Project information

Project title	Coordinating Optimisation of Complex Industrial Processes
Project acronym	COCOP
Project call	H2020-SPIRE-2016
Grant number	723661
Project duration	1.10.2016-31.3.2020 (42 months)

Document information

Deliverable number	D2.3
Deliverable title	System Requirements Specifications
Version	1.0
Dissemination level	Public
Work package	WP2
Authors	IDENER, TUT, VTT, OUTOTEC, OPTIMATION, TECNALIA, SIDENOR, TUDO
Contributing partners	IDENER, TUT, VTT, OUTOTEC, OPTIMATION, TECNALIA, SIDENOR, TUDO
Delivery date	29/09/2017
Planned delivery month	M12
Keywords	System, requirements, specifications, use cases, SRS

This project has received European Union's Horizon 2020 research and innovation funding under grant agreement No 723661.

VERSION HISTORY

Version	Description	Organisation	Date
0.1	Initial version	IDENER	15/03/17
0.2	Initial contributions	IDENER	15/05/17
0.3	New TOC	IDENER	05/09/17
0.4	New content	IDENER	07/09/17
0.5	Partner contributions	ALL	20/09/17
0.6	Further contriburions	ALL	22/09/17
0.7	Version for review	IDENER	25/09/17
0.8	Reviewed version	VTT, 2Control	26/09/17
1.0	Final version	IDENER, TUT	28/09/17

EXECUTIVE SUMMARY

The current (public) document includes a report of the design activities being performed in the framework of WP2 towards the definition of the scope and targets of the implementation to be done within the COCOP project. The title of this document is Deliverable *D2.3 System Requirements Specification* and is produced in the first year of the project schedule, specifically within the activities of WP2 and the associated task *T2.2 System requirements specifications*. SIDENOR, OUTOTEC, TECNALIA, Technical University of Dortmund, Tampere University of Technology, VTT, OPTIMATION and IDENER have contributed to this deliverable whose main responsible partners is IDENER as T2.2 main responsible and WP2 lead beneficiary. Background for this deliverable are documents, visits and workshops at the steel (Sidenor) and the copper (BOHA) factory, as well as the extensive meetings held between the project partners with the objective of defining the overall requirements and scope of the concept to be demonstrated within the COCOP project and accordingly, of the developments to be performed during the project execution. During the activities performed in the first months of the project which led to the production of Deliverable D2.1, extensive work was also done in order to start defining the requirements of the project main outputs. The information contained in this deliverable establishes thus the requirements that should be fulfilled in order to ensure that the required system that enables the demonstration of the COCOP concept with the two main pilot cases is properly implemented. This document presents a snapshot of this information, this is, of the actual set of use cases and requirements for the system. Nonetheless, the information herein contained will be continuously updated during the execution of the project in order to accompany the different developments achieved and to reflect the new discoveries found during this work. Some of the information present in this deliverable has been gathered from previous (private) deliverables and is repeated here in order to enable the readers of this document to properly understand it.

ABBREVIATIONS

Abbreviation	Full name
AC	Anode Casting
AF	Anode Furnace
BOHA	Boliden Harjavalta, the copper pilot process
CC	Continuous Casting
EAF	Electric Arc Furnace
FMU	Functional Mock-up Unit
FSF	Flash Smelting Furnace
HR	Hot Rolling
KPI	Key Performance Indicator
LCA	Life Cycle Analysis
ODE	Ordinary differential equation
OPC	Open Platform Communication (OLE for Process Control)
PDE	Partial differential equation
PSC	Pierce-Smith Converter
SM	Secondary Metallurgy

Software tools used in this project

JIRA: is an issue tracking product, developed by Atlassian. It provides bug tracking, issue tracking, and project management functions

Confluence: is a team collaboration software also developed by Atlassian. It provides several functionalities for the teams for collaborating in project managing and execution as on-line collaborative documents and shared calendars.

TABLE OF CONTENTS

1 Introduction 1

2 Methodology description 2

2.1 The COCOP approach 2

2.2 Use cases vs pilot cases 3

2.3 Use cases description 3

2.4 System Requirements 4

2.5 Abbreviations for use cases and requirements 6

2.6 Confidentiality in this document 8

2.7 Continuous Update of Information 8

2.8 Implementation of the backlog - Continuation of the COCOP work 9

3 COCOP System Requirements Specifications 10

3.1 Generic (non-pilot specific) aspects 10
3.1.1 Use cases 10
3.1.2 Requirements 12

3.2 Copper pilot case 16
3.2.1 Process short description 16
3.2.2 Main actors description 18
3.2.3 Models to be used 18
3.2.4 Defined use cases 19
3.2.5 Requirements 21

3.3 Steel pilot case 23
3.3.1 Process short description 23
3.3.2 Main actors description 24
3.3.3 Models to be used 25
3.3.4 Defined use cases 26
3.3.5 Requirements 28

4 Conclusions 30

TABLE OF CONTENTS

1 Introduction

The idea behind the COCOP project is that the European process-industry could be optimally run by operators under the guidance of a coordinating, real-time optimisation system. In order to reinforce such affirmation, the concept will be demonstrated within the project through the design of a model-based, predictive, coordinating optimisation conceptual system and its application to two real pilot cases. The underlying technical objective is to define, design and implement a concept that integrates existing industrial control systems with efficient data management and optimisation methods, which provides means to monitor and control large industrial production processes. This plant-wide monitoring and control require computationally intensive data analysis and large-scale optimisation, which are also enabled by the proposed platform. Additionally, the social objective behind this idea is to improve operator integration with the overall production process by increasing its plant-wide awareness. This improved awareness will allow the operator to understand the potential impact his control actions will produce in the overall production of the plant, allowing him to make better-supported decisions and consequently, reducing his mental workload. Lastly but yet, of utter importance, the COCOP project takes into account the comfortability of these operators with the idea of integrating such system in their plants. In the last years, the industries have frequently failed to bring into production many great technical developments, because the operators' feedback have not been considered in their design.

The novelty of the concept, especially considering these social aspects, made it necessary to establish a proper schedule for the demonstration of this idea. Accordingly, the most important group of project tasks is the definition of the scope and objectives of the actions to be taken. These tasks are covered and reported in WP2 and its associated deliverables. This deliverable constitutes a summary of the information retrieved not only within task T2.2, but also from all the activities performed in the first year of the project.

The present document summarises all this information providing it in a way that is useful for the later development of the activities to be executed within the project. The document begins with the "Methodology description" section in which the COCOP approach to organising the work regarding these activities is explained. Next, the "COCOP System Requirements Specifications" section summarises the main part of the knowledge generated within this first year, categorised as use cases and associated requirements. This information is split into three groups: the general information, which is applicable for the overall platform; and the copper and steel pilot case related ones. The document ends with some conclusions that provide a brief insight on how this document and the associated information will be used from now on, to accompany the project development activities.

2 Methodology description

2.1 The COCOP approach

In order to ensure the proper realisation of a technological idea, especially when its implementation relies on a multi-disciplinary team from different countries and areas of knowledge, it is of key importance to establish a proper methodology for the definition, planning and implementation of the elemental technical activities that are required. The European Material Modelling Council, a European cluster of stakeholders related to the material modelling area, remarks the problem with the research activities and projects for this in its 2015 Roadmap:

> *"It takes 10 to 15 years to move academic software to marketable software. There is hence a need to produce more industry ready software by academic modellers and to stimulate the transfer of academic software to industry".*

Although the sentence there specifically mentions the academic software, the idea behind it is that the conceptual developments produced in high scientific environments (not only academia but also very specialised companies and EU research projects) are frequently difficult to put into the market by the proper agents (i.e. big software and control companies). One of the reasons behind this is the differences between the implementation methodology between software companies and research projects. In the COCOP initiative, in order to boost the further development of the concepts analysed and researched in the project, the consortium has agreed to establish proper rules for the development activities. These rules are documented in the project handbook maintained by the project coordinator, which provides guidelines on how to proceed with different tasks. Inside these guidelines, indications on how to proceed with the definition, classification and description of the development activities has been included. Specifically, it has been decided to execute and coordinate the technical work following a software development-like methodology. This methodology has been adapted to the specific needs of the project but yet maintains the essential elements of a software implementation project, which will help in the future to enable the further development of the platform, either by the project members or by an external party interested on commercialising the researched technology.

The next sub-sections provide a summary of some of the main elements of this methodology. Before entering into detail of each of the elements described here, it is worth mentioning that as stated in other deliverables, two solutions from Atlassian has been selected to be used during the project for technical development organisation and tracking. Specifically, Confluence is used to handle the documentation and JIRA for keeping track of the associated requirements and tasks.

2.2 Use cases vs pilot cases

"Use case" is a commonly used term that can have several meanings. In software and systems engineering, a use case is a list of actions or event steps, typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system, to achieve a goal. Alistair Cockburn (*) provided the following definition of use case:

> *A use case captures a contract between the stakeholders of a system about its behavior. The use case describes the system's behavior under various conditions as it responds to a request from one of the stakeholders, called the primary actor. The primary actor initiates an interaction with the system to accomplish some goal. The system responds, protecting the interests of all the stakeholders. Different sequences of behavior, or scenarios, can unfold, depending on the particular requests made and conditions surrounding the requests. The use case collects together those different scenarios. Use cases are fundamentally a text form, although they can be written using flow charts, sequence charts, Petri nets, or programming languages. Under normal circumstances, they serve to communicate from one person to another, often to people with no special training. Simple text is, therefore, usually the best choice.* [1]

As introduced in Deliverable *D2.1 Use case definition document*, the COCOP consortium decided to make a clear distinction between the term "use case", which will be used within the project to refer to the software related use cases, and the term "pilot case", which will be used for referring to any of the two piloting activities that the project includes: "Steel and Copper manufacturing".

The pilot cases to be implemented within this project have been extensively described in other deliverables and a brief summary of them are included in the corresponding sub-sections of this document, enabling the reader to understand better the framework of the project.

2.3 Use cases description

In Cocop use cases are documented according to Allistair's model [1].

Use cases descriptions are written in Confluence. Each use case should be described as a separate page. A use case documents the operation and behaviour of a system (i.e. the COCOP solution) from the user point of view. This means that any actions and steps a user takes and what the expected outcome should be documented but without describing how it is implemented.

A use case contains the following elements:

---

[1] *WRITING EFFECTIVE USE CASES, Alistair Cockburn, Addison-Wesley, c. 2001.*

Element	Description
(Heading)	ID-NAME - Short description (e.g. "UC-C-PSC-CB: Copper Blow") The *ID* should be a string for referencing the use case. E.g. "UC-C-PSC-CB" means "copper, Peirce-Smith converter, copper blow". Hierarchical identifiers are encouraged. (See table for abbreviations in sect 2.5) The *name* part is there to provide an easily readable title.
Purpose	The purpose of the use case in short.
Actors	Any actors involved (e.g. operators).
Preconditions	What is the state of the system before the use case can begin; what has already been done.
Body	The actual body of the use case: the description of the scenario. For readability, it is encouraged to use numbered steps to structure the contents. Using *actor names* as the subjects of each sentence should be favoured.
Exceptions	Any exceptions of the scenario. What the exception is and why it occurs should be explained. An identifier (e.g. a few letters) for each exception should be assigned and references to it should be included in appropriate positions in the use case body.
Postconditions	The results of the use case if run successfully: e.g. some process phase has been finished.
Other remarks (optional)	Any other notes about the use case.
JIRA links (embedded in text; not a section)	Links to all related JIRA issues (issues, stories, tasks).

In practice, use cases tend to grow large. If some functionality is recurring, for example, it can be described once and the INCLUDED into and referenced by other use cases. Another common situation is when some functionality happens occasionally or due to an exception. In such cases, a new use case can be defined that EXTEND the first one. Proper referencing for traceability is also good practice.

2.4 System Requirements

JIRA is used to manage the backlog of the project. The backlog consists of the list of activities that will be done in each iteration of the solution development. The backlog consists of JIRA issues: mainly stories and tasks. Stories are used to describe the needed features, and to some extent, they also dissect the use cases into manageable items. A story is a software system requirement that is expressed in a few short sentences, ideally using non-technical language to ensure a proper understanding of it to all the people working around it (end users, programmers, integrators, etc.). JIRA supports creating sub-tasks for the stories and these can be made to divide and manage the technical work. The JIRA issues created in the COCOP project are linked to their related use case's descriptions in Confluence.

In order to establish a clear way of defining the requirements (stories) in JIRA within the COCOP project, the following rules / indications were provided to all project members:

Requirement naming: e.g. "REQ-C-PSC-CB-010" means "copper, Peirce-Smith converter, copper blow, requirement 010".

- Hierarchical identifiers are encouraged.
- By default, numeric IDs are incremented by 10 to enable adding new requirements in between existing ones. Numeric identifiers enable chronological requirement ordering (the order they apply in the production process).

When creating a requirement into JIRA **labels** should be used to denote the category of the requirement. The label candidates are:

- ***general*** *(system requirement)*
- ***copper-case*** *(copper case requirement)*
- ***steel-case*** *(steel case requirement)*
- ***req-non-functional*** *(non-functional requirement)*
- ***req-functional*** *(functional requirement)*
- ***req-security*** *(system security requirement)*
- ***kpi*** *(performance requirement)*

If the new JIRA story is created from **Confluence**, then a **link** to the (use case) page from which it was created appears automatically. But if requirements (stories) are created manually in JIRA one must **add the use case as a link to the JIRA item** for traceability whenever a story is related to a use case.

In order to write the requirements, the next wording recommendations from RFC 2119 were also provided:

1. "MUST" This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
2. "MUST NOT" This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.
3. "SHOULD" This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
4. "SHOULD NOT" This phrase, or the phrase "NOT RECOMMENDED" mean that there may

exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.

5. "MAY" This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein, an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

2.5 Abbreviations for use cases and requirements

In the below table, the columns Generic aspects, Copper pilot case and Steel pilot case denote the subsections where the abbreviation is used.

Abbreviation	Description	Generic aspects	Copper pilot case	Steel pilot case
ACC	Access	x		
ADD	Add	x		
AF	Anode Furnace		x	
AP	Acid Plant		x	
C	Copper case		x	
CB	Copper Blow		x	
CC	Continuous Casting			x
CON	Connectivity	x		
CONF	Configure	x		
CONSULT	Consultation			x
CONTROL	Control			x
COO	Coordination		x	
CRE	Create	x		
CUST	Customize	x		
DATA	Data	x		
DB	Database	x		
DCS	Distributed Control System	x		
DEL	Delete	x		
EDIT	Edit	x		

Abbreviation	Description	Generic aspects	Copper pilot case	Steel pilot case
EXE	Execute	x		
EXP	Export	x		
FB	Feedback	x		
FSF	Flash Smelting Furnace		x	
G	General	x		
GAM	Gamification	x		
HELP	Help	x		
HR	Hot rolling			x
INIT	Initialise	x		
INST	Installation	x		
KPI	Key Performance Indicator	x	x	x
LCA	Life Cycle Assessment	x		
MOD	Model	x		
MONITORING	Monitoring			x
OPT	Optimisation	x		
PRESENT	Present	x		
PSC	Pierce-Smith Converter		x	
REQ	Requirement	x	x	x
RESET	Reset	x		
S	Steel			x
SB	Slag Blow		x	
SCF	Slag Cleaning Furnace		x	
SCFNO	No SCF used		x	
SEC	Security	x		
SM	Secondary Metallurgy			x
SO2	Sulphur		x	
STATE	State	x		
SYS	System	x		
TAR	Target	x		
UC	Use case	x	x	x
UI	User interface	x		
UPD	Update	x		
USER	User	x		
VIEW	View	x		

2.6 Confidentiality in this document

The descriptions of each one of the COCOP project pilot cases (including process description, use case definition and associated requirements) include a high level of detail about the current operations of the associated companies and their processes. Therefore, most of this information is considered confidential by the companies. Accordingly, the present deliverable only includes a summary of these elements. Specifically, the process descriptions presented here are shorter versions of the information provided to the consortium and available in the internal Confluence system. Accordingly, both use cases and associated requirements are only listed in a table-style without including the extended description of these elements to avoid sharing confidential information.

2.7 Continuous Update of Information

Software development methodologies have changed severely in the last 25 years. From the tight patterns followed in the 80's to the currently used methodologies there has been a big change on the way to implement and control the software project execution. One of these methodologies, Agile, is defined as:

> *"Agile software development describes a set of values and principles for software development under which requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams. It advocates adaptive planning, evolutionary development, early delivery, and continuous improvement, and it encourages rapid and flexible response to change. These principles support the definition and continuing evolution of many software development methods."*

This definition fits perfectly with the characteristics of a research project like COCOP. Not tying the development to a fixed set of specifications (and keep evolving it) is well justified by two main reasons: the multidisciplinary nature of the members of the consortium; and the high novelty of the activities performed in the project. Both facts may lead to more than one change in the followed strategy during the implementation, reinforcing the need of a dynamic set of requirements. Accordingly, during the whole project, the partners will act as an Agile software development team, performing evolutionary development. This means reviewing at each stage the current status of the development and the needs required to reach the next step. Moreover, this also enables the end users of the system, to evaluate if the current implementation is going towards their needs and to identify potential mitigation changes when required.

2.8 Implementation of the backlog - Continuation of the COCOP work

The target of the COCOP project in terms of technology readiness level is to reach a level 6 of the overall defined solution. TRL 6, under the EU Commission definitions, is to have the technology demonstrated in a relevant environment (industrially relevant environment in the case of key enabling technologies). As commented at the beginning of this section, there is a big gap yet to translate a concept at TRL6 into a market-ready software solution. Besides the different actions taken within the project in the scope of the exploitation and dissemination activities, in the technical level it has been decided to not limit the definition of the backlog to the activities that will be executed within the project, but to extend its reach to the completion of the platform up to a commercial level. This means that although the development activities to be done within the project will be aimed to demonstrate the concept and to reach TRL 6, the consortium members will not only include those in the backlog of the system. Alternatively, all the implicated partners will continuously analyse the status of the platform, identifying the functionality (requirements) that should be included in a commercial solution exploiting the concept demonstrated in the COCOP project.

This decision also brings the need to establish a proper monitoring to define which requirements will be implemented during the project (and within each one of its development iterations) and which of them will be marked as required for its further development.

3 COCOP System Requirements Specifications

As mentioned in previous sections the system definitions have been split into three different categories. The first one, "Generic", contemplates all the aspects that are general to the COCOP concept and is not tied to a specific application of the platform. The two following ones include all the aspects that are specific to the two pilot cases of the project. Each one of these categories has been extensively defined in the IT platform of the project. Next, a summary of the main information contained in such system is included, providing a clear view of the scope of the activities and needs established (but yet maintaining the confidentiality of the specific details of each of the pilot cases as mentioned in previous subsections).

3.1 Generic (non-pilot specific) aspects

3.1.1 Use cases

ID	Group	Name	Short Description
UC-G-CON-DB	Connectivity	Connect historian databases	Provide connectivity to historian, MES and other systems through databases
UC-G-CON-DCS	Connectivity	Connect automation systems	Provide direct connectivity to automation systems.
UC-G-DATA-ADD-EDIT	DATA	Add a data item to communication	Create a new data item in the system providing connectivity to the corresponding data source
UC-G-DATA-DEL	DATA	Delete a data item from the system	Delete a data item from the system managing its dependencies
UC-G-HELP-ADD-EDIT	HELP	Adding or editing User Help contents	Create or update a new entry for the User Help system connecting it to a platform element
UC-G-HELP-PRESENT	HELP	Presenting User Help contents	Display the help for a given platform element retrieving it from the User Help System
UC-G-HELP-DEL	HELP	Delete a user help content	Delete a user help content from the User Help system

ID	Group	Name	Short Description
UC-G-INST-CUST	Installation	Install and customise the system	Provide functionality to customise the installation of the platform in a given system
UC-G-LCA-ADD-EDIT	LCA	Adding a new LCA model or update existing model	Install or modify an LCA model in the system defining its links to other system elements
UC-G-LCA-EXE	LCA	Execute on-line LCA calculation	Execute an LCA or provide a Sustainability Indicator(SI) for a given process/system
UC-G-MOD-ADD	Models	Adding a new model or update existing model	Install or modify a model in the system defining its links to other system elements
UC-G-MOD-EXE	Models	Execute a unit model once	Execute a model for a given set of parameters, providing the calculated output
UC-G-MOD-EXP	Models	Export model definition	Export a system's model definition into a file for enabling its future import.
UC-G-MOD-UPD	Models	Update parameters of an existing model	Update the parameters of a model that is available in the platform.
UC-G-MOD-DEL	Models	Delete a model	Delete a model from the system, handling its dependencies in the system
UC-G-OPT-CONF	Optimisation	Configure optimisation set	Configure the parameters of a optimisation set
UC-G-OPT-CRE	Optimisation	Create optimisation set	Create an optimisation set
UC-G-OPT-DEL	Optimisation	Delete optimisation set	Delete an optimisation set, handling its dependencies in the system
UC-G-OPT-EXE	Optimisation	Execute coordination optimisation once	Perform a coordination optimisation set

ID	Group	Name	Short Description
UC-G-OPT-INIT	Optimisation	Initialise the optimisation	Initialise an optimisation set by handling its prerequisites
UC-G-OPT-RESET	Optimisation	Reset the system	Reset the optimisation system
UC-G-OPT-VIEW	Optimisation	View optimisation set	Display the information about an optimisation set including its properties
UC-G-UI-CONF	User Interface	Configure user interface	To configure a customised user interface which the operator can use to access the COCOP system
UC-G-UI-PRESENT	User Interface	Present results of optimisation on the UI	Present results of the optimisation on the UIs of all associated sub-processes
UC-G-USER-CRE	Users	Create user	To add a new user account for the system
UC-G-USER-DEL	Users	Delete user	To add a new user account for the system
UC-G-USER-FB	Users	Collect user feedback on system usage	Collect information of how much the COCOP system is used by its users
UC-G-USER-EDIT	Users	Modify user profile	Modify the profile of an existing user, e.g., change the credentials
UC-G-USER-VIEW	Users	View user profile	View the profile of an existing user

3.1.2 Requirements

This sub-section details all the requirements that have been defined for the general implementation of the platform. These requirements have been split into non-functional, functional, security-related and performance-related ones. Please note that some of the requirements appear in several categories.

3.1.2.1 Non-functional requirements:

Summary
REQ-G-ACC-010 - Different user levels
REQ-G-CON-DB-010 - MySQL Connectivity
REQ-G-CON-DB-020 - OPC UA HA (Historical Access) client
REQ-G-CON-DCS-005 - DCS OPC required
REQ-G-CON-DCS-010 - Industry standard connectivity
REQ-G-GAM-PRESENT-010 - Defines the need to visualize the predicted trajectory
REQ-G-GAM-TAR-010 - Defines the need to evaluate how well a target is met
REQ-G-MOD-STATE-010 - State estimation
REQ-G-UI-010 - Meaning of colours on UI
REQ-G-UI-020 - Red and green colour on the UI
REQ-G-UI-030 - Symbols on the UI
REQ-G-UI-040 - Show system status on UI
REQ-G-UI-050 - UI consistent with real world process
REQ-G-UI-070 - Internal consistency
REQ-G-UI-080 - User error prevention
REQ-G-UI-090 - Memory supporting functions to UI
REQ-G-UI-100 - UI shortcuts
REQ-G-UI-110 - UI design
REQ-G-UI-120 - Error messages
REQ-G-UI-130 - Not crowded UI
REQ-G-UI-135 - User help
REQ-G-UI-140 - Pilot case specific terminology
REQ-G-UI-150 - Sustainability aspects to UI
REQ-G-UI-160 - Shift performance and UI
REQ-G-UI-170 - UI interaction experience and performance
REQ-G-UI-180 - Dig in to details

3.1.2.2 Functional requirements:

Summary
REQ-G-DATA-010 - Data quality bit
REQ-G-DATA-ADD-010 - OPC UA client address space browsing
REQ-G-INST-CUST-010 - Localization
REQ-G-LCA-010 LCA models
REQ-G-LCA-020 Input from the process to LCA
REQ-G-LCA-030 LCA model linking to other models
REQ-G-LOG-010 - System log
REQ-G-MOD-ADD-010 - Model help
REQ-G-MOD-ADD-020 - Indicates that the model import operation requires a description for it
REQ-G-MOD-ADD-030 - Model input validity ranges and engineering units
REQ-G-MOD-ADD-040 - Model output validity ranges and engineering units
REQ-G-MOD-ADD-050 - Model data validity ranges and engineering units
REQ-G-MOD-ADD-060 - Data modifications
REQ-G-MOD-UPD-020 - Model data checks
REQ-G-MSG-010 - Error message types
REQ-G-OPT-010 - Forecast of problems
REQ-G-OPT-EXE-010 - Check data before optimization
REQ-G-OPT-EXE-020 - Optimization algorithm
REQ-G-SYS-STATE-010 - Save system state
REQ-G-UI-060 - Undo
REQ-G-UI-CONF-010 - Connectivity with UI
UC-G-HELP-010 - User help

3.1.2.3 Security requirements:

Summary
REQ-G-MOD-ADD-070 - Access control to model data
REQ-G-MOD-UPD-010 - Data update access control
REQ-G-SEC-AUTHENTICATION-010 - System access authentication
REQ-G-SEC-AUTHORIZATION-010 - Authorization requirements in the platform
REQ-G-SEC-NETWORKING-010 - System access definition based on network parameters

3.1.2.4 Performance requirements:

Below is a list of requirements to measure social acceptance:

Summary
REQ-KPI-G-010 - Usage of system advice
REQ-KPI-G-020 - Acceptance of system advice
REQ-KPI-G-030 - Plant-wide processes as part of operator training ratio relative to baseline
REQ-KPI-G-040 - The level of understanding the plant-wide processes relative to baseline
REQ-KPI-G-050 - Operators' job satisfaction relative to baseline
REQ-KPI-G-060 - Participation ratio: plant-wide optimization

3.2 Copper pilot case

3.2.1 Process short description

The first pilot case of the COCOP concept demonstration platform will be applying it to a copper-smelting plant. The prototype will be installed in the Boliden Harjavalta smelter. The potential for improvement in this pilot case comes from increased production and reduced emissions related to more precise timing of the operations. Especially, the value of the production increase is significant. Since the copper refinery utilises the ingenious chemical energy of *Cu* concentrate, the energy savings and CO_2 emissions are minor. On the other hand, optimised process operation can further reduce CO_2 emissions per ton of concentrate and reduce the amounts of SO_2 released to the atmosphere through, for example, reduction of process variations or improved response to abnormal process conditions. When the process variations are in control the acid plant which process SO2 to sulphuric acid is able to capture the SO_2 containing offgas produced in Flash Smelting Furnace (FSF) and Peirce-Smith Converters (PSC).

COCOP includes Blend house, drying, Flash Smelting Furnace (FSF), Peirce Smith Converters (PSC) and Anode Furnace (AF) processes. Acid plant, refinery (electrolysis), slag concentrate plant and slow cooling are left outside COCOP. However, the capacity of the acid plant is viewed as a constraint and the actual capacity is monitored. Further, the effect of process parameters to slag chemistry to be able to improve the recovery of copper in slag concentrate plant is taken into account in the target function of the plant-wide optimisation. Also, the constraints the refinery sets to anode copper quality are included.

When the *Cu* concentrate is refined to *Cu*, the total production volume is an obvious objective. The yield of *Cu*, purity of end product, and reducing emissions (dust, SO_2 and CO_2) are also included in the objectives. A copper smelter consists of several unit processes, each having its own control room, typically spatially distributed and responsible for the local efficient operation. In such large-scale plants, there are several local practical but in a larger scale contradicting production objectives. In general, reduction of process variable variance allows operation of the process such that all the objectives can be reached. Thus, a general objective of process operation and control is to reduce variations and to make decisions that attenuate or do not generate variations.

Overall objectives are

- Increasing the capacity of the smelter
- Increasing the recovery of copper
- Decreasing the emissions per ton produced anode copper (SO_2, CO_2)
- Decreasing the use of fossil fuels and raw materials producing CO_2
- Better control of impurities in anode copper.

The image below depicts the flowsheet of the Copper converting process. The numbers (1-4) indicate the different main models to be developed within the project.

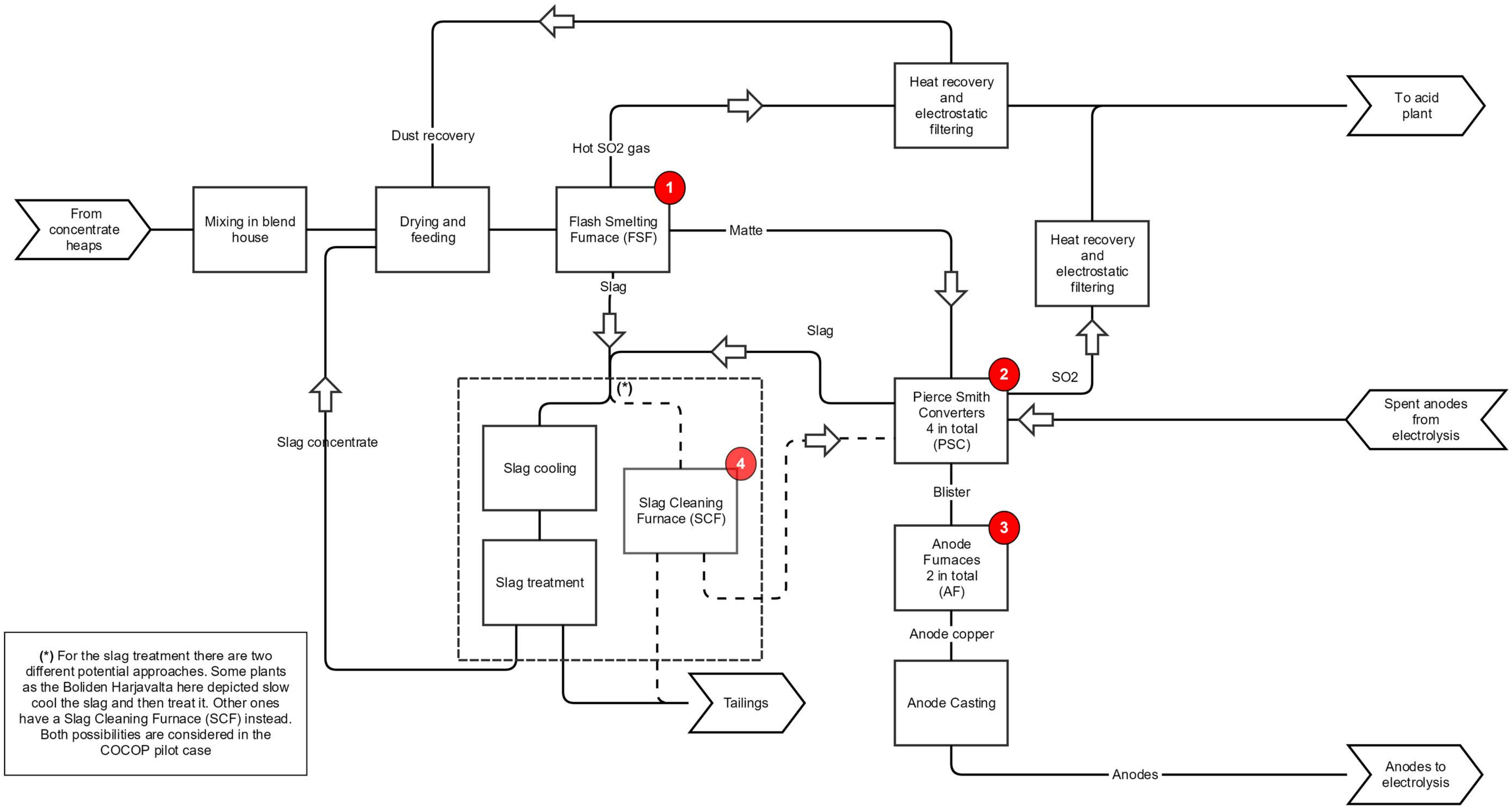

Figure 1 Simplified flowsheet depicting the material streams in the copper smelter process

3.2.2 Main actors description

The main actors of the use cases can be described as follows. The production operations organization has separate branches for smelters, copper refinery and acid plant with responsible manager. In each branch, there are production engineers and area supervisors for each main unit process. Production engineers and supervisors work mainly in the day shift. There are five shifts responsible for actual operational tasks. All the five shifts contain a shift supervisor and operators for each unit process.

In the copper pilot case, the main users of the COCOP platform will be the operators. Operators are the key personnel for an effective and efficient production. Their role and leeway in process controlling and the working places are of relevance.

3.2.3 Models to be used

Within the copper pilot case, and following the process break-down description provided in previous sub-sections, the following models are scheduled to be included in the COCOP solution for the Copper pilot case.

Model ID	Related Process	Actions	Model short description
C-FSF	1 - FSF	INCLUDE	The flash smelting furnace separates the input feed materials to matte and slag through oxidation.
C-PSC-SB	2 - PSC	UPGRADE	Model for slag-making blows where most of the iron compounds are removed from FSF matte in multiple blows.
C-PSC-CB	2 - PSC	UPGRADE	Model for composition and temperature of PSC contents during coppermaking blows
C-AF	3 - PSC	UPGRADE	Modelling oxidation and reduction requirements in AF.
C-SCF	4 - SCF	UPGRADE	Reduction of metal oxides contained in copper smelter slag to metals with coke.
C-SCHED	1-4	DEVELOP	Scheduling and optimisation specific modelling not obtainable from the above process models.
C-CRA	1-3	DEVELOP	Crane data handler needed to manage the mass transports from unit to unit.

Table 1 Copper pilot case - Model summary list

Additionally, the optimisation solution will require that the C-FSF, C-PSC-SB, C-PSC-CB, C-AF and C-SCF models are simplified when used in the optimisation problem. The different requirements for the models in the different roles are emphasised below but the models are not listed as different models. Crane operations have been differentiated from other logistical constraints incorporated into C-SCHED as the transported masses and their measurements are especially important also when the models are used in the advisory role.

3.2.4 Defined use cases

This sub-section details all the use cases that have been defined for the copper pilot case.

ID	Name	Short Description
UC-C-AF	Anode Furnace	Operation of the anode furnace, removing S and O from a blister copper ensuring enough high blister temperature for anode casting
UC-C-FSF	Flash Smelt Furnace	Operation of the flash smelt furnace fulfilling 3 objectives: · Producing matte to be further processed in PS-Converters with suitable matte grade

ID	Name	Short Description
		· To produce slag with optimal Si02 content and temperature · To maximise feed to the furnace. · To define the tapping strategy.
UC-C-PSC-CB	Copper Blow	Operation of the Pierce Smith Converters during the copper blow stage, removing the sulphur from white metal
UC-C-PSC-SB	Slag Blow	Operation of the Pierce Smith Converters during the slag blow stage, producing white metal
UC-C-SCF	Slag Cleaning Furnace	Operation of the Slag Cleaning Furnace, recovering copper from slag to matte, controlling its sulphur content.
UC-C-COO-AF-SCFNO	Coordination with AF limit, no SCF	Optimisation of the coordinated optimisation when the anode furnace is the bottleneck of the operation and there is no SCF in the smelter.
UC-C-COO-AF-SCF	Coordination with AF limit, SCF used	Optimisation of the coordinated optimisation when the anode furnace is the bottleneck of the operation and there is an SCF in the smelter.
UC-C-COO-FSF-SCFNO	Coordination with FSF limit, no SCF	Optimisation of the coordinated optimisation when the flash smelting furnace is the bottleneck of the operation and there is no SCF in the smelter.
UC-C-COO-FSF-SCF	Coordination with FSF limit, SCF used	Optimisation of the coordinated optimisation when the flash smelting furnace is the bottleneck of the operation and there is an SCF in the smelter.
UC-C-COO-PSC-SCFNO	Coordination with PSC limit, no SCF	Optimisation of the coordinated optimisation when the Pierce Smith Converters are the bottleneck of the operation and there is no SCF in the smelter.
UC-C-COO-PSC-SCF	Coordination with PSC limit, SCF used	Optimisation of the coordinated optimisation when the Pierce Smith Converters are the bottleneck of the operation and there is a nSCF in the smelter.
UC-C-COO-SO2-SCFNO	Coordination with SO2 limit, no SCF	Optimisation of the coordinated optimisation when the sulphur processing plant is the bottleneck of the operation and there is no SCF in the smelter.
UC-C-COO-SO2-SCF	Coordination with SO2 limit, SCF used	Optimisation of the coordinated optimisation when the sulphur processing plant is the bottleneck of the operation and there is an SCF in the smelter.

3.2.5 Requirements

This sub-section details all the requirements that have been defined for the copper pilot case.

3.2.5.1 Functional Requirements

Summary
REQ-C-010 - Free capacity on acid plant indication
REQ-C-020 - Bottleneck indication targets
REQ-C-030 - Consider work shift in scheduling
REQ-C-040 - Operators not following advises detection and consideration
REQ-C-050 - Temporal resource limitations
REQ-C-AF-010 - AF batch temperature
REQ-C-AF-020 - Batch composition estimation
REQ-C-AF-030 - Oxidation time calculations
REQ-C-AF-050 - New blister copper batch indication
REQ-C-FSF-010 - FSF feed rate indications
REQ-C-FSF-020 - Matte grade evaluation
REQ-C-FSF-030 - FSF Fuel and O2 indications
REQ-C-FSF-040 - Off-gas indications
REQ-C-FSF-050 - Oxygen flow control indications
REQ-C-FSF-060 - Matte tapping suggestions
REQ-C-FSF-070 - Slag tapping suggestions
REQ-C-PSC-CB-010 - Coolant requirements estimation
REQ-C-PSC-CB-020 - Blowing remaining time indication during Copper Blow
REQ-C-PSC-CB-030 - End temperature indication
REQ-C-PSC-SB-010 - Matte ladles scheduling in a PSC
REQ-C-PSC-SB-020 - Silica flux indications

Summary
REQ-C-PSC-SB-031 - PSC remaining capacity estimation
REQ-C-PSC-SB-040 - Slag blow duration indication
REQ-C-PSC-SB-050 - Slag generated in last Slag blow indication
REQ-C-PSC-SB-060 - Blowing remaining time indication during Slag blow
REQ-C-PSC-SB-070 - FeS monitorisation
REQ-C-SCF-010 - Temperature indications
REQ-C-SCF-020 - Coke dosage estimation
REQ-C-SCF-030 - Slag ladles timing indications
REQ-C-SCF-040 - FSF tapping scheduling

3.2.5.2 Performance Requirements

Summary
REQ-KPI-AF-FSF-010 - Oil usage in the anode furnaces and FSF
REQ-KPI-AF-PSC-010 - Stability Anode Composition due to advise tool
REQ-KPI-AF-PSC-020 - Scrap usage ratio
REQ-KPI-C-AF-010 - Propane usage relative to amount produced copper
REQ-KPI-C-AP-010 - Equipment load ratio: Acid plant
REQ-KPI-C-AP-020 - Stability Acid plant
REQ-KPI-C-FSF-010 - Equipment load ratio: Feed Rate FSF
REQ-KPI-C-FSF-020 - Stability Feed Rate FSF
REQ-KPI-C-FSF-PSC-010 - Copper content in waste slag relative to baseline
REQ-KPI-C-PSC-010 - Wear of bricklining (velocity) relative to produced blister
REQ-KPI-C-PSC-020 - Wear of bricklining (velocity)
REQ-KPI-C-PSC-030 - Amount of produced converter slag compared to optimal estimate from FSF matte analysis

3.3 Steel pilot case

3.3.1 Process short description

The steel pilot case will focus on the superficial quality of microalloyed steel grades produced by SIDENOR Basauri. Due to the global benefit that these steels offer to the different agents in the whole supply chain, microalloyed steels are a growing trend in special steel industry. But at the same time, they are quite demanding for the steel producer as one of their intrinsic properties is a low ductility through at some critical temperatures. The low ductility makes them very prone to superficial cracks as the billet surface suffers importantly from strains in different sub-processes.

The steel pilot case will address the three sub-processes of SIDENOR that have an influence in the generation of surface defects in these microalloyed steels: (1) secondary metallurgy station, (2) continuous casting process and (3) hot rolling. The next diagram depicts the production flow of SIDENOR Basauri Works.

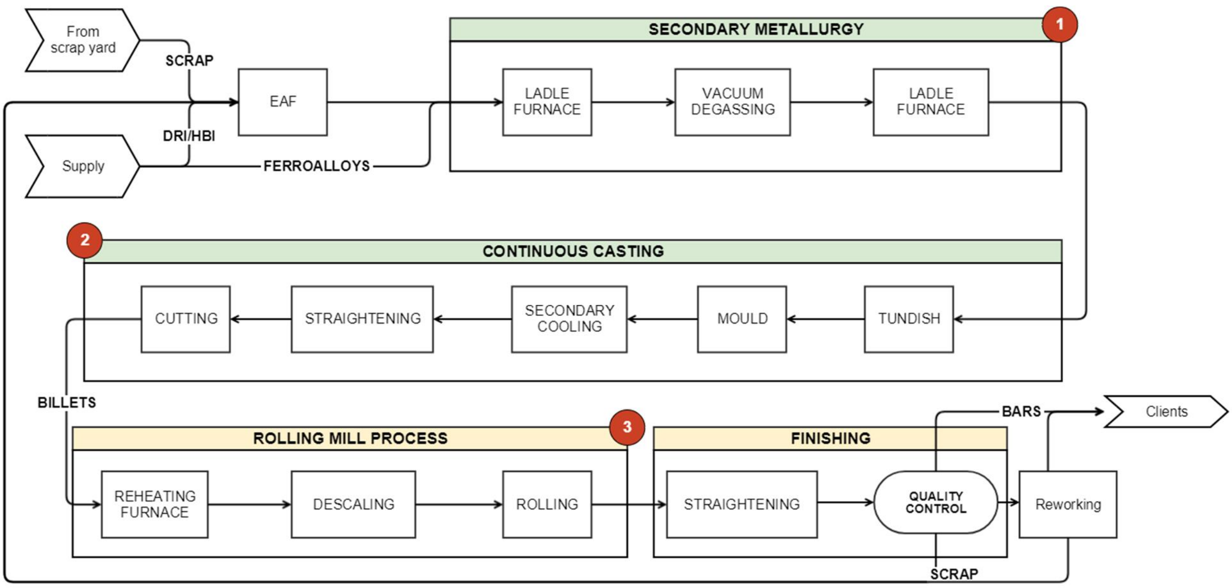

Figure 2 Flow diagram of the production of SIDENOR Basauri Works

The secondary metallurgy involves a set of processes and installations in order to refine the crude steel before casting. It basically involves adjusting the composition of the steel by the addition of ferroalloys (refining process) and removal of hydrogen and nitrogen, achieving good deoxidation and cleanness of steel. Sulphur is removed with the help of an appropriate slag and inclusions are removed or altered chemically to ensure that high-quality liquid steel is produced. To homogenize the temperature of the steel and the composition, the steel is stirred by injecting argon through a porous plug located at the bottom of the ladle.

Once temperature conditions and composition in the liquid steel are reached, the ladle is raised in a turret for casting liquid steel into a tundish with six strands. The continuous casting is the process whereby molten steel is solidified into a "semi-finished" billet, by means of a water-cooled open mold, secondary cooling combining water sprays and air, electromagnetic stirring, straightening and cutting the semi-product to the desired size. Steel solidification is a demanding process as liquid steel is above 1550 ºC and in the cooling process, the steel suffers from at least two phase changes, in a product that both in liquid and solid is dense and, consequently, heavy. The casting speed varies depending on the section to be cast, the steel composition and the temperature. After solidifying, the billets remain in any of the three previous cooling beds transfer to the rolling mill.

The billets are heated and hot rolled through several pairs of rolls to reduce the thickness and to obtain the desired shape and thickness of the bars. The steel quality is usually improved when it endures mechanical stresses in hot temperature. This allows steel structure to reduce average grain size and equalize them at the same time whereas other solidification defects as segregation or central pores are reduced significantly. For all these reasons, as-rolled steel is usually better than as-cast steel.

Finally, the 100% of the bars are inspected by Eddy Current technology to detect surface cracks. After this automated crack detection, there is a manual inspection of every crack-detected bar checking if the crack can be repaired.

3.3.2 Main actors description

The organisational structure of the areas considered in this pilot case is depicted in the three images below:

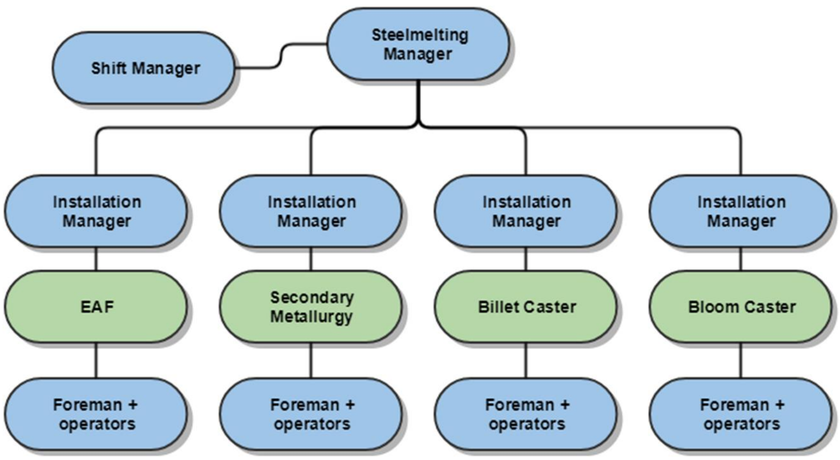

Figure 3: Organisation Chart Steelmaking (Secondary Metallurgy / Continuous Casting)

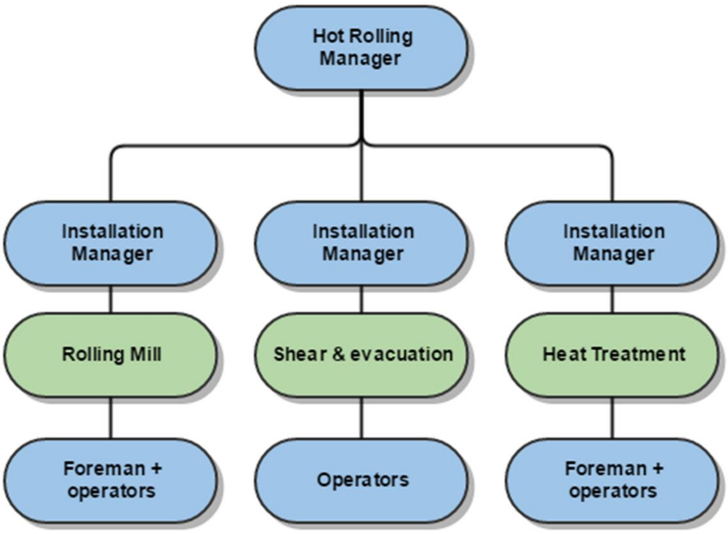

Figure 4: Organisation Chart Hot Rolling

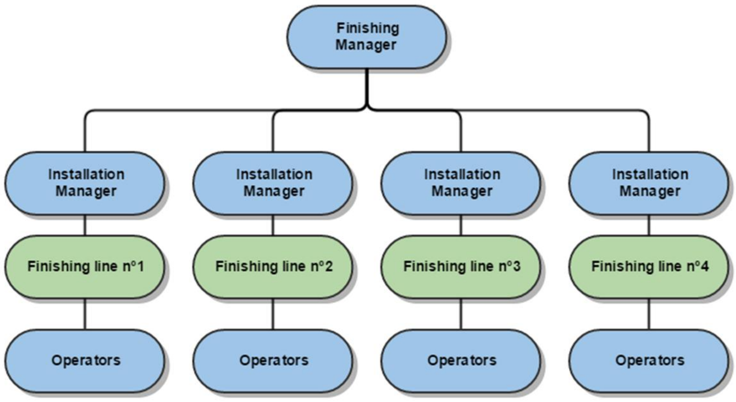

Figure 5 Organisation Chart Finishing Line

As users of the COCOP solution, the focus will be on the key personnel with direct responsibilities and intervention possibilities, named in the organisation charts: shift and installation managers, the foremen and operators. We will also take into account further responsible persons that might be affected by the innovation process later.

3.3.3 Models to be used

The figure below shows the models that will be developed for the Steel pilot case and their relationship. After the identification of the key parameters of each sub-process for the defects generation, a Data Based Model will be developed with these key parameters in order to predict the final defects of the Steel product: SteelDefects. In addition, three Data Based Models will also be developed for each subprocess: SteelSM, SteelCC and SteelHR. These models will simulate a good operation practice of each subprocess.

Finally, a Mathematical Model of the Continuous Casting (MatCC) process will be developed in order to calculate some relevant variables that cannot be measured by physical means such as solidified Shell thickness or the temperature in a point of the strand. This model will be used as a software sensor. The values of these variables will be included in the SteelCC model.

The goal will be to find the best combination of the values for the key defect-related parameters that minimise the generation of defects assuring a good performance of each sub-processes.

Model ID	Related process	Actions	Model short description
SteelDefects	1-3 All	Develop	Model to predict the surface defect generation
SteelSM	1 - SM	Develop	Simulate good operation practice of the Secondary Metallurgy process
SteelCC	2 - CC	Develop	Simulate good operation practice of the Continuous Casting process
SteelHR	3 - HR	Develop	Simulate good operation practice of the Hot Rolling process
MatCC	2 - CC	Upgrade	Model to calculate the thermal and solidification evolution along the Continuous Casting

Table 2 Steel pilot case - Model summary list

3.3.4 Defined use cases

This sub-section details all the use cases that have been defined for the steel pilot case.

ID	Name	Short Description
UC-S-SM-OPT - Secondary Metallurgy Optimization	Secondary metallurgy planning optimisation	Optimal definition of the Secondary Metallurgy Process Parameters to minimise final defects,being compatible with a good operation of the sub-process to ensure a good castability of the heat. Used by the secondary metallurgy manager.
UC-S-CC-OPT - Continuous Casting Optimization	Continuous casting planning optimisation	Optimal definition of the Continuous Casting Process Parameters to minimise final defects, being compatible with a good operation of the sub-process that ensures a good temperature of the billet along the process. Used by the continuous casting manager.
UC-S-HR-OPT - Hot rolling planning	Hot rolling planning optimisation	Optimal definition of the Hot Rolling Process Parameters to minimize final defects, being

ID	Name	Short Description
optimisation		compatible with a good operation of the sub-process. Used by the hot rolling installation manager.
UC-S-SM-MONITORING - On-line alarm tool in Secondary Metallurgy	On-line alarm tool in secondary metallurgy	Alarm system indicating one of the following situations: · Predicted defects on end product for the actual process parameters · Bad castability index predictions of the models for the actual process parameters · Predicted temperature of the casting process is out of the established bounds. Used by the shift manager
UC-S-CC-MONITORING - On-line alarm tool in Continuos Casting process	On-line alarm tool in continuous casting process	Alarm system indicating one of the following situations: · The current production parameters might produce a high number of defects in the end product. · The temperature of the billet before the straightening is not enough considering the ductility trough. · The medium thickness evolution at the output of the mould is not good to avoid a break-out. · There are differences between the target and real parameters Used by the shift manager
UC-S-CC-CONTROL - Off-line tool for defect prediction in Steeelmaking	Off-line tool for defect prediction in steelmaking	To predict billet defects after continuous casting based on real process data Used by the steelmaking quality manager
UC-S-CC-CONSULT - Off-line tool for validation and consultation of defect prediction	Off-line tool for validation and consultation of defect prediction in steelmaking	To compare the defect prediction based on real process data with the real performance. And in case of having a high number of real defects, to allow analysing which would have been the best process parameters to avoid them. Used by the steelmaking quality manager

3.3.5 Requirements

This sub-section details all the requirements that have been defined for the steel pilot case.

3.3.5.1 Functional Requirements

Summary
REQ-S-CC-CONTROL-010 - Quality evaluation for billets
REQ-S-CC-CONTROL-020 - Check by heat number
REQ-S-CC-CONTROL-040 - List of billets to be shown to the user
REQ-S-CC-CONTROL-050 - Number of estimated bars with defects
REQ-S-CC-CONTROL-060 - Preserve estimated quality values for the billets
REQ-S-CC-MONITORING-010 - Explain the alarm causing parameter.
REQ-S-CC-MONITORING-012 - Prediction of defects with the current SM operation parameters
REQ-S-CC-MONITORING-014 - Evaluation of temperature and ductility through
REQ-S-CC-MONITORING-016 - Prediction of break-outs
REQ-S-CC-MONITORING-018 - Real time comparison of operating parameters with recipe ones
REQ-S-CC-MONITORING-020 - The tool has to be configurable respect to alarm causes.
REQ-S-CC-OPT-010 - Set of initially optimized casting parameters for a new steel grade.
REQ-S-CC-OPT-020 - Casting parameters tuning recommendation for one heat.
REQ-S-CC-OPT-030 - The tool has to be configurable
REQ-S-HR-OPT-010 - Hot rolling thermal profile recommendation for one steel grade
REQ-S-SM-MONITORING-010 - Explain the alarm causing parameter.
REQ-S-SM-MONITORING-020 - The tool has to be configurable respect to alarm causes.
REQ-S-SM-MONITORING-030 - Extract alarm values from MES
REQ-S-SM-MONITORING-040 - Notify low castability
REQ-S-SM-MONITORING-050 - Prediction of surface quality defects

Summary
REQ-S-SM-MONITORING-060 - Prediction of the casting temperature
REQ-S-SM-MONITORING-070 - Real time comparison of operating parameters with recipe ones
REQ-S-SM-OPT-010 - Set of initially optimized secondary metallurgy process parameters for a new steel grade.
REQ-S-SM-OPT-020 - Secondary metallurgy process parameters tuning recommendation for one heat.
REQ-S-SM-OPT-030 - Retrieve SM information for a given heat
REQ-S-SM-OPT-030 - The tool has to be configurable
REQ-S-SM-OPT-040 - Storing recommendations in MES
REQ-S-SM-OPT-050 - Retrieve SM information for a given steel family

3.3.5.2 Performance requirements

Summary
REQ-KPI-S-010 - Relative Scrap ratio - Rejection on the finishing line relative to baseline due to surface quality
REQ-KPI-S-020 - Relative Rework Ratio - Reworking on the finishing line relative to baseline
REQ-KPI-S-030 - Relative Scrap Ratio - Rejection in the continuous casting relative to baseline

4 Conclusions

This document presents the current status of the definition of the COCOP system requirements. The IT platforms used in the project to manage both requirements and use cases will be continuously used, monitored and updated during the project, in order to ensure the proper execution of the project. Finally, a set of "Further tasks to be committed" report will be generated at the end of the project. It will include the information contained in the backlog that has been scheduled for its implementation after the completion of the project. It will allow other agents (and the project participants) to further develop the COCOP system into a commercial solution.