OSLC User Group Scenario Template

Project Number:	332933
Classification:	Public
Work Package(s):	WP1
Milestone:	
Document Version:	V0.2
Issue Date:	16.07.2015
Document Timescale:	Project Start Date: October 1, 2013
Authors:	Ian Giblett,

RECORD OF REVISION		
Date (DD.MM.JJ)	Status Description	Author
18.05.2016	Draft structure of template	Ian Giblett
16.06.2016	First two scenarios added	Ian Giblett
7.7.2016	Third EDA scenario added	Ian Giblett/ Sara Sillaurren

Table of Contents

1 Introduction

This document provides scenarios for the OSLC human factors working group. It is intended that these scenarios could be used to drive a fully-fledged OSLC specification at a later date.

There are 3 scenarios to date.

1. Usability Engineering	A scenario based around a process of assessing the usability of a system.
2. HF Design Traceability	A scenario based around linking evaluation data to an architecture framework for capturing the reasons behind architectural decisions.
3. Experimental Data Archiving	A scenario based around linking requirements and user needs to an executable model for testing and deployment of the finished system.

2 Scenario 1 – Usability Engineering

2.1 Business context

There is a need in many engineering domains to focus on usability for the end user. This could help less experienced operators make fewer mistakes.
User errors in many industries have become a particular concern since the potential for disaster is huge and yet many interfaces are designed in a non-intuitive way and are difficult to learn and use.
Examples might be the use of medical equipment, heavy plant machinery, automotive vehicles and factory control systems.
The aims of Usability engineering intended to reach better usability which will minimise use errors and associated risks. Traceability is a crucial part of demonstrating that the usability engineering process has been applied.

2.2 Proposed solution

The solution would comprise 3 tools and technologies which cover the areas of talk analysis and modelling, usability data collection and filing of evaluation results.

The toolchain consists of three tools, which range from task analysis and modelling, over usability data collection to the filing of the usability evaluation results. It allows the creation of task models in a graphical format and transfers them to the table-based data collection tool U-DAT through a common intermediate file format.

The usability experts will then perform a usability evaluation based on the data collected in the U-DAT tool and file the evaluation results in the dedicated tool HF Filer.

HF Filer is structured around the concept of evaluation plans, which in turn provide the structure for the evaluation reports. The initial evaluation plan can be created automatically in HF Filer from the same intermediate files used to create the forms in U-DAT.

It should be noted that the step from the data gathered in U-DAT to the evaluation report filed in HF Filer a manual one, as it involves the usability experts analysing the test results and providing a professional evaluation.

The complete toolchain is illustrated in Figure 1.

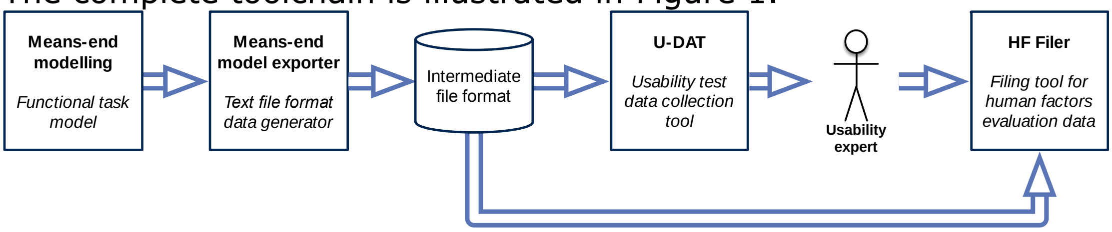

Figure 1: Chain of lifecycle tools for usability evaluations of a Usability Engineering example

2.3 Proposed results

The ultimate result for this scenario would be a design process that allows a usability practitioner to enter link usability data tightly to the relevant engineering artefacts.

The results would be a robust format that is easily shared between different tools. The file format would be XML and conform to the datatypes in Table 1.

Table 1 - Data types for the XML format

Property	identifier	Type	Attr./ elem.		Constraints
System map (XML root element)					
ID	ID	String	Attribute	required	xs:id
Name	name	String	Element	[1..1]	1 to 40 chars
Task map					
ID	ID	String	Attribute	required	xs:id
Name	name	String	Element	[1..1]	1 to 40 chars
Context of use	contetOfUse	String	Element	[1..1]	Free text
User					
ID	ID	String	Attribute	required	xs:id
Role name	name	String	Element	[1..1]	1 to 40 chars
Characteristics		String	Element		Free text
Need					
ID	ID	String	Attribute	required	xs:id
Name	name	String	Element	[1..1]	1 to 40 chars
Need	need	String	Element	[1..1]	1 to 256 chars
Description	description	String	Element		Free text
Category	category	String	Element	[0..1]	1 to 40 chars
Relation					
ID	ID	String	Attribute	required	xs:id
Type	type	String	Attribute		Enumeration: include, specialization, optional
Parent	parent	String	Element	[1..1]	xs:id
Child	child	String	Element	[1..1]	xs:id
Execution Order	executionOrder	Number	Element	[0..1]	>= 0
Task					
ID	ID	String	Attribute	required	xs:id
Name	name	String	Element	[1..1]	1 to 40 chars
User goal	userGoal	String	Element	[1..1]	1 to 80 chars
Description	description	String	Element		Free text
Step					

Property	identifier	Type	Attr./ elem.		Constraints
ID	ID	String	Attribute	required	xs:id
Name	name	String	Element	[1..1]	1 to 40 chars
Description	description	String	Element		Free text

3 Scenario 2 Decision Traceability for Model Based System Engineering.

3.1 Business context

Considering that the lifecycle from conception to retirement of modern systems can be many years, there is a clear need to capture decision making process behind architectural choices.
Currently, human factors work, if done at all, is regularly documented as unstructured data in many files. At best, these large documents are referenced in an architectural component. If lucky, the modeller will have a page number to the reasoning behind an architectural decision in the document. Often that information is not available and sometimes the document isn't even to hand. Locating that document and then finding the relevant part of the document is very time consuming.

3.2 Proposed solution

The proposed solution incorporates a modelling and some means for of storing human factor evaluative data.
The requirements that also drive the architecture come requirements. Since requirements management already exist for OSLC, they won't be discussed any further here.

Architectural Modelling
In this solution, the modelling tool will be Sparx Enterprise Architect. Enterprise Architect is architectural tool used widely across industry. It supports Architectural Frameworks such as DODAF, TOGAF and NAF. The purpose of such models is to create an architecture that matches the goals of a system to the physical assets and infrastructure that supports those goals.

HF Filer

This tool is intended to store human factors evaluation reports in a structured manner. Evaluation Items and Evaluation reports can be held on the HF Filer with each item of information being held under a separate URL.

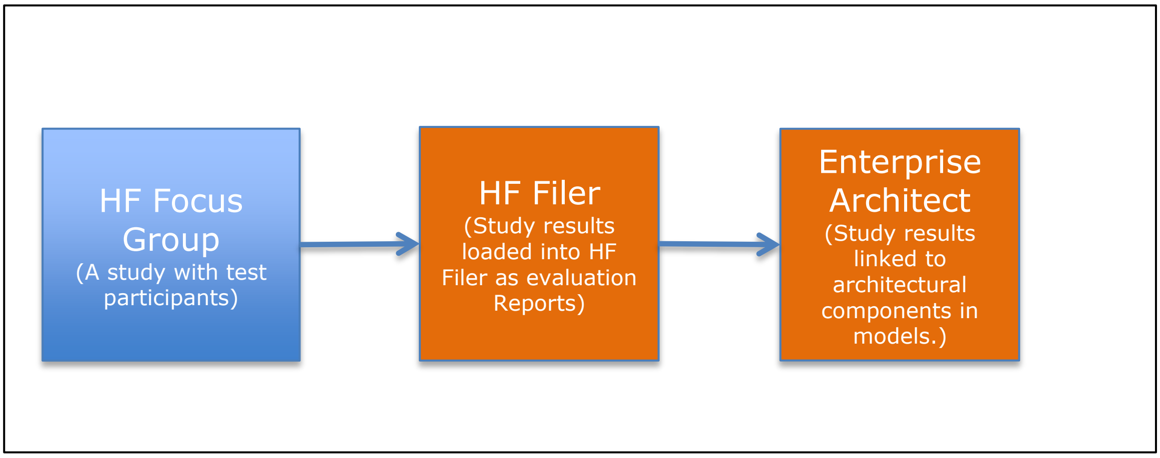

Figure 2 - MBSE and HF Traceability

3.3 Proposed results

The results of this scenario are a design process that can incorporate human work into the architectural model of a system in a more stream linked process.
The OSLC interface of a human factors service provider such as the HF Filer would allow the evaluation items to be made available to the modeller in Enterprise Architect.
The ability to store pertinent information about an architectural decision which has come from Human Factors studies will save time in future when one wishes to re-discover the direction in which a design decision has been made. Furthermore, this will demonstrate traceability between a system implementation and findings from HF studies has been maintained.

4 Scenario 3 Experimental Data Archiving

4.1 Business context

EDA is a centralized archive for HF data acquired in various development phases of a project. EDA stores acquired data together with meta-information that links the data to the project structure. The projects structure comprehends the phases executed within each project such as requirements, design, implementation or evaluation. Within a phase there are a number of data collection sessions which is the lowest level of meta-information hierarchy.

EDA is provided by Tecnalia Research & Innovation (TEC) based on the requirements defined by Honeywell (HON). EDA tool is a document management tool designed and developed in Python/Django to allow the definition and storage of multiple projects including their full data structure through a web interface. It provides web services. EDA is a centralized archive for HF data acquired in various development phases of a project. EDA stores acquired data together with meta-information that links the data to the project structure.

There are a number of activities where data from subjects is collected. Depending on the phase of the project, data is used for definition of a concept (voice of a customer) and requirements as well as for validation of prototypes.

The tool is designed to store project structures and the data generated when confronting these projects with subjects in a centralised archive.
The initial work prepared requirements to design the system. Here are some example requirements.

· **EDA should provide interface to record events in the database. Adding events to the database should be possible during online data storage as well as during playback.**
· **User should be able to find and list all data tables that contain keywords (projects, data collection sessions, observations, etc.) with respect to selected keywords.**
· **User should be able to find and list data records based on defined conditions of subject demography.**

4.2 Proposed solution

Based on the requirements, the tool would be designed, developed and integrated with RTMaps.

This EDA tool has been designed for the aeronautic application domain to keep all the relevant information about experiments set previously in RTMaps.

A data collection session will contain a number of data records, being one of the four basic methods. There are variants to the basic methods, from the point of view of data storage, the following generalisations are used:

- **Observation**: activity where the user is observed in his work environment while doing tasks that should be addressed by the new system. The observer makes notes while observing the user and the user can comment on his actions. A camera or a voice recorder can be used to support data analysis.

- **Interview**: a guided process where the subjects are presented a set of open ended questions and they freely elaborate each of them. The questions are forged to direct the subjects to talk about specific aspects of a problem

- **Questionnaire**: allows subjects to select defined options to the presented questions. As in case of interview, the questions are created with an intention to get feedback on specific aspects of a problem. However, the knowledge of the problem is deep enough to state options for answers.

- **Experiment**: activity when a subject works out his task in more or less realistic environment. The task is selected to address the problem being studied and usually it is a dynamic process, during which a number of parameters is recorded in time.

For results, there are recordings of specific parameters. Many of them create a time series of parameter's value (such as ECG record or simulator log). Additionally video and voice can be recorded and notes taken by the experimenter can be compiled as a list of events
(annotations). The recordings are linked to subjects and then placed in the context of a project following the data model of EDA.

EDA is publicly available for testing and evaluation purposes from TEC servers after a user's validation.
The tool defines a set of public methods to be consumed as web services along with an FTP to allow integration with other tools. It has been integrated with RTMaps to allow the archiving of the files generated during an experiment within a project.

The interfaces

The EDA tool uses as input description data and meta-data about various projects and their HF related data collected in experiments, interviews etc. The tool provides a web interface to
specify project data and rest based communication to simulation platform (such as RTMaps) to automatically collect experiment meta-data.
To define a project different general data should be entered:

· Description (plain text)
· Start date / End Date (date type)
· Team leader, HF Focal & SW leader (plain text)
· Description (plain text)

Each project executes a number of phases, which information kept is the category (plain text), and with different values as requirements, design, implementation and evaluation. And within a phase there is a number of data collection sessions, which are associated to a project and a
phase, and as data required have:

· Start (date)
· End (date)
· Observations (structure)
· Interviews (structure)
· Questionnaires (structure)
· Experiments (structure)

Each session has a start and end date and contains two files – data collection plan describing what and how it will be done and report/summary file describing results of analysis of the acquired data. It also requires:

· Location where the session is being performed
· Responsible of the collection
· Responsible of the analysis

Other information stored in this EDA tool is about the subject doing the experiments. In this case, the stored information will be:

· Email (text with format email)
· Name (plain text)
· Last name (plain text)
· Year of birth (date)
· Gender (female/male)
· Rank (plain text)
· Flight hours (number)
· Aircraft flown most often (plain text)

4.3 Proposed results

First of all the user must authenticate himself in order to use the EDA tool. There has been developed a permissions management module, each user will only see the public projects and the ones assigned to him via users groups.
Navigating from the left panel menu, Figure 6, the user can access the different modules that compose EDA. The main categories are:

- *Projects*: Projects defined in the tool, each of them with its structure of phases, sessions and methods. These subcategories can be accessed from within a project (seeing only the ones contained in the current project) or from direct links to all Phases or Sessions.

- *Methods*: The methods link allows the direct access to all defined methods belonging
to any project. Within each of these methods (Observations, Interviews, *Questionnaires* and Experiments) the user can access and modify its properties and see all the subjects that have participated,

- *Subjects*: Access to all participant subjects in any method, and within each subject, access to all the method in which he/she has participated,

- *Configuration*: Configuration of keywords, phases, categories, locations, event categories, users of the EDA tool (not participants) and their groups.

EDA would be connected to RTMaps to improve the process of data acquisition and archiving. The tool chain assures correct links between meta-data and real data, it supports synchronization
between various data sources and provides robust data structure.

The flow chart in Figure 3 describes the execution of an experiment. The steps involved are the following:

· User selects a task and request its execution through EDA
· EDA communicates via socket with the master instance of RTMaps
· The master instance of RTMaps starts the experiment and communicates with other potential slaves instances of RTMaps.
· Optionally the user could finish the experiment from EDA. If so EDA communicates via socket with the master instance of RTMaps.
· Finally, once the experimet is finished, the master instance of RTMaps sends back the metadata of the generated files and the events that took place during the experiment to EDA.

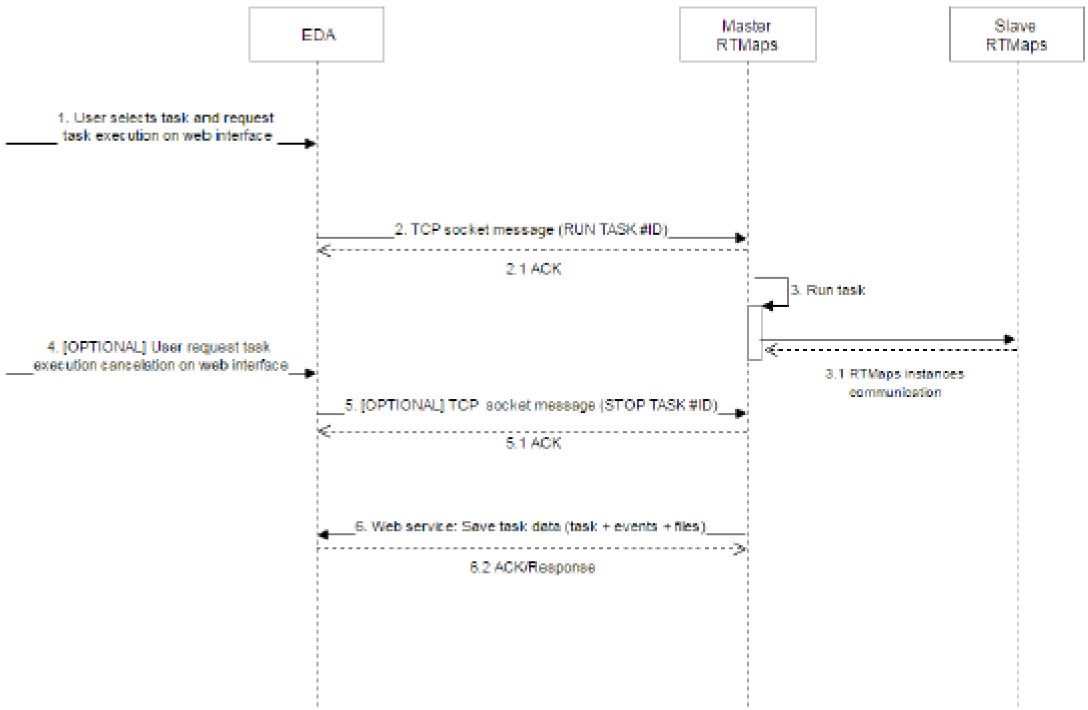

Figure 3 - Communication between EDA and RT-MAPS

Benefits

1. Data collection is simplified allowing HF experts focus on expert work
2. Managed data with a simple web interface allows for data sharing within a team and among teams
3. Flexible meta-information allows keeping relations among various types of information
4. Look-up times for information related to evaluations are reduced, especially in project hand-over or when working with certification agencies, i.e. providing data recorded in past.