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

D2.2, D3.2, D4.2, D5.2 – Integration Plan

Project Number:	332933
Classification:	Public
Work Package(s):	WP2, WP3, WP4, WP5
Milestone:	M1
Document Version:	V1.0
Issue Date:	13.06.2014
Document Timescale:	Project Start Date: October 1, 2013
Start of the Document:	Month 6
Final version due:	Month 7
Deliverable Overview:	Main document: D2.2_Integration_Plan_for_MTT_into_HF-RTP **Annex I:** D2.2-D5.2_ Integration_Plan_Common_Part.docx - Public
Compiled by:	Jan-Patrick Osterloh – OFF
Authors:	Jan-Patrick Osterloh – OFF Robert Hofsink – PHI David Käthner – DLR
Reviewers:	Susanna Donatelli - UTO Morten Larsen - AWI
Technical Approval:	Jens Gärtner, EAD-DE-IW
Issue Authorisation:	Sebastian Feuerstack, OFF

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

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

RECORD OF REVISION		
Date (DD.MM.JJ)	Status Description	Author
14.05.2014	Initial Version	JPO
27.05.2014	Added content	JPO
27.05.2014	Corrections and Additions	DK
28.05.2014	2nd round	DK

Table of Contents

1 Introduction

During HoliDes, Methods, Techniques and Tools (MTTs) will be developed, with three purposes:

1. MTTs supporting the development process of AdCoS, and
2. MTTs supporting the adaption within the AdCoS.
3. MTTs supporting the analysis (model-based or empirical).

The MTTs will be developed in four WPs, as shown also in Figure 1:

- WP2 will develop MTTs for Modelling,
- WP3 will develop MTTs for Adaption,
- WP4 will develop MTTs for model-based Analysis, and
- WP5 will develop MTTs for empirical Analysis.

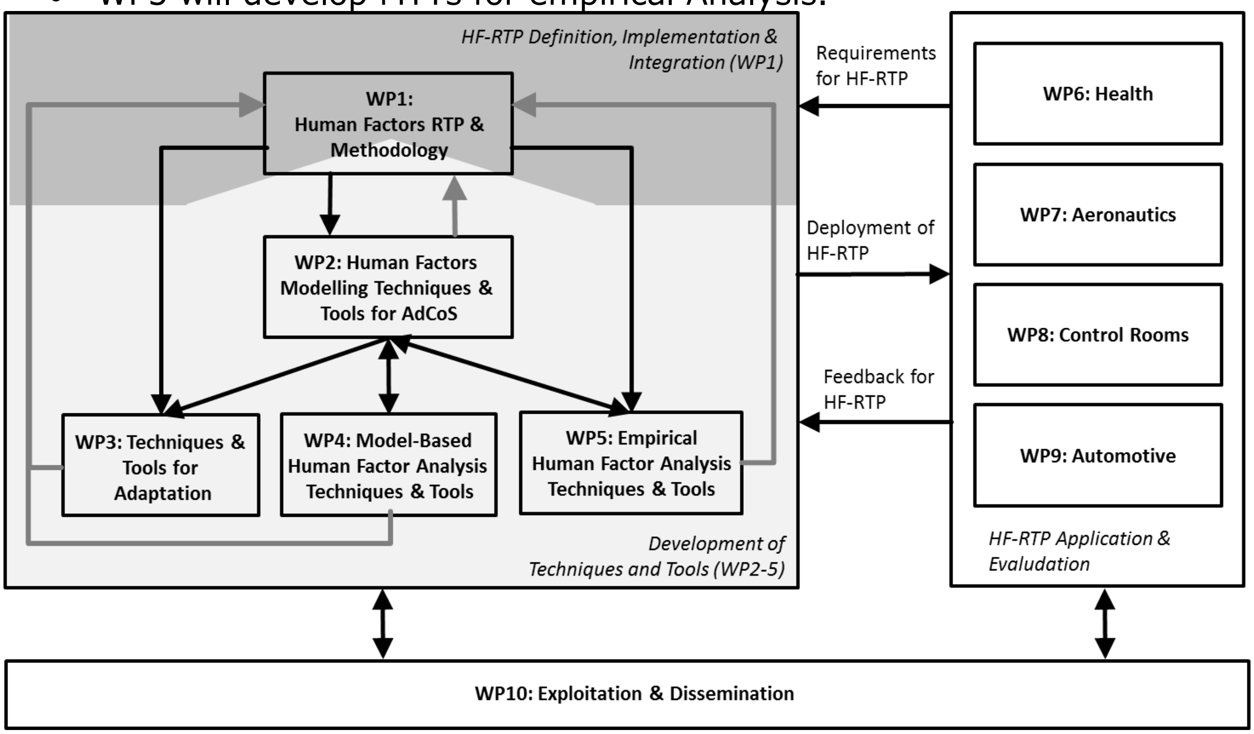

Figure 1: Work Package Overview

This report will describe our plans for integration of the MTTs into a HF-RTP and the accompanying methodology, by describing the successive steps to be followed to guarantee a smooth and iterative integration in the three Project Cycles, as shown in Figure 2.

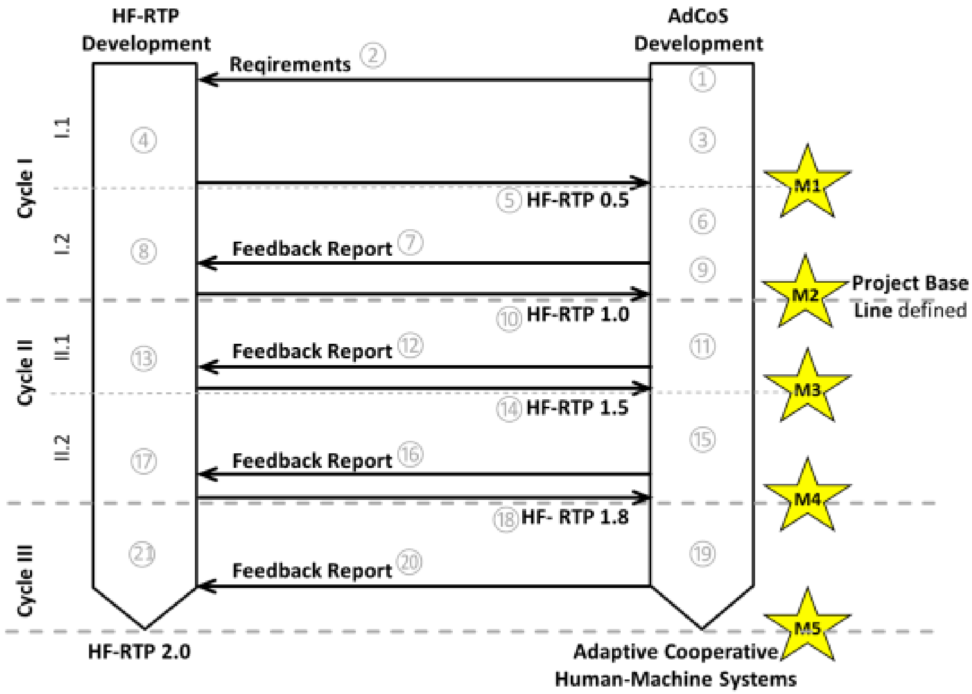

Figure 2: Project Cycles

In Project Cycle I the requirements ① for the HF-RTP and methodology (to be delivered to the HF-RTP development ②) as well as for the AdCoS in the four domains have been defined (see Deliverables D1.1 – D9.1). Furthermore, we will specify, model and assess ③ alternative designs for the AdCoS using pre-existing development methods, techniques and tools (MTTs). These, along with COTS tools, are provided mainly by the Research Partners. These MTTs are well suited for the development of cooperative human-machine systems, but not yet extended to the development of adaptive systems. The AdCoS will be built starting from pre-existing non-adaptive systems (incl. hardware and software) provided mainly by the Industrial Partners. In parallel to the AdCoS design, we will define the HF-RTP and will start the development of new and/or extended MTTs ④, by adding capabilities in particular for modelling and model-based analysis of adaptive systems. Cycle I will be divided into two sub cycles: I.1 and I.2. At the end of Cycle I.1 milestone M1 has been reached and a first definition of the HF-RTP (Vs0.5) will be available and will be delivered to the AdCoS development ⑤. The HF-RTP definition will be discussed with Industry Partners and will be assessed in the context of AdCoS development and qualification ⑥. Feedback (and requirements updates) will be given to the

HF-RTP development ⑦, where the definition will be improved accordingly ⑧ and where the development of new MTTs continues (towards Vs1.0). At the end of Cycle I.2 a Project Base Line will be defined ⑨ based on the application of the pre-existing MTTs as well as pre-existing non-adaptive systems.

At the beginning of Project Cycle II a new version of the HF-RTP (Vs1.0) will be delivered ⑩ (milestone M2). Vs1.0 consists of a set of MTTs (developed in ④) suited especially for the human centred design of adaptive systems and a definition (done in ⑧) of how these MTTs should be used in industrial development. Cycle II is again divided in two sub cycles: II.1 and II.2. In Cycle II.1 we will use the now extended MTTs (in HF-RTP 1.0) for modelling adaptive features of the AdCoS designs ⑪. Using model-based analysis we will assess the AdCoS designs and will "optimize" them according to innovative metrics for adaptive systems ⑪. Feedback (and requirements updates) on the application of HF-RTP VS1.0 will be given ⑫. The HF-RTP will now be implemented ⑬: software development to support work flows, the meta-model, infrastructure, methods as concepts with defined in- and outputs as well as generic services. Furthermore, MTTs will be improved based on feedback from their application and new MTTs will be added in particular for implementing adaptation: AdCoS context assessment, re-configuration and communication. At the beginning of Cycle II.2 a new version of the HF-RTP (Vs1.5) will be delivered ⑭ (milestone M3). Vs1.5 is the first fully implemented HF-RTP enabling interoperability of all MTTs. Vs1.5 will be used to implement ⑮ the modelled AdCoS in the four domains using specific new techniques and tools for implementing AdCoS. Feedback (and requirements updates) on the application of HF-RTP VS1.5 will be given ⑯. The HF-RTP will be improved ⑰ based on feedback on its application and new techniques and tools will be added focussing on empirical evaluation of AdCoS.

At the beginning of Project Cycle III a new version of the HF-RTP (Vs1.8) will be delivered ⑱ (milestone M4). Vs1.8 consists of a set of integrated and interoperable MTTs (developed and improved in ④, ⑬, ⑰) suited for modelling, model-based analysis, implementation and empirical analysis of AdCoS together with a definition and implementation (done in ⑧, ⑬, ⑰) to support their integrated and interoperable usage across all development phases. We will apply HF-RTP 1.8 for empirical evaluation of the implemented AdCoS in the four domains ⑲. We will assess the development of the AdCoS (incl. modelling, model-based analysis, implementation and empirical evaluation) against the Project Base Line which was established at the end of Cycle I.2. Feedback (and requirements updates) on the application and evaluation of HF-RTP VS2.0 will be given ⑳. The HF-RTP will be finalized ㉑ based on final feedback on its application. At milestone M5 the final results of the project will be delivered: HF-RTP Vs2.0 and the final developed and qualified AdCoS.

1.1 Process Definition

Writing this deliverable has been started by analysing the requirements and use-cases described by the AdCoS work packages. The requirements have been assigned to the pre-existing MTTs and some MTTs to be developed in HoliDes that are already planned. These MTTs have been presented and discussed with the AdCoS Partners in more detail during a workshop, held in Naples in May 2014. After this workshop, an initial table, that lists which MTT is used in which AdCoS WP, has been specified for the first cycle. As described above, the first cycle will consist of pre-existing MTTs and AdCoS development tools.
This initial planning is described in section 2. This initial assignment will be re-planned in the following cycles, based on the feedback of the AdCoS WPs, and new MTTs may be added to the list.
In the appendixes, which represent the WP relevant parts, details for each MTT are described in more detail:

- Purpose: A description of the main purpose of the MTT.
- Use Cases: Description of the Use-Cases for this MTT, i.e. how can it be used a) for development of an AdCoS or b) as part of an AdCoS.
- AdCoS Use-Cases: In which AdCoS use-cases this MTT may be applied.
- Input: Description, which input is needed for the MTT to produce its output (i.e. what does it need from the HF-RTP in order to work correctly). These descriptions will be either in text or UML.
- Output: Description, which output is produced by the MTT when applied.

From this description, especially the input and output description, it can be derived where integration in the HF-RTP is necessary, i.e. which information must come from the RTP and what is provided back to the RTP.

1.1.1 HF-RTP Integration

Today, the development of embedded systems in air and ground traffic, in control rooms and in the health domain gets more and more complex, resulting in a heterogeneous software environment being used during the design cycle. This means not only different system engineering methods and processes, but also different MTTs for their employment. A Reference Technology Platform (RTP) shall provide a complete systems engineering environment with a set of management or engineering methods and processes, as well as engineering tools. Several EU projects, like CESAR, MBAT and CRYSTAL have defined or are defining an RTP for embedded system design. In HoliDes, this RTP concept should be extended to cover human factors aspects.

An important point in RTP is the integration and interconnection of tools between all relevant business and development processes. The main idea behind a RTP is to follow the Interoperability Specification (IOS) to enable the data exchange through unification / standardization of the underlying data models and data exchange format. IOS is a specification which is currently being developed by other ongoing Artemis research projects, such as Crystal [4]. Whilst the exact details of IOS are yet to be defined, the underlying principles are clear. That is, tools will all share data using the same technologies which made the internet so successful. Those technologies are HTTP, RDF, XML, REST and Linked Data.

Therefore, a key component of the IOS is the OSLC [1] (Open Services for Lifecycle Collaboration) standard. More details on RTP, and IOS can be found at the CESAR [2], MBAT [3] and CRYSTAL [4] project pages.

1.1.2 What is OSLC?

Open Services for Lifecycle Collaboration (OSLC) is an open community creating specifications for integrating tools. These specifications allow conforming independent software and product lifecycle tools to integrate their data and workflows in support of end-to-end lifecycle processes. The goal of OSLC is to create specifications for interactions between tools. OSLC is not trying to standardize the behaviour or capability of any tool or class of tool, like a test tool or a requirements management tool. The OSLC core specifies a minimum amount of protocol and a small number of resource types to allow two such tools to work together relatively seamlessly. Even within a particular resource types specified by an OSLC workgroup, the goal is to define only properties that are valuable for integration, not all the properties that might be present in a particular tool's resources. OSLC also tries to accommodate a wide variety of implementation technologies, and be equally relevant to both existing tools and to newly-built ones. OSLC is based on the W3C Linked Data[2]. The Linked Data Standard main rules are:

1. Use Uniform Resource Identifier (URIs) as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL).
4. Include links to other URIs, so that they can discover more things.

In OSLC, each artefact in the lifecycle – for example, a requirement, defect, test case, source file, or development plan and so on – is an HTTP resource that is manipulated using the standard methods of the HTTP specification (GET, PUT, POST, DELETE). Following the third rule of linked data, each resource has an RDF representation – OSLC mandates RDF/XML, which is the most widely adopted RDF notation - but it allows representations in other formats, like JSON or HTML.

The OSLC Core specification defines a number of simple usage patterns of HTTP and RDF and a small number of resource types that help tools integrate and make the lifecycle work. The OSLC domain workgroups specify additional resource types, specific to their lifecycle domain, but do not add new protocol. Thus, OSLC consists of two main concepts:

- Service Providers: Service Providers are the central organizing concept of OSLC, enabling tools to expose resources and allowing consumers to navigate to all of the resources, and create new ones.
- OSLC Resources: OSLC resources describe the domain specific concepts that are available via the Service Providers.

In other words, OSLC describes what concepts to share and how to share them.

In order to integrate the HoliDes MTTs into the RTP, the following steps have to be performed:

1) Use Case identification (HoliDes Deliverables D1.1 to D9.1)
2) Identify MTT to be connected to RTP
3) Defining the concepts to be shared via RTP
4) Implementation of IOS/OSLC connection to RTP for each MTT
5) RTP Tailoring by AdCoS developers to RTP instance
6) Deployment and configuration of AdCoS RTP instance

[2] http://www.w3.org/DesignIssues/LinkedData.html

2 MTTs for Integration

The following table gives an overview on the MTTs developed in HoliDes, and an allocation of the MTTs to the Use Cases for the first cycle. Not all Tools have been selected to be used in the first cycle, because of the tool availability and maturity. Not all Use-Cases have allocated tools to be used in the first cycle for the same reason.

			WP6					WP7		WP8		WP9				
			Safe patient transfer	Guided Patient positioning	Safe parallel transmit scanning	iXR 3D Acquisition	Patient data access	Diversion Assistance	Transition Training	Border Control Room	Energy Control Room	Frontal Collision Scenario	Overtaking			
Tool	WP(s)	Partner								area surveillance (CAS)	energy network surveillance (IRN)		fully automated vehicle (IAS)	driver adaptive maneuver preference (DLR)	TAK demon-strator	CRF demon-strator
GreatSPN	D2.2, D2.4	UTO		x	x			x			x	x				
RTMaps	D4.2	INT						x				x				
Anaconda	D4.2	BUT						x								
Data Race Detector & Healer	D4.2	BUT						x								
Search Bestie	D4.2	BUT						x								
HEE	D2.2, D2.4	OFF						x		x	x					
PED	D2.2	OFF							x							
Training Manager	D2.2	OFF							x							
Adaption Tool	D3.2	BUT						x								
Adaption Tool	D3.2	TWT						x								
Adaption Tool	D3.2	TEC						x								
Certifier Tool	D4.2	TEC						x								
BAD-MoB	D3.2	OFF														x
CoSimECS	D4.2	OFF						x								
CASCaS	D2.2	OFF						x			(x)				x	
djnn		ENA									(x)					
Driving Simulator	D5.2	DLR												x		
Driving Simulator incl. Pro-SIVIC	D2.2, D2.4	IFS										x				
Driving Simulator	D5.2	TAK													x	
Driving Simulator	D5.2	REL														x
Theater Technique	D5.2	DLR											x	x		
Driver State Classification	D2.2, D5.2	TWT														
Driver State Classification	D5.2	UTO														
Driver State Classification	D3.2	CRF														x
Driver State Classification	D2.2	IFS										x				
Virtual Co-Driver	D5.2	UTO										x				x
CONFORM	D5.2	DLR												x		

3 References

[1] OSLC - http://open-services.net/
[2] CESAR - http://www.cesarproject.eu/
[3] MBAT - https://www.mbat-artemis.eu/home/
[4] Crystal Project - http://www.crystal-artemis.eu/