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

D8.6 Tailored HF-RTP and Methodology Vs1.5 for the Control Room Domain

Project Number:	332933
Classification:	Public
Work Package(s):	WP8
Milestone:	M4
Document Version:	V1.1
Issue Date:	10.10.2015
Document Timescale:	Project Start Date: October 1, 2013
Start of the Document:	Month 20
Final version due:	Month 23
Deliverable Overview:	**Main document:** D8.6 Tailored HF-RTP and Methodology Vs1.5 for the Control Room Domain
Annexes	HOLIDES_Task Analysis_v2.docx
Compiled by:	Ian Giblett
Authors:	Ian Giblett, Martin Boecker, Elisa Landini, Andreas Riegger
Reviewers:	Johann Christian Koerber, Zdenek Moravek
Technical Approval:	Jens Gärtner, Airbus Group Innovations
Issue Authorisation:	Sebastian Feuerstack, OFFIS e.V.

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

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

RECORD OF REVISION		
Date (DD.MM.JJ)	Status Description	Author
15.07.15	Draft structure of deliverable	Ian Giblett
11.09.15	Early draft released	Ian Giblett
22.09.15	Contributions added	Ian Giblett
25.09.15	Edits accepted	Ian Giblett

Table of Contents

Executive Summary

The following document describes the process of building the HF-RTP, which is being developed in the HoliDes project (WP1), to the Control Room Domain, with a special focus on the description of the AdCoS and tool chains developed by the partners. It is the follow up deliverable of D8.2. It is explained how the tailoring methodology provided by WP1 is applied in the Control Room Domain.

This document is the result of many collaboration activities between the AdCoS developers in WP8 and the method tools and technology (MTT) providers in WP 2 to 5. There are many MTTs in HoliDes but not all of them are relevant for every AdCoS. Those which can assist in the Control Room Domain design processes are listed here.

1 Introduction

This deliverable describes how the HF-RTP methodology Vs1.5 and the HF-RTP, which are being developed in WP1, are applied and tailored in the Control Room Domain. In particular, it focuses on the application of the tailoring rules provided by WP1 and defined in D1.4.

1.1 Objective of the document

Deliverable D8.6 describes the results of the HF-RTP tailoring methodology applied to the Control Room Domain for the third project cycle.

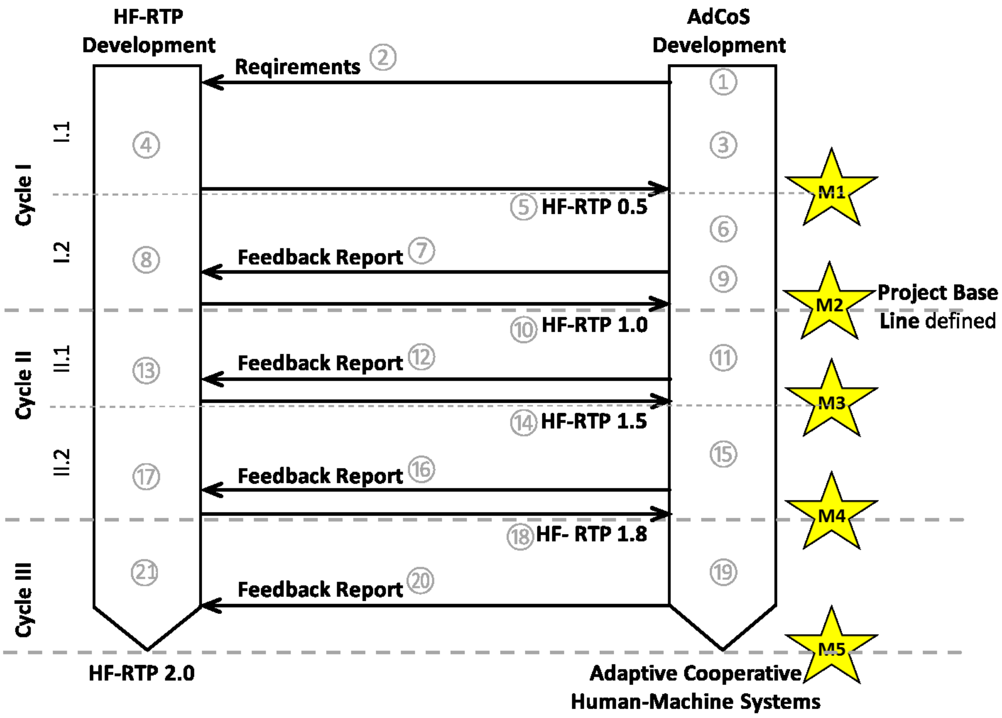

Fig. 1 - Overall workflow taken from the HoliDes proposal

The HF-RTP and the tailoring methodology (version 1.5) developed in WP1 and delivered in D1.4 are applied to the AdCoS of the Control Room Domain.

The previous version for tailoring of the HF-RTP in WP8 has already been provided in D8.4, which was based on the HF-RTP version 0.5 (M1).

1.2 Structure of the document

This document starts with a description of the common Meta Model proposed by WP2 and goes on to provide feedback. Next the platform builder from WP1 is assessed. Feedback on this is provided in a series of test cases.

After an insight into the platform builder, there is a short discourse on the current state of the AdCoS' of work package 8. Only a brief description is given here since the information is readily available in previous deliverables.

Section 3 is the core of this document since it describes the actual HF-RTP instance as used in WP8. It shows a list of MTTs utilised by both the Command and Control Room and the Energy Control Room and describes how each is used.

Finally, the concluding chapter seeks to summarise what has been done and looks forward to the next stages and identifies areas of work for the next stage.

1.3 Common Meta Model

Fig. 2 shows the Common Meta Model as proposed by Work Package 2. It is intended to be a logical data model used for the exchange of components in a Human Factors reference technology platform.

It covers all of the four domains (Control room, Aerospace, Automotive, and Healthcare) albeit in different levels of granularity.

The following table lists three observations of the Meta Model which should be addressed in order to maximise its potential.

Feedback from WP8

No.	Category	Feedback Comment
01	Modelling Structure	UML is not regarded as the best way to define a 'logical data model' due to its contamination with generalisations and definitions limited to only classes.

No.	Category	Feedback Comment
02	Human Operator Modelling	The Meta Model in its current format focuses heavily only the motor actions performed on a human and not so much on the analysis of humans. For example, information exchanges between MTTs covering the analysis of task difficulty would not be covered here.
03	Human Operator Modelling	Important concepts which come from the Human Views used in the NAF Human View framework are missing. These include meta data such as training, experience and qualifications. (For more information on the Human View modelling, please refer to D8.5.)

1.4 Meta Model Reference Implementation

So far, no reference implementation or guide explaining how to use the Meta Model has been provided by WP2. With the inputs provided to this date, it would be difficult for anyone in any work package to create a working integration between two tools.

What has been proposed recently by WP1 is that the integration between the HF Filer tool and Enterprise Architect is developed as the reference example for using the Meta Model to link two tools. This would be then used as a guide to assist the other application Work Packages 6 to 9.

This diagram shows the latest baseline version of the Meta Model at the time of writing.

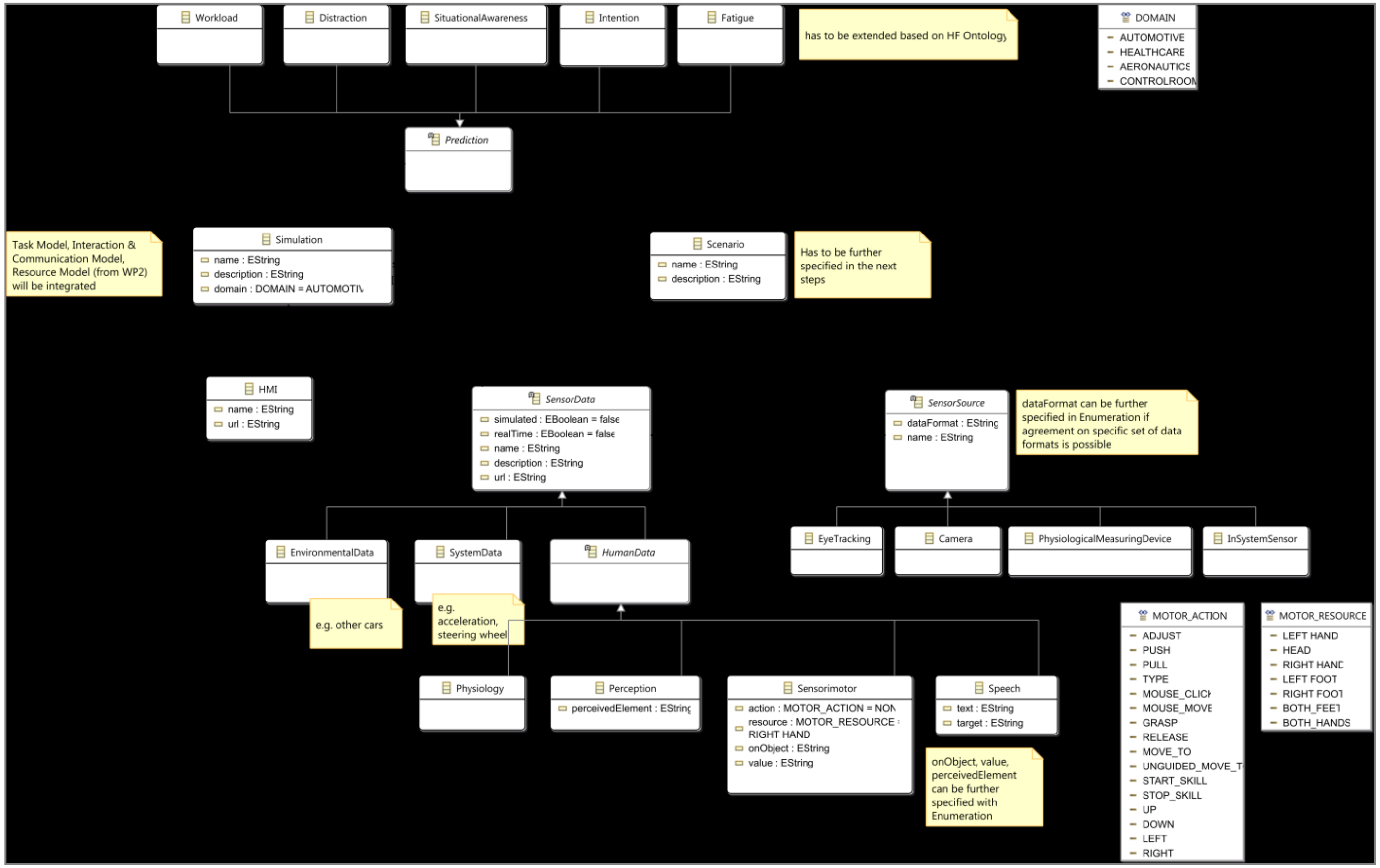

Fig. 2 - The Common Meta Model from WP2

1.5 Platform Builder

The platform builder in HoliDes from Work Package 1 has been made available through the HoliDes website: www.holides.eu/PlatformBuilder.

Fig. 3 shows the login to the platform builder.

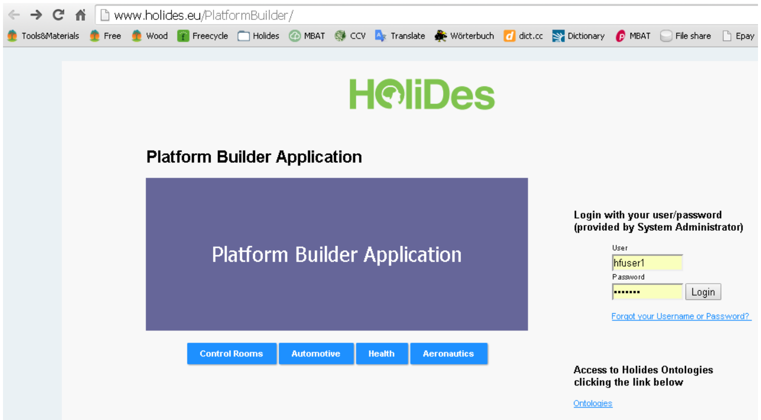

Fig. 3 - The Login Screen for the Platform Builder

The idea behind the platform builder is to allow someone developing a tool chain to see what is available in the Human Factors domain and let them make an informed decision as to what is available for them to use.

Fig. 4 shows the screen for querying the MTTs. The idea is that one selects the domain in which they are interested in.

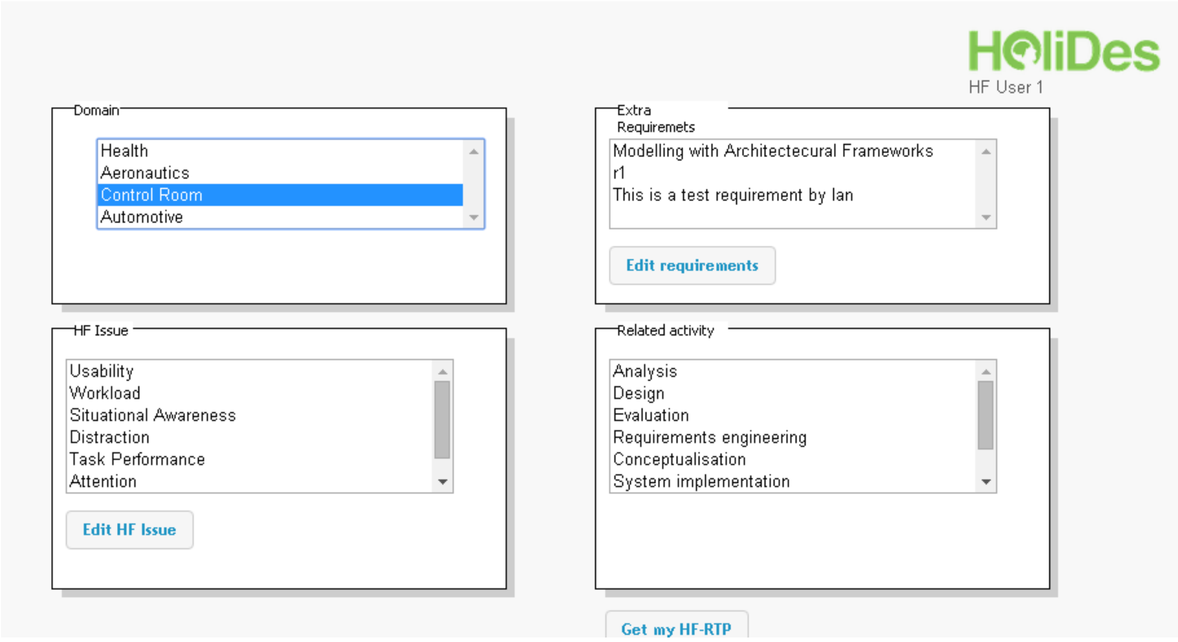

Fig. 4 – Screen for the querying of MTTS

1.5.1 Testing the platform builder

In the WP8 Command and Control Room use cases, the following tools had been identified as potentially useful and, therefore, would expect to be available through the platform builder:

- DOORS Next Generation
- LEA (Though later identified to be unsuitable and replaced by KNIME.)
- GreatSPN
- Enterprise Architect
- HEE Tool
- HF Filer Tool

For more information on the MTT investigation, please refer to D1.5.

Table 1 - Platform Builder test Cases

Platform Builder Test Cases			
Test No.	*Description*	*Expected Result*	*Actual Result*
1.	Select the Control Room Domain and under one of the HF Issues located the following MTT: **DOORS Next Generation**	**DOORS Next Generation** appears as an MTT on the screen.	DOORS appears when the user looks for requirements MTTs.
2.	Select the Control Room Domain and under one of the HF Issues located the following MTT: **LEA**	**LEA** appears as an MTT on the screen.	KNIME is being investigated as a substitute for LEA. Neither KNIME nor LEA appeared on the platform builder.
3.	Select the Control Room Domain and under one of the HF Issues located the following MTT: **GreatSPN**	**GreatSPN** appears as an MTT on the screen.	Did not appear on platform builder.
4.	Select the Control Room Domain and under one of the HF Issues located the following MTT: **Enterprise Architect**	**Enterprise Architect** appears as an MTT on the screen.	EA appears when the user looks at design MTTs.
5.	Select the Control Room Domain and under one of the HF Issues located the following MTT: **HEE Tool**	**HEE Tool** appears as an MTT on the screen.	The HEE tool appears when the user looks for HF Issues pertaining to Workload.

Platform Builder Test Cases			
Test No.	*Description*	*Expected Result*	*Actual Result*
6.	The user selects the Control Room Domain and under one of the HF Issues located the following MTT: **HF Filer Tool**	**HF Filer Tool** appears as an MTT on the screen.	Did not appear on the platform builder.
7.	The user is to create a project which contains all of the selected MTTS.	A named container is made available which houses all of the selected MTTs.	The feature existed to new MTTs, but the new MTTs could not be seen when creating a new HF-RTP instance.
8.	The user attempts to deduce which of the MTTs are currently tailored to which other MTTs.	Some sort of matrix or graphical representation is made available which allows the user to see which MTTs can collaborate with other MTTs.	Feature not found.

As not all of the MMTs were present, the following test cases are proposed in order to deal with missing information.

Platform Builder Test Cases			
Test No.	*Description*	*Expected Result*	*Actual Result*
9.	The MTT manager is used to add new MTTs into the Platform Builder Database.	The new MTT is now available for selection in the defining of a new HF-RTP instance.	MTT was not available.

Platform Builder Test Cases			
Test No.	*Description*	*Expected Result*	*Actual Result*
10.	The MTT manager is used to remove an MTT.	The MTT is now not available in the platform builder.	Deleting the MTT was not possible.

Results of the Platform Builder testing indicate that of the ten tests, only three pass and four fail. The passes are attributed to required MTTs being present. The fails are attributed to MTTs either not being available or the functionality to add them not existing or working.

This feedback will be sent to WP1 for improvements in the next version.

2 AdCoS Development

Fig. 5 shows the interaction diagram as used by the Command and Control Room. It clearly shows how the AdCoS is linked to the HF-RTP Instance. For more information on this please refer to "D8.5 Modelled and Model-based Analysis of the Control Room AdCoS and HF-RTP Requirements Definition Update".

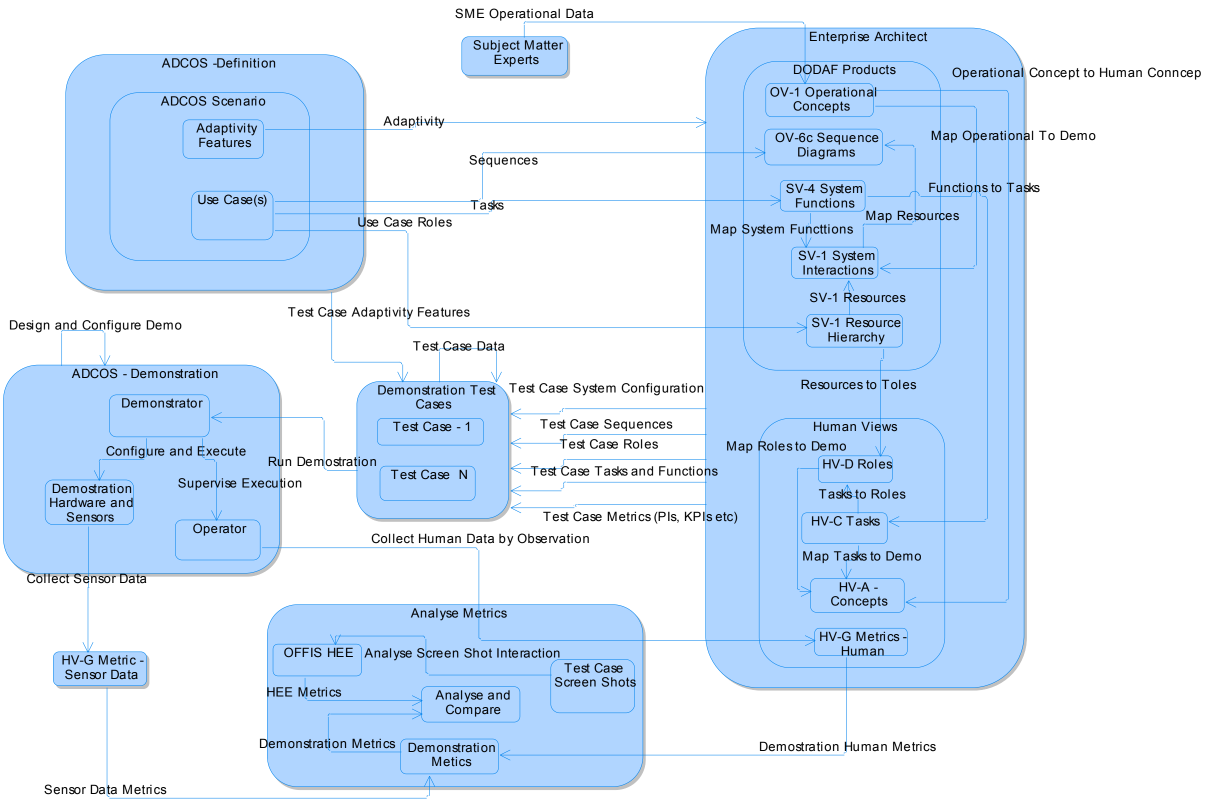

Fig. 5 – The interaction diagram for the Command and Control Room

2.1 Command and Control AdCoS

The Command and Control (C^2) AdCoS aims at increasing a C^2 organisation's effectiveness and security. The six use cases address the operator's mental and physical state (UC1, UC2, UC3), the identification of observable and exploitable behaviour patterns of operators (UC4), the re-distribution of work in case of high individual workloads of operators (UC5) and the assisted operator classification (UC6). More information on the 6 use cases can be found in D8.5 section 5.1.1.4.

2.1.1 Status of the AdCoS development

A draft specification document for the Command and Control AdCoS has been created that forms the basis for the further implementation of the five use cases to be demonstrated in the review event in November 2015 (Use Case 6 will be addressed in the event, but only as a poster and not in an implemented form).

The demonstration system will consist of a complete Command and Control network with two workplaces (a response operator workplace and a response supervisor workplace, see Fig. 6).

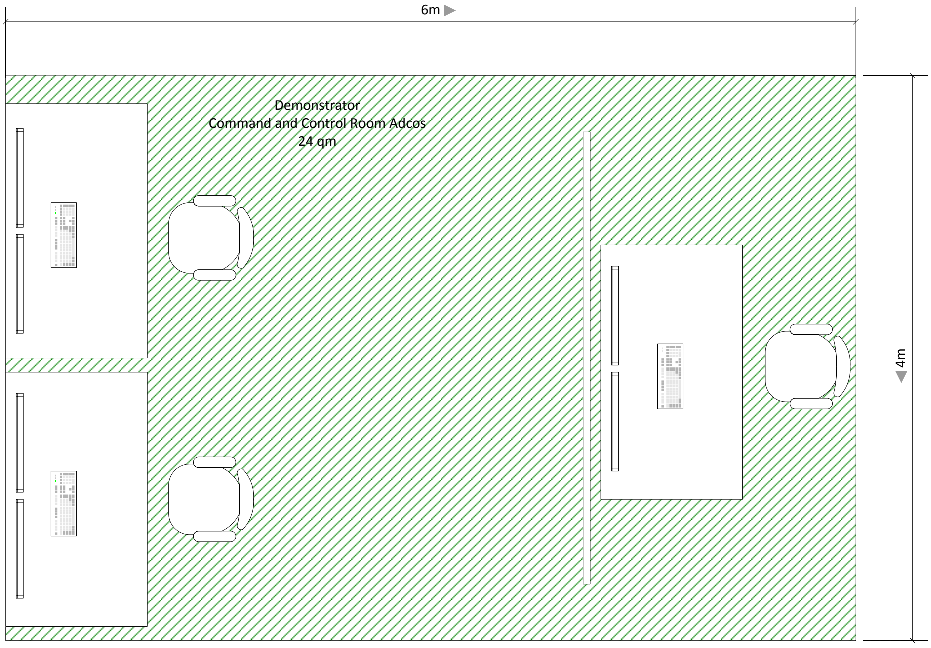

Fig. 6 - Spatial Arrangement of the WP 8 AIRBUS DS AdCoS Demonstration

The technical requirements for the demonstration system have been identified and missing hardware and software are being ordered.

A demonstration scenario is being specified that contains all information pertinent to the demonstration of the AdCoS.

The demonstration of the AIRBUS DS AdCoS will be conducted as follows:

- Explanation of the status quo ante (i.e. the current situation in the Control Room without the AdCoS adaptation). This includes an

explanation of the problems the AdCoS tries to solve (in particular, increasing the organisation's effectiveness and security).
- Explanation of the approach chosen for the AdCoS adaptation.
- Explanation of the sensor-based interaction technologies and the opportunities they offer.
- Explanation and demonstration of the five use cases supported by the demonstrator.

2.2 Energy Control Room AdCoS

2.2.1 Task analysis

A detailed task analysis (described in the Annex I) has been conducted by IRN and REL. It takes into account the work done by the operators once the user request has been categorized and included into the system.
The approach followed during the task analysis is top - down (from tasks to functions).

<u>Tasks of the Control Room operator</u>

1. Call receiving
   - 1.1. Collecting info about the malfunction (4W: who, what, where, when)
   - 1.2. Giving instruction to the user for a safe behaviour
   - 1.3. Launch of the intervention management
2. Management of emergency events
   - 2.1. Emergency team activation
   - 2.2. Deletion of the task from the pending list
3. Management of routine events
   - 3.1. Selection of the most suitable technician
   - 3.2. Deletion of the task from the pending list

<u>Tasks of the on field technician</u>

4. Call receiving from the Control Room
   - 4.1. Collecting info about the malfunction (4W: who, what, when, where)
   - 4.2. Accepting the task
   - 4.3. Rejecting the task

As the domain literature explains (Harrison et al. 2002), the designer of an adaptive automated system bases his/her decisions about automation of the studied functions on their use in the context of the scenario under consideration. Suitability for total automation is not based solely upon the technical feasibility of a solution. It is also based upon the function's relation to the roles. If a function is not seen to be separable from an operator's role then it cannot be totally automated. Doing so would interfere with the operator's ability to do the job effectively.

This is the reason why the tasks that refers with the user front-end cannot be handled by the AdCoS. This means that the workflow of tasks 1.1, 1.2, 1.3 cannot be optimized by the intervention of the AdCoS, hence the related task analysis and function analysis will not be changed.

For each task a deeply structured analysis has been performed. The applied methodology has followed the following steps:

1. Selecting suitable functions for the AdCoS: Firstly the function suitable for the automation (AdCoS) are detected.
2. Task review: then, according to what achieved in step 1, the task is reviewed, highlighting which subtasks become fully or partially automated.
3. Task redesign: since the introduction of automation allows at improving the workflow deleting some manual processes, the task was redesigned, shifting where possible from partially automated functions to fully automated functions. The task redesign process usually lead toward a subtask reduction.
4. Function optimization: according to the redesigned task, the function allocation is optimized, allocating to the automation the subtasks best suitable for automation, and allocating to humans task best suitable to humans.

The introduction of the AdCoS in the Control Room allowed:

- to optimise the workflow, reducing the number of subtask for each task and in some cases to unify some tasks (-35%)
- to reduce the tasks manually performed (-77%)
- to introduce a consistent number of fully automated tasks (+35%)
- to reduce the tasks which require a communication by phone (-81%)

Feedback to WP2 regarding the task analysis

The task analysis methodology was very useful to identify that tasks that could be improved by introducing the AdCoS and quantifying this improvement.

A potential improvement would be to use a tool that can easily and graphically support the designer in the creation of the task modelling, by facilitating the inserting of the information about the way used by the operator/technician to perform the task (e.g. manual, by phone), associate an estimated duration to each task, and categorize it for the functional analysis (information, decision, action).

2.2.2 Focus group

A focus group has been structured and performed by SNV in order to collect feedback and comments on the AdCoS.

Feedback to WP5 regarding empirical support for the design

The support of SNV for the design of the empirical activity in collaboration with IRN (i.e. the focus group) was fundamental to collect structured information that provided valuable data for the improvement of all systems (server, HMI for the operators and mobile app for the technicians).

A potential improvement for this methodology would be the integration with the HF Filer, in order to be able to share the design of the experiment with other partners, to easily access it for other focus groups (e.g. to repeat the same empirical activity during the last project cycle) and to facilitate the access of the data for the analysis.

2.2.3 Status of the AdCoS development

Fig. 7 shows the overall architecture of the IRN AdCoS.

It includes two macro elements:

- A server
- An HMI application for the Control Room operators
- A mobile app for the technicians in the field

The server includes:

- an Entity-Relationship Data Base with the data about malfunctions and technicians
- the engine with the **Decision Algorithms** for the automatic selection of the most appropriate technician for each intervention
- a **proxy** to dispatch the information to the operators and the technicians (respectively through a web browser interface and a mobile app)

The HMI application for the Control Room operators is a web-based application that lets them access the list of interventions, assign an intervention to a specific technician (according to the priority list provided by the server) and see which technician is in charge of which interventions.

The **HMI Application for the Technicians** lets them accept an intervention assignment and access the corresponding data (type of intervention, address, etc.).

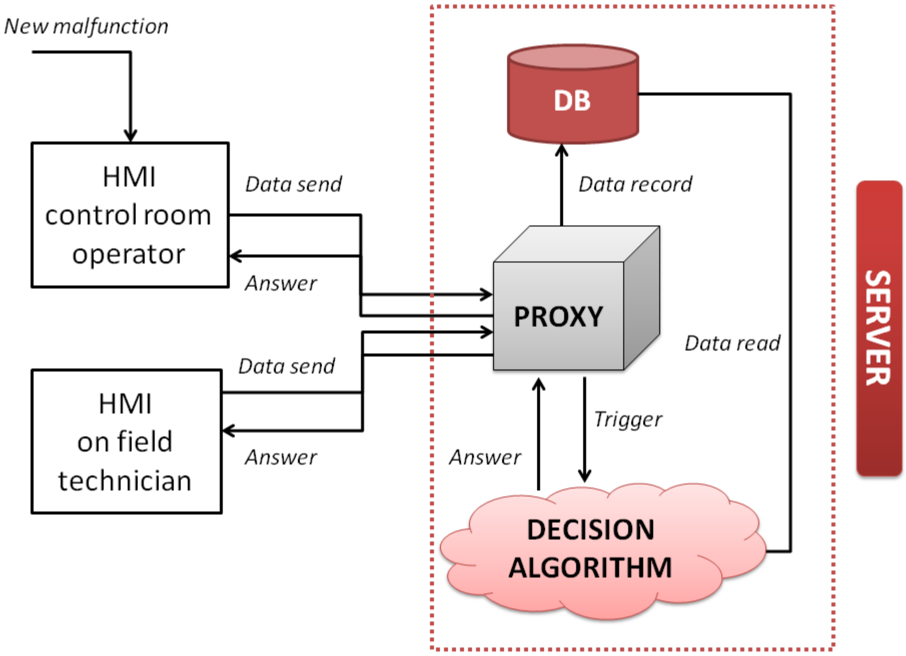

Fig. 7 - Architecture of the Control Room AdCoS

In order to implement the communication and the interactions among the elements, a sequence diagram has been drafted (shown in Fig. 8).

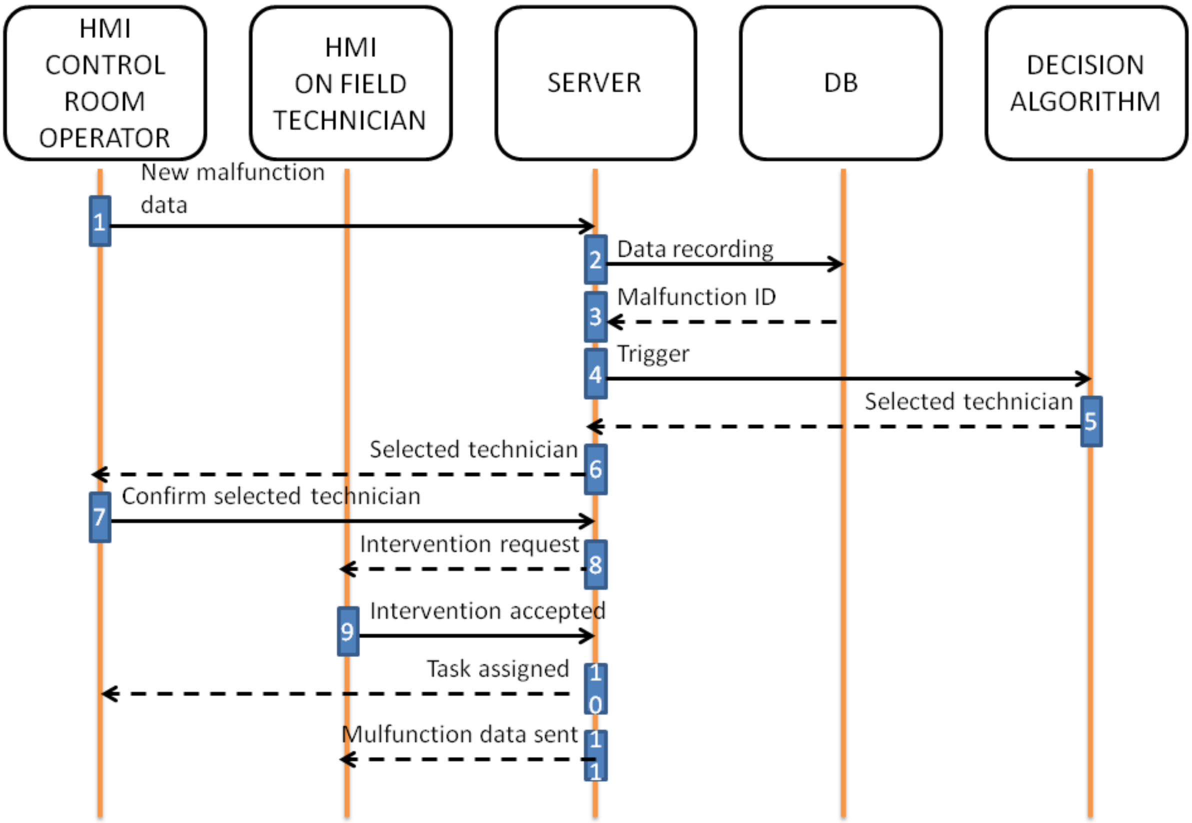

Fig. 8 - Sequence Diagram of the Control Room AdCoS

As shown in step 6, the server (and in particular the Decision Algorithm) is in charge of automatically define a priority list of the most appropriate technicians to be assigned to the intervention.

The criteria taken into account by the Decision Algorithm to find the most suitable technician are 5, presented according to the filter order:

1. Geographic zone of authority
2. Skills
3. Engagement
4. Work shift
5. Actual position

The logic of the algorithm is the following:

1. Check malfunction zone

   Nb The malfunction zone is the geographic area in which the potential fault has been detected.

2. Extract the list of technician for that zone
3. Check the skills of each technician
4. Match technician skills and malfunction type
5. Extract the list of technician with the required skills
   {if there is a technician with the required skills go to step 6
   else change zone and go to step 2}
6. Select a technician with the most suitable skills
7. Verify the current technician engagement
   {if the technician is available go to step 8
   else go to step 5 and update the list deleting the not selectable technician}
8. Verify the work shift
   {if the technician is available go to step 9
   else go to step 5 and update the list deleting the not selectable technician}
9. Verify the actual position
   {if the technician is in the suitable geographic range go to step 10
   else go to step 5 and update the list deleting the not selectable technician}
10. Assign the priority to the technician

The abovementioned rules will be refined, adding a weight for each instance of each filter. Moreover in the third year of HoliDes Project, the Decision Algorithm will be improved by adding a learning capability, in order to make the algorithm able to assess the reason of task reassignment by the operator or the reason of the rejection of the technician in order to adapt the weight of the instances of each filter, learning new data at each transition and refining in this way its effectiveness.

The information of the system has been included in a database (a relational database has been designed and implemented as a SQL server). Its overall structure is shown in Fig. 9.

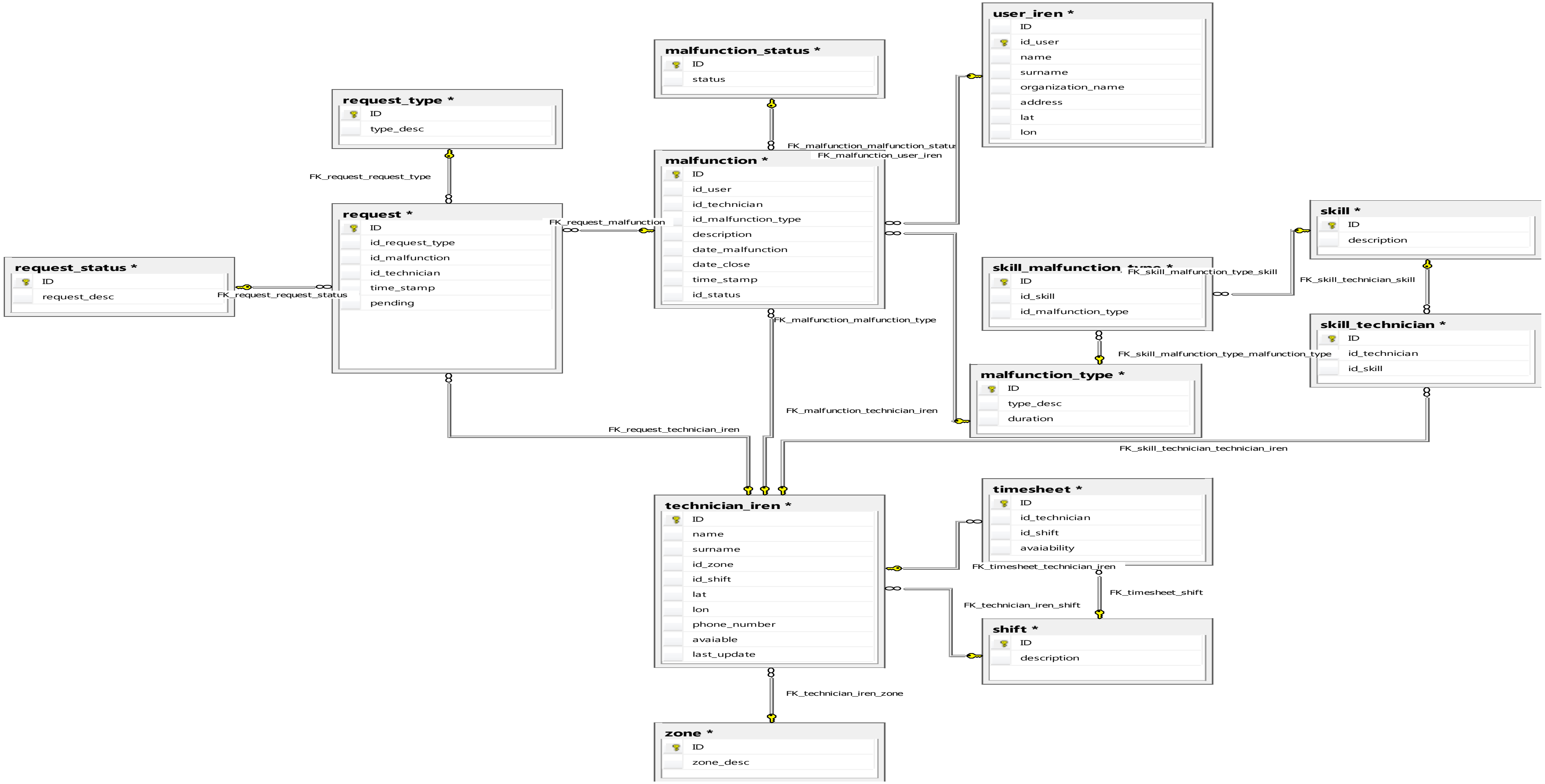

Fig. 9 - ER DB structure of the Control Room AdCoS

As regards to the HMI application for the operators, a web application has been implemented.
An additional HMI must be designed by taking into consideration that the operators of the Control Room must monitor heterogeneous information distributed on (at least) 3 screens by using different applications and operation systems (mainly windows and unix). Therefore we designed a web-based interface with the following characteristics:

- It occupies a narrow zone in the screen, preferably on the right of the main screen.
- It can be opened in a browser to be easily accessible on different devices (different hardware and software systems).

The HMI provides the operator with the following information:

- Malfunction date
- Technician name (selected according to the Decision Algorithms that is the core of the system)
- Status of the malfunction
- Close date

It also includes an OK button to assign the operator to the intervention (because the operators explicitly requested to have the final decision on assigning the technician, and do not make this operation completely automatic).

The Control Room operator can see the malfunction list split according the malfunction status. The statuses of the malfunctions are:

- **Pending**: if the task is waiting for an assignment to a technician
- **Ongoing**: if the task has been assigned to a technician and he accepted it
- **Closed**: if the technician finished the intervention and communicates it.

As described in the sequence diagram in Fig. 8, at step 4 the algorithm is triggered and the task is assigned if the technician accepts it.

However, if the technician rejects the task the malfunction is still in the “pending” list in the operator, so the AdCoS has to trigger again the Decision Algorithm. For this reason a fourth malfunction status has been created: “rejected”. This status is just a machine status: the HMI will still display "rejected" malfunctions as "pending", that is the most suitable semantic category for the human operator, while the machine syntax will distinguish between "rejected" and "pending".

Once a malfunction is rejected, the algorithm is retriggered, and once a new technician is selected, the malfunction status turns "pending", because the system is waiting again for the technician's answer.
This logic has been modelled as the Finite State Machine (FSM) described in Fig. 10. The HMI corresponding to this FSM is shown in Fig. 11, Fig. 12 and Fig. 13.

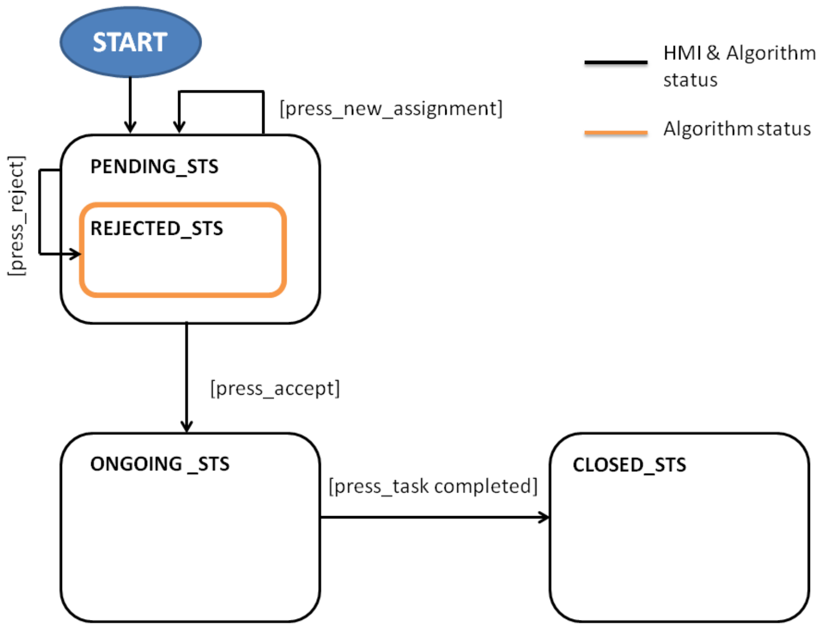

Fig. 10 - Finite State Machine for the management of the malfunction status

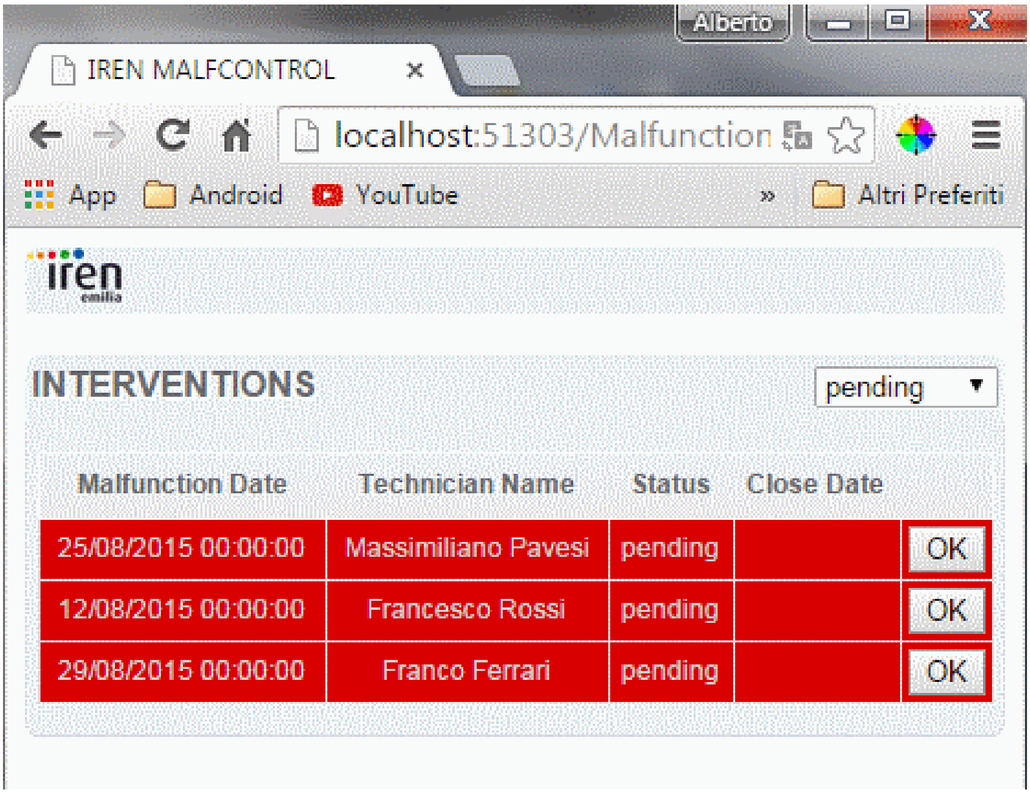

Fig. 11 - HMI for the Control Room operator: pending list of intervention

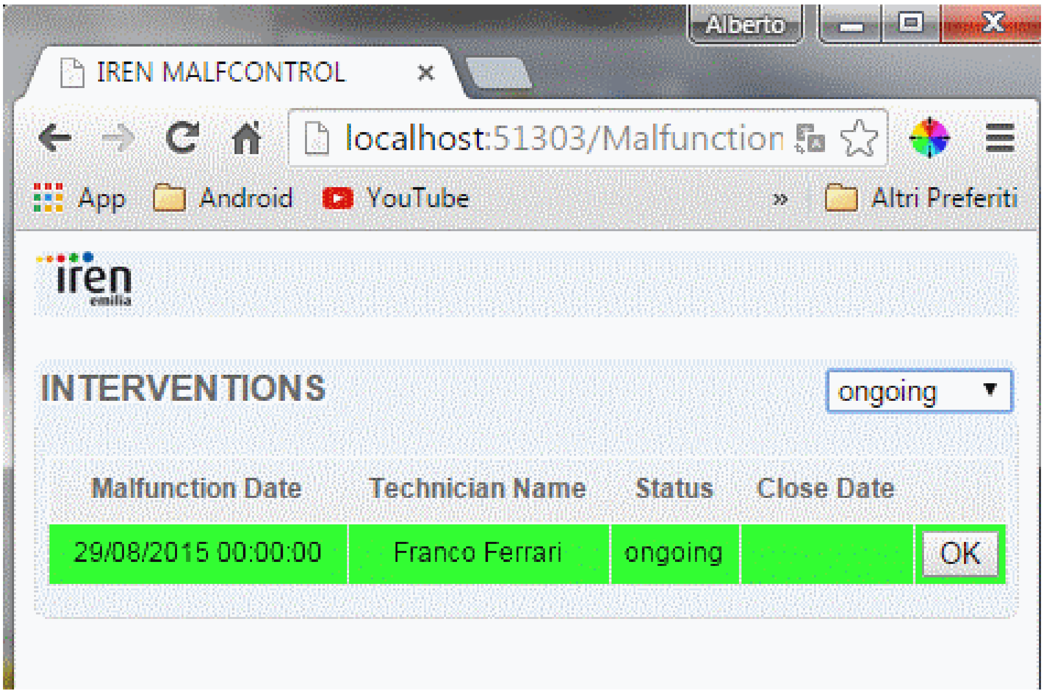

Fig. 12 - HMI for the Control Room operator: assigned and ongoing interventions

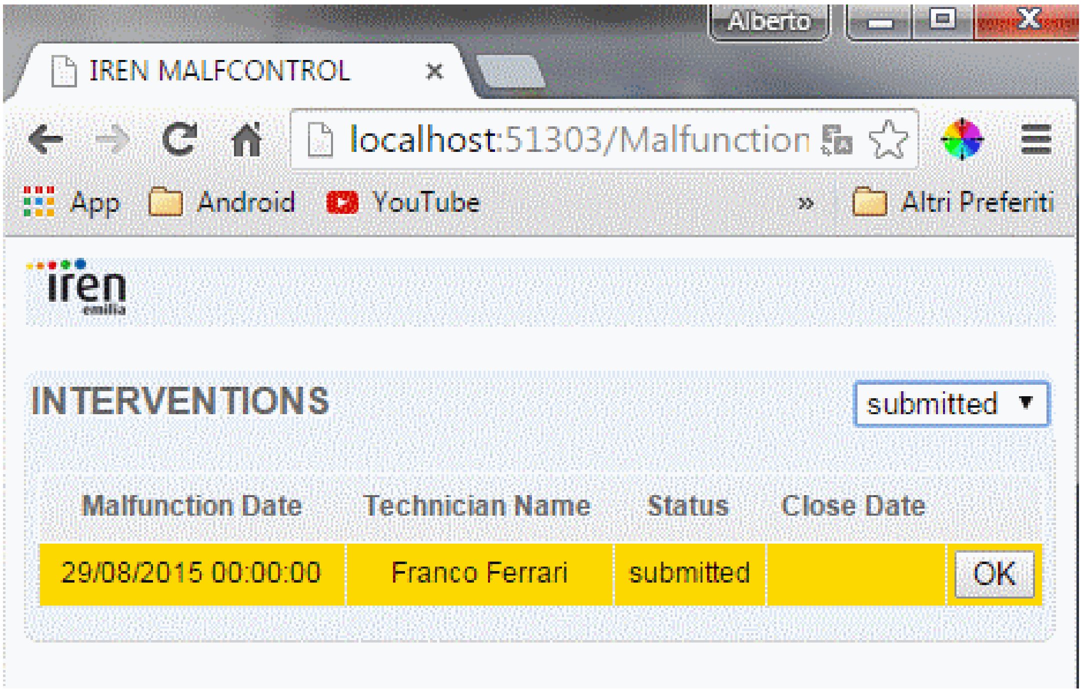

Fig. 13 - HMI for the Control Room operator: assigned and closed interventions

Finally, the app for the technicians has been implemented (a screenshot is provided in Fig. 14) to show the interventions assigned to each technician, and provides them with the corresponding information (details of the malfunctions, address of the interventions, etc.).

Fig. 14 – Screenshot of the HMI for the technician

3 HF-RTP Instance for WP8

3.1 Tailoring Rules

Tailoring rules are intended to a guide of best practises to tailoring the tools found in the HF-RTP for use in your development environment.
So far, no tailoring rules have explicitly been made available from WP1, only activities. As such, a short step of tailoring rules by WP8 has been identified.

Tailoring Step 1: Problem Analysis
Determine the Human Factors knowledge gap in your design process. I.e. perhaps a proper task analysis is missing from your current design process? A proper assessment is needed into what knowledge you can already extract from your current design process. For example do you have a models and documentation you can pull data from?

Tailoring Step 2: The Platform Builder
Consult the Human Factors Platform builder to identify tools and techniques which could assist you paying particular attention to MTTs which are already tailored to your design process. If for example, MTT "xyz" is already tailored to your petri-net modelling tool, then unless you need more capability, further tailoring is not necessary.

Where your chosen tools are not already tailored to products in your design process, look for tools which are currently OSLC compliant since this will make the integration easier.

Tailoring Step 3: The information to exchange
Assuming that your chosen tool is not already tailored to your design process you need to plan what information you wish to exchange in order to solve the problem in Step1. At this stage you should have identified which tool is 'consuming' the information and which tool is 'providing' the information.
Let's say for example that you have an analysis tool which is used to collect user error statistic from an empirical test. The results from this might need to feed into software HMI design tool in order demonstrate traceability between the faults found during testing and the resulting architecture. In this case, the analysis tool could be thought of as the provider and the HMI development tool would be the consumer.

Tailoring Step 4: Creation of the adapters

At most, two software adapters would need to be made in order to exchange the information. This is a more complicated step since it requires software development. The main things to mention here are:

- The information which is exchanged should conform to the HoliDes meta model seen in Figure 2.
- The exchange mechanism to used shall conform to the OSLC principles. i.e REST based using the linked data approach.

3.2 EA and DOORS

Part of the HF-RTP instance will comprise DOORS and Enterprise Architect (EA). An OSLC adapter for EA has been developed by AGI UK which a user of EA to access the requirements of particular project using standard web technologies.

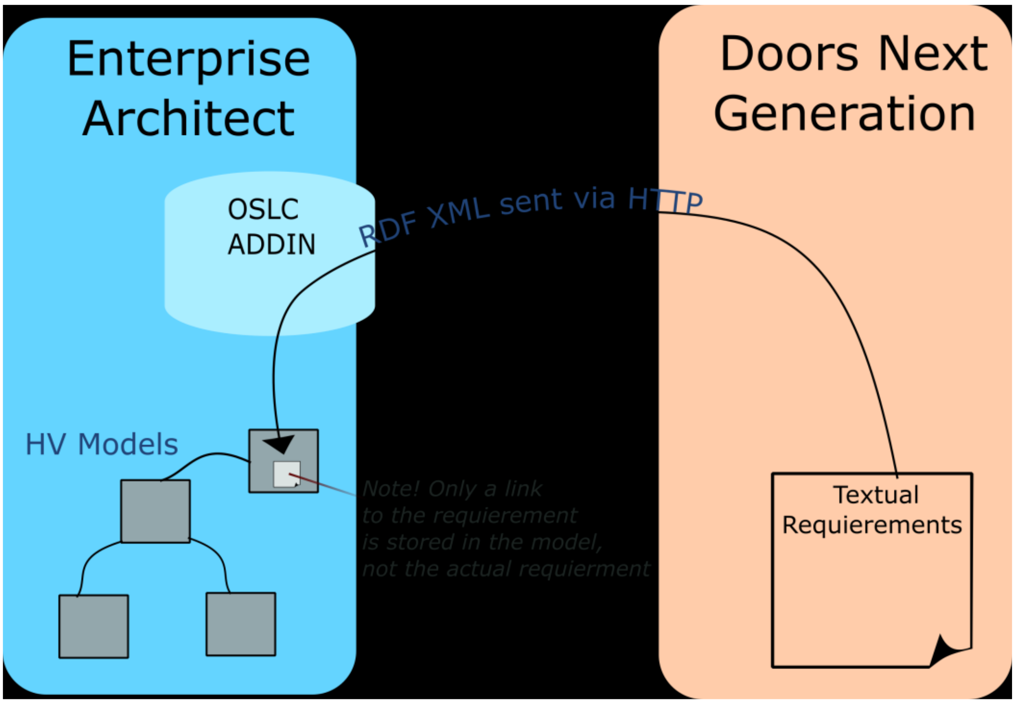

Fig. 15 - EA and DOORS sharing requirements

3.3 EA and the Human Views

Whilst developing the Human Views in EA is not strictly an example of two tools being integrated in the typical sense, it is an important part of the HF-RTP and worthy of mention. For a detail on the Human Views, please refer to D8.4 - Modelled and Model-based Analysis of the Control Room AdCoS and HF-RTP Requirements Definition Update. *Section 5.1.1.8 Human View Products.*

Enterprise architect comes bundled with many modelling frameworks such as UPDM, DODAF, UML, SysML, TOGAF and Archimate.

These Frameworks are housed in EA under a technology known as MDG; a proprietary technology of SPARX. It is possible, using EA for one to develop 3rd party MDGs for the creation of customised models. This allows the functionality of EA to be extended. In D8.5, the Human View models were described using standard UML models. This isn't ideal since there is no control over what can be drawn. A far better solution is to develop an MDG which can describe the models as listed in the NAF Human Views handbook.

<u>Components of MDGs in for Human Views</u>

In order to define the Human Views in EA as an MDG, the following MDG Components need to be in place:

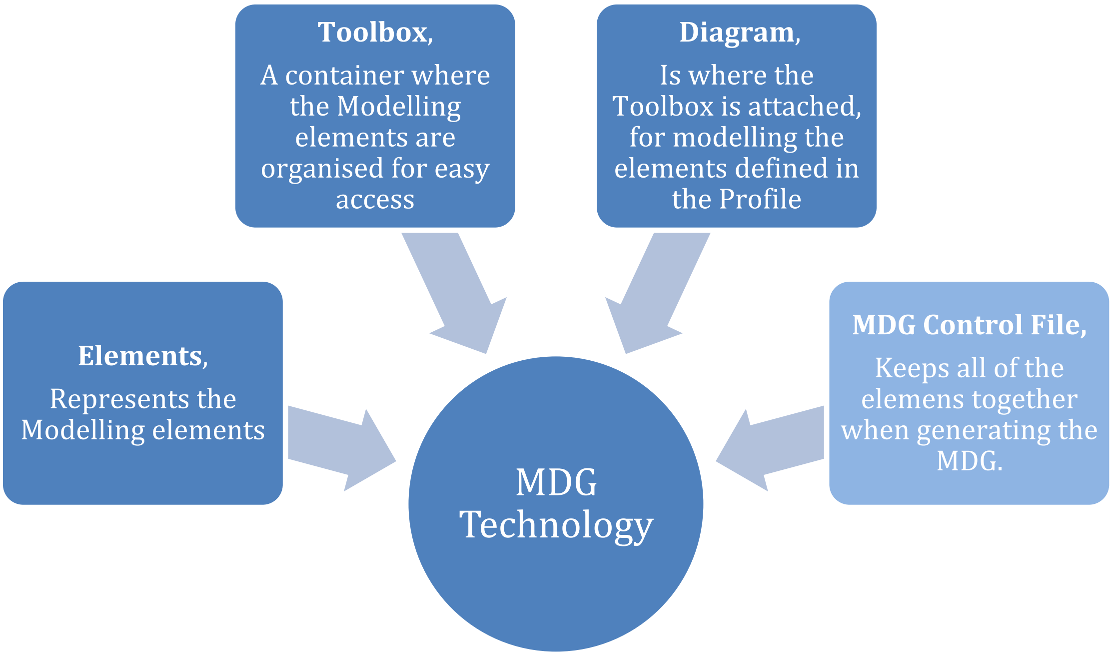

Fig. 16 - Components of an MDG technology

All parts of the MDG technology except the MDG control file are modelling in EA itself. Fig. 17 shows the process in EA for developing an MDG. The profile, (circled in red) is where the stereotypes that correspond to the Human Views will be defined.

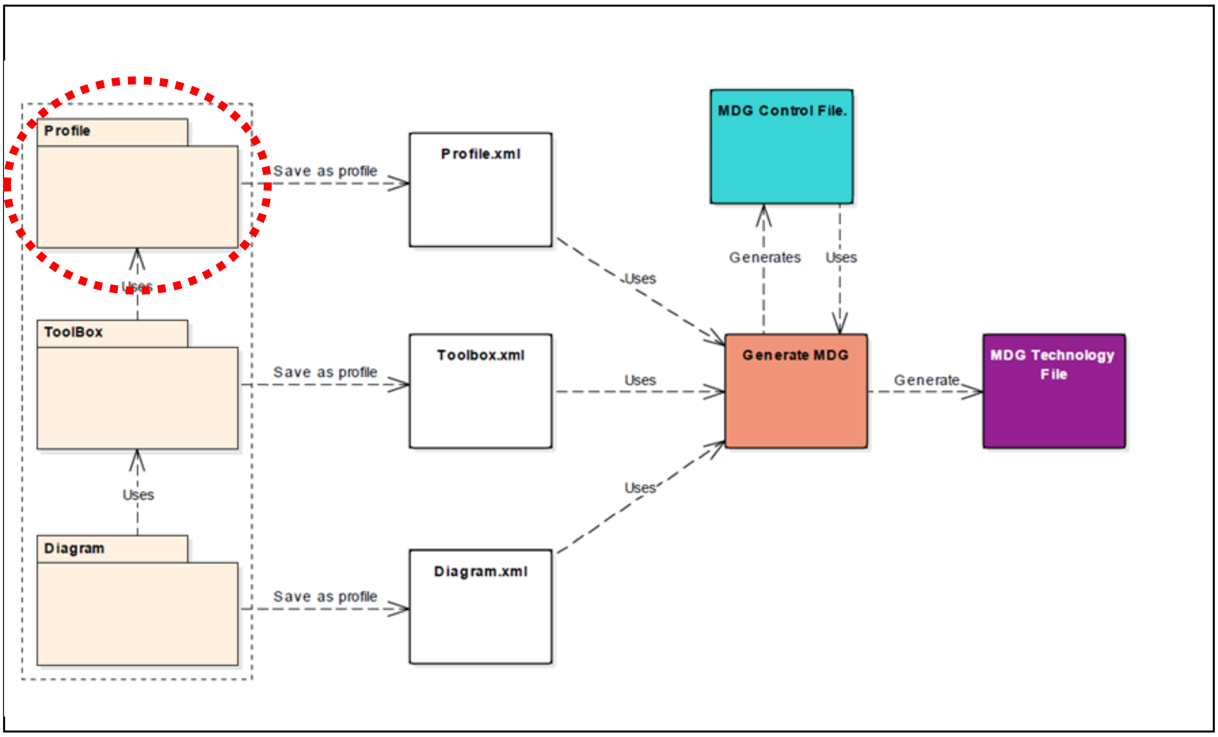

Fig. 17 - The MDG Creation Process

3.4 EA and the HF Filer

The HF Filer tool is an application for managing evaluation plans of a product or service.

Each plan would comprise a number of topics which would be answered by a Human Factors expert. By using the HF Filer tool, responses from HF-Experts would be logged in such a way that they could be accessed and used by other tools in the tool chain.

This accessibility will be based on OSLC principles and mapped to the properties of the OSLC Quality management domain.

The green area on the left of Fig. 18 shows the external project data in the existing tool chain. For the Command and Control Room this would be held as Human View models in Enterprise Architect.

The orange area represents the Evaluation plan. An Evaluation plan can involve a number of techniques, including such things as procedure evaluations (complexity, number of steps, nomenclature, etc.; reach analysis via computer modelling; time-line analysis for assessing task demands and workload; or other methods, depending on the issue being considered.)

The red area represents the actual AdCoS evaluation taking place. The plan is followed and the Human Factors expert completes the plan and enters the results into a web based interface.

Finally, the blue area on the right of Fig. 18 shows the Evaluation report. This represents the captured information from the evaluation. It is in three layers, items, reports and evaluation text. The report is a container for all of the evaluations which take place for a given plan. The item evaluation contains the results of a particular part of the evaluation and finally the textual data which makes up the evaluation item are stored as "Item Evaluation Text".

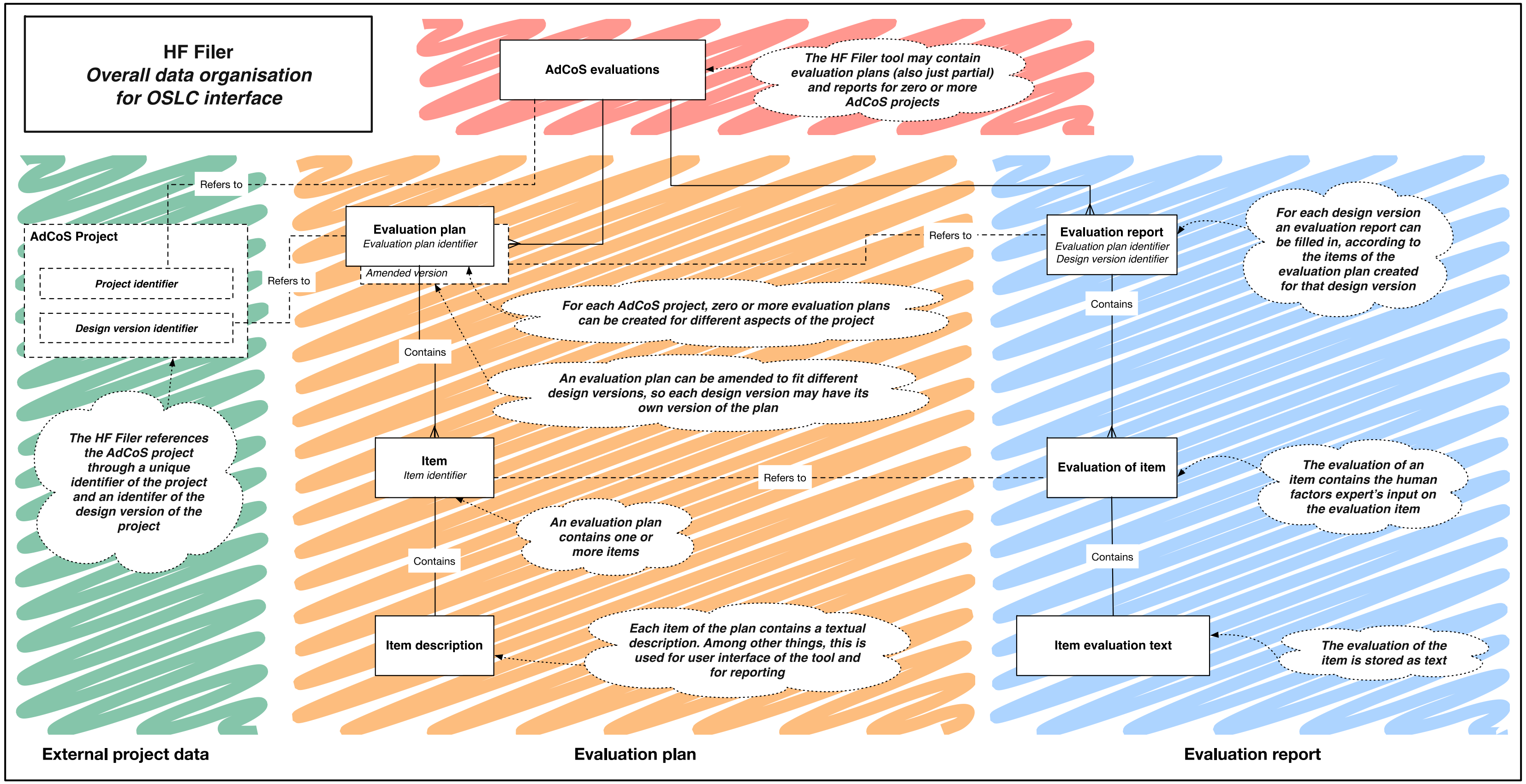

Fig. 18 - The internal Data structure of the HF Filer

Fig. 19 shows a table with the attributes available for each evaluation. Once completed, the information based on this meta data would be available for consumption by other tools in the tool chain. In the case of the Command and Control Room it would be Enterprise Architect.

Identifier	Data type	Description
Content	TextArea	User inputted response.
Created	DateTime	Moment the response was initially created.
Description	TextArea	'Question' asked to user.
design_version_identifier	Tag	Version ID of evaluation package.
evaluation_identifier	Tag	Evaluation plan identifier.
evaluation_name	String	Evaluation plan name/title.
Id	Integer	Response identifier.
last_modified	DateTime	Moment the response was last modified.
project_identifier	Tag	Project identifier.
project_name	String	Project name/title.
Reporter	String	Person that generated this response.

Fig. 19 – HF Filer Response Attributes

The comments from WP8 so are regarding the Meta Model in Fig. 18.

- Currently, the reports which result from the AdCoS evaluations are not explicitly linked to resources in the project tool chain. Currently it is only referred to through the evaluation plans. The thinking in WP8 is that an engineer who wishes to view the evaluation report from the context of a Human View would probably not be concerned as to the evaluation which was performed on an area of the system but more what the results of that evaluation were.

- Currently, the Meta Model only links to a project data at project level. That is, the entire results of the evaluation are linked to the entire project. WP8 believe that this is too much information at too high a level. A more practical and useable solution would be to access individual evaluation items and link them to the modelling components on which they are based. For example, if an evaluation plan is executed on the efficacy of the eye tracking, then the results of that evaluation should be made easily available to the engineer who is modelling the eye tracking component in Enterprise Architect. How this would work is show in Fig. 20.

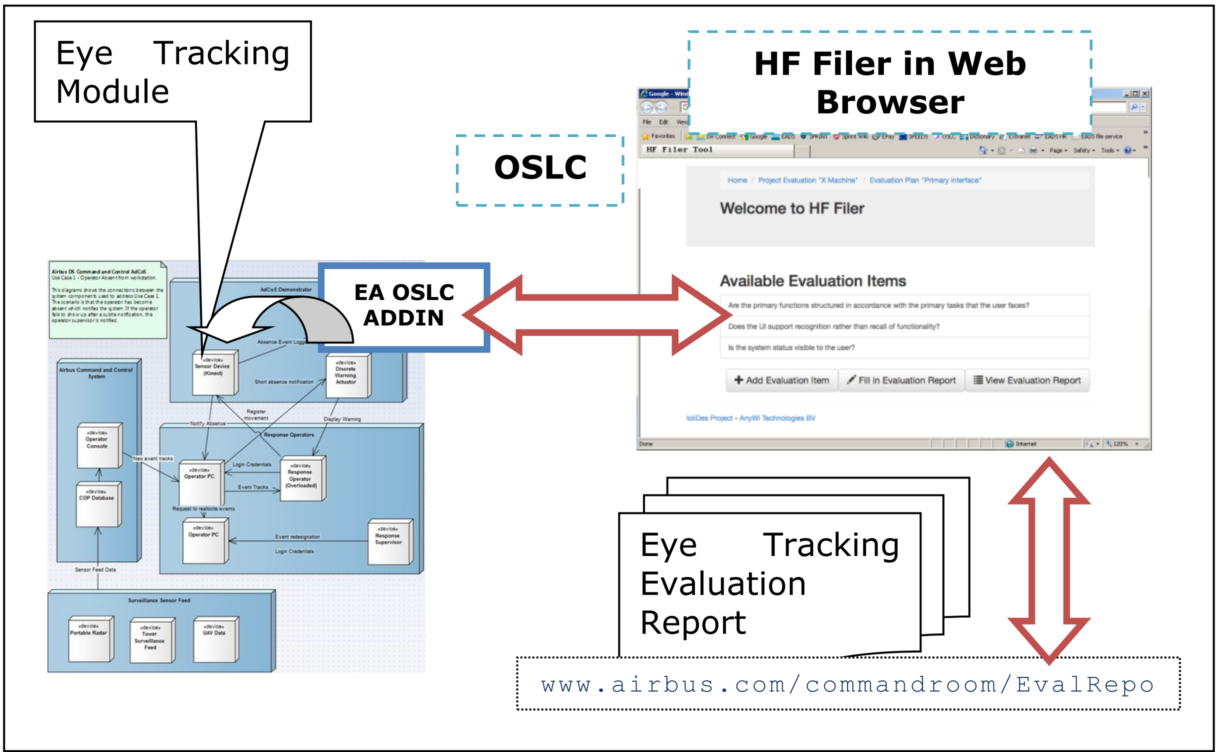

Fig. 20 - The HF Filer interacting with EA

After the evaluation has been completed, a report would be published online using the HF Filer. An evaluation report for the eye tracking module would have an online presence and therefore a URI. Using the OSLC Add-in, the URI of the report would be embedded inside a model element of a Human Views model, thus storing the traceability.

At a later date, an engineer wishing to view the evaluation report from the model would do so from EA. The OSLC services available from inside the modelling tool would bring up the report to the engineer through a web browser.

3.5 GreatSPN

GreatSPN is a modelling tool for the definition, animation and verification of Petri nets. Petri Nets are a formalism of discrete event dynamic systems and suited to describing concurrency in systems where a workflow is involved.

GreatSPN seeks to answer questions as:

- How many machines do I have idle in my system?
- What is the distribution of resources in my workflow?
- Is my system deadlock free?
- Are there times all of my machines are busy/free?

GreatSPN has been identified as a modelling tool that could be used to model a Command and Control workstation for the purpose of identifying such things as:

- What is the effect on my system if the operator becomes absent?
- How many tracks would be missed if a response operator is not available?
- How many operators would I need to deal with an expected quantity of incoming tracks?

NB – tracks is a command and control room term for entities which have been detected and require the presence to be monitored.

The Petri net model

Fig. 21 shows the so GreatSPN model so for the process analysis of tracks in the Command and Control Room. The events that take place are described in *Table 2*.

Table 2 - Steps in Petri net model

Step No	Description
1	As tracks are detected they move from the state of 'undetected' to 'detected' where they wait to be assigned to an operator.
2a	As tracks are assigned, the pool of working operators is depleted by 1.
2b	When there are no tracks waiting to be worked on, a tedium probability is encountered which causes operators to be bored and they reduce the Working operator pool. An operator will not become bored in a situation where tracks are waiting to be assigned. A bored operator is in effect unavailable for work since they are assumed to be absent from their work place.
3	If a track is determined to be irrelevant, the operator is returned the available operator pool.
4	Should a problematic track be discovered, it's then passed onto the response team. This allows the surveillance operator to be returned to the resource pool but consumes a response operator.
5	The role of the response operator is to assign guard to investigate the problematic track and keep the guard up to date with developments through observation.
6	During investigation, a guard resource is consumed. When the investigation is complete, both the guard and response operator are returned to the resource pool. To complete the petri net model, the track is 'returned' to the track pool though in reality, a collection of suspicious entities is not maintained!

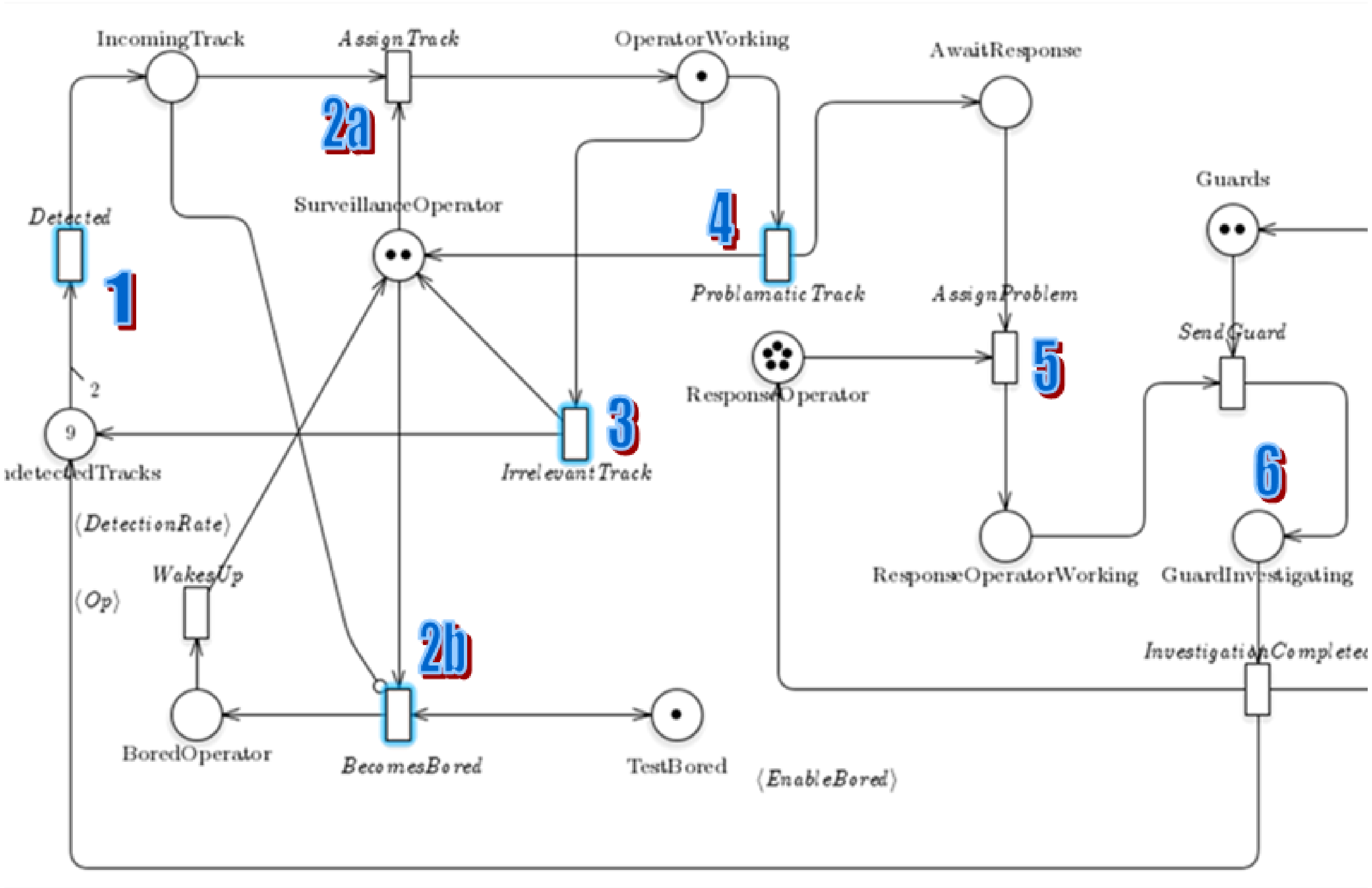

Fig. 21 - A Command and Control Room modelled in GreatSPN

3.6 KNIME

KNIME is a Java based open platform for data analytics developed by the University of Konstanz and is released under the General Public License (GPL), version 3. KNIME includes several components for data-transformation, data-processing, data-analysing, data-exploring and data-visualization (see Fig. 22).

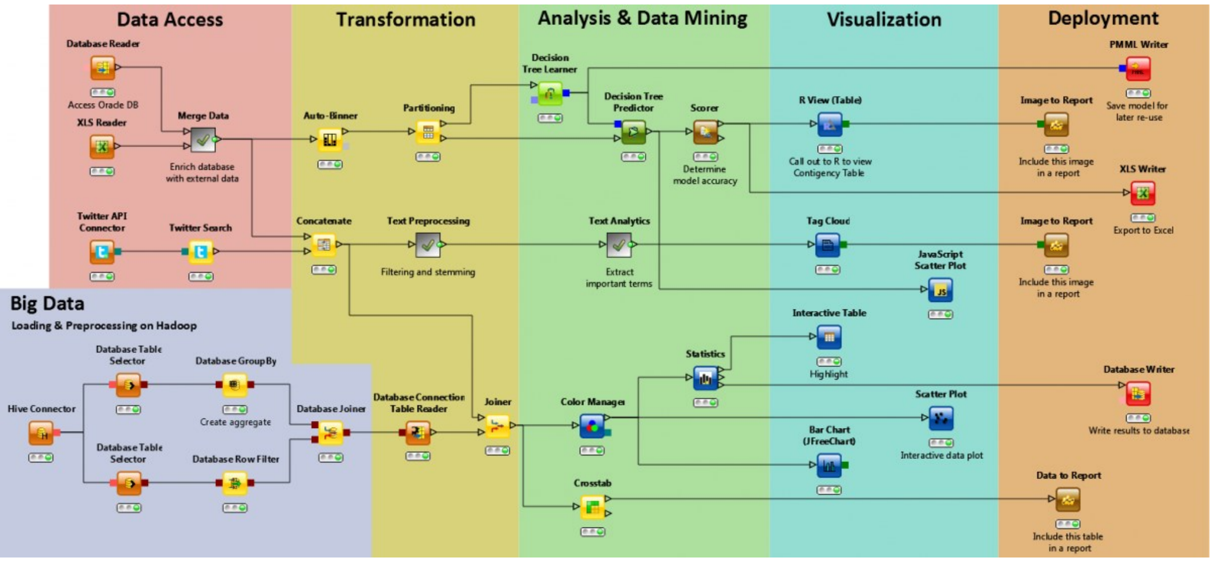

Fig. 22 - KNIME sample nodes

For the purpose to develop a KNIME workflow (model) it is recommended to use the KNIME Analytics Platform. The user interface (see Fig. 23) allows you to select from a large set of routines, called nodes, to develop a data-driven workflow to investigate incoming data for identifying potential design patterns and detecting unknown anomalies within the data.

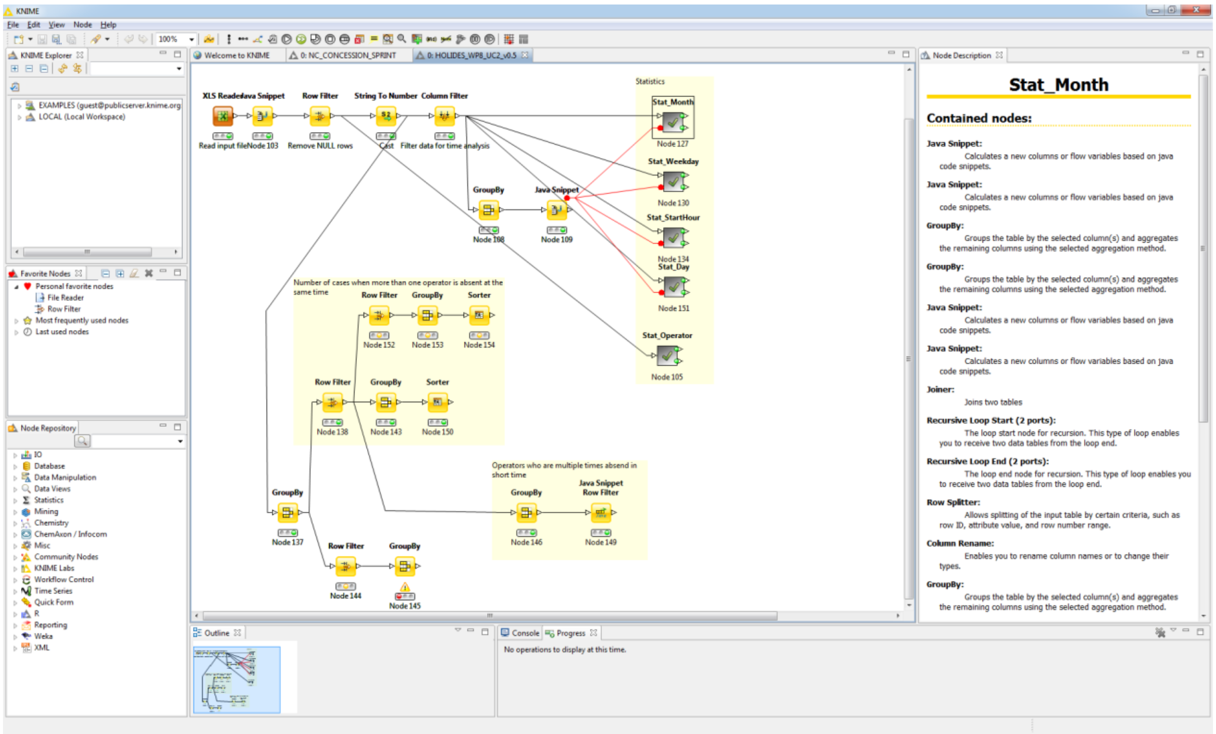

Fig. 23 - KNIME Analytics Platform

KNIME Analytics Platform offers multiple opportunities to manage data sources. In addition to the import of data from spreadsheets in excel or csv format it is possible to plug-in databases or web services. Also the outcome from a workflow could be managed in several forms. Results could be stored in text files or databases. Furthermore it is also an option to generate reports as word or as pdf file. Such reports could include graphs to visualize the outcomes of the workflow in a proper way.

For the execution of KNIME workflows there are two different opportunities. The first option would be to run the workflow within KNIME Analytics Platform itself. This option is quite useful during the development phase because the workflow would be traceable for the user and also allows looking at the output of each single node in the workflow. The second option would be to run the workflow in a batch execution node. Once the workflow is fixed there is no further need of the KNIME Analytics Platform. Furthermore this approach allows integrating the workflow execution in existing ICT processes.

4 Conclusion and Summary

This document has taken inputs from WP1 and WP2 to assist in the building development of HF-RTP instances for the Command and Control and Energy Control Rooms. WP1 provided the Platform builder for the identification and selection of MTTs available from the HF-RTP. Some usability issues were encountered and not all of the MTTs used by the WP8 were available but this should be easily rectified.

WP2 provided a Meta Model which eventually will be used to define the concepts that will be exchanged between MTTs in WP8. However, some key concepts are missing and so currently, the implementations so far demonstrated will not be exactly compliant with the Meta Model.

This document then described the latest status of the AdCoS' in WP8. In the Command and Control Room, interactive sensor technologies are being used to improve the efficacy of operatives by responding to their fatigue, attention and workload. In the Energy Control Room, improvements are being made in the area of decision making through the use of algorithm improvements and modelling.

These improvements have been made possible because of the MTTs available. These MTTs are what form the HF-RTP instances for use AdCoS and are detailed in chapter 3.

5 Way forward and upcoming activities

5.1 Command and Control Room

HF-RTP

With regards to the RTP work, the Command and Control Room will focus on the following:

- Integration of Human Factors Analysis between Enterprise Architect and the HF Filer tool. (With the aim of it being a reference implementation for the rest of the project.)
- Implementation of the Human Views in Enterprise Architect using MDG technologies.

AdCoS

The next milestone in the development of the AdCoS demonstrator is the implementation of the version to be presented at the external review event in November in the Netherlands. That version will cover Use Cases 1-5.

In 2016, the work on the implementation of the AdCoS demonstrator will continue, focussing inter alia on Use Case 6 and any changes suggested by the reviewer comments.

The second important focus of the work will be the evaluation activities of the AdCoS. This work will be closely co-ordinated with the partners in WP 5 and AIW."

Glossary

ACC = Adaptive Cruise Control
ADAS = Advanced Driving Assistance Systems
AdCoS = Adaptive Cooperative Human-Machine Systems
Archimate = Modelling Tool
DAS = Driving Assistance Systems
DODAF = Department of Defence Architecture Framework
DOORS = Dynamic Object Oriented Requirements System (Requirements management tool by IBM)
EV = Ego Vehicle
FCW(S) = Forward Collision Warning (System)
GreatSPN = Graphical Editor and Analyzer for Timed and Stochastic Petri Nets (Petri Net modelling tool)
HEE = Human Efficiency evaluator
HF = Human Factors
HF-RTP = Human Factors Reference Technology Platform
HMI = Human Machine Interaction
HMS = Human Machine Systems
HoliDes = Holistic Human Factors Design of Adaptive Cooperative Human-Machine Systems
IRN – Iren Emilia S.p.A
KNIME = Java based open platform for data analytics
LEA = Pattern recognition software
MOVIDA = Monitoring of Visual Distraction and risks Assessment
MTT = Method Tools and technology
NAF = NATO Architecture Framework.
OSLC = Open Services for Lifecycle Collaboration
PADAS = Partially Autonomous Driving Assistance Systems
RTP = Reference Technology Platform
SPARX = Software development company which make Enterprise Architect.
SysML = UML profile for systems engineering
SNV = Universita Degi Studi Suor Orsola Benincasa
TOGAF = The Open Group Architecture Framework
UC = Use Cases
UML = Unified Modelling Language
UPDM = Unified Profile for DODAF
WP = Work Package

6 Appendixes

HOLIDES_Task Analysis_v2.docx
This task analysis takes into account the work done by the operators once the user request has been categorized and switched by means the Genesis software.