Project Deliverable

Project Number: 604068	**Project Acronym:** MORE	**Project Title:** Real-time Monitoring and Optimization of Resource Efficiency in Integrated Processing Plants

Instrument: COLLABORATIVE PROJECT	**Thematic Priority** NMP

Title D4.1 Requirement specification for the integrated deployment platform

Due Date: Month 6	**Actual Submission Date:** Month 6

Start date of project: November 1st , 2013	**Duration:** 36 months

Organisation name of lead contractor for this deliverable: LeiKon	**Document version:** V 1.0.1

Dissemination level (Project co-funded by the European Commission within the Seventh Framework Programme)

PU	Public	**X**
PP	Restricted to other programme participants (including the Commission)	
RE	Restricted to a group defined by the consortium (including the Commission)	
CO	Confidential, only for members of the consortium (including the Commission)	

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 604068.

Abstract :

The objective of the task 4.1 is to develop an integrated deployment platform to build resource efficiency monitoring and decision support applications. The platform should offer a reliably real-time runtime environment of resource efficiency indicators based on computationally intense algorithms as e.g. multi-criteria optimization. It should also provide the benefit of model based software engineering in the area of industrial solutions for process control and supervision. The purpose of this report is to describe the organization and the initial results of the requirement specification for the integrated deployment platform.

Authors (organisations) : LeiKon, UVA, INEOS in Köln, Petronor

Validated by: INEOS

Keywords : Integrated deployment platform, Engineering Environment, real time runtime environment

Disclaimer

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Any liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No license, express or implied, by estoppels or otherwise, to any intellectual property rights are granted herein. The members of the project MORE do not accept any liability for actions or omissions of MORE members or third parties and disclaims any obligation to enforce the use of this document. This document is subject to change without notice.

Revision History

The following table describes the main changes done in the document since it was created.

Revision	Date	Description	Author (Organisation)
V0.5	2014-02-13	Creation	LeiKon
V0.9	2014-04-10	Revision	LeiKon and task partner INEOS
V1.0	2014-04-22	Completion of HMI Requirements	LeiKon
V1.0.1	2014-04-30	Final Version	LeiKon

Table of Content

List of Figures

List of Tables

1. Introduction

This report includes the requirement specification for the integrated deployment platform. In chapter 2 the process to find and to discuss the requirements is described. Some requirements could be derived from the REI definitions made in D1.1. Especially the specific indicators which are related to measurements of material flows or energy flows imply needs for specific data models and runtime capabilities like data coupling to measurements in DCS or PIMS. Most of the requirements were collected and analyzed based on a web based requirement specification management tool. To describe the requirement specification for the integrated deployment platform the partners of T4.1 also looked for similar tools like energy management systems (concerned with ISO 50001) and commercially available plant information systems.

The requirements for the integrated deployment platform were classified into the following groups:

- requirements for the runtime/calculation environment
- requirements for the engineering environment to specify REI calculations and their assignment to specific plant/site units or to specific product families
- requirements for the human machine interface (HMI) and reporting facilities.

This requirement specification (D4.1) will be modified continuously during the project. The web based requirement specification management tool will also be used to manage and supervise the progress.

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 604068.

2. Management of the requirement specification phase

2.1 Overview

Based on the first results the deliveries of work package 1 (D1.1: "Analysis of the state of the art of REIs" and especially D.1.2: "Definition of indicators of integrated plants") requirements for the integrated deployment platform can be derived. This will be done by reviewing the deliveries with focus on the data models needed and the expected complexity to calculate indicators.

The requirements derived from the first deliveries of Work package 1 were analyzed and prepared by the partners of Task 4.1 and were discussed with all partners by document exchanges and communication by phone.

2.2 Requirements specification process

In order to find a suitable way to collect and weight requirements for the usage of the integrated deployment platform an interactive web based Requirement Specification Platform was implemented. The advantages of using such an interactive way to collect requirements are:

- Requirements can be collected in a systematic manner.
- Each requirement can be classified in one of predefined requirement classes. That helps to structure the requirement list, to organize the development and later on to test and validate the results.
- During the whole development process it is clear who initiated a specific requirement. This helps to clarify open questions directly between development responsible and the user.
- Additional information like notices, web links, drawings or documents can be linked to each individual requirement.
- Developers can give feedbacks such as remarks or questions either directly to the initiator of a request or to the member of the platform at all. Misunderstandings are transparent and can be solved in early phases of a development process.
- The collection of requirements can be expanded during the whole project phase in order to react on modified constraints.

As interactive, web based requirement specification tool LeiKon chose the open source platform for bug and ticket tracking "Mantis". This tool including the GUI was customized in order to manage the requirement specification process. All partners were given an introduction and lecture explaining how to use the tool. In addition a short manual was written by LeiKon including the most important aspects to use the tool.

The state chart in Figure 1 illustrates the requirement specification process provided by "Mantis". The process is initialized by reporting a new idea, which is assigned to the state "NEW". A developer has to document the realization of requirement and to change the requirement state to "PROVIDED". If a reporter does not agree on realization of the requirement provided by a developer, a reporter can enhance own requirement description, which resets the requirement state to "NEW" and reinitializes specification process. These steps can be repeated until both the reporter and the developer agree on technical realization. Developer is allowed only to initiate a state transition and to change the state of a requirement.

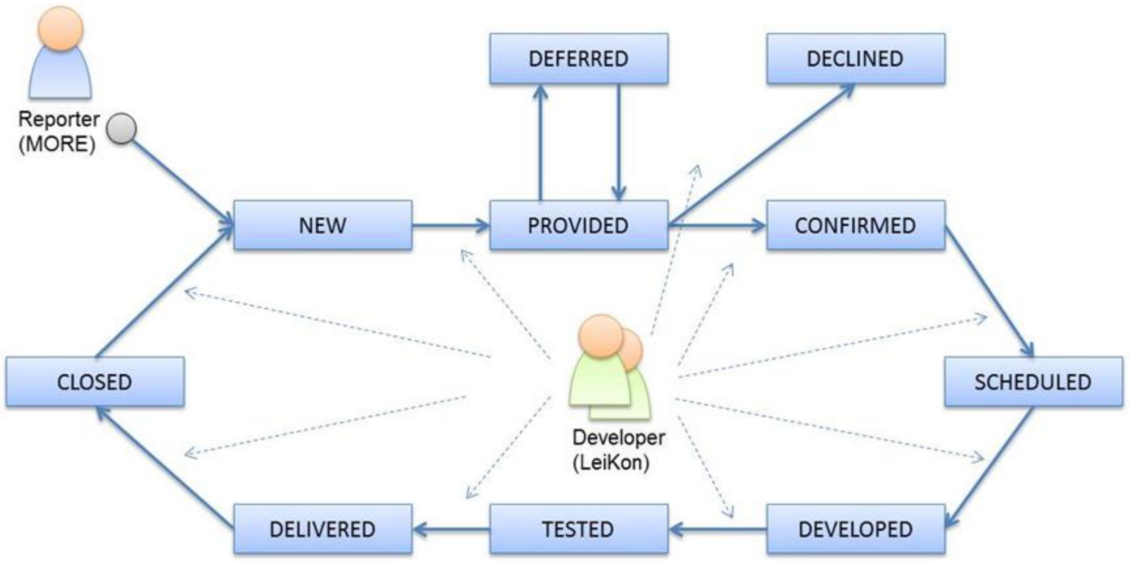

Figure 1: Requirement specification process managed by "Mantis"

An overview of available states is provided in the Table 1.

Table 1: Requirement states

State	Description
NEW (NEW)	A reporter posted a new idea or a feature request. A developer has to review the issue and provide documentation for the execution, development efforts etc.
PROVIDED (PRO)	The requirement was documented and given to the members of T4.1, who will make a decision whether it will be used now or in the next release.
DEFERRED (DEF)	The committee switched the requirement to the next release.
DECLINED (DEC)	The committee declined the requirement.
CONFIRMED (CON)	The committee checked and accepted the requirement.
SCHEDULED (SCH)	The requirement is ready and planned for developing.
DEVELOPED (DEV)	The requirement is developed and incorporated.
TESTED (TST)	The requirement is tested.
DELIVERED (DEL)	The system is in production or delivered to customer.
CLOSED (CLS)	The requirement is archived for possibly later use.

The following chapter summarizes the most important requirements which were given by the partners of MORE.

3. Requirements for HMI

R.1.1 An interactive web-based user interface for REI monitoring and reporting. With web-based applications, users access the system via a uniform environment—the web browser. Unlike the desktop application, a web-based application is available anytime and anywhere via a PC with a TCP/IP-Connection to the Webserver. It noticeably reduces efforts of management and deployment too, since the web-services are provided by a single web application.

R.1.2 Host-independent web-application code. The web-application should be deployable on a Windows and on a Unix/Linux Webserver System.

R.1.3 Configurable HMI. The HMI should be configurable in WYSIWYG[1]-fashion using different Layout-Strategies and UI-Components such as Button, Edit Field or Diagrams.

R.1.4 Extendible HMI. The HMI should also provide an extension interface for integration of new UI-components. Based on a SOA[2] of HMI-Runtime new UI-Components should be provided as modules (bundles) and integrated into the HMI at the runtime.

R.1.5 User-/Access-Management and Authorization. HMI should provide basic services for User-/Access-Management and Authorization. There should be a configurable access restriction for all elements of the HMI. A default management console within the HMI should provide standard UI-Interface for configuration of authentication and authorization services.

R.1.6 Reporting services. HMI should provide a reporting service (engine) which can process customized report-templates. The model of a report template consists of parameterized queries for structured data sources, data transformation logic and a presentation (layout) configuration. All reports within the HMI should be generated on-demand by real-time data or archived data. The IDE should provide a WYSIWYG-Editor for design of the report templates.

[1] What You See Is What You Get

[2] Service-oriented Architecture

4. Requirements for the runtime environment

R.2.1 Stand-alone calculation/runtime environment. The runtime environment is stand-alone application running in a user-mode or as an OS-service.

R.2.2 Multi-platform. The runtime environment should be a multi-platform application, which is able to run on Windows and Unix/Linux OS.

R.2.3 Design of the REI calculation based on functional block diagram. Functional block diagram is a common approach for system description. A system function is represented by a block containing input and output parameters. A connection between parameters represents the relationship between functions. A functional sequence (network) is a sequence of interconnected functions. Single functions within a functional sequence should be triggered periodically by an adjustable clock pulse. The order of triggered functions should be configurable by the user.

R.2.4 Runtime Monitoring. The runtime environment should provide a common service for the runtime monitoring of single functions and functional sequences in order to guarantee a deterministic execution of a REI calculation.

R.2.5 A function should be composed by a functional sequence (functional decomposition).

R.2.6 A function (routine) should be implemented by a scripting language e.g. Lua, Python etc. or by a high level programming language e.g. C/C++.

R.2.7 A function defined by a specific interface (ports) and a specific routine should be assigned as function type. Several function types can be grouped into a type library.

R.2.8 Functional extension. A dynamic, functional extension of the runtime environment should be provided by loading (deploying) of one or more type libraries into the runtime.

R.2.9 Integration with the state-of-the-art calculation and modeling tools e.g. “MATLAB®” or the open source alternative “Scilab”. Calculation model designed by the modeling tools should be executed within a functional sequence.

R.2.10 Communication interfaces to the HMI and to the data sources like DCS, PIMS, LIMS, File and Database systems. Communication driver (type) libraries should provide special function types for interfacing the communication system of different data sources.

R.2.10.1 OPC-DA

R.2.10.2 OPC-HDA

R.2.10.3 OSI-PI API

R.2.10.4 CSV-Files

R.2.11 Management and Deployment. The runtime environment should provide an exclusive communication interface and high priority services for management and deploying purposes. This ensures that user always is able to stop/start the execution of REI calculation and to load/update single function types or type libraries into the runtime.

This project has received funding from the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement no 604068.

5. Requirements for the engineering environment

R.3.1 Single integrated development environment should provide user assistance and the services for engineering and deployment of the REI calculation and HMI (incl. report-templates).

R.3.2 Multi-Platform. The IDE should be running on multiple computer platforms (Windows/Unix/Linux).

R.3.3 Project management. All configurations and calculation designs should be managed within a project by the virtual folder/document hierarchy. In order to resume the development process the projects including configurations, designs and current development states should be persisted to local/network file storage.

R.3.4 Help system. The IDE should provide an extensible help system for accessing common knowledge base and/or context sensitive topics.

R.3.5 Graphical design of REI calculations based on functional block diagrams. The IDE should provide a graphical editor for the functional block diagram. Each diagram should be represented as a node in project tree. Available function types should be provided by a function pallet.

R.3.6 Repository for the management of the function types. Function types are grouped into type libraries. A common UI-Interface should provide a list of available type libraries, function types and brief summary of the function. A user should be able to filter the list using the name of the function or using the keyword within function type description.

R.3.7 Formal description of the site/plant topology including resource and information flow. A user should be able to model a site/plant topology, material, resource and information flows as well as the mass balance, which will be used for the REI calculation. Furthermore these model elements should be assignable to a REI calculation.