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

D4.3 – Metrics for Model-based and Empirical AdCoS Qualification

Project Number:	332933
Classification:	Public
Work Package(s): **Milestone:** **Document Version:** **Issue Date:** **Document Timescale:** Start of the Document: Final version due:	WP4 M1 V1.0 13.06.2014 Project Start Date: October 1, 2013 Month 07 Month 07
Deliverable Overview:	**Main document:** D4.3_Metrics_for_Model-based_and_Empirical.doc (Public)
Compiled by:	Bohuslav Křena, BUT
Authors:	Aleš Smrčka, BUT Bohuslav Křena, BUT Daniel Prun, ENA David Käthner, DLR Fabio Tango, CRF Gert Weller, TAK Holger Bracker, EAD-DE-CAS Jan-Patrick Osterloh, OFF Mark Eilers, OFF Martin Krähling, IAS Susanna Donatelli, UTO Thierry Bellet, IFS Zdeněk Morávek, HON

Reviewers:	Ian Giblett, EAD-UK (external) David Käthner, DLR (external) Stefan Rieger, TWT (internal) Holger Bracker, EAD-DE-CAS (internal)
Technical Approval:	Jens Gärtner, EAD-DE-IW
Issue Authorisation:	Sebastian Feuerstack, OFF

DISTRIBUTION LIST		
Copy type[1]	Company and Location	Recipient
T	HoliDes Consortium	all HoliDes Partners

[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
15. 04. 2014	Initial version	Bohuslav Křena, BUT
04. 05. 2014	Version for project meeting in Naples	Bohuslav Křena, BUT
25. 05. 2014	Version for internal review	Bohuslav Křena, BUT
30. 05. 2014	Version for external review	Bohuslav Křena, BUT
13. 06. 2014	Final version for submission	Bohuslav Křena, BUT

Table of Contents

List of Acronyms

ACC = Adaptive Cruise Control

AdCoS = Adaptive Cooperative Human-Machine System(s)

AER = Aeronautics

AUT = Automotive

CTR = Control rooms

D = Deliverable

EFB = Electronic Flight Bag

HEA = Health

HF-RTP = Human Factors Reference Technology Platform

HMI = Human-Machine Interaction

HoliDes = Holistic Human Factors Design of Adaptive Cooperative Human-Machine Systems

HQ = Headquarter

REQ = Requirement

T = Task

WP = work package

1 Introduction

This document provides the definition of metrics for requirements that have been selected within the preceding requirement analysis (Deliverable D4.1). Using such metrics, requirements imposed on particular AdCoS can be verified and if the AdCoS satisfies all the requirements then the AdCoS is qualified for usage. In the second part, this document contains brief description of model-based and empirical techniques and tools that could be used for verification of these requirements.

Inputs:

- The most essential input for this document is deliverable D4.1: Requirements Analysis for Model-based Analysis Techniques and Tools – Vs 1.0 – 10/04/2014 that provides an analysis of requirements imposed on AdCoS and in which a subset of requirements has been selected for further examination.
- The original requirements from WP6-WP9 can be found in the following documents:
  - D6.1: Health related scenario descriptions – Vs 1.1 – 15/02/2014
  - D7.1: Requirements Definition for the HF-RTP, Methodology and Techniques and Tools from a Aeronautics Perspective – Vs 1.0 – 12/02/2014
  - D8.1: Requirements Definition for the HF-RTP, Methodology and Techniques and Tools from a Control Room Perspective – Vs 0.8 – 14/02/2014
  - D9.1: Requirements Definition for the HF-RTP, Methodology and Techniques and Tools from an Automotive Perspective – Vs 0.1 – 14/02/2014
- List of tools: Tool listing V4 (file "HoliDes - Tools-listing.xlsx") dated 17/03/2014 has been used, however, enriched by additional information requested from people involved in WP4 and WP5.

Outputs:

This document contains metrics for requirements selected for further examination in Deliverable D4.1 and mapping between requirements and model-based and empirical techniques and tools that could be used for their verification.

2 Background

Apart from other qualities, requirements must be described in a way that allows objective determination whether a property imposed on a system by a requirement is satisfied by the implemented system or not. For quantitative requirements, we can say that requirements must be **measurable**. If objective evidence that the requirement is met by the system cannot be provided, contribution of the requirement to the system development is rather disputable. At first, developers do not need to care about it if they know that there is no way how to check its satisfaction. And even if the developers take the requirement into account, without objective metrics, they can understand it in a different way than the customer and thus, they can develop something different than the user expected (e.g., the requirement on user-friendly interface can be implemented by developers as a command line interface with pretty compact set of commands while users could expect windows-based graphic interface controllable by a mouse). The main goal of this document is to make sure that the selected requirements provided from WP6 to WP9 are measurable (and for qualitative requirements, that they comprise unambiguous fit criteria indicating requirement's fulfilment) and thus, leading to an AdCoS which is in line with the users expectations.

Within the software engineering field, many metrics have been proposed and used targeting different aspects of product quality (e.g., functionality, usability, portability, reliability, maintainability), process quality as well as development costs (e.g., [1], [2], [3], [4]). In the following, we concentrate on metrics (and fit criteria) for requirements that are very diverse as the systems to be developed are also very diverse. Nevertheless, the important role is, in this document, naturally played by the metrics related to model-based and empirical verification of AdCoS regarding human factors and safety issues:

- Metrics directly related to an AdCoS itself for evaluating, measuring, or assessing its efficiency (based on an *engineering* point of view: does the AdCoS work as technically expected) and its effectiveness (based on an *ergonomics* point of view: does the AdCoS provide an assistance to the human that really corresponds to the aid function as initially specified and derived needs), according to what was initially expected from this AdCoS, when it was designed.

- Metrics related to human operator for (1) assessing or diagnosing, for instance, some risky inadequate behaviours and (2) comparing or analysing effects of AdCoS on the human's activity (e.g., whether and how the driving task changes according to AdCoS action).
- Metrics related to Human-Machine Interaction for investigating and evaluating HMI issues (e.g., task allocation, workload effects of AdCoS, etc.).
- Metrics concerning the Human-Machine System performance as a whole (e.g., human alone versus human + AdCoS versus full automation), additional safety gained by the AdCoS and potential risks liable to be introduced by the AdCoS.

3 Metrics for selected requirements

This section constitutes the core of this deliverable as it defines metrics for requirements selected within the preceding requirement analysis. For some requirements, several metrics may be appropriate, the most suitable ones will be determined in the course of the project. If applicable, appropriate thresholds may be defined (or refined) in the course of the project.

3.1 Health requirements (WP6)

3.1.1 WP6_ATO_HEA_REQ25

Requirement:
The system should be operational in case of failures.

Metrics:
If we are strict, this requirement cannot be satisfied as it is impossible to develop a system that is able to handle any kind of failure. Thus, the requirement needs to be refined, e.g., the system should be able to tolerate predefined set of faults. Then, one can consider the following metrics:

- After encountering a fault, the system will recover within predefined time (e.g., 1 second).
- After recovering, the system must contain all patient data as before the fault occurrence.

3.1.2 WP6_AWI_HEA_REQ01

Requirement:
The operator model should be able to identify the operators' skill and experience level through their (overt) actions.

Metrics:
Operator's skill classification (the following classes of operators are anticipated: basic, advanced, expert) provided by the system should be precise enough (e.g., at least 90 % of operators are correctly classified regarding their skills) in comparison with classification by expert operators.

3.1.3 WP6_AWI_HEA_REQ03

Requirement:
The human factors models should allow a simulation of an operator conducting an MRI scan with the relevant guidelines, such as procedure archetypes.

Metrics:
The simple metrics can be an answer "yes" or "no" based on a check whether the system provides a simulation of an operator or it does not. More detailed metrics can require that the human factors models allow to simulate predefined set of test scenarios (need to be specified later) or at least its substantial subset (e.g., at least 90 % of test scenarios).

3.1.4 WP6_AWI_HEA_REQ07

Requirement:
The human-machine interaction model should be able to handle actions in the physical world that are outside of the control of the system, and still adapt and give proper guidance to the operator.

Metrics:
The following two metrics shall be considered:

- The number of correctly detected unexpected patient states to the number of all unexpected patient states with respect to expert judgement.
- The number of proper guidance (with respect to expert judgement) provided by the system to the total number of detected unexpected patient states.

As not all possible external events (i.e., actions in the physical world) can be taken into consideration by a HMI model, the events that need to be considered in a MHI model should be specified explicitly later. Then, the ratio of correctly handled external events by the HMI model can be evaluated too.

3.1.5 WP6_AWI_HEA_REQ11

Requirement:
The AdCoS should support dynamic sharing of model or situation identification between operator and system based on image representations or similar data.

Metrics:
The simple metrics is answer “yes” or “no”, however, for reasonable evaluation, one should examine numbers and types of functionalities available according to operator experience. The ratio of correctly assessed operator experience by the system against opinion of experts should be, for instance, at least 90 %.

3.1.6 WP6_AWI_HEA_REQ25

Requirement:
The AdCoS should adapt to both medical and procedural context when the operator requests remote assistance.

Metrics:
The adaptation of operator request handling should take into account all significant data from medical as well as procedural context. Considerable portion (e.g., at least 85 %) of request handling adaptation should be classified by experts as adequate.

3.1.7 WP6_AWI_HEA_REQ35

Requirement:
After adding new features, automatic tests should show that the entire system, including the UI, is still running as it is supposed to run.

Metrics:
After a new feature is added to the system, all automated regression tests should pass. In addition, one shall also consider metrics based on various coverage criteria [5] to ensure that set of automatic tests is adequate.

3.1.8 WP6_AWI_HEA_REQ43

Requirement:
After checking in with the user's credentials the UI can automatically adapt to their personalised settings.

Metrics:
The simple metrics can be an answer “yes” or “no” based on a check whether the system provides an adaptation or it does not. More detailed metrics can

consider the number of UI features that can be personalised. If needed one may also examine performance issues here, e.g., the adaptation of UI should be finished within predefined time (for instance, 100 ms) after the operator is recognized.

3.1.9 WP6_IGS_HEA_REQ03

Requirement:
The system shall be able to represent activities that are performed by operators. It includes estimated, execution times, periodicity, staff involved, prerequisites …

Metrics:
All the performed actions of operators (from predefined set) are recorded in a log file with correct time stamps (begin and end of actions) and right operator identification.

3.1.10 WP6_IGS_HEA_REQ09

Requirement:
The platform shall ease methods and tools to measure the usability of application.

Metrics:
The number of performance indicators that are provided by the platform (probably not related directly to WP4 and WP5 techniques and tools as it targets HF-RTP itself).

3.1.11 WP6_IGS_HEA_REQ10

Requirement:
The platform shall ease the model and implementation of decision making.

Metrics:
Average time of decision making with the support to average time of decision making without the support (probably not related directly to WP4 and WP5 techniques and tools as it targets HF-RTP itself).

3.1.12 WP6_PHI_HEA_REQ08

Requirement:
Tooling shall allow fast iteration to rapidly validate various concepts interactively.

Metrics:
The platform should allow iterative development, i.e., it must support models refinement between iterations as well as models sharing among tools that can be employed in different phases of software development (probably not related directly to WP4 and WP5 techniques and tools as it targets HF-RTP itself).

3.2 Aeronautics requirements (WP7)

3.2.1 WP7_HON_AER_REQ30

Requirement:
The system should provide a consistent and intuitive user interface, within and across the various hosted applications; including, but not be limited to, data entry methods, colour-coding philosophies, and symbology.

Metrics:
Certain average value (e.g., at least 4 points on 5 points scale) should be received in questionnaires filled in by pilots based on experiments with the system.

3.2.2 WP7_HON_AER_REQ44

Requirement:
The system should be designed to minimise the occurrence and effects of flight crew error and maximise the identification and resolution of errors; for example, terms for specific types of data or the format in which latitude/longitude is entered should be the same across systems. Data entry methods, colour-coding philosophies, and symbology should be as consistent as possible across the various hosted EFB (Electronic Flight Bag) applications. These applications should also be compatible with other flight crew compartment systems.

Metrics:

This requirement aims at different properties and can be split into a set of more focused requirements. The following metrics have been identified:

- Scores in system usability scale (SUS) must be above 68 (high system usability).
- Occurrence of errors during interaction with the system must be lower than 5% (including data interpretation and maneuver execution).

3.2.3 WP7_HON_AER_REQ45

Requirement:

The EFB system should be capable of alerting the flight crew of probable EFB system failures.

Metrics:

- The system should detect 100 % of predefined error states.
- The flight crew should detect 100 % of failure states.

3.2.4 WP7_HON_AER_REQ46

Requirement:

The system should provide feedback to the user when user input is accepted. If the system is busy with internal tasks that preclude immediate processing of user input (e.g. calculations, self-test, or data refresh), the EFB should display a “system busy” indicator (e.g. clock icon) to inform the user that the system is occupied and cannot process inputs immediately.

The timeliness of system response to user input should be consistent with the application’s intended function. The feedback and system response times should be predictable to avoid flight crew distractions and/or uncertainty.

Metrics:

- Responsiveness to pilot action should not exceed 250 ms.
- 90th percentile of responsiveness to all actions is below 100 ms.
- If response time exceeds 250 ms a "busy" indicator must be displayed.

3.2.5 WP7_HON_AER_REQ49

Requirement:
The positioning and procedures associated with the use of the EFB should not result in unacceptable flight crew workload. Complex, multi-step data entry tasks should be avoided during take-off, landing, and other critical phases of the flight. An evaluation of the EFB intended functions should include a qualitative assessment of incremental pilot workload, as well as pilot system interfaces and their safety implications.

Metrics:
The system should have no or minimum impact on pilot workload. In general, it means value below 5 in Bedford workload scale (subjective measurement) or no significant difference between experimental and control condition for NASA-TLX.

3.2.6 WP7_HON_AER_REQ66

Requirement:
Create a common GUI that will allow to show dynamic logs, physiology recordings, event lists etc. at one time and that will allow for annotations of a situation.

Metrics:
Application can read a set of test signals from files with predefined format and visual inspection verifies alignment of the data (features in defined positions should overlap with 100% accuracy). To avoid ambiguity, the remaining elements in the requirement should be specified.

3.2.7 WP7_HON_AER_REQ71

Requirement:
Create a tool that is able to automatically evaluate the quality of an artifact according to general rules. The artifact may be defined as a screenshot or element description etc.

Metrics:
The simple metrics can require that 100 % of errors must be detected in the test samples. More detailed metrics focusing on other aspects of artifact's quality may be proposed later, of course, with regards to selected rules.

3.2.8 WP7_HON_AER_REQ78

Requirement:
Create a tool/methodology that is able to classify an action of an agent (human, machine) as 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.

Metrics:
This requirement is considered as satisfied if 75% success in classifying relevance of a taken action when compared to the consensus of experts is achieved.

3.3 Control rooms requirements (WP8)

3.3.1 WP8_ADS_CTR_REQ17

Requirement:
The system shall be able to analyze the status and workload of adjacent HQs (Headquarters) and subsequently offer support to transfer events to them.

Metrics:
The simple metrics can be an answer "yes" or "no", however, for reasonable status and workload analysis, the system shall monitor the following characteristics:

- HQ response time to the last incoming external event
- HQ mean response time to the last incoming external events during the last 5 minutes
- Current number of external events under treatment at the HQ level
- Offer to the superior HQ capabilities to transfer from one HQ to another (yes/no)
- Offer to the superior HQ capabilities to monitor current workloads of each HQ (yes/no)
- Identification of overloaded HQ (yes/no)
- Identification of underloaded HQ (yes/no)

3.3.2 WP8_ADS_CTR_REQ18

Requirement:
The system shall be able to analyze the workload of operators in one HQ and subsequently offer support to the supervisor to redistribute events among them.

Metrics:
The simple metrics can be an answer “yes” or “no”, however, for reasonable status and workload analysis, the system shall monitor the following characteristics:

- Operator response time to the last incoming external event
- Operator mean response time to the last incoming external events during the last 5 minutes
- Current number of external events under treatment
- Offer to the supervisor capabilities to transfer from one operator to another (yes/no)
- Offer to the supervisor capabilities to monitor current workloads of each operator (yes/no)
- Identification of overloaded operators (yes/no)
- Identification of underloaded operators (yes/no)

3.3.3 WP8_ADS_CTR_REQ22

Requirement:
The system shall offer scaled functionality (based on operator experience).

Metrics:
The simple metrics is an answer “yes” or “no”, however, for reasonable evaluation, one should examine numbers and types of functionalities available according to operator experience (the following classes of operators are anticipated: basic, advanced, expert). The ratio of correctly assessed operator experience by the system against opinion of experts should be at least 85 %.

3.3.4 WP8_IRN_CR_REQ04

Requirement:
The AdCoS shall normalize the workload, either low or high, on the operator.

Metrics:
The simple metrics can be an answer “yes” or “no”, however, for reasonable workload analysis and normalization, the system shall consider the following characteristics:

- Operator response time to the last incoming external event
- Operator mean response time to the last incoming external events during the last 5 minutes
- Number of system/operator interactions during the last minute
- Current number of external events under treatment

3.3.5 WP8_IRN_CR_REQ09

Requirement:
The AdCoS shall adapt to the competence and expertise level of the operator.

Metrics:
The simple metrics is an answer “yes” or “no”, however, for reasonable evaluation, one should examine numbers and types of functionalities available according to operator experience (the following classes of operators are anticipated: basic, advanced, expert). The ratio of correctly assessed operator experience by the system against opinion of experts should be at least 85 %.

3.3.6 WP8_IRN_CR_REQ10

Requirement:
The AdCoS shall adapt to the psycho-physical status of the operator (e.g., high/low workload, time pressure, physical features).

Metrics:
The simple metrics can be an answer “yes” or “no”, however, for reasonable status and workload analysis, the system shall consider the following characteristics:

- Operator response time to the last incoming external event
- Operator mean response time to the last incoming external events during the last 5 minutes
- Number of system/operator interactions during the last minute
- Current number of external events under treatment

- Offer to the supervisor capabilities to transfer from one operator to another
- Offer to the supervisor capabilities to monitor current workloads of each operator
- Identification of overloaded operators (yes/no)
- Identification of underloaded operators (yes/no)

3.3.7 WP8_IRN_CR_REQ11

Requirement:
The AdCoS shall adapt with respect to the role assigned to each operator for incoming calls.

Metrics:
The simple metrics with an answer "yes" or "no" can be used, however, if the requirement is refined to be more specific, other metrics can be considered as well, evaluating, for instance, the number of correct assignments of incoming calls to the total number of incoming calls during examined period or number of features that can be adapted according to the operator's role.

3.3.8 WP8_IRN_CR_REQ12

Requirement:
The AdCoS shall adapt to the language competences of the caller.

Metrics:
There are two simple metrics with "yes" or "no" answers assessing whether the system is able to detect caller language (from predefined set of languages) and then to adapt the interface according to the caller language. More detailed metrics can examine the number of correct detections of the language with the following interface adaptation to the total number of incoming calls during the examined period.

3.3.9 WP8_IRN_CR_REQ13

Requirement:
The AdCoS shall adapt to the geographical localization of the caller and of the target installation.

Metrics:
The system should provide visualisation on a map of the caller localization ("yes" or "no" answer) and evaluate ratio of correct localisation during the examined period.

3.3.10 WP8_IRN_CR_REQ15

Requirement:
The AdCoS shall adapt to the frequency of incoming calls.

Metrics:
This requirement needs to be refined to be more specific to be sure that the following interpretation is right. The system shall measure number of incoming calls per minute and provide appropriate actions for (1) increasing frequency (e.g., indicating to supervisor a need for additional operator) as well as for (2) decreasing frequency (e.g., indicating that some operator can take a pause). The number of appropriate actions with respect to opinion of experts to the total number of taken actions should be high enough (e.g., at least 90 %). The number of situations where some action of the system has been necessary but not provided by the system to the total number of situations requiring an action of the system with respect to opinion of experts should not exceeds some threshold (e.g., 20 %).

3.3.11 WP8_IRN_CR_REQ17

Requirement:
The AdCoS shall adapt to the priority level of the malfunctioning detected and the type of service addressed.

Metrics:
The following characteristics will be evaluated:

- Percentage of correctly classified priority levels of malfunctions
- Percentage of correctly detected types of services
- Percentage of adequate actions taken by the system with respect to opinion of experts
- Percentage of missing actions from all the situations where the action should be taken by the system

3.3.12 WP8_IRN_CR_REQ18

Requirement:
The AdCoS shall adapt to the asynchronous between the call and the malfunctioning detection.

Metrics:
Two simple metrics with "yes" or "no" answer should assess whether the system offers a mechanism to the operator to manage several incoming calls related to a same malfunction and the ability to associate a call to a particular malfunction. More detailed metrics can examine the ratio of correctly classified calls.

3.3.13 WP8_IRN_CR_REQ19

Requirement:
The AdCoS shall adapt to the historical intervention gathered on a target installation.

Metrics:
The simple metrics with "yes" or "no" answer should assess whether the system has a mechanism for detecting a recurring problem. More detailed metrics can examine the ration of correctly detected recurring problems.

3.3.14 WP8_IRN_CR_REQ20

Requirement:
The AdCoS shall adapt to the number of operators available.

Metrics:
The following characteristics should be monitored and evaluated:

- Minimum 90 % of phone calls received with response time under the maximum allowed, i.e., 120 seconds (yes/no)
- Offer to the supervisor capabilities to transfer from one operator to another (yes/no)
- Offer to the supervisor capabilities to monitor current workloads of each operator (yes/no)
- Identification of overloaded operators (yes/no)
- Identification of underloaded operators (yes/no)

3.4 Automotive requirements (WP9)

3.4.1 WP9_TAK_AUT_REQ07

Requirement:
The urgency of a blind-spot warning should be determined and then communicated to the blind-spot audio/visual feedback interface.

Metrics:
The percentage of blind spots for which urgency has been correctly determined and communicated will be measured. Urgency is related to TLC values (Time-to-line or lane crossing), i.e., the lesser TLC the higher the urgency. An increase in urgency could be an increase in blink frequency, an increase in loudness or brightness or a change of color towards red.

3.4.2 WP9_TAK_AUT_REQ08

Requirement:
Blind-spot indicator designs shall reflect the actual situation (speeds and relative positions of objects) and propose appropriate actions to the user by indicating directions and user actions that avoid collision.

Metrics:
This requirement is fulfilled when it can be shown by user tests or expert evaluation, that the indicator design reflects the situation at hand and proposes appropriate reactions to the driver and that reaction times are not delayed over normal reaction times for warning devices. These requirements include different metrics for the different aspects:

1. Indicator design is satisfactory (subjective test by experts).
2. Reactions suggested to the driver are adequate (subjective tests by experts).
3. Reaction times are "good", i.e., not delayed over normal reaction for warning devices (objective, if a target reaction time is defined, then the measured reaction time can be measured and compared against it).

3.4.3 WP9_TAK_AUT_REQ09

Requirement:
Blind-spot detection should be reliable and detection failures (failure to detect object in blind-spot/wrong detection of object in blind-spot) should be minimized. If complete reliance is impossible, prediction of the reliability by the driver should be supported by avoiding irregularity of detection failures. The reliance shall have low specifity and be applicable to all driving conditions.

Metrics:
The requirement is fulfilled when high reliability of detection and low situation specifity of detection rates are shown. We can consider two metrics:

1. True positives (correctly identified objects) – measured as number of correctly identified object over total number of objects in the scenario.
2. Misdetections (detection of objects which are not part of the scenario) – measured as number of objects in the scenario over total number of objects claimed (by the detection system) to be in the scenario.

3.4.4 WP9_TAK_AUT_REQ18

Requirement:
Ideally, the system shall use a combination of critical events, operator performance measures, operator modeling and physiological assessment of the operator to determine timing of automation mode transitions.

Metrics:
This functional requirement is fulfilled, when it can be checked against a more detailed specification that an adaptive automation algorithm incorporates several of the methods proposed above and the system is able to reliably determine the driver's state and from this derive the timing for mode changes of the automation.

3.4.5 WP9_TAK_AUT_REQ25

Requirement:
The urgency of an ACC warning should be measured and then communicated to the ACC audio/visual/haptic feedback interface.

Metrics:
Fulfilment of this functional requirement is shown, when urgency of a situation can be measured by the system and communicated to the warning system. Typical measurements are longitudinal TTC (Time-To-Collision), THW, Human risk feeling (combination of TTC and THW).

3.4.6 WP9_TAK_AUT_REQ27

Requirement:
ACC detection should be reliable and detection failures (failure to detect object in front/wrong detection of object in front) should be minimized. If complete reliance is impossible, prediction of the reliability by the driver should be supported by avoiding irregularity of detection failures. The reliance shall have low specifity and be applicable to all driving conditions.

Metrics:
The requirement is fulfilled when high reliability of detection and low situation specifity of detection rates are shown. We can consider two metrics:
1. True positives (correctly identified objects) – measured as number of correctly identified object over total number of objects in the scenario.
2. Misdetections (detection of objects which are not part of the scenario) – measured as number of objects in the scenario over total number of objects claimed (by the detection system) to be in the scenario.

In addition, a metrics that measure how regular the detection errors are, for example, assuming that a classification ("type") of possible objects to be detected is available (for example: car, pedestrian, tree, ball) we can consider that we aim at a low number of different types of misdetected objects over total number of object types.

3.4.7 WP9_TAK_AUT_REQ35

Requirement:
If the user ignores the lane departure/blind spot warning and continues to steer off road/into traffic the system shall issue an alarm through an appropriate channel. It shall also propose counter-actions to relax the situation.

Metrics:
The requirement is fulfilled when an alarm is triggered by an impending collision caused by inappropriate road behaviour of the user. The requirement can be formalized in TCTL (a timed temporal logic) as:

("lane departure" AND "blind spot warning" AND "user steering") -->
(
AF-before-time t (system gives an alarm) AND
AF-before-time t' (counter action are proposed)
)

where (1) "lane departure" , "blind spot warning", "user steering" are atomic propositions (statements that are either true or false in each system state), and (2) the logical operator "--> " means "implies" (3) "AF-beforetime t" means "for all executions in the future, before time t" which assume time 0 is the one in which the user ignores the lane departure/blind spot warning and continues to steer off road/into traffic. The formula reads as: if the car is departing from the lane, and the blind spot warning is on and the user is steering, then for all possible evolutions of the system it is true that, before t time units, the system gives an alarm and, after the alarm, in all possible executions, a counter action will be proposed within t' time units.

3.4.8 WP9_TAK_AUT_REQ36

Requirement:
If the user senses the automation is decelerating the vehicle and uses the brake pedal, the ACC should be deactivated and longitudinal control re-issued to the driver.

Metrics:
The requirement is fulfilled when the use of the breaking pedal by the user immediately terminates the automation's longitudinal control. In TCTL:

("automation decelerate" and "break pedal pressed") -->
(
AF-before-time t ("automation deactivated" and
"user has longitudinal control")
)

3.4.9 WP9_TAK_AUT_REQ37

Requirement:
If the user senses the automation is decelerating the vehicle and uses the accelerator pedal, the accelerator pedal shall provide a to be determined resistance and a warning shall be issued by the system.

Metrics:
The requirement is fulfilled when the system is able to provide an appropriate feedback to the user about the appropriateness of the user's intentions. In TCTL:

("automation decelerate" and "accelerator pedal pressed") -->
AF-before-time t ("accelerator pedal resists" and "warning is issued")

3.4.10 WP9_TAK_AUT_REQ64

Requirement:
System shall use minimal correction (just the amount necessary to avoid collisions) for user input error. User shall be supported in finding and avoiding false input by preventing input that will lead to undefined conditions, informing about corrections and giving the opportunity to postpone error treatment for non-critical errors.

Metrics:
System should satisfy the following TCTL property:

"input error" and "correction=X" and AF-before-time t "no-collision" -->
for all Y<X: "input error" and "correction=Y" and EF "collision"

In this case the t of "AF-before-time t" could be set equal for the time of a cycle in our decision process.

3.4.11 WP9_IFS_AUT_REQ02

Requirement:
The RTP platform should allow replaying of simulations cases/tests.

Metrics:
The answer "yes" or "no" should be determined by testing (probably not relevant for WP4 neither WP5).

3.4.12 WP9_IFS_AUT_REQ03

Requirement:
Data synchronization coming from different simulation tools (e.g. driver models, car sensors, road environment simulation, AdCoS, etc.) should be recorded in a synchronized way.

Metrics:
No particular metrics is provided. It relies on the data synchronization feature of RT-Maps tool which has been already validated in other contexts.

3.4.13 WP9_IFS_AUT_REQ04

Requirement:
A virtual car shall be able to be dynamically piloted by the driver model.

Metrics:
The answer "yes" or "no" should be determined by testing or one can define a "percentage of success" metrics. Here "dynamically" means "moving", and, for the requirement to make sense, we need to associate to this requirements the use cases of interest that define the traffic scenarios in which the virtual car is successfully driven by the driver model (automation system), and possibly define what successful means (e.g., percentage of the scenario, in meters driven without accidents or percentage of the number of obstacles that were correctly circumnavigated/avoided).

3.4.14 WP9_IFS_AUT_REQ05

Requirement:
Road environments and traffic events shall correspond to the WP9 scenarios, where the driver model can drive a virtual car.

Metrics:
The answer "yes" or "no" should be determined by testing or a percentage of environment features and traffic events available, over the one cited in the WP9 scenarios can be defined.

3.4.15 WP9_IFS_AUT_REQ06

Requirement:
Driver mental model should be built and updated in a synchronized way with the Simulated Road Environment and Event (traffic scenarios).

Metrics:
This requirement can be described as a TCTL formula

("element X on sensors") --> AF-before-time t ("driver model includes X")

which reads as "if an element X is detected by the sensors, then, in all possible behaviours of my driver model, the element X is taken into account by the driver model within t units of time". Of course, one needs to define what "an element" is and what does it mean "to take into account in the driver model" in the context of considered AdCoS.

3.4.16 WP9_IFS_AUT_REQ07

Requirement:
The system should record and use eye-tracking data to assess driver's visual distraction.

Metrics:
The objective of this metric is to assess the ability of the visual distraction module of the driver model to account for eyes tracking data. Correctness of the module can be measured "directly" against input/output pairs, if the distraction module has an associated function, or, if such a function is not available (as it is likely if we introduce learning techniques), "indirectly" with respect to the general behaviour of the driver. Another correctness criterion can be a sort of "sensitivity" analysis, to measure how a change in the eye-tracking data corresponding to visual distraction is reflected in the driver model.

Note:
This requirement has been included as relevant for analysis by model-based and empirical techniques and tools despite it has not been considered in Deliverable D4.1.

3.4.17 WP9_IFS_AUT_REQ08

Requirement:
Analysis (Strore) of eye-tracking data as a mean of assessment of visual distraction of the driver / IFS model

Metrics:
As the previous requirement, it can be measured "directly" against input/output pairs or "indirectly" with respect to the general behaviour of the driver. Some sort of "sensitivity" analysis can be considered here as well.

Note:
This requirement has been included as relevant for analysis by model-based and empirical techniques and tools despite it has not been considered in Deliverable D4.1.

3.4.18 WP9_IFS_AUT_REQ10

Requirement:
Virtual simulation of car sensors (radar, camera, telemeter), as components of AdCos1 to be simulated and tested with the RTP during the Project

Metrics:
Either the simulator of car sensors is included in the AdCos1 or it is not, thus, an answer “yes” or “no” should be determined by testing (probably not relevant for WP4 neither WP5).

3.4.19 WP9_IFS_AUT_REQ11

Requirement:
Algorithms for AdCoS (or support for AdCoS Simulation) TBD: from Target System definition to be simulated to algorithms to be developed for driver monitoring and adaptive & cooperative assistance / HMI

Metrics:
An answer “yes” or “no” should be determined by testing (probably not relevant for WP4 neither WP5).

3.4.20 WP9_IFS_AUT_REQ12

Requirement:
Car-To-Car communication simulation, as component of AdCos2 to be simulated and tested with the RTP during the Project

Metrics:
Either the simulator of car-to-car communication is included in the AdCos2 or it is not, thus, an answer "yes" or "no" should be determined by testing (probably not relevant for WP4 neither WP5).

3.4.21 WP9_CRF_AUT_REQ09

Requirement:
When the driver has indicated his/her intention to change lane and there is not a side lane, or there is a side obstacle, or there is an incoming obstacle from the rear on the side lane, the driver should be warned so that he/she does not start the lane change maneuver. Driver's state shall be considered as well.

Metrics:
It can be measured as a number of given warnings over the total number of warnings that should have been given.

3.4.22 WP9_CRF_AUT_REQ16

Requirement:
When the driver is facing at the same time with more conditions that could generate an indication or a warning from the system, only the most critical indication should be given to the driver.

Metrics:
It can be measured as a number of indications of the most critical conditions over the total number of generated indications. A measure for criticality needs to be provided.

3.4.23 WP9_OFF_AUT_REQ02

Requirement:
The Bayesian driver model must be able to update its initial (offline) learned parameters using inputs of the driver (steering angle, brake pedal position, throttle position) and available sensor data while driving assisted.

Metrics:
Simple metrics with answer "yes" or "no" based on a simple check of whether the parameters have been changed according to the plan, e.g., *parameter_new = parameter_old + update*. More complex metrics can be based on computation of correct classification rate of appropriate samples for updating. It should be at least 90 %.

3.4.24 WP9_OFF_AUT_REQ04

Requirement:
The Bayesian driver model must be able to return meaningful results after a fixed amount of computation time.

Metrics:
Mean percentage of how accurately the provided approximative result approximates the exact belief state.

3.4.25 WP9_OFF_AUT_REQ09

Requirement:
After an initial offline learning phase, the driver model must be able to classify the currently shown driving style (e.g. aggressive, sporty, ecp, normal) with a Correct Classification Rate (CCR) of (80÷85)% and provide information about the driver's profile (e.g., mean speed, mean TTC).

Metrics:
The number of correctly identified classifications on a test set (True Positives + True Negatives) over the total number of classifications performed (True Positives + False Positives + False Negatives + True Negatives) should be at least 85 %.

3.4.26 WP9_IAS_AUT_REQ06

Requirement:
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.

Metrics:
The number of overrides over the number of attempted overrides should be 100 % as this is the safety feature.

3.4.27 WP9_IAS_AUT_REQ10

Requirement:
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.

Metrics:
This requirement can be described as a TCTL formula

("automation on" AND "steering wheel manually activated") -->
(
AF-before-time t ("automation complete" AND
"automation considers driver will")
)

3.4.28 WP9_IAS_AUT_REQ11

Requirement:
The automated vehicle shall be able to change the lane for an overtaking maneuver. It shall adapt the speed according to the traffic in the neighboring lane and maintain a safe spacing to other traffic participants.

Metrics:
This requirement can be described as a TCTL formula

("overtaking action on") -->
A ("safe spacing" U-before-time t "change lane action completed")

where *"A (prop1 U-before-time-t prop2)"* means that, for all executions, *prop1* is true at least until we come to a state in which *prop2* is true.

4 Techniques and tools for evaluating metrics

This section briefly describes model-based (WP4) and empirical (WP5) techniques and tools that could be used for evaluation of the metrics defined above. More detailed information about the techniques and tools can be found in other documents (e.g., Deliverable D4.2 – Plan for Integration of Model-Based Analysis Techniques and Tools). The list of techniques and tools is not considered as fixed now as it can be extended later by existing tools as well as new tools developed or employed due to the project needs. One can also expect that the mapping between metrics connected with particular requirements and techniques and tools – provided at the end of this section – will be enhanced during following evaluation of AdCoS designs and implementations using model-based and empirical techniques and tools.

4.1 Model-based techniques and tools

Model-based techniques and tools offer different approaches for addressing the requirement evaluation – from model checking represented here by GreatSPN, through formal simulation of the system represented here by RTMaps + ProCIVIC + COSMO-CIVIC and CoSimECS + CASCaS, to abstract interpretation represented here by AnaConDa + Race Detector & Healer + SearchBestie. The model checking and the formal simulation approaches are suitable mainly to evaluate a design of AdCoS using a model of the developed system while AnaConDa and Race Detector and Healer are suitable for evaluation of the system implementation.

4.1.1 GreatSPN

GreatSPN is provided by University of Turin, Department of Computer Science. GreatSPN is a suite of tools for modelling, validation, optimization, and performance evaluation of complex systems using Generalized Stochastic Petri Nets and their extensions such as, for instance, Stochastic Well-formed Nets and Markov Decision Petri Nets. It provides a user friendly framework to experiment with stochastic Petri net based modelling techniques and thanks to the implementation of efficient analysis algorithms it can be used also to study real complex applications. It is intended to be used for model checking of AdCoS designs.

4.1.2 RTMaps + ProCIVIC + COSMO-CIVIC

These tools are introduced in the same section because we plan to use them jointly in order to support design, development and evaluation techniques of AdCoS. While RTMaps and ProCIVIC can be used independently of the application domain (equally on WP6, WP7, WP8 and WP9 application domains), COSMO-CIVIC is dedicated to automotive system domain.

In particular, **RTMaps** is provided by INTEMPORA. It is intended to be used to support design and development phases of AdCOS, and specifically for WP4 their verification and validation through the evaluation technique. RTMaps is a rapid and modular development environment for real-time multi-sensor (more generally multi-I/O) applications. It allows to very easily acquire and process data from various data sources such as cameras, audio, eye-trackers, biometric sensors, motion capture, CAN bus, GPS, IMUs, Lidars, Radar, etc. It provides data samples time-stamping functionalities and allows real-time recording and playback of the data for post-analysis, self-confrontation, and so on. It provides a graphical environment for rapid development based on existing components, and a C++ SDK for integration of third-party libraries into components.

ProSIVIC is provided by CIVITEC and is intended to support, in the context of WP4, the evaluation technique. ProSIVIC is a modelling and simulation software for 3D environments and multi-frequency sensors such as cameras, Lidars, Radars, IMUs, GPS etc. It helps designing and validating applications from the early development stages. It is oriented towards embedded systems with perception capabilities, with or without human interaction in the simulation.

COSMO-CIVIC is provided by IFSTTAR and is intended to support, in the context of WP4, the evaluation techniques. COSMO-SIVIC is a simulation research tool designed during the ISI-PADAS project (2008-2011), integrating a driver model (named COSMODRIVE for COgnitive Simulation MOdel of the DRIVEr) able to drive a virtual car within a virtual environment (based on a SiVIC precommercial version of ProSIVIC). During the HoliDes project, we plan to interface this research tool with ProSIVIC and RTMaps, in order to support virtual simulation of future AdCoS use by human drivers (simulated by COSMODRIVE) and then to support a “Human Centered Design approach” of Cooperative driving Aids in WP9. Moreover, COSMO-SIVIC could be also used as a driving simulator, for implementing experiments and tests among real human drivers.

4.1.3 CoSimECS + CASCaS

CoSimECS is provided by OFFIS and is intended to support the evaluation techniques. CoSimECS is a tool, allowing the specification of a system in terms of agents, tasks and resources, simulation in terms of assigning simulators for the agents and resources, as well as automated generation of configuration files.
CoSimECS also allows setting up and controlling the simulation, based on the OFFIS simulation platform. The OFFIS simulation platform is based on the High Level Architecture standard (IEEE1516). Development of CoSimECS has been started in D3CoS, and will be continued in HoliDes. Currently it has been only used internally at OFFIS, but when tool maturity allows, a release to the OFFIS partners using CASCaS for evaluation and simulation is planned.

CASCaS is provided by OFFIS and is intended to support the evaluation technique. CASCaS is a cognitive architecture, intended to allow simulation of human behaviour, based on psychological and physiological sound models of human behaviour. When connected to a simulator, CASCaS performs actions and made decisions by applying a procedures model based on the current status of the simulation.

4.1.4 AnaConDa + Race Detector & Healer + SearchBestie

These tools are provided by VeriFIT research group from Brno University of Technology and are intended to support checking of concurrent software. ANaConDA is a framework that simplifies the creation of dynamic analysers for analysing multi-threaded C/C++ programs on the binary level. The Java Race Detector & Healer is a prototype for a run-time detection and healing of data races and atomicity violations in concurrent Java programs. SearchBestie (Search-Based Testing Environment) is a generic infrastructure that is designed to provide environment for experimenting with applying search techniques in the field of program testing (e.g. to find optimal settings of injected noise to increase efficiency of AnaConDa and Race Detector & Healer).

4.2 Empirical techniques and tools

Empirical techniques and tools are in many cases less precise than techniques and tools based on models with precisely defined semantics but can help in situations where model-based tools are not applicable. Thus, these techniques and tools provide a complementary approach to model-based techniques and tools. In the following, several tools and techniques for empirical evaluation are identified, including simulators, a machine learning tool, techniques for monitoring physiological characteristics and techniques based on individual evaluation.

4.2.1 Flight simulator

Flight simulator can provide a very realistic simulation of aircraft cockpits, however, it is quite expensive evaluation approach. Thus, it should be used as a complement of other techniques and tools, mainly for final evaluation of aeronautics requirements (WP7). Access to flight simulator can be managed by Honeywell.

4.2.2 Driving simulator

Driving simulator simulates driver environment and thus it is planned for evaluation of automotive requirements (WP9).

4.2.3 Theatre technique

Theatre technique is some sort of "Wizard of Oz"-technique [6] where two electronically and/or mechanically coupled seat boxes let a person in one seat box imitate a driver assistance system, while a subject in the other seat box experiences these effects. This technique is intended for evaluation of system designs and is provided by DLR.

4.2.4 Weka

Weka is a collection of machine learning algorithms for data mining tasks. It contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. Weka [7] is open source software developed by The University of Waikato, New Zealand.

4.2.5 Eye tracking

Eye tracking can measure a percentage of dwell time for a certain area of interest.

4.2.6 Electroencephalography

Electroencephalography (EEG) can be used to determine what a human operator concentrates on. To provide useful results, the evaluation of neural activity should be adjusted for particular application.

4.2.7 Pupillometry

Pupillometry is based on measuring changes in pupil diameter that can be used for estimation of human workload.

4.2.8 User questionnaire

Questionnaires allow users to rate the system under investigation from usability, acceptance, attractiveness and other rather subjective points of view.

4.2.9 Expert review

Expert review is similar to questionnaire technique based in a sense that is based on human opinion about the system, however, there is a difference in people involved in the evaluation and the freedom the evaluators have.

4.3 Mapping between the requirements and the techniques and tools

The first attempt to provide mapping between the metrics associated with requirements and the techniques and tools that could be used for verification of the requirements based on the associated metrics is shown in the following tables.

The symbol “x” means that that the technique or tool (in the column) seems suitable for verification of the requirement (in the row), the symbol “(x)” means that the technique or tool (in the column) may be used for verification of the requirement (in the row) while the symbol “-” means that the technique or tool (in the column) seems as not applicable for verification of the requirement (in the row). This mapping does not imply that the tool is finally used in HoliDes for verification of the requirement; it is just a suggestion that it could be used. The final decision if the tool is used needs more detailed discussion, as other aspects like effort of the partner providing the tool has to be taken into account, too.

4.3.1 Health AdCoS

Mapping between WP6 requirements and techniques and tools that can be used for their verification:

Requirement ID	GreatSPN	RTMaps + ProCIVIC + COSMO-CIVIC	CoSimECS + CASCaS	AnaConDa + Race Detector & Healer	Flight simulator	Driving simulator	Theatre technique	Weka	Eye tracking	Electroencephalography (EEG)	Pupillometry	User questionnaire	Expert review
WP6_ATO_HEA_REQ25	(x)	(x)	-	(x)	-	-	-	-	-	-	-	(x)	(x)
WP6_AWI_HEA_REQ01	(x)	(x)	-	-	-	-	-	(x)	-	-	-	(x)	x
WP6_AWI_HEA_REQ03	-	-	x	-	-	-	-	-	-	-	-	(x)	x
WP6_AWI_HEA_REQ07	(x)	(x)	-	-	-	-	-	(x)	-	-	-	(x)	x
WP6_AWI_HEA_REQ11	(x)	(x)	-	-	-	-	-	(x)	-	-	-	(x)	x
WP6_AWI_HEA_REQ25	(x)	(x)	-	-	-	-	-	(x)	-	-	-	(x)	x
WP6_AWI_HEA_REQ35	-	-	x	-	-	-	-	(x)	-	-	-	(x)	(x)
WP6_AWI_HEA_REQ43	(x)	(x)	-	-	-	-	-	(x)	-	-	-	(x)	x
WP6_IGS_HEA_REQ03	(x)	(x)	(x)	-	-	-	-	(x)	-	-	-	(x)	x
WP6_IGS_HEA_REQ09	-	-	(x)	-	-	-	-	-	-	-	-	(x)	(x)
WP6_IGS_HEA_REQ10	-	-	-	-	-	-	-	-	-	-	-	(x)	(x)
WP6_PHI_HEA_REQ08	-	-	(x)	-	-	-	-	-	-	-	-	(x)	(x)

4.3.2 Aeronautics AdCoS

Mapping between WP7 requirements and techniques and tools that can be used for their verification:

Requirement ID	GreatSPN	RTMaps + ProCIVIC + COSMO-CIVIC	CoSimECS + CASCaS	AnaConDa + Race Detector & Healer	Flight simulator	Driving simulator	Theatre technique	Weka	Eye tracking	Electroencephalography (EEG)	Pupillometry	User questionnaire	Expert review
WP7_HON_AER_REQ30	-	(x)	x	-	x	-	-	-	(x)	(x)	-	x	(x)
WP7_HON_AER_REQ44	-	(x)	-	-	x	-	-	-	(x)	(x)	-	x	x
WP7_HON_AER_REQ45	x	(x)	-	(x)	x	-	-	-	(x)	(x)	(x)	x	(x)
WP7_HON_AER_REQ46	x	(x)	-	(x)	x	-	-	-	(x)	(x)	(x)	x	(x)
WP7_HON_AER_REQ49	-	(x)	x	-	x	-	-	-	(x)	(x)	(x)	x	(x)
WP7_HON_AER_REQ66	-	(x)	x	-	x	-	-	-	-	-	-	(x)	x
WP7_HON_AER_REQ71	-	(x)	x	-	(x)	-	-	-	-	-	-	(x)	x
WP7_HON_AER_REQ78	-	(x)	x	-	(x)	-	-	-	-	-	-	(x)	x

4.3.3 Control rooms AdCoS

Mapping between WP8 requirements and techniques and tools that can be used for their verification:

Requirement ID	GreatSPN	RTMaps + ProCIVIC + COSMO-CIVIC	CoSimECS + CASCaS	AnaConDa + Race Detector & Healer	Flight simulator	Driving simulator	Theatre technique	Weka	Eye tracking	Electroencephalography (EEG)	Pupillometry	User questionnaire	Expert review
WP8_ADS_CTR_REQ17	(x)	(x)	x	-	-	-	-	(x)	-	-	-	(x)	x
WP8_ADS_CTR_REQ18	(x)	(x)	x	-	-	-	-	(x)	-	-	-	(x)	x
WP8_ADS_CTR_REQ22	-	(x)	-	-	-	-	-	(x)	-	-	-	(x)	x
WP8_IRN_CR_REQ04	(x)	(x)	-	-	-	-	-	(x)	-	-	-	(x)	x
WP8_IRN_CR_REQ09	-	(x)	-	-	-	-	-	(x)	-	-	-	(x)	x
WP8_IRN_CR_REQ10	(x)	(x)	-	-	-	-	-	(x)	-	-	-	(x)	x
WP8_IRN_CR_REQ11	(x)	(x)	-	-	-	-	-	(x)	-	-	-	(x)	x
WP8_IRN_CR_REQ12	(x)	(x)	-	-	-	-	-	(x)	-	-	-	x	x
WP8_IRN_CR_REQ13	(x)	(x)	-	-	-	-	-	(x)	-	-	-	x	x
WP8_IRN_CR_REQ15	(x)	(x)	-	-	-	-	-	(x)	-	-	-	x	x
WP8_IRN_CR_REQ17	(x)	(x)	-	-	-	-	-	(x)	-	-	-	x	x
WP8_IRN_CR_REQ18	(x)	(x)	-	-	-	-	-	(x)	-	-	-	x	x
WP8_IRN_CR_REQ19	(x)	(x)	-	-	-	-	-	(x)	-	-	-	x	x
WP8_IRN_CR_REQ20	(x)	(x)	-	-	-	-	-	(x)	-	-	-	(x)	x

4.3.4 Automotive AdCoS

Mapping between WP9 requirements and techniques and tools that can be used for their verification – 1st part:

Requirement ID	GreatSPN	RTMaps + ProCIVIC + COSMO-CIVIC	CoSimECS + CASCaS	AnaConDa + Race Detector & Healer	Flight simulator	Driving simulator	Theatre technique	Weka	Eye tracking	Electroencephalography (EEG)	Pupillometry	User questionnaire	Expert review
WP9_TAK_AUT_REQ07	-	(x)	(x)	-	-	x	(x)	(x)	(x)	(x)	(x)	(x)	x
WP9_TAK_AUT_REQ08	-	(x)	-	-	-	x	(x)	(x)	(x)	(x)	(x)	(x)	x
WP9_TAK_AUT_REQ09	-	(x)	-	-	-	x	(x)	(x)	(x)	(x)	(x)	(x)	x
WP9_TAK_AUT_REQ18	-	(x)	x	-	-	x	(x)	(x)	(x)	(x)	(x)	(x)	x
WP9_TAK_AUT_REQ25	-	(x)	-	-	-	x	(x)	(x)	(x)	(x)	(x)	(x)	x
WP9_TAK_AUT_REQ27	-	(x)	-	-	-	x	(x)	(x)	-	-	-	(x)	x
WP9_TAK_AUT_REQ35	x	(x)	(x)	-	-	x	(x)	(x)	(x)	(x)	(x)	(x)	(x)
WP9_TAK_AUT_REQ36	x	(x)	(x)	-	-	x	(x)	(x)	(x)	(x)	(x)	(x)	(x)
WP9_TAK_AUT_REQ37	x	(x)	(x)	-	-	x	(x)	(x)	(x)	(x)	(x)	(x)	(x)
WP9_TAK_AUT_REQ64	x	(x)	(x)	-	-	x	(x)	(x)	(x)	(x)	(x)	(x)	(x)
WP9_IFS_AUT_REQ02	-	-	-	-	-	-	-	-	-	-	-	(x)	x
WP9_IFS_AUT_REQ03	-	x	x	-	-	-	-	-	-	-	-	(x)	x
WP9_IFS_AUT_REQ04	-	(x)	x	-	-	x	(x)	-	-	-	-	-	x
WP9_IFS_AUT_REQ05	(x)	(x)	x	-	-	x	(x)	-	-	-	-	(x)	x
WP9_IFS_AUT_REQ06	x	(x)	x	-	-	x	(x)	(x)	-	-	-	(x)	(x)

Mapping between WP9 requirements and techniques and tools that can be used for their verification – 2nd part:

Requirement ID	GreatSPN	RTMaps + ProCIVIC + COSMO-CIVIC	CoSimECS + CASCaS	AnaConDa + Race Detector & Healer	Flight simulator	Driving simulator	Theatre technique	Weka	Eye tracking	Electroencephalography (EEG)	Pupillometry	User questionnaire	Expert review
WP9_IFS_AUT_REQ07	-	(x)	-	-	-	x	(x)	(x)	(x)	(x)	(x)	(x)	(x)
WP9_IFS_AUT_REQ08	-	(x)	-	-	-	x	(x)	(x)	(x)	(x)	(x)	(x)	(x)
WP9_IFS_AUT_REQ10	-	-	-	-	-	x	(x)	-	-	-	-	(x)	x
WP9_IFS_AUT_REQ11	-	-	-	-	-	x	(x)	-	-	-	-	(x)	x
WP9_IFS_AUT_REQ12	-	-	-	-	-	x	(x)	-	-	-	-	(x)	x
WP9_CRF_AUT_REQ09	-	-	-	-	-	x	(x)	-	-	-	-	(x)	x
WP9_CRF_AUT_REQ16	-	-	-	-	-	x	(x)	-	-	-	-	(x)	x
WP9_OFF_AUT_REQ02	-	-	-	-	-	(x)	-	-	-	-	-	(x)	x
WP9_OFF_AUT_REQ04	-	-	-	-	-	(x)	-	-	-	-	-	(x)	x
WP9_OFF_AUT_REQ09	-	-	-	-	-	(x)	-	-	-	-	-	(x)	x
WP9_IAS_AUT_REQ06	(x)	(x)	(x)	(x)	-	x	(x)	(x)	-	-	-	(x)	(x)
WP9_IAS_AUT_REQ10	x	(x)	-	-	-	x	(x)	(x)	-	-	-	-	-
WP9_IAS_AUT_REQ11	x	(x)	-	-	-	x	(x)	(x)	-	-	-	-	-

5 Conclusions

This document defines metrics for selected requirements and identifies which metrics could be evaluated by which model-based and/or empirical analysis techniques and tools.

The definition of metrics as well as the assignment between the metrics and the techniques and tools can be refined later in the project due to several reasons, including refinement of requirements (e.g., more precise specification of expected AdCoS behaviour), selection of additional requirements for model-based and/or empirical analysis, identification of another useful techniques or tools that already exist or developing new tools that address the needs of the project, development of human operators models (within Task T2.4) and human-machine interaction models (within Task T2.5). The future improvements of this document are fully consistent with the planned timing of the work packages as the duration of Task T4.2 is planned from Month 4 till Month 18. This deliverable is planned at the end of Month 7 and thus, it can capture only the first results of the task.

References

[1] Victor Basili. Software Modeling and Measurement: The Goal/Question/Metric Paradigm http://dl.acm.org/citation.cfm?id=137076

[2] Peter Farrel-Vinay. Manage Software Testing. CRC Press. 2008. http://books.google.cz/books/about/Manage_Software_Testing.html?id=ijQElKfBaHEC

[3] ISO/IEC TR 9126-2:2002. http://www.cse.unsw.edu.au/~cs3710/PMmaterials/Resources/9126-2%20Standard.doc

[4] Techrep Johansson2004 from WP01-T1.3 Regulations Database. http://www.aide-eu.org/pdf/sp2_deliv_new/aide_d2_2_1.pdf

[5] Hong Zhu, Patrick A. V. Hall, John H. R. May. Software unit test coverage and adequacy. In ACM Computing Surveys, vol. 29, 1997.

[6] Wikipedia. *Wizard of Oz experiment*. Last modified on 2 January 2014 at 14:24. http://en.wikipedia.org/wiki/Wizard_of_Oz_experiment

[7] Machine Learning Group at the University of Waikato. Weka 3: Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/weka/

*** End of document ***