HoliDes

Holistic Human Factors **Des**ign of Adaptive Cooperative Human-Machine Systems

D1.5 - HF-RTP Version 1.5 incl. Methodology and Requirements Analysis Update

Project Number:	332933
Classification:	Public
Work Package(s):	WP 1
Milestone:	M 3
Issue Date:	<16.06.2015>
Document Timescale:	Project Start Date: October 1, 2013 Written Feb 2015 - Jun 2015
Deliverable Overview:	**Main document:** D1.5 – HF-RTP Version 1.5 inc. Methodology and Requirements Analysis update. <Public>
Compiled by:	Sara Sillaurren – TEC
Authors:	Sonia Bilbao – TEC, Sara Sillaurren – TEC, David Kaethner – DLR, Elisa Landini – REL, Fabio Tango – CRF, Ian Giblett - EADS-UK, Denis Martin – TWT, Marie-Cristin Ostendorp – OFF, Nacho González – ATO, Jordi Fonoll – ATO, Linda Onnasch - HFC, Harald Kolrep - HFC, Ernst Hermens, PHI
Reviewers:	Internal Reviewer: Morten Larsen, AWI External Reviewer: Robert Hofsink, PHI
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
10.02.2015	Started writing initial document	Sara Sillaurren, TEC
18.03.2015	Received initial comments	David Kaethner, DLR Elisa Landini, REL Fabio Tango, CRF
23.03.2015	Contributions for Command and Control Room Workflow and review of section	Ian Giblett, EADS-UK
16.04.2015	Update on Command and Control Room workflow	Ian Giblett, EAD-UK
17.04.2015	Contributions for Requirements Update	Denis Martin, TWT
17.04.2015	Contributions for the Common Meta Model	Marie-Cristin Ostendorp, OFF
22.04.2015	WP6-9 AdCoS HF data interchange workflows described in UML diagrams	Nacho González, ATO
27.04.2015	Update on Method Library (Descriptor manual and excel file)	Linda Onnasch, HFC
12.05.2015	Update for the Common Meta Model section	Marie-Christin Ostendorp, EAD-GE
28.05.2015	Update on HF ontology section	Sonia Bilbao, TEC
29.05.2015	Update for Annexes related to Platform Builder	Jordi Fonoll, ATO
29.05.2015	Update in section about Tailoring rules	Harald Kolrep, HFC
15.06.2015	Methodology for Integration Plan	Elisa Landini, REL Sonia Bilbao, TEC Ian Giblett, EAD-UK
15.06.2015	Contributions for status baseline and PIs	Ernst Hermens, PHI
25.06.2015	Internal/External Review	Morten Larsen, AWI Robert Hofsink, PHI
03.07.2015	Final version	Sara Sillaurren, TEC

Table of Contents

Table of Figures

Table of Tables

Introduction

This document describes the progress of the HoliDes consortium to develop a Human Factors Reference Technology Platform (HF-RTP), at its version 1.5. 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.

Section 1 will review what has been achieved for the moment in version 1.0 of HF-RTP and presented in previous *deliverable D1.4*.

Section 2 will give the reader an overview of the HoliDes approach which led to the third version of the HF-RTP: version 1.5. In this section, a new version of the updated requirements will be delivered. Also it will be analysed the Human Factors data flows in the different AdCoS. Another subsection will be related to the Common Meta Model and the HF ontology (based on the defined Human Factors method library).

In this HF-RTP version 1.5, after having delivered the Tailoring rules in the previous version, the results of the tailoring applied in the different AdCoS will be analysed.

Also in this section 2, it will be defined a methodology for an Integration Plan into the HF-RTP, that will be used for all the MTTs being developed in WP2-WP5.

The main advance in this version of the HF-RTP is the development of a Platform Builder, a solution for improving the configuration and instantiation capabilities of the HoliDes HF-RTP.

1 HoliDes HF-RTP version 1.0

The previous version of the HoliDes HF-RTP was delivered in *D1.4*. In that version the different tasks of WP1 were analysed, except the one referring to Human Factors and Safety Regulations (T1.3) as these were fully reported in *D1.2*.

A brief summary of version 1.0 of HoliDes HF-RTP is given in Appendix A.

2 HoliDes HF-RTP version 1.5

The focus of this deliverable version 1.5 has been shifted compared to the previous *deliverable (D1.4)*.

First, the HF-RTP requirements have been updated from the previous version 1.0. TWT has gathered feedback from the domain workpackages 6-9, with the task of re-evaluating the importance of the HF-RTP requirements with respect to their expectations (i.e., what they do expect from the HF-RTP).

Second, the section in *D1.4* about the Human Factors integration concepts has evolved to a section concerning the HF-RTP Common Meta Model and the Human Factors Ontology. In this chapter, the meta-models concepts that are under current discussion in HoliDes are presented. The second subsection deals with a definition of a Human Factors method library, which is the basis for the creation of the HF ontology (which defines the interoperability standard to exchange information related with human factors including methods, human factor issues and regulations). These three concepts together will serve as basis to state the interoperability and integration of the MTTs in the HF-RTP.

In *D1.4* the AdCoS methodology and HF-RTP architecture were defined. In this deliverable they are no more referred to, as no updates have been made.
Third, the HF-RTP processes and workflows which were analysed in the previous deliverable have evolved to the AdCoS Human Factors Workflows, in

order to state the Human Factors data flow in the different AdCoS (as basis for the HF ontology definition).

Fourth, a new section by EAD-UK has been included in this deliverable about tracking the RTP tailoring. The main goal of this section is to clarify the difference between tailored and non-tailored components with a fictitious example. At the same time, the RTP integration plan has been defined. To achieve this goal, a methodology for this integration plan has been defined in this deliverable. This methodology will be followed in the WP2 to WP5 (and in their correspondent deliverables for this milestone). The result of achieving the first step of this methodology will be reflected in an Annex about MTTs integration.

Furthermore, the first steps for defining an HF-RTP infrastructure & interoperability standard have been given in the correspondent section.

As an evolution from the previous version 1.0 of the HF-RTP in which the tailoring rules were defined, in this version 1.5 an analysis of the tailoring rules applied to the different AdCoS (and reflected in Dx.4) has been done and broadens this section.

Fifth, another main innovation in this version 1.5 of the HF-RTP is the Platform Builder, as a solution for improving the configuration and instantiation capabilities of the HoliDes HF-RTP. Based on an HF problem description, the Platform Builder aims to set up an instantiation of the HF-RTP for a specific domain project.

Last but not least, the baseline definition process was defined in the previous version 1.0 of the HF-RTP (D1.4). These were used for the comparison between this reference and target states according to the selected Key Performance Indicators (KPIs). In this deliverable a summary of the work done so far for the baseline and Performance Indicators is done, together with the next steps to follow in this 2^{nd} project year.

2.1 HF-RTP Requirements Update

The last version of HF-RTP requirements compiled in *D1.4* summarized requirements that are necessary for the development process of AdCoS being designed to improve human-machine interaction. Moreover, these requirements were estimated to be of cross-domain relevance, and thus not only being requested from one of the HoliDes domains but from several HoliDes domains (which are Health Care, Aeronautics, Control Rooms, and Automotive).

For the current version of the requirements (see Annex I: Requirements Update) and as part of Task 1.1, feedback from the domain workpackages 6-9 was collected. Their task was to re-evaluate the importance of the HF-RTP requirements with respect to their expectations (i.e., what they do expect from the HF-RTP). Besides rating the general comprehensibility of the requirements, they specifically needed to state whether each requirement was relevant in their domain. In addition, they had the opportunity to add requirements that were missing until now.

With the collected feedback, smaller changes were made concerning the phrasing of the requirements in order to define them more precisely. In addition, three requirements were removed (cf. Table 1) while two others were added (cf. Table 2).

ID	Requirement Name	Reason for removal
WP1_HFRTP_REQ08_v2	Status reporting	Merged with WP1_HFRTP_REQ05_v3 (now "Error and Status Reporting")
WP1_HFRTP_REQ24_v2	Output report with screenshots	Too specific (screenshots are not always applicable)
WP1_HFRTP_REQ28_v2	Tester-independent validation	Not realistic. While tester-independent validation of, e.g., HMIs would be valuable in some scenarios, its applicability is situational and automated testing without real test persons is a very challenging task. Thus, no general solution can be offered by HoliDes.

Table 1: HF-RTP requirements that were removed during this iteration

ID	Requirement Definition	Requirement Type
WP1_HFRTP_REQ38_v3	The HF-RTP shall provide at least one MTT for the management of textual requirements.	MTT
WP1_HFRTP_REQ38_v3	The HF-RTP shall provide at least one MTT for the modelling of human views in an architectural framework.	MTT

Table 2: New requirements added to the list of HF-RTP requirements. For more details see Annex I Requirements Update.

The feedback from the application domains not only verified the necessity of the remaining requirements but also helped to identify synergies between domains in order to develop a holistic RTP for all application work packages. In this process, Task 1.1 could establish new discussions between partners from different application domains (e.g., with respect to the requirement WP1_HFRTP_REQ04_v3 – "Storage of experimental data").

The feedback from the application domains shows that most requirements were categorized as relevant in at least two domains. Some, however, were not marked as relevant by any domain or by just one domain. Those requirements are listed in Table 3.

ID	Requirement Definition	Requirement Type
WP1_HFRTP_REQ03_v3	HF-RTP MTTs shall work according to standard Meta Models and the IOS in order to provide consistent information exchange, so that all related MTTs work with the same information and are able to interact with each other.	Interface
WP1_HFRTP_REQ21_v3	The HF-RTP shall provide MTTs for knowledge storage. A design-time knowledge base can store, e.g., checklists for HF-related design issues and thus, can be reused for different projects (learning from experience).	MTT, Workflow
WP1_HFRTP_REQ30_v3	All RTP tools must be IOS compliant (IOS: InterOperability Standard of the HF-RTP).	Interface
WP1_HFRTP_REQ32_v3	The HF-RTP shall provide means for transporting project identifiers to distinguish each AdCoS design and development project to be used by all tools in the same RTP instance.	Workflow
WP1_HFRTP_REQ33_v3	The HF-RTP shall provide means for transporting unique version identifiers to distinguish each design and development cycle of the same project with the same RTP instance. The identifier must be available to all tools in	Workflow

	the RTP instance that require it.	
WP1_HFRTP_REQ36_v3	If a single RTP instance should support the development of multiple AdCoS, there must be one MTT which can create a new project for the development of the new AdCoS.	MTT

Table 3: HF-RTP requirements only considered relevant by one or none domain. Note that the descriptions here already contain the updated versions and not the versions for which the feedback was collected.

Requirements of type "Interface" do not originate from domain requirements but are requirements induced by the RTP concept: Communication between RTP tools must be standardized in order to allow a tailored setup of an HF related development cycle. Since tools currently used by the respective domains (i.e., non-HF-RTP tools) already can interoperate within the respective domain, it does not seem to be required to create such an interoperability standard (IOS) from the perspective of a single domain. However, to allow cross-domain usage of the different MTTs, an IOS is essential. Thus, related HF-RTP requirements are kept with a priority of "High" to "Very high".

Requirements 32, 33 and 36 result from the needs of some partners to use a single RTP instance for developing multiple AdCoS or multiple versions of the same AdCoS. An alternative to such an approach is to use multiple RTP instances, where one instance is used for only one AdCoS (resp. one version of the AdCoS). As a consequence, we changed the priority of those requirements from "High" to "Low".

Last but not least requirement 21 (integrated tool support for knowledge storage across RTP instances) was only deemed relevant by the Health Care domain. While knowledge storage itself is relevant in all domains (and present in the form of personal experience and shared knowledge storages such as wikis), an integration of this knowledge storage into a development tool chain doesn't seem to be important in other domains. Thus, we reduced the priority of this requirement from "High" to "Medium".

The HF-RTP requirements will be continuously tracked and updated for future RTP versions. With the next update in *D1.6*, we will focus on the current coverage of the requirements.

2.2 AdCoS Development Processes and Work Flow

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

2.3 HF-RTP Common Meta Model and Human Factors Ontology

"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 deliverable of WP1 the development workflows have been defined, so that common concepts can be identified for a meta-model.

As a starting point section 2.3.1 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-Models concepts that are under current discussion in HoliDes are presented. At the current stage none of these models is already finished and some are only identified as candidates and have not been discusses a lot. Thus the overall Meta-Model will develop over the next few deliverables and the current state can only be considered as intermediate. The next steps are presented in the following section.

2.3.1 Common Meta Model definition

This section is linked to Task 1.6 (Definition & Implementation of the HF-RTP Meta Model). According to the proposal, in Task 1.6 the Human Factor Common Meta Model (HF-CMM) will be defined and implemented with new concepts for AdCoS processes, techniques and tools. The following section presents the specification of an initial version of the HF-CMM including a first definition for a standardization of shared artefacts. This HF-CMM will be used in Task 1.7 as a basis to define and implement the interoperability standard.

"If tools want to communicate, they need to understand the data they exchange. […] The RTP provides an interoperability platform which enables communication between tools. The [Common Meta Model] enables common understanding of exchanged data among tools. "
CESAR Deliverable D_SP1_R3.2_A_M2

This aspect is visualized exemplarily in Figure 1, showing an example for work to be done to achieve the HF Common Meta Model (HF-CMM) specification: During the development process of an AdCoS, different Tools and Techniques can be applied. These tools and techniques have different underlying models. To allow data exchange between different tools, these aspects need to be synchronized. Therefore equal concepts – which have the same semantic in different models but using a different wording – have to be identified and a common element in the HF Common Meta Model has to be created.

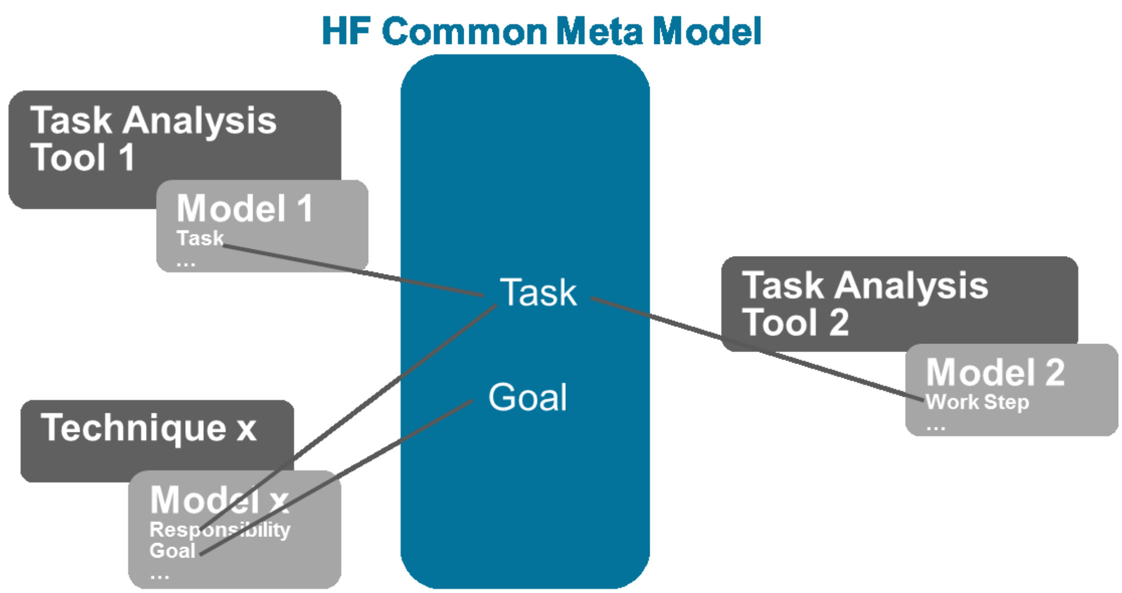

Figure 1: Example for work to achieve HF Common Meta Model – synchronizing in- and output of different tools and techniques in development processes for AdCoS.

So far in the previous deliverable of WP1 the development workflows have been defined, so that common concepts can be identified for a Meta Model.

As next steps the data that can be exchanged via tools in a tool chain needs to be identified and formalized as part of the HF Common Meta Model. As presented in *deliverable D1.*3 the HF-CMM needs to only be a **minimum** set of data that is exchanged between each stage. There will inevitably be huge amounts of data types that are used at each stage which will only be interesting to the tool or specialist concerned at that particular stage. The

Common Meta Model will comprise concepts that are exchanged between tools during the design process. There is no point in spending work effort to define information types that are simply not interesting to other tool in the tool chain. Therefore in the **Annex VIII:** Table for the specification of the HF-CMM is described a table which provides help to establish the data that can be exchanged between different tools and allow identifying equal concepts and connections between elements of the HF Common Meta Model.

The table has been completed using the information specified in the workflows, the results presented in *D 1.4* and the MTT table filled during the Consortium Meeting in March 2015 in Berlin. Tools with no interest and no usage has been ignored (these are AnaConDA, Race Detector & Healer, Predator, ADAS, AEON, AMAS, Operator Pattern Classifier). The completed table can be found in the Appendix C (
Figure 24).

With help of the table presented in **Annex VIII**, common concepts and data can be identified and synchronized. The synchronized elements have then been specified as part of the HF-CMM in Ecore.

The initial version of the HF-CMM is shown in Figure 34, Appendix C.

The common concepts which could be identified and which are exchanged by tools are:

- *Simulations* – which can have a name, a description and a domain
- *HMI* – which has a name and a url (where a picture or similar can be found),
- *Scenarios* – which are used in simulations and
- *Sensor Data*. Sensor Data has as attributes a name, a description and a url (where concrete values can be found).

Furthermore it can be specified if sensor data is real time and/or simulated or real. The sensor data can be further subdivided into:

- *environmental data* e.g. other cars in a traffic simulation,
- *system data* e.g. data concerning the steering wheel or acceleration in a traffic simulation and
- *human data*.

Human data can for instance be data about:

- the *physiology*,
- about the *perception* (having then a perceived element as attribute),
- about *sensorimotor* actions (having then an action (e.g. push),
- a resource (e.g. left hand),
- a string defining the object on which the action is performed and a value (e.g. if something is adjusted to a specific value)).

Moreover human sensor data can also be *speech* with the attributes text and target.

All sensor data have a *sensor source*. This source defines the data format for the sensor data and can be an eye tracking device, a camera, a physiological measuring device or an in system sensor.

Further steps to be taken in this task are the integration of the Human Factors Ontology (see *Prediction* which can be subdivided into a prediction of *workload, distraction, situational awareness, intention* or *fatigue*). Furthermore the common modelling framework provided by WP2 will be included into the HF-CMM. Thereby the task model, the resource model, the communication model, the cooperation model, the human operator model, the HMI interaction model and the training model will become part of the HF-CMM presented in
Figure 24.

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." The integration into the CESAR/CRYSTAL MM is covered in an upcoming project called CPSETIS (www.cp-setis.eu) and thus can be neglected in HoliDes.

2.3.2 Human Factors method library

In task 1.4 the need of a formalization of methodologies in relation to human factors (HF) issues is posed. This addresses two issues:

- First, the question arises which concrete techniques and tools are needed and where to integrate these within the engineering life-cycle.
- Second, clear guidelines are needed for answering requirements posed by authorities by suggesting different HoliDes techniques and tools as means of compliance.

The human factors method library addresses these issues by collecting and structuring different methods that can be applied to evaluate HF issues within the engineering life-cycle.

The goal of the human factors method library is to collect different methods that can be applied to HF issues like usability, situational awareness, workload and user distraction and to categorise them in a standardised manner. The motivation for building such a method library is that most method integration approaches to this day do not provide an appropriate level of formalisation. Most general approaches as well as standards and regulations only suggest frameworks for HF-related workflows but fail to describe concrete methods to be used.

To overcome this lack of formalisation, a defined set of descriptors is needed, which can be applied to different HF methods to make those comparable. The descriptors should give exclusive information about

- which HF issue is considered by the method in question,
- how the method can be used,
- what is needed to apply the method,
- how results are achieved and
- how the outcome can be interpreted.

Based on these requirements, a first set of 13 descriptors was developed. The descriptors should be sufficient to describe a method in such detail that it can be compared to other methods that are available for a certain HF activity within a system development process. The information provided by the descriptors should furthermore enable non-HF experts to choose a suitable method for their purpose, e.g. a method to assess situational awareness in the evaluation phase of system development.

To allow other parties to work with and extend the Human Factors method library with new HF methods, a manual was written, which explains the descriptors in detail. The manual is arranged in table format. The following figure explains how to read the manual (Figure 2).

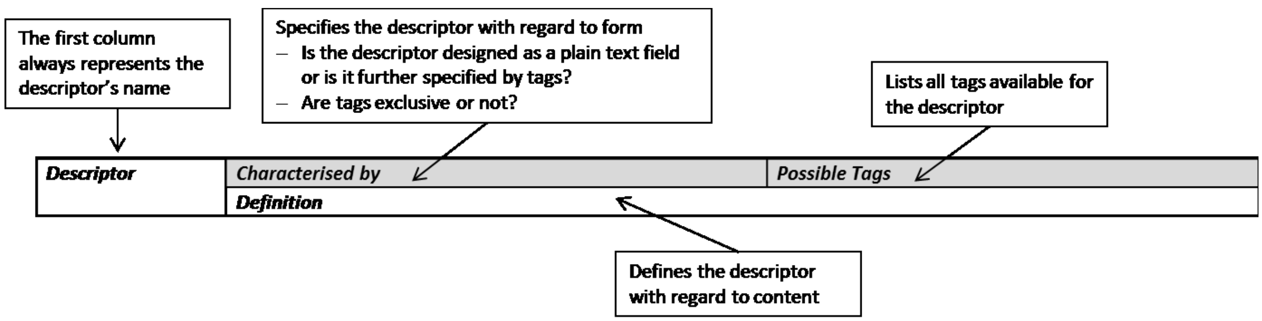

Figure 2: Table structure

The manual is presented in Table 4: Descriptor manual for the Human Factors method library

ID (Identifier)	Text field	
	Name of the described method	
Short Summary / Claim	Text field	
	This descriptor provides a short summary of the method and what the method claims to do, respectively.	
HoliDes V-Model Phase	1 or more of 7 possible tags	**requirements engineering** **conceptualisation** **design** **system implementation** **evaluation** **certification** **deployment**
	In which phase within the system development process is the method applied? The seven possible tags represent phases of the v-model of system development that is applied in HoliDes. The v-model links HF activities and methods to the classical engineering approach of system development (see Figure 3). The tags are related to the latter approach, the system engineering. This makes it more feasible for non-HF-expert to select appropriate HF methods.	

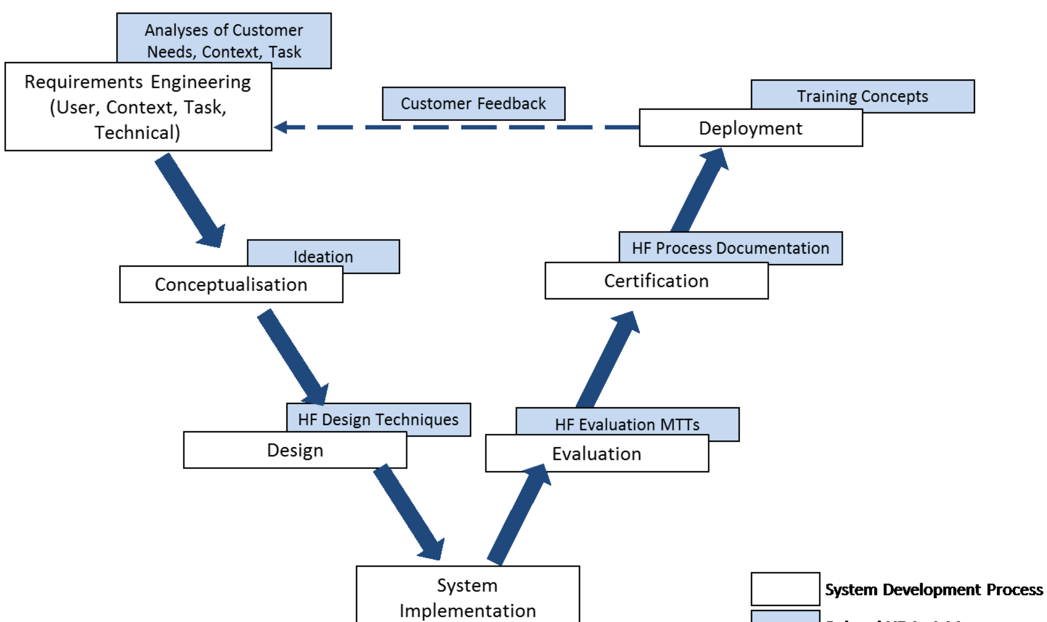

Figure 3: V-Model of System Development as applied in HoliDes

HF Phase	1 or more of 3 possible tags	**concept** **prototype** **operation**
	When in the system lifecycle does it happen? As a very simple first approach to this dimension, we distinguish just three lifecycle phases that are **concept, prototype,** and **operation**. The basic idea is that each of the phases might be iterated several times until a satisfactory intermediate result has been found before turning to the next phase.	
HF Issue	1 or more of 3 possible tags	**usability** **situational awareness** **workload** **user distraction**
	Which HF problem is addressed? Current options (possible tags): **Usability, situational awareness, workload, user distraction**	
Data Type	1 or more of 4 possible (sub-) tags	**subjective:** **actor** **observer** **objective:** **psychophysiological** **performance**
	What kind of data is recorded? This descriptor differentiates whether data comes from subjective assessments or objective assessments. **Subjective**: A user of a system (**actor**) fills out a questionnaire; an **observer** collects data by observing how a user performs a task. **Objective**: Data is obtained by **performance** measures like reaction times or by **psychophysiological** measures like EEG, EDA or eye-tracking.	
Mode of Data Collection	1 or more of 5 possible tags	**experiment** **expert inspection** **observation** **interview** **questionnaire**
	In which way is the data collected? Does the method require conducting an **experiment**, does it need an expert to analyze the task (**expert inspection**), can data be collected by **observation**, in form of an **interview** or **questionnaires**?	

Time of Data Collection	1 or more of 3 possible tags	**prospective** **real-time** **retrospective**
	When is the data obtained? Methods can be either applied **prospective**, **real-time** or **retrospective**. Prospective means that people never have interacted with the system they are going to evaluate, e.g. because the system does not exist right now or to ask for expectations people have of the system they are going to interact with. Real-time data collection means that the data is obtained while someone is performing a task with the system in question. Retrospective means that the data is collected after the user has interacted with the system. For example, questionnaires are often provided to users after they performed a task to evaluate workload or usability.	
Method applied by	1 or more of 3 possible tags	**HF-expert** **domain expert** **non-expert**
	Is expert knowledge mandatory to apply the method? The descriptor determines what kind of knowledge level someone must have to apply the method and to appropriately interpret the gained data. Does the method require an **HF-expert**, a **domain expert** or can the method in principle be applied by everyone (**non-expert**).	
Interpretation of Outcome	Text field	
	This descriptor specifies mandatory requirements for the interpretation of results.	
Effort (time)	1 of 2 possible tags	**high** **low**
	How much time is needed to apply the method? This descriptor not only takes into account the actual time of method application, but also time that might be needed to adapt the method to the situation or domain it should be applied to and the time that is needed for analysis of results.	
Costs	1 of 2 possible tags	**high** **low**
	Is it expensive to apply the method? Whereas the use of a questionnaire might not be expensive as only paper & pencil is needed (minimum requirement) the use of other methods like EEG are expensive in terms of asset costs and method application which could be, for example, time-consuming. Other aspects that can raise the costs are the need to conduct an experiment in a laboratory environment with a lot of participants or the need of an external HF-expert.	

Resources	1 or more of 8 possible tags	**paper & pencil** **eye-tracker** **simulation environment** **experimental lab** **participants** **EEG, …** **Video / Audio Recording** **computer**
	This descriptor contains a checklist defining the technical and human resources needed to apply the method.	

Table 4: Descriptor manual for the Human Factors method library

The result of applying the descriptors to different HF-related methods is a table which structures the methods according to the descriptors. This output is meant as a basis for the HF ontology. The goal of the HF ontology is to describe HF concepts, methods and their outcomes. The method description is covered by the Human Factors method library. Furthermore, the method library is an essential basis for building future HF-RTP instantiations using the Platform Builder. One part of building an HF-RTP instantiation is to select appropriate methods to consider HF issues during the system development process. The descriptors within the Human Factors method library can be used as selection criteria and therefore provide a sound basis within the Platform Builder to decide whether a method is suitable or not for an actual scenario.

Furthermore, the Human Factors method library will be part of the RTP-Tracker. The RTP-Tracker was established to track the status of the different RTP components. Sections in the RTP-Tracker include an RTP Integration Plan and also the HF-RTP integration methodology. The method section is covered by the Human Factors method library presented in this chapter.

The work done so far can be reviewed in the Annex II: HF Method Library.

2.3.3 Human Factors Concepts Ontology

The HF ontology is part of the work developed in task 1.7: Definition & Implementation of the HF-RTP Infrastructure & Interoperability Standard & Generic Services.

This ontology defines the interoperability standard to exchange information related with human factors including methods, human factor issues and regulations that apply to each of the four domains covered in HoliDes project. This ontology formalizes in OWL language the concepts identified in the HF-CMM (Task 1.6)

The methodology used to build the HF ontology is based on the methodology by Uschold and King[2]. It contains the following steps[3]:

1. Identification of the purpose. It is important to be clear why the ontology is being built and what its intended uses are. In HoliDes, the ontology is built to provide a common model and format for human factors modelling and exchange.
2. Identification of the key concepts and relationships in the domain of interest, i.e. scoping (in all AdCoS and HF-RTP tailoring processes).
3. Provide precise unambiguous text definitions for such concepts and relationships
4. Identification of the terms to refer to such concepts and relationships (classes and properties of the ontology)
5. Coding of the ontology in RDF/OWL language. This step involves explicitly representing the knowledge acquired in previous steps in a RDF/OWL language.
6. Integrating existing ontologies. During either or both of the capture and coding processes, there is the question of how and whether to use ontologies that already exist.

Following the above methodology, the HF ontology will be used to model:

[2] Uschold, M. Building Ontologies: Towards A Unified Methodology. Expert Systems 96. Cambridge. 1996

[3] Mariano Fernández-López and Asunción Gómez-Pérez. 2002. Overview and analysis of methodologies for building ontologies. Knowl. Eng. Rev. 17, 2 (June 2002), pages 129-156.

- HF issues definition and evaluation
- HF methods description
- Regulations, standards and guidelines related to HF issues

Figure 4 shows the relationship between HF issues, HF methods and regulations.

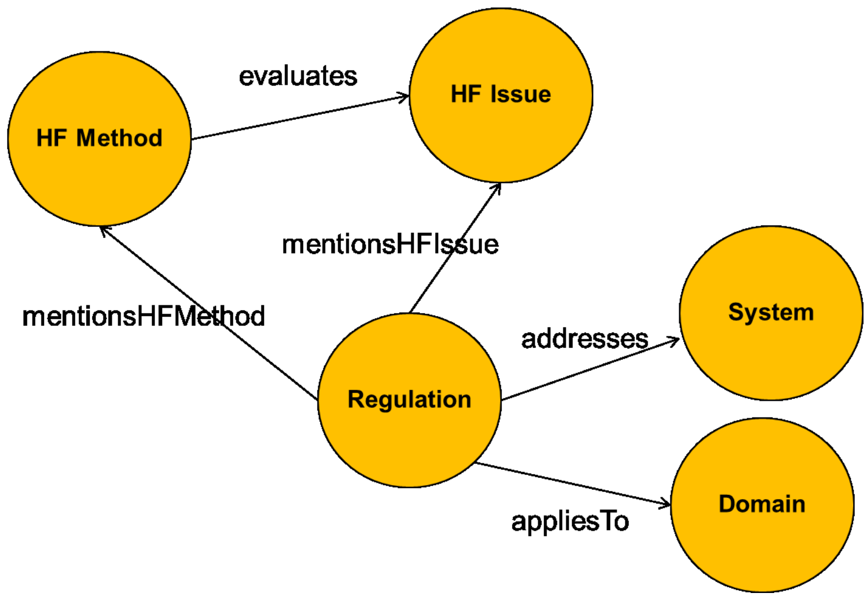

Figure 4: General schema of the HF ontology

The key concepts or main classes of the HF ontology are the following:

- Descriptor
- Domain
- HFIssue
- HFMethod
- HumanEfficiencyEvaluation
- Regulation
- System

2.3.3.1 HFMethod

Categorises or describes a HFMethod using a set of descriptors.

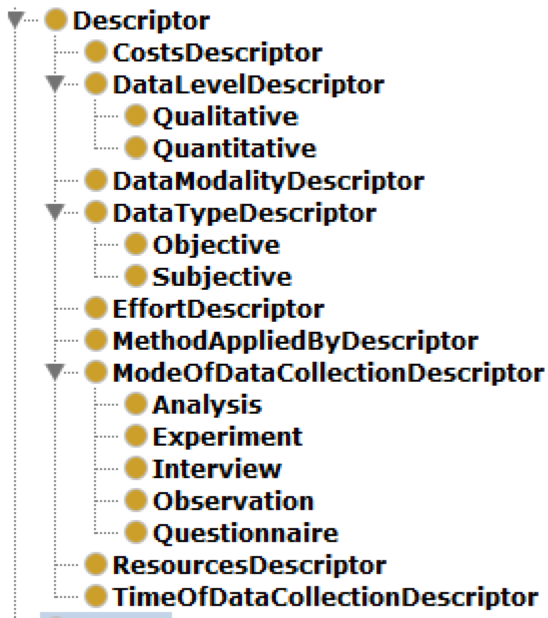

Figure 5: Descriptors for HF methods

The HFMethod concept has the following properties:

Property	Description	Range
terms:title	Identifier or name of the method	string
terms:abstract	Short summary of the method	string
appliedInPhase	Phase within the system development process in which the method is applied	VModelPhase ◆ certification ◆ conceptualisation ◆ deployment ◆ design ◆ evaluation ◆ RequirementsEngineering ◆ SystemImplementation
measurementSource	Type of data recorded: subjective vs objective	MeasurementSourceDescriptor • Subjective

		◆ **actor** ◆ **observer** • Objective ◆ **performance** ◆ **psychophysiological**
empiricalMethodType	Process used to collect the data	EmpiricalMethodTypeDescriptor ◆ **experiment** ◆ **expertInspection** ◆ **interview** ◆ **observation** ◆ **questionnaire**
timeOfDataCollection	Time of data collected	TimeOfDataCollectionDescriptor ◆ **prospective** ◆ **realtime** ◆ **retrospective**
methodApliedBy	Skills needed to apply the method	MethodApliedByDescriptor ◆ **domainExperts** ◆ **HFexpert** ◆ **HFnonexpert**
outcomeResult	Text description for specifying mandatory requirements for interpretation of results	string
effortTime	Time effort needed to apply the method	EffortDescriptor ◆ **highEffort** ◆ **lowEffort**
costs	Costs needed to apply the method	CostsDescriptor ◆ **highCosts** ◆ **lowCosts**
resources	Checklist defining the technical and human resources needed to apply the method	ResourcesDescriptor

		◆ **computer** ◆ **EEG** ◆ **experimentalLab** ◆ **eyeTracker** ◆ **paperAndPencil** ◆ **participants** ◆ **simulationEnvironment**
addressesHFProblem	Human factors problems addressed by this method	HFIssue

Table 5: Properties of the HFMethod concept

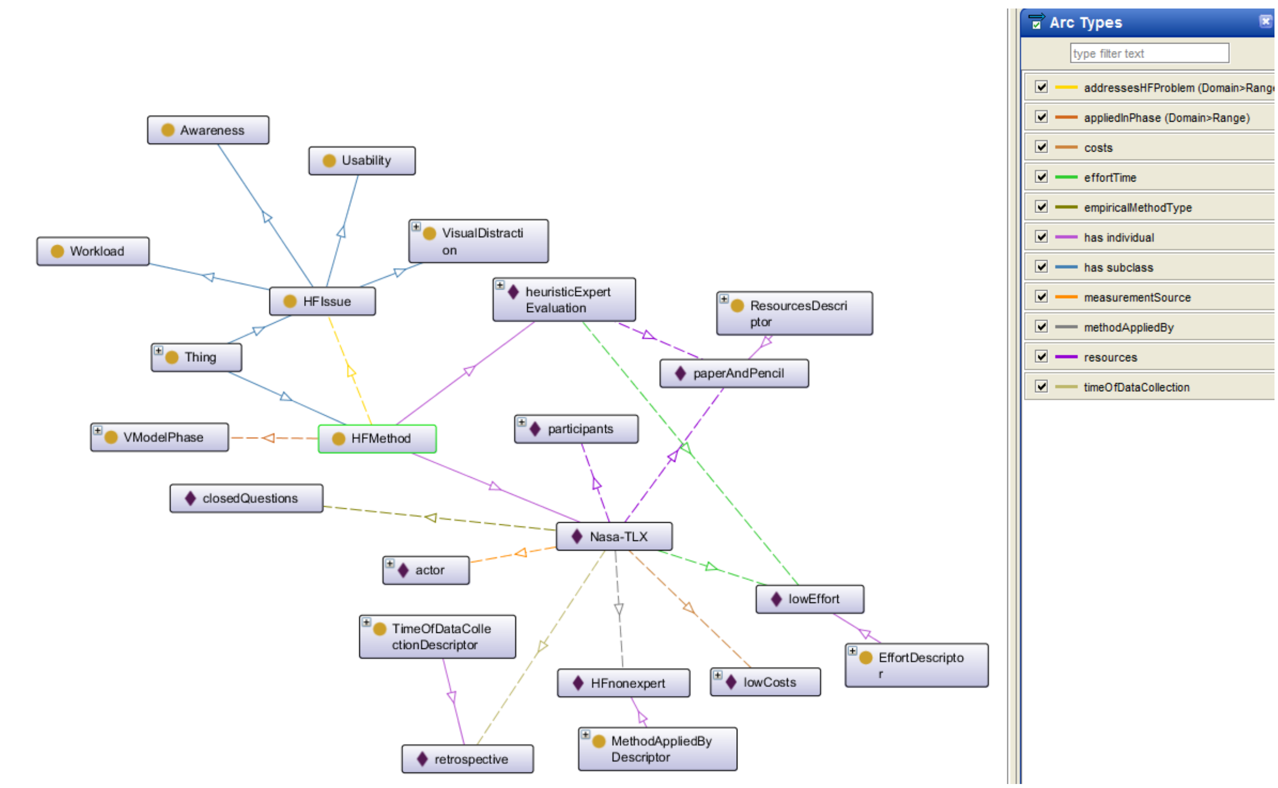

Figure 6: HF Method properties

2.3.3.2 HFIssue

This concept describes a Human Factor Issue and its different states. For the moment the following HF issues are considered: distraction, situation awareness, usability, workload, fatigue, cognitive capacity limits, trust in automation, technology acceptance, satisfaction, desire for control and perceived risk. Although not all HF issues are at the same level or category, the ontology considers them as a list as the purpose is to link each issue to relevant HF methods and regulations.

Figure 7: HF Issue class hierarchy

For each of these HF issues different states can be defined. E.g. in the case of fatigue, a two state classifier could be proposed: fatigued or not fatigued. In the case of visual distraction 3 possible states could be defined: fully distracted, partially distracted and not distracted. In addition, the property accuracy will allow providing a number that represents the accuracy of the classifier for the given state. This is a first approach to classification outcomes. However, in the next iteration of the ontology concepts to define the human operator models developed in WP2 will be included, among others: type of model (e.g. neural, mathematical, SVM, etc.), applicable domains or domain independent, type of predictions (qualitative vs quantitative), evaluated HF issues or output description. It is envisaged that this classification outcomes will be more detailed.

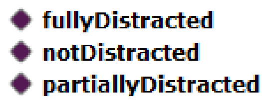

Figure 8: Visual distraction states

2.3.3.3 Regulation

This term represents a Human Factors Integration Concept or Regulation for any of the four HoliDes domains: aeronautics, control rooms, automotive and health care. These concepts and regulations were identified in *deliverable D1.2*.

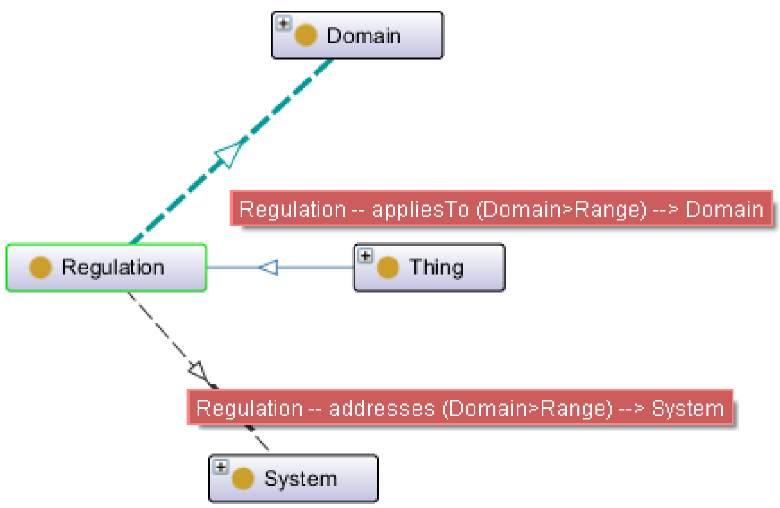

Figure 9: Human Factors Integration Concept or Regulation

The Regulation term has the following properties:

Property	Description	Range
appliesTo	The domain to which the HF integration concept or regulation applies	Domain: • aeronautics, • control rooms, • automotive • health care
addresses	The system addressed by the HF integration concept	System • ATM

	or regulation.	• EFB • Cockpit • Etc
mentionsHFIssue	The HF issues that are mentioned in the HF integration concept or regulation	HFIssue
mentionsHFMethod	The HF methods that are mentioned in the HF integration concept or regulation	HFMethod

Table 6: Properties of Regulation term

2.3.3.4 HumanEfficiencyEvaluation

This term measures the human efficiency. It has the following properties:

Property	Description	Range
attentionAllocationValue	The value that measures the attention allocation of the operator	measurement
operatorWorkloadValue	The value that measures the operator's workload	measurement
reactionTimesValue	The operator's reaction times of different HMI designs	measurement
taskPerformanceExecutionTime	The operator's task performance	measurement

Table 7: Properties of HumanEfficiencyEvaluation term

The class Measurement is related to:

- a value (i.e. quantity, integer, double, etc),
- a scale (optional), i.e. a minimum and maximum possible values
- measurement units (optional) from the Measurement Units Ontology (MUO) available at http://idi.fundacionctic.org/muo/muo-vocab.html

2.4 Tracking the RTP tailoring

2.4.1 The difference between tailored and non-tailored components

There is a subtle but important difference between a tool being classed as part of the RTP and tailored in the RTP. To say that a tool is part of the RTP represents a commitment to say that a tool should share resources in compliance with the HoliDes IOS. (I.e. Tools will share data using RDF XML which is compliant to a HoliDes Meta Model and made available through a Restful interface). In order to integrate a tool, i.e. to make it part of the HF-RTP, the HF-RTP integration methodology has been defined (See 2.4.4).

Note, an interface which conforms to REST is known to be “restful”. REST is an architectural style used in the development of web services. However, a full description of REST is beyond the scope of this document.

To say that a tool is tailored in the HoliDes RTP you are making a commitment to that two tools are exchanging information in accordance to the IOS to satisfy a Use Case. One tool is making its data available as RDF XML and is considered to be an OSLC provider. The second tool is asking for the data and is considered to be a consumer. In essence if a tool is tailored it ‘understands’ how to use that data. The methodology to tailor tools in the HoliDes RTP is defined in section 2.6.

This next chapter aims to make that clearer with an example.

2.4.2 Tailoring in a fictitious Example

Take a simple use case where with two tools, one is a drawing tool, e.g. Photoshop and the other tool is a word processor, e.g. MS Word. The design flow involves taking a picture from Photoshop and using it in a document.

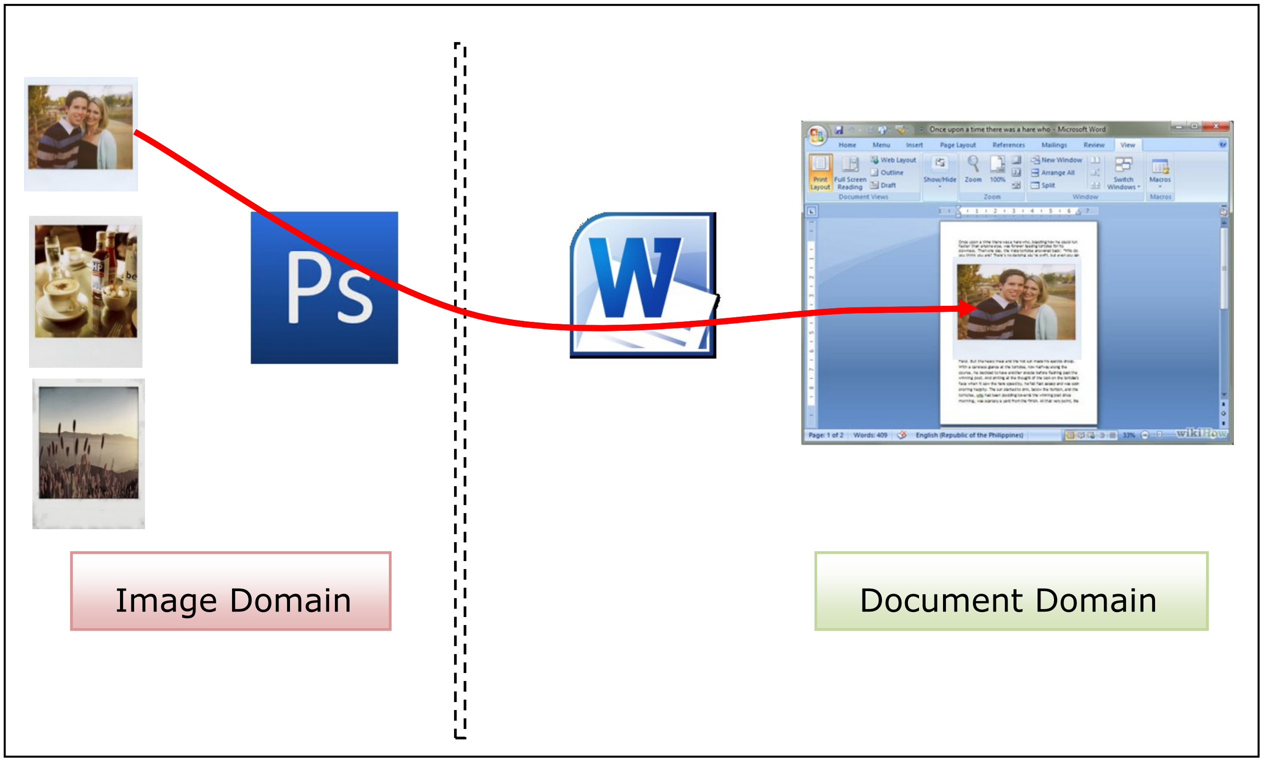

Figure 10: A fictitious Tailoring example where images are embedded into documents

One might copy and paste an image from Photoshop into the document in Microsoft word. Yes, you could argue that Photoshop and Word have worked together but there are drawbacks with this method.

I. **Duplicity**: The picture now exists in two separate places. Now you have all of the problems with data redundancy such as maintaining all copies of the image. If the image is modified, multiple documents would now need to be updated.

II. **Location:** The image needs to exist locally on the user's machine. Of course, the person creating images in Photoshop could email you a copy but then the problem of file formats appear. Do you take the jpg, bmp or a native Photoshop file? In Addition, you are back to the first problem of duplicity and redundancy.

III. **Scalability:** Let's assume that you now need to bring a 3rd document tool into the chain. There's a chance the copy and paste feature might not be available for the image format you want. A proprietary plug-in could be developed or at worst, you simply don't use that particular tool.

Let's now assuming we wish to improve the workflow and incorporate our two tools into the HF-RTP. First of all we need an ontology of what we want to exchange. For arguments sake, let's assume in HoliDes we have defined an image Meta Model. This Meta Model describes what an image should look like, its resolution, colour pallet etc.

The next stage is to modify Photoshop to make this image available in accordance to the HoliDes IOS. This means that some plug-in or modification is installed which allows the image to be accessed with RDF XML using HTTP. It's important to note that the image is still stored by Photoshop in the same way that it has always done. Just because Photoshop is now IOS compliant, this does not mean that all images in Photoshop are stored as RDF XML. What it means, is that there is now a representation of that image as RDF XML available for other tools to use.

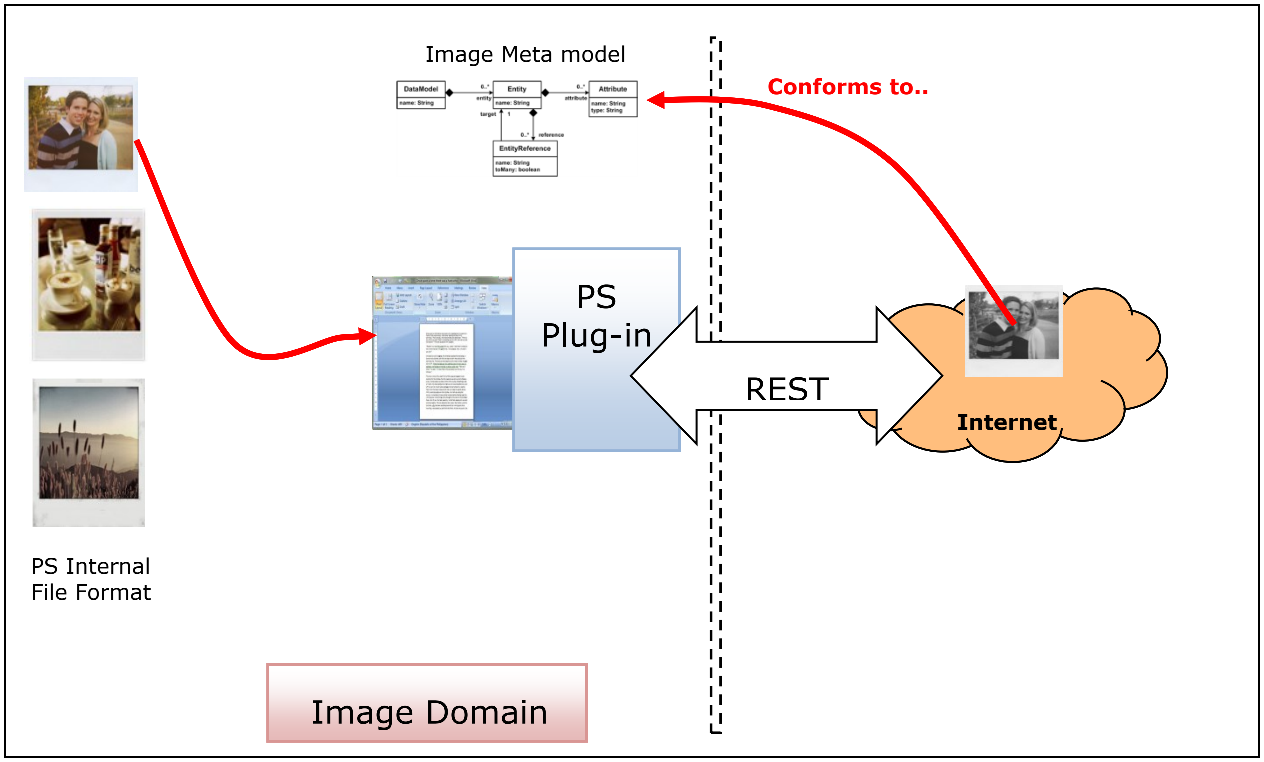

Figure 11: An Image tool which is now part of the RTP but no yet tailored

Now that this adapter is installed, Photoshop (or its adapter) now has a web presence. That means it is now possible to visit Photoshop in a web browser or any other application that works with HTTP. Using HTTP, one can browse for the image they wish to use. At this point you could consider Photoshop as part of the HF RTP but not yet tailored as nothing consumes its data.

Now, in this example, the consumer application will be MS Word. Let's assume that word is already set up with a plug-in which will allow it to act as OSLC consumer of Human Factors images. By the term "Human Factors Images" it is meant that the images it can work with are images that conform to the fictitious 'image Meta Model'.

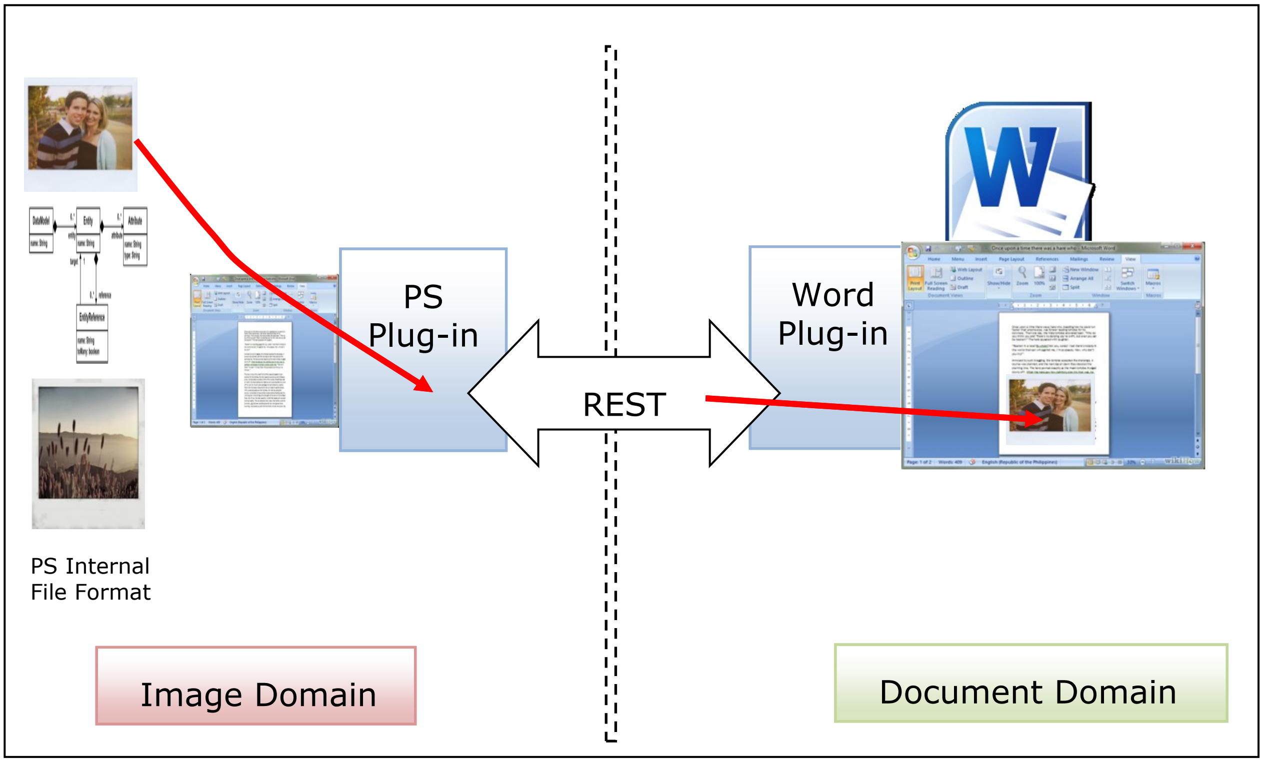

Figure 12: Use of OSLC plugin

The user of word is now able to start the OSLC plugin and send a request to the Photoshop URL. The Photoshop service provider responds by sending back a list of all the available images. The list which is sent back contains the name, description and URL of each of the available images. The word OSLC consumer then sends another request but this time it's for a particular image URL.

The Photoshop plug-in once again responds by sending via HTTP all of the information it has on that particular image. Of course what is wanted is the actual image which the consumer now has the URI.

The following figure shows how an exchange of data might look like:

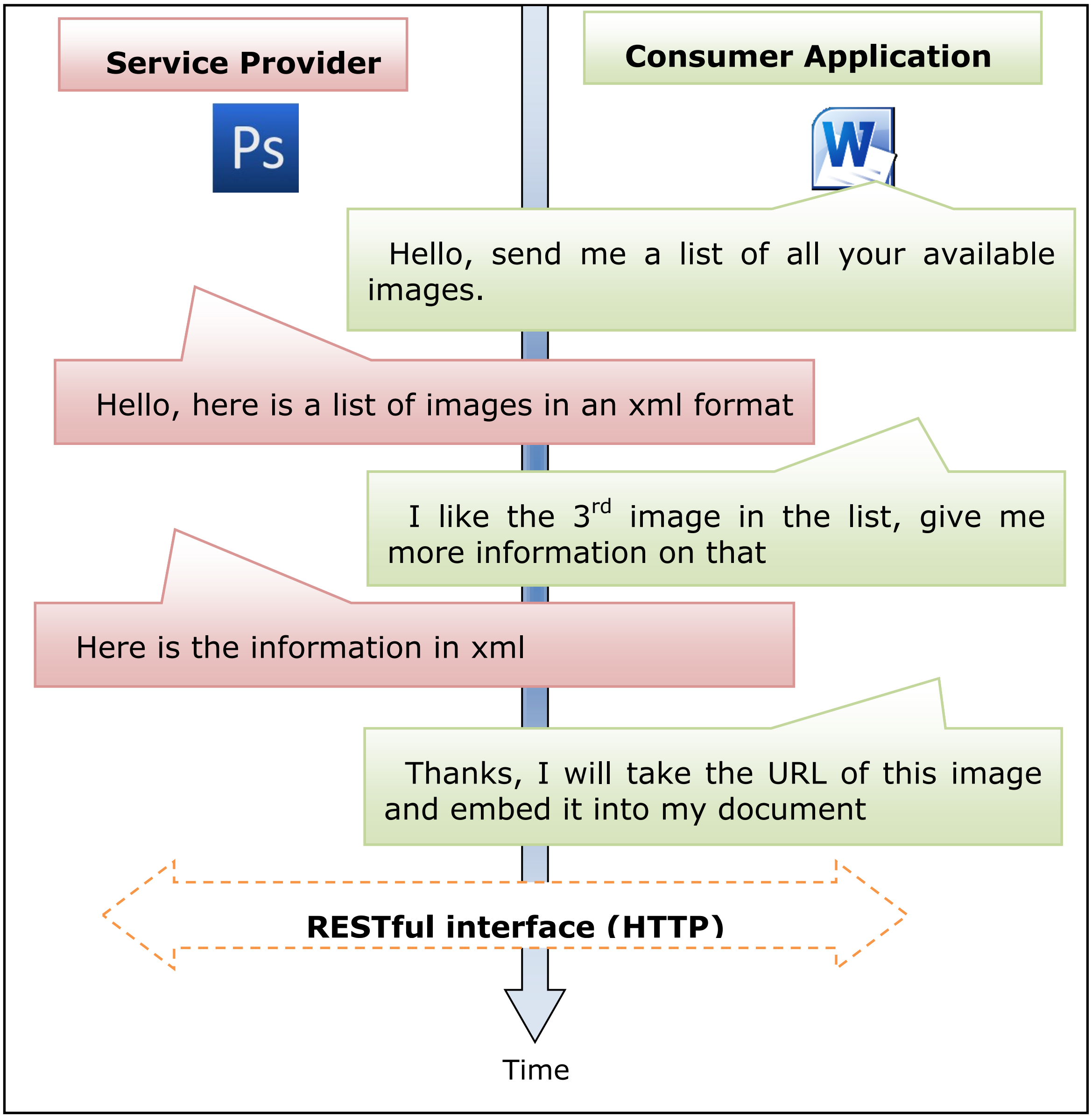

Figure 13: A typical RTP exchange

Note *It depends on the particular scenario how this information is used. It could be that a copy of the data is kept and MS word works with a copy but this is problematic for the reasons of duplicity already discussed. What would make better sense is if MS Word stores only a link to the image.*

The image URI is now embedded into the document. From now one, whenever that document is opened, MS Word will perform an HTTP GET on that stored URI and get the latest image. Thus, if the image is updated the changes would be reflected in the document. Now that these two tools are working together we can consider them tailored. By using the same principles and protocols which made the internet so successful we strive for similar levels of interoperability in applying Human Factors to Systems Engineering. For example, suppose that Photoshop is no longer wanted and MS Paint is now preferred for creating document images because it's a cheaper tool. So long as MS Paint can make its resources available over a rest-ful format with RDF that conforms to the HF Image Meta Model, there should be little or no effort in moving from Photoshop to Paint in your design process.

Further Tailoring

The benefits of the IOS are made clear with an example in which new domains are added. Let's expand the work flow to include a fictitious analysis tool. The purpose of this fictitious analysis tool is to track the features used in Photoshop. For this, it needs to know about the numbers of times the user of Photoshop selects a certain feature; it doesn't need to know anything about images at all.

A new Photoshop plug-in is created which adds another service to its existing IOS adapter. The beauty of this method is that the delivery mechanism already exist i.e. the Restful interface. As the existing restful interface can be reused the development effort is significantly reduced. Thus, a tool which was already part of the HoliDes RTP has been tailored now to work with two tools.

RTP Building

When someone wishes to leverage the power of the HoliDes RTP they will create an RTP instance and to assist them with that they will use the HoliDes RTP builder. When someone does this, they will have two questions to ask to the platform builder:

I. Which Methods and Tools are available to me?
II. Which of those Methods and Tools are integrated together?

The platform builder will in essence be a collection of tools and tailorings. Throughout HoliDes this list of tools and tailoring will grow and so will need to be managed for the duration of the project.

2.4.3 The RTP Integration Plan

The RTP Integration plan is a spreadsheet which will be a live document for the duration of HoliDes. The spreadsheet was started by Work Package 2 to 5 and has been adopted by Work Package 1 since integrating tools is at the heart of the RTP. The idea behind it is to keep a log of the current list of tools and their tailored connections to other tools in the HF-RTP.

Methods, Tools and Technologies

Typically, an RTP is made up of engineering tools but in HoliDes, MTTs exist as documentation. Where MTTs exist as only documentation some other software could be used to give the resources a web presence. For example, the linker tool or any database with an OSLC interface.

Entry Name	Entry Description
MTT ID	A unique ID number
Tool Name	The name of the Tool
Description	A brief sentence describing what the tool does
OSLC / IOS	A yes/no answer denoting OSLC compliance.

Compliant *	
3rd Party Tool (y/n)	A yes indicates that the tool provider is an active partner in the HoliDes Consortium.
Use Case ID	A list of all of the use cases in which this tool is used.
Position on the V model	A phase name indicating where on the V model this tool should be placed.
MTT is or has dedicated software to run it. (y/n)	A **yes** denotes that the MTT is something that can be installed and used straight away. A **no** denotes that no specific software exists to use it and the results likely exist as documentation.

Table 8: Description table for Methods, Tools and Technologies

MTT List with which to build the HF RTP

The following table lists all of the MTTS which have been made available in HoliDes from Work Packages 2 to 5. Note, techniques have been listed here as well tools. A technique can't exists in an RTP by itself since an RTP component must be IOS compliant and a technique can't collaborate with other tools without a software mechanism to work with. For partners to make use of the technique MTTS, they must also choose a software application to accompany it.

This list also contains tools which are not provided by HoliDes partners but are required for some use cases. These are listed as external to HoliDes.

Table 9: The MTT list with which to build the HF-RTP

WP 2			
CASCaS	Simulation	Human Behaviour Model	OFF / Lars Weber
COSMODRIVE	Simulation	Human Driver Model	IFS / Thierry Bellet
djnn	Tool	UI / Interaction Model	ENA / Mathieux Magneudet

Driver distraction model	Simulation	Human Distraction Model	TWT/ Denis Martin
Driver distraction classifier	Tool	Visual Distraction Classifier	UTO / Marco Botta
GreatSPN - MDP part	Tool	Petri Net	UTO, Susanna Donatelli
Human Efficiency Evaluator	Tool	Task Model, Human Behaviour Model (CASCaS)	OFF / Sebastian Feuerstack
MagicPED	Tool	Task Model	OFF / Jan-Patrick Osterloh
Training Manager	Tool	Task Model, Training Model	OFF / Jan-Patrick Osterloh
Pilot Pattern Classificator	Technique	Human Behaviour Model	TEC/Sara Sillaurren
Bad MoB	Simulation	Human Behavior Model	OFF / Mark Eilers
WP 3			
CBR: Case Based reasoning	Technique	return of experience	EAD-F / Nicolas Schneider
Hgraph: hierarchical graph modeling	Technique	Task Model	EAD-F / Nicolas Schneider
Hgraph: hierarchical graph modeling	Technique	Adaptive analysis	EAD-F / Richard LeBlond
LEA: Learning classifier system	Tools	Learning behaviors	EAD-F / Nicolas Schneider
APA	Tools	Sequential patterns assesment	EAD-F / Nicolas Schneider
Uppaal	Tools	Time automata	UTO, Susanna Donatelli
Movida	Tool	Cosmodrive / Monitoring	IFFSTAR / Thierry Bellet
AMAS: Advanced multi-agent system	Technique	multi-agents	EAD-F
CONFORM	Tool	Behaviour Model	DLR / Stefan Griesche

Driver intention recognition	Tool	Behaviour Model	OFFIS/ Mark Eilers
Operator Pattern Classifier	Tool	Behaviour Model	TEC/Sara Sillaurren
HF-Guidline	Technique		EAD-IW-DE / Martina Becker
WP 4			
RT-MAPS	Tool	Base for HF-RTP	INT / N. Dulac
Anaconda	Tool	Dynanmic analysis of C++ programs	BUT / B. Krena
Search Bestie	Tool	Support for Testing	BUT / B. Krena
Data Race Detector & Healer	Tool	Data race detection for Java programs	BUT / B. Krena
Pilot Monitoring	Tool?		BUT / Adam Herout
I-DEEP	Tool	Observation and playback	INT / N. Dulac
COSMO-SIVIC	Tool	Human operatormodel & simulation	IFS / T. Bellet
GreatSPN	Tool	Petri Net	UTO Susanna Donatelli
Experiment Database Management	Tool		TEC/Sara Sillaurren
CaSACaS	Simulation	Human Behavior Model	OFF / Jan-Patrick Osterloh
djnn	tool	UI / Interaction Model	ENA / Mathieu Magnaudet
PROSIVIC	Tool	Virtual car and 3D road environment simulator	CVT / K. Kayvantash
WP 5			
HF Filer	Tool	OSLC data / filings of human factors evaluations	AWI / Morten Larsen
Modelling of AdCoS data from a means-ends perspective	Technique	observations / task model	AWI / Morten Larsen

Detection of operators' head orientation	Tool	videos of human operators' heads / description of operators' heads orientation	BUT, HON / Adam Herout
Operator state detection from implicit hand gestures	Tool	whole-body videos / description of operator gestures	BUT, HON / Adam Herout
CPM-GOMS task analysis of a Lane Change for manual and automated driving	Technique	video data, driving data, verbal data, eye tracking data / detailed description of a driver's cognitive, perceptual and motor activities	DLR / David Käthner
Methods and techniques for the driver adaptive parameterization of a highly automated driving system	Technique	manual driving data / preference for automation driving style	DLR / David Käthner
Theater Technique for acceptance tests during AdCoS design	Technique	interaction concept for the overtaking manoeuvre / qualitative data from exploration	DLR / David Käthner
Tests for cognitive task models	Technique	empirical data from task analysis / MTTs for testing predictions of task models	HFC / Dora Gardas
MEBEF (Mean number of Examinations Between Examination Failure)	Tool	systematic log-file analysis and reporting on routine system use in the installed base	PHI / Paul Kaufholz
Behavioural Validation Tool	Tool	XML representation of finite-state machine / spreadsheet with report	REL / Elisa Landini
Empirical analysis of cognitive and communication processes	Technique	empirical data	SNV / Simona Collina
Detection of driver distraction based on in-car measures	Tool	driving data, audio data, video data / measure of distraction	TWT / Svenja Borchers
Surrogate Reaction Task	Tool	empirical data	DLR / David Käthner
Detection of driver distraction based on data on vehicle dynamics	Tool	driving data / qualitative description of driver's degree of inattention	UTO / Marco Botta
MTTs External to HoliDes			
Doors Next Generation	Tool	Requirements managing tool.	Airbus, Control Room.
Enterprise Architect	Tool	Architectural modelling tool.	Airbus Control Room.

MTT Integration

The tables in the Annex III MTT Integration Plan have been built up from the integrations proposed by the application use cases from the AdCoS in WP 6 to 9. It is not a static list, but a living document since it will be changed constantly throughout the project. Each row represents a pair of tools which will be connected by the end of HoliDes are listed here. The 'type' column indicates whether the tool is a provider or consumer of resources. The input data column describes the data which is given by the provider application and the output data describes what is made available with the input data with the capability of the consumer application.

Platform builder and the Integration Plan

The tables in the Annex III MTT Integration Plan will be used in the future as input into the Platform Builder. The Platform Builder will depicts the connections shown here as possibilities for users wishing to developing their tool environment.

2.4.4 HF-RTP integration methodology

The integration of MTTs into the HF-RTP will be based on a detailed plan and methodology. In order to define it, it is important to distinguish between Methods/Techniques and Tools.

The methods and techniques that have been developed in the HoliDes project have been delivered in the form of innovative algorithms, procedures and guidelines for the different phases within the AdCoS development process in the different domains.

Most of the methods and techniques are not delivered as a piece of software. Therefore, they cannot be actually integrated into the HF-RTP and no interoperability with the other tools can be achieved. However, the outcomes of these methods and techniques will be gathered in documents that will be inputs to other MTTs.

Therefore, the integration of the methods and techniques (including guidelines and design of empirical experiments) will be integrated by uploading the documents into MTTs which are part of the HF-RTP. An example of such a tool is the HF Filer tool.

Moreover, they will be easily accessible by using the Platform Builder developed in WP1 that allows finding the most suitable method and technique according to the specific need they are meant to address.

We are aware that this process cannot be considered a full integration as defined in CESAR project, regarded as a reference for the development of HF-RTP in HoliDes.

The main difference is that in CESAR each tool was mapped on the development process (represented with the V-model). V-model and adapters have been developed to make these tools fully interoperable.

However, as stated in the DOW, HoliDes aims to go beyond the RTP developed in CESAR.

In particular, the HF-RTP will extent the CESAR RTP with human factors engineering activities and will integrate the HoliDes techniques and tools by defining RTP workflows and processes, a Common Meta Model and an interoperability standard tailored on the needs of the Human Factors.

What we claim is that the CESAR roadmap has lacked appropriate means for human-centred development of AdCoS. Therefore, the coexistence of methods/techniques and tools is implicitly due to the nature of HoliDes, since the HF-RTP will enable development and qualification of AdCoS from both human factors and technical perspectives, in a holistic way, where methods and techniques represent an explicit and usually informal and unstructured part of the methodology for the design, analysis and evaluation of adaptation in the human-machine cooperation, thus a key element of the engineering process for the development and qualification of an AdCoS.

As regards the tools, two different categories of software have been developed (as shown in Figure 14):

- The tools that are meant to be integrated into the tool chains of the AdCoS owners to improve their development process and deal with lifecycle data such as requirements, test cases, analysis results and behavioural models (e.g. Magic-PED of OFF for the task modelling)
- The tools that have been developed to be embedded into the AdCoS to improve its functionalities (e.g. the Driver Distraction Classifier of TWT).

In order to avoid any confusion between them, from now on the latter will identified as “modules” instead of “tools”.

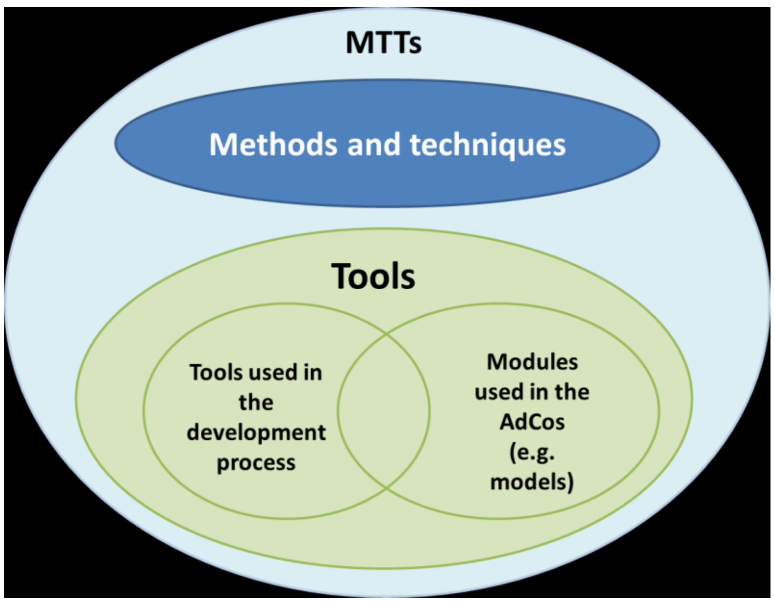

Figure 14: MTTs – methods, techniques and tools

Moreover, some of them present a double nature: according to the context of use, they can be either used in the development process or “as bricks” of an AdCoS (e.g. a module for the detection of the distraction can be included in the AdCoS to improve its functionalities and it can be employed as a tool for the evaluation of an AdCoS to assess if it induces distraction in the operator).

For the tools, the integration implies the development of specific piece of software (i.e. adapters) to implement the OSLC specification and allow the sharing of data with other tools.

A similar approach can be also applied to the modules: they do not deal with the lifecycle data management but they share data with other pieces of software within the AdCoS. In particular, most of these modules share real-time data with RTMaps for the implementation of the AdCoS.

Exchanges of data which take place at design time will be done with using the OSLC principle. Exchanges which take place at run time will not be done using OSLC adapters. With MTTs which deal in both design and run time there will inevitably be some cross over but the distinction can always be made. For example, RTMAPs can be used at both design and run time.

During design time, the engineer building the model would do so in accordance with systems engineering principles and thus the design will have a lifecycle and be linked to other life cycle artefacts. This is where OSLC would be useful. For example, the requirements defined in another tool would impact the design model in RTMAPS in some way and this traceability needs to be maintained for stakeholder validation.

When the model is completed in RTMAPS it generates the code which controls sensors and actuators. The data that flows around the physical system does so in run time and as such the exchange is beyond the scope of OSLC.

It will allow reusing the same module within different AdCoS (i.e. to make the modules interoperable across different AdCoS and domains).

The overall methodology for the tools and modules foresees the following activities to be integrated into the HF-RTP:

1) Identify their inputs and outputs, related MTTs or AdCoS. Moreover, only for tool, identify their compliance with OSLC and the estimated date for integration into HF-RTP.
2) Identify the correspondence between its inputs and outputs and the concepts of the HoliDes Meta Model.

3) Only for tools: develop the adapters or parsers between the concepts and the data managed by the MTT (parse numerical values and strings into RDF and from RDF to numerical values and strings).

The next steps will be to tailor the tool/module as defined in section 2.6 according to its use as part of the development process and/or as part of an AdCoS. For tools different adapters may be developed according to the context (i.e. the other software is it expected to share data with) and the different information requested.

For the tools, the methodology will be similar to the tailoring steps defined in D1.4 and applied in D6.4, D7.4, D8.4 and D9.4 to tailor the HF-RTP to the needs of the different demonstrators.

The next section 2.5 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.

2.5 HF-RTP Infrastructure & Interoperability Standard

To define the real interoperability we will follow these three steps for each use case:

1. For the specific AdCoS of each Use Case, there should be specified the type of OSLC adapters used currently (if not used, justification for not using them)
2. Different OSLC domains used in each use case (core, requirements, architecture management, automation, monitoring, quality …)
3. Define the specifications for OSLC Human Factors domain

TEC together with ATO has had a series of telcos with the different Use Cases leaders and participants, in order to define which type of data is exchanged among the different MTTs. The idea is analyse if OSLC standard fits well in the different MTTs of each AdCoS and if not, explain which the reason to adopt a different standard.

As stated in previous *deliverable D1.4*, HoliDes will build upon the approaches of the CESAR and MBAT projects, with the OSLC standard as basis, but defining the concepts specific for Human Factors, which do not exist at the moment.

A working group[4] will be proposed to the OSLC consortium with the results of HoliDes. Currently we are working on the charter description to be submitted.

Besides, HoliDes is in contact with the CP-SETIS project consortium. This project has two main goals:

- The alignment of all IOS-related forces within Europe to support a common IOS Standardisation Strategy, aiming at a formal standardisation process of the IOS
- The definition and implementation of sustainable IOS Standardisation Activities supporting both, formal standardisation of “stable” IOS

[4] http://open-services.net/participate/creating-new-workgroups/

versions as well as extensions of IOS, if possible within existing structures that survive the lifespan of single projects.

HoliDes IOS is based on RDF/XML and REST (Representational state transfer) technologies. This means that tools will share data which is compliant to the HF ontology defined in section 2.3.3 and made available through a Restful interface.

Adapters consist of Service Consumers and Providers. "Consumers" are clients consuming services whereas "Providers" are servers providing services following a client-server architecture.

2.6 HF-RTP Tailoring Rules & Deployment

The development of tailoring rules is part of the task 1.8. Their application in the AdCoS development is performed in the task 6.1, 7.1, 8.1 and 9.1 for the different application domains. The most recent tailoring status has been reported in the *deliverables D6.4, D7.4, D8.4* and *D9.4* and will be summarized in section 2.6.3.

2.6.1 Generic tailoring steps

The basic definition of tailoring has been outlined in *D1.3*. Tailoring is about selecting tools and services and integrating them by providing suitable ways to exchange data between them. The result of a tailoring process is thus an instance of the HF-RTP addressing the specific needs of the project.
However the integration of Human Factors into the development processes, the requirements related to adaptivity, and the HF-RTP supporting them show a need for more adapted and more specific description of the tailoring steps. In *D1.4* the tailoring process has been described in four generic steps:

1. Identification of the purpose of the project and the used tool chain
2. Selection of methods and tools
3. Definition of semantics and information mapping between methods and tools
4. Implementation of information models and connectors.

By actually tailoring the HF-RTP on the real needs of the AdCoS owners in the WP6 to 9, and through discussion with the partners in WP1 the steps have been slightly rephrased. The new version of the methodology, as a result of the collaboration of the AdCoS owners (WP6-WP9) with the partners in WP1, includes the following steps:

1. Identification of issues in the existing development process
2. Selection of methods and tools (MTTs)
3. Integration and interfaces
4. Implementation of information models and connectors.

2.6.1.1 Tailoring Step 1: Development process and issues

The first step is to provide a description of the overall purpose of the HF-RTP instance within the development process to be supported. This describes the specific workflow of the development process of an AdCoS, and describes the issue faced by the AdCoS owners in the development of the adaptive system, mainly in terms of Human Factors and adaptivity. Such issues might include a lack of traceability of engineering data or issues using information from one tool in another.

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

2.6.1.2 Tailoring Step 2: Selection of methods and tools

In the second step, related methods and tools are selected to fit the needs of the project purpose. A detailed rationale is given for each of the selected MTTs which will be included in the workflow.

A library of generalized descriptions of methods supports the methods selection (methods library described in section 2.3.2). A HoliDes tool list supports the tool selection and will be integrated into the platform builder (described in section 2.7)

The definition of step 2 in *D1.4* also covers some issues about OSLC compliance of selected tools, which can be achieved for some tools during the project, but is not available yet.

2.6.1.3 Tailoring Step 3: Integration and Interfaces

The next step is to describe the information exchange needed between the methods, techniques and tools in the actual tool chain. A common semantic approach allows the generalization of the information exchange across several methods and tools. 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.

In particular, this mapping step describes the interfaces that define which and how information is exchanged between MTTs and existing tools.

2.6.1.4 Tailoring Step 4: Implementation

Step 4 is the implementation of the mappings defined in step 3 (i.e. implementation of parsers to allow the tools to correctly interpret the information receive according to a predefined communication protocol).

The actual implementation of the communication protocols and parsers for the sharing of data among tools and with the RTP relies on the definition of the information to be shared.

2.6.2 HF-RTP Components supporting the tailoring

A number of HF-RTP-components have been planned and/or developed to support the tailoring process:

- A list of tools is available within HoliDes for all partners. It includes standard of the shelf tools used in the AdCoS development processes as well as tools developed or under development by the partners.
- A human factors method library is to collect different methods that can be applied to HF issues like usability, situational awareness, workload and user distraction and to categorise them in a standardised manner. A set of standardised descriptors for the different methods allows to overcome the low level of formalisation of many of these methods and to make them available for the tailoring process via the platform builder described below. The details of the Human factors methods library are being described in section 2.3.3.
- The HF ontology is part of the common Meta Model. To allow data exchange between different methods and tools, the common semantics need to be described and synchronized. The HF ontology defines the common semantics related to Human Factors. It includes the descriptors used in the method library as well as descriptors for Human Factors issues and regulations related to them in the different application domains.

- The Platform Builder aims at providing a software application to support the configuration and instantiation of the HoliDes-RTP. It gives access to the HF-RTP components, libraries and models and supports the choice of tools and methods for a specific project. The details of the Platform Builder are being described in section 2.7.

2.6.3 Effective tailoring in the AdCoS work packages

In WP6 the status of the tailoring has been described in *D6.4*. Each AdCoS of WP6 has progressed in identifying Methods, Techniques and Tools (MTT's) that will help to improve their specific development process and product quality. Most AdCoS are working on the specification of information exchange (semantics) between MTTs (step 3 of the tailoring). The work on the actual implementation will be done in a next iteration.

One of the challenges in WP6 is to design optimal user interfaces and product behaviour for a large variance of user roles, skills and ways of performing the tasks. An example of adaptivity is to play with level of user guidance, target information to the right users concerned with the related task and remember repetitive actions/preferences.

In WP7 the tailoring is reported in *D7.4*. For the two AdCoS under development in WP7 (Airport Diversion Assistant DiVa, Enhanced adaptive transition training EATT) the tailoring has been improved and appended based on the previous version in *D7.2*.

For DivA, the specific issues within the development processes are being described (step 1). The choice of used methods and tools has been appended, adding a new tool for experimental tagging and annotation and a tool for fatigue modelling and pilot state classification in DivA. The central tool in this scenario is RTMaps, which is used to integrate data from different sources including the other used tools. The data flows are described based on the integration via RTMaps (Step3) in order to be able to meet the time constraints real time data transmission from sensors. Some of the adaptors are already implemented (Step 4).

For EATT, the workflows and related issues are identified (step1) and two tools added, MagicDraw and Training Manager (step 2) together with the expected data flows. The connections between the tools have been defined (step3) and need to be implemented.

As for WP8, the deployment of the tailoring is being reported in D8.4. For both application cases (Border security, Energy network control room) the workflows of the development process have been described and associated issues are identified (step 1 of the tailoring).

In the energy network control room scenario, the information flows between the different steps have been described as well. In step 2 of the tailoring, the MTTs that have the potential to address the issues have been identified and selected to be tested. No adequate tool has been found to address all issues raised on requirements and this issue remains unsolved for now. In the command and control room scenario, the requirements are managed with DOORS which is already compliant with OSLC.

Finally, the outputs (and the corresponding format) for each existing tool have been identified in order to understand how to interface them with the MTTs and the HF-RTP (Step 3). However even though the tools have been identified, the tool chains are not yet RTP compliant as they don't all yet exchange data using linked data which conforms to the Meta Models as defined by the HoliDes project.

The status of the tailoring deployment in WP9 is reported in *D9.4*. For each of the AdCoS, the choice of methods and tools has been made (step 2 of the tailoring) and the tools have been tested as to whether they meet the needs of the development process. The integration of the tools (step 3 of the tailoring) has been defined for some but not all of the AdCoS development processes. At this stage of the project, not all MTTs planned to be used in the development process share resources in compliance with the HoliDes IOS. Therefore the integration in accordance to the IOS needs to be carried out in later versions of the tailoring. Step 4 of the tailoring will be carried out in later versions of the tailoring.

As a general conclusion within this WP the HF-RTP provides already a good overview of available tools for developing an AdCoS, although it is in an early stage. The tailoring backed by the basic architecture of the HF-RTP solves some issues in the development and design phases. However other issues still remain unsolved. For some issues, no suitable MTTs are available, for other issues, MTTs are available, but are not selected, because they do not show a clear benefit in the design process. The ability to improve the development process in terms of more efficiency is the strongest requirement for the choice of MTTs.

Moreover there is a need for integration of more empirical methods for analysis and evaluation. The integration of the HF methods library in the RTP should allow valuable support here.

A special situation for WP9 is that the results of the intended development are prototypes rather than products. For this reason, the requirements on the HF-RTP may differ from other work packages. Standard office tools are used for most of the development steps. In contrast, product development includes validation and certification which adds the need for powerful requirement tracking tools.

2.6.4 Conclusions and next steps

Based on discussions with partners from WP1, the tailoring rules have been further fitted to the real needs of the AdCoS owners. This progress has proven successful as the redefined tailoring steps could be easily applied by the AdCoS owners (at least partially, according to the status of system development). Thereby, the system development process is getting more structured, reasonable and transparent.

According to the system development phase AdCoS developers are currently working on, tailoring step 1 has been successfully accomplished. Tailoring step 2 has been completed in a first version but will be updated on a regular basis. Step 3 at least has been started in most cases. Step 4 has not been started for most AdCoS.

Tailoring step 1 requires a clear definition of issues that have to be faced, particularly with regard to human factors and adaptivity. By explicitly stating issues that are going to be addressed, further steps can be derived in a reasonable way.

Step 2 covers the selection of tools. This step is work in progress as the tool data base has not been completed yet. The collection of tools represents a very laborious task. Tools do not only have to be collected, but have to be categorized in a standardized way to make them comparable and therefore reasonably selectable (Why is tool A selected and not tool B?). In this regard, the latest efforts were made for MTTs dealing with human factors analysis and evaluation. Results are available in form of a Human Factors method library and will be deployed in the next version of the tailoring.

Step 3 describes the information exchange needed between the methods, techniques and tools in the actual tool chain and step 4 will address the implementation of the mappings defined in step 3.

To ease the tailoring process, it should be supported by the platform builder which is being implemented and reported in this deliverable. The first instance of the platform builder will mainly address step 2 of the tailoring and support the selection of methods and tools. Later versions of the platform builder together with the RTP-tracker should also support step 3 by allowing graphical representation of needed information flows between methods and tools, and step 4 by keeping track of already implemented interfaces and connectors and thus providing them for other projects.

2.7 Platform Builder

The documentation of the Platform Builder will be provided in the attached documents in order not to overload the deliverable. Here we will provide the most transversal and generalist information about the solution.

This Platform Builder is part of the work done so far in the development of components and application of their technical integration and seamless interoperability achievement, referred to Task 1.7 about definition and implementation of the HF-RTP infrastructure & interoperability standard and generic services.

2.7.1 Platform Builder overview

The Platform Builder is a solution for improving the configuration and instantiation capabilities of the HoliDes HF-RTP. Based on a HF problem description, the Platform Builder aims to set up an instantiation of the HF-RTP for a specific domain project.

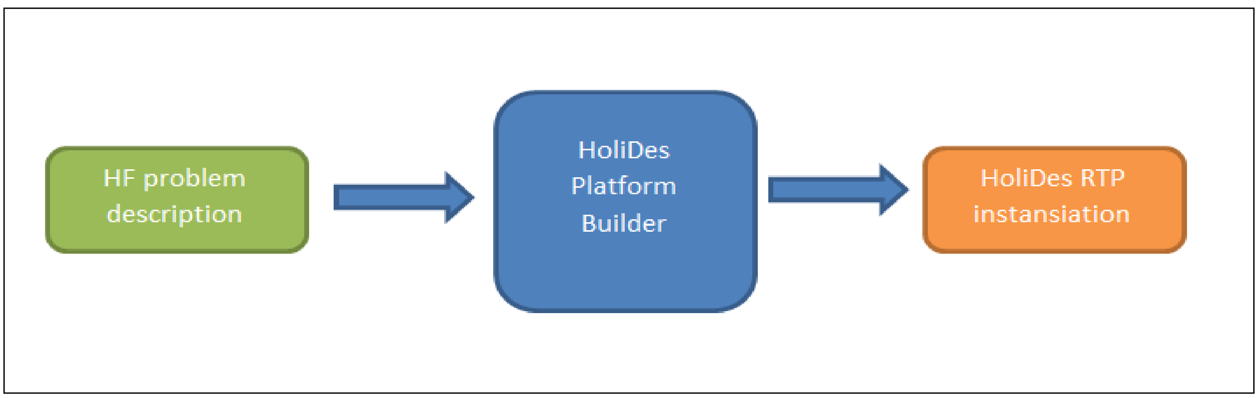

Figure 15: Platform Builder conceptual Design

For the accomplishment of the objectives and goals of the Platform Builder, a web application will be developed using a three-tier architecture (detailed in the sub-section Platform Builder functional analysis document).
Based on HF problems descriptions have given by HF partners (of all domains) to propose the correct MTT tools existing in HoliDes Project.

The idea is entering in the Platform Builder application a specific domain and other criteria (linked by domain as HF Issues, Extra Requirements and Related activity) to propose specifics MTTs to help the Users of HF to choose a decision "what is the better Tool for my HF issue".

2.7.2 Platform Builder functional analysis

There are two documents attached to this section, functional analysis document and requirements document.

In the Functional Analysis document "Platform Builder functional analysis" has included the architecture & technical structure and the Graphical User Interfaces with a mock-up for each screen, also the database model entity-relation with a description for each entity and the relationship between them.

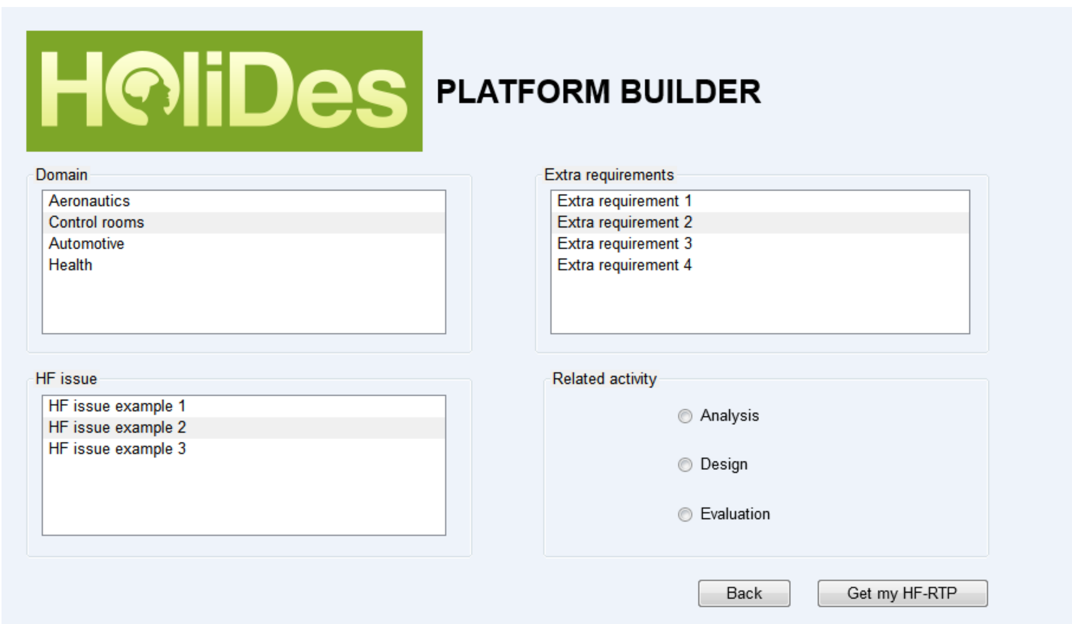

Figure 16: Mock up for PB Input screen

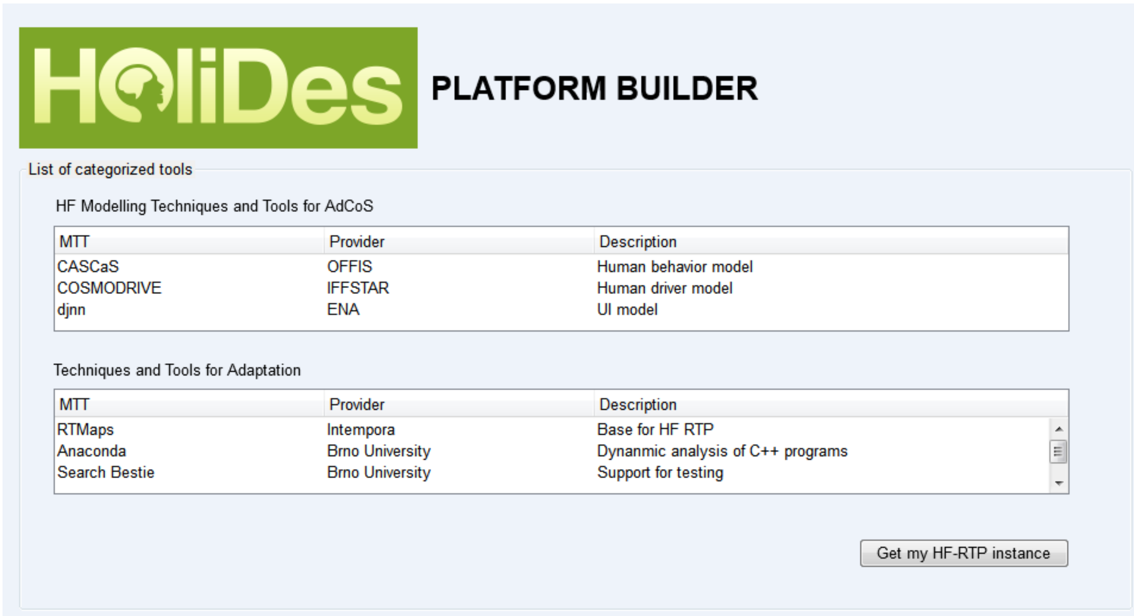

Figure 17: Mock Up for PB Instance HF-RTP, List of Categorized tools

Database Model Database Entity-Relationship Model (ER):

Contains the entities and the relationship between these ones storing all data interesting for resolve the HF Issues proposed. All MTT included in HoliDes should be stored in this database, the advantages and disadvantages for each MTTs in order to allow to the system to propose the correct "List of categorized Tools".

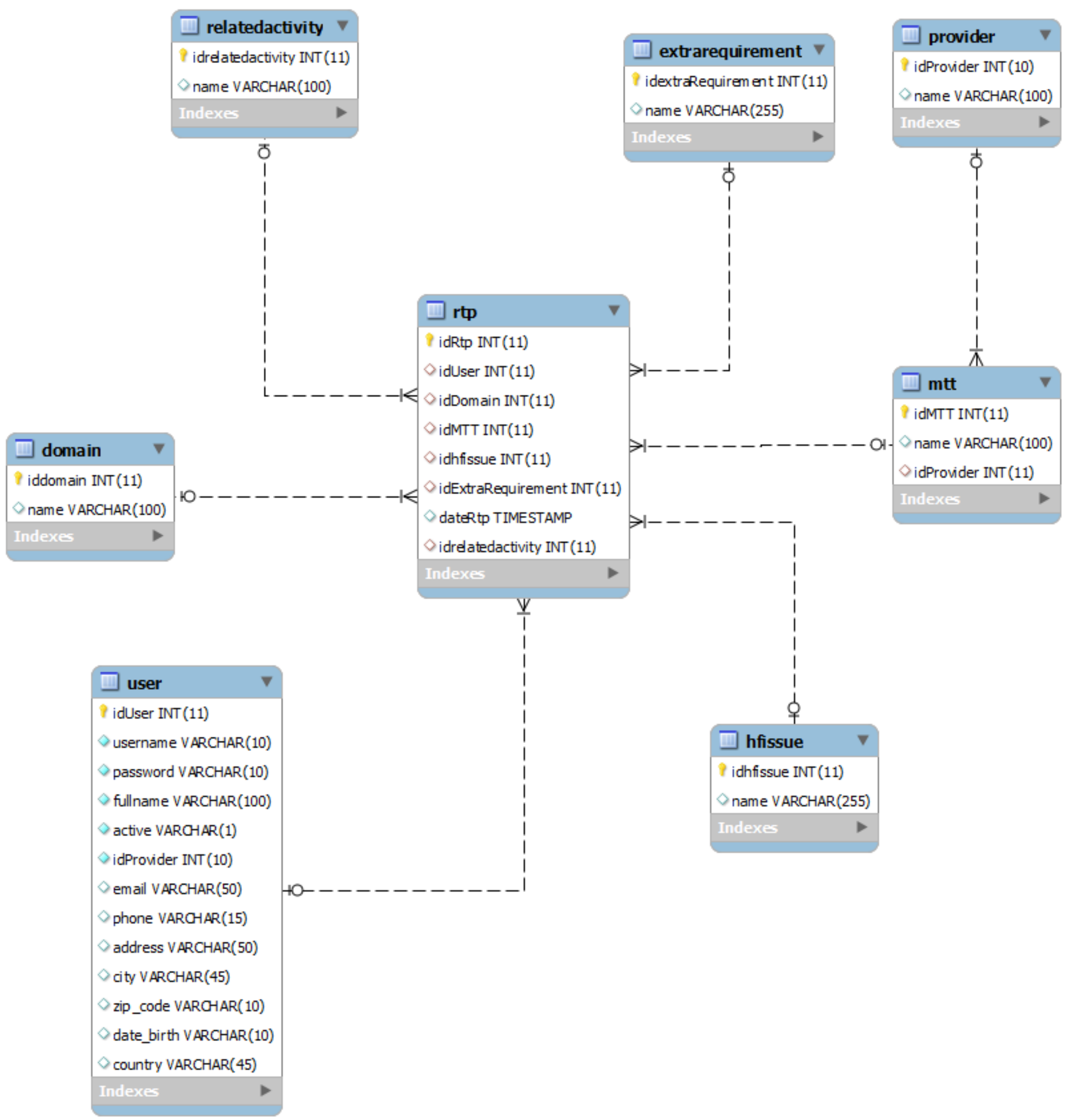

Figure 18: Platform Builder Database structure

In the Requirements document "Platform Builder requirements" are described the uses cases and all the requirements (functional or non-functional) that should be covered by Platform Builder application.

A description and picture for each use case of Platform Builder are detailed below:

HF-RTP Input Use Case

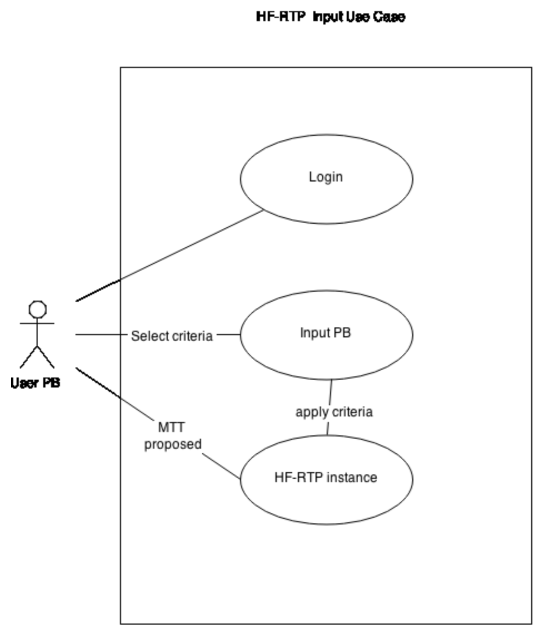

Figure 19: HF-RTP input use case

1. The use case starts when User PB click in the Login button, using user/password fields as identification to access it.
2. The System checks user/password in the database and shows next screen if the credentials are correct.
3. The System displays the data for "Domain" field from database.
4. The System filters each "Extra requirement", "HF Issue" depending on the "Domain" field selected.
5. The System proposes three options in the field "Related Activity": "Analysis", "Design" and "Evaluation".
6. The "User PB" click on "Get my HF-RTP" to get the specific MTT for the problem proposed.
7. The System filter by criteria selected by "User PB" and obtains the list of MTTs from database.

8. The Systems show the screen "List of Categorized tools" with the list of MTTs proposed based on the filters selected.

MTTs Comparator Use Case

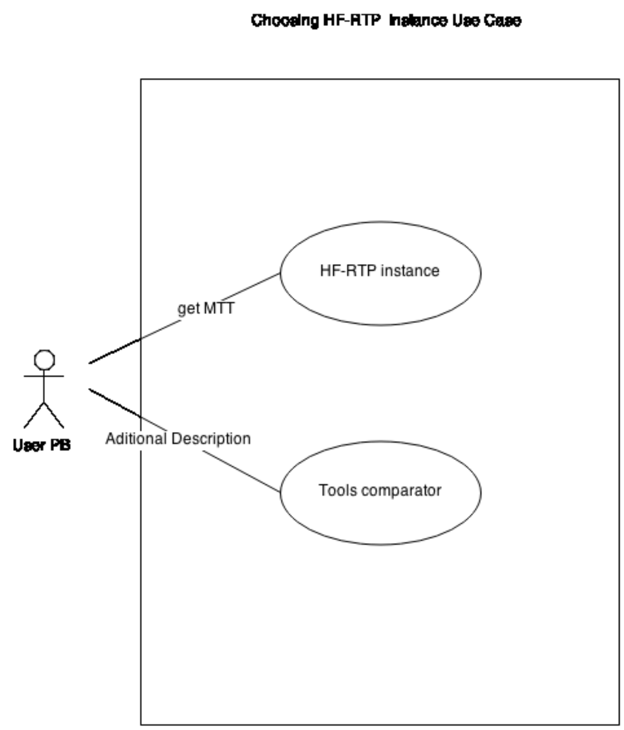

Figure 20: Choosing HF-RTP instance use case

1. The Systems shows the screen "List of Categorized tools" with the list of MTTs proposed based on the filters selected in the "Input HF" Screen.
2. The System retrieve from database all additional data from each Tool
3. The System put additional data in a comparative frame, showing what are the advantages and disadvantages for each tool.
4. The "User PB" can choose the best MTT for their issue, taking in account the data shown in the comparative details frame.

A list of most important requirements to be covered by Platform Builder is detailed in the document "Platform Builder requirements". This document has functional and not functional requirements.

2.7.3 Platform Builder user's handbook

The attached document "Platform Builder user handbook" details how the platform builder should be use, and it's a user guide with explanations, showing screens and the steps to resolve a HF issue, remembering that the aim goal of this application is to propose a list of MTT tools to help the User to take a decision.

2.7.4 Validation

The attached document "HoliDes Platform Builder validation report v0.1" checks all the requirements proposed in the document "Platform Builder requirements v0.1" are accomplished.
Each requirement (functional or non-functional) proposed in the requirements document should be checked and validated in the Platform Builder Application.

Requirement ID	REQ-NF03	**Type**	Non functional
Name	Compliance		
Description	The Platform Builder may be developed in compliance with specifications created by reference industry bodies, such as the IETF.		

Table 10: A non-functional requirement example

In this case should be checked and validated that: Platform Builder is accomplishing the IETF standard, checking the software and methodology used.

2.8 Status Baseline Assessments and Performance Indicators

The HoliDes baseline contains a set of measurements to evaluate the progress and results of the project according to a set of Performance Indicators (PI's). In the project, two types of Performance Indicators are defined: Generic Key PI's on project level and PI's on AdCoS level.

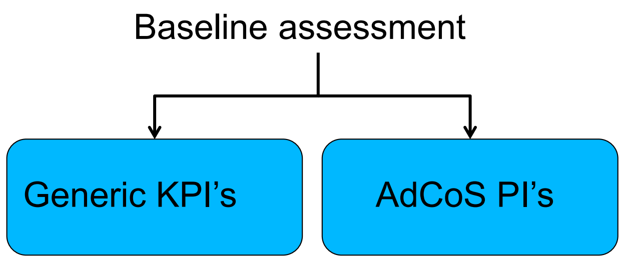

Figure 21: Baseline assessments

During the first project year already the assessment of the Generic KPI's was started-up and first results were incorporated into *D1.4* that was submitted January 2015. These Generic KPIs were defined more on an overall level (e.g. Time, Cost, and Effort). The KPI's are intended for project use, it addresses the major goals of the project to improve system development like Time, Resources, Cost.

In addition to the Generic KPI's also specific Performance indicators for each AdCoS are defined. In the TA a number of specific PIs of each work package are suggested and most of them have been adapted by the four AdCoS WPs (6-9). In some case new or modified PIs have been provided.

The specific PI's will be measured and subsequently reported at the end of the 2nd project year.

2.9 Conclusions

This deliverable has demonstrated that Milestone 3 has been reached in HoliDes. The RTP definition is now more mature and widely understood than previous releases. There now exists a clear understanding of the tools and techniques available through the RTP which can be utilised by the application work packages. The Human Factor issues and challenges have now been identified and are being addressed by the MTT owners.

The AdCoS have been defined in more detail than previous versions with the workflows listed current being implemented. The techniques and tools for the AdCoS modelling and analysis have also been improved through iterative cycles.

Through meetings and presentations AdCoS owners interacted with MTT developers to match problems with solutions to tailor them where necessary.

The techniques and tools have been integrated as per the use cases and are now listed as such on the MTT integration plan. Tools such as GreatSPN, Enterprise Architect and RT-MAPS have furthered the modelling capabilities of the AdCoS teams.

Appendix A

In version 1.0, an update of requirements was performed, but this only referred to the category of requirements for the HF-RTP, which will guide the developments of the project. After that, the project baseline was described, containing a set of measurements to evaluate the progress of the project according to a set of KPIs. This is the result of the WP and not a specific task. In that version of the HF-RTP the Human Factors integration concepts were also analysed, from the perspective of the different domains.

The conclusion was that when one refers to Human Factors, they are referring to frameworks for workflows and not the methods or tools. This is due to the informal nature of the human factors work. For example, descriptions in observation reports which are recorded as written text are very hard to analyse with algorithms. It would be very hard to for a computer to perform any kind of analysis on this data. Instead a human is required to perform some kind of analysis in order to deduce any kind of meaning.

An example of a more formalised method in action could be modelling behaviour as a state chart. The states and transitions represented as boxes and lines could be considered more formal since they have syntax. What's more, it would be very easy for software to perform some kind of analysis on a state chart to produce some meaning. This formalised syntax means that the state chart could easily be linked to other artefacts in your design process, for example, test cases and requirements, this enhancing traceability.

There are MTTs in HoliDes which perform analysis on textual descriptions and empirical testing. The outputs or results of these MTTs need to be formalized in compliance to HoliDes IOS standards. In this case, the appropriate standards are the Meta Models defined in WP2. These metamodels will be implemented in RDF XML following OSLC specifications.

The definition of the AdCoS methodology for the HF-RTP architecture was done in two approaches. The first was for the application domains to work with all of the MTT work packages to understand the capabilities of each method tool and technology. Based on these discussions WP 6 to 9 could make an assessment to understand if each MTT could assist them with their AdCoS development.

The second approach was for each of the domain work packages to create a workflow diagram depicting their current design process. This flow chart would span 5 areas of the lifecycle from requirements to testing showing the flow of information between operations. Additionally, areas of concerns where difficulties lie with their current method are highlighted with a yellow marker. Figure 22: The Command and Control Room Work FlowFigure 22 shows a workflow created by the Command and Control domain.

This process which describes the development of a control room highlights the current issues with requirements management, modelling and evaluation.

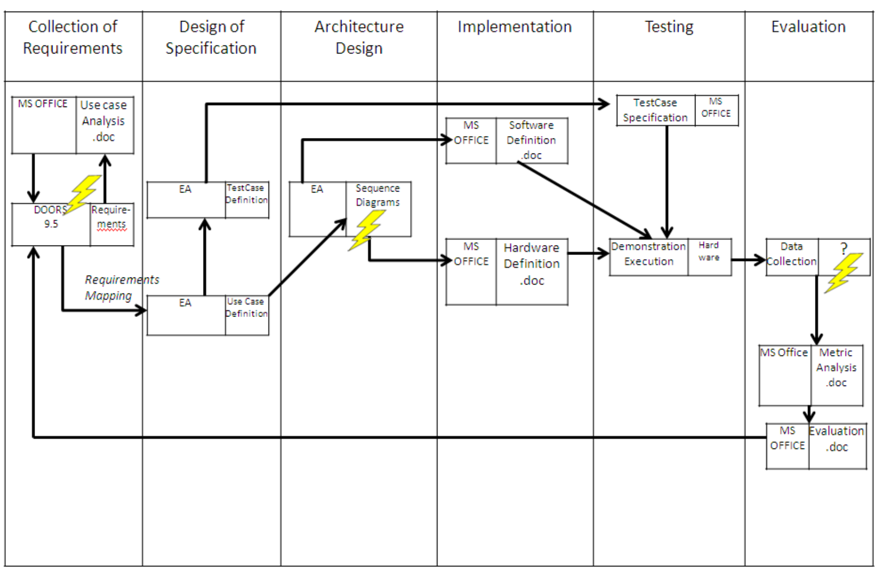

Figure 22: The Command and Control Room Work Flow

These diagrams which will be provided by WP6 to 9 along with descriptions of the design process were then made available to MTT providers so they could analyse the sequence of events to assess what contributions their tools could make. These diagrams were delivered in the version 1.0 of HF-RTP.

Due to the parallel nature of the activities in HoliDes, the workflows have had to be defined before the Common Meta Model. A regular iterative updating of workflows and Meta Models throughout the project will ensure that both model and workflow are relevant and in line with each other.

Version 1.0 of the HF-RTP deliverables explains why Meta Models are useful and how they can help to set the common basis for data exchange and interoperability. For that version none of the models is already finished and some models have been identified as candidates, but not discussed in detail. In next version of HF-RTP the overall meta-model will be developed.

In version 1.0 of HF-RTP, the infrastructure & interoperability standard provided the state of the art regarding existing interoperability standards, tool protocols, programming interfaces, runtime aspects, communication paradigms and exchange formats. Also the data management requirements and the generic engineering services were analysed in that version 1.0. HoliDes will build upon the approaches of the CESAR and MBAT projects, with the OSLC standard as basis, but defining the concepts specific for Human Factors, which do not exist at the moment.

The last section for version 1.0 of the HF-RTP was the one related to the tailoring rules & deployment. The result of the tailoring will be a specific instance of the RTP. In that version were defined a series of four steps to achieve the tailoring for each AdCoS in each domain. These rules will be followed in the Dx.4 for each WP domain, and adapted to the different processes.

Appendix B

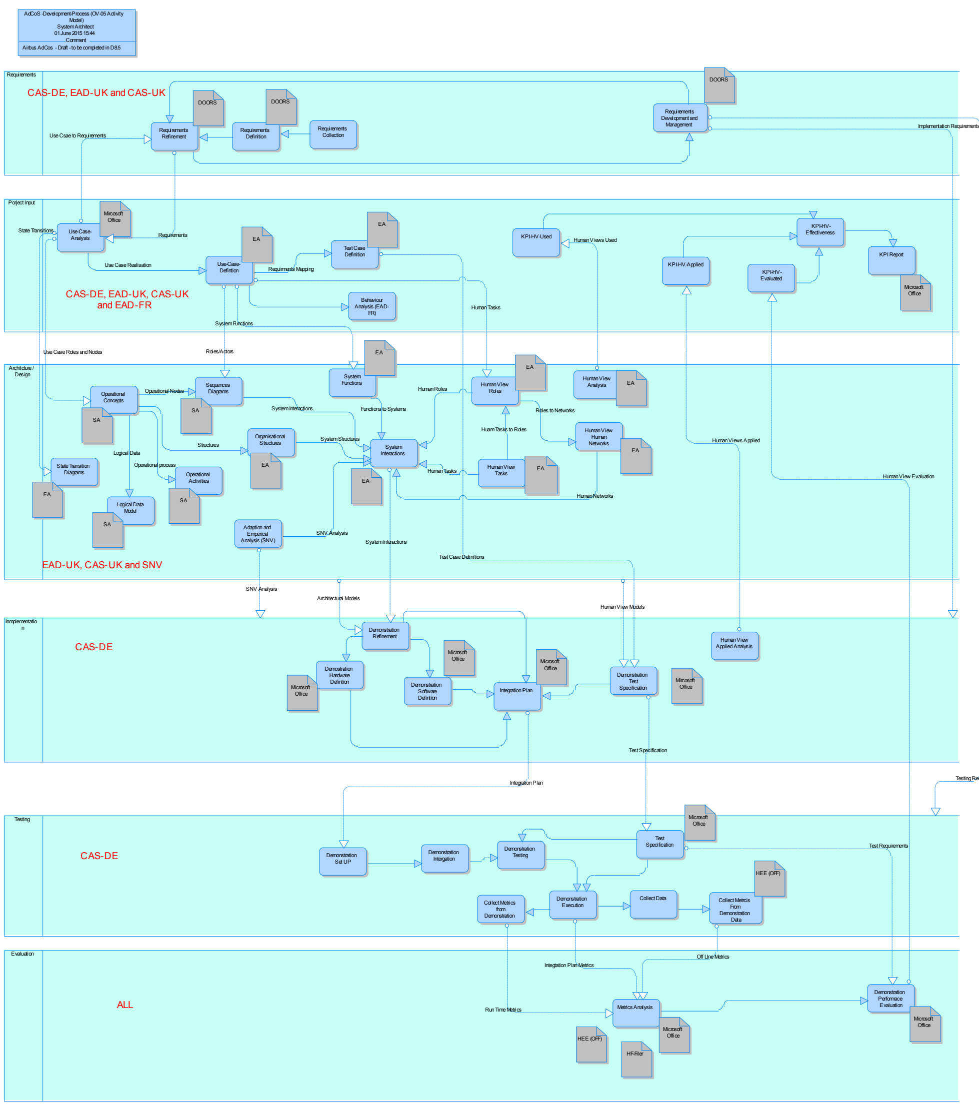

Figure 23: Complete view of the Command and Control room workflow

Appendix C

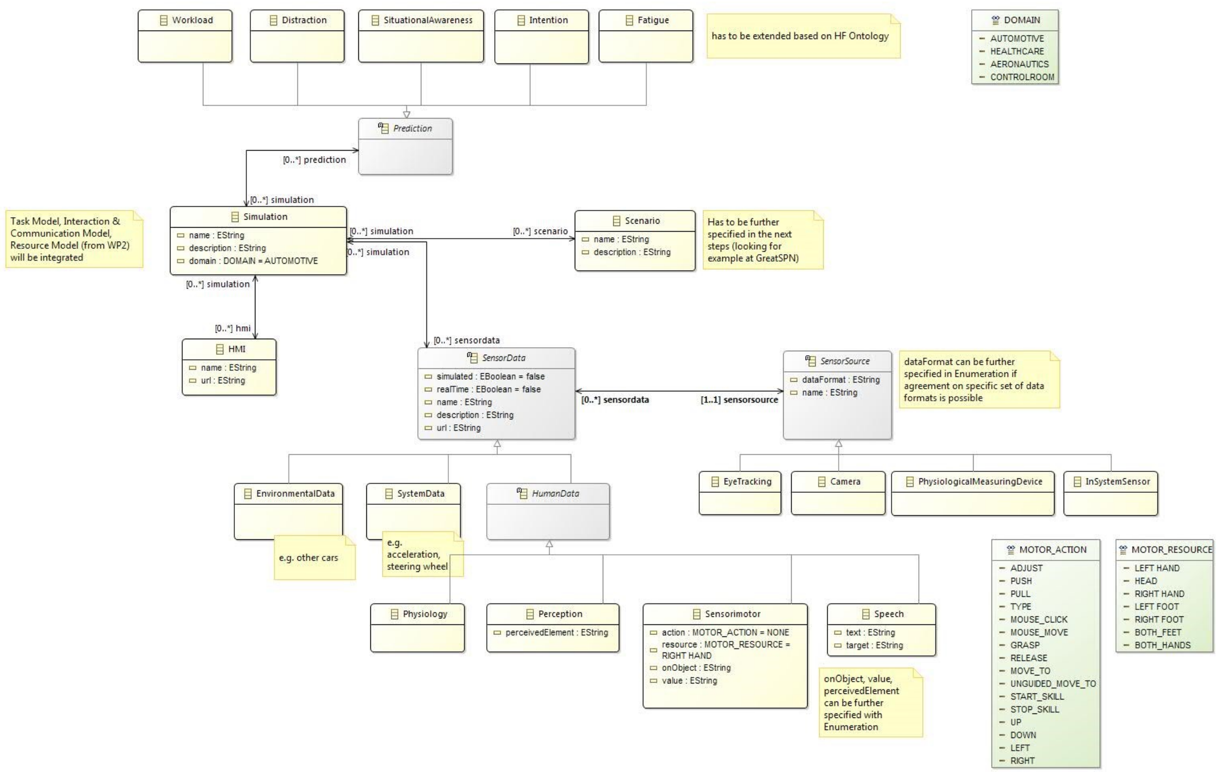

Figure 24: Initial version of the HF-CMM as Ecore Model