognition, and Action (PCA) Requirements of Tasks**
ID:	WP6_HEA_ixR_UC01_3D_acquisition_REQ8
Ver:	0.1
Description:	The tooling should be able to capture what the human operator needs to perceive for a task (sensory input like vision and hearing), whether the task involves higher cognition, and how s/he has to perform the task (manually, speech).  Device User Interface in Operational Context (adapted from Redmill and Rajan, 1997).
Validation & Verification	

Method:	Ability of the tool to capture the abovementioned task requirements.
Metric:	Checklist
Success:	completeness of checklist: yes
Comment:	N/A

6.9.4 Feedback from AdCoS owner (PHI)

HF-TA is used in the healthcare domain of HoliDes, for the VCG Triggering and 3D acquisition use cases (PHI). Feedback for the HF-TA tool was gathered by applying it to selected tasks of the HoliDes WP6 use cases in user sessions and expert evaluations.

Pro's	Improvement points
• Method and Tool in one	• Add contextual information in the model (user / context of use, user needs and goals)
• Addresses many relevant modules for medical industry usability activities like task analysis, test observations and human error template	• Support export to TMM (XML file)
• Extra tooling for optional usability tools like card sorting and layering.	• Optimize tools for use in a (medical) industry environment: • give more freedom / options for the workflow of HF-TA steps (e.g. Integrate HET with observation and interview questions)

	• support more possibilities to create task layering • support root cause analyses
• tablet-based solution that facilitates in-sito use for usability studies	• improved support for observations of multiple participants
• The concept of connection nodes enables more advanced execution patterns (flow) of tasks	• improve usability for touch and pen usage on tablet
• Supports post-test questionnaires (interview and top-level interview)	
• Imports task model from TMM file (XML)	

6.9.5 Further Feedback resulting from benchmark for the IREN AdCoS (WP8, Caterina Calefato)

For the IREN AdCoS, it was evaluated how the HF-TA tool compares to the usual method of manually doing task analysis in tabular format. For the benchmark of methods for task analysis, the following tasks from the IREN AdCoS task analysis were selected:

1. Call receiving	
	1.1. Collecting info about the malfunction (4W: who, what, when, where)
2. Management of routine events	
	2.1. Selection of the most suitable technician 2.2. Deletion of the task from the pending list

All the above mentioned tasks are performed by control room operators. Those tasks were selected because they present some modelling criticalities, hence the way they are solved is interesting, in order to compare different frameworks. For further details about the task analysis for IREN AdCoS please refer to D8.6.

The first criticality is in task 2.1 in step 5: “to refer to task 1.1” that supposes a routine. The second criticality is in task 3.1 in step 6, where the typical condition IF-THEN-ELSE occurs (If the technician accepts the task, go to task 3.2 OR If the technician rejects the task restart task 3.1).

Besides these criticalities, the benchmark has the aim of evaluating how the selected tools support the analyst and the designer in defining the following features:

- Subtask features
- Employed tools
- Subtask description
- Relationships among tasks
- Total amount of steps
- Total amount of steps with a particular feature/tool
- Routines management (ref. first criticality)
- Conditions management (ref. second criticality)

The methodologies analysed in the benchmark are:

- Task analysis by tables
- Task analysis by the HF-TA tool

6.9.5.1 Task analysis by tables

Below, the task analysis performed by tables (for the selected tasks) is presented. To make the document easier to read, tasks were renumbered.

TASK 1 CALL RECEIVING

TASK 1.1 Collecting info about the malfunction (4W: who, what, when, where)				
To ask the user who s/he is	To ask the user what happened	To ask the user when happened	To ask the user where happened	**4 steps**
Manual	*manual*	*manual*	*manual*	
by phone	*by phone*	*by phone*	*by phone*	

TASK 2 MANAGEMENT OF ROUTINE EVENTS

TASK 2.1 Selection of the most suitable technician							
Printing the working shift sheet of all technicians (once a day)	Looking at working shift sheet	Selecting the most suitable technician	Calling technicians by mobile/radio	Providing the technicians with the 4 W information (refer task 1.1)	The technician refers about his own skills in solving the malfunction and its time availability	If the technician accept the task, go to task 3.2 **OR** If the technician rejects the task restart task 3.1	**7 steps**
Manual	*manual*	*manual*	*Manual*	*manual*	*manual*	*manual*	
by PC	*cognitive*	*cognitive*	*by phone*	*by phone*	*by phone*	*cognitive*	

TASK 2.2 Deletion of the task from the pending list			
Receiving a call from the technicians that inform that the malfunction is	Recording the time and the duration of the intervention on the	Deleting the intervention from the pending list	**3 steps**

solved	network		
Manual	*manual*	*manual*	
by phone	*by PC*	*by PC*	

6.9.5.2 Benchmark of the task analysis by HF-TA tool

Below, the task analysis performed by HF-TA tool (for the selected tasks) is presented. Since the HF-TA tool is a tool for Hierarchical Task Analysis, tasks were renumbered, according to HTA-model implemented in HF-TA.

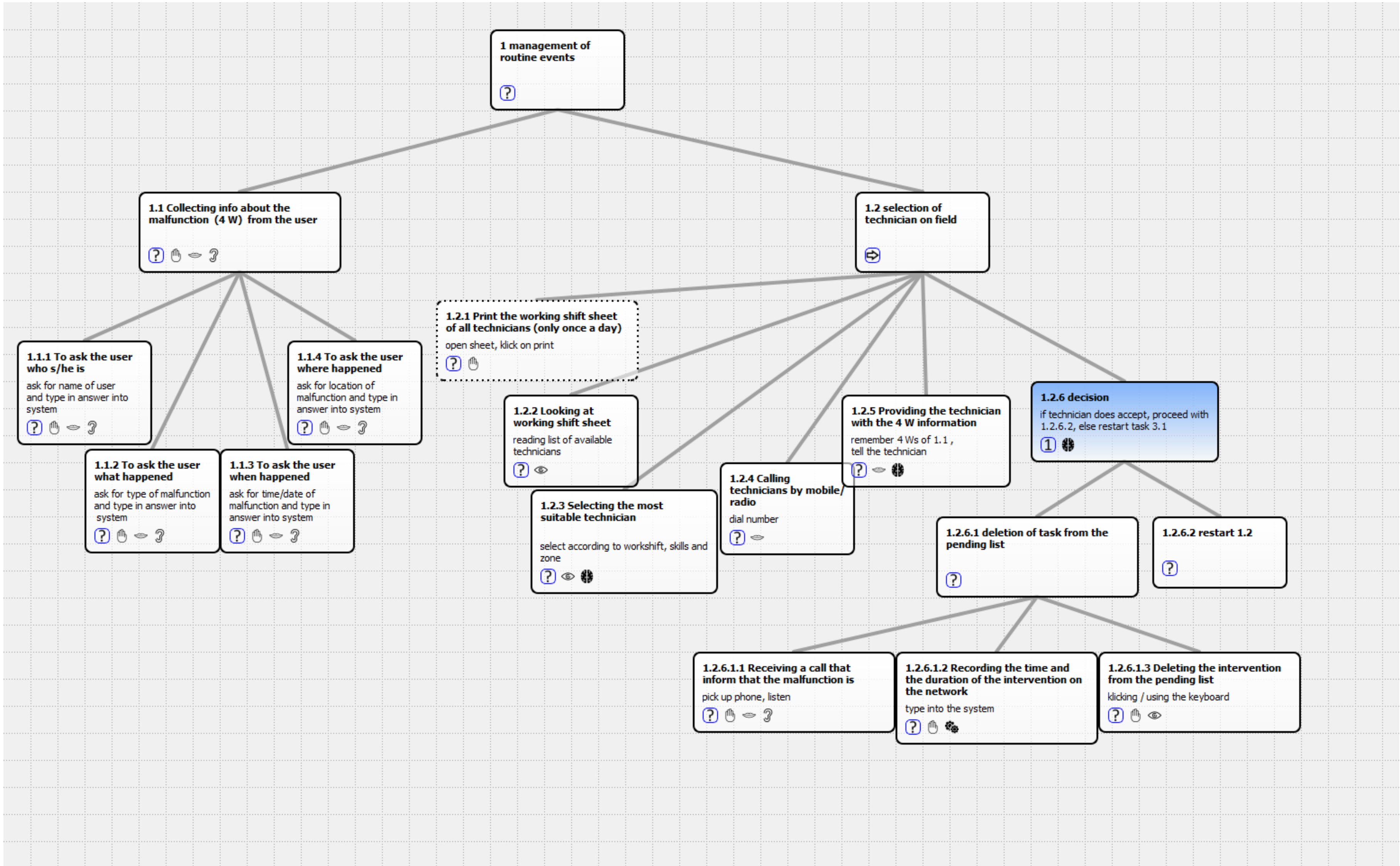


Figure 39: Theoretical Hierarchical Task Analysis of IREN task performed by HF-TA tool: tree chart output

ISSUES	TABLES	HF-TA TOOL
Subtask features	To add a feature it is needed to add a row under the subtask, and in each cell, one should indicate the option among the available alternatives.	There is a list of features that can be selected with a flag, and the corresponding symbols is displayed in the subtask box
Employed tools	To add a tool employed during task performing, it is needed to add a row under the subtask, and in each cell, one should indicate the option among the available alternatives.	There is the form “control/display” in which typing the name of the tool in.
Subtask description	The description is added directly in the cell.	It is possible to define a subtask title and to refine the descriptions using the following forms: “mandatory/optional”, “action”, “system reaction/feedback”, “description”.
Relationships among tasks	Tasks are presented in sequential way; hence, it is possible to elicit just the temporal relationship. For other type of relationships, it is needed to write a note.	It is possible to elicit which tasks are subtasks of other tasks (parent-child relationship). The temporal relationship and execution order is indicated by numeration and by the type of plan.
Total amount of steps	One has to count steps manually	Steps can be counted using a formula on the xls/csv file output.
Total amount of steps with a	One has to count steps	Steps can be counted using a formula on the

particular feature/tool	manually	xls/csv file output.
Routines management	One has to insert a note (i.e. "refer to task xx") in the subtask.	The reference to a routine can be added in the form "action".
Conditions management	One has to insert a note (i.e. "IF-THEN-ELSE to task xx") in the subtask.	Condition can be managed as a normal hierarchy, defining a colour convention for a task named "decision" (i.e. blue, as in task 1.2.6). Then the following subtasks represent the option of the conditions.

6.9.5.3 Further remarks

During the use of the HF-TA tool some comments about its functionalities and features were gathered:

1. Since the aim of this task analysis is to design an automated and cooperative system, namely the AdCoS for IREN Control Room, it is very important to identify which tasks are performed by humans or by the automation. In HF-TA tool the feature "manual" stands for something done by the hands. Hence, it misses a feature that identifies the role human/machine. Since the tool is under development, it is just a matter of adding new labels.
2. In case of modelling conditions and related options (IF-THEN-ELSE), it could be useful to have a symbol that stands for "process ended" (for example a dark dot, as software designers use in algorithm modelling or in Finite State Machine formalism). It can be used also for identifying the end of the whole task.
3. The HF-TA offers to the Human Factor expert a platform for ergonomics analysis: besides the task analysis tool, there are other useful tools to

model human error and to easily record card sorting and interview results. What is missing instead is a tool to perform function allocation. The function allocation method, which is based on task analysis, is one of the most used method in the design of automated and cooperative system. The implementation of a framework for the function allocation method can be taken into consideration for the further developments of HF-TA tool.

6.10 U-DAT (PHI)

6.10.1 Overview

U-DAT (Usability-Data Acquisition Tool) is a tool to structurally capture information for usability activities.

During the development process of an AdCoS typically a number of usability activities have to be performed. Each usability activity will generate its own usability-related data relevant for the subsequent steps in the development process:

1. Detailed user needs and tasks
2. Scenarios
3. Use FMEA - Failure Mode and Effect Analysis based on potential use errors
4. Usability test script
5. Usability test results

Performing these usability activities typically require a lot of manual work. This tool limits the need for manual translation and capturing of data by formalizing the data structures and to the extent possible automation.

HoliDes has been an ideal project to develop this tool as it brings together other usability tooling and applicable domain AdCoS to make sure proper integration with other relevant tools is possible and to test the use of these MTTs beyond a theoretical model in the context of a realistic AdCoS.

6.10.2 Functionality

The screens below illustrate the different modules as stated above.

1. Detailed user needs and tasks

Task Analysis									
Task ID	Context of Use	User Role	Parent	Relation	Task	Step	User Goal (User Story) *As a <user> I can <activity> so that <business value>*	Description	Covered in Scenarios
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									

This module captures all identified user needs / tasks / steps. The Task Map Model – capturing the relationships between different entities – can be imported from the TMM-based XML file. This will populate the blue fields. The green fields represent the information that needs to be added in this module. In this case, detailed information about each entity and the correspondence to scenarios. Some of this data may be automated in the future as well.

2. Scenarios

Use Scenario Nr	Scenario Narrative	Hazardous
1		
2		
3		
4		
5		
6		

Scenarios describe a textual and realistic flow through the task model. A scenario touched upon one or more tasks in the task model. The total set of scenarios should cover all identified tasks at least once.

3. Use FMEA - Failure Mode and Effect Analysis based on potential use errors

Hazard Analysis - Use FMEA										
									Criticality	
Task ID	Task	Step	Failure Mode ID	Hazard	Hazardous Situation	Hazardous Scenario Nr (leading to possible use error / failure mode)	Use error by PCA analysis *what PCA component(s) led to the use error?*	Harm (effect of failure)	Hazard	Essential

Based on the identified tasks, this module provides the template to analyse possible use errors that might lead to harm. Again, the blue parts stem from earlier modules, populated automatically, and the green fields are required to be filled in. If a potential use-error has been identified the corresponding PCA (Perception, Cognition, Action) component(s) are identified that lead to the use-error. Eventually, criticality is indicated for each task. If a task is labelled as critical it requires proper safety-related follow up.

4. Usability test script

Test Script

Scenario 1

					Criticality		Task Score				
Task ID	Task	Step	Expected result / SCRIPT	User Goal	Hazard	Essential	Succes	Close call	Fail	Comments	Root cause analysis (close call / fail)
Scenario 1											
Scenario 2											

Post-test question:	
Post-test question:	
Post-test question:	

The test script is meant to be used for usability validation with end-users. For each relevant scenario to be tested the corresponding tasks that are part of the scenario flow are depicted. This template will be used for each test participant to capture scoring and feedback results.

5. Usability test results

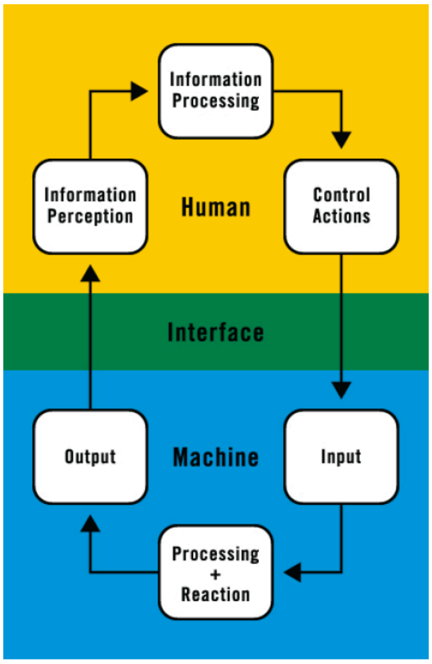
pp no	Task/Step no	Task	Step	task score	Comments	Root cause analysis (close call / fail)	Issue
pp1				success			
pp1				success			
pp1				success			
pp1				success			
pp2				success			
pp2				success			
pp2				success			
pp2				success			
pp3				success			
pp3				success			
pp3				success			
pp3				success			

The test results per participant are combined automatically in this results overview. This module is not for capturing information but to facilitate to filter and analyse results.

6.10.3 U-DAT: Evaluation

Requirement:	**Task Modelling**
ID:	WP6_HEA_ixR_UC01_3D_acquisition_REQ6
Ver:	0.1
Description:	The tooling should be able to model the work flow of the tasks with indication of their dependencies and order of execution. It should also capture details of each step of the interaction, like interfaces, the controls and displays involved.
Validation & Verification	
Method:	Check: Ability of the tool to model the abovementioned details for all important steps under specified circumstances (scenario).
Metric:	Successfully modelled sequence of interaction in a given scenario.
Success:	yes
Comment:	As stated in the previous section the tool enables to capture abovementioned characteristics of identified tasks.

Requirement:	**Task analysis data gathering**
ID:	WP6_HEA_ixR_UC01_3D_acquisition_REQ7
Ver:	0.1
Description:	The tooling supports task analysis procedure and collection of data from: • document analysis • observations • expert interviews • capturing of users` mental models • error analysis The tool combines these in one platform such that they can be further analysed together.
Validation & Verification	
Method:	Success is measured by the ability of the tool to gather data in the specified procedures of task analysis in one or more use cases.
Metric:	completeness of checklist above
Success:	yes
Comment:	As stated in the previous section the tool enables to capture and merge this information in a structured way and be able to further process it in a semi-automated manner.

Requirement:	**Perception, Cognition, and Action (PCA) Requirements of Tasks**
ID:	WP6_HEA_ixR_UC01_3D_acquisition_REQ8
Ver:	0.1
Description:	 Device User Interface in Operational Context (adapted from Redmill and Rajan, 1997). The tooling should be able to capture what the human operator needs to perceive for a task (sensory input like vision and hearing), whether the task involves higher cognition, and how s/he has to perform the task (manually, speech).
Validation & Verification	
Method:	Ability of the tool to capture the abovementioned task

	requirements.
Metric:	checklist
Success:	yes
Comment:	As stated in the previous section, the tool facilitates capturing PCA (Perception, Cognition, Action) based analysis for each relevant task. See module Use FMEA.

6.10.4 Feedback from AdCoS owners

U-DAT has been used for AdCoS Guided patient positioning and ECG Triggering in WP6 – Health domain. As it has been used multiple times in the last 1-2 years U-DAT has been improved based upon feedback and learning from using it in real user studies.

Pros:

- Addresses the different usability stages required in a regulated medical industry.
- Provides structure to the data needed in different stages of the process
- Provides to some extent automation, which saves a lot of time.

Improvement points:

- More automation
- Enhanced connection to other usability tooling, like an updated version of TMM.
- Export possibilities to requirement management tooling

6.11 Empirical Analysis (SNV)

As described in previous deliverables (D5.4; D5.5), Empirical Analysis (hereafter EA) is a technique introduced by SNV in the HoliDes project to the aim to collect real data to be used in different phases of the development process of two AdCoS, DiVA system in Aeronautic domain and Lane Change Assistant system (hereafter LCA) in the Automotive domain.

6.11.1 Overview

In the Aeronautic domain EA has been introduced to collect data about fatigue effects to be used to train the Pilot Pattern Classifier, able to detect fatigue effects in pilots. In the Automotive domain EA has been introduced to detect distraction effects in drivers in a real car to learn a Distraction Classifier and to test the performances of the final version of the LCA System. The great advantage that promoted the introduction of this methodology in the HoliDes project is the possibility to use labelled datasets representative of the real context of interest and to collect information from potential end-users.

6.11.2 Functionality

In the Aeronautic domain EA contributed to the development of the DiVA AdCoS, in particular of the module for the pilot state detection. The Pilot Pattern Classifier is a machine learning tool able to timely recognize the workload level of the pilot on the basis of EEG and eye-tracking information. In the training phase, PPC needed to be fed with labelled datasets providing examples of EEG and eye-tracking data associated with different (and explicitly indicated) workload conditions.

EA was the method used to conduct experiments to collect sufficient amount of data to train the classifier and to estimate its theoretical performance. To this aim SNV developed and previously tested a protocol to detect fatigue effects. Behavioural and psycho-physiological data have been registered and synchronized by means of RTMaps. All the details about this study have been described in D7.9.

Also in the Automotive domain, EA has been employed in the development phase of the LCA system to train the algorithm for distraction detection. To this aim an in-vehicle experimental campaign has been realized. Data about the driving behaviour of the participants and psycho-physiological data, as eyes orientation and head orientation, have been collected and synchronized by means of RTMaps. Further details about this study are available in D9.5.

EA has also been adopted for the evaluation of the LCA system. LCA is a driving assistant system able to help drivers during lane changing manoeuvres presenting them with “keep lane” and “change lane” alerts. The complete version of the system integrates video, audio and haptic warnings. SNV defined the protocol to test the performances of the complete system. To this aim an experiment in a driving simulator has been realized to compare the performances and the subjective judgements of participants driving with the complete LCA system and with a baseline version without haptic support. The results highlighted an enhancement of the safety and subjective appreciation in the complete version of the LCA respect to the baseline. The only drawback for the complete version has been that participants judged it more cognitively demanding respect to the baseline version. To further test the differences between the complete and the baseline version of the LCA a second empirical study focused on HMI has been realized. The detailed description of these studies is included in D9.9 and D9.10.

6.11.3 Evaluation

EA has been inserted in HoliDes to the aim to address HF related requirements. In particular, EA covered the need to investigate the cognitive states of the users of adaptive systems. In addition, EA has been applied to evaluate the performances of the LCA System and to deeply investigate the HMI and the communication strategies between that system and the real user.

Requirement**:**	**Classification of physiological output**
ID:	WP7_HON_RTP_REQ87
Ver:	0.1
Description:	Define methods and tools for classification of measured physiological signal and related level of stress/workload. Do it in real time.
Validation & Verification	
Method:	Experiment
Metric:	Behavioural and psycho-physiological data
Success:	Yes
Comment:	SNV developed the experimental protocol for the data collection, realized the experimental setting, included the sensors synchronization, and performed a statistical analysis of such data to detect if the human behaviour, in term of response times to a stimulus onset and related psycho-physiological data, significantly changes in the different cognitive states. The quality of the real data collected and analysed contributed to the satisfaction of the requirement about the classification of physiological signals in real time.

Requirement:	**Classification of driver's cognitive state**
ID:	WP9_CRF_AUT_REQ3
Ver:	1.0
Description:	The classifier of the driver cognitive state shall be able to do that with a CR > (80÷85)%.
Validation & Verification	
Method:	CR evaluation in the test phase of the classifier
Metric:	CR
Success:	yes
Comment:	The construction of a classifier able to detect the driver cognitive state leverages the data collected and analysed by SNV. SNV developed the experimental protocol for the data collection on the road and performed a statistical analysis of such data to detect if the driver behaviour significantly changes in the different cognitive states. The quality of the real data collected and analysed contributed to the satisfaction of the requirement about the CR of the final classifier.

Requirement:	**Lane Change Assistant (LCA).**
ID:	WP9_CRF_AUT_REQ10
Ver:	1.0
Description:	When the vehicle aims at leaving the current lane (e.g. for an overtaking) the system shall assist her/him, indicating the right time and moment, taking into account the internal and external situation.
Validation & Verification	
Method:	Experiment in a driving simulator
Metric:	Behavioural and subjective data
Success:	yes
Comment:	SNV has collaborated with CRF and REL to realize the evaluation of the LCA system. In particular, SNV defined the protocol for data collection and analysed subjective data, as reported in D9.9.

Requirement:	**HMI for LCA**
ID:	WP9_CRF_AUT_REQ11
Ver:	1.0
Description:	HMI shall be appropriate and distinguishable by the driver, with different channels and modes, depending on the internal (state) and external (environment) situation.
Validation & Verification	
Method:	Evaluation of the communication strategies by means of questionnaires.
Metric:	User judgement of the HMIN of the AdCoS respect to different PIs: • Comprehensibility; • Distinguishability; • Perceptibility; • Effectiveness.
Success:	yes
Comment:	SNV has defined the protocol to conduct the experiment and prepared the questionnaire to collect subjective data, as described in D9.10, to verify that the HMI realized from REL for the LCA system was appropriate. SNV has been also responsible for data analysis.

6.11.4 Feedback from AdCoS owners

SNV received positive feedbacks from the partners. In particular in the Automotive domain SNV contribution has been evaluated as a very useful MTT to collect significant data. The results of the in-vehicle study have also been presented at the Conference of the European society for

Cognitive Psychology and the presentation aroused a great interest between the experts of the field.

Also in the case of the study about fatigue effects in the Aeronautic domain the results have been accepted for presentation at HCI Aero 2016 Conference.

6.12 Focus Group (SNV)

Focus Group (hereafter FG) methodology has been introduced in the HoliDes project from SNV to the aim to identify and address HF issues. In particular it has been applied in the development and evaluation of Safe Parallel Transmit Scanning in WP6 and for the evaluation of the Airbus AdCoS in WP8.

6.12.1 Overview

As explained in D5.5, FG is a methodology developed in the sociological field to the aim to collect relevant information about a specific topic from a group of people potentially or effectively interested in that topic. This MTT is useful to identify HF related problems and to envision a possible solution. The great advantage of this methodology, which has been adapted for HoliDes scopes, is the flexibility that facilitates the adaptation to different contexts and arguments.

6.12.2 Functionality

FG has been used for the development of the HMI of the UMC AdCoS. At the beginning of the HoliDes project UMC team provided the system with a preliminary version of the HMI. In cooperation with SNV, a first FG session has been organized to evaluate the preliminary layout for the interface realized by UMC team in order to acquire information about HF needs. A second version of the HMI has been developed following the indications emerged from the FG and addressing HF requirements. A second FG has been realized to evaluate the new HMI. The results of both studies have been reported in HF filer. All the details about those studies are inserted in D6.7 and D6.9.

For evaluation purposes, FG has been also applied to the Airbus system. The object of the study was the evaluation of potential end-users' attitude towards the advanced control room features introduced in the AdCoS, and above all the load balancing functionality. The activity has been developed following the guidelines for the evaluation developed in WP5. In particular, it has been focused on the subjective PI "Perceived usefulness". This activity is reported in detail in D8.9.

6.12.3 Evaluation

FG has been evaluated as a useful methodology in the HoliDes context where HF factors are the core of the project. The addressing of HF issues is fundamental also for exploitation purposes, as emerged in the Border Security use case.

Requirement:	**Load balancing Operator Level**
ID:	WP8_CAS_CTR_REQ021
Ver:	1.0
Description:	The system shall be able to analyse the workload of operators in one HQ and subsequently offer support to the supervisor to redistribute events among them.
Validation & Verification	
Method:	Users judgement
Metric:	Consensus degree
Success:	yes
Comment:	SNV prepared the protocol, realized the FG and analysed data to the aim to verify if this functionality of the AdCoS was appreciated from the point of view of potential users to help AdCoS owners to better understand the potentiality of the AdCoS for exploitation purposes.

6.12.4 Feedback from AdCoS owners

SNV team received a great appreciation for their effort in defining and conduction FGs. In particular, SNV contribution has been fundamental for the development of the HMI of the Safe Parallel Transmit Scanning, as stated in D6.7 and D6.9.

6.13 Cognitive Distraction Classifier (TWT)

The Cognitive Distraction Classifier (CDC) is a tool, or more accurately, a framework incorporating an applied cognitive and computational model, aimed to detect cognitive distraction, for example, while driving. It has the capability to record and classify several types of physiological and behavioural data offline (e.g., post-experimental) or online (i.e., near-to-real-time). The focus of the offline CDC has been on employing facial video data and behavioural driving data, whereas the current version of the online CDC uses facial video data only.

6.13.1 Overview

The CDC classifies the current cognitive distraction of a user into a predefined level. This MTT may help to overcome or avoid the negative effects of distracted drivers (such as accidents and collisions), when a system responds adequately to the detected level of distraction. A Human-Machine Interface or adaptive system (AdCoS) can use the classified level of cognitive distraction of the user either to adapt the task to demand less cognitive resources, or the system can try to reclaim the cognitive attention of the driver when needed.

Most current human-state classifiers aim to recognize drowsiness or visual distraction, but not cognitive distraction. Visual distraction is measured when the user is looking away from the subject it should focus to. Additionally, the CDC can be used online, meaning that the output levels have minimum delay and can be considered (near-to-) real-time. Within the HoliDes project, AdCoS were developed. These systems adapt to operator states, preferences or environmental conditions. The need for an online cognitive distraction classifier is therefore present within the HoliDes project. The CDC can be used in AdCoS focusing on safety, or to adapt the level of automation or preferred driving style to.

6.13.2 Functionality

To create a Cognitive Distraction Classifier, it is important to understand what cognitive distraction is, how it is defined, and how it can be measured. [6] described driver distraction such that *"a driver's attention is, voluntarily or involuntarily, diverted away from the driving task by an event or object to the extent that the driver is no longer able to perform the driving task adequately or safely."* Unfortunately, no direct measurements of cognitive distraction have thus far been discovered. What has been found out is that cognitive distraction can be measured with subjective measures and with objective measures that correlate highly with cognitive distraction. Two main categories of features correlating with cognitive (driver) distraction are ‘driving performances’ and ‘physical measures’. Driving performances can be measured through fluctuations in steering, use of the throttle pedal and fluctuations in speed [7] but also through measuring distance to the lead vehicle, reaction time to lead-vehicle position and changes in speed, and task completion time [8]. There are also physical measures that correlate to cognitive distraction. Facial features such as eyebrow and mouth movements have been linked to cognitive distraction [9]. Eye movements or eye gaze are very good measures for cognitive distraction. Scanning patterns and duration of fixations can indicate cognitive distraction [10] [11]. More precisely, when participants are cognitively distracted from a main task, they avert their gaze, or they move their gaze to walls, ceilings, floors, or they cover or close their eyes all together [12] [13].

Whether the physiological measures mentioned above might be useful in the CDC, is tested by TWT. The most recent offline CDC uses driving behaviour features and facial features (including eye blinks) to classify cognitive driver state. While collaborating with TAK, it was investigated whether adding Eye Tracking features to the CDC would increase its classifying accuracy. In TAK’s driving simulator a virtual driving experiment was conducted. Cognitive distraction was induced through a secondary task. For this the n-back task was used in which the participants repeat a set numbers, either immediately after they heard a number, or with a delay. During the driving and the n-back task, facial features, driving performance features and eye tracking features were measured.

The first statistical analyses compared driving behaviour features and eye tracking features between different experimental cognitive distraction states. The results showed that variation in cognitive distraction conditions did only influence one of the driving behaviour features. This is in line with results from previous experiments that showed that the CDC bases its distraction classification more on the facial features than on the behavioural driving features. For the Eye Tracking features, the same type of analyses were used to investigate whether the features regarding pupil size change due to induced cognitive distraction. With data from only nine participants the results are, however, not decisive. A trend was found towards increased pupil height when participants are more distracted. Another finding is that both driving behaviour features and pupil features are participant dependant. For Machine Learning techniques used within the CDC this is not a problem. Further investigations with Machine Learning techniques are still needed to investigate if adding an eye tracking system to the current set-up will increase the accuracy of the cognitive distraction classification.

6.13.3 Evaluation

The CDC has recently been investigated in an in-house experiment that evaluated the accuracy of both offline and online classifications. During this experiment, participants drove in a driving simulator, followed a lead car, and performed the n-back task in order to induce cognitive distraction. Several features were measured while participants experienced 3 levels of cognitive distraction. With machine learning techniques, these features were used to train the CDC during the experiment and after the experiment. Drivers' cognitive states were therefore either classified during the experiment by the online CDC or after the experiment by the offline CDC.

During offline (post-experimental) analysis of facial video and behavioural driving data collected in the experiment, we have been able to predict the level of distraction for all participants analysed so far at rates significantly above chance level (i.e., 33,3% for 3 classes). Most of the true positive rates are even higher than 60%. Compared to previous online classifications that led to high rates of misclassification, some adaptation were made for the online analyses. In addition to a (possible) inferior classification algorithm and a drastically reduced training set, feature pre-processing had to be simplified in our online classification code in order to meet real-time requirements. The current experiment also resulted in high rates of misclassification for the online CDC.

Evaluations of the in-house experimental data show great differences in the accuracies between the offline and online classifications of the CDC. One reason may be the difference in available data for the training set, which was much larger for offline analyses. We are further investigating these differences. Both feature pre-processing and machine learning can be tuned by various parameters. We will gain further insight by conducting a detailed study on the variation of relevant parameters and optimizing our classification analysis methods for online processing. Finally, extending the CDC with other physical signals such as eye-tracking and audio data may significantly improve results.

Further improvements should be made to the online set-up of the CDC to obtain more reliable results. We have developed the CDC, suited for online analysis, with a framework that can be extending with different types of data (such as audio voice, and eye-tracking data). Next steps include continuation of collaboration with partners to integrate the CDC in AdCoS, allowing adaptation of the level of automation, to the cognitive state of the driver, or appropriate systems-driver communications.

Requirement:	**Algorithm for analyses of distraction**
ID:	WP9_TWT_AUT_REQ04
Ver:	01
Description:	Distraction level classifier algorithm for feedback app
Validation & Verification	
Method:	Implemented into the TWT RTMaps diagram, steering the TWT driving simulator
Metric:	Different input signals for the algorithm have been tested and evaluated on their usefulness.
Success:	Offline classification results are highly above chance level (>60% over 33,3% chance). The online classification may need larger amounts of training data to approach offline results. More investigation is needed. Adding multiple types of data to the CDC may improve classification accuracies, as in indicated by advanced statistical analysis of eye-tracking and voice audio signals.

Requirement:	**Feedback rendering modality of estimated distraction level**
ID:	WP9_TWT_AUT_REQ14
Ver:	02
Description:	The estimated output level can be used to be integrated with other driving functionalities or as direct visual feedback as level of distraction
Validation & Verification	
Method:	The classified distraction level will be integrated into AdCoS
Metric:	For the IAS AdCoS, the CDC output is transferred into a single number from 0 to 1.
Success:	The CDC output level should be able to adapt system outputs. The CDC is currently being integrated into the IAS AdCoS.

6.13.4 Feedback from AdCoS owners

Feedback from AdCoS owners - TAK adaptive HMI

The adaptivity of the Adaptive HMI AdCoS is based on driver distraction in combination with environmental conditions. With the visual sense being the most dominant in driving, driver distraction is usually associated with visual distraction. However, based on the increase of both nomadic and in-vehicle communication devices in the car, cognitive distraction gains increasing importance. The Cognitive Distraction Classifier (CDC) developed by TWT reacts to this change and allows adapting the HMI accordingly.

TAKATA supported TWT in developing the CDC by integrating two experimental conditions in the simulator study that was conducted in HoliDes to evaluate the Adaptive HMI AdCoS. In the two conditions in the simulation cognitive distraction was implemented via an n-back task, specifically MIT's Delayed Digit Recall Task [14] [15]. This task was used in two versions, as 0-back task in the easy and as 2-back task in the difficult condition.

In order to be used as input for the Adaptive HMI AdCoS, the CDC must give feedback on the state of cognitive distraction. Preferably, this feedback would be in real time. However, the adaptivity of the AdCoS must be based on a certain length of distraction in order to ensure user acceptance. This would allow for a certain small time-lag in this information.

Because the CDC was still in the development phase at the time the experiments were conducted, experiences with the CDC cannot be reported here. However, different requirements can be named. This concerns the integration in the AdCoS as such that must be easily achievable. Furthermore, the CDC must be reliable in order to ensure user acceptance. Finally, a certain amount of adaptivity should be possible in order to adjust user preferences.

Feedback from AdCoS owners – IAS Adapted Automated

The CDC was successfully integrated and tested on the AdCoS. Further integration and functional tests are planned for the remainder of the project duration. The performance of the MTT in the context of the AdCoS could not be evaluated due to technical and legal constraints. This issue has been described in previous deliverables of WP9. There were no major issues with the integration of the MTT on the AdCoS. The final experiments will give an indication for the performance, even though a scientific evaluation with different subjects is not possible. The benefit that comes with the integration of the MTT on the AdCoS will be visible from the final evaluation that is carried out in simulation.

6.14 Classifier of visual driver distraction based on data on vehicle dynamics (UTO/CRF)

6.14.1 Overview

Driver distraction and inattention are an important safety concern [16]. Deriving knowledge about the human operator can be very valuable in the system validation phase. While interacting with a prototype or some modules of the AdCoS, the operator's degree of visual distraction can be evaluated.

The purpose of this system is to classify driver distraction based on vehicle dynamics using machine learning techniques using the Visual Driver Distraction Classifier (VDC).

The module provides feedback whether or not a new system increases or decreases the operator's degree of distraction, and this information can be used to design how to adapt the interface of the AdCoS.

In particular, it consists of two modules: the first module works offline and learns a classifier from sensory data. The second module works online and makes predictions on the status of the driver using the knowledge acquired offline.

Therefore the module can be used online to classify the driver's distraction not only during the testing phase of a prototype, but also during everyday interaction with the AdCoS. This online measure of distraction could in turn be used to adapt the degree of automation of the AdCoS to the driver's state.

6.14.2 Functionality

This module will help the lane change assistant functionality and it will be applied in the frontal collision use case from WP9, and also used in the overtaking use case of the same WP9. The module provides feedback whether or not the operator is distracted.

Data from the system dynamics during driving have been collected from driving session performed on an equipped vehicle provided by CRF. 29

test subjects drove for about 1 hour on normal and motorway roads and they had to perform a SuRT task while driving.

In particular, the vehicle dynamics data considered are the following:

- Speed [m/s]
- Time To Collision [s]
- Time To Lane Crossing [s]
- Steering Angle [deg]
- Lateral Position [m]
- Lane Width [m]
- Road Curvature [%]
- Heading Angle [deg]
- Position of the accelerator pedal [%]
- Position of the brake pedal [%]
- Turn indicator [on/off]
- X,Y coordinates of car in front (if any)
- Speed of car in front (if any)

These values are directly available on the prototype vehicle CAN bus or can be derived from those (e.g., time to collision is computed from vehicle speed and car in front data). The frequency of data collection was 20 Hz (1 data-point each 0.05s), which is the output rate of the simulator. Several experiments have been carried on by varying the period over which values are averaged from 1s to 2s. In order to be consistent with the target variable (distracted or not-distracted), data have been labelled distraction if the driver was not looking at the road for the whole considered period, not distracted otherwise. For the time being, we just considered only two possible levels of driver distraction.

Because of the way the experiment was designed, we consider here the visual distraction (eyes off the road). Although we cannot directly address other types of distraction (e.g. cognitive) by this experiment, nonetheless visual distraction is associated with greater odds to crash-relevant conflict than cell phone conversation (cognitive distraction).

Examples of what you could use:

- schematics
- screenshots

- verbal descriptions
- pictures
- class diagrams
- flow charts

6.14.3 Evaluation

We made a large number of experiments by varying the learning algorithm parameters, such as the number of neurons, learning rates, number of training instances, etc. Anyway, here we only report results obtained by building and testing a model on every given subject separately by performing ten-fold cross validation on each dataset. The table below reports the correct classification rates.

Subject #	Training time (s)	% correct predictions
1	0.83	97.5313
2	0.54	97.2761
3	0.73	95.6618
4	0.31	96.4755
5	1.11	95.3813
6	0.72	94.6761
7	1.98	96.3278
8	0.46	97.3091
9	0.42	96.4286
10	0.47	97.2863
11	0.64	98.0251
12	0.23	95.4148
13	0.43	94.8718
14	0.73	96.1206
15	0.54	94.2682
16	0.4	95.915
17	0.31	93.8942
18	0.87	96.6554
19	0.38	96.0703
20	0.4	95.8815
21	0.55	95.5548
22	0.53	97.3172
23	0.34	93.8014

24	1.07	94.8655
25	0.48	95.0667
26	0.5	92.8363
27	0.41	95.5286
28	0.71	94.7718
29	0.78	95.7729
Average	**0.62**	**95.758**

In a second experiment, we evaluated the possibility to generalize the proposed classifier method by performing training on a dataset containing data from 28 subjects and testing on the left out subject.

The table below reports training times and correct classification rates in this case.

Subject #	Training time (s)	% correct predictions
1	120.64	76.963
2	135.86	70.7317
3	101.95	75.4417
4	104.63	83.6237
5	99.4	78.679
6	84.14	79.4025
7	100.76	76.2582
8	125.35	80.0305
9	135.06	74.2788
10	91.9	80.7527
11	89.15	81.0773
12	76.39	80.4533
13	135.39	78.7482
14	113.42	80.2001
15	129.72	78.391
16	85.86	77.5732
17	103.48	77.1776
18	95.8	77.8656
19	86.72	80.294
20	93.19	76.1682
21	94.74	79.5762
22	90.23	77.2691

23	94.94	74.9064
24	101.58	82.5044
25	100.65	76.2222
26	85.09	78.9192
27	117.55	83.7456
28	113.97	76.0965
29	124.27	80.4752
Average	**104.54**	**78.407**

A general comment about the reported results is that the learned model performs very well (better than required) when trained and tested on the same subjects, but does not generalize as well when trained on a set of subjects and tested on a different subject, even though the average performances are anyway acceptable.

Requirement:	**Driver distraction classification**
ID:	WP9_CRF_AUT_REQ03
Ver:	2.0
Description:	The AdCoS should constantly monitor the driver and classify his/her mental state (i.e. distraction) by using real-time data.
Validation & Verification	
Method:	Classification rate on validation set
Metric:	CR% (Classification Rate Percentage)
Success:	CR% > 80%

Comment:	The requirement is satisfied if we consider each subject separately on his own, and it is only satisfied in 10 out of 29 cases when built from all subjects but one used for testing.

6.14.4 Feedback from AdCoS owners

The feedbacks from AdCoS owners are related to the use of two MTTs: the Driver Visual Distraction Classifier (DVDC) and the SuRT.

For the first, all details are presented in the previous sections. As mentioned, the performances of DVDC have been surely acceptable, considering “one model for one subject”.

The difficulty in this field of study lies in the fact that different mental workload levels could correspond to the same behaviour, above all when different users are involved in the training process [17]. Some of the existing workload classifiers are subject-specific, meaning a new classifier has to be trained for each subject and session (as pointed out from Aeronautics domain in D9.7, where a detailed analysis is provided). In fact, this MTT does not achieve good performances when classifying a group of subjects with the same training dataset and also the same subject in different sessions. Here it seems that it would be necessary to tailor the model for each pilot (what we have done). This research has not been able to find a direct transfer learning. Statistical distribution of the data varied across subjects as well as across sessions, even within individual subjects, limiting the transferability of our trained models between them.

Further steps involve the use of other signal sources (e.g. EEG sensors, HR sensors, etc., especially for cognitive distraction) as well as a deep investigation of the methods and tools for the data collection, to be used for the classifier development (in particular, the ground-truth sensor to obtain the target-set for training the model).

For what concerns the SuRT, in the “Adaptive Assistance” AdCoS, it was used in two experiments on CRF real-car and in REL driving simulator, in order to induce visual distraction in the users, with a twofold goal:

- to collect data about driver's distraction, then used to develop the related MTT (visual distraction classifier)
- in the evaluation phase, to create the "distractor" (i.e. the source of distraction) and thus to assess the adaptivity of the AdCoS.

The total number of participants (considering both experiments) was around 60 people. The conditions with SuRT varied between the two experiments, it was used in one condition in the first experiment on the road and in two out of five conditions in the second experiment in the simulator.

For the rest, we found the same results as TAK, that is, the SuRT allows the easy implementation of a large variety of conditions (target and non-target colour, size, timing, etc.) via configuration file. Moreover, the SuRT was easy to explain and was understood by all participants without problem. Interacting with the SuRT resulted in the expected level of distraction and thus fulfilled the requirements.

The data output saved by the SuRT also fulfilled the requirements and could easily be summarized in any form needed (especially in MATLAB format, used to develop the classifier).

During the experiments, only very few times the SuRT produced technical problems (block and crash of the related program in RT-Maps) and thus proved to be very reliable.

The integration in RT-Maps and in the driving simulation tool of REL (always by means of RT-Maps) was FULLY achieved. As TAK pointed out, although precompiled libraries are available, the source code is not. This makes a full integration in different programs difficult and requires that the SuRT runs as a standalone window in the background (and this can create some run-time problems in certain cases). Thus, a full availability of the source code would be appreciated.

7 References

[1] B. E. John, “Extensions of GOMS analyses to expert performance requiring perception of dynamic visual and auditory information.,” in *Proceedings of the SIGCHI conference on Human factors in computing systems.*, 1990.

[2] B. John and D. E. Kieras, “The GOMS family of user interface analysis techniques: Comparison and contrast.,” in *Transactions on Computer-Human Interaction*, 1996.

[3] K. J. Vicente, Cognitive work analysis, Mahwah, N.J.: Lawrence Erlbaum Associates, 1999.

[4] L. Onnasch, J. Bürglen, S. Tristram and H. Kolrep, “Break it down! A Practicioner's Software Tool for Human Factors Task Analysis and Human Error Idnetification.,” in *Proceedings of EAAP 2016*, 2016.

[5] L. G. Militello and R. J. Hutton, “Applied cognitive task analysis (ACTA): a practitioner's toolkit for understanding cognitive task demands.,” *Ergonomics,* vol. 41, no. 11, pp. 1618-41, 1998.

[6] K. Young and M. Regan, “Driver distraction: A review of the literature,” in *Distracted driving*, I. J. Faulks, M. Regan, M. Stevenson, J. Brown, A. Porter and J. D. Irwin, Eds., Sydney, NSW, Australasian College of Road Safety, 2007, pp. 379-405.

[7] L. Jin, Q. Niu, H. Hou, H. Xian, Y. Wang and D. Shi, “Driver cognitive distraction detection using driving performance measures,” *Hindawi publishing corporation discrete dynamics in nature and society,* p. 12,

2012.

[8] D. B. Kaber, Y. Liang, Y. Zhang, M. L. Rogers and S. Gangakhedkar, “Driver performance effects of simultaneous visual and cognitive distraction and adaptation behavior.,” *Transportation research part F: Traffic psychology and behaviour,* pp. 491-501, 2012.

[9] D. Metaxas, S. Venkataraman and C. Vogler, “Image-based stress recognition using a model-based dynamic face tracking system.,” in *Computational Science - ICCS 2004*, Berlin, Heidelberg, 2004.

[10] M. A. Recarte and L. M. Nunes, “Effects of verbal and spatial-imagery tasks on eye fixations while driving.,” *Journal of Experimental Psychology: Applied,* vol. 6, no. 1, p. 31, 2000.

[11] M. A. Recarte and L. M. Nunes, “Mental workload while driving: effects on visual search, discrimination, and decision making,” *Journal of experimental psychology: Applied,* vol. 92, no. 2, p. 119, 2003.

[12] G. Doherty-Sneddon, V. Bruce, L. Bonner, S. Longbotham and C. Doyle, “Development of gaze aversion as disengagement from visual information,” *Developmental psychology,* vol. 38, no. 3, p. 438, 2002.

[13] A. M. Glenberg, J. L. Schroeder and D. A. Robertson, “Averting the gaze disengages the environment and facilitates remembering,” *Memory & cognition,* vol. 26, no. 4, pp. 651-658, 1998.

[14] B. Reimer and B. Mehler, “The impact of cognitive worload on pysiological arousal in young adult drivers: a field study and simulation validation,” *Ergonomics,* vol. 54, no. 10, pp. 932-942, 2011.

[15] B. Mehler, B. Reimer and J. A. Dusek, “MIT AgeLab Delayed Digit Recall Task (n-back),” 2011.

[16] M. A. Regan, C. Hallett and C. P. Gordon, “Driver distraction and driver inattention: Definition, relationship and taxonomy,” *Accident Analysis & Prevention,* vol. 43, no. 5, 2011.

[17] G. Borghini, L. Astolfi, G. Vecciato, D. Mattia and F. Babiloni, “Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness.,” *Neuroscience and biobehavioral reviews,* vol. 44, pp. 1-18, 2012.

[18] A. Schieben, M. Heesen, J. Schindler, J. Kelsch and F. Flemisch, “The theater-system technique: Agile designing and testing of system behavior and interaction, applied to highly automated vehicles.,” in *Proceedings of the First International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, Essen, 2009.

[19] G.-B. Huang, Q.-Y. Zhu and C.-K. Siew, “Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks.,” *Extreme Learning Machine: Theory and Applications,* no. 70, pp. 489-501, 2006.

[20] C. Huang, X. Ding and C. Fang, “Head Pose Estimation Based on Random Forests for Multiclass Classification,” in *20th International Conference on Pattern Recognition (ICPR)*, 2010.

[21] F. Tango and M. Botta, “Real-Time Detection System of Driver Distraction using Machine Learning,” *Intelligent Transportation Systems,* vol. 14, no. 2, 2013.

[22] H. Lausberg and S. Kita, “: The content of the message influences the hand choice in co-speech gestures and in gesturing without speaking,

Brain and Language," *Brain and Language,* vol. 86, no. 1, pp. 57-69, 2003.

[23] C. Xu, S. Li, G. Liu, Z. Zhang, E. Miluzzo, Y. Chen, J. Li and B. Firner, "Crowd++: Unsupervised Speaker Count with Smartphones.," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'13)*, Zurich, Switzerland, 2013.

[24] T. Horberry, J. Anderson, M. A. Regan, T. J. Triggs and J. Brown, "Driver distraction: the effects of concurrent in-vehicle tasks, road environment complexity and age on driving performance.," *Accident Analysis Prevention,* vol. 38, no. 1, pp. 185-191, 2006.

[25] M. A. Regan and K. L. Young, "Driver distraction: a review of the literature and recommendations for countermeasure development.," in *Proc. Australas. Road Safety Res. Policing Educ. Conf.*, 2003.

[26] L. M. Trick, J. T. Enns, J. Mills and J. Vavrik, "Paying attention behind the wheel: a framework for studying the role of attention in driving.," *Theoretical Issues in Ergonomics Science,* vol. 5, no. 5, pp. 385-424, 2004.

[27] J. Artho, S. Schneider and C. Boss, "Unaufmerksamkeit und Ablenkung: Was macht der Mensch am Steuer?," Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 2012.

[28] Q. Ji and X. Yang, "Real-time eye, gaze, and face pose tracking for monitoring driver vigilance," *Real-time imaging,* pp. 357-377, 2002.

[29] H. Saito, T. Ishiwaka, M. Sakata and S. Okabayashi, "Applications of driver's line of sight to automobiles - what can driver's eye tell," in *Vehical and information systems conference*, 1994.

[30] Y. Ebisawa, “Unconstrained pupil detection technique using two light sources and the image difference method,” *Visualization and intelligent design in engineering,* pp. 79-89, 1989.

[31] P. Jimenez, L. M. Bergasa, J. Nuevo, N. Hernandez and I. G. Daza, “Gaze fixation system for the evaluation of driver distractions induced by IVIS,” *Online,* 2011.

[32] M.-H. Sigari, M. Fathy and M. Soryani, “A driver face monitoring system for fatigue and distraction detection,” *International journal of vehicular technology,* 2013.

[33] M. F. Khan and F. Aadil, “Efficient car alarming system for fatigue detection during driving,” *International journal of innovation, management and technology,* pp. 480-486, 2012.