



COORDINATED PRODUCTION
FOR BETTER RESOURCE EFFICIENCY

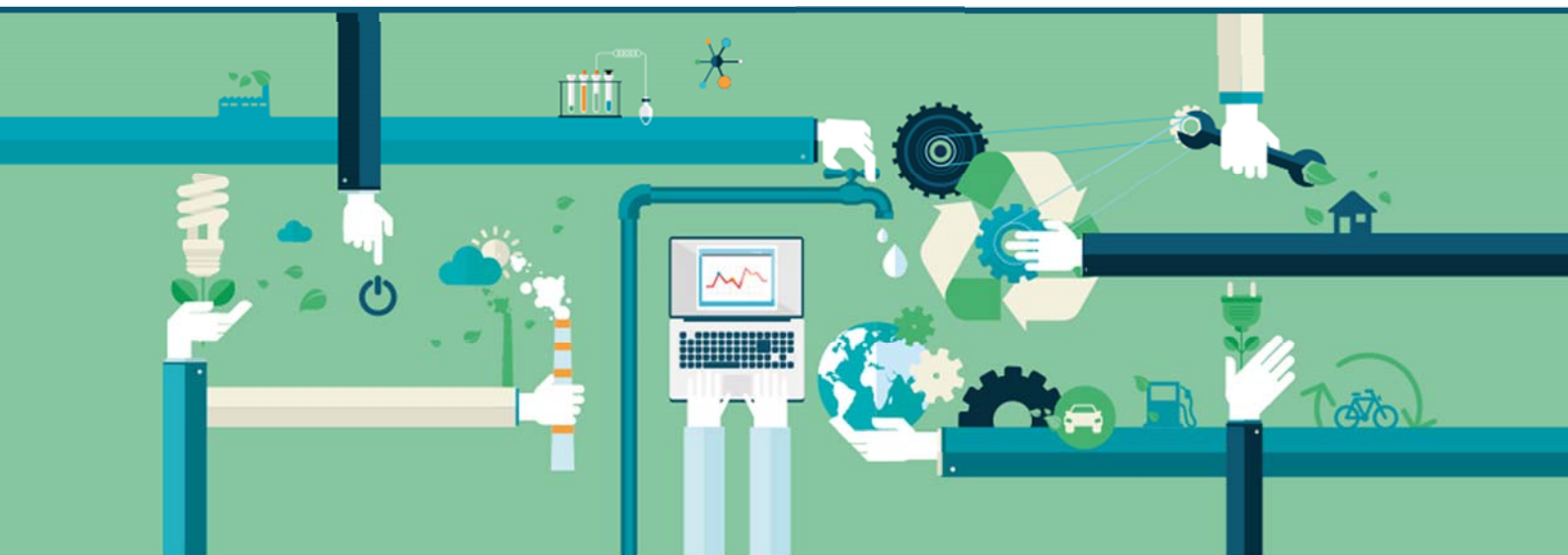
D6.4 HMI Requirements

Dr. Tim Butters

Head of R&D, Sabisu - UK

January 2019

www.spire2030.eu/copro



Project Details

PROJECT TITLE	Improved energy and resource efficiency by better coordination of production in the process industries
PROJECT ACRONYM	CoPro
GRANT AGREEMENT NO	723575
INSTRUMENT	RESEARCH AND INNOVATION ACTION
CALL	H2020-SPIRE-02-2016
STARTING DATE OF PROJECT	NOVEMBER, 1ST 2016
PROJECT DURATION	42 MONTHS
PROJECT COORDINATOR (ORGANIZATION)	PROF. SEBASTIAN ENGELL (TUDO)

THE COPRO PROJECT

The goal of CoPro is to develop and to demonstrate methods and tools for process monitoring and optimal dynamic planning, scheduling and control of plants, industrial sites and clusters under dynamic market conditions. CoPro pays special attention to the role of operators and managers in plant-wide control solutions and to the deployment of advanced solutions in industrial sites with a heterogeneous IT environment. As the effort required for the development and maintenance of accurate plant models is the bottleneck for the development and long-term operation of advanced control and scheduling solutions, CoPro will develop methods for efficient modelling and for model quality monitoring and model adaption.

The CoPro Consortium

Participant No	Participant organisation name	Country	Organisation
1 (Coordinator)	Technische Universität Dortmund (TUDO)	DE	HES
2	INEOS Köln GmbH (INEOS)	DE	IND
3	Covestro Deutschland AG (COV)	DE	IND
4	Procter & Gamble Services Company NV (P&G)	BE	IND
5	Lenzing Aktiengesellschaft (LENZING)	AU	IND
6	Frinsa del Noroeste S.A. (Frinsa)	ES	IND
7	Universidad de Valladolid (UVA)	ES	HES
8	École Polytechnique Fédérale de Lausanne (EPFL)	CH	HES
9	Ethniko Kentro Erevnas Kai Technologikis Anaptyxis (CERTH)	GR	RES
10	IIM-CSIC (CSIC)	ES	RES
11	LeiKon GmbH (LEIKON)	DE	SME
12	Process Systems Enterprise LTD (PSE)	UK	SME
13	Divis Intelligent Solutions GmbH (divis)	DE	SME
14	Argent & Waugh Ltd. (Sabisu)	UK	SME
15	ASM Soft S.L (ASM)	ES	SME
16	ORSOFT GmbH (ORS)	DE	SME
17	Inno TSD (inno)	FR	SME

Document details

DELIVERABLE TYPE		REPORT
DELIVERABLE NO		6.4
DELIVERABLE TITLE		HMI Requirements
NAME OF LEAD PARTNER FOR THIS DELIVERABLE		Sabisu
VERSION		0.3
CONTRACTUAL DELIVERY DATE		31ST OCTOBER 2018
ACTUAL DELIVERY DATE		7TH JANUARY 2019
Dissemination level		
PU	Public	X
CO	Confidential, only for members of the consortium (including the Commission)	

Abstract

The CoPro project looks to address plant coordination challenges, moving production to a site-wide optimum rather than each plant running at its local optimum, which may be sub-optimal when considering the whole site performance. Along with mathematical methods to determine this site-wide optimum, it is important that the users can easily interact with the optimisation system. Here we outline the requirements for this interface.

REVISION HISTORY

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

Revision	Date	Description	Author (Organisation)
V0.1	20/11/2018	Original draft	Tim Butters (Sabisu)
V0.2	27/11/2018	Revision	Lukas Maxeiner (TUDO)
V0.3	05/12/2018	Final Version	Tim Butters (Sabisu)
V0.3	07/01/2019	Final Approval	S. Engell (TUDO)

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 CoPro do not accept any liability for actions or omissions of CoPro members or third parties and disclaims any obligation to enforce the use of this document. This document is subject to change without notice.

Table of contents

1	Executive summary	5
2	Site-Wide Scheduling Tool	6
2.1	System Inputs & Outputs	6
2.1.1	Input Requirements.....	7
2.1.2	Output Requirements.....	8
2.2	Layers of abstraction	8
2.3	Timing Considerations	8
2.4	Validation and optimisation	8
2.5	What-If? Scenario Testing	9
2.6	Communicating Changes	9
2.6.1	Minimising Communication	9
2.6.2	Limiting Locations in Visualisation	9
2.7	Changing the Optimisation Objective	9
2.8	Communities & Security	10
3	Flexibility of Solution	10

1 Executive summary

The Human-Machine-Interface (HMI) is a key part of any computer system, as it provides the mechanisms through which the users will interact with the software. When developing an HMI it is important to establish how the system will be used and what it needs to deliver. From this, a list of functional requirements are produced. It is imperative that the key requirements are met for the system to be useable.

By building a detailed set of requirements like this, the HMI can be designed in a way that ensures that all of the key requirements are met, and as many optional requirements are considered without impacting the efficiency of the core function-set. In this document we outline the requirements of the scheduling tool HMI.

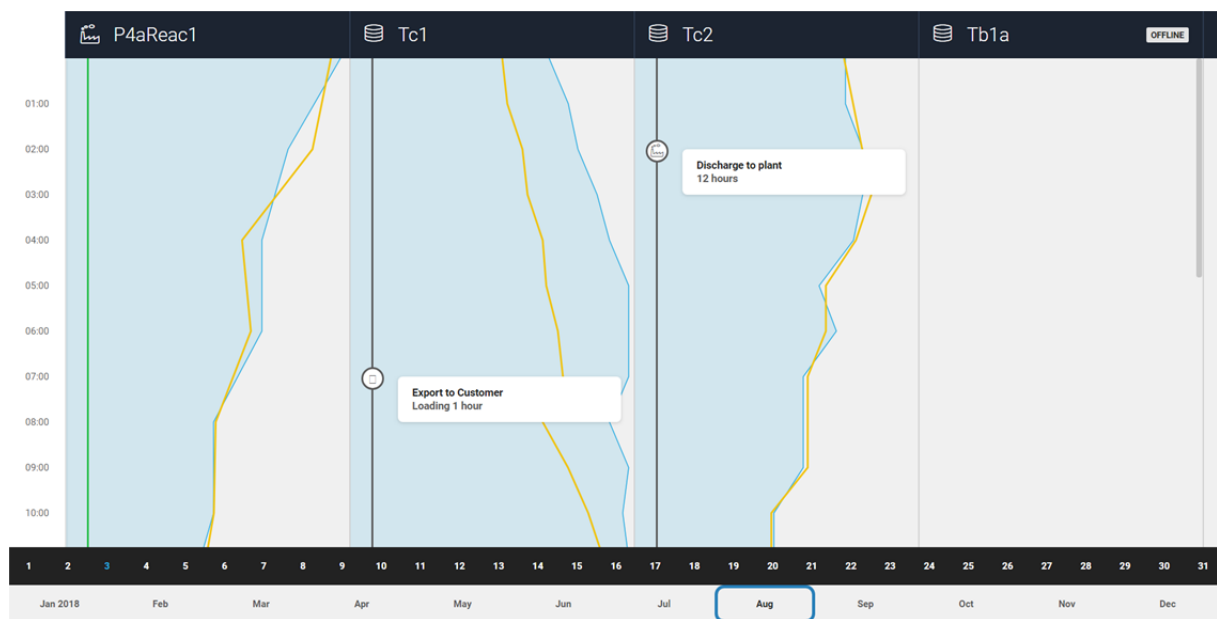


Figure 1: Schedule view screen of the prototype Human-Machine-Interface (HMI) delivering continuous as well as discrete information caused by operator interactions.

2 Site-Wide Scheduling Tool

The site-wide scheduling tool addresses a complex coordination problem, bringing together information from different systems. Although the task itself is complex, it is important that as much complexity as possible is abstracted away from the end user, while still providing the adequate functionality to satisfy differently skilled operators. This will ensure that the system is easily used, which in turn will help to drive adoption across the business.

2.1 System Inputs & Outputs

An important starting point for HMI development is the consideration of what data is provided as inputs, and what information is given as outputs.

The scheduling system takes in data including:

- Initial conditions of the site
 - Operational state of each plant/unit.
 - Tank/storage levels.
- Appointed logistics information:
 - Raw materials being delivered to the site.
 - Products being taken from the site.
- Constraints around the operation of each plant/unit:
 - Maintenance events.
 - Planned downtime.
 - Rate reductions.
- (Time varying) Pricing information:
 - Electricity
 - Raw material costs
- Weather reports:
 - Predicted ambient temperature.
 - Predicted amounts of precipitation.
- The optimisation objective (e.g.):
 - Profit.
 - CO₂ Emissions.
 - Energy usage.

From this an optimal schedule is determined. The results of this optimisation are:

- The production of each unit.
- The fill level of tanks.
- The status of each unit and tank.

This information would need to be displayed along with any discrete events such as deliveries, exports of product, and maintenance events.

2.1.1 Input Requirements

The HMI should be governed by simplicity and intuitiveness to provide a concise and straightforward user experience. At the same time, the challenge is to enable operations on different levels of abstraction.

The basic functionality should allow unexperienced operators achieve the following:

- The user must be able to enter information for discrete events such as maintenance and logistics.
- These events will have an impact on the constraints of the scheduling model such as:
 - The availability of units.
 - The capacity of units.
 - The timing and size of imports and exports.

Much of the listed input information can be gathered without active user interaction, provided a suitable IT infrastructure layer is implemented. However, the HMI needs to provide the functionality to input this information in case:

1. Such an interface layer is not available.
2. The layer provides information that needs to be edited. For example, due to changes in schedules not reflected in the relevant systems.

Some of the more complex HMI elements are around the users' interactions with the constraints of the algorithms. This is due to the wide-ranging nature of the potential changes. It is therefore beneficial to separate out the potential changes into commonly used sections:

1. Changes to, or addition of, logistics events.
2. Changes to, or addition of, maintenance events.
3. General constraint changes.

The changes to logistics events would have a common format:

- Date of event.
- Type of material.
- Direction of stream (from/to)
- Which tank/storage ID is involved.
- Amount of material.
- Method (train, ship, etc.)

Maintenance events would have a format determined by the unit on which the event is applied, an example would be:

- Date of event.
- Plant/unit or tank/storage ID
- Duration.
- Shutdown required?

- Reduction in throughput (%).

General constraint changes would depend on the constraints available for the selected unit. With this list of constraints, it is possible to show the default value, the current setting (in the case that it is different to the default), and an interactive element to allow its value to be changed.

2.1.2 Output Requirements

The operator should be able to understand the generated schedule, while being able to access all further relevant data if needed.

- Overview of schedule should be on one screen, with the information in one place.
- The system must be able to display the scheduling results listed above.

The outputs of the system must be clear and concise, whilst including everything needed to convey the optimal schedule.

To achieve this, a method of displaying both discrete and continuous information is required. This will allow logistics and maintenance events to be shown along with time series data such as tank levels and reactor throughput.

2.2 Layers of abstraction

The HMI will be layered, providing high level results suitable for all users. If required, more information will be available through drill-downs and other click-through methods to expose more detailed levels of information. This ensures that the complexity of the HMI does not inhibit less advanced users from extracting the information they need, while still allowing experienced users to access the more advanced elements of the scheduling system.

2.3 Timing Considerations

Due to the complexity of the scheduling system, it is not necessarily possible to return the results instantly, where the time taken can depend on the size of the problem and complexity of the system. The HMI therefore needs to account for the fact that users may have to wait some time to receive the results. In order to provide a pleasant user experience the HMI needs to be responsive at all times, even when calculations are running in the background.

The added communication resulting from this separation into frontend and backend should integrate seamlessly into the workflow, e.g., cancellation or rerunning with changed inputs should always be possible and some measure of the progress of calculations should be available to the user. Completion of the computation and availability of results could be brought to the user's attention through alerts, in the forms of platform notifications or emails.

2.4 Validation and optimisation

An important aspect of the scheduling tool is the validation of schedules produced by ERP systems. These schedules could be manually entered into the system, or automatically imported through the data integration framework. The HMI should clearly display whether the schedule is feasible, and if

the information can be returned from the algorithm, highlight the section causing issues in the case that the schedule is infeasible.

The optimisation algorithms will aim to learn about the system, building a knowledgebase of information regarding the common causes of infeasibility. This would allow more feedback to be provided to the user regarding requested schedules.

The HMI should be able to present this information in a clear and concise manner, helping the user to make appropriate changes quickly. These could be highlighted visually, with the potential for text feedback. For example: *given the discharge state of tank T1 and the throughput of reactor R1, the selected maintenance window is not be possible.*

2.5 What-If? Scenario Testing

As the tool will be used for exploratory scheduling, the HMI needs to accommodate “What-If?” scenario testing. This allows users to enter one or more batches of changes, representing possible schedule changes. The scheduling tool then calculates which of the proposed schedules is optimal.

It is important that the HMI provides functionality to enable users to make multiple changes to a schedule. They should be able to temporarily save the set of changes, so they can put together an alternative set. Once the user has built the change sets required for comparison the scheduling tool can show which is optimal. This batching of computations makes it efficient for advanced users to explore different potential schedules quickly.

It may also be useful to have visual feedback on the comparisons so that the distance in optimisation space between the potential solutions can be shown.

2.6 Communicating Changes

Schedules are not created and consumed by the same person, the results therefore need to be communicated to the relevant people.

Users must be able to “accept” a specific schedule. This action sets the selected schedule as the live plant schedule and communicates out the changes.

2.6.1 Minimising Communication

Plant personnel should only be notified of an updated schedule if their specific area is affected. This reduces the amount of communication in general, and importantly, eliminates notifications that require no action and provide no useful information.

2.6.2 Limiting Locations in Visualisation

To ensure people are only shown relevant information, the visualization aspect of the HMI should be controlled by groups of users so that groups only see the areas of the site they are concerned with.

Users can see multiple areas by joining several of these user groups.

2.7 Changing the Optimisation Objective

Users should have the ability to change the objective function to re-weight the overall objective based on specific business needs. This would allow temporary emphasis to be applied to, for example, CO₂ emissions or energy usage.

2.8 Communities & Security

The site schedule is an important entity that dictates the high-level activities of plants. It is therefore crucial that it is only edited by authorised people. A community structure is required to ensure that the appropriate levels of access are given to each user of the system.

Community	Access Rights
Schedulers	<ul style="list-style-type: none"> • View schedule. • Perform “What-If?” scenario testing: <ul style="list-style-type: none"> ○ Enter/Edit Logistics Events. ○ Enter/Edit Maintenance Events. ○ Change Constraints.
Senior Schedulers	Same access rights as Schedulers. Plus, the ability to accept new site schedules and communicate them out to the plants.
Plant Operators / Engineers	<ul style="list-style-type: none"> • View schedule. • Perform “What-If?” scenario testing: <ul style="list-style-type: none"> ○ Enter/Edit Maintenance Events.
Senior Plant Operators / Engineers	Same access rights as Plant Operators / Engineers. Plus, the ability to accept new schedules for their plant/area ONLY.

This community structure would also be used to satisfy the requirements in Section 2.6.2, limiting the visualised areas to the relevant site sections for each community.

3 Flexibility of Solution

The HMI developed for the CoPro project should fit the specified use-case(s) within the project, but also be applicable to other industrial sites and industries. To accommodate this, the HMI elements should be configurable by suitable knowledgeable persons to reflect the specific needs of the site. This would include the ability to change the following:

- The default event types that can be added.
 - Maintenance and Logistics will suit most, but not all industries.
- The time series shown along with the discrete event data.
- The community structure and permissions.

The HMI should be developed using technologies and standards that allow it to be used with a wide variety of data pipelines and systems, ensuring that it can be easily installed and configured at any industrial site.