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

D1.4 - HF-RTP Version 1.0 incl. Methodology and Requirements Analysis Update

Project Number:	332933
Classification:	Public
Work Package(s): **Milestone:** **Issue Date:** **Document Timescale:**	WP 1 M 2 <13.02.2015> Project Start Date: October 1, 2013 Written Oct 2014 - Feb 2015
Deliverable Overview:	**Main document:** D1.4 – HF-RTP Version 1.0 inc. Methodology and Requirements Analysis update. <Confidential>
Compiled by:	Lars Weber, OFF
Authors:	Lars Weber, OFF Nacho González, ATO Jordi Fonoll, ATO Jose Daniel Gómez de Segura, TEC Sonia Bilbao Arechabala, TEC Sara Sillaurren, TEC Svenja Borchers, TWT Denis Martin, TWT Ian Giblett, Airbus-UK Dorota Gardas-Schmid, HFC Harald Kolrep, HFC Nicolas Schneider, EAD-FR
Reviewers:	Internal Reviewer: Svenja Borchers, TWT External Reviewer: Zdenek Moravek, HON
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

[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	Status Description	Author
12.09.2014	Started writing initial document	OFF
04.12.2014	First text version for Task T1.6 CMM	OFF
09.01.2015	Added section for Task 1.5 / 1.7 RTP Workflows / IOS	ATOS / TEC
09.01.2015	Received updated Annex with Requirements Table T1.1.	TWT
13.01.2015	Added baseline section	HFC
15.01.2015	Added HF Integration section	HFC
16.01.2015	Adaptation Framework Meta Model	OFF, EAD-FR
23.01.2015	Final contributions for internal review	OFF, HFC, TWT, EAD-UK
23. - 29.01. 2015	Internal Review	TWT
29.01.2015	Updated WP6-9 development workflows UML diagrams	ATOS
30.01-05.02. 2015	External Review	HON
05.02-11.02	Changes to the sections based on reviews	ALL
11.02	Changes due to the reviews integrated into final document	OFF
13.02	Delivery Final public and confidential versions to the EU	OFF

Table of Contents

Introduction

This document describes the progress of the HoliDes consortium to develop a Human Factors Reference Technology Platform (HF-RTP), at its version 1.0. In software industry, version number 1.0 typically refers to the first official release of a new piece of software or tool. Thus, it seems to be appropriate to describe the overall approach which led to this important milestone of the project.

Before any details are explained it should be clarified: "What is the HoliDes HF-RTP?" According to the proposal, the HF-RTP developed in HoliDes consists of a set of *processes*, *techniques* and *software tools*, which can be used for the development of *Adaptive Cooperative Human-Machine Systems* (AdCoS) according to domain specific regulations and standards. The three components (processes, techniques and software tools) are briefly explained below.

All four application domains of HoliDes (Health, Aeronautics, Control Rooms and Automotive) are safety and security critical. Each of them has its own regulations and standards for the development of systems. This imposes a lot of tool qualification effort on the tool vendors and product certification effort on system developers. Thus, an essential part of the HF-RTP is to consider system development compliant to standards and regulations. Previous RTP related projects like CESAR and MBAT or the current project Crystal deal with development process topics from the perspective of system architecture modelling and safety engineering. HoliDes instead pays attention to the integration of *human factors into the engineering processes*.

The term *technique* refers to specific development and analysis techniques, like for example model-based or contract-based design of hardware and software. There are dozens of traditional engineering techniques which are worth to be considered, but HoliDes specifically cares about the integration of *human factors techniques* (e.g. task analysis / task modelling, usability studies, workload impact analysis) into traditional system design processes.

The general purpose of *software tools* is to support an engineer in his development and analysis tasks. Especially in the human factors domain a lot of techniques are not strictly formalized and not supported by specific tools. Results are often stored in descriptive paper form or excel sheets. One of the major goals of the HF-RTP developed in HoliDes is to close the gap between 1) the engineering disciplines which are to a large extent supported by tools and computer aided techniques and 2) the human factors discipline.

Figure 1 below is taken from the HoliDes proposal and shows the cyclic approach for the development of the HF-RTP. With this deliverable the end of the first cycle at milestone M2 is reached, HF-RTP version 1.0 is delivered and the project base line is defined.

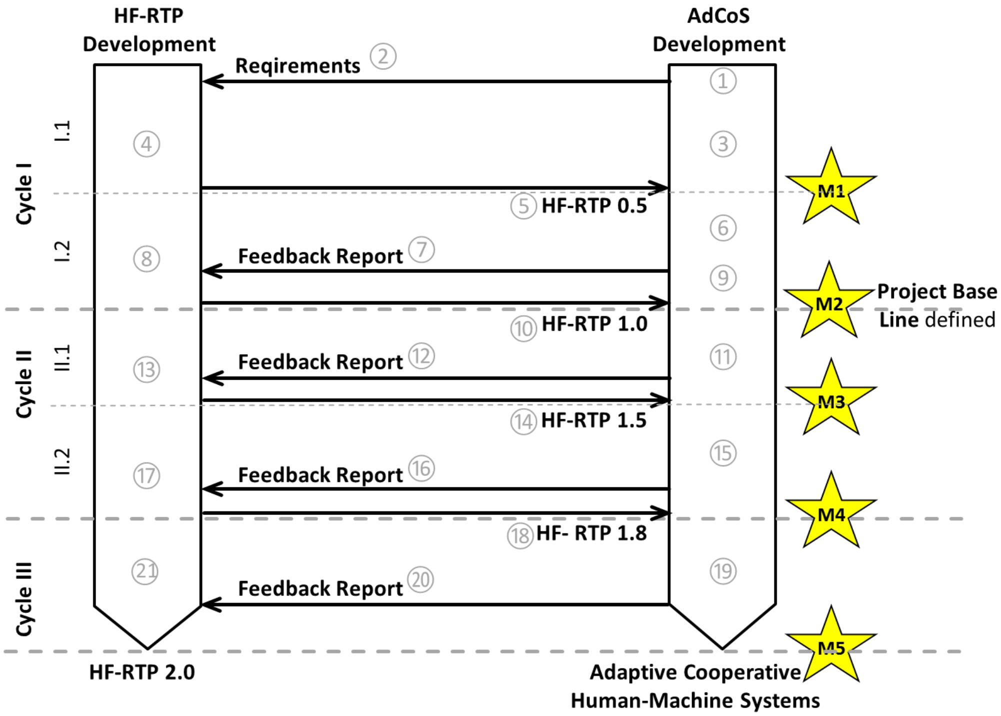

Figure 1: Overall workflow taken from the HoliDes proposal.

The steps of the approach outlined in Figure 1 will be explained over the course of this document. Chapter 1 will give the reader an overview of the HoliDes approach which led to the first version of the HF-RTP: version 0.5. Afterwards, chapter 2 will deal with the effort that was invested to upgrade the HF-RTP version from 0.5 to 1.0.

1 HoliDes HF-RTP version 0.5

Besides the management and dissemination work packages (WP10, WP11), the research work packages of HoliDes are split into two categories (see Figure 2): the research work packages (WP 1-5) which take care of the development and integration of processes, techniques & tools into the HF-RTP, and the application work packages (WP 6-9), which have a need to integrate new technology (human factors methods and supporting software tools) into their development processes.

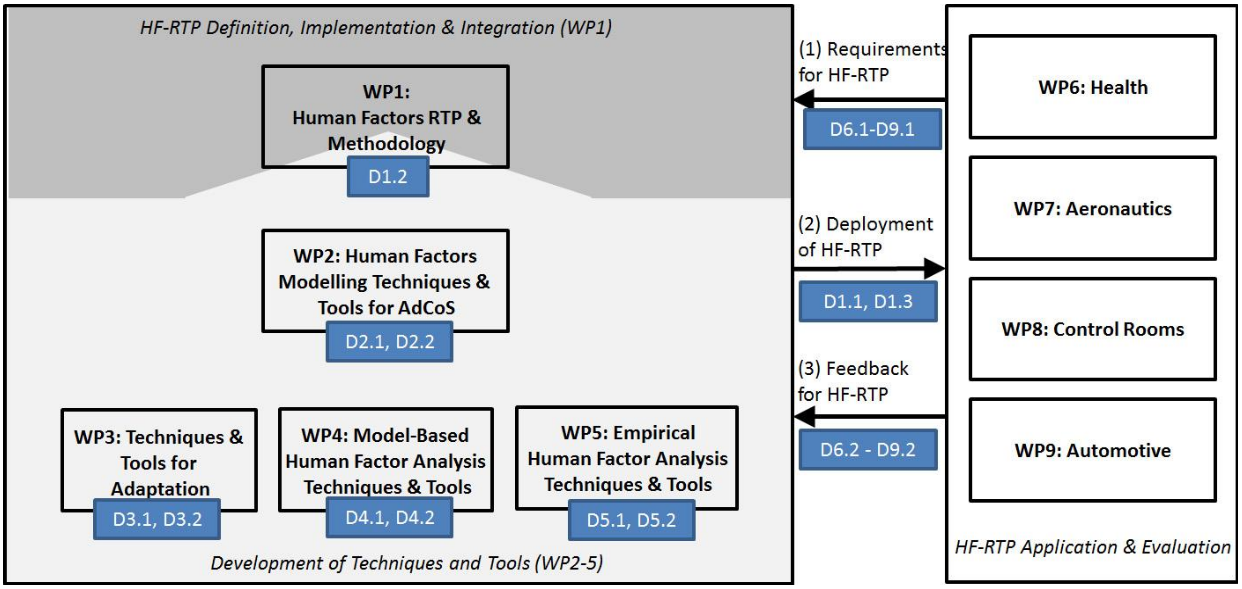

Figure 2: Research work packages (left) and application work packages (right); Blue boxes show the deliverables that were exchanged until HF-RTP version 0.5. (1) Domain specific requirement deliverables. (2) D1.1 HF-RTP Requirements; D1.3 HF-RTP version 0.5 including workflows; (3) D6.2-D9.2 workflow details. Other deliverables were used WP internally. Hint: to simplify the diagram, the interaction between the research work packages 1-5 was removed compared to the diagram in the proposal, but it still exists.

The interaction between research and application work packages as visualized by Figure 2 contains three phases (1) - (3):

1. *Requirements for HF-RTP*: the first action taken by the consortium was the specification of requirements driven by the application domains (deliverables D6.1, D7.1, D8.1 and D9.1). Besides the specification of application scenarios, use-cases and technical requirements for the AdCoS, the first four deliverables contained a number of high-level requirements on tools and interoperability issues, as well as on the domain specific development processes and standardization issues.
2. *Deployment of HF-RTP*: the research work packages started their work based on the input of all four requirement deliverables. WP1 categorized them into domain specific / independent as well as HF-RTP related requirements. Based on this categorization WP1 defined a first set of cross-domain HF-RTP requirements in D1.1, while WP2-5 investigated the requirements according to the question: "Where can a method or tool

developed in our WP be applied?" An overview of tools and techniques as well as a mapping between requirements and tools was documented in the deliverables D2.1 – D5.1. WP 2-5 also presented their methods and tools on a workshop and several webinars which was useful for the industrial partners to identify possible candidate tools that could be applied in their development process. The most important *output of the deployment step* was deliverable D1.3 which was a joint effort by all nine work packages. For each domain, it contains a number of development workflows for the design of the different AdCoS. Each of those workflows tackles specific human factors problems that should be solved by including new methods and software tools from WP2-5. Additionally, D1.3 describes the technical backbone of the HF-RTP (based on OSLC), but no tool chains were implemented.

3. *Feedback for HF-RTP*: The workflows described in D1.3 were specified in more detail by the industrial partners in the deliverables D6.2 - D9.2.

With the delivery of the feedback (D6.2-D9.2) and the evaluation of the HF-RTP version 0.5 the conceptual development of the RTP was outlined. As cornerstones for the development of HF-RTP version 1.0 the following deliverables were used:

- D1.3 HF-RTP version 0.5
- D1.2 Human Factors Integration Concepts and Regulations
- D2.2-D5.2 Brief method and tool descriptions (without any relation to workflows)
- D6.2 - D9.2 Feedback about the workflows and HF-RTP version 0.5 (D1.3)

2 HoliDes HF-RTP version 1.0

While the previous deliverables of WP1 were all dedicated to a specific topic, e.g. Requirements (D1.1) or Human Factors Integration Concepts (D1.3), this deliverable is organized according to the different tasks of WP1. The sections are ordered according to the tasks T1.1 - T1.8 and address the goals originally stated in the proposal with the only exception of Task 1.3. Its results have already been fully reported in deliverable D1.2.

Tasks of WP 1

T1.1	Analysis of Requirements
T1.2	Review of existing Human Factor Integration Concepts & Commercial Tools
T1.3	Review of Human Factor & Safety Regulations
T1.4	Integration of Review Results & Definition of the AdCoS Methodology and HF-RTP Architecture
T1.5	Definition & Implementation of the HF-RTP Processes and Work Flow
T1.6	Definition & Implementation of the HF-RTP Meta Model
T1.7	Definition & Implementation of the HF-RTP Infrastructure & Interoperability Standard & Generic Services
T1.8	Definition of HF-RTP Tailoring Rules & Deployment of the HF-RTP to WP6-9

The first section of this document starts with an update of the requirements. In HoliDes three different categories of requirements exist: 1) requirements for the AdCoS (WP 6-9), 2) requirements for the *processes*, *techniques* and *software tools* (WP 2-5), and 3) requirements for the HF-RTP (WP 1). In this document we will only refer to the latest category of requirements. The requirements will guide the developments of the project. Afterwards subsection 0 of this deliverable describes the *project baseline*, its purpose and use. The baseline contains a set of measurements to evaluate the progress and results of the project according to a set of *Key performance Indicators* (KPIs). It is the result of the work package and not a specific task.

The subsequent sections 2.4 - 2.8 contain the parts dedicated to the development of the HF-RTP, starting with the identification of HF concepts and methods. In Section 2.4 the terms *complexity* and *determinism* are discussed concerning potential problems that arise during the development and certification of AdCoS. Section 2.5 contains a UML representation of the development workflows as they were specified in the deliverables D6.2 - D9.2 by the application work packages.

For the development of the HF-RTP the identified HF concepts of section 0 are part of the *Meta-Modelling* section. The goal of this Task (1.6) is to formalize these concepts step by step. The different parts of the meta-model can be understood as an abstract data type definition which can be used afterwards to implement concrete tool adaptors. These adaptors are necessary to exchange data between different software tools. Section 2.7 will shift the focus on the concrete implementation issues for an HF-RTP, while section (2.8) deals with tailoring a concrete instance of the RTP for a development process. Finally, we address the reviewer's comments from the first year project review in section 2.9.

2.1 HF-RTP Requirements Update

The initial version of the HF-RTP requirements was a collection of cross-domain requirements, i.e. requirements coming from more than one application domain investigated within HoliDes. With the initial version of the HF-RTP and the collected application domain workflows in D1.3, the initial HF-RTP requirements have been refined to be more precise and targeted exclusively at the HF-RTP. This refinement includes a more detailed description of existing requirements, the addition of new requirements, and the removal of some requirements.

In addition to these general refinements, categories were defined to support the requirements management process and to help deciding whether a requirement is relevant for the HF-RTP. Requirements for the HF-RTP either address the methods, techniques and tools (MTTs) that will be connected to the RTP, the interface of these MTTs, the workflow (i.e., dependencies between MTTs) and finally the concepts that are specific to the Human Factors (HF) domain (addressing the HF issues mentioned in Section 0 and in D1.1). These categories are described in more detail in Table 1.

Requirement Type	Description	Examples
MTT	MTT RTP-requirements describe what MTTs for design / experimentation / simulation / modelling must be part of the RTP.	(1) The RTP must provide a tool X for modelling / simulation task Y. (2) The RTP must provide a model X for describing human factor Y (e.g., a model for human visual distraction, a model for human cognitive distraction). (3) Methods which do not use tools as well as their inputs and outputs can be described in compliance to the HF ontology.
Interface	Interface RTP-requirements are targeted to the common interface description (e.g., input/output) and / or technology between design-time	(1) All RTP tools must implement an OSLC compatible interface. (2) Data X of tool A must be provided as common data type Y (Y to be defined by the RTP). Tool B then

	/ workflow tools.	can interpret this data type and convert it to its internal data type. Example: Timestamp formats and timestamp basis.
Workflow	Workflow RTP-requirements describe needs imposed on the development cycle / life-cycle used in tailored versions of the RTP.	(1) Traceability / logging of data exchanged between workflow tools. (2) Steps within the workflow can be identified and related activities attributed.
Concept	Concept RTP-requirements describe necessary human factors for modelling / describing human state / behaviour in specific scenarios. These address HF issues, e.g., workload, distraction / attention, situation awareness. These definitions (or even models) on the one hand help integrating MTTs into the RTP and help defining the interfaces / data descriptions. On the other hand, these concepts can be used for developing new models for human factors.	(1) The RTP must provide taxonomies describing human factors (e.g., for human visual distraction, for human cognitive distraction). (2) The RTP must describe the relation between workload and distraction.

Table 1 Categories of requirements that are relevant for the HF-RTP.

Requirements concerning the run-time of an AdCoS (e.g., the AdCoS must be able to process data of source X in real-time) or specific MTT requirements (e.g., an MTT providing X must have property Y) are requirements that are not relevant for the HF-RTP and are thus not described here.

A refinement of the first version of HF-RTP requirements (compare D1.1) can be found in Table 2. Each requirement is assigned at least to one of the categories described in Table 1 in order to structure the requirements and to verify their relevance for the HF-RTP. Requirements that could not be assigned to one of the categories were removed (example: WP1_HFRTP_REQ 06, 07, 09, and 10 describe AdCoS requirements and, thus, were removed). For all requirements more details were added, either by an elaborated description, by additional rationales, or by splitting the requirement if it describes multiple aspects (e.g., WP1_HFRTP_REQ13, see below). With a clearer understanding of the capabilities of the HF-RTP after its first release, new requirements were added (WP1_HFRTP_REQ22 and above). A large part of those target MTTs developed within HoliDes.

The full table of these requirements (in excel format) with a rationale for each requirement as well as other attributes describing each requirement in more detail can be found in the attached document Annex_Req_HF_RTP.xlsx.

(*) updated requirement; (**) new requirement

Requirement ID	**Name**	**Description**	**Require-ment Category**
WP1_HFRTP_REQ01_v2(*)	Synchronized data processing	The HF-RTP shall provide MTTs that are able to process multimodal data sources in a synchronized manner.	MTT
WP1_HFRTP_REQ02_v2(*)	Data logging and tracing	The HF-RTP shall provide means for standardized logging and tracing of data so that logs of MTTs are accessible to other MTTs and events can be sent from one MTT to another during development / design time.	Workflow
WP1_HFRTP_REQ03_v2(*)	Consistent information exchange	The HF-RTP shall present consistent information on all MTTs so that all MTTs work with the same information and are able to interact with each other.	Interface
WP1_HFRTP_REQ04_v2(*)	Storage of experimental data	The HF-RTP shall provide MTTs for an efficient storage of experimental data using standardized formats for constant accessibility.	MTT
WP1_HFRTP_REQ05_v2(*)	Error reporting	The HF-RTP shall be able to produce standardized alerts or reminders when ambiguities or errors are detected by any of the MTTs or with respect to communication issues between the MTTs ensuring a reliable process.	Workflow, MTT
WP1_HFRTP_REQ08_v2(*)	Status reporting	The HF-RTP shall provide information about MTT availability and failure status at any time.	Workflow
WP1_HFRTP_REQ11_v2(*)	Support of different user classes	The HF-RTP shall provide MTTs that allow a distinction of user classes. These classes may, for instance, depend on the expertise level of the user for a given task. Depending on the user's class, the task may offer	MTT

		more or less information and/or control.	
WP1_HFRTP_ REQ12_v2(*)	Appropriate HMI / GUI designs	The HF-RTP shall provide MTTs for design of HMI / GUI that consider physical properties of the target display device. If the target device cannot display the design, MTTs should provide means for adaptation with respect to reordering or reducing of HMI information.	MTT
WP1_HFRTP_ REQ13a_v2(**)	Adaptability to user's physiological properties	The HF-RTP shall provide MTTs that allow determination of the user's physiological properties.	MTT
WP1_HFRTP_ REQ13b_v2(**)	Adaptability to user's current state	The HF-RTP shall provide MTTs that allow determination of the user's psychological and physiological state, e.g. MTTs for monitoring the cognitive load of the operator or operating team.	MTT
WP1_HFRTP_ REQ14_v2(*)	Learning and prediction	The HF-RTP shall provide MTTs that allow machine learning based analysis and prediction of online and offline data.	MTT
WP1_HFRTP_ REQ15_v2(*)	Selective display of information	The HF-RTP shall provide MTTs for prioritizing / weighting information based on the user's state and size of the display device.	MTT
WP1_HFRTP_ REQ18_v2(*)	Workload management	The HF-RTP shall provide MTTs for task management during specific, intense time phases (e.g., for multi-step tasks) depending on the operator's workload.	MTT
WP1_HFRTP_ REQ19_v2(*)	Questionnaires for automatic statistics generation	The HF-RTP shall provide MTTs for creating questionnaires, which can be evaluated automatically and which can generate predefined statistics.	MTT
WP1_HFRTP_ REQ20_v2(*)	Scenario modelling	The HF-RTP shall provide MTTs for modelling or recognizing scenarios (e.g., overtaking on a highway or crossing an intersection) in order to trigger appropriate reactions with respect to the scenario.	MTT, Workflow
WP1_HFRTP_ REQ21_v2(*)	Knowledge storage	The HF-RTP shall provide MTTs for knowledge storage. A design-time knowledge base can store, e.g.,	MTT, Workflow

		checklists for HF-related design issues and thus, can be reused for different projects (learning from experience).	
WP1_HFRTP_ REQ22_v2(**)	Validation of AdCos prototypes	The HF-RTP shall provide MTTs that support and facilitate the validation of the HMI of the AdCos.	MTT
WP1_HFRTP_ REQ23_v2(**)	Empirical validation tests	The HF-RTP shall support workflows that allow the developer to conduct empirical (not model-based) validation tests.	Workflow
WP1_HFRTP_ REQ24_v2(**)	Output report with screenshots	The HF-RTP shall provide MTTs specific for the validation process producing a report as output including the screenshot of the wrong behavior of the AdCos.	MTT
WP1_HFRTP_ REQ25_v2(**)	Validation of human factor models	The HF-RTP shall provide MTTs that support and facilitate the creation and validation of human factor models (i.e., task models, operator models etc.).	MTT
WP1_HFRTP_ REQ26_v2(**)	Automated generation of test cases	The HF-RTP shall provide MTTs that automate the generation of test cases for the validation of AdCoS behaviour.	MTT
WP1_HFRTP_ REQ27_v2(**)	Formal description of input	The HF-RTP shall provide MTTs that formally describe the generic target scenarios, use cases and requirements in order to use it as input for the validation of the AdCos.	MTT
WP1_HFRTP_ REQ28_v2(**)	Tester-independent validation	The HF-RTP shall provide MTTs that conduct the validation of the AdCoS behaviour in a tester-independent way according to the AdCos requirements and the specifications.	MTT
WP1_HFRTP_ REQ29_v2(**)	Adaptive behaviour validation	The HF-RTP must validate adaptive behaviour aspects of the AdCos, e.g.the ability of the HMI of the mobile app to adapt to the different Android phone models it can be installed on.	MTT
WP1_HFRTP_ REQ30_v2(**)	OSLC interface	All RTP tools must implement an OSLC compatible interface.	Interface
WP1_HFRTP_ REQ31_v2(**)	Conceptual HF issues	The RTP must provide taxonomies describing human factors relevant for the specified AdCoS developments (e.g., for human visual distraction, for human cognitive distraction). From	Concepts

		these taxonomies it shall be possible to infer e.g. the relation between workload and distraction.	
WP1_HFRTP_ REQ32_v2(**)	Project identifiers for AdCoS development	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 RTP.	Workflow, MTT
WP1_HFRTP_ REQ33_v2(**)	Provide a unique version identifier for a design version	The HF-RTP shall provide a unique version identifier to distinguish each design and development cycle. The identifier must be available to all tools in the RTP that require it.	Workflow, MTT
WP1_HFRTP_ REQ34_v2(**)	Creation and storage of evaluation plans	The HF-RTP shall provide MTTs to create and store itemised evaluation plans. The evaluation plans must be incrementally modifiable for different design versions and refer to a unique AdCoS project.	MTT
WP1_HFRTP_ REQ35_v2(**)	Storage of textual evaluation reports	The HF-RTP shall provide MTTs for a storage of textual evaluation reports based on itemised evaluation plans. The evaluation reports must be traceable to design versions.	MTT
WP1_HFRTP_ REQ36_v2(**)	Creation of new projects	The HF-RTP shall provide the functionality to create new projects for the development of new AdCoS.	MTT
WP1_HFRTP_ REQ37_v2(**)	Creation and management of users	The HF-RTP shall provide the functionality to create and manage multiple users who use the HF-RTP for the design and evaluation of AdCoS.	MTT

Table 2 List of refined HF-RTP requirements (version 2), (*) Updated requirement (**) New requirement.

Although those requirements are based on the HoliDes AdCoS use cases, some requirements were defined in a broader context. Therefore, not all requirements may be covered at 100% within HoliDes, but they provide directions for the further development of the HF-RTP beyond HoliDes.

2.2 .Project Base Line

The goal of the HoliDes Project is to develop and establish an HF-RTP, which supports and optimizes the AdCoS development. In comparison to a conventional AdCoS development process, the use of an HF-RTP should lead to improvements primarily in terms of significant economic savings and reduction of the investments needed for the development (15% according to DOW, p. 298). The objective of the baseline assessment is to verify, whether the intended enhancement is being achieved in the HoliDes project. There are several working steps required during the project run time to enable the verification:

1. Baseline definition,
2. Assessment of the reference development process (before integration of any HoliDes methods),
3. Assessment of the target state - development process supported by HoliDes,
4. Comparison of the reference and target state according to selected Key Performance Indicators (KPIs).

The baseline definition is done next in section 2.2.1. The assessment addresses the AdCoS' developed in WP6-9 and is described afterwards in 0. For this reason the application partners from each domain have been asked to select at least one AdCoS (related to the use cases) and provide the assessments regarding to them.

2.2.1 Baseline definition

To accomplish the complex task of defining a baseline, the consortium installed a baseline working group consisting of the following partners: HFC, PHI, ANYWI and AIRBUS DE. The objectives of the baseline working group are to 1) define the baseline assessment process, 2) review and discuss the assessment results, 3) improve the assessment process where needed and 4) report results in the deliverables. The baseline definition process is part of WP1 and based on the approach given in the DOW. Two workshops (Naples 08-05-2014 and Eindhoven 14-08-2014) were conducted with the aim to create a framework that enables the measurement of the added value of the HoliDes project to development and qualification processes. The framework should allow collecting information about what has to be done in an AdCoS development process to be compliant with HF and safety requirements and which effort is involved. The framework should be applied for both, reference development process (without HoliDes) and development process supported by HoliDes. The subsequent comparison of the assessments shows, which parts of the process are affected by HF-RTP and the resulting effort savings.

We have drafted a template for the assessments of the AdCoS development process including the related HF activities. In the template the overall process has been split into two phases: The core development process and the after sale phase (the period of time after the market introduction, where the true product quality is monitored, potential system weaknesses are exposed and the product might be improved). The latter phase also includes the provision of user trainings which might be necessary and the handling of complaints. Furthermore five sub-processes of the HF development process have been identified:

- Analysis
- Concept
- Design
- Implementation
- Evaluation

The given process structure covers all possible activities, which can be undertaken during development of an AdCoS. It is formulated in a general manner, to allow its application by all partners of WP 6-9, even if their processes widely differ from each other. The task of the application partners is to not only describe the actual development work but also the resources and efforts which are necessary after the development has finished and the product is on the market. The partners were asked to split the development processes into subsequent steps (e.g. according to the V-Model) for more detail or accuracy of the assessment. In some AdCoS development processes within HoliDes, only subsets of the proposed phases which are defined in the baseline are conducted. In such cases the baseline assessment is adapted accordingly.
The baseline definition includes key performance indicators that enable the assessment of effort that has been invested. The **Key Performance Indicators (KPIs)** used in the assessment are the following:

- TIME: in person-month (PM) refers to the effort to perform the given activity (assuming 160 hours per month)
- RESOURCES: number of persons required (size of the team)
- COST: cost of the Human Factors process of AdCoS creation. It includes all kinds of required cost. The value corresponds to thousands of Euros (k€). Costs are divided into:
  - COST RC/RECURRING COST: costs that occur in a repeating fashion. Mostly the activities performed and expenses incurred in

the production life-cycle phase are classified as recurring (e.g. production labor; direct material; production process costs). The recurring costs are rarer (less frequent) in the development process, but they can occur. Examples include: efforts required for redesign, modifications, rework and replacement; tool maintenance; training.

- COST NRC/NON RECURRING COST: costs expected to be incurred only once in the life-cycle of a system and generally associated with pre-production efforts. Examples of nonrecurring costs are diverse development costs: system test, basic design and development through the first release of engineering drawings and data, basic tool and production planning through initial release, engineering models built for development or test purposes only, needed equipment and MTTs and specialized workforce training.

The classification of NRC and RC is sometimes difficult and there are different interpretations possible. The most important thing is, to stay consequent and use the same classifications in all baseline assessments during the project. The following table has been used as template and was used for the assessment.

Domain/WP		**AdCoS**	**Use Case**
Description			
Assessment code	**Possible:** **0** - Measurement of Initial State (=without HoliDes) **1a** - Estimation of expected Target State (=with HoliDes) **1b** - 1a update of expected target state **1c** - 1b update of expected target state		
Datum			
Development Process			
Sub-Process 1 (Analysis)			
Description			
Step	1.1	1.2	1.n
Detailed Description			
Effort - Generic KPIs			
time (MM)			
resources			

cost RC (Euro)			
cost NRC (Euro)			
Sub-Process 2 (Concept)			
...use the template above			
Sub-Process 3 (Design)			
...use the template above			
Sub-Process 4 (Implementation)			
...use the template above			
Sub-Process 5 (Evaluation)			
...use the template above			
After Sales Phase			
...use the template above			
Summary effort for whole development process and after sales phase - Generic KPIs			
time (MM)			
resources			
cost RC (Euro)			
cost NRC (Euro)			

Table 3 Template for the assessment of the project progress.

2.2.2 Assessment

In HoliDes, we will measure the improvements introduced by the new development processes for each of the workflows individually with four subsequent assessment steps. :

1. **Assessment 0** – is an estimation of effort and costs needed for the described process before the methods, techniques and tools to be developed in HoliDes are being introduced. In the assessment procedure the partners start with describing the AdCoS development processes and workflows according to the given table. They specify the process steps of their workflows and calculate the KPIs for each of them. This first step serves as a **baseline reference assessment**.

Further assessments will then estimate how the introduction of HoliDes MTTs and HF-RTP will affect the resources and effort needed and will be compared to this baseline.

2. **Assessment 1a (estimation of target state after introduction of HoliDes MTTs and HF-RTP)** – using with the same assessment table the development process supported by HoliDes MTTs and HF-RTP will be described. It is the starting point for an estimation of the expected added value of HoliDes in the future. The elaboration of this second step occurs in the middle of the second cycle at milestone M3. At this stage of the project the HF-RTP and MTTs are still under development and it might be difficult to describe the future process of AdCoS development in detail, but it is at least possible to estimate the potential of the changes introduced by the HoliDes methodology.
3. **Assessment 1b (target state with HoliDes MTTs and HF-RTP)** – will be an update of Assessment 1a and should be delivered at the end of the second cycle. The progress achieved in HoliDes should enable more accurate evaluation of the improvements.
4. **Assessment 1c (realized target state with HoliDes MTTs and HF-RTP)** – a final update of the assessment will be conducted until milestone M5. This assessment should provide a clear overview of the changes in the AdCoS creation process and the resulting economic benefits based on the final outcome of the tailoring and use of the RTP in the AdCoS development. While the Assessment 1a is an estimate of the added value of the HoliDes RTP in the development process, the Assessment 1c should be based on realized improvements.

The assessments are conducted per application domain in the respective work packages. Each of the WP6-9 has selected at least one of their AdCoS (in relation to one or more use cases) developed within HoliDes for the assessments. Table 4 shows an overview of the assessed AdCoS with related use cases.

WP	Provider	AdCoS	Use Case

WP 6	PHI	MRI Trolley	Safe patient transfer
	PHI	Robust ECG triggering system	Robust ECG triggering
	PHI	Guided patient positioning system	Guided patient positioning
WP 7	HON	Diversion assistant	Diversion assistant
WP 8	Airbus DS	Border Security	UC 1 Presence / UC 4 Behaviour Patterns (combined in one baseline, UC 1 is a prerequisite for UC 4)
	Airbus DS	Border Security	UC5 Load Balancing at Operator Level
	REL	Energy Network Control AdCos	The Communication between the operator and the operational teams in the field
WP 9	REL	LCA	WP9_AUT_U2.1_CRF_v01 WP9_AUT_U2.2_CRF_v01 WP9_AUT_U2.3_CRF_v01

Table 4 AdCoS Use-Cases per WP.

The assessments 0 and 1a are being delivered in confidential document only. The assessment tables include detailed information about the development process of each AdCoS without and with HoliDes and effort (expressed in terms of KPIs) needed for realisation of each stage of work. In all cases the benefits have been identified in terms of the applied time, resources and costs. Dependent of the AdCoS and activities conducted within the single processes, there is different support needed and expected from HoliDes. It should be noted, that the assessment of the target state with HoliDes is initial and will be updated twice in the future, when the development of the HF-RTP has progressed. Currently, expected changes can only be estimated. All assessments which have been done so far can be found in. The final paragraph of this section will discuss the benefits resulting from the comparison between the first two assessment steps, which have already been conducted:

WP6

The developers of the MRI Trolley expect the concept and design, as well as the evaluation and after sale process to become optimized in the future. The savings according to the initial estimation total up to about 31% of time, 18% of resources, 25% of recurring costs and 33% of non-recurring costs.

The development process of the Robust EKG triggering system, as well as of the guided patient positioning system will be affected by HoliDes in the same way. The estimated changes are related to all of the specified sub-processes: analysis, concept, design, implementation, evaluation and after sale phase. The effort will be reduced by 19% by time, 10% by resources, 20% of the non-recurring costs and 50% of the recurring costs.

It needs to be noted that the recurring costs have been identified by PHI only in the after sale phase and result from effort spent in user training and customer complaint handling.

WP7

HON specifically targets the evaluation phase of the diversion assistant within HoliDes. For this reason the assessments are not based on the proposed sub-processes of AdCoS creation. Instead of it, there have been sub-processes specified within the evaluation process: software testing and profiling, scenario design and modelling, data synchronisation, data organisation, data playback and module re-use. HON provided very detailed information about content of work with and without HoliDes support. The estimated effort reduction of the assessed process amounts to nearly 58% of time, 68% of resources (interpreted as material costs), 62% of recurring and 16% of nonrecurring costs. According to HON the assessed evaluation phase has been recognised as the most expensive within the whole development process. Without HoliDes the realisation of the evaluation required about 50% of the whole budget. Due to improvements brought by HoliDes in this phase the total cost of the whole development process will be reduced by about 24%.

WP8

The AIRBUS DS provided two separate assessments referring to the development process of the border security. The first one addresses the development process related to two closely related use cases (UC1/ Presense and UC2/Behaviour patterns). The second development process which was assessed is the UC5/Load Balancing at Operator Level. In both use cases there are changes expected within analysis, concept, design and implementation sub-processes. The evaluation and after sale phase are estimated not to be affected by HoliDes in the future. The estimated savings in development regarded to UC1/UC5 amount to about 3.5% of time and 3.5% of non-recurring costs. The effort within the process related to UC5 will be reduced by about 6.2% of time and non-recurring costs.

In the development process of Energy Network Control concerning the HMI of the mobile application for the communication between the operator and the teams which work in the field has been assessed by REL. The steps that will be modified by HoliDes have been identified within sub-processes: analysis (steps: task modelling and task analysis) and implementation (step: validation). The remaining development activities will not be modified by HoliDes. The expected savings sum up to 6.4% of time, 4.8% of resources and 7.6% of recurring costs.

WP9

The assessed development process of Lane Change Assistant is focussed on the HMI of the mobile application. The specification of the process is equal to the case of the development of the HMI of Energy Network Control assessed in WP8. It probably results from the similar goal of the processes, even when different AdCoS are developed. The steps affected by HoliDes are task modelling and task analysis (analysis sub-process) and validation (implementation sub-process). The expected cost reduction amounts 6.4% of time, 4.8% of resources and 7,6 % of recurring costs.

2.2.3 Further steps

Further assessments have been scheduled within the project. Furthermore, the collected assessments results will be reviewed and discussed in the baseline working group. The applied key performance indicators seem to be interpreted differently by the assessment providers, e.g. time is in most of cases directly translated into RC or NRC costs, but in the assessments from the health domain this KPI is not related to any kind of cost. The inconsistencies need to be cleared and if necessary the assessment data will be adapted. It will also be discussed, whether the list of performance indicators should stay constant, or further indicators should be added.

2.3 Human Factors Integration Concepts

This section is divided into three parts: The first subsection 2.3.1 identifies elements relevant for the HF development process, which should be addressed in the HF-RTP. The following section 2.3.3 introduces some of the most prominent methods, which are applied in industrial context to deal with the HF issues explained in the first section. Finally, subsection 2.3.4 gives initial ideas how the methods can be integrated into the HF-RTP.

2.3.1 Selection of Human-Factors Issues

The review of the HF integration concepts and regulations (D1.2) and the requirements collected in WP 6-9 have been matched and brought about the conclusion that the main common link, along which the different human factors activities during all stages of the development process are oriented, are the HF issues. These aspects are related to human-machine-interaction, which need to be handled to ensure the intended quality of the end product. In this section we identify the HF issues most relevant for HoliDes, which is the basic input to T 1.6 that will define a meta-model for HF issues. In the following we will present the most promising HF integration concept for each of the four HoliDes application domains. For detailed explanations of the concepts mentioned below please refer to deliverable D1.2.

Aeronautics (WP7)

Three HF integration concepts have been reviewed for the aeronautics domain:

- The HF Case: Guidance for HF Integration provided by Eurocontrol
- Integrated Human Centred Systems Approach (IHCSA) to the Development of Advanced Cockpit and Air Traffic Management Systems.
- Honeywell internal HF integration process.

All the concepts point out that the emphasis of HF is to ensure a *safe* and *efficient* overall system performance, which is simultaneously driven by technical and human capabilities coupled with operational requirements. The main focus of HF Integration is set on addressing human performance related issues throughout the system development process.

The IHCSA Concept concentrates on the human operator as a key element of the closed loop information system. It also stresses the potential overall system performance degradation resulting from the limited human capabilities. Particular attention is paid to human-automation issues emerging in the Air Traffic Management (ATM) Domain: system performance (comparing performance of automated and non-automated alternatives in full range of potential operating conditions), situational awareness, attention limitations, information and task overload, understanding of automation criteria, changes in authority, changes in communication modes and human reliance on automation. The other Human-Factors integration concepts do not define which HF-issues are relevant and should be addressed in the system development process. It is mentioned that every development project is unique. It follows that project specific HF-issues have to be identified every time. In an optimal way this task should be accomplished during the early stage of the development. The HF Case approach establishes well-structured actions to specify the project crucial HF issues systematically.

Part of the regulations reviewed in task 1.3 are dedicated to the electronic flight bag and address issues related to HMI aspects (use of colours, legibility of text and symbols, lighting conditions, adjustment to changed human vision by older users), hardware and software design (e.g. position, power connection, data connectivity), and operational use conditions. It is also mentioned, that the used procedures should be designed to mitigate and/or control workload. Primarily, the regulations provide requirements, which should be fulfilled by the system. Here again a number of HF aspects are addressed: work place layout, physical limitations, human machine interaction (e.g. presentation of information, user specified settings, operations), and system reliability. The analysis of the requirements collected in WP7 related to AdCoS dealt with system features, which are closely related with workload and usability.

Control Rooms (WP8)

The HF Integration concepts addressing control rooms are: the HSI Human Systems Integration Concept and the NATO Human View (HV) concept handle the topic of HF issues in a general manner. HSI is a comprehensive framework for the design and evaluation of the interplay of hardware, software and humans, covering nine functional areas of HF issues including: workplace ergonomics, human-computer interaction, health and safety aspects, and organisational and social aspects. The NATO Human View Concept (HV) covers human factors problems from six different perspectives (the human views): general concepts (understanding the human dimension in relation to operational demands), human characteristics, tasks, roles, human network and training. ISO 11064 Control Centre Design Standard consisting of several parts focuses on aspects of the design of control centres: workplace layout, workstation layout, alarms, display and controls. These are mostly related to ergonomics and usability. Additional general environmental requirements like temperature, lighting, and noise are addressed. The MIL-STD-1472G for military systems in control centres provides mainly requirements focusing on classical ergonomical issues. The number of requirements collected regarding the demonstrators developed in WP8 dealt with characteristics of a system and operations that should be fulfilled to support usability, operator situation awareness and avoid operator over- and under-load. Furthermore the system should be able to analyse and adapt to the psycho-physiological state of the operator.

Health (WP6)

The concepts for human factors integration and regulations in the domain of healthcare are primarily concerned with the HMI and safety. Mostly, the concepts cover issues, such as interface usability, information requirements and representation, and task demands in order to develop a safe and sound system. The system is seen as usable, when it can be used effectively, efficiently and with satisfaction. It is underlined that the end user should be included throughout the whole system development process. Usability is also perceived as the key HF issue in WP6. The developers pointed out, that the demonstrators need to be easy and intuitive in use, reliable and provide fast feedback. They additionally mentioned that the MTT's for testing usability need to be provided via the HF-RTP. Furthermore, the interface should be applicable by different devices (e.g. laptop, tablet, smartwatch) and using contexts (e.g. hand-free modus).

Automotive (WP9)

The concepts dedicated to the automotive domain are: The Code of Practice (CoP) and the Detectability concept focus on Human Machine Interaction (HMI): The Detectability concept mainly focusses on visual output devices; The CoP mainly on safety/controllability related issues. Additionally the CoP also provides some guidance for the System reliability issue.

The HF-Standards have put their focus on several aspects of HMI, mainly on ergonomical and usability issues. Therefore, some ISO standards provide information on a specific topic: dialogue management, auditory signals, transport information, and control systems. Other HF publications concern the automation and allocation of function between human and machine. In general, the focus is put on human limitations like working memory and attention on the traffic situation. The interaction with the AdCoS should provide all necessary information to ensure the situation awareness of a driver and at the same time does not lead to distractions. Situation awareness, workload and avoiding of distraction signals are also the main issues addressed in the requirements collected in WP9. Furthermore the use of the system should be easy-to-learn and the interaction should result in comfortable driving experience and safety. It is also important, that the driver trusts the system's reliability.

2.3.2 Summary

To sum up, this overview took into account

- General strategies of Human-Factors integration described in the review in deliverable 1.2.
- Human-Factors and safety related standards and regulations, and
- The application use-cases and their defined HF requirements.

The overlap of the four WPs point to the HF issues considered as the most important within HoliDes and which should be addressed in a development process of the AdCoS in different domains and need to be supported in the HF-RTP. These are:

- usability,
- situation awareness,
- workload and
- user distraction.

2.3.3 Methods

For the relevant Human-Factors issues identified in deliverable 1.2 and in section 2.3.1 several methods are established common practice in the Human-Factors domain. Most, but not of all of these methods are concerned with evaluation. Table 5 represents a collection of methods related to the identified issues.

Human Factors Issue	Common Practice Methods	Related Activity
Workload	Task performance measures o Primary task measures o Secondary task measures	Evaluation

	Physiological measures ○ Pupil diameter variation ○ Heart rate variability ○ Skin conductance response ○ EEG and evoked potentials (P300, N100)	Evaluation
	Subjective workload assessment ○ NASA Task Load Index (NASA TLX) ○ Bedford Scale ○ Modified Cooper Harper Scale ○ SWAT – Subjective Workload assessment technique ○ Workload Profile Technique	Evaluation
	Cognitive and/or Modelling Techniques ○ Cognitive task load Analysis ○ GOMS Analysis and Prediction	Analysis Evaluation
Situational Awareness	SA Requirements Analysis	Analysis
	Freeze Probing SAGAT – Situational Awareness Global Assessment technique.	Evaluation
	Online Probing SALSA	Evaluation
	Situation Awareness Rating Technique (SART)	Evaluation
Usability	Questionnaire based Techniques ○ ISONorm ○ IsoMetrics	Evaluation
	Usability Testing	Evaluation
	Heuristic Evaluation	Analysis Evaluation
	Walktroughs ○ Cognitive Walkthrough ○ Heuristic Walkthrough	Evaluation
	Inquiry Methods ○ Contextual Inquiry ○ Focus Groups ○ Card Sorting ○ Workshops ○ Subject Matter Expert Interviews	Analysis

Distraction	Detection of driver distraction based on in-car measures: Combines different measures like o performance-related measures o eye-gaze recordings o workload-related measures o physiological data	Evaluation

Table 5: Common practice methods related to the relevant Human-Factors Issues.

Most of the listed measures are related to evaluation or to validation activities within the development process. Only very few relate to the analysis and/or design activities.

2.3.4 Integration into the HF-RTP

One result of the compilation of HF integration approaches in D1.2 was the observation, that most general approaches suggest frameworks for HF related workflows but do not describe methods or suggest tools to be used. The standards and regulations reviewed in the same document often give very precise indications as to which HF issues should be observed and sometimes which workflows should be adopted. But the information about methods and tools to be used lacks precision or is missing.

To be able to integrate HF activities into the HF-RTP we need to overcome a number of difficulties related to the low level of formalisation or even completely informal nature of most of the methods mentioned in the previous section. For the purpose of certification to human factors criteria, the results of human factors activities need to be traceable at least in a textual way. At least a verbal description of what has been done and achieved should be available.

To overcome these difficulties we propose to develop a semantic framework for the different human-factors activities using a defined set of descriptors. The descriptors should

- be generalizable for different kinds of activities,
- be domain independent,
- provide a common framework to exchange information between activities, as well as methods and tools and at the same time make them accessible for the RTP

- make the information traceable for certification purposes.

The definition of a set of descriptors should eventually lead to the definition of a Human Factors ontology that can be used as a generalized HF semantic framework within the HF-RTP and furthermore lead to the definition of a HF domain within the OSLC standard.

Keeping in mind a proper documentation and traceability of Human-Factors activities, which is important for certification purposes, the first approach to a set of descriptors is focused on the outcomes or results of Human-Factors activities. The descriptors should give information about the following basic questions:

- Which category of HF activity has been performed?
  As a first approach, we distinguish three main HF-related activities which are analysis, design, and evaluation. As described in D1.1 and D1.2 these activities might be repeated several times within a development workflow.
- What is the outcome or result of the activity?
  Different activities have different types of outcomes. In many cases the result of analysis type activities are related to selecting which HF-issues are relevant for the given project. The design-type activities describe how these issues are being tackled by design decisions and the evaluation activities assess whether the result is acceptable.
- Which Human-Factors issues have been considered?
  The issues are a connecting link between different types of activities throughout the development workflow.
- How has the outcome been achieved? Which method has been used and how?
  The used methods (and eventually techniques and tools) should be included in the set of descriptors. Depending on used methods, more descriptors will be needed to describe details of the exact procedure (experimental settings, sample size etc.) as well as the gathered empirical data and to provide access to the data.

Figure 3 give an overview of these ideas for the definition of basic descriptors. Table 6 summarizes brief descriptions of a basic set of descriptors.

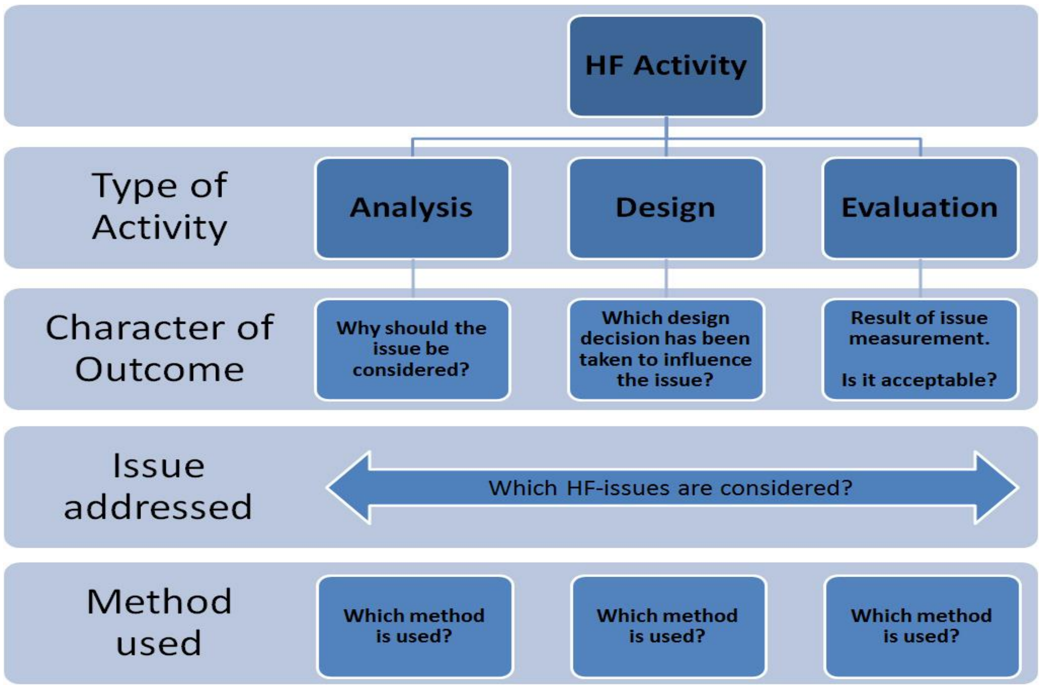

Figure 3: Basic descriptors for Human-Factors activities.

Descriptor	**Content**
ID	Identification of Activity
Type of Activity	<Analysis/Design/Evaluation>
Outcome/Result	Description of what the result of the activity is and in which way it is being used in the project
Issue considered	One or more HF-Issue taken from a list of possible Issues
Method used	Name of the method (details of the procedure should be described in subordinate descriptors)
Data	Format and content of available empirical data if applicable.
Relation to other HF activities	Which relation is there to precursors or consequent HF activities?
Standard or regulation touched	To which standard or regulation is this activity related?

Table 6: Basic set of descriptors for Human-Factors activities.

A sketch for a tool that could support a traceable documentation of evaluation activities has been included in D5.3 (HF-Filer). The proposed set of descriptors should lead to the development of an HF ontology suited to the needs of the HF integration into the RTP.

2.4 Definition of the AdCoS Methodology and HF-RTP Architecture

2.4.1 Adaptation

D3.3 makes the point that humans behaviour is (to a large extent) goal driven and in order to fulfil these goals a number of tasks has to be accomplished. If weaknesses occur in a human's ability to achieve his goals, assistance systems may help and tasks can to be allocated to a machine agent. To which these tasks are automated is well known as levels of automation. The levels of automation which are employed have significant bearing on the system that is being developed. Table 7: lists 10 levels of automation ranging from 1 where the human has 100% control with no assistance from machine to level 10 where the machine decides everything, has full control and ignores the human.

1	The computer offers no assistance: human does it all
2	The computer offers a complete set of decision/action alternatives
3	The computer narrows the selection down to a few
4	The computer suggests one alternative
5	The computer executes that suggestion if the human approves
6	The computer allows the human a restricted veto time before automatic execution
7	The computer executes automatically, then informs the human
8	The computer informs the human only if asked
9	The computer informs the human only if it, the computer, decides to
10	The computer decides everything, acts autonomously, ignores the human

Table 7: Different levels of automation (see D3.3 also).

When defining tasks, which are automated to some degree in HoliDes, the relevant WP should use these levels and definitions. However, this table alone is too simplistic. WP3 proposes 5 stages which are involved when goal oriented behaviour has to be described, namely *Perception, Evaluation,* *Decision Making, Action Planning* and *Action Implementation*. Each of them can be done by either the human, a machine or by cooperation between them. Hence, different levels of human computer automation can be used for each of these 5 stages. These concepts will feed into the adaptation meta-model which will feature later on in this document. D3.3 gives examples of a model, which shows all 5 of these stages.

Figure 4 illustrates one of the use cases in the border surveillance domain (WP8) which deals with the problem of absent control room operators.

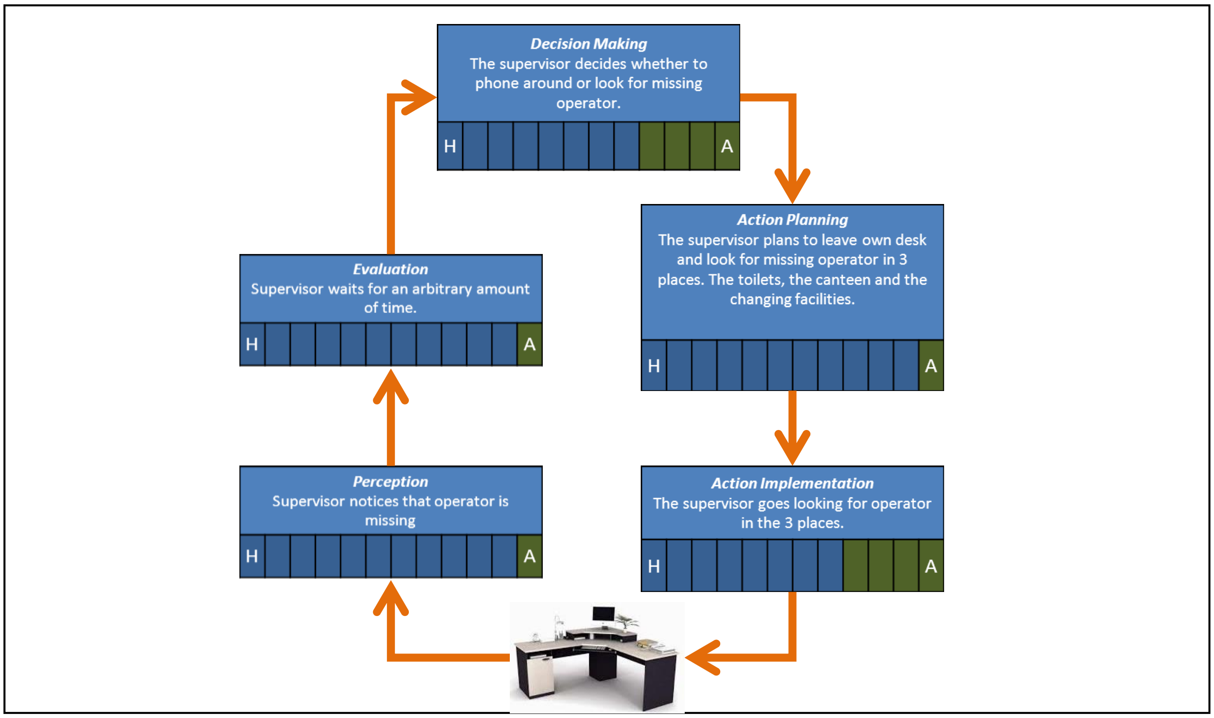

Figure 4: Current Task Allocation for Absent Control Room Operator.

In the current setup, the absence is not detected automatically until the supervisor notices that someone has been gone. This is reflected in Figure 4 with the lower levels of automation around the task allocation process.

The plan is to raise the level of automation when dealing with absent operatives, which is the aim by the end of HoliDes. The benefit of modelling the levels of automation in this way is that it's instantly clear what is going on in the feedback loop and the level to which the computer takes over tasks from the human operator.

Proposed Approach

As a methodological approach it is proposed that for all AdCoS use cases in HoliDes similar diagrams should be constructed. Along with a pictorial view, each use case will have a textual description explaining what is going on throughout the 5 stages. This information will feed back into WP 3 to make informed decisions, as to what decision making methodologies are available.

2.4.2 Complexity

Complexity has different meanings depending on which WP you speak to. If you speak to the modelling experts in the application work packages, they will define complexity by the sophistication of the models, i.e., the more components with more details the higher the complexity.

In WP 3, complexity can be found within the algorithm that defines the decision making in adaptation. A more complicated algorithm might have to rely on additional sensors and inputs. For example, in the border control room there could be multiple sensors indicating a number of states such as fatigue, stress, boredom and so on. In this case, more sensors mean more complexity.

Any number of inputs could be taken into account, e.g., the operator's experience or training, recent behaviour, current workload and the time of day. All of these factors need to be taken into account before arriving to a correct action plan so this scenario could be considered to be more complex.

In WP 1 complexity is used to refer to the problems that arise from a human and machine task sharing. There are 3 areas of complexity in developing an AdCoS: 1) Computational Complexity, 2) Cognitive Complexity and 3) Regulatory Complexity

Task allocation is at the core of any human machine interaction. A task represents something, which requires an action to complete it. How that action allocation is split between a computer and a human agent there needs to be explicit delegation. It is not just the task, which needs to be considered in delegation but also the authority. If the machine intervenes, is the human still accountable? The co-operation of multiple agents needs to be clearly defined and understood [2].

There are three ways in which this complexity could be modelled and managed.

- *Deterministic:* a philosophical viewpoint which states that for every event there must be a number of preconditions that could cause no other event. Examples for managing complexity in a deterministic manner could involve the use of lookup tables and fully deterministic rule based systems.

- *Non deterministic with some predictability and certification possibilities: a* probabilistic model is one in which there could be many outcomes. Examples of this could be stochastic rule based systems in which the current state of the system is not known and only given through probability.
- *Non deterministic with not enough predictability and impossible to certify:* examples of these are neural networks, genetic algorithms and genetic regulatory networks, as the clear input output relation cannot easily be described. The more deterministic a system is, the easier it becomes to certify it from a safety perspective.

2.5 HF-RTP Processes and Work Flow

The content of this section is not available in the public part of the deliverable.

2.6 HF-RTP Common Meta-Model

According to the proposal "The Human Factor Common Meta-Model (HF-CMM) will be defined and implemented as an extension of the CESAR CMM with new concepts for AdCoS processes, techniques and tools." So far in the previous deliverables of WP1 the meta-model task had minor contributions only. To a large extend this was the case, because the development workflows had to be defined first before common concepts can be identified for a meta-model.
As a starting point section 2.6.1 explains why meta-models are useful and how they can help to set a common basis for data exchange and tool interoperability. Afterwards the meta-model concepts that are under current discussion in HoliDes are presented in section 2.6.2. At the current stage none of these models is already finished and some are only identified as candidates and have not been discussed in detail yet. Thus, the overall meta-model will develop over the next few deliverables and the current state can only be considered as intermediate.

2.6.1 Cesar Meta-Model and Common Meta-Model

In the CESAR project two different terms were used, a CESAR Meta-Model and a Common Meta-Model. The *CESAR Meta-Model* is a reference ontology about common terminology used by four engineering disciplines in different safety critical domains: systems engineering, safety engineering, product line and performance engineering. The common terminology contains concepts for e.g. contract-based requirements engineering, component-based design as well as validation and verification. The ontology remains a descriptive model whereas the *Common Meta-Model* is a concrete instantiation / implementation of the ontology into the RTP. The instantiation enables tool interoperability based on the concepts defined in the descriptive ontology.

The Meta-Model in CESAR is based on concepts taken from the Systems Modelling Language (SysML[2]) and Heterogeneous Rich Component Model (HRC[3]). Thus, it contains basic elements to describe a system, like *Components*, *Variables, Ports*, *Connectors* or *Services*. Figure 5 was taken from Baumgart (2010) [1] and shows basic elements of a component-based system design according to the HRC language. Typically for this approach, the system under development is partitioned into smaller subsystems (components). Interfaces of the components are specified and the internal behaviour of the components is defined (e.g. via state charts). Component-based design is closely related to contract-based design. A *contract* for a component specifies that "if assumption X holds for the input data of the component, the component guarantees a specific output Y". In this way a contract is a specification about guaranteed behaviour if the data is fed into the component according to the input interface specification. Such contracts may be specified for a lot of different properties of a component, e.g. time constraints, value ranges, processing time. These contracts are often directly linked to requirements.

[2] http://sysml.org/sysml-specifications/
[3] http://www.speeds.eu.com/index.php

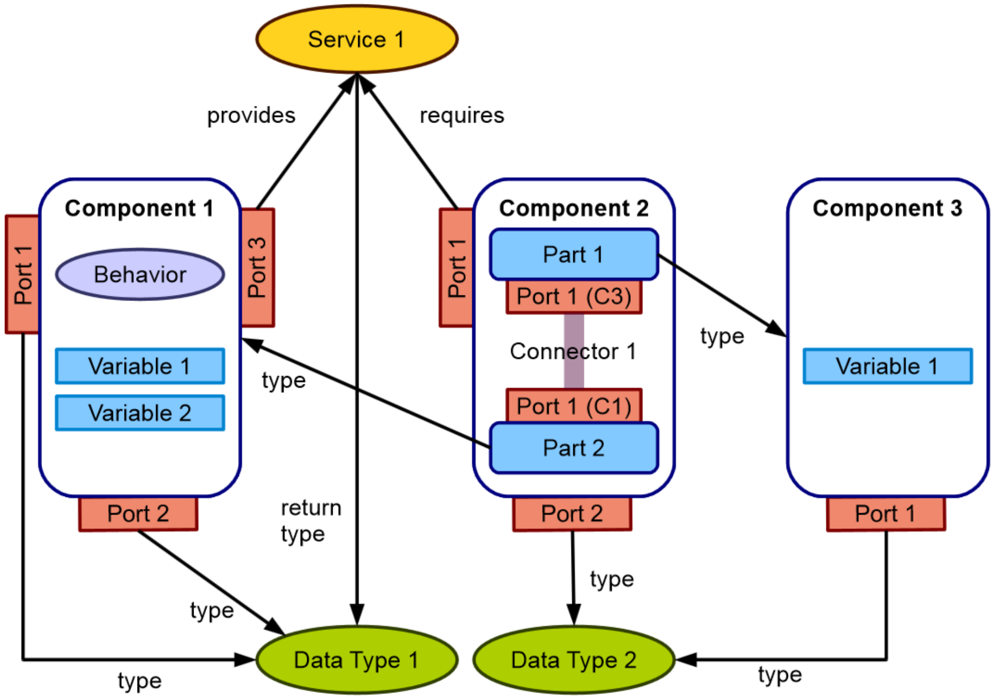

Fig. 2. The common meta-model and component based design.

Figure 5: taken from [1]

Referring to Figure 5, each element is specified in the CESAR meta-model, e.g. what is a *Component*, what is a *Port* or how can a *Behaviour* of a component be specified[4]. A concrete instantiation of this model can be used to create a diagram editor which allows the specification of a component-based system model according to the CESAR meta-model.

Why is such a meta-model and component based approach useful for e.g. the traceability during the system development? Assume there is another meta-model for requirements, which specifies the basic attributes of a requirement and has a concrete instantiation in a requirements engineering tool. Both, the component-based modelling and the requirements tool store

[4] The Cesar Meta-Model also specifies the elements for contract-based design, here only the component model can be seen.

their internal *models* (component-based system design model, requirements model) based on the predefined format of the respective meta-models.

The next step would be to establish links between the requirements model and the component-model. A requirement might be linked to any item of the component diagram, e.g. a component itself, a behaviour implementation or even a port or data type. A typical link between a requirement and a system model element is "component X *satisfies* requirement Y". Such links improve the traceability during the design process. Very similar, this process could be extended by e.g. a meta-model for test case generation or empirical validation issues.

As it was already reported in previous deliverables HoliDes will rely on the OSLC specifications, which itself contains a number of meta-models for different domains, e.g. Requirements Engineering, Architecture Management, Change Management or Quality Management. If all tools that are required in a certain development process implement the full standard, linking between elements from all these domains could be done easily in all tools. Unfortunately, the reality is a bit more complex and two issues will arise in almost every development process:

1. A problem with the OSLC standard might be that for example the requirements specification contains only one mandatory attribute: the name of a requirement. But there are about 40 additional attributes which are all optional. If tool vendor A considers only the name relevant for his tool while a requirements tool B may have implemented additional elements, tool A will not be able to deal with those attributes. Thus it cannot link those more specific requirement elements which might lead to traceability issues. It remains an open question if a bigger set of mandatory elements would be more useful to describe elements like a requirement in a more standardized and complete way.
2. Another issue is that in a lot of industrial development processes proprietary tools are integrated, which are 1) not OSLC compliant at all and 2) no adaptor can be integrated directly because no API exists.

In both cases tools will not be able to exchange data, even if they rely on defined meta-models. As a backup solution an external linker tool can help. Figure 6 describes the situation where neither tool (requirements tool and system modelling tool) has implemented an appropriate set of attributes to

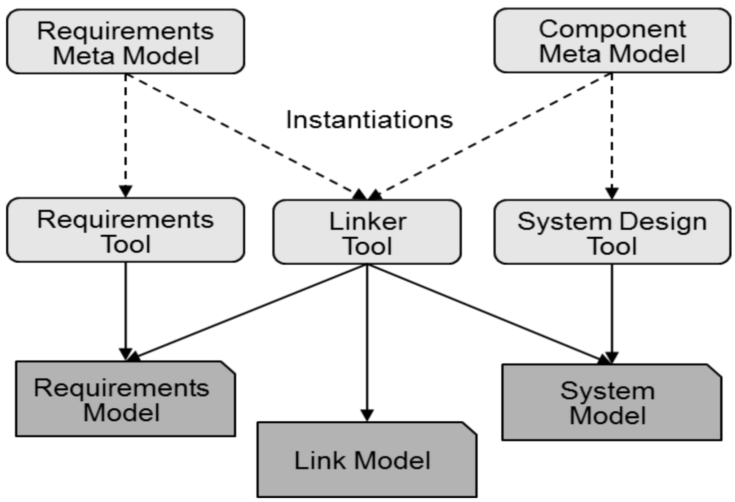

Figure 6: Two Meta-Models are instantiated in three different tools. None of the tools has implemented an interface to get access to the other tools model elements directly. At this point, a specifically implemented linker tool is able to read both concrete model formats and is used to connect system model components with requirements. The links are stored in a specific linker model.

be able to understand the other tools' model format. In that case, a specific linker tool can be implemented which is able to read both meta-models and can link elements of both.

There are a lot more things that might be discussed here but for further information please refer to the OSLC webpage, the project webpages of Crystal or MBAT or existing literature about meta-models. The next subsection will introduce the meta-model concepts that are going to be developed in HoliDes.

2.6.2 Identify Human Factors Concepts in HoliDes

The goal of HoliDes is to come up with a Human-Factors (HF) Meta-Model, which can be instantiated by specific HF tools. It should serve as an initial approach to integrate Human-Factors methods on a more formal basis into the classical systems engineering disciplines, processes and tools.

To create a HF Meta-Model, alternative approaches could be followed: one could start to aggregate a huge number of Human Factors concepts based on literature research. Afterwards a categorization and formalization could lead to a Meta-Model. This approach did not seem to be feasible for HoliDes due to the number of existing concepts. Instead the defined RTP workflows (D1.3, D2.2 - D5.2) and the tools developed by HoliDes partners in WP2-5 are the starting point. Additionally, the work done in Task T1.2 Human Factors Integration was used as input here. This reduces the overall number of candidate concepts and focusses on those which are relevant to the project partners. Nevertheless, the resulting HF Meta-Model could be extended later on by other projects.

The first action of Task 1.6 was to go through the previous deliverables D2.2 - D2.5 and categorize the available tools developed by the project partners. The deliverables contain example design workflows that are used to solve specific Human-Factors problems. Table 8 shows an initial set of three concepts which are tackled by several workflows and tools. Those concepts are potential candidates for a Human-Factor Meta-Model. The *Producer* column means, that e.g. workload or distraction is measured by a certain method / tool which produces an output "*workload measurement*". The *Consumer* column contains methods and tools that are interested in analysing the data of the Producers.

Section 2.3.1 of this document mentions four HF concepts: *Workload, Distraction, Situation Awareness and Usability*. The first two could be found as a part of the development processes while the latter two concepts could not be assigned to a tool listed within the workflows. In general the creation of meta-models is ongoing work that needs to be done by this task in cooperation with the tool suppliers and application partners.

HF-Concept	Producer	Consumer
Task Model	• PED, HEE (OFF)	• HEE (OFF) • TrainingManager (OFF) • TA for adaptation of automation (DLR) • Theatre Technique (DLR) • Communication Processes (SNV) • CASCaS (OFF) • COSMODRIVE (IFS) • Hand Gesture Detection (BUT) • Head Orientation Detection (BUT) • Means Ends Modelling (AWI)

Distraction	Observed Human Behaviour
	• Test vehicles (CRF, IBEO)
	• Eye-tracker (ERG)
	Cognitive Models
	• CASCaS (OFF)
	• COSMODRIVE (IFS)
		• Gesture Recognition & Head Orientation (BUT)
		• Driver Distraction Detection
		  ○ Behaviour Data (TWT)
		  ○ Vehicle Data (UTO)
		• BAD-Mob (OFF)
		• MIVIDA (IFS)

Workload	Observed Human Behaviour • Test vehicles (CRF, IBEO) • Eye-tracker (ERG) Cognitive Models • HEE (OFF)	• ICA (ERG)

Table 8: A first number of HF concepts which were identified in a number of workflows and tools in the deliverables D2.2-D5.2

Table 9 lists two specific topics / techniques which are very important for HF methods and the analysis of HF concepts. Empirical investigations are an important method for 1) information gathering and 2) validation purposes.

	Producer	**Consumer**
Real-time data streams	Test Vehicles (CRF, IBEO) Driving Simulators: • COSMO-CIVIC (IFS) • SILAB (OFF, TAK) • DOMINION (DLR) Devices: • Eye tracker (ERG) • Pro-Sivic (CVT)	• System Monitor (ATOS) • RT-Maps (INT) • Gesture Recognition & Head Orientation (BUT) • COSMODRIVE (IFS) • Online gaze tracking & ICA (ERG)
Test scenario generation	• Simulator Exp. Design (REL) • GreatSPN, automatic scenario variations (UTO)	

Table 9: Specific technologies that are important for most HF methods / analysis in the deliverables D2.2-D5.2

HF in general has a specific need to deal with huge amount of empirical data and the creation of specific test scenarios. For this reason, a common understanding of these two topics seems to be important and may also be part of a HF Meta-Model discussion. Reasonable arguments for this are:

- The definition how data during empirical investigations should be produced, how it should be recorded / stored and how it should be used during system development or analysis could lead to suggestions / requirements on data processing tools.

- Empirical investigations like simulator studies heavily rely on a well-designed experimental paradigm and the correct choice and variation of independent variables. A structured approach for test scenario generation or even an automatic test case generation would be useful for a lot of HF methods.
- The analysis of a specific HF concept like workload poses certain requirements on the data recording and the test scenario.
- If for example a new flight system requires a new operating procedure this is typically associated with a specific flight scenario which needs to be tested (simulated). In turn, to analyse the effect of the new flight procedure specific data collection capabilities are necessary.

It can already be concluded, that both issues are important to almost every HoliDes RTP-workflows. It remains to be clarified if a common understanding can be found and how this can be part of the HF-RTP.

2.6.2.1 Task Meta-Model

The task model which we use in HoliDes (see Figure 7) as Meta-Model is based on an ongoing standardization initiative of the W3C consortium[5].
Entry point for the model is the abstract data type *Task* (see Figure 7), which can be dedicated to one of four categories: *User, Interaction, System* and *Abstract*:

- User task: a task of a human operator, such as selecting a strategy to solve a problem
- System task: performed by the application itself, such as generating the results of a query
- Interaction task: user actions that may result in immediate system feedback, such as editing a diagram

[5] http://www.w3.org/TR/task-models/

- Abstract task: a task that has subtasks belonging to different categories,

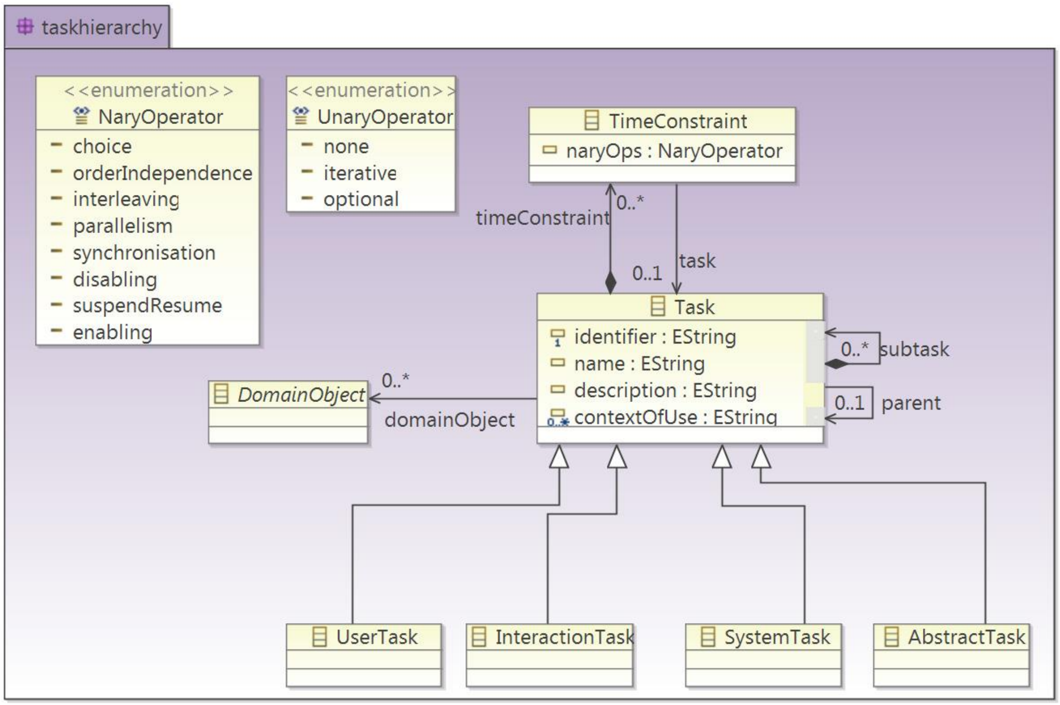

Figure 7: The task meta-model specified as Ecore model.

and thus cannot be allocated uniquely using the previous three categories

A Task can have several sub-tasks (subtasks association). By applying one of the unary operators it can be expressed that a task is either

- *Iterative* (T*): the task is performed iteratively: when it terminates, its execution is started again from the beginning.
- *Optional* ([T]): the task is optionally performed.

Between two tasks, a *TimeConstraint* can be added. These constraints are specified as N-aryOperators and have the following semantics:

- *Interleaving* (T1 ||| T2 ||| ... TN): The connected tasks can be performed concurrently, without any specific constraint.
- *Order independence* (T1 |=| T2 |=| ... TN): the tasks can be performed in any order (DEFAULT).
- *Synchronisation* (T1 |[]| T2 |[]| ... TN): The tasks are concurrent and can exchange information among them.

- *Parallelism* (T1 || T2 || ... TN): The tasks are performed in true parallelism.
- *Choice* (T1 [] T2 [] ... TN): in this case it is possible to choose one task from a set of tasks and, once the choice has been made, the chosen task can be performed.
- *Disabling* (T1 [> T2 [> ... TN): the left-hand task is deactivated once the right-hand task has started.
- *Suspend-Resume* (T1 |> T2|> ... TN): The right-hand task interrupts the left-hand task one. When it is finished, the left-hand task can be reactivated from the state it was before the interruption.
- *Enabling* (T1 >> T2 >> ... TN): There are two cases:
    - Without information passing - when T1 completes it enables T2, when T2 completes it enables T3, and so forth through TN.
    - With information passing, as before, except information is passed from each completed task to the next task.

The Task Meta-Model likewise the CESAR Meta-Model is instantiated in *WP2-Task 2.7 Common Modelling Languages* using a rule-based format which is used for example by the PED and HEE tools of WP2. Please see deliverable *D2.4 Modelling Techniques and Tools version 1.0*. The next deliverable will contain additional Meta Models for Agent and Resource as well as for the HF concepts workload and distraction.

2.6.2.2 Further Meta-Models

Currently the HF concepts for *workload*, *distraction*, *agent* and *resource* are identified but the discussions about them have just started and no common idea can be presented here. Initial models will be presented in the next deliverable D1.5. Finally, *Situation Awareness* and *Usability* are concepts that we consider, but they are not directly part of any development workflow. Therefore it remains open if we will be able to define any meta-model for those two.

For dissemination purposes and also further discussions about HF concepts with a larger community, our plans are to start a new OSLC working group on HF as soon as we have some stable proposals for our meta-models.

2.6.2.3 Adaptation Framework Meta-Model

Adaptation is one of the major issues in HoliDes and WP 3 is dedicated to investigate adaptation concepts and the impact on complexity of the system itself as the development processes and certification. As input and in cooperation with Task 1.6 a meta-model for adaptation is under development and its first ideas are presented here. The goal of this meta-model is a little different compared to the previous ones, e.g. Task: the final version should allow the modeller to get an overview how to deal with the different dimensions of adaptation. Which adaptations are possible and how do they affect the operators and the systems. The underlying concepts and explanations as well as the state of the art on adaptation can be found in the deliverables D3.3 and D3.4 of work package 3.
In WP 3 adaptation is approached on a number of dimensions:

1. A 5-stage process in form of a sequential control loop is defined (Perception, Evaluation, Decision Making, Action Planning and Action Implementation): a high level description how systems as well as humans process information and decide about upcoming actions.
2. Each of the five stages can be performed at a different degree / level of automation.
3. As already stated in the HoliDes proposal adaptations can be made on the level of *Resources, Agents and Tasks*.

Raw Data | Task Specific Data | Selected Action Plan | Selected Actions

Process / Target	Perception	Evaluation	Decision Making	Action Planning	Action Implementation
Task	Required Information	Constraints on integrating data	Proposed Procedure	Proposed Actions	Proposed Use
Agent	Modalities: Vis. Hapt. Acoust.	Data Fusion, how is data integrated	Who decides? Decision Strategy	Who plans action?	Who executes actions?
Resource	Sensor Display	Data Storage / Fusion Software / Hardware	Knowledge Data Base	Which input device(s)	Actual Device use (Interaction)

Figure 8: Two dimensions of adaptation and examples for each category

Figure 8 is a first attempt to create a high level overview how the different dimensions may fit together. The upper column contains the five stages of the control process. The arrows on top visualize concrete input from one stage to the next. They symbolize the information flow, which is step by step condensed from *raw data* until the execution of an appropriate *action*.

The *Target* (also called the "What" in WP3) specifies the three different adaptation targets *Resource, Agent and Task*. Each entry of the matrix lists example items or categories which are impacted if a target at a specific process step is going to be adapted. The arrow on the left symbolizes a hierarchical relation where the upper levels constrain the selection of alternatives on the lower level. This is explained at the end of this subsection. Beforehand, each of the categories is explained in more detail.

Perception Stage

- *Task*: The task puts constraints on the information that need to be perceived by the agent. Thus, if the task is adapted it is likely that these constraints change the way information needs to be perceived. Furthermore, the required sensor hardware (Resource) needs to be capable to gather the required data with the necessary precision, timing and value range.
- *Agent*: The presentation of information to a human operator can be based on different perceptual modalities, e.g. a visual or a haptic display can be used or acoustic warning signals can be given. Dependent on the agent (human / machine) who needs to perceive the data, its abilities constrain the selection of the resources. Adapting one of these signals, e.g. increase the volume of a warning signal, changing the colour of a display or switch from a pure visual to a multimodal information presentation has an impact on the operator who has to perceive the information. In higher levels of automation this additional complexity is removed, as the information can be fed directly into the next automated processing step and not specific presentation devices are necessary.
- *Resource*: An adaptation here impacts the low-level information gathering. The selection of specific technical sensor concepts is dependent on the working environment, the signals that have to be detected, the agents that have to process the signals and finally also the task constraints. If redundant sensors are equipped, adaptations at runtime might be useful to overcome problems with either a sensor defect or if a sensor perceives inaccurate or even no data due to specific environmental conditions, which limit e.g. sensor range. If the data of a

sensor must be perceived by a human agent a display solution has to be selected to present the information.

Evaluation Stage

- *Task*: Certain constraints on how the different data has to be weighted, e.g. the relative importance of the information for a task needs to be taken into account.
- *Agent*: During operation each agent continuously perceives new information during the first phase. Some might be relevant for the current task; others are bottom-up distractions and not directly related to the current task. Nevertheless, they may be important to deal with anyway (e.g. a warning). The agent (human or machine) needs the ability to select the appropriate information for the task and how he has to deal with distracting input. For a human operator the correct information selection is dependent on 1) the knowledge which information is best suited, as well as 2) the level of practical experience. For a classic machine agent the knowledge is programmed at design time, while adaptive systems may also come up with some kind of learning algorithms for data fusion. The most common adaptation on this level would be the integration of raw data into higher level measures.
- *Resource*: Some automatic pre-processing of data can be accomplished using specific data-base algorithms. If large amounts of data have to be processed or real-time requirements exist specific hardware might be necessary.

Decision Making Stage

- *Task*: In certain work areas standardized procedures reduce the number of possible decisions. Adaptations in these procedures can impact the distribution of responsibilities between the agents or the number of decision alternatives that can be chosen.
- *Agent*: One adaptation in this category is the selection of the responsible agent for the decision making. This might be a machine / human or a combination of both. It is also possible to adapt the decision making strategy at this level. Example: an operator is unsure about which alternative to choose and decides to follow the advice of the system or an assistance system contains a set of problem solving algorithms and either they system or a human operator chooses online which strategy is best

suited to solve the current input. Adaptations specifically on this level are a huge part of the research about levels of automation.

- *Resource*: The procedures might be stored for example in an electronic knowledge database or they might be delivered on paper (e.g. handbooks). An adaptation might lead to changes in the work process, shifting from the necessity to manually read a handbook.

Action Planning Stage

- *Task*: The task proposes a number of (alternative) actions that can be taken to achieve the goal of the (sub)-task and can also constrain the resources that have to be used and agents that are allowed to select actions.
- *Agent*: The responsibilities for choosing the correct action plan can be moved from one agent to another or a cooperative planning can be done. It may be a little difficult to subdivide decision making and action planning. One possibility to distinct these two is: decision making selects between concurrent / alternative goals and the order how to achieve them, while the action planning is to choose the appropriate sequence of actions to achieve the selected goal(s).
- *Resource*: Possible adaptations in this category can vary the input devices, which are manipulated by the agent: lever, potentiometer or buttons.

Action Implementation Stage

- *Task*: Certain input devices may have dedicated instructions how to use them. It is for example not advisable to abruptly move the side stick on an aircraft. Adaptations of such restrictions typically change restrictions on the value range, allowed accelerations amplitudes or frequencies and many more.
- *Agent*: Which agent is actually responsible to manipulate e.g. the control process?
- *Resource*: The actual manipulation of e.g. a control lever, the adjustment of process parameter via a potentiometer, etc. can be adapted to situation specific values. For specific manipulation purposes different manipulators with different value ranges (e.g. one for large adjustments and one for fine granular adjustment) can be used.

Within this 2-dimensional categorization the task is interpreted as a set of instructions and constraints how to accomplish a certain goal / action plan and it impacts each level of the process. The constraints and restrictions affect the resources as well as the agents. Changes / adaptations to the task therefore need subsequent analysis if dependent resources can be used or if agents are capable of executing the procedure. Vice versa, adaptations on resource and agent level need to be crosschecked against the restrictions of the task. Accordingly the selection of the appropriate agent puts additional constraints on the resources that can be used, e.g. not every sensor can be used by a human operator directly, if complicated pre-processing is necessary. Also not every process manipulation can be made by all agents in an equal manner (qualitatively and quantitatively). Thus task, agent and resource can be related in a top down hierarchy, where the upper levels put constraints on the selection of alternatives on the lower levels.

What is missing so far is an additional dimension about possible causes for adaptations. The examples mentioned are mostly technical issues, but also HF issues like workload, distraction or fatigue might lead to a necessary adaptation. As already stated in the introduction the table above and its content are ongoing work and it is yet open how it will evolve within the next version. To our opinion it might be a useful entry point to systematically explore possible adaptations of a system starting from a high level process perspective.

The next section will now shift to the implementation issues of meta-models into an Interoperability Standard (IoS) and the necessary infrastructure to set up an HF-RTP.

2.7 HF-RTP Infrastructure & Interoperability Standard

This section is divided into four parts: subsection 2.7.1 provides a definition of interoperability and the levels it considers: organizational, semantic and technical. HoliDes will cover interoperability at both the technical and semantic levels. In addition, it describes the state of the art regarding existing interoperability standards, tool protocols, programming interfaces, runtime aspects, communication paradigms and exchange formats.

In the second subsection 2.7.2, a relation of the different inputs and outputs of a list of tools has been analysed, in order to define a complete specification of the HF-RTP interoperability services that cover all the requirements and needs of HoliDes use cases.

The third subsection 2.7.3 defines generic services useful in HoliDes for parsing and transforming data to HF-RTP format, plug & play installation of components, testing HF-RTP components, and subscribing services. Finally, conclusions are drawn in paragraph 2.7.4.

2.7.1 HF-RTP Interoperability standard

One of the most relevant aspects in this document is to investigate interoperability based on definition of common interfaces for data exchange (defined in the HF-CMM) between models, tools and techniques. The IEEE[6] Glossary defines interoperability as:

- The ability of two or more systems or components to exchange information and to use the information that has been exchanged.

In addition, according to ISO definition:

- A standard is a set of agreed rules and guidelines for common and repeated use for a particular, pre-defined purpose.

Interoperability can be considered at three levels[7]:

- Organizational level. This aspect of interoperability is concerned with how organizations cooperate to achieve their mutually agreed goals. In practice, organizational interoperability implies integrating business processes and related data exchange.
- Semantic level ensures that the precise meaning of exchanged information is understood and preserved throughout exchanges between parties;
- Technical level covers the technical aspects of linking information systems. It includes aspects such as interface specifications, interconnection services, data integration services, data presentation and exchange, etc.

HoliDes covers interoperability at both the technical and semantic levels. Semantic level encompasses two aspects:

- Semantic interoperability is about the meaning of data elements and the relationship between them. It includes developing vocabulary to describe

[6] http://www.IEEE.org
[7] European Interoperability Framework (EIF) for European public services, version 2.0. ISA programme. Interoperability Solutions for European Public Administrations Programme

data exchanges, and ensures that data elements are understood in the same way by communicating parties.

- Syntactic interoperability is about describing the exact format of the information to be exchanged in terms of grammar, format, and schemas.

To support and investigate interoperability at technical and semantic levels, this section is divided into: interoperability standards, tool protocols, programming interfaces, runtime aspects, communication paradigms and exchange formats, explained below.

2.7.1.1 Interoperability Standards

Widespread interoperability across varying systems can only be achieved through reliance upon standard technology interfaces that establish clear rules for communication. While some of these technology standards are *open*, others are *restricted* or *closed*, meaning they don't achieve the highest possible level of interoperability. Open standards have the following essential features[8]:

- Its development and management process must be collaborative and democratic.
- Transparent evolution and management processes are open to all interested parties.
- Approved through process by consensus among participants.
- Faithful implementations of the standard must interoperate.
- Platform-independent, vendor-neutral, and usable by an unrestricted number of competing implementations.
- Openly published including specifications and documentation.
- Available royalty-free or at minimal cost, with the only licensing restrictions being reciprocity and defensive suspension.

Some examples of open standards are:

- HTML is the lingua franca of the Web, the language spoken by every web page and browser
- REST is a standard interface for Web applications.
- CSW is a standard defined by the Open Geospatial Consortium for publishing catalogue records for geospatial data.

[8] http://www.ecis.eu/open-standards/

DCAT is a standard vocabulary for describing datasets in data catalogue

2.7.1.2 Tool Protocols

HTTP is used as a protocol in OSLC. Open Services for Lifecycle Collaboration (**OSLC**) is an open community creating specifications for integrating tools. These specifications allow conforming independent software and product lifecycle tools to integrate their data and workflows in support of end-to-end lifecycle processes. OSLC offers two primary techniques for integrating tools – "Linking data via HTTP" and "Linking Data via HTML User Interface". Both of these techniques build on the HTTP and RDF foundation of OSLC.

Linking data via HTTP: OSLC specifies a common tool protocol for creating, retrieving, updating and deleting (CRUD) lifecycle data based on internet standards like HTTP and RDF using the Linked Data model. This protocol can be used by any tool or other programmatic client to talk to any other tool that implements the specifications. Linking is achieved by embedding the HTTP URL of one resource in the representation of another.

Linking Data via HTML User Interface: OSLC specifies a protocol that allows a tool or client to cause a fragment of the web user interface of another tool to be displayed. This allows a human user to link to a new or existing resource in the other tool or see a preview of information about a resource in another tool. This enables a tool or other client to exploit existing user interface and business logic in other tools when integrating information and process steps. In some circumstances this is more efficient and offers more user function than implementing a new user interface and then integrating via an HTTP CRUD protocol.

RTPS (Real-Time Publish Subscribe) protocol is the interoperability protocol used to allow multi-vendor Data Distribution Service (DDS) implementations to communicate. RTPS protocol is designed for use with Internet Protocol (IP) one-to-many Multicast and connectionless best-effort transports such as IP User Datagram Protocol (UDP).

TCP-IP / UDP connections are often used to integrate real-time simulation components into simulation environments. This can be done either by point to point connections using UDP or TCP/IP sockets or integrating all components into an HLA simulation platform.

2.7.1.3 Programming Interfaces

This section is divided into two parts: the first, presents a study of interfaces of the Distributed Computing Reference Model, and the second presents a list of standardized tools and frameworks that could be used in HoliDes.

Cloud computing tools

There is an interesting study by The Open Group[9], discussing about the interfaces of the Distributed Computing Reference Model (DCRM), describing the standardization work that is needed for cloud computing portability and interoperability.

The table described below shows which components expose and use each interface, why standardization might be beneficial, and which standards or emerging standards might apply.[10]

Interface	Exposed By	Used By	Type	Reason for Standardization	Standards
Data Model	Data	Applications	Description or Shared Understanding	Data Portability Application Interoperability	SQL XML RDF OWL UDEF
Application-Application Interfaces	Applications	Applications	Web Service API Content	Application Interoperability	N/A (but principles of arrangement are important)
Application Management Interfaces	Applications	Management Systems	Web Service API	Management Interoperability	CDMI TOSCA OCCI
Platform Management Interfaces	Platforms	Management Systems	Web Service API	Management Interoperability	CAMP TOSCA OCCI
Infrastructure Management Interfaces	Infrastructure	Management Systems	Web Service API	Management Interoperability Platform Portability (machine image)	CIMI-CIM VMAN OVF TOSCA OCCI Openstack
Publication	Marketplace	Platforms	Web Service	Publication and	ODCA

9 http://www.opengroup.org/aboutus
10 http://www.opengroup.org/cloud/cloud/cloud_iop/interfaces.htm

Interfaces	s		API	Acquisition	SLA@SOI TOSCA
Acquisition Interfaces	Marketplaces	Platforms	Web Service API	Publication and Acquisition	HTTP OVF ODCA SLA@SOI TOSCA
Application-Platform Interfaces	Platforms	Applications	Programmatic	Application Portability	UNIX Linux MS Windows Android iOS Openstack
Platform-Platform Interfaces	Platforms	Platforms	Web Service API Envelope, HTTP and Internet Service Discovery	Platform Interoperability	TCP/IP HTTP SOAP WSDL WADL WS-I UDDI JSON
Platform-Infrastructure Interfaces	Infrastructure	Platforms	Various	Platform Portability (platform source)	Openstack

Table 10: Cloud computing interoperability and standardization issues (by Open Group)

Interfaces Tools/frameworks

Next, a list of standardized tools and frameworks is provided:

- Functional Mockup Interface: defines a standardized interface to be used in computer simulations to develop complex cyber-physical systems.
- Fitman Data interoperability Platform Services Enabler (DIPS) is web Based platform for the management of Data Interoperability services.
- IBM Rational Jazz: integrates task tracking, source control, and agile planning with continuous builds and a configurable process to adapt to the way you work.
- Apache CXF is an open source services framework. CXF helps you build and develop services using frontend programming APIs, like JAX-WS and JAX-RS. These services can speak a variety of protocols such as SOAP, XML/HTTP, RESTful HTTP, or CORBA and work over a variety of transports such as HTTP, JMS or JBI.

In the automotive domain:

- OpenXC: is a combination of open source hardware and software that lets you extend your vehicle with custom applications and pluggable modules. It uses standard, well-known tools to open up a wealth of data from the vehicle to developers. This open source allows consumer devices, such as smart phones, to access data from any vehicle.
- AUTOSAR *(AUTomotive Open System ARchitecture)*, is an open and standardized automotive software architecture, jointly developed by automobile manufacturers, suppliers and tool developers.

2.7.1.4 Runtime Aspects

Models at runtime can be defined as abstract representations of a system, including its structure and behaviour, which exist in tandem with the given system during the actual execution time of that system. Furthermore, these models should be causally connected to the system being modelled, offering a reflective capability. Significant advances have been made in recent years in applying this concept, most notably in adaptive systems.

An interesting paper analyzing some different runtime models and runtime interoperability is [2]. This paper explains how the runtime models can support decision making and reasoning based on knowledge unforeseen prior to the time of execution, but which emerges during execution.

2.7.1.5 Communication paradigms

Different communication paradigms applied by these communication systems require elaborated mapping approaches, which must be standardized in the future. To do so, there are interesting standards in use:

- IEEE Standard for modelling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules.
- ETSI has a number of standards developed to suit different cases in response to market demand, mainly standards for the Radio & Telecommunications Terminal Equipment.

2.7.1.6 Exchange Formats

A data exchange language is a language that is domain-independent and can be used for any kind of data. Most commonly used languages and data formats are RDF, XML, JSON and CVS. E.g. OSLC uses RDF that is based on

XML; Internet media types for data (JSON, XML, Atom and images) are used in Restful services.

For syntactic interoperability, the following tools are needed to transform and translate between data formats:

- Tools to convert between different data formats (XML, JSON and CVS).
- Tools to parse data between the same formats (e.g.: between XML and XML)

2.7.2 Definition

A relation of the different inputs and outputs of a list of tools has been analysed, in order to define a complete specification of the HF-RTP interoperability services that covers all the requirements and needs of HoliDes use cases. Note that these inputs and outputs are intended for the communication requested in the different RTP instances for solving the human factors problems presented in the scope of HoliDes:

SearchBestie

- *Type:* Software analysis
- *Provider:* Brno University of Technology
- *Inputs:* What search engine to be used, setting of execution module, definition of state space, fitness function that provides evaluation of particular states (e.g. test configurations), and tasks to be executed before and/or after exploration process.
- *Outputs:* Statistics about explored state space.

Race Detector & Healer for Java

- *Type:* Software analysis
- *Provider:* Brno University of Technology
- *Inputs:* What program to be analysed, what detection algorithm to be used (Eraser algorithm for detecting violations in a locking policy, AtomRace for detecting both data races and violation of predefined atomicities, healing of detected atomicities, obtaining of correct atomicity) and settings (e.g., what type of noise to be injected, on which program locations, and how strong the noise should be)
- *Outputs:* Warnings about possible data races and atomicity violation and detected atomic sections.

AnaConDA

- *Type:* Software analysis
- *Provider:* Brno University of Technology
- *Inputs:* A path to the analyser to be used, path to the program to be analysed and a space-separated list of arguments passed to the program
- *Outputs:* Data races detected by AtomRace algorithm and all received events

Predator

- *Type:* Software analysis
- *Provider:* Brno University of Technology
- *Inputs:* What program to be analysed.
- *Outputs:* Identified bugs in manipulation with pointers and linked lists.

ProSIVIC

- *Type:* Simulator
- *Provider:* CIVITEC
- *Inputs:*
    - If a real human driver or a driver model is in the simulation loop: pedal stroke/speed orders, steering wheel orders, and others human command (e.g. blinkers, gears).
    - If an external model for the vehicle simulation and/or traffic simulation is used: position and state of each external vehicle.
    - If a base scenario is to be modified by another tool: script files or scripts commands (in Pro-SiVIC® language) associated with modifications to operate.
    - All properties of objects can be changed during simulation run by other applications. The properties to be changed are determined according to specific scenario requirements (for a camera sensor others process could set resolution, frequency, focal, …)
    - If objects that are not provided with Pro-SiVIC® are required: 3D representation of objects (in Pro-SiVIC® format).
- *Outputs:*
    - Simulated raw sensor data (e.g.: for a camera sensor: image set or stream).
    - “Simulated ground truth” information (object position, status, …).
    - States vector of vehicles (CAN bus).

COSMO-SIVIC

- *Type:* Driver model and virtual simulation tool, to support a Human Centred Design approach of Driving Aids
- *Provider:* IFFSTAR
- *Inputs:*
    - Environment information (one time, before simulation):
    - Environment description (building, trees, sign, etc.), and also a road description
    - Environment information (at each step of the simulation), e.g. a movable position
- *Outputs:*
    - Steering wheel
    - Acceleration
    - Gaze position

RTMaps

- *Type:* Software framework
- *Provider:* INTEMPORA
- *Inputs:*
    - Real sensors (including potentially Ergoneers eye-tracker)
    - SQL database (to be developed by DLR for instance to import data in RTMaps which they already have stored in a database)
    - Potentially Tobii glasses
    - Potentially FaceLab 5
- *Outputs:*
    - Simulink (The Mathworks)
    - dSpace microautobox.
    - Real actuators

PED

- *Type:* Task Model and specification
- *Provider:* OFFIS
- *Inputs:* All domain information obtained in advance, i.e. through a Hierarchical Task Analysis, Cognitive Work Analysis, either formalised or as text, which allows the modeller to model the tasks. A list of used systems and its interactions (in form of variables) is necessary for accessing the systems in the procedure.

- *Outputs:* A fine granular formal model of tasks, which an operator has to perform to reach a certain goal, i.e. flying an aircraft or driving a car. More details on PED can be found in Lenk et al (2012).

CASCaS

- *Type:* Cognitive Architecture
- *Provider:* OFFIS
- *Inputs:* Procedure file and variable specification file
- *Outputs:* Requirements addressing

Human Efficency Evaluator

- *Type:* Software
- *Provider:* OFFIS
- *Inputs:* Images of the HMI, task or process model
- *Outputs:* Comma-separated value format file

TrainingsManager

- *Type:* Adaption of Training Syllabi
- *Provider:* OFFIS, TRS
- *Inputs:* Formal task descriptions, as specified with PED
- *Outputs:* A training syllabus in form of a report, containing different Simulator Lessons

GreatSPN

- *Type:* Software framework
- *Provider:* University of Torino
- *Inputs:* All input is provided by the tool together with the modelling and domain knowledge a user needs to have.
- *Outputs:* Result of the use-case is a scenario report – description of how an experiment will be conducted. The tool has no reporting ability, but it would be useful to have one specific for this use-case. Such reporting should read the model and evaluation results (as XML) and build a textual description

CognitiveDistraction

- *Type:* Algorithms / Tool
- *Provider:* TWT
- *Inputs:* In-car audio recordings, face data tracking from the driver
- *Outputs:* Prediction of the driver's degree of cognitive distraction

MOVIDA

- *Type:* Simulator
- *Provider:* IFFSTAR
- *Inputs:*
  - Driver status based on COSMODRIVE (Gaze position)
  - Ego car status (Position on road, Speed)
  - Virtual sensor(Camera, Radar, Telemeter)
  - Driving assistance system (Alert/value from ADAS outputs)
- *Outputs*
  - Rules for adaptation of IFS ADAS
  - (Situational risk estimation)
  - (Driver behaviour consistency)

ADAS

- *Type:* Simulator
- *Provider:* IFFSTAR
- *Inputs:* Virtual sensor
- *Outputs:* Alert or wheel/acceleration control

2.7.3 Generic engineering services

This section describes some of the generic services that could be offered by the HoliDes RTP. Each service description is divided into:

- Id: Service name.
- Input: Data input to treat into the service.
- Output: Data Service Result
- Description: How is the service working? What are the most important features?

Id	**ParseData**
Input	Specific Data Format
Output	Holides Data Format
Description	This component parses and transforms specific formats to HF-RTP formats to achieve interoperability between tools. Standardized data types (strings, integer, date) and generic formats (XML, RDF) allow connectivity between tools.

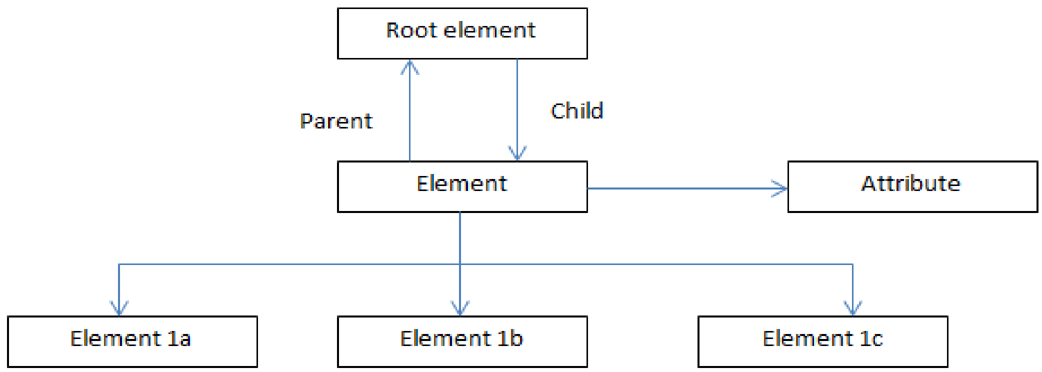

Figure 9 Standard data format for parsing

Id	**InstallComponent**
Input	Connect new component
Output	Install of the new component (not) successful
Description	This component allows installing and configuring new sensors/collectors in HF-RTP as a plug and play feature.

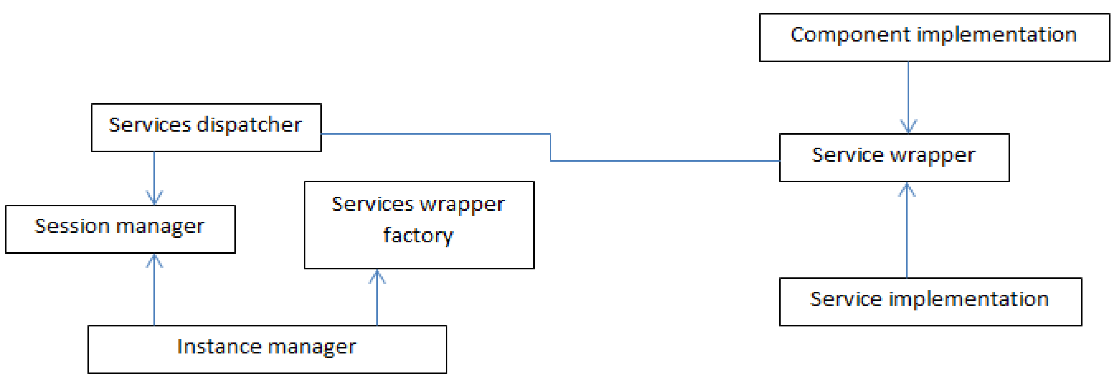

Figure 10 Component installation service diagram

Id	**TestComponent**
Input	Specific Data Format
Output	Holides Data Standard Format
Description	This component tests HF-RTP components against the HF-RTP constraints. It runs as a backend support service, receives

	specific data formats and shows results in a report form.

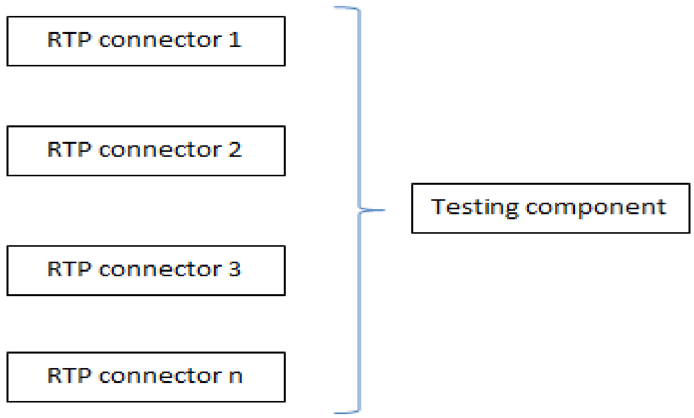

Figure 11 Testing component diagram

Id	**SubscribeServices**
Input	Request to subscribe in a service.
Output	Boolean
Description	Response from publisher/administrators if your petition is accepted or not. This service enables tools to automatically receive messages from a publisher.

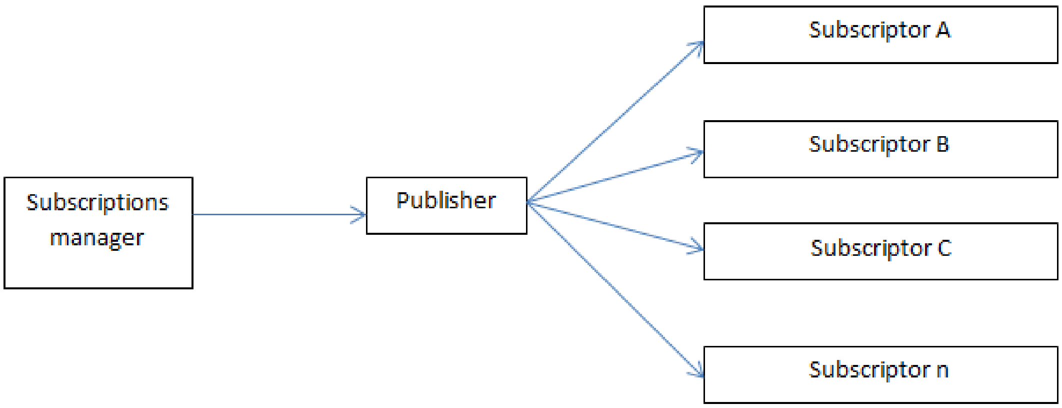

Figure 12 Subscription manager diagram

2.7.4 Interoperability conclusions

This section provides the state of the art regarding existing interoperability standards, tool protocols, programming interfaces, runtime aspects, communication paradigms and exchange formats as well as the data management requirements and the generic engineering services proposed for designing a HF-RTP infrastructure.

As mentioned in previous deliverable D1.3 (see section 2.9 Interoperability Syntax and 5 Conclusions), HoliDes RTP will be built using the same principles as other RTPs and will define the concepts specific to Human Factors which do not exist at the moment. Deliverable D1.3 contains detailed technical specifications and conclusions about why this technology suits for the technical communication requirements of the HoliDes project.

To sum up, HoliDes will reuse the approach of projects CESAR, MBAT and Crystal and will use OSLC specifications for interoperability in HF-RTP. The decision for OSLC was made, because the project on the one hand reuses results of CESAR and shall also be compatible with future developments in the project Crystal.

2.8 HF-RTP Tailoring Rules & Deployment

2.8.1 First approach to Tailoring – Tool selection and integration

A first approach to the tailoring process has been sketched in D1.3. According to this approach, tailoring is about

"*selecting tools and services for the project at hand and creating an instance of the RTP which should be apt to address the needs of a specific project.*"

If we start from the assumption that an RTP is basically represented by a list of tools and services which are all IOS compliant, then the basic idea for the tailoring process consists in performing two steps:

1. From the list of tools and services describing the RTP, select those tools and services to be used in a given (development) project.
2. Integrate the tools and services by providing a suitable way to exchange data between them.

The result of this tailoring will be a project-specific instance of the RTP. When tailoring the RTP for a domain specific development process, meta-models and respective access methods have to be adapted in a way that these meta-models can be used in the RTP (CESAR Project-book).

The second point (the integration of tools) relies on the interoperability specification. All the tools are supposed to comply with the IOS as a precondition for their integration. One important part of the IOS will be that the tools and services apply to the principles for linked data fixed in the OSLC specification (Open Services for Lifecycle Collaboration).

Human-Factors integration into the tailoring process

However, there are a number of reasons, why this approach based on the tailoring approach in the CESAR project does not completely address the needs of human-factors-integration into development processes:

Human-Factors-related activities in many cases are not clearly formalized and might rely on informal methods with unclear specification how results should be presented. Activities connected to certification needs defined in the application domains sometimes use "common practice" methods rather than specific tools to fulfil certification requirements. These practices may include, for example, the Code of Practice (automotive)[11] or methods derived from User Centred Design (medical). Again, in these domains regulations are rather unspecific concerning HF activities. An overview of relevant regulations in the application domains has been provided in D1.2. Many concepts and regulations provide guidelines on issues or requirements to be met. But most of the concepts and regulations are unclear about methods or tools that shall be used. Subsequently the level of formalization of results of HF activities is low. Data formats are left open.

To be able to integrate informal methods into the RTP as well as with the supporting tool-chain, we suggest defining a library of scripts describing common practices of used methods. A script contains descriptions of sets of methods including instructions about how to apply the methods and how to document the methods and the results. These scripts can be used to fulfil the certification requirements, thus providing standardized guidelines for developers and authorities.

The tailoring process then needs to be appended by the choice of methods and their documentation. The description of methods and their results should use sets of shared descriptors, defined in an HF-ontology. The Tailoring then should include the following steps:

Step 1: Identification of the purpose of the project and the used tool chain

The first step of the tailoring process will be the definition of the overall purpose of the RTP within the (development) process to be supported. The purposes should be described in terms of the planned workflows; the expected outcomes of each step within the workflows and include the human-factors and certification criteria.

As far as the description of workflows is concerned, the tailoring process uses parts of the regular engineering process. Moreover it also references to an HF integration plan and to HF standards and regulations where needed.

Step 2: Selection of methods and tools

[11] http://www.acea.be/uploads/publications/20090831_Code_of_Practice_ADAS.pdf

In the second step, related methods and tools are selected to fit the needs of the project purpose. A library of generalized descriptions of methods should support the method selection. A HoliDes tool list supports the tool selection. A lot of methods, tools and services are already used in the development workflows described in the HoliDes application work packages (WP6-9).

Problems might be caused by existing gaps of missing methods, not described by a script in the method library, by missing tools, or by insufficient integration or interoperability. However, not every method and tool needs to be integrated with every other method and tool. Therefore, the integration engineer has to analyse the missing pieces of the current development environment with respect to integration or interoperability aspects on the one hand. On the other hand, the engineer needs to take into account the development process and specific workflows (CESAR project book).

In the CESAR project, the next tailoring step was to select an integration platform. Late in CESAR a decision was made to use OSLC, which is also used by successor projects MBAT and Crystal. Therefore in HoliDes, we decided to rely on the OSLC specification as a general precondition of implementing and tailoring an RTP instance.

Step 3: Definition of semantics and information mapping between methods and tools

The next step is to decide which level of information exchange is needed for each of the RTP instances. Within the RTP approach, we try to avoid a direct mapping between methods and tools. In theory, direct mapping of methods and tools would lead to *(n*(n-1))/2* connectors needed in case of unidirectional data exchange. This means a quadratic growth of the number of connectors with growing numbers of methods and tools in an RTP instance. Instead a common semantic approach is needed to generalize the information exchange across several methods and tools. Theoretically, this reduces the integration effort to a linear complexity. If a tool is integrated to the common semantic representation it is integrated to all other methods, tools and services. The common meta-model approach should support this together with the Human-Factors ontology.

One of the lessons learned from RTP implementation in other projects is, that a promising strategy is to mix the different approaches of point-to-point and common semantics. In several domains already a lot of domain knowledge is included in standardized models such as requirement descriptions with ReqIF, testing concepts with UML Testing Profile (UTP) or business process descriptions with SPEM and BPMN. The potential numbers of connections can be significantly reduced by introducing domain specific languages (CESAR project book). However a Human-Factors specific semantic is missing. Therefore a Human-Factors ontology needs to be developed as a basis for both, the description of informal methods and their outcomes as well as the common semantic basis for the human-factors and certification activities with their related methods to be integrated in the HF-RTP.

Step 4: Implementation of information models and connectors.

Once the common semantics and mappings have been decided upon, the information management has to be implemented. This is mainly an engineering task.

Conclusions

To be able to apply the tailoring process to the HoliDes use cases, a number of preconditions need to be fulfilled:

With respect to the development workflows (WP6-9):

- the purposes and intended workflows of the projects need to be described (done in deliverables D6.2 - D9.2), and
- a choice of methods and tools to be used needs to be compiled (partly done so far for only some workflows).

With respect to the research work packages which develop the HF-RTP (WP1-5):

- A library of informal methods should be provided (including methods used for certification purposes). Each of the methods is described by one or several scripts, explaining what should be done when and how with which outcome (upcoming work for deliverable D1.5).
- A Human-factors ontology provides generalized descriptors for the scripts as well as for a Human-Factors semantic domain (upcoming work in the next deliverable D1.5).
- A Common Meta-Model is provided (Task 1.6).
- An interoperability standard is agreed upon (done: OSLC).

The development tools created by HoliDes partners should be OSLC compliant or in case of new HF tools for which no OSLC domain exists so far they should rely on the data types defined in the HoliDes Common-Meta-Model.

2.9 Addressing Reviewers Comments First Year

The content of this section is not available in the public part of the deliverable.

3 Literature

[1] Baumgart, A. (2010). *A common meta-model for the interoperation of tools with heterogeneous data models*. ECMFA 2010, 3rd Workshop on Model-Drive Tool & Process Integration.

[2] Bencomo, N., Bennaceur, A., Grace, P., Blair, G. and Issarny, V. *The role of models@run.time in supporting on-the-fly interoperability.* Computing, March 2013, Volume 95, Issue 3, pp 167-190