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	D4.4
Deliverable title	Modelling guideline document and demonstration development kit
Version	1.0
Dissemination level	Public
Work package	WP4
Authors	VTT
Contributing partners	TUDO, OPT
Delivery date	27.3.2018
Planned delivery month	M18
Keywords	workflow, modelling, requirements, use cases, LCA, KPI, social perspectives, legacy model, main action, model simplification

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	Created	VTT	21/04/17
0.2	Modelling chapter text	VTT	→ 31/12/17
0.4	Overall workflow text	VTT	→ 26/01/18
0.4	Major revisions to overall workflow text	VTT, TUDO, OPT	→ 21/02/18
0.5	Final touches before internal review	VTT	→ 28/02/18
0.6	Corrections by the reviewers' comments	VTT	→ 25/03/18
1.0	Finalisation for submission	VTT	27/03/18

EXECUTIVE SUMMARY

This document helps a stakeholder of an industrial plant to assess, whether the COCOP methodology fits in a targeted process, and guides in the system development, commissioning and maintenance. The guideline addresses the question "What needs to be done when the COCOP concept is applied to a given plant?" from several different angles. Tools to estimate the current state in the target plant and identify key factors for a successful COCOP commissioning are given. COCOP pursues the use case approach to establish distinct and practically oriented steps to lead the development.

Key Performance Indicators (KPIs) are developed for setting a baseline, quantifying the benefits, and also, for thoroughly discussing and understanding the priorities of the targeted solution. As special characteristic and novelty in the development process, COCOP applies Life Cycle Assessment (LCA) and human factors and social perspectives; this guideline gives tools and ideas in terms of incorporating these approaches in the system development phase, and also in the operational use. A successful COCOP solution calls for a mindset change among the plant personnel.

As computational simulation models are in the core of the COCOP approach, this guideline helps for pursuing proper models for the COCOP optimisation. We introduce and use concepts of Legacy Models and Main Actions to outline different models and modelling needs, and to process them for the use in COCOP. While it is most reasonable to use the models developed to other purposes, the computationally intensive nature of the COCOP simulations often forces the developers to Upgrade, Transform and Simplify these Legacy Models. The Main Actions are discussed and guidelines provided. The model simplification is shown to be highly an application specific task, having no general solution available. Thus, different methods and related applications are broadly reviewed, and a decision flowchart given to assist in selection of the most suitable method for different needs. Selected methods are demonstrated by illustrative examples.

An updated version of this guideline, D4.6 "Modelling guideline document and demonstration development kit (update)", will be published later during the project (M28). It will present updates in the topics of this document, and cover the implementing aspects of the system as well. Using the holistic approach, COCOP strives for outstanding technical capability, but as importantly, for full acceptance of the plant personnel and environmental sustainability. Fulfilling the conditions guarantee significant benefits, also in financial respect.

ABBREVIATIONS

Abbreviation	Full name
ANN	Artificial Neural Network
APC	Advanced Process Control
ACT	Advance Control Technology
DACE	Design and Analysis of Computer Experiments
DCS	Distributed/Digital Control System
FMI	Functional Mockup Interface
FMU	Functional Mockup Unit
FSF	Flash Smelting Furnace
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
KPI	Key Performance Indicator
LCA	Life Cycle Assessment
ODE	Ordinary Differential Equation
OPC	OPen Connectivity via open standards (formerly: OLE for Process Control)
PDE	Partial Differential Equation
POD	Proper Orthogonal Decomposition
PS-Converter	Pierce-Smith Converter
RBF	Radial Basis Function
REI	Resource Efficiency Indicator
RMSE	Root Mean Squared Error
SCADA	Supervisory Control And Data Acquisition
SCF	Slag Cleaning Furnace

Abbreviation	Full name
SUS	System Usability Scale
SVM	Support Vector Machine
SW	Software
UI	User Interface
WP	Work package

TABLE OF CONTENTS

1 Introduction 1

2 Overall workflow 2

2.1 Feasibility 2
2.1.1 Need 2
2.1.2 Benefit 3
2.1.3 Current state 4
2.1.4 Gaps 6

2.2 Design of customisation 7
2.2.1 Use cases and setting a baseline 7
2.2.2 Requirements 10
2.2.3 Action plan for implementation 14

2.3 Implementation 15

2.4 Testing 15
2.4.1 Off-line testing 15
2.4.2 On-line testing 19

3 Modelling 20

3.1 Workflow 20

3.2 Legacy Models and Main Actions 21

3.3 Proper modelling 22

3.4 Model simplification 24
3.4.1 Introduction 24
3.4.2 Surrogate modelling 26
3.4.3 Other views to simplification 33

3.5 Model inclusion 36
3.5.1 Model inclusion of online LCA with FMI example 37

3.6 Model upgrading 39

3.7 Model transformation 39

4 Conclusions and further work 41

References **42**

Appendix A: Example of a use case **48**

Appendix B: Example of a requirement **51**

Appendix C: Sub-classes of the unrevealed proper modelling techniques **52**

Appendix D: Application examples from the literature **55**

1 Introduction

As defined in the DoA, this deliverable (and later the updated one D4.6 "*Modelling guideline document and demonstration development kit (update)*") will elaborate on

1. "a generic guideline for the modelling work related to decomposition-coordination optimisation of process operations, making the optimisation approach of the project more usable after the project in other processes
2. exemplify development of new or integration of existing software tools to be used to transform (legacy) models of different principles to the necessary form via e.g. such co-simulation approaches as FMI/ FMU standard and Simantics platform "

During the project implementation it has become evident that the guideline should not only relate to models, but rather try to answer the question: "What needs to be done when the COCOP concept is applied to a given plant?". One of the reasons for this is that applying the COCOP concept is a large project. It affects a plant in several ways: introduces new software and hardware; affects the plants sociotechnical system, e.g. introduces new ways of working; and requires many people to be convinced of the expected benefits. Thus this deliverable is not restricted to the pilot cases (steel and copper), and is not restricted to modelling, but will rather try to take into account the generalizability of the model-based decomposition-coordination optimisation. Also, some demonstrative example studies and implementations were conducted and they are briefly presented. The deliverable is divided into two major parts. The first "Overall workflow" deals with the question above, whereas the second part "Modelling" delves into the modelling and simulation issues. Some techniques are also demonstrated with small studies conducted. Supporting tables and examples were placed in Appendices in order to facilitate readability.

2 Overall workflow

The overall workflow when implementing the COCOP concept to a new plant follows a quite straightforward automation delivery project workflow, which is split into four phases according to the figure below.

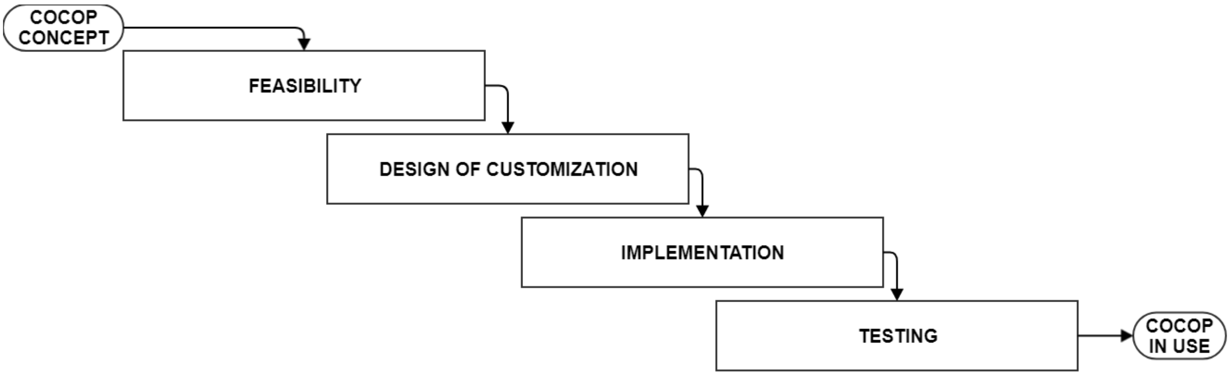

Figure 1 COCOP workflow phases

In each of the four major phases outlined above, several issues need to be considered. The current understanding of such are outlined in the subsequent sections.

2.1 Feasibility

The feasibility phase can be broken into several analysis steps: Need, Benefit, Current state, Gaps. Firstly, for the COCOP approach to be feasible at given plant, there must exist a need for it. Secondly, in order to proceed COCOP should bring some benefit to the plant, personnel or parent company. Thirdly, to apply the COCOP approach, the current state needs to be analysed and possible gaps between it and the requirements of COCOP must be identified.

2.1.1 Need

The first step is to be executed immediately when application of COCOP methodology is considered at a given plant. The goal is to recognise whether the actual situation at the plant (e.g. a specific problem or overall need for improved operation) needs the COCOP approach in order to be solved. The situation may be that, initially application of the COCOP approach seems promising, but a closer inspection should be conducted. This step can be represented as the following flowchart.

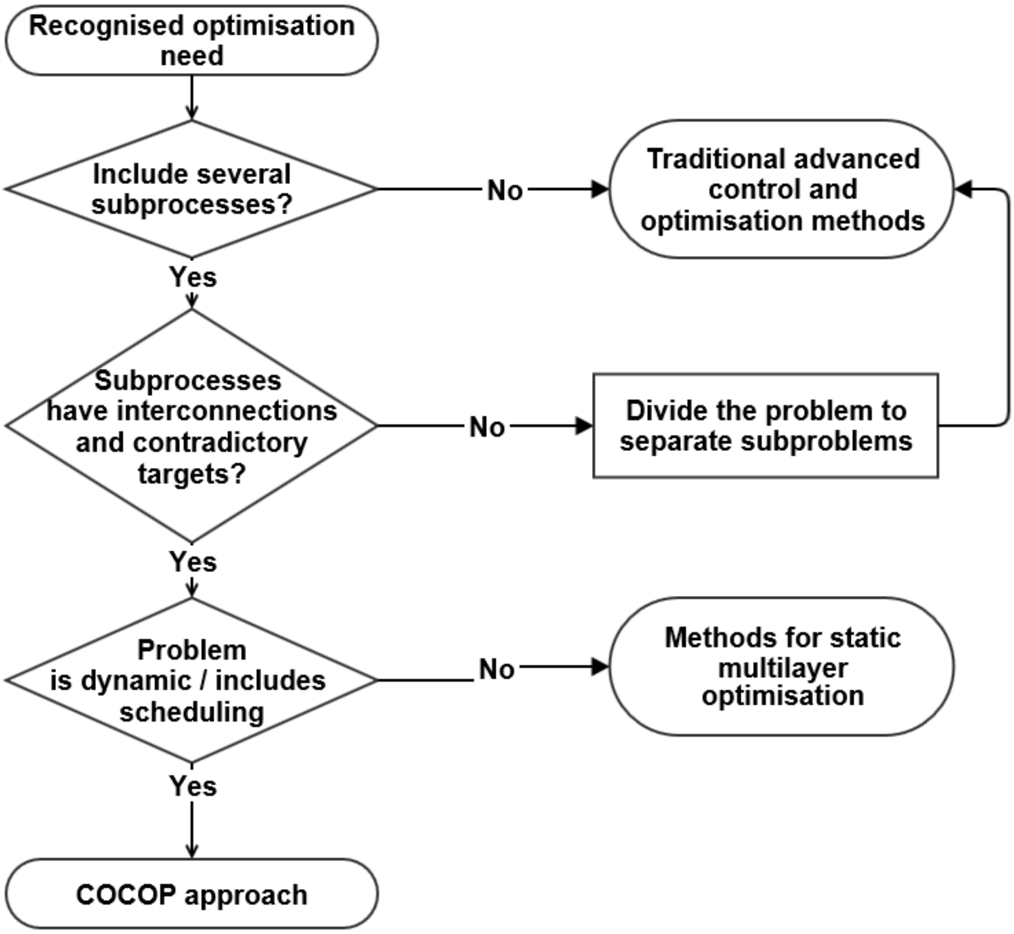

Figure 2 Flowchart to determine suitability of COCOP

First in the flowchart, a check is made whether the situation involves more than one subprocess. If not, then, by definition, the decomposition-coordination approach of COCOP is not suitable and it is advisable to resort to "traditional" approaches. If the situation indeed involves several subprocesses, then the next check is to see whether they have interconnections and involve targets that are contradictory. Both criteria should be met, in order to necessitate the COCOP approach. Note though, that these criteria are not necessarily sufficient to warrant the use of the COCOP approach, because, if subprocesses are interconnected but the whole optimisation problem is so simple that it can be solved in real time, then COCOP approach is not needed. Finally, it is advisable to determine whether dynamics and/or scheduling type of problems play a key role.

2.1.2 Benefit

Having established the need for the COCOP approach, next, its potential benefits should be analysed. Typically, monetary benefit is emphasised, but other types should also be considered: environmental, societal, safety, process operation, quality.

To exemplify this, we present two simple benefit estimation from the COCOP project proposal:

> *One [copper] smelter, 5 per cent of EU's copper output: Avoidable economic cost 7 M€/a, avoidable CO2 emission 4 200 t/a (8 per cent reduction)*
>
> *One [steel] plant, 10 per cent of EU's special steel output: Avoidable economic cost 4-5 M€/a, avoidable CO2 emission 26 000 t/a (20 per cent reduction)*

Such first estimates of the benefits can be typically obtained using quite simple calculations and thus should be done early on. Furthermore, the initial benefit estimates can also be qualitative in nature. For example, the effect of COCOP on various aspects can be estimated with a table such as the one below.

Table 1 Estimation of benefits of applying COCOP in the target system.

Effect Aspect	very negative	somewhat negative	neutral	somewhat positive	very positive
Operator work					
Maintenance					
Safety					
Ease of operation					
Environment					
Quality					
Societal aspects					
...					

In the table above, the list of aspects is an exemplary one and should be adjusted to each plant. Here the effect of COCOP is divided into seven categories and if many of the aspects are expected to worsen, application of COCOP might not be the best solution. Note, that if an investment decision in COCOP is made, then more exact measures of the implementation's impact should be defined. This is elaborated further later in this document.

2.1.3 Current state

If the outcome from the previous step is that COCOP approach should be applied, then the next step is to chart out the current state at the plant. To aid in this, the following set of questions can be used as a checklist. This list has been composed from the experiences of the COCOP project team during the initial phases (esp. WP2) of this project. The list includes an analysis of current:

- **Personnel**

1. Who are the envisioned end users?
2. What are the characteristics of the end users' work / task?
3. Are the end users familiar with Advanced Process Control (APC) or optimisation?
4. Would APC or optimisation be acceptable tools for the end users?
5. Would APC or optimisation support the end users?
6. Will a change management process support the implementation of the new sociotechnical system realized in the COCOP approach? E.g.
    a. Are there "thought leaders", who could advance the uptake of this approach, identifiable among the end users?
    b. Are managers appropriate role models for the new thinking of a plant-wide optimisation?)

In this project, this kind of analysis was conducted in tasks T2.1 *“Use case definition and operator work”* and T2.4 *“Operator work and co-creation requirements”*.

- **Process control and IT infrastructure**

1. What is the current automation system (DCS, SCADA, ..) in use, or are there many?
2. What communication interfaces and protocols is the automation system provided with (e.g., OPC)?
3. Is the existing DCS/SCADA's User Interface (UI) functionality sufficient for COCOP? For example, is it possible to present future predictions as trends?
4. What is the look and feel of the current UIs? If the existing DCS/SCADA UI functionality is not sufficient and specialized COCOP UIs need to be made, then they must be similar to existing ones.
5. Are there any existing APC applications, e.g. model predictive controllers? Should and can these be integrated into the COCOP solution?
6. Does the current infrastructure match for the COCOP architecture?
7. What measurement databases are used and how can they be accessed?
8. What kind of IT security policies exist?

In this project, the analysis of the current infrastructure at the pilots was conducted in WP2 and also in WP3.

- **Models**

1. What computational process models exist currently?
2. Are they up-to-date?
3. Are they validated?
4. How are they used, e.g. in closed-loop control or on an engineer's desktop?

In this project, the analysis of existing models and development needs for the pilot cases was analysed in task T4.1 *“Model specifications for case processes”*.

- **Measurements**

1. What process variables are measured?
2. Are the measurements online or sampled/laboratory analyses?
3. What is the measurement frequency of each?
4. What is the perceived quality of each measurement (e.g. trusted, only indicative)? Does the quality change in time (e.g. periodical calibration practises)?
5. What other characteristics there are for the data (e.g. high level of noise, frequent outliers)?
6. How are the measurement data pre-processed and stored, as raw values or minute/hourly averages?

In this project, the measurement issues were looked into in WP2 and WP3. They will also come up in the modelling tasks of WP4.

When gathering this information, it is likely that the COCOP experts would need to make a few visits to the plant. The visits would entail meeting with IT responsible and automation department personnel. And, it is likely that COCOP experts would need to discuss with the automation system provider or with the system maintenance provider. Finally, care must be taken, when forming a picture of the current state, since an ad-hoc answer from an external stakeholder might not be as precise as a careful analysis of the system.

2.1.4 Gaps

From the above two steps one can form a picture of what gaps there are, and use these as starting points in planning of how to customise the COCOP architecture to the plant. Such gaps could be categorised as follows.

Expertise gap: Current envisioned user do not possess sufficient motivation/expertise to use COCOP. Potential remedies include a change management process including:

- Involve "thought leaders" and "role models" in project phases
- Apply co-creation approach in later phases
- Implement training.

Infrastructure gap: Current IT / process control infrastructure is not sufficient or does not match the COCOP architecture. Also, preventive plant IT security policies may exist. Potential remedies include:

- Buy new DCS/APC or other IT hardware (PCs, servers, …)
- Buy new software / development of SW plugins or adapters
- Negotiate with IT department on possible security aspects.

Model gap: There are not all needed models for the COCOP approach. Potential remedies include:

- Apply Main Actions to Legacy Models, see Chapter 3.2.
- Develop models from scratch.

Measurement gap: Not all variables that COCOP approach needs are measured, or the measurements' quality is not sufficient. Potential remedies include:

- Calibrate existing sensors
- Buy new sensors
- Use soft sensors
- Re-arrange data storage to suit COCOP approach
- Select appropriate pre-processing toolset.

In all cases where a gap exists, the situation should be analysed and the cost of bridging the gap estimated. Naturally, if it seems that the cost is high, or the gaps cannot be closed at all, then application of the COCOP approach is not feasible.

2.2 Design of customisation

The second phase of the overall workflow is to carefully plan, how to implement or customize the COCOP approach.

2.2.1 Use cases and setting a baseline

In this phase, writing use cases of the COCOP approach at the given plant may prove useful. With use case, we refer to what that term means in the software engineering community: use case is a list of actions defining the interactions between an actor and a system to achieve a goal. At this stage, the actor can be a human (e.g. an operator) or a software (e.g. the COCOP software). The system refers here to either the plant (or part of it) or the COCOP software. The rationale of writing use cases is that they focus on the use of the system and thus guide the subsequent efforts to the correct direction. Also, they are quite concise and arguably allow for better communication stemming from the use of structured templates and natural language. Thirdly, since the use cases chart out basic flow of events and possible exceptions to it, they can be used to derive explicit requirements for the software, both in normal situations and in exceptional ones. Also, the use case process was harnessed to help in defining Key Performance Indicators (KPIs). Finally, use cases can be used to design tests for the software and form a basis of a user manual. Thus, general use case templates can help in structuring the subsequent work and refining the actual goal of the COCOP approach to the plant. We think that this use case approach can result in a more thorough analysis and clearer estimates of the required development efforts.

An example of a use case from the project's copper pilot case is given in Appendix A. The use case is explicitly linked to several requirements, show as texts like "TUTCOCOPDEV-55".

In addition to helping in charting software requirements, the use cases aid also in other aspects of design of implementation:

- Giving final scope of the COCOP approach. In other words, what parts of the plant are addressed and how they are broken down into manageable subprocesses.
- Estimating the effort needed to formulate optimisation problems both on subprocess and coordination level
- Estimating implementation costs

Finally, they can help in defining a baseline for evaluation of the COCOP approach's effect on plant operations. At this time, the project team should define criteria/KPIs by which the implementation of the COCOP approach is evaluated and finally accepted, since in the design and maintenance of plant-wide control and advisory systems measuring the performance is fundamental. The following sub-chapter delves a bit more deeply on this.

Key Performance Indicators and baseline LCA

It is important to be able to quantify the benefit obtained with the COCOP approach in order to justify the investment. Tools for this are KPIs and Life Cycle Assessment (LCA).

Each defined KPI can be used to explicitly elicit and measure one or several top level impacts achieved with the COCOP implementation and the KPIs can be viewed as a clear and understandable feedback that contributes to the overall equipment effectiveness and environmental goals of the production process. In the first stage, no numerical goals are presented for the KPIs, but rather they provide a way to define, when and how impact should be measured. Background material for the KPIs needs to be gathered from visits at the production plants and meetings with plant and supplier representatives in order to find measurable indicators and defining a baseline. With the COCOP approach, we envision that three different sets of KPIs can be used: *technical*, *social* and *development.* This differentiation brings a structure to the KPI definition, and also simplifies the exchange of KPIs between different process industries and departments within these industries. Performance measures found to be particularly meaningful for the realisation of operational performance improvement have been recently published in the ISO22400 standard. The purpose of these KPIs is to measure the performance of plant operations, and provide decision making support to the enterprise level. When moving to numerical evaluations, the KPIs are calculated using aggregated measurements from the control layer and an analysis needs to be done to define the numerical values, which correspond to desired/undesired performance. To help in this, a baseline must be defined. In other words, the KPI values should be evaluated before the implementation of the COCOP approach. After the implementation and a test period, these KPIs will be re-evaluated in order to sum up the impact. If possible, the analysis should be done automatically by the system.

As far as possible, the KPIs defined in ISO 22400 should be considered and used, yet some modifications might be needed to fit the needs of the plant. It is recommended to use a few understandable and measurable KPIs. Further, the KPIs should be simple and reviewed regularly. The KPIs in the standard were initially aimed at discrete manufacturing plants, but can also be applied to continuous and batch processes. In the COCOP project, we have designed a template for consistency of KPI definition, which was elaborated in *Task 2.3 – "Impact evaluation criteria"* and the resulting deliverable D2.2 "*Impact evaluation criteria"*. We also note that the results of EC funded MORE project (http://www.more-nmp.eu/) form an interesting source of possible KPIs. In that project, resource efficiency was the focus and thus the term Resource Efficiency Indicator (REI) was used. Finally, the KPIs could be a part of the plant wide optimisation solution, i.e. their values would be maximized/minimized continuously. If this is not feasible, it is recommended to use at least some of the KPIs as online indicators, i.e. values that are only shown to the plant operators.

As stated in D4.2: "LCA is a standardised method (ISO 14040-44) to assess the overall environmental impacts through the value chain. LCA includes measuring the individual ingoing resources both process and site levels. Upstream and downstream effects from raw material acquisition, production and use to the end of life cycle stages are taken into account. LCA as a tool is most commonly used in process and product benchmarking and development, strategic decision-making, and environmental reporting and communication. Data for LCA is typically average data collected on yearly basis". With this in mind, the environmental effect of the COCOP implmentation can be assessed with LCA. In principle, there should be two LCA evaluations: before and after the implementation. In other words, the LCA can complement the analysis described above. Finally, a novel development of performing LCA continuously at the running plant is being demonstrated in the COCOP project. This "online LCA" is elaborated in a later Chapter 3.5.1.

2.2.2 Requirements

This section gives a brief overview of requirement elicitation processes by reviewing various sources and by taking into account the technical and human factors (resp. social) perspectives in the formation of requirements.

2.2.2.1 Technical perspective to requirements

For example IEEE Standard Glossary of Software Engineering Terminology states a requirement to be:

1. A condition or capability needed by a user to solve a problem or achieve an objective.
2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document.
3. A documented representation of a condition or capability as in 1 or 2.

In the COCOP project, requirements have been used in going deeply into the pilot processes as well as in definition of the COCOP architecture. During this work, it has become evident that two types of technical requirements naturally arise: general and implementation-oriented. On one hand, there are general requirements, which seem to apply to all situations in which the COCOP approach/architecture is applicable. On the other hand, numerous requirements have been written that are more implementation-oriented and apply more to one plant (e.g. the copper or steel pilot case plants). The general requirements are especially useful in the COCOP project in developing the architecture and the related research, but when applying the finalised architecture to a new plant, their usefulness is smaller. In that situation, which is in the scope of this deliverable, implementation-oriented requirements become relevant. Requirements elicitation is a "process through which the acquirer and the suppliers of a system discover, review, articulate, understand, and document the requirements on the system and the life cycle processes" (ISO, 2011). In the following, the focus is set on the suitability of requirements to develop software solutions in the COCOP project. The COCOP project mainly follows the agile methods approach, see (Beedle et al., 2001), and in this section, we first give an introduction to the challenges of this task. Then we give an overview of the agile methods and traditional requirement elicitation processes. Finally, we make some observations.

In general, requirement elicitation in traditional and agile software development are very different. Roughly speaking, requirement elicitation in traditional requirement engineering is that all requirements are first discovered together with the stakeholders. The requirement elicitation processes in agile methods are based on discovering the requirements together with the stakeholders during the project. Agile methods fit very well to cover human factor requirements. In a first survey, future users and other stakeholders are not able to define their requirements very detailed. At this stage, only roughly described user requirements are collected. Sometimes, requirements are formulated as processes that have to take place in a company to enable successful implementation of a new software (as part of a changing sociotechnical system), e.g. a process how to define and meet new skills requirements. They are not well known right from the beginning. Therefore, it has to be an iterative process that is supported by agile methods.

Agile methods were designed for use in small, single-team projects, but the potential benefits have attracted also large projects to use them (Dikert, 2016). Agile methods are more difficult to implement in large projects, and there is also evidence that agile methods may not be a good fit for large projects (Dikert, 2016). However, neither approach covers how to handle COCOP domain specific requirements like general concept and human factor related requirements. Note that these may be crucial in obtaining a good return of investment for the COCOP domain problems.

A general note, when the agile development approach is taken in a COCOP concept implementation project. The reference (Cline, 2015), an agile method development guide, describes the use of context diagram, feature catalog, use cases, user stories, requirements traceability matrix, release plan, etc.. It is recommended to use processes and tools described in such a guide to a large extent, because they increase understanding, commitment and probability of success, even though, they are, in reality, used to various extent and they also consume time. This is also motivated by the fact that the software implementation part is a crucial part, but not the only part, in a large COCOP implementation project. Furthermore, following the guidelines may open up possibilities to include human factor processes into the development process that further increase customer satisfaction and the value of the COCOP solution.

2.2.2.2 Human factors and social perspective to requirements

The COCOP project develops a software concept to solve the following problem: How to optimally control a complex plant consisting of sub-processes, i.e. to solve COCOP domain problems. The concept covers both technical and human factors related aspects in developing a new software solution for a COCOP domain problem. When developing new software from the human factors resp. social perspective, there are many different focus points:

1. Part of the requirements is related to the interface between the system and the user. The importance of a functional and at least a good-enough user interface is commonly acknowledged and the qualities of a UI, appropriate from the usage point of view, are contemplated in widely known sources such as System Usability Scale (SUS) [https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html] or Nielsen's usability heuristics [https://www.nngroup.com/articles/ten-usability-heuristics/].
2. The human factors related requirements are targeted to the software or system itself. For instance, future users may wish that the functionalities that the system provides support the user in work-related problems. Such a wish as such and, furthermore, the nature of those problems can be identified best by studying the work through operators' experiences and conceptions.

3. The new system is to be used in a specific context, which may facilitate or hinder the usage of the system. If the objective is not only to implement a functional system but also ensure it will be used, also such matters must be taken into account. Thus, some requirements are targeted to this context. In the case of COCOP system, operators should become informed about plant-wide processes in order to understand and accept the suggestions the new system provides. If the future users have no conception of the processes and their interrelations in the factory, then the suggestions that the software provides to users will not make sense to users and the suggestions become easily neglected.

All in all, human factors related and general concept aspects should and can be recorded in requirements. However, there are some differences between elicitation of technical and human factor (also called social) requirements. Related to the human factor requirements, COCOP pilot cases has not started with describing use cases, due to the mere fact it was not the original plan. What has been done within the human factors approach, future users and other stakeholders have been asked to describe their tasks and their experiences with computer based work. Based on that, as well as on human factors expertise, mainly related to beneficial user interface qualities, human factors requirements are being described. After collecting these roughly outlined requirements, it has to be checked, whether they suit to existing (technical) use cases or whether new use cases have to be defined. The features, how to elicitate human factor requirements will be described in deliverable D4.6 to make sure that these requirements will affect the software developing process (it is still work in progress).

2.2.2.3 Requirements elicitation

The following describes requirement elicitation by paraphrasing or citing Cline (Cline, 2015). In agile development, requirements are written with use cases and user stories, which are progressively improved during project iterations, which is what COCOP project is doing. A context diagram should be made before the core development begins, which shows the workflow between the new product and the actors. The context diagram defines the actors, data, and workflows and shows the boundaries for the responsibilities of the project. The workflows are defined at a high level, and show the direction and kinds of data (or control) that the new product provides between actors. The workflows can be broken down into use cases during the requirement elicitation. No technical details are given in use cases because it is a functional requirement: it defines what must be done but not how. The use cases and detailed use cases may be rigorously validated to ensure that the requirements are complete, logically consistent, have no missing data or control flows, and are defined for all user types. If so, they should meet the IEEE 830 standard (IEEE, 2009) to be a requirements specification and consequently avoid a critical deficiency. The product backlog (feature catalogue) is a list of all the identified large-grain scope items within the project, where the features are prioritised. "The release plan is a simple calendar schedule showing when each fixed-length iteration starts and ends. At a minimum, the release plan will have an Iteration 0 (start-up), a number of productive iterations (delivering business value), and a release iteration at the end of the project." Furthermore, there may be hardening iterations, where all technical debt is repaid: refactoring that should have been done, gaps in requirements filled in, necessary documents completed, and any other work that resets the project to full production quality. In iteration 0 of the project, the development and test environments are set up, tools put in place, architecture defined, and the first pass of the requirements completed. The release plan (sometimes called iteration schedule) is a list of approximately two-week calendar intervals that are labelled Iteration 0 to Iteration N. "Each iteration is time-boxed so regardless of the scope being implemented in those iterations, each iteration has a fixed start and end date." In subsequent iterations features from the product backlog are implemented. The described thoughts are provided by Cline in (Cline, 2015).

Let us now consider how to elicit (draw out) requirements for a requirement specification, which is typically used in a traditional software development project. According to Koelsch (Koelsch, 2016), the best elicitation technique is whatever technique that allows you to capture the requirements, hence, it depends on your situation, your environment, your experience, your stakeholders' experience, your judgement, your management commitment, and your timeline to capture requirements. Additionally, he lists the following recommended requirement sources and elicitation techniques:

- **Questionnaires/surveys**
- **Group meetings**
- **Interviews**
- Following people around/observation
- **Models (e.g. Data Flow Diagrams and Unified Modeling Language)**
- **Use cases/scenarios/user stories**
- Document analysis
- Request for Proposals
- Prototyping of requirements
- Work in the target environment
- **Reverse engineering**
- Tools (e.g. user stories and requirements management)

Thus, this gives an overview of the different methods that can be used, but simultaneously implies that there is no general best method to elicit requirements. In COCOP the techniques shown above **in bold** have been employed, more or less rigorously.

2.2.2.4 Observations

Let us now make some observations. A traditional requirement specification is not part of an agile method process. Regarding traditional requirement specification, the human factors approach is applied at the end of the design process if at all, to correct usability related flaws or to add some nice or good-to-have qualities to the object in question, related to superficial features in the user interface. The situation is somewhat similar with agile methods - concept, human factors and non-functional requirements have no clear place in the development process. A solution that considers all these points is needed. We note that a more rigorous requirement validation avoids critical deficiencies and may be considered in complex or large projects. We also note that requirements discovery and validation may be performed by someone else than software developers. Furthermore, we observe that it may be favourable to write use cases instead of functional requirements because use cases are an integrated part of agile method processes. Hence, it may be worth to consider that requirements can be turned into use cases whenever it is possible. Concept, non-functional and human factors related requirements may be scheduled to be revisited during a suitable stage in the agile methods development process, e.g. during hardening iterations. Initially, the main human factor related requirements are defined. During the requirement revisiting time points the requirement implementation progress is reviewed and, furthermore, the requirements are improved with contextual and more detailed information, acquired through becoming more familiar with the context related demands. For example, new skills requirements become more and more clear during the project. So, they can be translated in new or revisited training concepts. Tentatively, the first human factor related main requirements could be as follows:

1. The system and skills development is of the kind that the user is able to use the system (personal abilities vs. system related requirements); for example, the usability related qualities of user interface or curricula for operator training
2. The user wants to use the system (personal motivation vs. system related requirements); for example, the meaningfulness of functionalities from the users' perspective, or the meaning of the system for the user (supporting the user performing his/her task or threatening his/her job), or that in the organisation, the usage of the system is supported or awarded in some way
3. The usage context is of the kind that the user is able to use the system (context vs. system related requirements); for example, the work does not proceed in such a high pace that there would not be sufficiently time to use the software and do the related decisions, or face-to-face meeting between installation managers take place to support coordinating the sub-process related optimisation

Finally, as an example, in Appendix B, we give on functional requirement from the steel pilot case (see Deliverable D1.2 "Use case definition document" for details). The requirement has been written into a requirement management system called JIRA. Use of such tools is recommended.

2.2.3 Action plan for implementation

A detailed timetable for the implementation and testing tasks must be specified. At this stage, the COCOP implementation will likely follow normal practices of automation implementation project at the plant, but adjustments to facilitate the COCOP approach may have to be done. At the time of writing this deliverable, the present COCOP project has not yet progressed to this phase, and thus more detailed discussion is deferred to D4.6. Nonetheless, the initial thoughts on implementation are shortly presented in the next section.

2.3 Implementation

Implementation of the COCOP approach ideally will proceed on several tracks in parallel, shown in the following table.

Table 2 Implementation tracks for COCOP

Track	What	Who
Measurement	Install new sensors, if needed Configure data collection Store collected data Analyse collected data in order to select appropriate data pre-processing tools	Plant personnel, e.g. automation or maintenance departments, or automation service provider
Modelling	Construct needed models Take already existing models into use Validate models	Modelling expert (quite possibly external to the plant), with the assistance of plant personnel
User	Construct user interface mock-ups for comments from end users Construct final user interfaces Produce documentation, including user instructions Operator/user training	External software development team, with feedback from end users
Software development	Buy or program missing software components	External software development team, plant automation and/or IT department
Integration	Integrate the models, interfaces, data pre-processing tools, optimisation solvers etc.	External software development team and plant personnel (automation and/or IT department)

Furthermore, since these tracks are dependent on each other, it seems to be advisable to apply approaches from agile software development.

2.4 Testing

2.4.1 Off-line testing

Testing of the COCOP software implementation should ideally start as early as possible. One approach to this is to perform off-line testing in a virtual environment. In other words, if there is available a process simulation software during the implementation, the COCOP software should be connected with it in order to perform the tests. Such approach is called *simulation-aided automation system testing*.

Simulation-aided automation system testing allows testing the functionalities, performance, reliability, and usability of the automation system. It allows testing situations that would be dangerous, expensive, or otherwise difficult to test. The main goal is to weed out bugs and gross misconfigurations of the software so that when the software is installed to the plant, it will not produce completely untrustworthy results to the end user. This is important, since the immediate time after the installation is crucial for the end user acceptance of the software.

Simulation-aided automation system testing is not a new technique and proprietary applications have been done for a few decades. However, a breakthrough of the OPC interface that started in the late 90s allowed generic solutions to be developed. For instance, Rinta-Valkama et al. (2000) designed an architecture for simulation-aided testing which they applied for verifying the automation system of a power plant process. That work was continued by Laakso et al. (2005), who took a broader scope for the entire automation delivery process dividing it to the following phases (as shown in the figure below): specification, system design, implementation, installation, commissioning, and validation. Their study concluded that simulation is a useful tool in almost every phase of an automation delivery project; in addition to enabling validation of system requirements and verification of the implemented automation system, it enhances communication between process engineers, automation engineers, and operators.

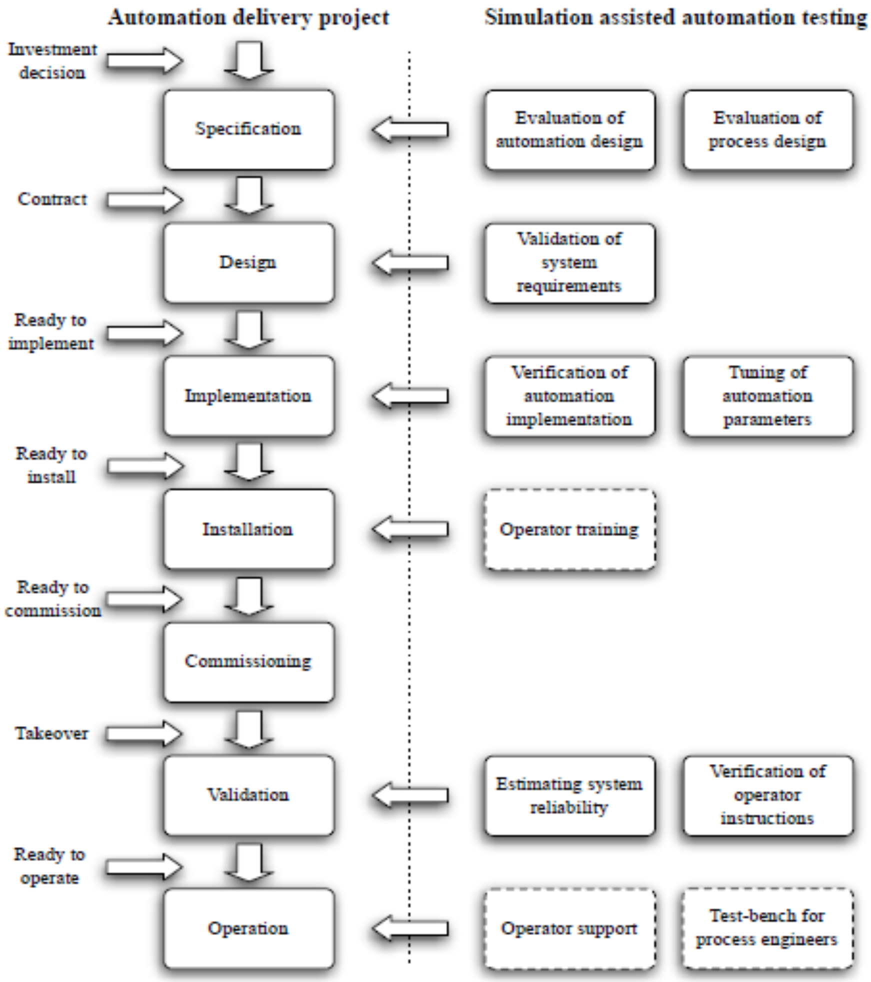

Figure 3 Simulation assisted automation testing in an automation delivery project (Laakso et al. 2005)

In particular, the study by Laakso et al. introduced a tool for semi-automatic simulation-aided automation testing. The work has been applied in many real world cases to test the automation system of different processes, for instance, the tool was used for testing the automation system in the automation renewal project of a nuclear power plant (Tahvonen et al., 2009) and for operator training of the same plant (Näveri and Laakso, 2008). The tool allows creating input sequences for the automation system and the connected process simulator. Once the sequence is configured, the simulation can be run automatically or alternatively the user can interact with the simulation execution. A complete automatisation level cannot be achieved, though, because the result of a run is given in a graphical format, and hence interpretation of the result is difficult to automate. As a solution for that problem, Siivola et al. (2016) have applied metric temporal logic for runtime verification of automation system requirements against a simulated process. Furthermore, Savolainen et al. (2017) combined the approach by Laakso et al. with the work of Siivola et al.: they used keyword-driven testing to simplify the creation of the sequences for the simulation in their testing environment that allowed the requirements of the automation system to be validated automatically after a test run.

Simulation-aided automation system testing has been studied also in other projects: An approach similar to the work by Siivola et al. has been suggested as the future work by Adiago et al. (2015) with addition to use *model checking* as another way to test the automation system. As another example, Peltola et al. (2013) proposed a model-based testing concept, in which the requirements of the automation system were appended to the P&I diagrams, and in which OPC communication is used to access the automation system.

As said, an important task in simulation-aided automation testing is to check that the outcome of the system is correct, which means that the outcome is corresponding to the requirements of the automation system. With complex automation, not all states of the automation application can be tested and, therefore, a systematic approach needs to be taken to allow the automation system testing to have the best coverage of the application. Different heuristics can be used to resolve this issue, for instance, sequences for how the automation system is controlled is extracted from the operating manual of the system.

As the COCOP solution is intended to have the operator in the control loop, the operator would have to be involved in some manner in the testing of the automation system to get a proper response. A straightforward approach for that would be to use a technique presented by Laakso et al.; a rigorous simulation model of the target process is connected with the automation system (including the COCOP system) that the operator can access. In this approach, the automation system and the process model are running in real time with an operator using the automation system (with the COCOP system) as if it was connected with a real process. Additionally, a framework similar to the work of Siivola et al. can be included to allow runtime verification of the automation system against its requirements. A strong point in this approach for testing is the realistic operator action, in addition to which the same environment can be used for operator training as well. The drawback in this approach for automation system testing is that the procedure cannot be always run faster than real time due to constraints imposed by older automation systems. This is alleviated by some newer systems. In addition, with the current COCOP cases, the tests with actual operator in a loop would not be deterministic, which however could be a minor drawback by the side of gaining valuable user feedback and probable easier acceptance. In practise, probably the largest drawback of simulation-aided automation system testing is the relatively high development cost of the system.

As a substitute for using a real operator in the control loop, also the operator actions could be modelled. For instance, a simple operator model would be to apply the suggestions of the COCOP system to the control system as is, potentially after a small delay. The keyword-driven testing technique proposed by Savolainen et al. supports creating such functionality, and hence the requirements for the automation system can be validated against the process model with that technique. A clear drawback in this technique is that the know-how of the operator is disregarded completely. This issue can be relieved by having a plant operator as a follower in the tests, at least occasionally.

Naturally, using Legacy modes in testing the automation system raises the question on what is the relation of the simulator models and the COCOP models inside the software. A general answer is difficult to give, but the relation can be characterised as follows:

- the simulator models typically are slower in computation speed than the COCOP models
- the simulator models may be more detailed (both in comprehensiveness of covered unit processes and physical/chemical phenomena) than the COCOP models
- the COCOP models may be constructed from the simulator models. In this case the simulator models fall under the Legacy Model category.

2.4.2 On-line testing

Between the off-line testing, the software is installed to the plant for the end users for on-line testing. And, a testing period, which may last even a few months, starts. The timeframe is relatively long, in order to ensure that during testing the plant will experience different operating points and transients between them. This facilitates comprehensive testing and allows for final adjustments.

At the final stages of the on-line testing, it is advisable to conduct again the same LCA that was done before the installation. Also, recalculation of the KPIs is done. Comparing the results with the previously calculated allows for a "before and after" type of impact assessment to be conducted.

Finally, the acceptance criteria set forth are checked and upon passing them, the COCOP software enters full operation.

3 Modelling

As the original focus of this deliverable (as per the project proposal) was in modelling, we delve deeper into this in the following subchapters. This deliverable does not describe modelling work done in the project's pilot cases, because that is the domain of D4.3 *"Case process simulation models"* (confidential). Here we are taking a more generic view.

3.1 Workflow

Modelling is not concerned with only construction of the mathematical equations, but it is a rather more involved process. A typical workflow is depicted in the figure below (Hangos & Cameron, 2001)

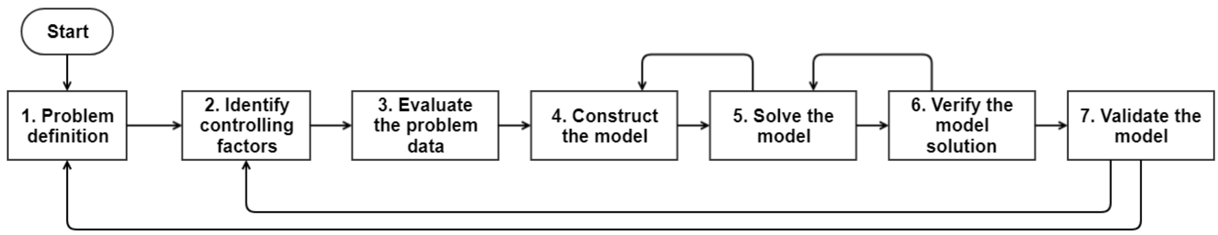

Figure 4 Generic modelling workflow

As is evident from the figure, modelling is an iterative process and it involves typically the work of several people, not just the modeller. Especially important are people who work at or operate the process being modelled and are thus able to guide the modelling work. In fact, the situation is a bit more complex than depicted since the figure ends at a validated model, but this only is the beginning of its use in the COCOP system. Then, the question of model maintenance arises, as experience has shown that over time models and the real process tend to drift apart. For example, there may be gradual changes, such as fouling, in the process equipment. On the other hand, processes undergo revamps which easily cause the model to become obsolete. In the latter case, it is easy to detect that at that point in time the model should be updated, but in the first situation it is not so evident. A possible way to alleviate this would be to have periodic maintenance of the COCOP system and the models therein. This periodic maintenance would entail at least re-estimation of model parameters, but could also lead to more extensive model upgrading. Additionally, any changes in the process that might have influence on the models' performance, should raise a notification to consider and check the models' validity. Examples of this could be a re-calibration of a central process sensor, or cleaning of a heat exchanger. Ideally, this kind of notification could be automatically managed by a maintenance or an Industrial IoT system. In practise, the responsibility must be addressed within the plant organization. Due to the nature of the issues, it would be best to allocate the task and responsibility for a team of maintenance, operating and engineering persons, and include the topic in other periodic meetings. Furthermore, as the COCOP approach is in use at a given plant, personnel may change. Altogether these aspects lead to the conclusion that the models must be well documented, assumptions made explicitly stated, the simulation code clearly written and version managed, and general qualifications for system's faultless operation taken care by the plant organization.

3.2 Legacy Models and Main Actions

In the COCOP approach, the models are at the heart of the concept, but not all kinds of models are suitable. This means that even though, at given plant, there might already exist several models prior to implementation of COCOP, they might not be directly usable. This has led to the concepts of Legacy models and so called Main Actions on them. These are summarised in the table below.

Table 3 Legacy models and Main actions

Legacy Model	Any model that exists before start of the COCOP work
Main Action (on Legacy Model)	Some effort on Legacy Model which makes it a COCOP Model Main Actions are: Include: Use model as is but with added interface to COCOP Upgrade: Extend with new functionality Simplify: Make it faster/less complex e.g. generate a surrogate, delete non-important parts, linearize, discretize Transform: Change but do not simplify or extend e.g. Matlab to C++ e.g. PDE to ODE system, discretize

The goal of the Main Actions is to bring structure and aid in evaluation of needed work load in bringing a Legacy Model into the COCOP concept. It is noted that these actions are not mutually exclusive. The four Main Actions bear a resemblance to what Ersal et al. define as two broad actions on models (Ersal, Fathy, Rideout, Louca, Stein, 2008):

- Model deduction: begin with simple models and increment their complexity until satisfied. This is related to "Include" and "Upgrade"
- Model reduction: begin with too complex models and reduce them until satisfied. This is related to "Simplify" and to some extent "Transform"

In modelling and simulation practise, there always exists a trade-off between accuracy and simplicity of a model. Model accuracy is demanded to capture the real life phenomena in such level of details that ensures the relevancy of the simulation results in the targeted application. Simplicity is generally a desired feature for any system, especially when agile development and implementation, and easy maintainability of a system, is demanded. As mentioned above, increasing modelling accuracy adds complexity due to, e.g., higher diversity of phenomena (e.g. time scales) and increased states, interconnections and parameters. All this leads to the concept of a proper model.

3.3 Proper modelling

A proper model is defined to be a dynamic system model, when it provides the accuracy required for the task with minimal complexity (Wilson and Stein, 1992). Complexity itself could be managed in many cases, but typically it entails slower speed of the model execution. In the context of COCOP, and similar applications, the execution speed is one of the major factors making the model proper or not proper, for the intended task. Thus, in order to reach the properness for the models, we often need to simplify Legacy Models, or even reject them and build up new, simplified models from scratch. An extensive literature review on proper modelling was conducted in (Ersal et al., 2008) and they pointed out that there is no universal proper modelling technique suitable for all modelling problems and all applications. While coming up with a proper model seems to be highly dependent on the modeller, they managed to classify different proper modelling methods they found, and we exploit this work in the following. It is worth mentioning that this classification is not strict; some methods could be placed in more than one class (Ersal et al., 2008). This classification focuses on dynamic system models and deterministic modelling techniques, making it well aligned with the COCOP modelling activities. The following table describes the method classes, lists the references mentioned in (Ersal et al., 2008) and additional ones found in our literature survey. In the table, the first mentioned references are not included in the literature references of this report, due to the large number, instead the reader is advised to refer to the original article (Ersal et al., 2008). To give a rough idea of the number and type of these articles, the references are divided into two groups: articles introducing a scientific method and the ones that are more application oriented. The term realization-preserving (Ersal et al., 2008) is used to characterize the methods, whether they preserve the original states (the dominant ones) or make use of some kind of state transformation and lose the connection with the original states.

Table 4 Classification of proper modelling techniques. The classification and descriptions mostly origin from (Ersal et al., 2008). Note, the references in brackets refer to their references and are accompanied by a rough article type.

Method class	Short description	References
Frequency-based	These modelling techniques assume, in general, that the salient dynamics of a given system occur over a fairly limited range in the frequency domain. Some methods aim to approximate the low-frequency behaviour, some methods are capable to approximate several frequencies of interest, and one method addresses the system's entire frequency response. Most often used for linear systems, yet many can be generalised to nonlinear ones. See sub-classes in Appendix C, Table 6.	See Appendix C, Table 6

Method class	**Short description**	**References**
Projection	These methods assume that the salient dynamics of a given system are limited to a portion of the system's entire state space. They search for this subspace by searching for the basis vectors spanning it; they differ in the ways they choose the basis vectors. See sub-classes in Appendix C, Table 7.	See Appendix C, Table 7
Optimisation-based	These methods typically seek to minimize the L_2, H_2,or H_{inf} norm of the difference between a given full model and its proper counterpart, subject to a constraint on the model order (i.e. complexity). May or may not be realization-preserving.	Method [134–144] Application [138] (Yousefi, 2006)
Energy-based	These methods are built on the intuitive fundamental premise that in an energetic system the most important components to model accurately are those characterized by the largest magnitudes of energy (or power) flow. These algorithms simplify a given model by eliminating less energetic components, while trying to minimize the effect of the elimination on the overall energy flow. E.g. Rayleight-Ritz method, statistical energy analysis, power-based model reduction algorithm	Method [50, 145–148] Application [151–153]

The classes of frequency-based and projection-based proper modelling techniques have several sub-classes each, which are presented in Appendix C. In most cases, the proper modelling techniques are based on simplification, which is why we go into that issue in the next Chapter. We note that the use of various model simplification techniques are often limited by the modelling language or software environment used, as it is not necessarily easy to export or convert the model format into such a form that the simplification algorithms can be used. Also quite many methods are applicable only for linear systems, which is reflected for example in Figure 5. This limits the general applicability of the model simplification methods with respect to COCOP type of applications. Furthermore, many of the techniques seem to provide rather modest potential computational speed-up. Speeding-up a computationally expensive Legacy Model by 2–10 times, for example, is not necessarily enough for the COCOP purposes. Employing a novel mathematical method also contains a risk for errors, until it is well adopted. These issues might get the engineer to give up, unless the model execution time is really critical for the success. In this respect, most potential seems to be in the projection based methods, which rely on extracting empirical data from the expensive model for conducting the simplification. This is a general idea in surrogate modelling and is discussed in more details in Section 3.4.2. Finally, there are methods which seem to have excluded from the above classification, such as ANN, but which we have included in the surrogate modelling discussion.

3.4 Model simplification

3.4.1 Introduction

In modelling and simulation, model complexity originates from different sources, e.g. inclusion of processes that contribute little to model performance, or/and too many state variable (Innis and Rexstad, 1983). Often it brings along the fact that execution of the simulation becomes slow. (Chen et al., 2011) described that in chemical engineering, complexity mainly originates from the physical scales being considered. For example, a plant-wide (flowsheet) model often becomes complex due to a high number of unit processes and connections between them. They give another example of a simple reactor model, based on ordinary differential equations (ODEs), which becomes more complex, if the spatial variation within the reactor is not negligible, but instead partial differential equations (PDEs) have to be applied (Chen et al., 2011). More examples in reaction engineering are easy to find: For example, the number of reactions and intermediate products may become high in polymerisation applications. Auxiliary systems, like heating and cooling, may play a key role, thus demanding inclusion to the model scope.

When having a computationally too expensive, or otherwise complex, dynamic Legacy Model at hand, the first thing to check is whether additional hardware or parallel computation would be a feasible solution. If not, we basically have to simplify the model to suit the COCOP optimisation applications. Innis and Rexstad (1983) have presented a usable flow chart for model simplification. We took their work as a starting point and applied it into the following figure. Some of the steps have been renamed and some of the original paper's steps omitted. Such omitted steps dealt with stochastics, which we have ruled out of the scope for now.

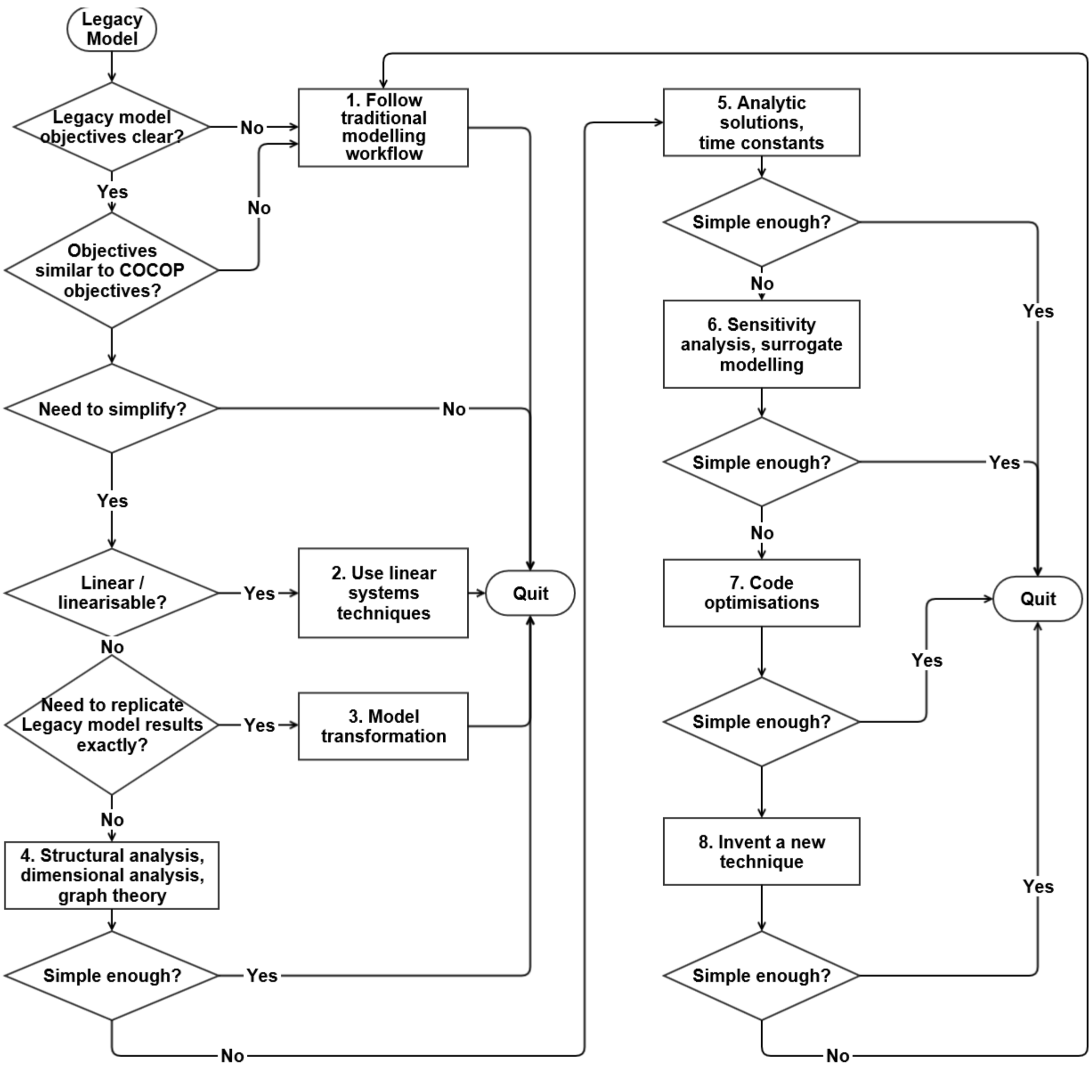

Figure 5 Model simplification flow chart, applied from Innis and Rexstad (1983).

The first two checks in the workflow strive to determine whether Legacy Model at hand is at all applicable in the present situation. If not, then the Legacy Model needs to be abandoned and the COCOP model constructed from scratch (Step 1 in Figure 5). If Legacy Model indeed is suitable, then naturally the need for simplification should be determined. The first actual step of simplification is to determine whether the model is linear or linearisable since there is a lot of methods and tools for those (Step 2). If linearity cannot be reached, other means of simplification need to investigated. Before doing this, another check is to be made: do we need to replicate all Legacy Model results accurately? As Innis and Rexstad note: "If the modeller decides that an exact duplication of original model output is desired, simplification may not be possible". Possibly, then model transformation, e.g. rewriting the computer code, (step 3) may prove useful. Otherwise, Rexstad and Innis recommend to proceed with techniques that are quite formal in nature, for example structural analysis, dimensional analysis or graph theory (Step 4). In structural analysis the idea is to take a look at the model equation and to determine whether optimisations might be possible there, e.g. by converting the model into some "standard format". Dimensional analysis uses so called dimensional variables/groups to simplify the model. This is a bit similar to what has been done in the copper pilot case with the flash smelting furnace composition and temperature estimation. The graph theory approach naturally is applicable to models, where a clear network/graph structure is present. We give an example of this later in this document. Next possible simplification approach is to see, whether part or the whole model could be solved analytically (Step 5), or in the COCOP terminology, transformed. We give an example of this in the Model Transformation chapter of this document. Also, an analysis of the time constants of the model may provide avenues for simplification. Phenomena which are very fast (small time constant) with respect to the time resolution of the simulation can be approximated with algebraic equations describing the equilibrium state. On the other hand, very slowly changing variables can be replaced by constants. In Step 6, techniques such as sensitivity analysis and surrogate modelling come into play. Sensitivity analysis can provide information on inputs/phenomena, which have little or no effect on the model outputs. Step 7 delves into the simulation code employed. Yet compilers or simulation platforms themselves are able to do considerable optimisations nowadays, the impact of selecting the best algorithms and proper code structuring should be kept in mind. Finally, the original paper gives, rather humoristically, as the last option (Step 8) "Invent a new technique". While this is true, it may be difficult to come up with one in a practical situation, where the COCOP approach is being implemented. Thus, it seems advisable to go directly back to Step 1.

3.4.2 Surrogate modelling

Surrogate modelling is always data driven, while many other model simplification techniques work on the mathematical formulation of the model. A surrogate model, sometimes known also as response surface model or metamodel, is a fast-to-evaluate approximation of a computer model output. It is fitted to a known sample of input–output data points, and can be used to predict the output response at untried points/configurations (Beck et al., 2015). These snapshot-based techniques are widely applied for nonlinear systems. In other words, a surrogate model provides a way to run the time-expensive, high-fidelity model fast and accurately enough for some specific need. It seems that surrogate modelling is most commonly used for spatially oriented applications, such as design optimisation in aerospace engineering. Even though many applications of surrogate modelling are related to static applications, they are looked into in COCOP. This is justified since in many cases the available Legacy Models may contain static relations (e.g. due to very fast dynamics). After reviewing surrogate modelling methods from a number of references, including (Zheng et al., 2015), (Vincenzi and Gambarelli, 2017), (Bremer et al., 2017), (Song et al., 2017)) we can list the most common methods as follows: Kriging and its different variants, Artificial Neural Networks (ANNs), Polynomial response surfaces, Proper Orthogonal Decomposition (POD), Support vector machine (SVM). It is noteworthy that (Ersal et al., 2008) included also surrogate modelling methods in their subclass of Karhunen-Lóeve expansion, but do not use the term “surrogate model” at all. However, many of these surrogate methods can be classified as projection based methods. It is worth mentioning that ANN modelling was excluded from the Ersal's classification, yet it is widely used for dynamic systems as well.

3.4.2.1 Surrogate model type selection

A key question in surrogate modelling is that what type of model to choose. This question has gained lot of attention in the literature. It is common that the modelling method needs to be changed or updated during its use. (Rocha, 2009) demonstrated with the popular radial basis interpolation methods that it is dangerous to make a priori selection of the function, comparing cubic spline, thin plate spline, multiquadratic and Gaussian functions. They concluded that the choice of the Radial Basis Function (RBF) should be part of the optimisation problem. This same message is clear throughout the literature, for example (Xia et al., 2016) advise that Kriging with fixed basis function is not adaptable to complex engineering problems. (Peng and Wang, 2016) pointed out that if any of the sample points used to construct the model are too close, the correlation matrix can become singular, and large amounts of computer memory and CPU time are needed for constructing models with high dimensionality. Accordingly, various methods to select the best model and adapt the model according to the needs have been published. For example, different extensions have been introduced to Kriging, which is probably the most commonly used surrogate modelling method, to address the selection of the sampling and basis function, e.g. (Xia et al., 2016). A highly valued feature of Kriging is that it provides an estimate of prediction variance, which is a useful for adaptive sampling strategies in surrogate-based optimisation (Rogers and Ierapetritou, 2015). (Pilát and Neruda, 2013) presented a framework, which in each generation (i.e. step when updating the model) selects the most suitable surrogate from a set of models based on some pre-defined criteria. Four different model selectors are used, one being the most obvious approach, using the lowest mean square error (MSE) on a validation set. Another one used the bias and variance of the error on the validation set, and one of the selectors used the so called relation preservation. (Pilát and Neruda, 2013) emphasize that the best model may be different in different independent runs of the algorithm, so some kind of automatic selection is necessary to ensure good convergence speed and the quality of the individuals. This frequent need of model adaptation, more or less autonomously depending on the application, actually categorises the surrogate modelling into the domain of machine learning.

The figure below gives a simple illustration of one important aspect in the model selection in optimisation applications. Namely, the model with the smallest prediction error does not necessarily lead to find the correct location of the optimum. In the example, the RMSE for the Surrogate 1 (green) is more than 9 times that of Surrogate 2 (red), but still Surrogate 1 would lead to a significantly better estimate for the optimal decision variable value i.e. location of the minimum on the x-axis.

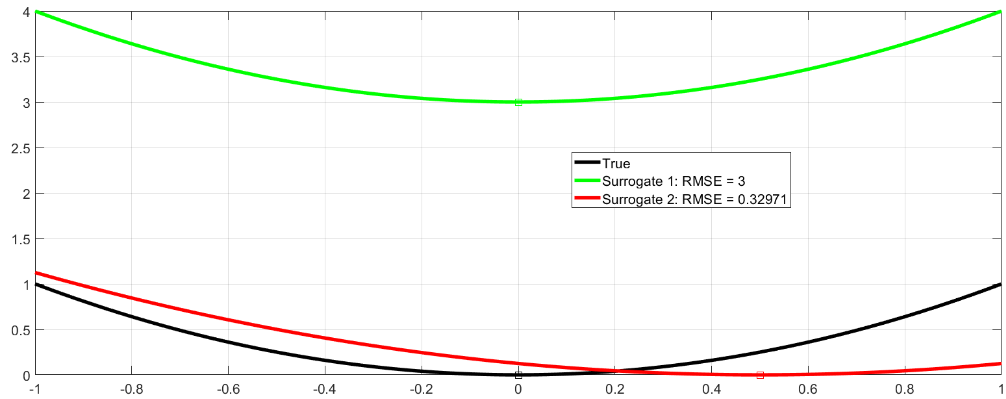

Figure 6 An example of model fitness evaluation. The model with the smallest mean square error (Surrogate 1 in this case) does not necessarily lead to the optimum.

(Rogers and Ierapetritou, 2015) pointed out that global convergence to a true optimal solution is difficult to guarantee in surrogate-based optimisation. The popular Kriging, for example, is an interpolating method, so the surrogate model cannot be used to extrapolate beyond the range of available design sites. Similarily, (Fouladinejad et al., 2016) emphasized that the surrogate model was capable of capturing the essential system dynamics in pre-defined parameter ranges and the scenarios applied, but the accuracy is presumably lower for different scenarios. And, (Bremer et al., 2017) concluded that the performance of their POD-DEIM type surrogate model does not exclusively depend on the surrogate itself, but also on the operational conditions and scenarios it is used for. Also, developing an accurate reduced-order model for a process with a large number of uncertain parameters is problematic, particularly if the feasible region is nonconvex with respect to several of these parameters.

Even though Kriging was originally developed for static processes, it has been applied for dynamic processes as well. The term dynamic Kriging is understood in different ways in literature. One way to categorize the different types of research could be as follows:

1. A method of how to select the design sites that are used for Kriging dynamically (e.g., Kleijnen, 2009)
2. Simulation of dynamic processes with outputs and controllability in only small number of points in time or space (e.g., Rogers and Ierapetritou, 2015)
3. Controllability after each time step (e.g., Hernandez and Grover, 2011; Amicarelli et al., 2014; and Shokry et al., 2015).

The first definition for dynamic Kriging is *not* for dynamic simulation. In the second case, the method of how Kriging is applied leads to the curse of dimensionality as the number of training samples grows very fast. The last one is of interest here, as Kriging is applied iteratively so that its time complexity is linear in respect to the size of the states and time steps. The results from the iterative method also seem convincing.

Surrogate modelling clearly provides good potential to simplify expensive simulation models for time-critical applications like optimisation-based process control. This approach relies on data samples and success in applying the method greatly depends on the sampling strategies and the models used. The studies reported show that most cases have reached their accuracy targets with a simulation speed-up. However, from optimisation point of view, the speed-up is often quite modest, so that more than 10x faster implementations are in minority. It is worth emphasizing the surrogate models' incapability to perform in exceptional conditions that many authors are pointing out. This is not surprising for a data-based methodology, and must be carefully considered in the implementation.

3.4.2.2 Application examples and available software

This section gives further insight to the model simplification and surrogate models through software tools available. Use case examples reported in literature are summarised in Appendix D. The programs that were considered most interesting are listed in Table 5. It must be noted that plenty of other tools exist, many of them freely downloadable, and furthermore, the situation is in continual change. So it is highly recommended to make a search for the specific topic, for example in the Internet site of MathWorks/Community.

Table 5 Software tools available for model simplification, surrogate modelling and supporting the use of simulation for optimization purposes.

Software	Description	Note
Matlab/Robust Control Toolbox	10 functions including e.g. Balanced Truncation, Hankel minimum degree approximation, Modal form realization.	Commercial
Matlab/Statistics and Machine Learning Toolbox	Functions to describe, analyse, and model data using statistics and machine learning. Regression and classification algorithms (including Kriging) to draw inferences from data and build predictive models. Functions to identify key variables or features that impact the model with sequential feature selection, stepwise regression, principal component analysis, regularization etc. Provides machine learning algorithms, including support vector machines (SVMs), boosted and bagged decision trees, k-nearest neighbour, k-means, k-medoids, hierarchical clustering, Gaussian mixture models, and hidden Markov models.	Commercial
Matlab toolbox DACE (Design and Analysis of Computer Experiments)	For working with Kriging approximations to computer models. To construct a Kriging approximation model based on data from a	Free of charge

Software	Description	Note
http://www2.imm.dtu.dk/projects/dace/ Also available: DACE for Scilab Kriging toolbox https://atoms.scilab.org/toolboxes/dace_scilab	computer experiment, and to use this approximation model as a surrogate. Also addressing the design of experiment problem, i.e. choosing the inputs at which to evaluate the model for constructing the Kriging approximation.	(the changelog ends 2002) The Scilab version created 2012.
MATLAB toolbox ooDACE (object oriented Design and Analysis of Computer Experiments) by Ghent University http://sumo.intec.ugent.be/ooDACE	For building Kriging surrogate models of a given data set. The models can be used efficiently for design automation, parametric studies, design space exploration, optimisation, yield improvement, visualisation, prototyping, and sensitivity analysis. The ooDACE Toolbox provides a flexible implementation, easily extendable and well-suited to test and benchmark new Kriging flavours.	Both commercial and Open Source versions Matlab 2015a or newer
Matlab toolbox SUrrogate MOdeling (SUMO) Toolbox by Ghent University http://sumo.intec.ugent.be/SUMO	For automatically building accurate surrogate models of a given data source within the accuracy and time constraints set by the user. The toolbox minimizes the number of data points (which it selects automatically) since they are usually expensive.	Both commercial and Open Source versions
Matlab toolbox mGstat (Geostatistical Matlab Toolboox) http://mgstat.sourceforge.net/	Provides geostatistical algorithms including Simple Kriging, ordinary Kriging and Universal Kriging and interfaces to some other geostatistical software.	Free of charge (apparently)
GPML Matlab code by C.E. Rasmussen and H. Nickisch (http://www.gaussianprocess.org/gpml/code/matlab/doc/)	Provides the Gaussian Process algorithms demonstrated in the book "Gaussian Processes for Machine Learning" by Rasmussen and Williams (2006). A link to other available Gaussian Process SW given.	Applicable for Octave and Matlab. Released under FreeBSD License
Python Kriging Toolbox pyKriging http://www.pykriging.com/	To make Kriging easily accessible in Python. Provides n-dimensional Kriging.	Under GNU General

Software	Description	Note
		Public License v2.0
Kriging Toolbox for Python pyKrige https://github.com/bsmurphy/PyKrige	The code supports 2- and 3-dimensional ordinary and universal Kriging. Standard variogram models (linear, power, spherical, gaussian, exponential) are built in, also custom variogram models can also be used.	Under GNU General Public License v2.0
UQLab (The Framework for Uncertainty Quantification) by ETH Zurich http://www.uqlab.com/	General purpose Uncertainty Quantification framework to carry out uncertainty propagation through Monte Carlo sampling, sensitivity analysis, reliability analysis (computation of rare event probabilities), build surrogate models (polynomial chaos expansions, Kriging, low-rank tensor approximations, etc.), etc.	Free of charge for academic Non-academic licenses available
OptoEcon Tolbox by RWTH Aachen University, http://www.avt.rwth-aachen.de/cms/AVT/Wirtschaft/SoftwareSimulation/~kqzp/OptoEconToolbox/?lidx=1 (Elixmann et al., 2014)	For economic NMPC and dynamic real-time optimisation of chemical processes. For development of controllers and estimators for large-scale nonlinear process models and their arrangement into arbitrary single- or multi-layer control architectures. Includes interfaces to numerical software for formulating surrogate models in different modelling languages, such as Modelica or gPROMS.	Free of charge (apparently)
MOSAIC by Technische Universität Berlin http://www.mosaic-modeling.de/ (Kraus et al., 2014)	Web-based modelling, simulation, and optimisation environment. Based on a LaTeX-style entry method for algebraic and differential equations. Large-scale chemical engineering applications flowsheets, optimisation problems, etc. can be built. Provides an automatic code generation for simulation and optimisation environments, such as AMPL, Aspen Custom Modeler, GAMS, gPROMS, MATLAB, Modelica, and for solvers interfaced via C++, FORTRAN, Python, etc.	Free of charge for academic purposes

3.4.2.3 Surrogate model example

To exemplify use of Matlab and its Statistical and Machine Learning Toolbox, we present a surrogate modelling case from a commercial simulator (Apros, www.apros.fi) model into a Gaussian Process Regression model.

The original simulator model is a chemical reactor with a recycle and two control loops, where a catalytic conversion of CO_2 and H_2 in to methane (CH_4) takes place. While this process is not a pilot case, it has been chosen as an example in order to stress the cross-sectorial potential. The process is continuous and dynamic, and a control-wise challenging one, especially when driven by H_2 produced by electricity obtained from a renewable and intermittent source such as solar or wind power.

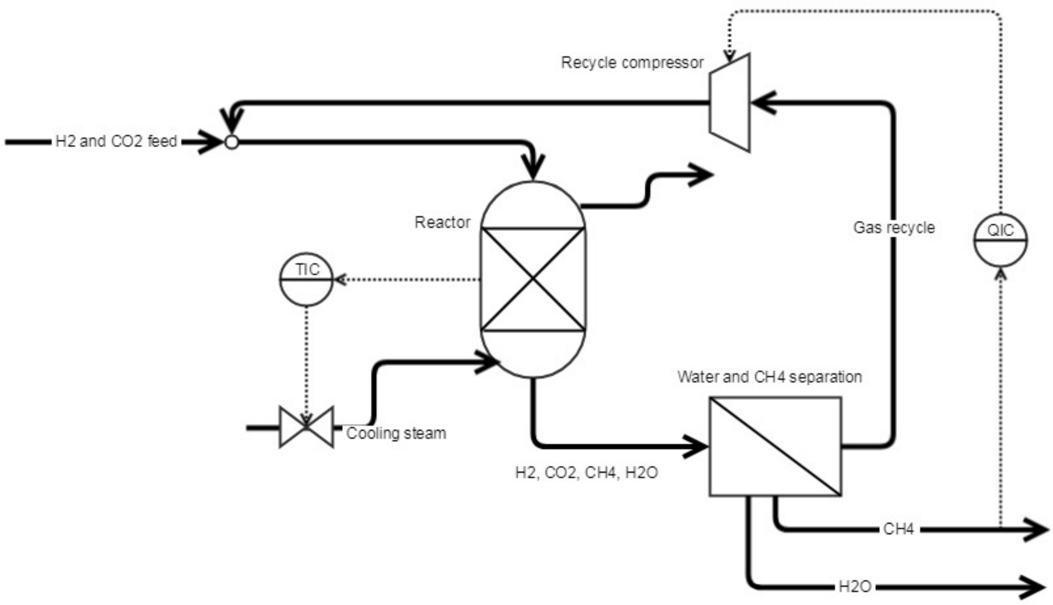

Figure 7 Power-to-gas process

In the original model there are numerous inputs, such as input H_2 and CO_2 feed rates to the reactor, pressures and temperatures. Also, the number of outputs was high, since pressures, temperatures, flow rates, concentrations, energy consumptions were calculated all through the process. The mathematical complexity was also high, since all the flows were described with partial differential equations, chemical reactions kinetics were non-linear, etc. All this resulted in a model, whose execution speed on a standard laptop was less than 100 times real time. Given that scheduling of such a process should have a prediction horizon of one or two days, one simulation would last more than 30 minutes wall clock time. Thus, in order to be used in COCOP-like applications, model simplification would be needed.

First, the relevant inputs and outputs for the surrogate model were chosen. As inputs were chosen: 1) H_2 feed mass flow, 2) reactor temperature controller set point, 3) molar ratio of H_2

to CO_2 at the reactor feed, 4) CH_4 molar fraction controller set point, and 5) reactor pressure. As outputs were chosen: CH_4 mass flow, reactor cooling duty, recycle mass flow, CH_4 purity, compression power, and steam outlet temperature. Next, the inputs were varied over pre-defined ranges according to a Box-Behnken experimental design, and thus input time series were generated and fed into the Apros model. These simulations produced output data that was logged for surrogate modelling. The surrogate model was constructed in Matlab. An example result is shown below. On the left, the CH_4 purity is depicted, and on the right, outlet temperature of cooling steam is shown. The blue curve indicates the training data from Apros, while the red and orange curves are the GPR-surrogate model output and validation data from Apros, respectively. On the x-axis is time in multiples of the data logging time step (10 s). In other words, the slightly over 10 000 data points long trend represents approximately 28 hours of operation. In this demonstration, there were no exact requirements set for the prediction accuracy; different parameters of the surrogate modelling were experimented aiming at qualitatively acceptable (using visual evaluation) result. In this respect, the results were satisfactory, as can be seen in the figure below. These initial results showed a speed-up of 13 times compared to the original simulation.

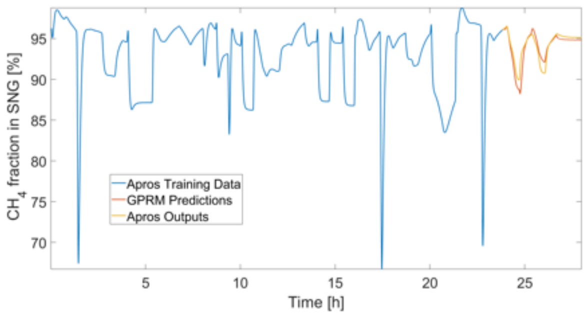

Figure 8 Methane purity surrogate model, training and validation

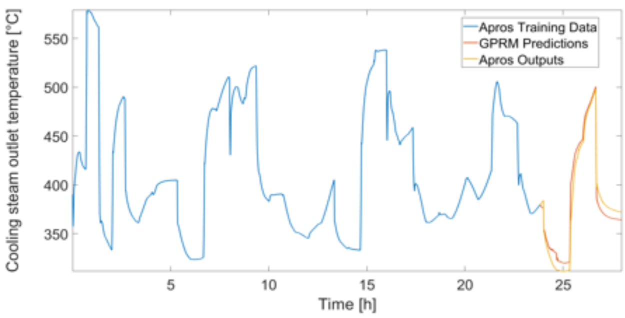

Figure 9 Cooling steam outlet temperature surrogate model, training and validation

3.4.3 Other views to simplification

3.4.3.1 Simplifications in the modelling phase

Instead of considering model simplification as a step of its own, it can be considered as an implicit part of the modelling process. For example, in the polymerisation process the products and intermediate products are characterised with continuous molecular weight distribution. Thus, it might take very large number of states if the molecular weights are covered with high resolution. A method of moments has been used to overcome this complexity source, see e.g. (Disli and Kienle, 2012; Soumitri et al., 2015). In chemical reaction engineering, also the reaction mechanisms are often subject to simplifications. So generally speaking, assumptions done in the modelling phase could be considered as model simplifications. Typical such approximations that significantly help to simplify the model include:

- Flowchart simplifications: reduction of units and/or connections between them e.g. combining separation steps, lumping together tanks and other process volumes, leaving out small volumes and streams in a piping network models
- Lumping in spatial dimensions within a unit process, modelling in one dimension (e.g. neglecting the radial direction in flow channels)
- Ideal mixing i.e. first order dynamics and homogeneous properties of the calculation volumes
- Simplifying geometries, e.g. complicated area/volume is described by a circle/sphere etc.
- Simplified material properties calculation: constant properties, or only temperature dependent
- Reducing the number of chemical components in the system
- Thermal and/or chemical equilibrium
- Reaction mechanisms, neglecting less significant reactions and making fast reactions instantaneous
- Simplified material moving mechanisms, e.g. neglecting momentum balance in flow calculations
- Simplified phase interactions, e.g. ideal or no dissolution/release of gases to the liquid phase
- Simplified heat transfer mechanisms, e.g. use of constant heat transfer coefficient, and/or overall heat transfer coefficient instead of detailed modelling of heat transfer in the surfaces and thermal conduction through the heat exchanger walls
- Energy balance, e.g. cooling/heating approximations like constant heat transfer fluid temperature, or constant heat duty
- Neglecting heat losses to environment

3.4.3.2 Application of sensitivity analysis

Given a complex model, especially of black-box nature, it is intuitive to use sensitivity analysis for assisting model identification, simplification or surrogate modelling tasks to find the most relevant set of model inputs and parameters. The method may be based on interpretation of more or less ad-hoc experimenting and observations, or some of the well-known sensitivity analysis methods. In both ways, the system's dominant physical features (whether real world target or a model) can be identified, and consequently, the needed simulation test campaign and the outcoming model complexity are reduced. Examples are given by (Brown et al. 2008, Moghaddam et al. 2014; Wu et al. 2011). For example, (Wu et al., 2011) used a batch reactor application to demonstrate a model selection criterion for selecting the optimal number of parameters to estimate from ranked parameter lists obtained using estimability analysis. (Moghaddam et al., 2014) used sensitivity analysis method to simplify a generator model. Taking this thinking a bit further, sensitivity analysis can prove indication of which parts of the plant are important or non-important for the modelling or control purposes (see for example (Savolainen, 2013; Santillán Martínez et al., 2016)).

3.4.3.3 Directed graph and graph theory

(Szimandl and Németh, 2015) presented an approach of model simplification, where the resulted simplified models preserve the physical meaning of the variables and parameters, while their complexity is decreased significantly. Their approach has systematic features, while it also seems to be rather application dependent and largely based on engineering judgment and physical insight of the model. They employ a method called directed graph for illustrating the model variables. They also use performance and size/complexity index to evaluate, whether the simplifying goals are fulfilled. The procedure was applied to a dynamic hybrid model of an electro-pneumatic clutch system, reducing the number of states from 16 to 3, which reduced computational time and memory demands. The proposed use for the achieved model was control design. It is characteristic for many general approaches of model simplification that the methodology is domain specific, another such example is (Picco et al. 2014), for modelling of buildings.

As stated previously a graph simplification approach may prove useful in COCOP applications. For example, process industry plants typically involve pipelines that connect and branch, thus making up a network, or in other words, a graph. Such structures also arise naturally elsewhere and as an example of such, a waste water network model simplification is described.

> *In this case study, a dynamic, partial differential equation based two phase flow model is used to describe a small residential area, where waste water (sewage) is produced from several houses, and then transported in a flow network towards a waste water treatment plant. The partial differential equations describing the flows are automatically spatially discretised based on network topology data obtained from the network operator. This discretisation is then simplified in order to speed up the calculation. The basic idea is to treat the sewage network as a graph. The key to simplification is to remove vertices with exactly two edges connected, because these represent spatial discretization nodes within one sewage line. In other words, they are not sources of sewage water and they are not connection points of two sewage lines, in which cases they could not be removed. When removing such a vertex, also one edge is removed, or to be more exact, it is combined with another edge. Since each edge represents a pipe with a definite length, the removed edge's length is added to the remaining one. This is shown in the following figure.*

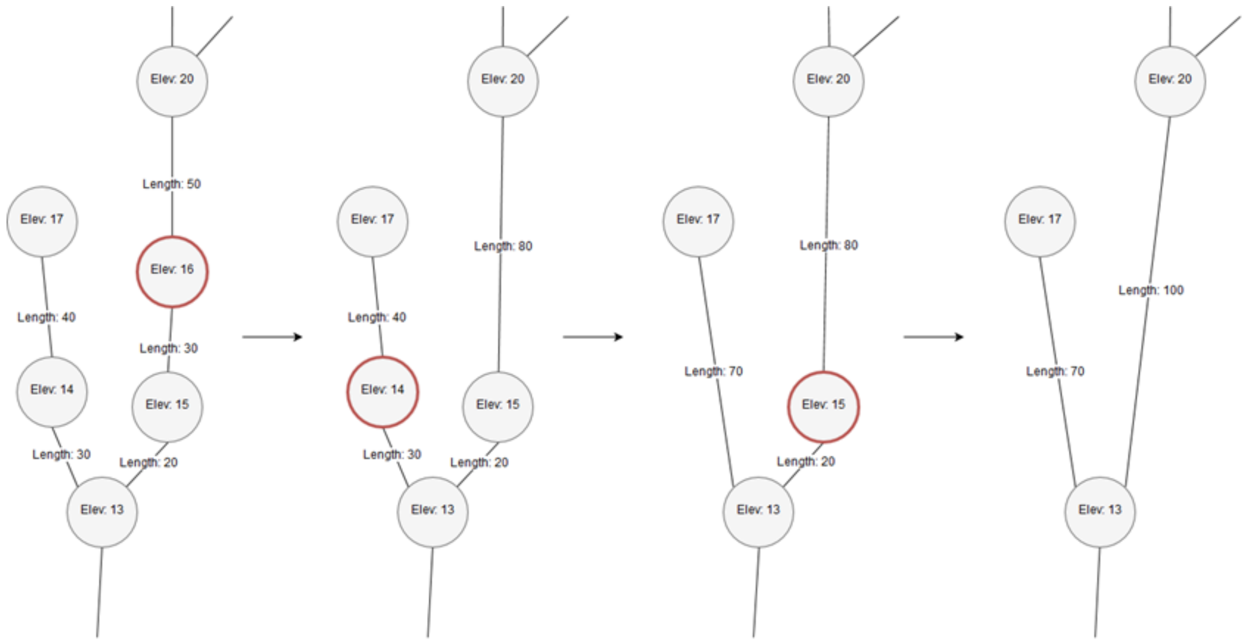

Figure 10 Sewage network simplification example

In this study, a network of 40 nodes was originally included. The simplification algorithm removed 80 % of the discretisation nodes resulting in doubling of the computational speed, without undue deterioration of the model results.

3.5 Model inclusion

Model Inclusion, in the COCOP context, has been defined as "Use model as is but with added interface to COCOP". This definition has two parts. Firstly, the model is used "as is" i.e. it needs not to be simplified or re-programmed; its accuracy and computational speed are sufficient. Secondly, the definition acknowledges that in order to be included into the COCOP system, an "added interface" may be needed. This interface will allow model execution to be started by the COCOP systems as well as allow for input and output data to be transferred. This interface thus connects the model to the COCOP architecture, drafted in D3.1 and later updated in D3.7.

How to include a model in the COCOP system is another aspect of connecting models together. The Functional Mockup Interface (FMI, http://fmi-standard.org/) is a promising tool-independent standard to support model exchange and co-simulation. It was first released in 2010 as a result from the ITEA2 project MODELISAR (2008-2011) and updated to FMI 2.0 in 2014. The basic concept is the Functional Mockup Unit (FMU), which is a file that contains an XML description of the interfaces and implementation of the model either as C code or as binary. The FMU can be included as an external part of a simulation tool, which reads the XML file in order to obtain information of the model. The model is then executed by the simulation tool's solver. This is referred to as Model Exchange. In addition, the standard also deals with Co-Simulation, where the FMUs contain also the solver. In this way, several simulation tools can be coupled and each tool deals with one part of the problem. The tools solve their own subsystem models independently and exchange data only at discrete communication points.

It should be noted, that considering the platform-independent vision of the COCOP architecture, FMI is a low-level technology. Thus, FMI is not a solution for system-to-system communication, but it is applicable in the scope of individual simulation modules.

3.5.1 Model inclusion of online LCA with FMI example

In COCOP we exemplify FMI Co-Simulation with a Life Cycle Assesment (LCA) model. This is also the first example of the online LCA referred to earlier.

> *The LCA model, created with an LCA software, the Simantics-based SULCA (https://www.simantics.org/ and https://www.simulationstore.com/sulca), is transformed into an FMU, which then can be connected to a running plant's control system. In this example, the FMU is called by Outotec's ACT control software (http://www.outotec.com/products/analyzers-and-automation/act-advanced-process-control/). Between the FMU and ACT, there are connector and wrapper codes that enable the use of the FMU over the OPC communication standard's different versions: Data Access (DA) and Unified Architecture (UA). The use of OPC makes the FMU-based LCA model usable directly from a plant's digital control system (DCS). The FMU itself consists of two parts. Firstly, the model which describes the physical process for which the LCA is calculated and secondly, the mathematical solver which solves the resulting equations. The architecture is depicted in the figure below.*

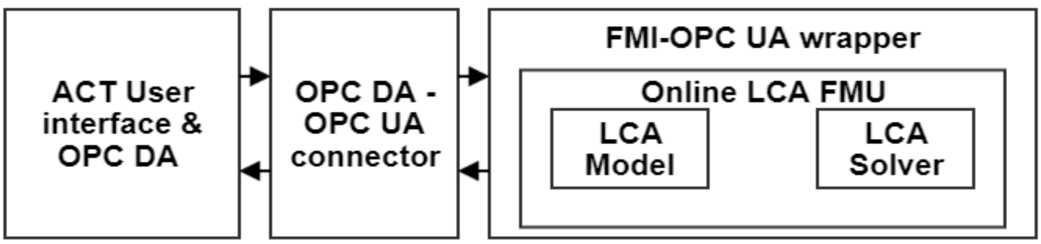

Figure 11 Data flow in FMI-based online LCA

> *In this example, the workflow to launch an online LCA calculator is a two step process. In the first step, the LCA model is constructed in the offline SULCA tool, parametrized and then exported. This step is illustrated in the screen capture below, see Figure 12.*

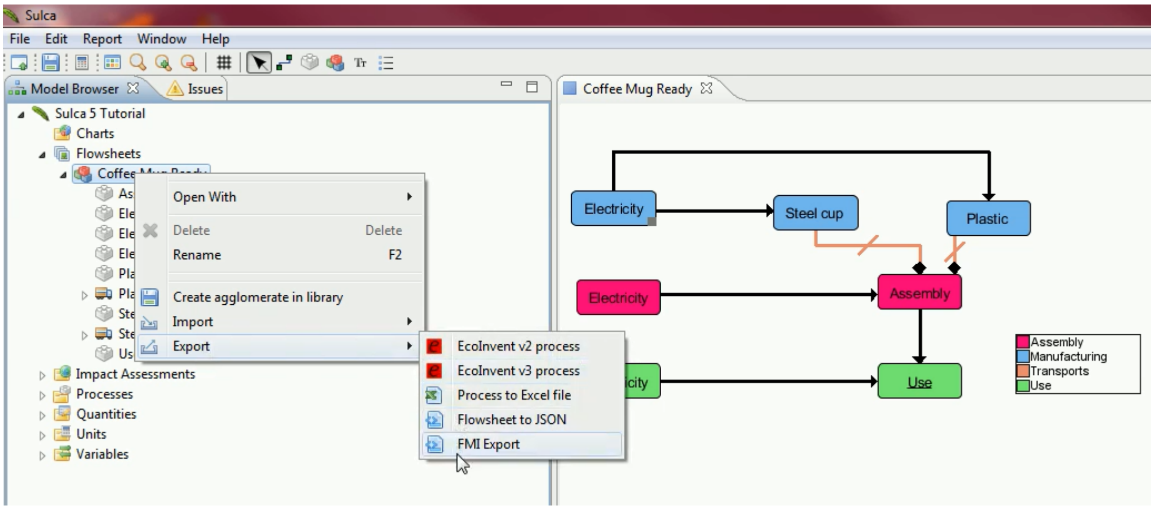

Figure 12 Exporting an LCA model as an FMU

The export functionality generates the FMU, which is, as described in Figure 11, a package with two parts (a zip file). In the second step, in ACT Designer software, the OPC connection is done and rest of the automation application is configured, see Figure 13.

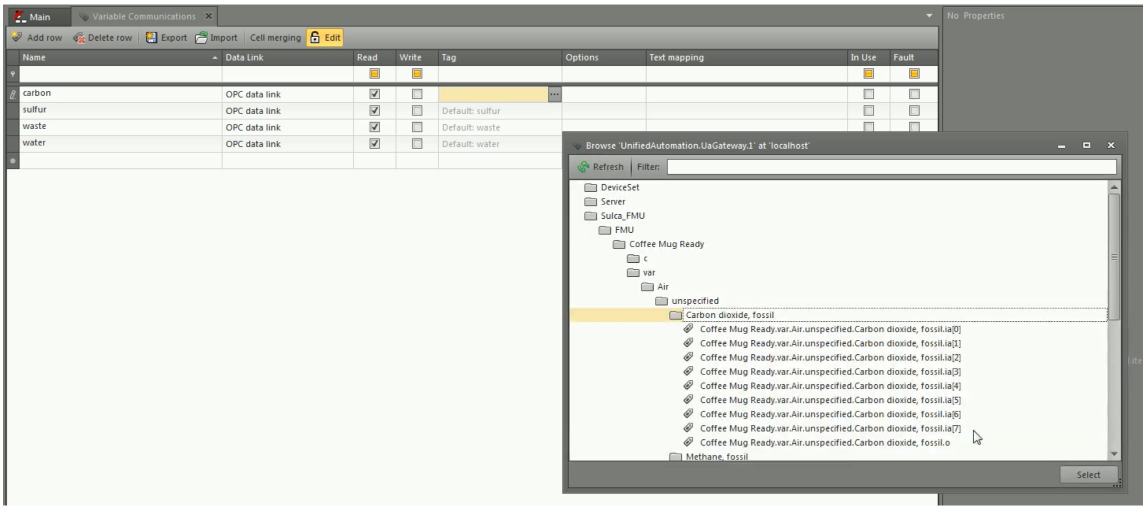

Figure 13 Configuring ACT's OPC communication with the FMU

Finally, the user interface is started, in this case in a web browser as shown in Figure 14, and the FMU calculation is executed.

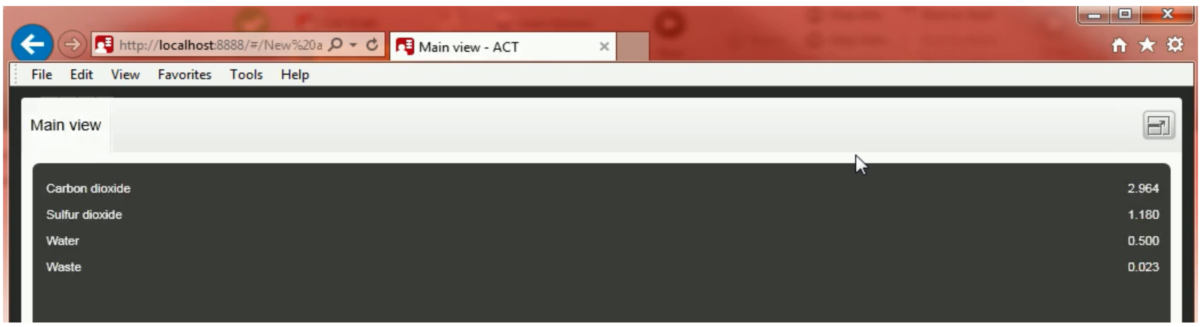

Figure 14 Very simple ACT UI for online LCA

3.6 Model upgrading

As stated above model upgrading is defined as "Extend with new functionality". This kind of action seems quite appropriate since Legacy Models are, by definition, developed before the decision to implement the COCOP approach to a given plant. This means that the model may have been developed for some other purpose than operator support or process control. As was stated in D4.1, Main Actions on Legacy Models are not mutually exclusive, and such a situation may arise quite easily with Model upgrading. As the name implies, in Model upgrading the model becomes more complex and care must be taken that it does not become too complex, especially with respect to simulation speed. This in turn may result in a situation where some other part of the model needs to be simplified. A generic recipe on how to do this is difficult, if not impossible to give, but one potential avenue could be sensitivity analysis, which was mentioned also above in conjunction with Model simplification. In sensitivity analysis, one problem setting discussed in the literature (see e.g. Saltelli et al., 2008) is parameter screening. The screening aims at identifying those parameters that have no, or only a minor effect, on the output of interest. In the present case, if the existence some part of an upgraded model can be described by one parameter, then a screening study may give indication whether that part of the model could now be removed.

In the COCOP project Model upgrading has been applied, for example, to the flash smelting furnace model of the copper pilot case. Those developments are summarised in D4.3, and thus are not repeated here.

3.7 Model transformation

In model transformation a Legacy Model is changed, but not simplified or extended. Such a change may be necessitated, for example, by computational speed requirements. A model transformation is applicable also when a model is taken from one modelling environment to another. As an example we summarise the work from (Björkqvist, 2016), where an iterative copper smelting balance model was converted into an explicit model by using symbolic manipulation.

> *In this work, a mass balance, done with Outotec's HSC-Sim software, for a copper flash smelting furnace was developed. The execution of the model required an iterative solution, which was deemed to be too slow for real-time optimisation purposes. To alleviate this the authors tried out symbolic manipulation of the model equations using Matlab's Symbolic Math Toolbox with the goal of obtaining explicit equations for the desired outputs. The transformation resulted in very long expressions, which are not feasible for human*

> *manipulations, but present no problems for a computer. The authors note that prior to the transformation, care must be taken in model formulation, and after the transformation, a comparison with the original model must be performed. Furthermore, as the expressions obtained are complex, editing them (if needed) directly is not feasible. This is, though, not a hindrance since editing the original model is easy and intuitive and the transformation can be done again.*

Another view of model transformation is automatic generation of executable program code from mathematical equations. In other words, a mathematical model is transformed into a computer simulation model. To this end, (Esche et al., 2017) published an interesting view on the relationship of simulation and optimisation. They have noticed that a modelling engineer commonly formulates models for simulation, and then re-implements them for optimisation purposes. Typically, the existing sophisticated simulation models are either used as a starting point for the reduced order models, or to generate training data for surrogate modelling. The major drawback in surrogate modelling lies in the narrow range the snapshot based surrogate models can handle in the optimisation, which leads to repeating new, time consuming surrogate modelling steps. They also point out that an optional development path among simulation providers is to extend the existing modelling and simulation tools towards optimisation. The research group at Technische Universität Berlin (Esche et al., 2017) looked at this simplification issue from a different angle. For several years already, they have been developing a free, web-based software platform MOSAIC (Kraus et al., 2014) (http://www.mosaic-modeling.de/) to provide a tool for combined simulation and optimisation needs. In the system, algebraic and differential equation systems are entered by the user in LaTeX and stored in MathML. Given the formulation in MathML, the codes of fully instantiated models can then be exported to various modelling platforms for simulation and/or optimisation purposes. The background and strongest application area for this development is chemical engineering. It is worth mentioning that also modelling and simulation platform providers are working hard to provide capabilities for optimisation within the same platform, where the modelling work is conducted.

4 Conclusions and further work

This document works as a guideline for any interested person, for example a plant owner or a development engineer, to assess whether the COCOP methodology fits in his/her industrial plant, and when found promising, helps with the practical actions in the system development, commissioning and maintenance. In addition, as simulation models are focal in this approach, this guideline helps in pursuing proper models for new COCOP solutions. In the modelling domain, we introduce and use concepts of Legacy Models and Main Actions to outline different models and modelling needs the target organisation and partners have, and to further process them towards the optimisation use in COCOP. This modelling effort can be on the plant responsibility or done by a separate solution provider. We also discuss the roles of different actors and give initial (conventional) proposals in the use case descriptions.

We address the question "What needs to be done when the COCOP concept is applied to a given plant?" from several different angles. We give tools to estimate the current state in the target plant and identify key factors for a successful COCOP commissioning. We promote the use case approach to establish distinct and practically oriented steps to lead the development. We emphasize use of KPIs to set a baseline and quantify the benefits. Moreover, the KPI definition helps to set and understand priorities, and thus it directs the development. Special novelty in the COCOP development process comes by the application of LCA, and human factors and social perspectives. COCOP reaches for improvements in the plant operation in many respects, thus being - or sometimes even demanding - a significant change of mindset of the personnel. The new way of thinking and the influences on the plant's sociotechnical system require many people to be convinced of the expected benefits. With this holistic approach, we want to guarantee that the COCOP solution is not only technically viable, but also accepted by the plant operators and other personnel, and very importantly, enhances environmentally sustainable production.

This guideline pertain to the situation at the time of writing of this deliverable. To be more precise, at the moment the project has addressed requirement specification, use cases, modelling and the underlying architecture in different level of details, but regarding implementation to pilot plants and verification and validation, the work has not yet been started. Consequently, these topics are covered in less detail here, while an updated version of this guideline: D4.6 "*Modelling guideline document and demonstration development kit (update)*" will cover the implementing aspects as well. In respect of Legacy Models and Main Actions, we continue searching and studying different software tools available and potentially applicable in the COCOP approach. For example, the MOSAIC tool will be investigated further. Also, the concept of online LCA will be pushed further in more realistic test cases.

References

Adiego, B. F., Vinuela, E. B., Tournier, J. C., Suárez, V. M. G., & Bliudze, S. (2013, July). Model-based automated testing of critical PLC programs. In *Industrial Informatics (INDIN), 2013 11th IEEE International Conference on* (pp. 722-727). IEEE.

Amaro, B., Immanuel, C.D., Pistikopoulos, E.N., Daiß, A., Hungenberg, K., Saraiva, P.A., 2010. Dynamic process optimisation in free-radical multicomponent polymerisation: Butyl methacrylate and butyl acrylate case study, Computer Aided Chemical Engineering. Elsevier B.V. doi:10.1016/S1570-7946(10)28097-5

Amicarelli, A., Quintero, O., Sciascio, F., 2014. Behavior comparison for biomass observers in batch processes. Asia-Pacific J. Chem. Eng. 9, 81–92. doi:10.1002/APJ.1748

Ansari, A.B., Esfahanian, V., Torabi, F., 2016. Discharge, rest and charge simulation of lead-acid batteries using an efficient reduced order model based on proper orthogonal decomposition. Appl. Energy 173, 152–167. doi:10.1016/j.apenergy.2016.04.008

Beck, J., Friedrich, D., Brandani, S., Fraga, E.S., 2015. Multi-Objective Optimisation using Surrogate Models for the Design of VPSA Systems. Comput. Chem. Eng. 82, 318–329. doi:10.1016/j.compchemeng.2015.07.009

Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Schwaber, K., Sutherland, J., Thomas, D., 2001. Manifesto for Agile Software Development, http://agilemanifesto.org/ [Accessed 19.1.2018]

Björkqvist, T., Suominen, O., Vilkko, M., Korpi, M., 2016. From Iterative Balance Models to Directly Calculating Explicit Models for Real-time Process Optimization and Scheduling, in: 9th EUROSIM Congress on Modelling and Simulation2. Oulu, Finland, pp. 184–188. doi:10.1109/EUROSIM.2016.9

Bouabaz, K., Zhu, Q., 2016. Improved numerical technique for industrial robots model reduction and identification. Proc. 2016 IEEE 11th Conf. Ind. Electron. Appl. ICIEA 2016 1032–1038. doi:10.1109/ICIEA.2016.7603734

Bremer, J., Goyal, P., Feng, L., Benner, P., Sundmacher, K., 2017. POD-DEIM for efficient reduction of a dynamic 2D catalytic reactor model. Comput. Chem. Eng. doi:10.1016/j.compchemeng.2017.02.032

Brown, T.M., Brouwer, J., Samuelsen, G.S., Holcomb, F.H., King, J., 2008. Accurate simplified dynamic model of a metal hydride tank. Int. J. Hydrogen Energy 33, 5596–5605. doi:10.1016/j.ijhydene.2008.05.104

Chen, T., Hadinoto, K., Yan, W., Ma, Y., 2011. Efficient meta-modelling of complex process simulations with time-space-dependent outputs. Comput. Chem. Eng. 35, 502–509. doi:10.1016/j.compchemeng.2010.05.013

Chu, Y., You, F., 2014. Integrated Planning, Scheduling, and Dynamic Optimization for Batch Processes : MINLP Model Formulation and Efficient Solution Methods via Surrogate Modeling. Ind. Eng. Chem. Res. 53, 13391–13411. doi:10.1021/ie501986d

Cline, A., 2015. Agile Development in the Real World, Apress, ISBN-13: 978-1484216781

de Pina, A.C., de Pina, A.A., Albrecht, C.H., Leite Pires de Lima, B.S., Jacob, B.P., 2013. ANN-based surrogate models for the analysis of mooring lines and risers. Appl. Ocean Res. 41, 76–86. doi:10.1016/j.apor.2013.03.003

de Pina, A.C., Monteiro, B. da F., Albrecht, C.H., de Lima, B.S.L.P., Jacob, B.P., 2014. ANN and wavelet network meta-models for the coupled analysis of floating production systems. Appl. Ocean Res. 48, 21–32. doi:10.1016/j.apor.2014.07.009

Demiray, T., Kertscher, P., Andrin, M., Arnold, T., Weber, H., Michael, H., 2011. Identification and Reduction of Hydro Power Plant Models based on On-Site Measurements.

Dikert K., Paasivaara M., Lassenius C., 2016. Challenges and success factors for large-scale agile transformations: A systematic literature review". Journal of Systems and Software 119, 87-108. Elsevier. doi: 10.1016/j.jss.2016.06.013

Disli, I., Kienle, A., 2012. Systematic evaluation of models of different complexity for a low-density polyethylene plant. Math. Comput. Model. Dyn. Syst. 18, 397–412. doi:10.1080/13873954.2011.642383

Dones, I., Preisig, H.A., 2010. Model simplification and time-scale assumptions applied to distillation modelling. Comput. Chem. Eng. 34, 732–743. doi:10.1016/j.compchemeng.2009.11.002

Elixmann, D., Puschke, J., Scheu, H., Schneider, R., Wolf, Inga, J.., Marquardt, W., 2014. A software environment for economic NMPC and dynamic real-time optimization of chemical processes. Automatisierungstechnik 62, 150–161. doi:10.1515/auto-2014-1020

Ersal, T., Fathy, H.K., Rideout, D.G., Louca, L.S., Stein, J.L., 2008. A Review of Proper Modeling Techniques. J. Dyn. Syst. Meas. Control. Trans. ASME 130, 610081–6100813. doi:10.1115/1.2977484

Esche, E., Hoffmann, C., Illner, M., Müller, D., Fillinger, S., Tolksdorf, G., Bonart, H., Wozny, G., Repke, J.-U., 2017. MOSAIC - Enabling Large-Scale Equation-Based Flow Sheet Optimization. Chemie Ing. Tech. 89, 620–635. doi:10.1002/cite.201600114

Fouladinejad, N., Fouladinejad, N., Abdul Jalil, M.K., Mohd Taib, J., 2016. Development of a surrogate-based vehicle dynamic model to reduce computational delays in a driving simulator. Simulation 92, 1087–1102. doi:10.1177/0037549716675956

Gong, W., Duan, Q.Y., Li, J.D., Wang, C., Di, Z.H., Ye, A.Z., Miao, C.Y., Dai, Y.J., 2016. An intercomparison of sampling methods for uncertainty quantification of environmental dynamic models. J. Environ. Informatics 28, 11–24. doi:10.3808/jei.201500310

Gong, X., Gu, Z., Ye, J., Yan, X., Zhao, Z., 2013. Surrogate Model for Aerodynamic and Handling Stability Optimization of a Tractor-Trailer in Crosswinds, in: Proceedings of the FISITA 2012 World Automotive Congress. Springer-Verlag Berlin Heidelberg, pp. 189–200. doi:10.1007/978-3-642-33741-3

Han, M., Wang, X., Wang, Y., 2008. Applying ICA on Neural Network to Simplify BOF Endpiont Predicting Model, in: 2008 International Joint Conference on Neural Networks. pp. 771–776.

Hangos, K., Cameron, I., 2001. Process modelling and model analysis, Academic Press, San Diego.

Hansen, A., Hedrick, J.K., 2015. Nonlinear Control Design within the High Level Modeling Framework for an Engine Cold Start Scenario *.

Hawe, G., Sykulski, J., 2007. Considerations of accuracy and uncertainty with kriging surrogate models in single-objective electromagnetic design optimisation. IET Sci. Meas. Technol. 1, 37–47.

IEEE, 2009. 830 Standards, 830-1998 - IEEE Recommended Practice for Software Requirements Specifications, reaffirmed in 2009, C/S2ESC. Software & Systems Engineering Standards Committee, sponsored by IEEE Computer Society. Piscataway, NJ, 1998.

ISO/IEC/IEEE. 2011. Systems and software engineering - Requirements engineering. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission/ Institute of Electrical and Electronics Engineers (IEEE), (IEC), ISO/IEC/IEEE 29148.

Hernandez, A.F., Grover, M.A., 2011. Comparison of Sampling Strategies for Gaussian Process Models, with Application to Nanoparticle Dynamics. Ind. Eng. Chem. Res. 50, 1379–1388. doi:10.1021/ie1007954

Kim, T., 2015. Surrogate model reduction for linear dynamic systems based on a frequency domain modal analysis. Comput. Mech. 56, 709–723. doi:10.1007/s00466-015-1196-4

Kodra, K., Zhong, N., Gaji, Z., 2016. Model Order Reduction of an Islanded Microgrid Using Singular Perturbations 3650–3655. doi:10.1109/ACC.2016.7525480

Koelsch, G., 2016. Requirements Writing for System Engineering, 1st edition, Apress Berkely, CA, USA, ISBN:1484220986 9781484220986

Kraus, R., Fillinger, S., Tolksdorf, G., Minh, D.H., Merchan-Restrepo, V.A., Wozny, G., 2014. Improving Model and Data Integration Using MOSAIC as Central Data Management Platform. Chemie Ing. Tech. 86, 1130–1136. doi:10.1002/CITE.201400007

Moghaddam, I.N., Salami, Z., Mohajeryami, S., 2014. Generator excitation systems sensitivity analysis and their model parameter's reduction. 2014 Clemson Univ. Power Syst. Conf. PSC 2014. doi:10.1109/PSC.2014.6808113

Mohamed, M., 2015. Model Simplification Methods for a Reduced Order System ofFlexible Aircraft, in: 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT). pp. 294–300.

Näveri, Jussi, and Pasi Laakso. "Instructor station for apros based Loviisa NPP training simulator." *System Simulation and Scientific Computing, 2008. ICSC 2008. Asia Simulation Conference-7th International Conference on*. IEEE, 2008.

Paluszczyszyn, D., Skworcow, P., Ulanicki, B., 2013. Online simplification of water distiribution network models. J. Hydroinformatics 15, 652–665. doi:10.2166/hydro.2013.029

Peltola, J., Sierla, S., Aarnio, P., & Koskinen, K. (2013, September). Industrial evaluation of functional Model-Based Testing for process control applications using CAEX. In *Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on* (pp. 1-8). IEEE.

Peng, H., Wang, W., 2016. Adaptive surrogate model-based fast path planning for spacecraft formation reconfiguration on libration point orbits. Aerosp. Sci. Technol. 54, 151–163. doi:10.1016/j.ast.2016.04.017

Pilát, M., Neruda, R., 2013. Surrogate Model Selection for Evolutionary Multiobjective Optimization. 2013 IEEE Congr. Evol. Comput. 1860–1867.

Rocha, H., 2009. On the selection of the most adequate radial basis function. Appl. Math. Model. 33, 1573–1583. doi:10.1016/j.apm.2008.02.008

Rogers, A., Ierapetritou, M., 2015. Feasibility and flexibility analysis of black-box processes part 1: Surrogate-based feasibility analysis. Chem. Eng. Sci. 137, 1005–1013. doi:10.1016/j.ces.2015.06.026

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S., 2008. Global Sensitivity Analysis: The Primer. John Wiley & Sons, Ltd.

Santillán Martínez, G., Miettinen, T., Aikala, A., Savolainen, J., Kondelin, K., Karhela, T., Vyatkin, V., 2016. Parameters selection in predictive online simulation, in: IEEE 14th International Conference on Industrial Informatics. Poitiers, France.

Savolainen, J., 2013. Global sensitivity analysis of a feedback-controlled stochastic process model. Simul. Model. Pract. Theory 36, 1–10.

Savolainen, R., Sierla, S., Karhela, T., Miettinen, T., & Vyatkin, V. (2017, July). A framework for runtime verification of industrial process control systems. In *Industrial Informatics (INDIN), 2017 IEEE 15th International Conference on* (pp. 687-694). IEEE.

Sen, O., Gaul, N.J., Choi, K.K., Jacobs, G., Udaykumar, H.S., 2017. Evaluation of Kriging Based Surrogate Models Constructed from Mesoscale Computations of Shock Interaction with Particles. J. Comput. Phys. 336, 235–260. doi:10.1016/j.jcp.2017.01.046

Shi, H., You, F., 2015. Adaptive Surrogate-based Algorithm for Integrated Scheduling and Dynamic Optimization of Sequential Batch Processes. Conf. Decis. Control 7304–7309. doi:DOI: 10.1002/aic.14974

Shokry, A., Audino, F., Vicente, P., Escudero, G., Moya, M.P., Graells, M., Espuña, A., 2015. Modeling and Simulation of Complex Nonlinear Dynamic Processes Using Data Based Models:

Application to Photo-Fenton Process, 12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering. Elsevier. doi:10.1016/B978-0-444-63578-5.50027-X

Siivola, E., Sierla, S., Niemistö, H., Karhela, T., & Vyatkin, V. (2016, July). Requirement verification in simulation-based automation testing. In *Industrial Informatics (INDIN), 2016 IEEE 14th International Conference on* (pp. 740-743). IEEE.

Sommerwerk, K., Michels, B., Lindhorst, K., Haupt, M.C., Horst, P., 2016. Application of efficient surrogate modeling to aeroelastic analyses of an aircraft wing. Aerosp. Sci. Technol. 55, 314–323. doi:10.1016/j.ast.2016.06.011

Song, W., Han, K., Wang, Y., Friesz, T., del Castillo, E., 2017. Statistical metamodeling of dynamic network loading. Transp. Res. Procedia 23, 263–282. doi:10.1016/j.trpro.2017.05.016

Soumitri, M.S., Majumdar, S., Mitra, K., 2015. Optimization using ANN surrogates with optimal topology and sample size. IFAC-PapersOnLine 28, 1168–1173. doi:10.1016/j.ifacol.2015.09.126

Szimandl, B., Németh, H., 2015. Systematic Model Simplification Procedure Applied to an Electro-Pneumatic Clutch Model. Period. Polytech. Transp. Eng. 43, 35–47. doi:10.3311/PPtr.7467

Vincenzi, L., Gambarelli, P., 2017. A proper infill sampling strategy for improving the speed performance of a Surrogate-Assisted Evolutionary Algorithm. Comput. Struct. 178, 58–70. doi:10.1016/j.compstruc.2016.10.004

Viswanath, A., Forrester, A.I.J., Keane, A.J., 2014. Constrained Design Optimization Using Generative Topographic Mapping. AIAA J. 52, 1010–1023. doi:10.2514/1.J052414

Wang, Z., McBee, B., Iliescu, T., 2015. Approximate partitioned method of snapshots for POD. J. Comput. Appl. Math. 307, 374–385. doi:10.1016/j.cam.2015.11.023

Wilson, B.H., Stein, J.L., 1992. Algorithm for obtaining minimum-order models of distributed and discrete systems, in: Proceedings of Winter Annual Meeting of the American Society of Mechanical Engineering. ASME, Anaheim, California, USA, pp. 47–58.

Wu, S., Mclean, K.A.P., Harris, T.J., Mcauley, K.B., 2011. Selection of optimal parameter set using estimability analysis and MSE-based model-selection criterion. Int. J. Adv. Mechatron. Syst. Mechatron. Syst. 3, 188–197.

Xia, B., Lee, T.W., Choi, K., Koh, C.S., 2016. A Novel Adaptive Dynamic Taylor Kriging and Its Application to Optimal Design of Electromagnetic Devices. IEEE Trans. Magn. 52, 2–5. doi:10.1109/TMAG.2015.2487380

Xia, B., Ren, Z., Koh, C.S., 2014. Utilizing kriging surrogate models for multi-objective robust optimization of electromagnetic devices. IEEE Trans. Magn. 50, 1–4. doi:10.1109/TMAG.2013.2284925

Xia, B., Ren, Z., Koh, C.S., 2015. Comparative study on Kriging surrogate models for metaheuristic optimization of multidimensional electromagnetic problems. IEEE Trans. Magn. 51. doi:10.1109/TMAG.2014.2361956

Yin, H., Wen, G., Gan, N., 2011. Crashworthiness Design for Honeycomb Structures Under Axial Dynamic Loading. Int. J. Comput. Methods 8, 863–877. doi:10.1142/S0219876211002885

Yousefi, A., 2006. A Compromise Between Simplicity \& Accuracy of Nonlinear Reduced Order Systems. Rt.Mw.Tum.De 6203, 1–25. doi:10.2316/Journal.205.2009.4.205-4808

Zheng, Z., Chen, X., Liu, C., Huang, K., 2015. Using support vector machine and dynamic parameter encoding to enhance global optimization. Eng. Optim. 273. doi:10.1080/0305215X.2015.1057056

Appendix A: Example of a use case

One use case for the copper pilot's FSF is given below. Like all the use cases, this has been created and updated in the project's Confluence system, and it links to several requirements in the JIRA system.

> ***UC-C-FSF-010: Flash Smelt Furnace***
>
> ***Purpose***
>
> *To produce matte to be further processed in PS-Converters with suitable matte grade (copper content in matte).*
> *To produce slag with optimal SiO_2 content and temperature.*
> *To maximize feed to the furnace. To tap FSF matte at suitable times. To tap FSF slag at suitable times.*
>
> ***Actors***
>
> *Operators, engineers*
>
> ***Preconditions***
>
> *There is feed mixture and utilities available to produce matte.* *TUTCOCOPDEV-55 - REQ-C-050: Temporal resource limitations*
> *The feed mixture composition is set.*
>
> ***Body***
>
> *The process is continuous rather than a batch process so there is no actual sequence of steps to be executed.*
>
> - *Process parameters are set as follows.*
>   - *The engineers set the target feed rate according to the capacity of the smelter. (Exceptions: CLEAN, MAIN)* *TUTCOCOPDEV-56 - REQ-C-FSF-010: FSF feed rate indications*
>   - *The engineers set the target matte grade so it is suitable for the current/next PSC slag blow. The operators control matte grade with oxygen coefficient. (Exceptions: GRADE)* *TUTCOCOPDEV-57 - REQ-C-FSF-020: Matte grade evaluation*
> - *The following properties are controlled.*
>   - *The operators control the slag silica content with flux ratio. Silica is added to wet concentrate before drying. (Expection: SILICA)*
>   - *The operators control slag temperature according to temptip measurements and feedback from the field operators about slag tapping velocity.* *TUTCOCOPDEV-58 - REQ-C-FSF-030: FSF Fuel and O2 indications*

        - *The operators control O2 content in input air (oxygen enrichment).*
        - *The operators utilise fuel burners to produce additional heat in the reaction shaft and in the settler part of FSF. The operators try to mimimize fuel usage in the long run. (Exception: FUEL, TEMP)*
        - *Other temperature control possibilities in some smelters*
            - *Oxide revert amount in feed (coolant)*
            - *Change dust feed (dust level in bin changes; dust is a coolant)*
    - *The operators control the off-gas amount within the limits of the off-gas line. The oxide revert feed and dust feed affect required oxygen enrichment and oxygen coefficient. TUTCOCOPDEV-59 - REQ-C-FSF-040: Off-gas indications*
    - *The operators control the dust feed according to the limits in the dust bin. Dust feed has strong effect to matte grade and slag temperature.*
    - *The operators control the dust sulfation oxygen and air. Feed back from acid plant weak acid production rate and visual inspection of heat recovery boiler and possible dust analysis. TUTCOCOPDEV-60 - REQ-C-FSF-050: Oxygen flow control indications*
    - *The engineers set the target for big iron amount/shift to control the bottom buildup.*
- *Tapping is performed as follows.*
    - *The operators tap matte when there is enough matte in the settler and there is a PS converter to receive matte. TUTCOCOPDEV-61 - REQ-C-FSF-060: Matte tapping suggestions*
    - *The operators tap slag when slag level is high enough and matte level is not too high. (Exception: SCF) TUTCOCOPDEV-62 - REQ-C-FSF-070: Slag tapping suggestions*

Exceptions

CLEAN: The concentrate burner or other equipment may require cleaning from buildups (temporarily feed off). TUTCOCOPDEV-55 - REQ-C-050: Temporal resource limitations
MAIN: There can be maintenance issues of equipment requiring temporarily feed off or lowered feed rate. TUTCOCOPDEV-55 - REQ-C-050: Temporal resource limitations
GRADE: The matte grade target may have to be lowered temporarily to decrease the SO2 load from the FSF. The matte grade target may have to be increased temporarily to produce heat.
SILICA: There can also be a separate silica feeder to the furnace; then, silica is added using the feeder instead of adding it to wet concentrate.
FUEL: The fuel is usually natural gas or oil.
TEMP: There are also other ways to control temperature like revert feed rate.

SCF: If slag is cleaned in SCF, also the SCF has to be ready. If using slow cooling, the slow cooling ladle has to be ready. It might happen that slag transportation capacity is limited.

Postconditions

FSF provides matte to PSC slag blows with optimal timing and matte grade. FSF produces slag with optimal SiO2 content and temperature.

Other remarks

Many variables affect matte grade, slag temperature, slag composition and off-gas amount. There can be different targets for variables regarding the situation.
The roles of engineers and operators may vary regarding the smelter.

Appendix B: Example of a requirement

An example of a technical requirement in the JIRA system is shown below.

[TUTCOCOPDEV-149] REQ-S-CC-MON-016: Prediction of break-outs Created: 21/Sep/17 Updated: 21/Nov/17	
Status:	To Do
Project:	COCOP Development
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Story	**Priority:**	Major
Reporter:	Carlos Leyva Guerrero	**Assignee:**	Unassigned
Resolution:	Unresolved	**Votes:**	0
Labels:	req-functional, req-steel-case		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original Estimate:	Not Specified		

Description	

The system must predict if the medium thickness evolution at the output of the mould is not enough to avoid a break-out

Generated at Thu Jan 04 09:20:42 EET 2018 by Jouni Savolainen using JIRA 7.4.0#74002-sha1:4bbb6c3997c184a0a30bb2830d61b4dc50b1a591.

Appendix C: Sub-classes of the unrevealed proper modelling techniques

As stated earlier, the frequency-based and projection-based proper modelling techniques have several sub-classes each. These are broken down in the subsequent tables, which include also short descriptions and references.

Frequency-based method sub-class	Short description	Method and application references
Aggregation	Number of system state variables is reduced, e.g. small time constants are ignored. Seems to be for linear systems only. Not realization-preserving.	Method [17–31]
Singular perturbation	Partitions the original model into two sub-models: driving and driven, for slow and fast dynamics, respectively. The fast sub-model uses the slow states as input variables. Particularly for numerically stiff systems. Realization-preserving.	Method [32–40] (Kodra et al., 2016) (Dones and Preisig, 2010)
Model order deduction algorithm (MODA)	The method starts with simple models and increments their complexity until the model captures the most relevant characteristic speeds of a given system for a given application. Realization-preserving.	Method [42–46]
Modal analysis	Modal analysis converts the model into modal representation, which allows elimination of the faster eigenvalues. In its simplest rendition, modal analysis focuses on linear, time-invariant, vector-second-order dynamic systems satisfying the principle of separation of variables. Not realization-preserving. Belongs to the projection-based techniques as well.	Method [49–50, 51–53]
Component mode synthesis (CMS)	An extension of modal analysis that is particularly applicable to large modular systems. It proceeds in two simple steps: i) it uses modal analysis to separately obtain a proper model of each module in the system, ii) it assembles these proper models into a system-level	Method [54–59] Application [60–62]

Frequency-based method sub-class	Short description	Method and application references
	proper model. This approach can be significantly less computationally expensive than direct CMS to the entire system model. Not realization-preserving.	
Polynomial approximation methods	Given a complex transfer function model, a lower-order approximation of the model is searched by constructing series expansion and retaining its first coefficients and truncating the rest. Limited to linear systems. Not realization-preserving. E.g. Padé approximation, Routh approximation.	Method [63–85] (Mohamed, 2015)
Oblique projection	This method simultaneously matches high and low frequency moments of the transfer function, and high and low power moments of the power spectral density. Not realization-preserving. A projection based method as well.	Method [86]
Optimal Hankel norm approximation	For a given, stable, linear, and time-invariant system G, Hankel norm approximation seeks an optimal reduced model G_r, whose order k is specified a priori by the modeller. The resulting optimal proper model minimizes the Hankel norm of the error $G-G_r$ over the set of all linear and time-invariant models of the desired order. Not realization-preserving.	Method [87–91, 92–95]

Table 6 Sub-classes of frequency-based proper modelling techniques (descriptions mostly origin from (Ersal et al., 2008). Note, the references in brackets refer to their references.

Projection-based method sub-class	Short description	Method and application references
Karhunen-Loève expansion	The method uses snapshots, i.e. observation data from a physical system or its model, to find a subspace that captures the dominant features. Specifically, using singular value decomposition, it finds the orthogonal basis	Method [96–98, 105, 107]

Projection-based method sub-class	Short description	Method and application references
	that optimally captures the energy (the dominant system dynamics) of the observation signals, in the least-squares sense. Projecting the system's model onto this subspace using the Galerkin projection method then furnishes the reduced model. Also known as: • Principal Component Analysis (PCA) • Method of empirical orthogonal functions • Proper Orthogonal Decomposition (POD) • Singular Value Decomposition (SVD) • Empirical eigenfunction decomposition • Method of quasiharmonic modes Popular in many fields including fluid dynamics, structural vibrations, image processing, and signal analysis. Applicable to nonlinear systems as well. Not realization-preserving.	Application [99–104, 106] (Ansari et al., 2016) (Bremer et al., 2017) (Sommerwerk et al., 2016) (Kim, 2015) (Wang et al., 2015)
Balanced truncation	Balanced truncation applies the Karhunen-Loéve expansion to find a balanced realization for the system. A system's realization is balanced, if its observability and controllability Grammians are equal, meaning that each state is as observable as it is controllable. When this is done, the less observable and less controllable states can be eliminated from the system's model to generate a reduced model. Significant research has also pursued on this method for nonlinear systems. Not realization-preserving.	Method [108–128] (Dones and Preisig, 2010) (Han et al., 2008) (Mohamed, 2015)
Component cost analysis	A specific cost function is defined for a linear stable system, and the reduced model is then obtained by truncating the low-cost states based on the rationale that the system cost should be perturbed minimally.	Method [129–133]

Table 7 Sub-classes of projection-based proper modelling techniques (descriptions mostly origin from (Ersal et al., 2008). Note, the references in brackets refer to their references.

Appendix D: Application examples from the literature

In the table below are listed several model simplification examples from the literature. The examples cover only approximate ten years, in order to save space. We have used subjective judging and listed the examples in the order of COCOP relevance.

Reference	Method	Application and results
(Chu and You, 2014)	Piecewise linear response surface model Complex large-scale MINLP problem. The integrated problem decomposed by using surrogate models to represent the linking functions among the sub-problems	Integrated planning, scheduling, and dynamic optimisation for sequential batch process. Better optimal (instead of sub-optimal) solutions found, and in a very large case found a solution (in 25 min) while the full model approach failed (in 50 h).
(Shi and You, 2015)	Piecewise linear response surface surrogate Complex large-scale MINLP problem. Bi-level optimization: the upper level is about the schedule, the lower level (using adaptive surrogate) about recipes and processing time.	Integrated planning, scheduling, and dynamic optimisation for sequential batch process. To maximize the total production profit over the scheduling horizon. Clearly larger profit with the new method compared to the conventional method reported, but also more computational time was needed!
(Amicarelli et al., 2014)	Estimators by 5 methods compared: a phenomenological estimator based on dissolved oxygen balance, an extended Kalman filter, a Gaussian process regression-based, an ANN-based, and finally, an estimator based on information fusion by a decentralized Kalman filter.	Biomass concentration estimation in a batch bioprocess, for control purposes. If a reliable and accurate model of the bioprocess is available, then simple phenomenological estimators are the first option. Beneficial properties (robustness) against model degradation can be gained by information fusion (two or more methods combined).
(Bremer et al., 2017)	Proper Orthogonal Decomposition, discrete empirical interpolation method (DEIM) The order of the original nonlinear reactor model was 4375, the	Two-dimensional model of catalytic, tubular reactor for CO_2 methanation in two dynamic scenarios: disturbed continuous operation and start-up.

Reference	Method	Application and results
	reduced models were of order 34 and 36. Each scenario had its own model realization.	The biggest contribution to the acceleration of the simulation (>10x) was found to be due to the DEIM approach, not due to the reduced order.
(Soumitri et al., 2015)	ANN (Sobol based) surrogate	Finding optimum processing conditions for operating the PVAc batch polymerisation reactor, leading to corresponding results 10 times faster.
(Disli and Kienle, 2012)	Physics based insights (e.g. method of moments, simplifications of flow diagram and reaction mechanisms, the energy balance, constant material properties and overall heat transfer coefficients). 4 levels of simplified models with model order reductions from the original 30000 to 5800/3400/2400/2400.	Low-density polyethylene (LDPE) production plant, featuring a long (over 1 km) reactor tube, and highly nonlinear characteristics due to the exothermic reactions and material recycles. The reactor profiles more or less aligned with the detailed model. Good dynamic prediction even with the simplest model, but incapable to perform in exceptional conditions.
(Han et al., 2008)	Independent component analysis (ICA) to reduce the model dimensions, ANN for modelling.	Prediction of carbon content and temperature in oxygen blowing endpoint for Basic Oxygen Furnace (steel making). The reference model not of very high dimension, so calculation speed-up was not a concern. They claimed that the simplified model gave even better results than original.
(Amaro et al., 2010)	Simplification in modelling phase (e.g. pseudo-homopolymerisation approximation) Modelling and optimisation at gPROMS platform	Free-radical copolymerisation. Optimal reactor conditions searched to attain target molecular weight distribution (MWD) and copolymer composition, by manipulating process variables such as monomer and initiator feed rate profiles. The desired MWD reached, but the process time not shortened.

Reference	Method	Application and results
(Beck et al., 2015)	Kriging surrogate GA used in multi-objective optimisation: the purity and the recovery maximised simultaneously in a Pareto sense.	Vacuum/pressure swing adsorption (VPSA) type separation (CO2/N2 from flue gas) process, operation cycle configuration. Computational reduced effort by factor 2–5.
(Shokry et al., 2015)	Ordinary Kriging (OK), ANN and Support Vector Regression (SVR) used for surrogate modelling (comparison)	A photo-Fenton batch pilot plant, monitoring purpose All three methods can be used as soft sensors. OK preferred due to its accuracy, flexibility, robustness and confidence interval information.
(Hernandez and Grover, 2011)	Kriging with different sampling strategies	Dynamic simulation of nanoparticle synthesis
(Dones and Preisig, 2010)	Order-of-magnitude assumptions, singular perturbation (SB) and lumping Simplifications in each tray called horizontal folding assuming fast heat and mass transfer, while vertical folding lumps the capacity effects and retains a network of stationary transfers.	Flash and distillation column models. No simulation results shown.
(Ansari et al. 2016	Proper Orthogonal Decomposition (POD)	Electrochemical model for a lead-acid cell during a cycle of discharge, rest and charge processes, for monitoring purposes POD-based model was computationally 15x faster than the classic CFD approach with the same accuracy.
(Demiray et al., 2011)	1) Utilising physical insights of the modelled hydro power plant 2) Standardisation of the sub-models maintaining the main features 3) Trajectory sensitivities to	Swiss electric power transmission system models, for coordination of operations Comparison of different simplifications. The reduced models were capable to

Reference	Method	Application and results
	perform the simplification and the following parameter optimisation	reflect the main characteristics of the measured hydro power plant in different operational modes, in particular islanded mode and interconnected mode.
(Paluszczyszyn et al., 2013)	Online model reduction method which emphasizes the preservation of the original model energy distribution. The operator can launch the model reduction to automatically produce an updated simplified model according to the current situation.	Water distribution network models for the purpose of online optimisation for energy and leakage management. The online model reduction helps to manage abnormal situations and structural changes to the water network, e.g. isolation of part of the network due to a pipe burst.
(Szimandl and Németh, 2015)	Approach with systematic features, based on engineering judgment and physical insights. Directed Graph used for illustrating model variables. Performance and size/complexity index used for evaluating the goals.	Dynamic hybrid model of an electro-pneumatic clutch system, for control design.
(Chen et al., 2011)	Kriging surrogate PCA-based approach with GP regression model, to represent tempo-spatial coordinates, which are highly correlated, by low-dimensional vector.	Scenario of the malicious release of hazardous materials Approach effective and efficient for meta-modelling of the time–space-dependent output variables.
(Gong et al., 2016)	Multivariate Adaptive Regression Spline (MARS), Kriging 8 sampling methods compared with Monte Carlo sampling, measuring methods' efficiency and effectiveness.	Uncertainty quantification of environmental dynamic models. According to the uniform metrics used, Symmetric Latin Hypercube (SLH) and Good Lattice Points (GLP) were the most efficient sampling methods.
(Sen et al., 2017)	Surrogate modelling with Dynamic Kriging Method (DKG) and Modified Bayesian Kriging Method (MBKG)	Particulate flows. In the absence of noise in the training data, the DKG method converges faster

Reference	Method	Application and results
		than MBKG. From numerical experiments in a multiscale modelling framework, MBKG is recommended.
(Rogers and Ierapetritou, 2015)	Kriging surrogate	Static and dynamic test problems and a case model of roller compaction (pharmaceutical manufacturing). Surrogate-based feasibility analysis demonstrated.
(Fouladinejad et al., 2016)	Surrogate modelling (Polynomial response surface, ANN, Kriging), sensitivity analysis, decomposition, sampling techniques. Decomposed the expensive model into smaller sub-models relevant for surrogate modelling (8 surrogates)	The Universiti Teknologi Malaysia Driving Simulator; real time responses crucial. Kriging approximation produced best accuracy when the responses were nonlinear. 5x faster than the expensive model, but the accuracy was acceptable only for 4 of the 6 outputs.
(Yousefi, 2006)	Approach that delivers models of reduced order and simple inner structure. The model structures were coded in binary strings and optimized using GAs and special fitness functions.	Hydropneumatic system in a car
(Bouabaz and Zhu, 2016)	A numerical procedure which categorizes the model parameters and analyses each for achieving a good balance between the model simplicity and accuracy.	Dynamic model of manipulating robot for torque prediction. The accuracy of the reduced model was considered very satisfactory.
(Mohamed, 2015)	5 methods compared: • Balanced Realization (BR) • Truncation • Residualization • Moment Matching • Pade's approximation	Model of pitching motion of a flexible aircraft, for control law design. BR, Truncation and Residualization considered best; they also remain the corresponding portion of the original model in the reduced model. BR is preferred when accuracy is required at high frequencies, Residualization instead for low frequencies.

Reference	Method	Application and results
(Hansen and Hedrick, 2015)	The simplification method (maybe a transformation method rather) is said to be exact, not-introducing any approximations.	Cold start dynamics of a four cylinder 2.4 L Toyota engine, for control design. The model obtained was successfully used for the controller design.
(de Pina et al., 2013, 2014)	ANN surrogate	Floating (marine) production systems.
(Xia et al., 2014, 2015, 2016)	Adaptive Dynamic Taylor Kriging (ADTK) surrogate Particle swarm optimisation (PSO)	Design of electromagnetic devices. ADTK selects the basis function by minimizing the fitting error. Combination of ADTK and PSO considered numerically efficient global optimisation approach.
(Hawe and Sykulski, 2007)	Kriging surrogate Two experimental designs are investigated: Latin hypercube, Hammersley sequence	Test functions on electromagnetic devices, for optimisation purpose. When the overall aim is to locate the optimum in as few iterations as possible, it is advantageous to start the iterative search early with a relatively inaccurate model.
(Peng and Wang, 2016)	Kriging, RBF (exponential function as the basis function) The nonlinear and multi-layer optimal control problem was decomposed using an adaptive surrogate model framework.	Optimal transfer paths of spacecraft formation (reconfiguration of 5–6 spacecraft) Most of the final relative errors were <5 %, the computational time of the RBF and Kriging surrogate models app. 5 % of the reference time.
(Gong et al., 2013)	Optimal Latin hypercube DOE method to analyse the sensitivity of the design variables. Kriging method for surrogate modelling. GA used in optimisation.	Wind deflector shape design for a tractor-trailer. The differences between the full model and the surrogate model results <1%. Time to perform the design reduced to below a week from one month with traditional methods.
(Viswanath et al., 2014)	Generative Topographic Mapping (GTM) uses initial DOE of the expensive model and provides a	Transonic aircraft wing design, transonic aircraft compressor rotor

Reference	Method	Application and results
	transformation into a low-dimensional manifold. GA employed in optimisation.	blade design (the last case involved an expensive solver) GTM reduced the model, found the optimum, satisfied constraints effectively. Suits well for the industry, where a near-optimal design that meets the constraints is more important than finding the exact optimum.
(Yin et al., 2011)	Optimal Latin Hybercube design (OLHD) for sampling. Polynomial functions, RBF, Kriging, Multivariate adaptive regression splines (MARS), and Support Vector Regression (SVR) for surrogates. Multi-objective particle swarm optimisation algorithm (MOPSOA) for optimisation.	Specific energy absorption and peak crushing stress of honeycomb structures under axial dynamic loading Quadratic and cubic polynomial functions were most accurate. The goal of design optimisation of the honeycomb structures with various cell specifications was reached.
(Zheng et al., 2015)	Support Vector Machine (SVM) for surrogate modelling Dynamic Parameter Encoding (DPE) for accelerating the convergence of the optimisation by compressing the solution space.	Design optimisation of an antenna. To meet the same optimisation objective > 60% saving in the computation time.

Table 8 Selected examples of model simplification and surrogate modelling studies.