COGNITWIN

Cognitive plants through proactive self-learning hybrid digital twins

DT-SPIRE-06-2019 (870130)

Deliverable Report

Deliverable ID	D5.1	**Version**	V2
Deliverable name	Baseline Hybrid AI and Cognitive Twin Toolbox		
Lead beneficiary	SINTEF (SINTEF AS)		
Editor	Stein Tore Johansen (SINTEF)		
Contributors	Enso Ikonen(UOULU), Istvan Selek (UOULU), Signe Mo (SINTEF), Nenad Stojanovic (NST), Tim Dahmen (DFKI), Pierre Gutierrez (Scortex), Sailesh Abburu (SINTEF), Michael Jacoby (Fraunhofer), Ljiljana Stojanovic (Fraunhofer), Jan Gunner Dyrset (CYB), Alexander Morin (SINTEF), Bjørn Tore Løvfall (SINTEF), Aslak Einbu (SINTEF), Stein Tore Johansen (SINTEF), Özlem Albayrak (TEKNO), Perin Ünal (TEKNO), Sudi Jawahery (CYB)		
Report review	Enso Ikonen (UOULU), Arne J. Berre (SINTEF)		
Due date	29.02.2020		
Date of final version	01.10.2021		
Dissemination level	PU		
Document approval	Frode Brakstad	01.10.2021	

The COGNITWIN project has received funding from the European Union's Horizon 2020 research and innovation program under GA No. 870130

Executive Summary

This D5.1 report is a documentation of the baseline technologies and methods that are the starting points for the "COGNITWIN Hybrid and Cognitive Twin Toolbox" and which will be applied in the developments for the COGNITWIN industrial pilots.
The information provided in the report will further be useful for aligning the concepts and available tools among the COGNITWIN partners but should also give external readers ideas about new Industry 4.0 possibilities.

The COGNITWIN projects aims toward supporting the digitalization of the European heavy industries. A main ambition of the COGNITWIN project is to develop cognitive digital twins that can support a significant improvement in industrial operation. To do so COGNITWIN will work with combining data, physics-based models, machine learning (ML) and Artificial Intelligence (AI) in the best possible manner to solve the industrial challenges. Cognition is introduced into the models through self-learning and AI. In COGNITWIN WP5, where this report belongs, the aim is to identify which ML/AI methods are suited for such problems and extend and/or develop new algorithms to further improve performances of the control systems. By developing a Cognitive Twin Toolbox, comprising methods to analyse data, exploit the information from physics-based models, combine information from data and numerical models, and demonstrating applications to process control, this can be applied more generally to support many different process industries. A Cognitive Twin Toolbox will be built out from the needs of 6 different industrial pilots, all with their specific and different challenges.
In this report we discuss the baseline technologies and methods that are the starting points for the Hybrid AI/Analytics and Cognitive Toolbox, and which will be applied in the developments for the pilots. The report gives a brief orientation about the 6 industrial pilot cases. Various toolbox elements are next presented, such as "Hybrid AI/Analytics and Cognitive Toolbox – overview and architecture",
"Plant Digital Twins with ML/AI", "Multi-variate Sensor analytics with Deep Learning", "Deep Learning Performance", "Hybrid Digital Twins" and finally "Cognitive Digital Twins". Relevance to the pilots is discussed, and an overview of the use or potential use of the toolbox elements in the various pilots is given. Details about the technologies which are available from the partners are presented in annexes.

The term initial is used throughout this report. This is used to express the toolbox elements developed by the partners before M6 and which has the potential to contribute to solve the pilot challenges. Initial does not allude to a follow-up report on D5.1. Further updates and actions will be integrated into the next further deliverables.
Status on the developments of the presented tools and methods is not presented herein but is given in the regular status reports.
Separate reports on the industrial pilots (D1.1, D2.1, D3.1), the "Baseline Platform, Sensor and Data Interoperability Toolbox" (D4.1) and the Key Performance Indicators (D6.1) and Data Management Plan (D8.1) are issued together with this report and will give a more complete picture about the COGNITWIN challenges and platform.

Table of Contents

Table of figures

Table of tables

Partner Technologies Acronyms

DSS	Decision Support System
MIS	Management Information Systems
ICT	Information and communications technology
IOT	Internet of Things
KPI	key performance indicators
Mgm	Management
DCS	Distributed Control System
HSE	Health, Safety and Environment
JSON	JavaScript Object Notation
ODE	Ordinary Differential Equation

Other acronyms are found at https://www.allacronyms.com/data/abbreviations.

1 Introduction to COGNITWIN Baseline Hybrid AI and Cognitive Twin Toolbox

The COGNITWIN projects aims toward supporting the digitalization of European heavy industries. These industries have specific challenges due to very complex processes and in many cases lack of relevant sensor data. By assembling a team of multiple skills within sensor technologies, physics-based modelling, data driven modelling, hybrid modelling and process control COGNITWIN aims to develop cognitive digital twins that can support a significant improvement in industrial operation.

This report deals with the baseline toolbox that is currently available to support " COGNITWIN for Industry Process Excellence", "Cognitive Digital Twins for Cognitive retrofitting" and "*Hybrid Twins for Optimized process performance".* The baseline technologies presented below and new developments during COGNITWIN will end up as an "Interoperability Toolbox" which can be offered "as a Service".

In COGNITWIN we have defined 6 Pilot cases where the objective is to apply the available digital technologies to improve well defined KPIs (Key Performance Index). We further plan to develop toolboxes which may be used in a number of other process industries. In this report we address the "Hybrid AI and cognitive twin toolbox" and the baseline will be related both to general industrial needs and the specific need for our pilot cases. The pilots will be introduced next.

2 State of the practice – COGNITWIN pilots

The COGNITWIN project is built around 6 pilots. Major elements will be to introduce robust, accurate and cost-efficient sensors using retrofitting as well as novel new sensors as needed to achieve the planned cognitive elements in form of proactive self-learning digital twins.

Although a full digitalisation of the plant is the aim, the technology demonstration will be shown in the most crucial selected parts of the industrial participants plants – i.e. the selected pilots, whereas the technology development can be transferred to the complete plant. The degree of digitalization aims to be 100% in the pilots at the end of the project.

The project has the following objectives of Improved Performance in Cognitive production Plants through Cognitive Digital Twins in the selected process industry types:

2.1 Hydro Pilot - Aluminium Production Process

The topic of the pilot is related to Reduced energy consumption in a selected Hydro GTC (Gas Treatment Centre).

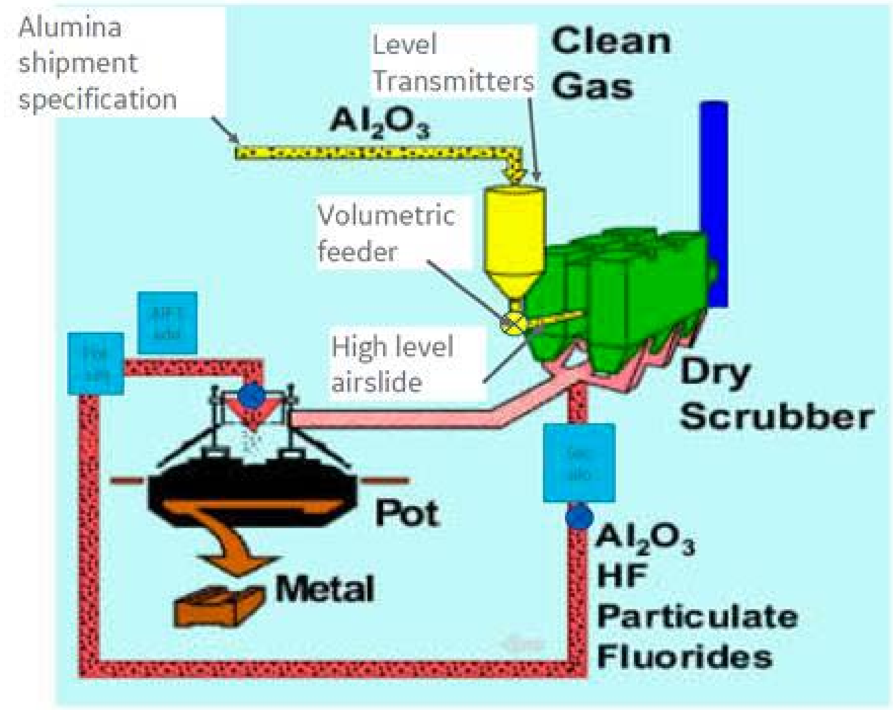


Figure 1 A schematic view of the Hall-Heroult aluminium production process.

The COGNITWIN work is related to the performance of the Gas Treatment Center (top right, Figure 1) and the interactions with the pots (reduction cells) and the inflow of fresh alumina. The ambition is to develop a Digital Twin that allows optimal operation, acceptable emissions of HF (Hydrogen Fluoride) and which can account for the variations in alumina quality from ship load to ship load.

The work will increase the overall energy efficiency as a result of a symbiosis between the actual production (electrolysis) and the gas cleaning plant (Gas Treatment Centre GTC). In particular, the work will lead to reduced environmental impact and an overall optimized energy consumption by maximizing the efficiency of the Gas Treatment Centre. By help of digitalization an average reduction in suction rate, without reducing the gas cleaning efficiency, will give a significant energy saving in the GTC. In addition, by raising the averaged off gas temperature another saving, in increased available recovered thermal energy, will be realized. Reduced energy consumption and/or replacement by thermal energy (heating) will save CO2 emissions caused by the current energy source. By improved observability and cognitive digital twins process disturbances can be reduced by help of preventive maintenance.

2.2 Elkem Pilot - Silicon Production Process

The topic is to optimize the post top-hole process in an Elkem Silicon plant.

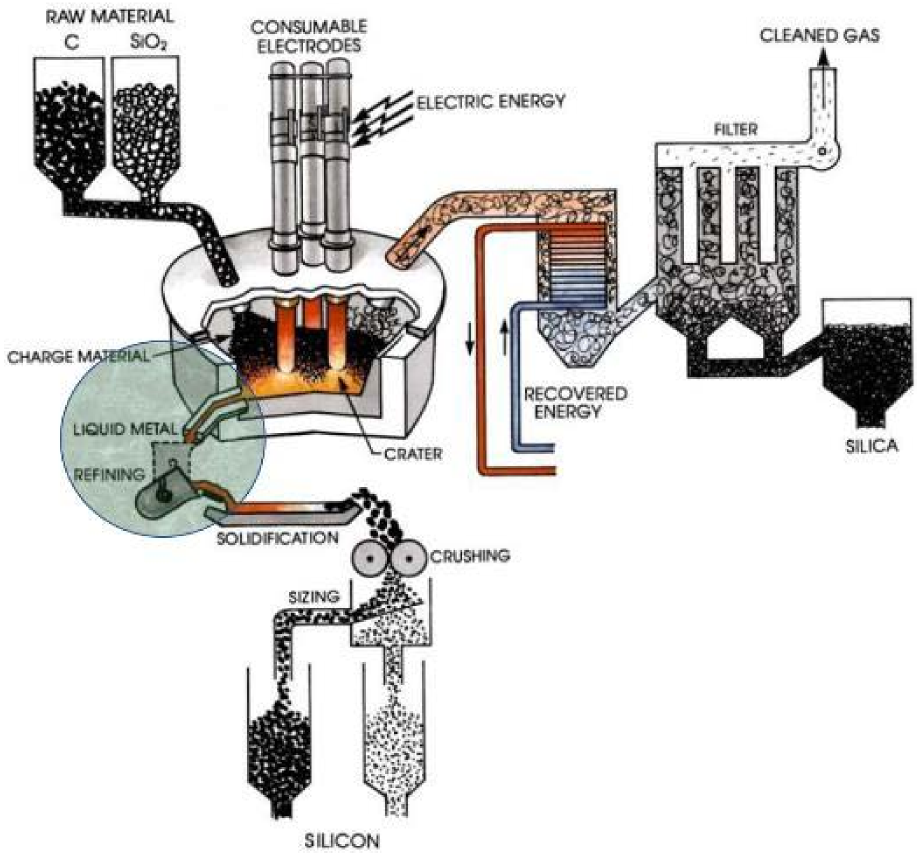


Figure 2 The figure shows the silicon process and where the post tap hole processes in question are found inside the circle (liquid metal / refining)

By application of digital technologies to the post tap hole processes (tapping into the ladle, silicon casting into molds) silicon yield can be increased, ladle lifetime can be increased, metal quality can be improved, and energy consumption reduced. By help of new measurement techniques COGNITWIN will help enabling on-line estimates of the actual silicon flow and its temperatures. Application of new and old data into cognitive hybrid models will be developed to improve the product quality due to more consistent quality and lead to more profitable operation.

2.3 Sidenor Pilot- Steel Production Processes – Ladle life time improvement

The topic is improved ladle lifetime in a Sidenor steel plant. The challenge is erosion of the refractory bricks. In Figure 3, upper part, we see a belt of dark grey refractory bricks which made of special material and which is harder to erode during the operations.

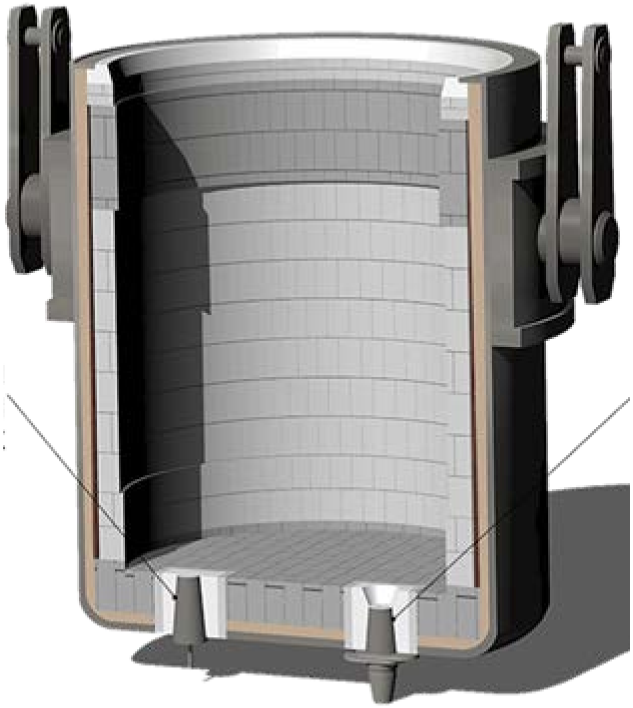


Figure 3 A steel ladle with porous bottom plugs for gas injection

In the steel plant ladles gas injection is applied for refining and stirring. It is observed that ladle lifetime varies a lot and depends on a large number of parameters. The COGNITWIN approach will be to develop a hybrid model that may exploit the large data that already exist. In addition COGNITWIN will apply physics based models that can handle the thermomechanical conditions in the ladle refractory, take advantage of the thermodynamic data for the steel-slag-refractory system, and account for multiphase and multicomponent mass transfer as well as the dynamic temperature variations in the system. Based one available data, new measuring techniques and physics-based modelling a Digital Twin for the ladle operation will be developed and used to optimize the ladle lifetime and reduce operational costs.

2.4 Saarstahl Pilot - Tracking system for rolled bars in the rolling mill

This pilot is owned by Saarstahl AG, and the topic is to provide a tracking system for rolled bars in the rolling mill

The COGNITWIN approach will enable a seamless tracking of individual billets throughout the rolling mill train, thus providing a linkage between various sensor data as well as other relational data on individual billets collected before and after the non-continuous part of the mill train. Combining the data from the rolling mill associated to the billet with data collected beforehand at the steel mill will allow the digital twin of the billet to span the entire production process and enable the twin to acquire cognitive elements/cognition. The digital resp. cognitive twin in return can then be used e.g. to optimize production processes, recognize causes for deviations and, depending on the specific situation, react in real time to prevent deviations from occurring. Another benefit of the envisaged COGNITWIN computer vision tracking system will be to detect deviations and erroneous billets.

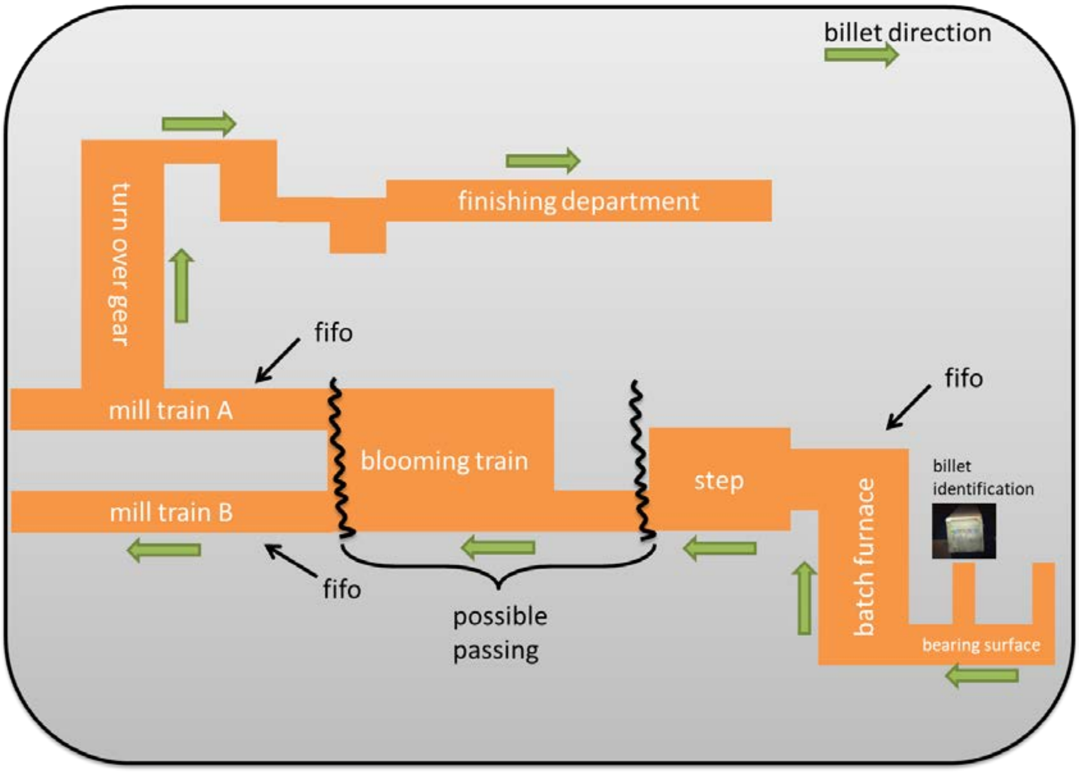


Figure 4 Schematic overview of a non-continuous Saarstahl rolling mill

2.5 Noksel Pilot - Digital Twin Powered Condition Monitoring

This pilot is owned by Noksel. The topic is to apply a cognitive digital twin to power condition monitoring (and control) in the steel pipe manufacturing industry

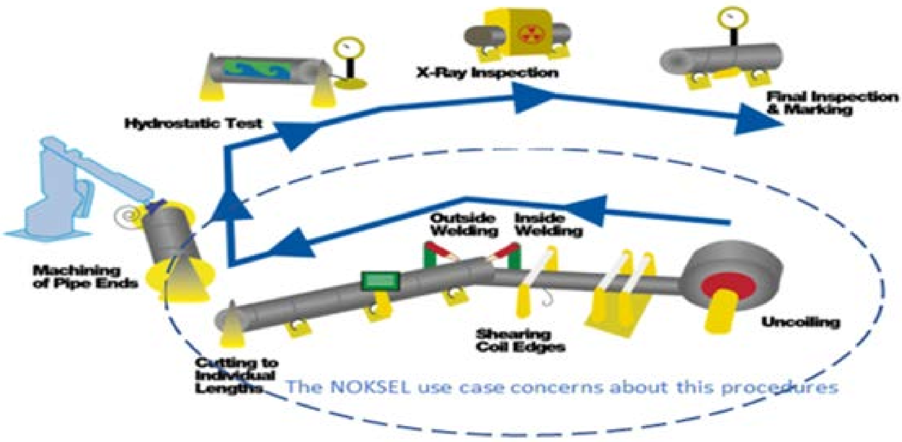


Figure 5 Noksel SWP processes for producing welded steel pipes

Noksel's pilot case aims to the development of a digital twin for the production process of Spiral Welded Pipes (SWP). The digital twin will collect and analyse multiple sensors' data in real-time, and enable a smart condition monitoring system for predictive maintenance. Real-time data acquisition, communication networks for monitoring, and automated recommendation generation are among the key innovative features of this pilot.

Smart components that use sensors to gather data about real-time status, working condition, or position will be connected to a cloud-based system that receives and processes all the data the sensors monitor. These inputs will be analysed against business and other contextual data through smart visualization systems. The digital twin model will allow joining physical and virtual worlds to create a new networked layer in which intelligent objects interact with each other to virtualize the steel pipe manufacturing process on the SWP machinery. The ambition is to reduce machine downtimes, decrease energy consumption, and increase total equipment performance.

2.6 Sumitomo Pilot - Engineering Boiler operations

This pilot case is owned by Sumitomo and where a main ambition is to allow plants to operate well even if the fuels quality and composition is changing faster than it used to do in the past. This can be made possible through the COGNITWIN cognitive digital twin development.

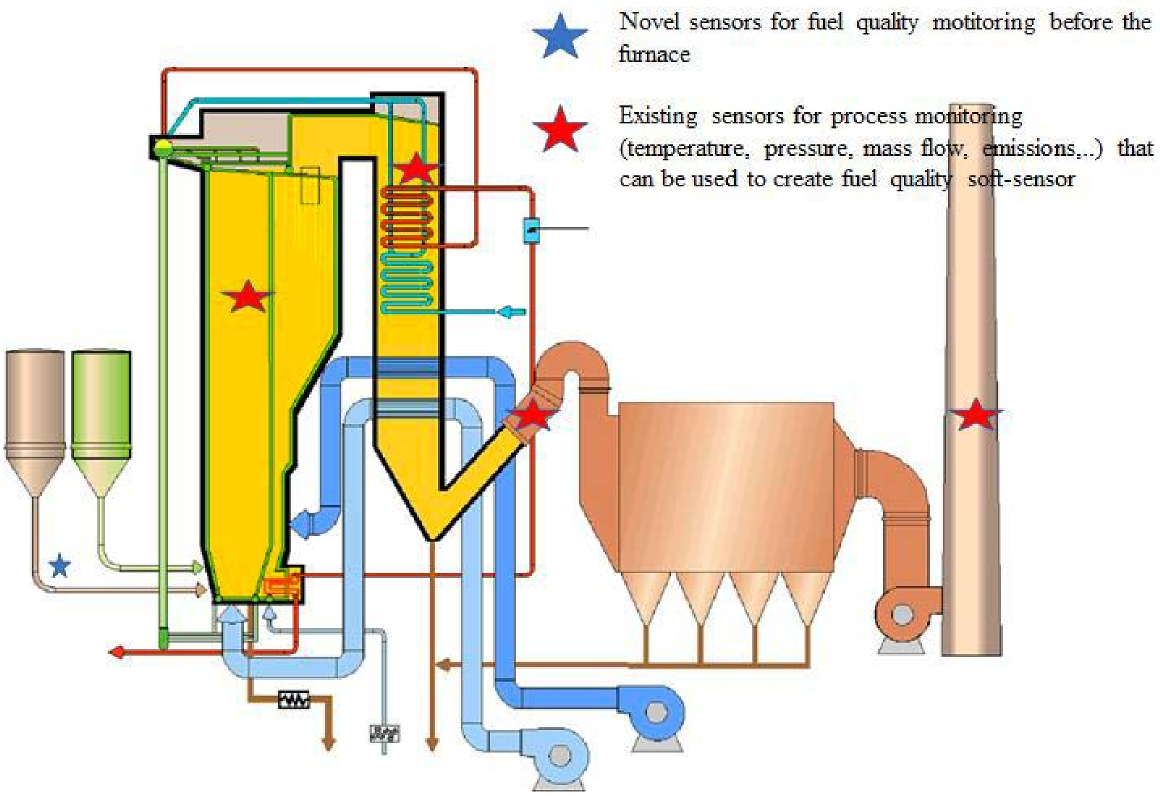


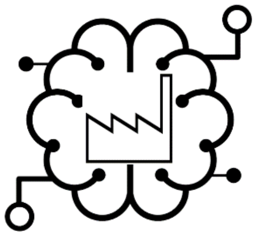
Figure 5 Overview of the boiler process in a Sumitomo made plant

The innovation and the cognitive element here is to introduce new measuring techniques, combine measured fuel quality data, process data from the power plant and existing physics based models. The developed digital twins should predict how fuel quality changes affect the process, which enables early detection of process disturbances and overall process optimization.

3 COGNITWIN Hybrid AI/Analytics and Cognitive Toolbox – overview and architecture

3.1 Introduction to COGNITWIN Toolbox architecture

Cognition is "a mental action or process of acquiring knowledge and understanding through thought, experience, and the senses" [1]. Modern process plants/operations feature cognitive capabilities are referred to as cognitive process plants. Cognitive plants leverage on advanced technologies such as, the Industrial IoT, and advanced analytics to digitize, to understand and optimize manufacturing processes[2].



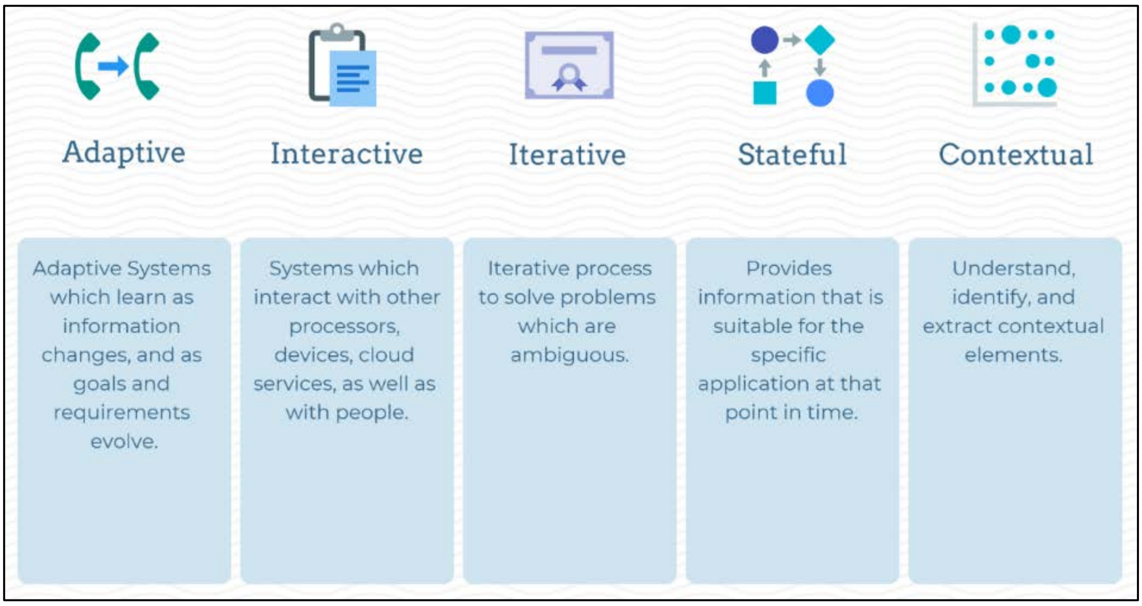


Figure 6 Different elements of cognition in an industrial context

3.2 Baseline Digital Twin, Hybrid and Cognitive Twin partner components

Figure 7 shows various technology components from the COGNITWIN partners, placed into the different WP5 areas of the COGNITWIN Hybrid AI and Cognitive Twin Toolbox.

The related deliverable D4.1, on Baseline Platform, Sensor and Data Interoperability Toolbox focuses on the underlying WP4 supporting areas of the COGNITWIN Platform, Data and Sensor Interoperability Toolbox.

[1] https://www.lexico.com/en/definition/cognition

[2] https://metrology.news/what-is-cognitive-manufacturing/

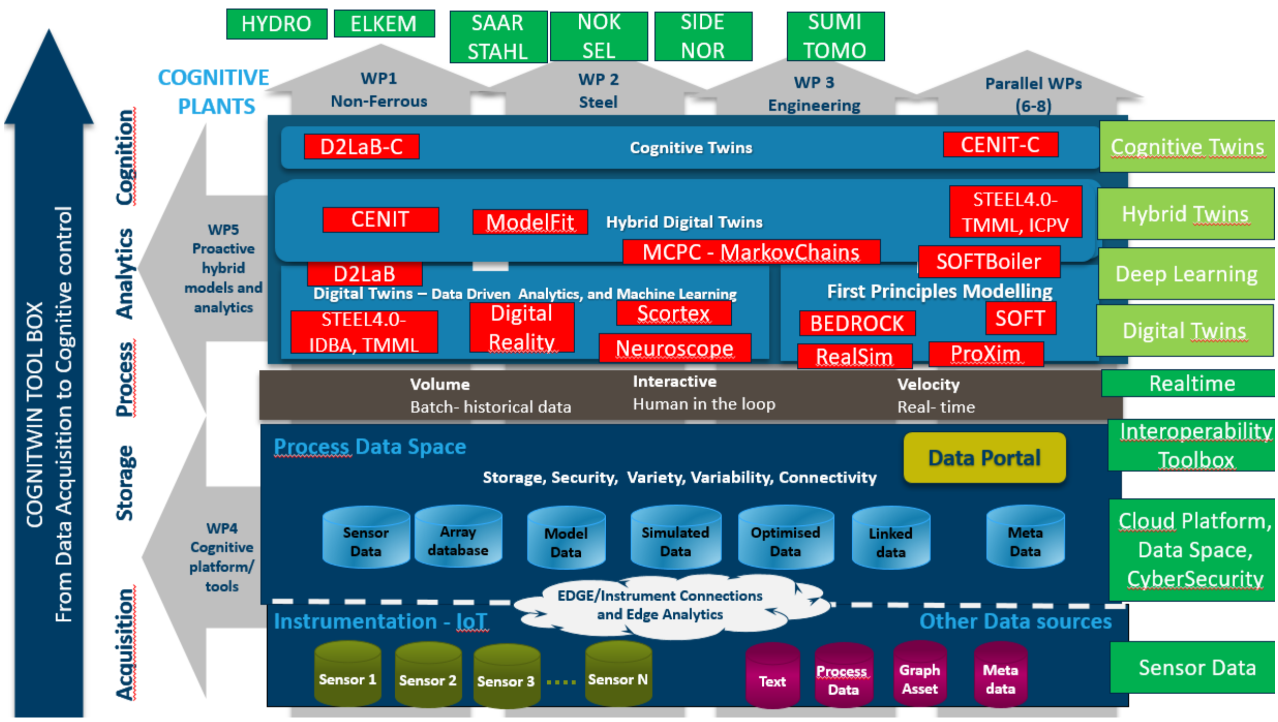


Figure 7 Technology components (in red) related to the areas of the COGNITIVE WP5

Digital Twins can be based on both Data Driven Analytics and Machine Learning and by First Principles Modeling. Data Driven Digital Twins are supported by D2Lab (by Nissatech) (described in D4.1), STEEL4.0-IDBA,TMML (by Teknopar), Scortex Image analytics/Machine learning and Neuroscope (by DFKI). First principles modeling is supported by BEDROCK (by SINTEF), SOFT (by SINTEF), RealSim and ProXim (by Cybernetica).

Hybrid Digital Twins are supported CENIT, ModelFit, MCPC-MarkovChains, STEEL 4.0-TMMI, ICPV (by Teknopar) and SOFTBoiler by OULU/Sumitomo. The emerging Cognitive Digital Twins will be supported by Cognitive extensions to some of the partner tools, including D2Lab-C(ogntive), CENIT-C(ognitive) and others.

The following sections and chapters describe these WP5 areas and some of the corresponding baseline partner components in more detail.

3.3 Twin Toolbox - Overview and architecture

The main objective of the Twin Toolbox is to support the life cycle of "digital replicas" of physical assets and processes.
We define the following phases in the lifecycle of a Cognitive Twin (CTwin):

1. Twin Creation (System/Process/Asset Behaviour modelling)
2. Twin Operation (Continuous Behaviour monitoring)
3. Twin Update (Behaviour model refinement), self-awareness
4. Twin Analysis & Management (System Behaviour analysis and optimization)

One of the most important functionalities provided by a of Cognitive Twin is the possibility to reason about the behaviour of the physical asset.

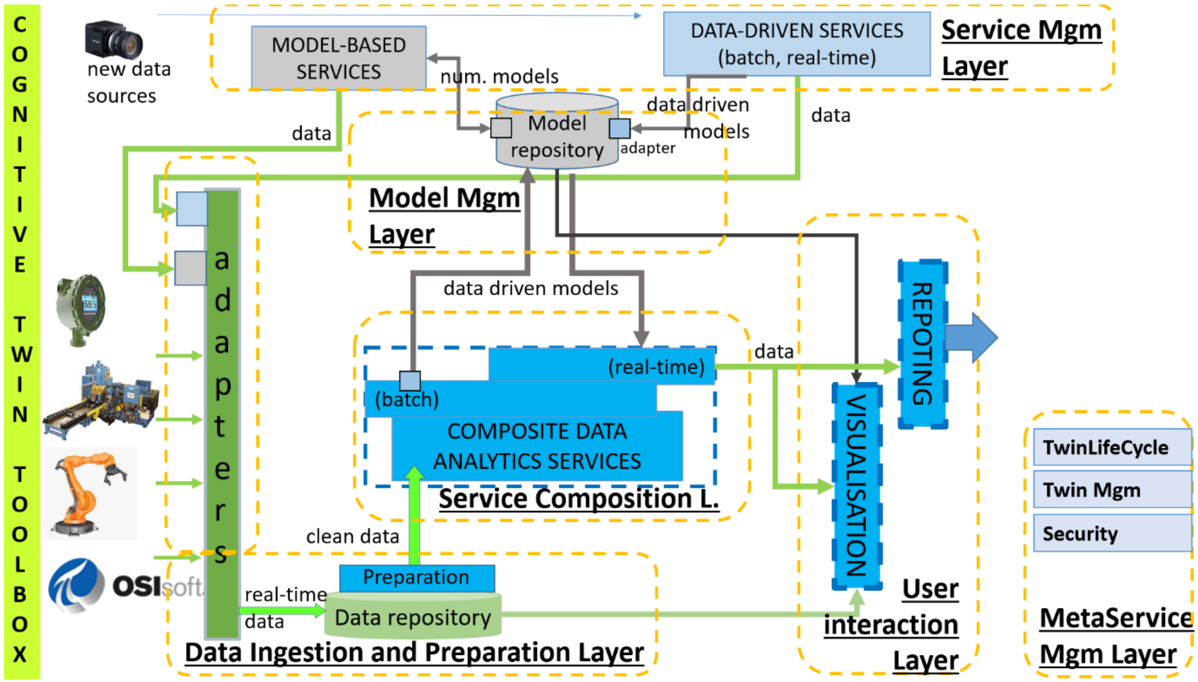


Figure 8: Five Layers in CToolbox

There are five main layers, described briefly in the following text.

<u>Data Ingestion and Preparation Layer</u>

Main role of this layer is to enable the integration of relevant data sources and the preparation of the data for further analysis. Primarily, it collects data, transforms in the TwinToolbox (TTbox) data format, stores in the Data (TTbox) repository and performs various cleansing steps to prepare data for the usage in analytics services.

Data will be published on the Broker that ensures the efficiency and scalability of the data communication.

Main components are:

- Data adapters (provided for each data source and service): enable adaptation of various data formats used by data sources and services, to the TwinToolbox data format (explained in report D4.1).
- Data repository: stores all collected data and makes it available for further processing
- Data preparation: uses different techniques for cleansing in order to make data useful for various analysis

<u>Model Management Layer</u>

One of the main advantages of TwinToolbox is that the models are treated as first class citizen, which is mainly realized by this layer. There are three main types of models:

- numerical, 1st principle models, related to the well-defined models of the system (Twin) behaviour. They describe **formal knowledge** known about underlying processes

- data-driven models, related to the models derived from past data. They describe **implicit knowledge** that can be learned using machine learning and AI methods

- knowledge-driven models, related to the **tacit knowledge** of the domain experts and human operators, based mainly on their huge working experience

Therefore, the main role of this layer is to ensure an efficient storage and access to the various types of models, provided by different services (model-based, data-driven). Primarily, it receives models from services, which are previously adapted to the TTbox model format, stores in the Model repository and manages the access to them. It includes also a registry of models for enabling various analyses of the available models (part of the model management functionalities).

Models can be published on the Broker that ensures the efficiency and scalability of the model exchange.

Main components are:

- Model adapters (provided by each service creating models): enable adaptation of various model formats used by services, to the Twin Toolbox model format (explained in report D4.1).
- Model repository: stores all collected models and makes them available for further usage
- Model registry: manages information about available models and their characteristics, based on the predefined classification of models
- Model analyser: enables complex views on the available models, based on the various comparison between models and groups (classes) of models. Important goals are to understand the dynamics of model changes (and corresponding Twin/system behaviour) and discover gaps/anomalies which can be resolved in the future (e.g. process optimization)

Service Management Layer

Main element of the Twin Toolbox are services. There are two main types of services:

- model-driven services, based on numerical, 1st principle models, delivering data created in rather complex computation processes (e.g. various types of simulations)

- data-driven services, related to the various types of data analytics services, delivering data and models which describe some aspects of the system behaviour, based on data-intensive computation.

There are two main types of data-driven services:

- batch processing services, related to the rather complex off-line learning process that works on huge amount of past data

- real-time processing services, related to fast pattern-detection or on-line learning processes, working on small-window-size real-time and streaming data

Main role of this layer is to ensure an efficient usage of all available services for resolving underlying domain problem. It is based on a complex orchestration of services, which creates added-value processing pipelines, combining data-driven and model-based services.

It includes also a registry of services for enabling an efficient discovery of services required for an orchestration process.

Services can publish the results on the Broker that ensures the efficiency and scalability of service communication.

Main components are:

- Service orchestrator: enables complex composition of heterogeneous services and various data sources to create value adding processing
- There are two types of composite (orchestrated) services:
- Batch composite services which delivers complex models of the system behaviour, requiring intensive computation processing since work on huge amount of past data

- Real-time composite services which usually apply previously learned models on the real-time data to get some important information to be reported immediately to the user
- Service registry: manages information about available services and their characteristics, based on the predefined classification of services
- Service analyser: enables complex views on the available services, providing information about (hidden) opportunities for processing

User interaction layer

Since Cognitive Twin (CTwin) represents a digital replica of a physical asset and models its behaviour, it is very important to support a user in exploring a CTwin (data, models) and its characteristics. In other words, an intuitive but explorative user interaction should be enabled.
There are two main types of users who can benefit from the interaction with the CTwin:
- process domain expert, who will use the Twin for a better understanding of the long-term behaviour of the related process/asset (esp. non-optimal behaviour) and consequently its (re-) design
- process engineers and operators, who will use the Twin for a better understanding of the medium- and short-term behaviour of the process, esp. anomalous situations and search for root causes and corresponding resolutions

Therefore, the main role of this layer is to enable an efficient visual exploration of the Twin models and characteristics for various types of users, i.e. their goals, as mentioned above. In addition, this layer supports the creation of various types of notifications
Main components are:
Visualization: offers a very illustrative and intuitive graphical user interface, enabling a deep, but a very efficient exploration of the Twin repositories (data and models).
Reporting: responsible for the generation of various reports and real-time notification that will be delivered through various communication channels

MetaService Management Layer

Since it models the behaviour of a physical asset, Twin represents a complex structure that should be managed in an efficient way. Being a digital replica of a physical asset, a CTwin should reflect its behaviour, e.g. through having model of the normal behaviour of the asset. On the other hand, a CTwin is a digital object that has own life cycle which is influenced by the physical asset, i.e. all changes in the behaviour of the physical asset should be reflected in Twin structure (esp. models), as soon as corresponding data is available in the physical world.
One of the main roles of this layer is to support the life cycle of a CTwin. We define the following four phases in the lifecycle of a CTwin:

1. Twin Creation
   - mainly dealing with asset/system behaviour modelling – using Machine Learning and AI methods for the creation of initial models from past data
2. Twin Operation
   - related to continuous behaviour monitoring – applying numerical models and models learned in the previous phase on real-time data which are sensed from physical assets, with the goal of detecting any unusual / anomalous behaviour (and their root causes)
3. Twin Update
   - focusing on behaviour model refinement – applying Machine Learning and AI for updating existing models (in the case of a model drift) or learning new ones

4. Twin Analysis
    - related to the system/asset behaviour analysis and optimization – using advanced data analytics techniques for finding opportunities for short-, medium- and long-term improvement of the physical asset

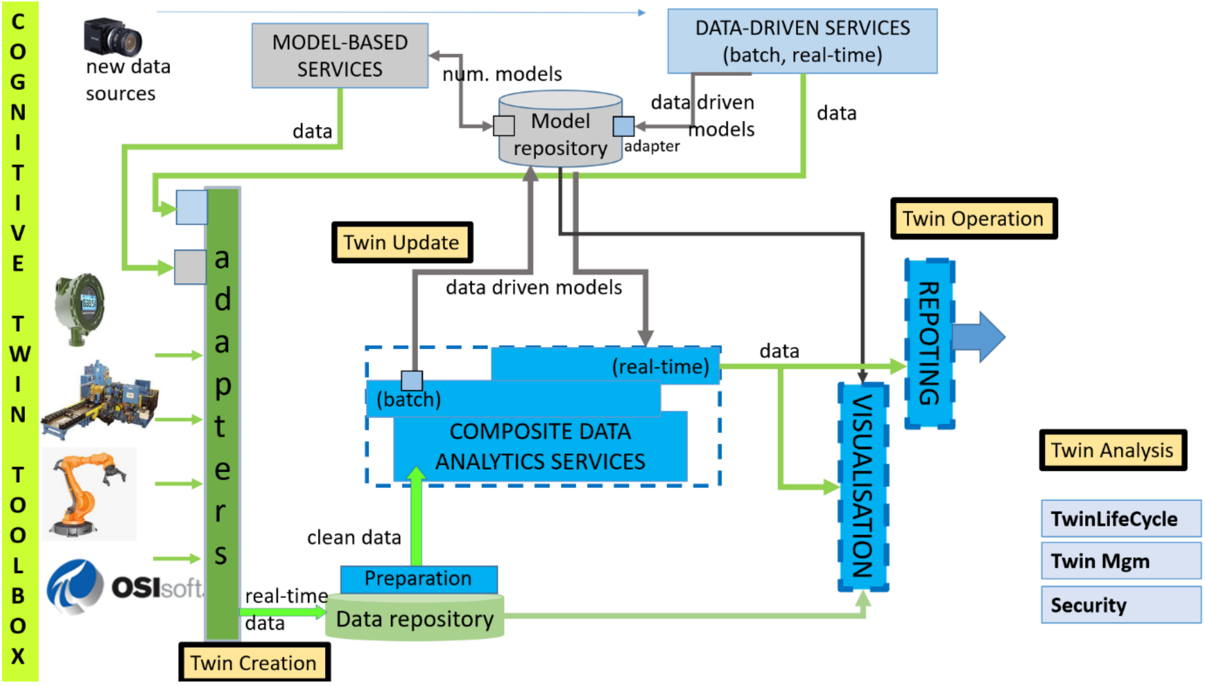


Figure 9: Supporting Twin lifecycle in CToolbox

In addition, this Layer is responsible for the ensuring the privacy of the Twin data and models and secure communication

4 Plant Digital Twins with ML/AI

4.1 Introduction

Currently, the huge potential of the digital twin technology is mainly reflected in the better design of an asset, based on the extensive simulations in various conditions. The models can be very detailed and enable powerful simulations.

In the heavy process industry, the models are often built on the basis of physical considerations, tuned and complemented in various ways by data from experimental tests. The complex industrial process environment poses several requirements, as the approaches need to be able to consider nonlinearities, variations in time delays and sparse/asynchronous measurements. The approaches need to be able to handle time-series and other sequential data. This task aims to use ML/AI methods suited for such problems and extend and/or develop new algorithms to further improve performances of the control systems. Recommendations on ML methods will be based on the works with pilot case problems.

The task on plant digital twins with ML/AI links closely with those on hybrid digital twins and cognitive digital twins.

4.2 State of the art for Plant Digital Twins with ML/AI

4.2.1 Plant DT/simulation

Today's process state estimation, monitoring, control and optimization is built on model-based methods. A model is always a simplification of real life, as relevant for the task(s) at hand. A starting point for modelling is the associated system, which provides the boundaries of the model with its inputs and outputs. The simplest models provide basic information on steady state causalities and gains, dynamic models extend to system time delays and time constants. Comprehensive plant models can be called plant digital twins, often covering large systems consisting of subsystems, etc. They can be seen as digital replicas of the real plant. Their main function is the ability to provide computer simulations of real-life plant behaviour. In many cases, also the process automation system (DCS for operation, safety, etc.) needs to be modelled to a reasonable extent. This enables what-if analysis and dedicated tasks, e.g., in state estimation, control and short-term optimization, such as in monitoring, operation/dispatching, production planning and maintenance.

4.2.2 Models/Ordinary Differential Equations

Plant models can be constructed using knowledge of the underlying physical (chemical, mechanical, etc.) phenomena and laws [1]. As an alternative, the process identification approach builds models using process measurement signals. Plants in industrial process engineering are typically slow and complex, which severely limits the application of purely data-based approaches. Therefore, a typical starting point is to construct the set of mass, energy (etc.) balances, associated with the physical transport phenomena and chemistry, etc. As plants typically are slow and continuous in time, the role of system dynamics is emphasized, leading to descriptions as differential equations.

Construction of process models can be time-consuming, and care needs to be taken to find a suitable level of detail for the model. More detailed models can require time/resource consuming computations, e.g. in iterative solutions, which also restricts the level of detailed modelling. This is pronounced in on-line applications, such as in monitoring and control. Therefore, use of specialized simplified/reduced/quantized (etc.) models, instead of the complete plant digital twin, are often a reasonable alternative.

4.2.3 Dynamics/time-series

Process identification typically relies on **time-series** structures, leading to NARX and NOE (etc.) type of model structures. Stochastic approaches may be considered to model the unknown/unmodelled quantities ([2],[3]). In these structures, the system inputs consist of delayed input and output signals, and the mapping f can be seen as a static map (ARX) or with external dynamics (OE), etc. If the mapping f is nonlinear, NARX and NOE (etc.) structures emerge. A structure composed of linear dynamics and a nonlinear static map leads to Wiener/Hammerstein model structure. The mapping f may also contain internal dynamics, leading to approaches such as recurrent neural networks, and many more.

4.2.4 Adaptation/learning/estimation

In process identification, the model structure contains a number of unknowns, which need to be estimated using plant data. One may attempt to estimate the parameters on-line, leading to adaptive and learning approaches ([4],[5]). Learning is an essential component of AI. Therefore, many of the artificial intelligence inspired approaches are well suited to applications in identification. However, many of the AI/ML approaches end up using gradient-based or random search, well known from other fields of research, fuzzifying the boundary between AI/ML and “traditional” optimization and estimation.

An important decision made during the design phase is to specify the model to be adapted. It would be straightforward if one could estimate the physical parameters in a digital twin and derive the simplified models from the up-to-date DT. However, this may be challenging in practice and the parameters of the simplified models be substantially easier to estimate from real life measurements. Model reduction may also be time-consuming to perform, e.g., in the case of on-line adaptation.

Crowdsourcing of plant information is an emerging trend, e.g. estimating unknowns in DT’s using a bank of plants with some overlapping parameters.

4.2.5 Tools/Matlab

Plant digital twins can be constructed in many ways. A typical approach is to build the basic structure of the simulation model using physical arguments, with initial parameters derived from idealized understanding. The effective parameters of the model are then sought by linking the data from real plant to the model. Such work requires a proper development environment. One such environment is provided by Matlab/Simscape. Matlab is a general-purpose coding environment, especially suitable for matrix computations and dynamic systems. Simscape provides a physical network approach for the physical modelling task, while the direct links to Matlab enable smooth transition to applications in parameter and state estimation, control and optimization. Matlab is proprietary software, but much of the code developed in Matlab language can be run using the freely available software Octave.

4.2.6 DT for state estimation methods

In complex models many of the system states cannot be directly observed, or the measurements are unreliable due to noise. Therefore, methods of state estimation are essential components in a plant digital twin development toolbox ([6]-[10]). The Bayesian approach to state estimation fuses information from both the model and measured data in a stochastic framework. The implementation depends on the type of model and assumptions on the noise. In the linear/Gaussian case, a Kalman filter can be applied. Extending towards nonlinear/non-gaussian models, extended Kalman filter (EKF), unscented Kalman filter (UKF), finite state cell filters, and various particle filters can be devised. While the modelling/assumption constraints are relaxed, the computational load increases. The cell filters provide an exact solution to the propagation of uncertainties in the Bayesian framework, but the approach is restricted by the curse of dimensionality. Particle filters can be seen as an AI-inspired population-based random search, dominated by parallel simulations.

4.2.7 DT for process control and beyond

The methods of state estimation are often closely linked with control algorithms. Model predictive control (MPC) is based on the ability to predict (e.g. simulate) future plant behaviour, for a finite time horizon ahead ([11]-[13]). The plant controls are then designed such that the cost function, defined in the given horizon, is optimized. It can be reasonable to base both state estimation and MPC on the same plant description, or (linearized, reduced, quantized, etc.) simplifications of the same model, such as a plant digital twin. Dynamic programming extends the MPC from open-loop to closed-loop optimization. Solutions can be sought using finite state Markov chain approaches (in small systems) or approximate dynamic programming (for larger systems).

DT predictions can be utilized as a part of optimization also in applications in fault diagnosis, maintenance, short-or longer-term production optimization, etc.

4.2.8 ML/AI with plant DT

The role of ML/AI with plant digital twins can be multifold ([14]-[17]). An important ability associated with ML/AI is that of learning from data. As an outcome, the ML/AI methods provide clusterings, classifications or mappings. Mappings f can be constructed from measured data, to provide those parts of the plant digital twin which are difficult to construct using physical knowledge, within a reasonable resource frame. Another common approach is to use the plant digital twin as a source of data. Here, ML/AI approaches are used to construct simplified mappings of plant digital twins, e.g. for real time control purposes. Neural networks (NARX, NOE, recurrent) and finite Markov chain models are typical candidates in this category. Clustering and classification can provide tools to analyse system properties. Methods familiar from the ML/AI context can also be applied for estimation of unknown parameters in the physical model, using data records measured from a plant.

4.3 Partner technologies

4.3.1 UOULU

The COGNITWIN algorithm development and applied work on the boiler case application by UOULU will be based on Matlab and SimScape tools, and past work on these tools. Tools on boiler furnace (hotloop), steam side dynamic models/simulators as well as tools on impacts to corrosion and fouling from Sumitomo will be available. The necessary tuning and adjustment of the models according to the Sumitomo pilot case problem is within the expertise at UOULU.

Tools also include development tools for model-based state estimation (especially on various Bayesian approaches) and control design (using MPC and ADP), as developed at UOULU. These include a Matlab library (MCPC-toolbox) on finite state Markov chain-based tools for state estimation and process control, which is available for the work in COGNITWIN and a number of other approaches developed and applied at UOULU in the context of process control engineering.
See more details in

Annex 9. UOULU tools.

4.3.2 Cybernetica CENIT

Cybernetica CENIT is presented in more detail in *Annex 7. Cybernetica Tool components*. The Cybernetica CENIT software has been used successfully to develop hybrid digital twin based applications for online estimation and nonlinear model predictive control, of which there exist several examples in literature ([18],[19]).

Cybernetica CENIT can be used to develop both soft sensors and control applications. Cybernetica CENIT implements both a variant of the Extended Kalman Filter (EKF) ([20],[21]) and a moving horizon estimator (MHE) ([22],[23]) that both can be used for online adjustment of process parameters. For control, a Non-linear Model Predictive Controller (NMPC) ([24]) is implemented.

Cybernetica CENIT is an industrially proven technology with several hundred soft sensor and controller applications already running on different industrial plants.

4.3.3 Cybernetica ModelFit

Cybernetica ModelFit is presented in more detail in section 18.2. It is a tool used for off-line model development and validation as well as off-line state and parameter estimation with Kalman filters or Moving horizon estimators. The ability to tune the models and Kalman filters off-line is an advantage as it can allow for the consideration of many different control and estimation strategies.

Cybernetica ModelFit is used industrially and is a ready-to-use tool.

4.3.4 Cybernetica OPC UA Server

Cybernetica OPC UA Server, described in section 18.6, can be used to collect and distribute real-time data in a standardized way to any application that implement an OPC UA Data Access client interface.

The Cybernetica OPC UA server can easily be extended to collect data from various data sources or proprietary protocols via plugins.

OPC stands for "Open Platform Communication" and UA stands for "Unified Architecture", which is the newest version. It is a standardized way to exchange data. The OPC UA Data Access specification is maintained by the OPC Foundation[3].

4.3.5 SINTEF BEDROCK

SINTEF BEDROCK is a software framework and underlying component for several SINTEF activities on advanced process control, digital twin, pilot plant operation and research data management for process plants.

The BEDROCK framework is a flexible, lightweight easily deployable software bundle of modules developed at SINTEF Industry applied as the foundation for digital twinning R&D activities and process

[3] http://www.opcfoundation.org

control. The purpose of BEDROCK is to enable a framework for building various process plant applications. More details are given in *Annex 2. SINTEFs BEDROCK.*

4.3.6 TEKNOPAR STEEL4.0 (IDBA and TMML)

IDBA and TMML components of TEKNOPAR support Noksel's SWP machinery and related processes digital twin by ML/AI. These two components of TEKNOPAR's STEEL4.0 provide meaningful information to support digital and hybrid twins of the SWP machinery and the related processes in steel pipe production. Collected data on production (machine), and operation (real-time and historical status data, configuration data and maintenance records) will be used to develop data driven models used to conduct descriptive, diagnostics and predictive analysis. Real time streams will also be analysed. Kafka, Casandra, Flink,NoSQL, Oracle, Python and Matlab technologies will be utilized. ICPV component of STEEL4.0 will be used for visualisation of the digital twin. In development of ICPV JSON, JSware, HTML, Java and Solidworks will be used. More details are provided in ANNEX 3. TEKNOPAR Tool Components.

4.4 Pilot requirements related to Plant Digital Twins with ML/AI

4.4.1 Non-ferrous pilots

Both non-ferrous pilots are related to high-temperature metal production ovens in full production. The pilots will require a close collaboration between the industry and research partners, systematic collection and sharing of process data and installation of new sensors required to measure essential quantities.

The COGNITWIN Hydro pilot revolves around the Gas Treatment Centre. The GTC digital twin (GTC-DT) is a data science & software development complement of the existing Digital Twin for Aluminium smelter. It will be developed leveraging the PRedix Industrial IoT platform and the related Machine Learning & Artificial Intelligence capabilities. Most necessary sensors (e.g., pressure, temperature, HF gas monitoring and ambient conditions related to humidity and weather) are installed, only lacking cost efficient and operational rough flow measurement.

The tapping process will be the main focus of Elkem's pilot in COGNITWIN. Today tapping conditions are measured using either manual sampling in the liquid silicon in the ladles or weighing the full ladles after the tapping, which has a high HSE risk. Elkem wants to develop remote operation of the tapping process, and online sensors will be giving information about furnace production rate. Similarly, real time data for the incoming metal can give information.

For both non-ferrous pilots, two main activities will be conducted in parallel: 1) Modeling of the process based on first principles and 2) initial analysis of the data (correlation analysis, testing of simple data-driven models). These two activities will provide input to one another, for instance on potential features/measurements to include and application of general domain knowledge, and will form the basis for a hybrid model further down the line. This approach can be generalized and integrated in the toolbox.

4.4.2 Ferrous pilots

Saarstahl: The Saarstahl Use-Case requires the identification of individual billets in a video data stream. In machine learning terminology, this problem is referred to as “instance segmentation on image data” and requires specific neural network architectures such as Mask R-CNN[4].

Currently, the visual debugger for neural networks (Neuroscope) developed in this project does not support instance segmentation architectures. To support this, the following requirements can be formulated:

- Enable opening a Mask R-CNN[5] or related architecture such that the network structure is visualized schematically in the architecture view
- Enable visualisations of a Mask R-CNN or related architecture such that the trained weights or the network can be inspected and understood

Sidenor: Sidenor has quite extensive industrial data. As this is heavy industry, the data will be processed, and missing / faulty data will be treated. Various machine learning methods will be assessed in order to produce a data-based prediction model for the ladle refractory lifetime, depending on the operation history.

Noksel: The spiral welding pipe production and the SWP machinery of Noksel will be the main focus of the pilot. Noksel wants to monitor the condition of the machinery in real time by means of multimodal sensors. The machinery is currently digitized with multivariate sensors and real time data is being acquired, processed and displayed in a digital twin enhanced with ML/AI. The aimed digital twin will support predictive maintenance of the machinery.

Specific pilot requirements have been formulated, including:

- Estimation of remaining useful life of the selected machine component.
- Fast and user-controlled visualization of data
- Alarms (potential defects)
- Learning from data to improve estimate of remaining useful life of the machinery
- Enable users to get reports of past, analyzed and estimated data

4.4.3 Engineering applications / boiler

A circulating fluidized bed boiler is a complex plant characterized by phenomena in combustion and heat transfer ([25],[28]). Plants are typically unique in construction, as they are adjusted to the local conditions, e.g., on fuel, size, CHP, etc. and the technology evolves in time. A particular focus in the pilot is set on flexibility and efficiency (in power production) as well as environmental issues (e.g. flue gas emissions).

The foreseen pilot problem focuses around the issue of fuel quality (to be described in WP3). As direct measuring is difficult and/or expensive, state estimation of unknown quantities is developed using plant DT and measurements available later in the process stream. Using plant models, the impacts of

---

[4] K. He et al., Mask R-CNN, Comput. Vis. (ICCV), 2017 IEEE Int. Conf. (2017) 2980–2988

[5] https://wiki.math.uwaterloo.ca/statwiki/index.php?title=Mask_RCNN

fuel quality to fouling and corrosion will be developed. Given this information, guidelines for optimal operation with available fuels are then determined.

First, the available plant knowledge/models need to be adjusted for the pilot case, leading to a plant digital twin. Matlab/SimScape is to be used for constructing the physical plant models. In order to approach the set plant development aims and KPI's, the problem is likely to be posed as a mode-based state estimation problem and extended to process control at later stages. The completion of this requires development tools for data driven process modelling and identification as well as for monitoring of process input and efficiency. Again, Matlab is a suitable platform. The DT is then further developed, with possible model simplifications, towards an engineering solution. This requires tools for process control and prescriptive maintenance to be developed. The overlap between plant DT with ML/AI and hybrid DT is significant, and the work/tools will extend to multiple tasks.

5 Multi-variate Sensor analytics with Deep Learning

5.1 Introduction

Multi-variate sensors range from spectra measurements systems, which have 1D data of 1000's of elements, e.g. FT-IR or imaging systems, which are 2D data sources that can be in the megapixel range. Multi-variate can also mean the combination of several sensors for analytics. Deep learning algorithms will be developed for pilot cases where traditional machine learning proves insufficient. The approach taken in this task will follow the Digital Reality concept of training Deep Neural Networks from in-silico data generated using parametric models. The process starts by creating partial models of Reality by modelling, capturing, or learning individual aspects such as geometry, physical properties including materials, behaviour, or lighting. The partial models are composed of parametric scenarios by manual configuration, data fitting, or machine learning. The composition is performed in such a way that all aspects of reality relevant to a specific simulation are provided. Setting all parameters of such a scenario to fixed values creates a concrete instance of the scenario, corresponding to a simulation-ready 3D scene. The scene is then rendered by a forward-simulation of the imaging process. The resulting synthetic images are used to train a machine learning system.
Using the Digital Reality approach, we gain fine granular control over the composition training data set, particularly in cases like production incidents, that happen so rarely that in-vitro data is simply not available in sufficiently quantities to train a system.
Deep Learning Reliability and Debugging: A critical factor in the deployment of Deep Learning systems to production environments is the capability of the deploying organization to react to errors of the network. Deep Learning systems are imperfect by definition (albeit they can outperform human operators in many tasks) and approach optimal behaviour only in an asymptotical way. This is acceptable in most situations given a high enough rate of correct detections. However, procedures must be established to react to situations with erroneous output that can occur a as result of the training meta parameters selection, network architecture, training data selection, or because of unforeseen situations and changed production circumstances. Understanding erroneous behaviour on

the network level is crucial to correct reaction during system operation. In this task we will transfer existing academic models for the visualization of internal states of classification networks to the practically more relevant class of networks for multi-object detection and semantic segmentation. By extending the Neuroscope software to these Use-Cases, we will provide a crucial tool for practical operations in the operation of software systems that contain Deep Learning components.

5.2 State of the art

One approach to understanding how a network works is to understand the underlying features. Feature visualization tries to improve understanding of neural networks by revealing representations of the input in hidden layers. In a network training process we do not specify important features, instead the feature selection process is done by the network. To understand individual features there are two main approaches (1) visualizing by dataset examples, and (2) visualizing by optimization. Finding dataset examples is relatively simple from an algorithmic point of view as we can search for examples from the dataset which maximally activate the desired feature [29]. The second method to find learned features in a trained network is generating representing images using optimization. Instead of searching through a dataset to find the examples, we create the images from scratch. To find out what kind of input would cause a certain behaviour, derivatives are used to iteratively change the input w.r.t the target [30]. Optimization is able to generate inputs that cause the desired effect, but it does not provide a unique answer and the results could be misleading [31]. There are good reasons to visualize what a trained model is really looking for by optimization; one reason is, it separates things causing behaviour from things that may correlate with the causes. This method also has the advantage of flexibility, as it is possible to find a representation for a group of several neurons (instead of just a single feature). On the other hand, visualizing features with optimization has several challenges. This is an optimization problem that may diverge to different solutions or end up with a high-frequency noisy pattern instead of generating meaningful images [32]. Also, we should keep in mind that achieving meaningful results is computationally expensive and needs different regulations [33]. Complex modern networks consist of thousands of units, so to generate desired results we must adjust several hyper parameters and compute the results for each individual unit.

Visualization methods in CNNs can be categorized into two groups: global and local visualizations. Kernel weights and features can be grouped as global visualizations. They are not related to a specific input and explain the desired network as a general case. On the other hand, attributions are local visualizations that are generated using selected input images. Neural network interpretability is a young field of research, so it does not yet have standardized terminology. Different literature used different names to address attributions i.e. “feature visualization” or “saliency maps”, but recently the term “attribution” is becoming more popular among others. Attributions are methods and algorithms that identify important features of inputs for a network decision [30]. These methods try to highlight important features (pixels) for the network in the input [6,7,8,9]. There are two types of methods to generate such a visualization, gradient-based method [34,38,39] and occlusion-based methods [33]. For each type several variants are available, DeconvNet [33], guided backpropagation [29], saliency maps [41,36,38], class activation map, Grad-CAM [34], layer-wise relevance propagation [39], integrated gradients [42], and Grad-CAM++ [38] are examples of gradient-based visualizations. Also, occlusion methods [33,36] usually differ in how they choose pixels to perturb, random pixels or superpixels, or how they perturb the inputs, by replacing by black, gray or blur masks [43].

For the synthetic generation of training data for neural network (Digital Reality approach) we have recently published an overview paper with a comprehensive state of the art [46] and refer to this document here.

5.3 Partner technologies

Method: Digital Reality Approach to the synthetic generation of training data.
Component: Neuroscope. For more details see *Annex 1. DFKI-Tool-Component.*

5.4 Pilot requirements related to Multi-variate Sensor analytics with Deep Learning

The requirements from the pilot refer to the multi object tracking system that is supported be the two technologies (Neuroscope and Digital Reality) presented here. The system needs to reliably track billets as they are being processed in the rolling mill under realistic production conditions. The role of the presented technologies is to facilitate this object tracking objective in two ways:

(1) By means of the Digital Reality approach, we will generate videos with a a total number of 10,000 frames to train a neural network for multi object tracking. This training data must involve a full coverage of conditions relevant to the imaging of the production, including different products, daytime, weather and lighting conditions insofar as they have effect on the image sensor output and potential changes to the factory layout.
(2) Using the Neuroscope software, any misbehaviour of the trained network will be investigated, the root cause identified and the error will be eliminated. The pilot requirement is hereby to avoid misclassifications.

The Hydro, Elkem, Sidenor, Noksel and Sumitomo pilots have no specific requirements for the task on multivariate sensor analytics with deep learning.

5.5 Initial recommendations/toolbox integration

As initial recommendations, we suggest the following integrative steps:

(1) As an initial integration with the Saarstahl Pilot, will create a first set of synthetic image and compare then to the characteristics of the sensory data from the optical cameras installed by now. Image based metrics will be used to characterize the real images.
(2) There is currently uncertainty on the quality requirements for the synthetic data for training purposes. A systematic study will be performed for this specific use-case to better understand, what aspects of the 3D renderings are relevant for the purpose of training a neural network.
(3) To integrate the Neuroscope software with platform components of other parts, we will provide the software to Saarstahl and Scortex and perform an integration test with their existing toolchains. This will provide a better understanding of operability. Resulting requirements will be formally specified.

6 Deep Learning Performance

6.1 Introduction

When moving from a TRL 5 laboratory environment to a TRL 7, computation resources become an important topic. Ideally, the developed solution should match the production line and hardware requirements in terms of speed, memory footprint and computation resources required.

A typical industry use case for defect detections may require images with high resolution (1200x1920 or 2048x2464) as well as a high frame per second (typically 20). Detection system may also need several cameras to cover a large field of view, look at the whole surface of a 3D geometric part, or need different simultaneous angles to view cavities in parts. For the solution to be affordable, the computation impact must still be as low as possible. Standard state of the art deep learning / convolutional neural networks (CNN) algorithms are composed of hundreds of layers which leads to high computation cost which in turn, makes their effective deployment in factories cumbersome. At the same time, the compute must stay at the edge as even network (uploading high resolution images in the cloud) can be too slow for real time usage.

An approach may be to downscale images resolution in order to speedup calculations. However, this is not always possible for applications where the system should be looking for small defects or objects. From our experience, down sampling images can quickly lead to loss in terms of accuracy.

In this project, we will work on deploying real life deep learning models in the factory and study this in two ways: hardware and software. First, we will replace the standard graphics processing unit (GPU) with a Field-programmable gate array (FPGA) which is more described in work package 4. Second, we will work on methods to fasten the network inferences using dedicated network architectures, compression and distillation algorithms, as well as quantizing the neural networks, a task necessary to run deep learning models on FPGA.

In summary, in this work package, we will address all required issues needed to achieve enough performance for using Deep Learning Systems in production.

6.2 State of the art

We focus here on the state of the art related to deep learning software. Report on work package 4 should have covered the hardware computation part.

There are mainly 4 approaches to reduce convolutional neural networks (CNN) computation costs, namely: architecture design, network distillation, network compression, and quantization.

The subject of convolutional network design has been prolific over the past years. Alexnet ([45]) was the first convolutional approach to achieve state of the art on the Imagenet benchmark. From there, the trend in architecture design was to be able to create deeper networks such as VGG [46], ResNet [47] or Densenet [48]. More recently, the need for inference on mobile or at the edge, called for more efficient CNN architectures, that is, for a similar budget in accuracy, minimize compute time.
This is typically what draw the design of MobileNet [49], a light architecture optimized for mobile devices. From that point, many contributions were made in this field such as Xception [50] or

Efficientnet [51]. Finally, “bag of tricks” enable the performance of light model to go further, such as architecture implementation of [52] and [53].
Finally, the study of fast inference using neural network is not limited to the image classification task as it has also been applied to detection (Yolo [54], [55] and RetinaNet [56]) or high resolution segmentation (Bisenet [57], Fast-SCNN [58]).

The second technique is network distillation. The original idea from [59] was to be able to replicate a network (the teacher) output by making another network (the student) learn to copy it. By extension, one can first train a very large network to achieve great accuracy before distilling the network into a smaller student. Several papers use this technique to compress a larger model into a smaller, more portable one ([53]).

The idea of network compression is that not all weights of a network are needed in order to make a good accuracy. Removing these weights, which is called “Pruning”, can allow to reduce network size and computation. Typically, while training, many weights converge to close to 0 values. In [60], the authors propose a scheme to iteratively prune the weights while training. In a more recent work, [61], the authors find an algorithm to automatically find the best subnetworks. Though this technique enables lower network size and memory footprint, it does not necessarily mean faster inference. Some deep learning framework such as keras, are not currently able to make the most of the sparsity induced by model compression.

In Finally, the last approach used is quantization. The idea is that the networks weights and activation may not need to be expressed in float32 to have a good precision. Some hardware such as FPGA are also optimized for quantized calculation making quantization necessary for this type of devices (see work package 4). A summary of recent methods can be found in [62]. One of the seminal paper is [63] in which the authors propose the first way to train quantized neural networks. Papers of interest on the topic are [64], [65] and [66] for further quantized models training optimization.

6.3 Partner technologies

See Annex 1. DFKI-Tool-Component, Annex 3. TEKNOPAR Tool Components, and Annex 8. Scortex Tool components.

6.4 Pilot requirements related to Deep Learning Performance

All Partners interested in using image analytics and deep learning may face at some point the need for inference speed. Elkem, Sidenor or Noksel are thinking about using deep learning or at least camera sensors at some point. Saarstahl is currently the most advanced pilot in terms of application of deep learning on images.
The Hydro and Sumitomo pilots have no specific requirements for the task on deep learning performance.

DFKI and Saarstahl are still working on what the final set up but in any case It will need to be able to run in real time as: 1) the tracking system needs to run fast enough to track bars and the faster the deep learning part is 2) the best frame per second rate can be obtained for the image analysis which will in turn benefit tracking performances. Indeed, if one or two seconds are waited for the next image, the metal bar may have moved too much and can be confused for another bar.

The current plan is to use 3 Full HD cameras to make sure the whole area is covered. These cameras will thus be streaming 1920x180 pixels images which is way larger than standard Imagenet images (224x224 pixels). For this reason, using standard neural network architectures designed for Imagenet at a high enough FPS (Frame Per Second, probably 20 FPS needed for reliable tracking) dataset is not computationally tractable. Several optimizations can be done: hardware (see work package 4) and software (deep learning optimization, this work package).

For the Sidenor case thermal video, giving surface temperature distribution of the outside of the ladle, may become available during the project. It may of interest to explore the presented deep learning capabilities for analyses of such images.

For the Noksel use case, there are requirements related to deep learning algorithms' performance measurement. The below provided requirements are accepted and partially implemented.

Noksel T5.3.1- TEKNOPAR's digital twin will calculate different metrics of deep learning algorithms' performances.
As an authorized user, **I want** deep learning algorithms metrics (TP, FP, TN, FN, Precision, Recall, F1-Score, AUC) to be calculated **so that** I select the one(s) I prefer based on the performance values.

Noksel T5.3.2- TEKNOPAR's digital twin will enable users to monitor and compare deep learning algorithms' performances in numbers and in graphs/charts.
As an authorized user, **I want to** monitor the performance of the deep learning algorithms presented in charts and numbers **so that** I select the one(s) I prefer based on the performance values.

6.5 Initial recommendations/ toolbox integration

Our recommendations are the following.

When starting a new deep learning project, one should not start by having the inference time and production requirements slow down the speed of iteration. It is better to start the project with:

1. A well-known and documented architecture, available in one of the deep learning frameworks (keras, tensorflow, pytorch, etc). Do not design your own architecture as it may introduce bugs or hinder convergence.
2. Preferably, use pretrained weights to allow faster convergence and have better accuracy and more robustness in low training data regime

Once a satisfying performance has been achieved, one can then wonder about production requirements, necessary hardware and inference time.

In order to be able to make your application faster:

- First, look at the available hardware. Optimized hardware may be available. The price of hardware should be balanced with the necessary research and development of speeding network techniques

- Then, consider if there are optimization that are task dependent but not network architecture dependent. The larger driver of inference time is the input image size. One must first consider if it is possible to lower this image size without losing too much precision by downsizing the image or inferring only on a region of interest
- Third, consider moving from your first architecture to another lighter one, also available on the internet. Typically use a MobileNet (or more recent light architecture) as a backbone.
- If the issue is network size or memory, consider compressing this network with an online compression library.
- To improve performance of this network, replace the training from scratch by a distillation scheme using your best network.
- If none of the above work, consider creating your own architecture by removing layers from standard architectures. Preferably, use well studied, standard blocks such as ResNet blocks.

In the project, we will work on a library that allow some of these functionalities.

7 Hybrid Digital Twins

7.1 Introduction

This task can be divided into three parts: 1) enhancement of hybrid digital twin technology with additional AI and machine learning functionality, 2) development of "soft sensing" applications based on hybrid digital twins, and 3) development of advanced, predictive and self-learning control applications based on hybrid digital twins.

7.1.1 Definition of hybrid digital twins

In the process industry, digital twins use a mathematical model to represent a physical process and/or unit operation. A monodisciplinary approach to developing a digital twin uses either detailed knowledge of the physics and chemistry of the process or data-driven methods to analyse input and output process signals. A hybrid version of the two approaches gives rise to a mathematical model that incorporates both physical governing equations and techniques from data science.

Several different approaches to physical, first-principles based models exist. A static definition of the process is possible, where the digital twin stores information on past and present system states. Static models can be useful for identifying trends and discovering anomalous situations in process data. However, the inability to relate current and future system states renders static models impractical for use in controls and optimization applications. For controls purposes, models should be dynamic and system states are appropriately described by differential equations. In the process industry, these differential equations are generally system-specific formulations of momentum, mass and energy balances. Such models are often called "white-box" process representations.

White-box process representations have several advantages. When designed well, white-box models produce results that are understandable for both model developers and process operators and can be related to the real world. Model parameters should similarly be rationalizable based on either engineering intuition or physical measurements. White-box models require the setting of realistic boundaries for input data and model states. While these input/output boundaries can simplify the model and results, they can also be a disadvantage because they limit the scope of the model predictions. White-box models developed to monitor and control a certain physical process are likely not suited to handle unusual and unexpected disturbances and will therefore fail when presented with anomalous data.

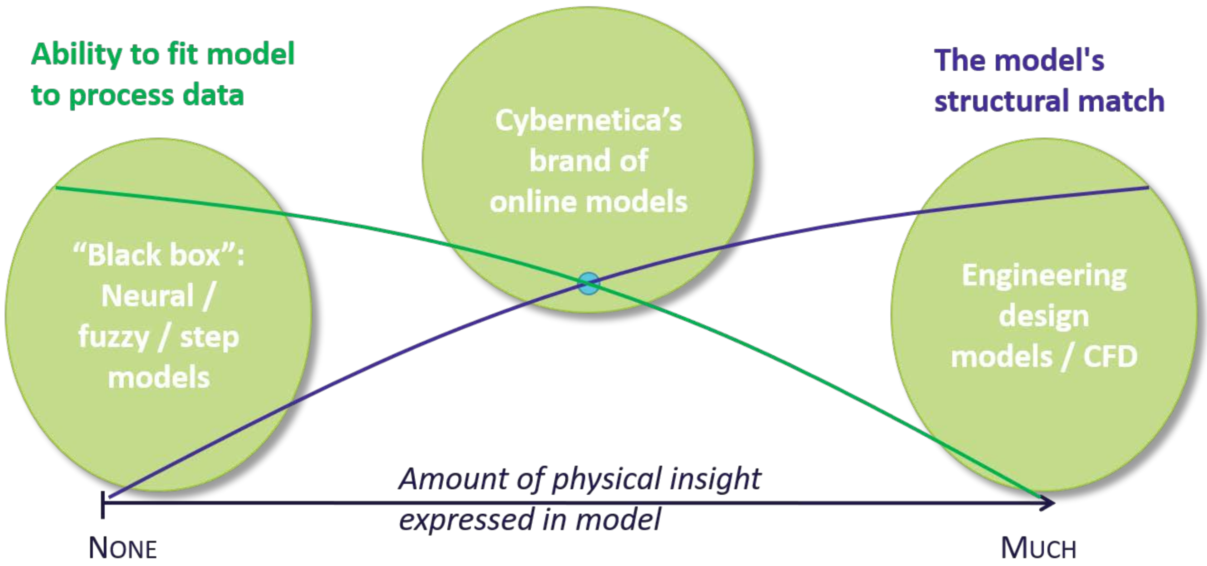


Figure 10 A schematic showing hybrid models as implemented in Cybernetica's software platform (Cybernetica ModelFit and Cybernetica CENIT).

The purely data science approach to digital twins does not take the process structure or physics into account at all, leading to a "black-box" representation of the process. Traditional data-driven approaches such as machine learning belong in this category. As with white-box models, black-box process representations have both advantages and limitations. Black-box models have the potential to be both versatile and unbiased by engineering design flaws. However, they are not by design physically rationalizable and it can be difficult to gain insight into their structure and dependencies. Furthermore, fully understanding why and when purely data-driven methods succeed and fail is presently an unsolved research challenge. Hence, applying them online to actively control unit operations in process industry systems with high safety requirements is currently not advisable.

Perfect process knowledge is never fully achievable. Complex systems, highly non-linear effects, unknown parameters and time delays are some of the challenges resulting in inaccurate model. In the scope of this project, mechanistic, white-box models resulting from first-principles will serve as a basis model for hybrid applications. This basis model will be supplemented by either data-driven sub-models or data-driven self-adaptation, resulting in "gray box" modeling via a hybrid digital twin.

The tools for development and maintenance of hybrid digital twins will include software for off-line, optimization-based parameter estimation and model validation. On-line data will be used for continuous correction of the model (combined state and parameter estimation) and prevention of model performance degradation over time. New functionality based on data-driven methods will enable detection of changes in equipment and sensors and facilitate self-learning and adaptation to process and feed-stock variations

7.1.2 Soft sensing applications

Physical sensors have several shortcomings. First, certain sensors are expensive and may constitute a large part of the total cost of a control system. Other sensors introduce errors such as stochastic noise and biases. Furthermore, certain states and signals are simply too challenging to measure directly and must be estimated. A convenient application of hybrid digital twins is the development of soft sensors, wherein the hybrid model estimates a state variable that may not be available as a physical measurement. Soft sensor applications will therefore facilitate on-line prediction of unmeasurable variables for control and information purposes.

Soft sensor applications are particularly advantageous because they allow one to approach the challenge of unmeasurable variables using both first-principles modeling and data-driven techniques. One of the strengths of data-driven methods is their ability to learn complex nonlinear correlations. They are therefore excellent tools for estimating unknown quantities given measured data. Soft sensor applications based on hybrid digital twins will use available measurements to perform dynamic model calibration in order to follow process variations. They will also have functionality to adapt to data from various sources with different data collection frequencies and varying measurement delay.

7.1.3 Predictive and self-learning applications

Tailor-made plant models will be developed for the different use cases, consisting of mechanistic models combined with data-driven methods, utilizing the available measurements in an optimal way for system identification and control. The models and the identification methods will be integrated in the Cybernetica CENIT software and demonstrated in simulations for relevant use cases (WP 1-3).

Current technology will be extended with improved cognitive capabilities, as discussed in more detail in section 8. The model’s prediction errors will be monitored and analysed using data-driven techniques. In this way, we expect to be able to detect abnormal situations where the hybrid model no longer is valid (fault detection), and in some cases determine the cause (fault isolation). As a special case, instrument failure should be detected, and faulty measurements prevented from entering the control system.

The system will be adaptive, with learning capabilities as well as predictive. The methodology builds on existing technology for nonlinear model predictive control. The flexibility afforded by data-driven approaches will make hybrid model self-learning more robust. Methods such as neural networks can also be used to identify parts of an ODE model and may be able to fix flaws in model structure. Extensions will be made, to incorporate adequate actions as response to new, cognition-based information on abnormal situations.

7.2 State of the art

Mathematical models based on physical principles are useful tools for the purpose of estimation and control. However, designing a model that remains robust and accurate under changing process conditions is a challenge. Assumptions and simplifications that are valid under certain circumstances do not always hold. As a result, state of the art technologies that use models for process optimization and control incorporate data-driven methods to improve model adaptability.

7.2.1 Offline and online parameter optimization

A hybrid digital twin can be established by first developing a basis model from first-principles and using state and parameter estimation with logged input and output process data. When evaluating mechanistic models of physical processes, there exists uncertainty associated both the process input and the model itself. State and parameter estimation is a strategy for addressing this uncertainty in and adapting the model to better follow process data. Selecting which states and parameters to optimize based on real-time process measurements is non-trivial for complex models, and the decision is often made by a trial-and-error procedure and/or engineering intuition. There is room for improvement in the scope of this project for data-driven identification of appropriate states and parameters for estimation.

State and parameter estimation using Kalman filters are implemented in Cybernetica ModelFit (offline, described in section 7.3.2) and Cybernetica CENIT (online, described in section 7.3.1).

7.2.2 Sub-model identification

If a part of the process operation is not possible or feasible to model by first-principles and there exist relevant measurable input and output signals, it may be possible to apply black-box modelling to only this part of the process. In this way the model incorporates both mechanistic and data-driven methods, but unlike the case of state and parameter estimation a part of the model structure itself is data-driven.

7.3 Partner technologies

7.3.1 Cybernetica CENIT

Cybernetica CENIT is presented in more detail in *Annex 7. Cybernetica Tool components*. The Cybernetica CENIT software has been used successfully to develop hybrid digital twin-based applications for online estimation and nonlinear model predictive control, of which there exist several examples in literature ([67],[68]).

Cybernetica CENIT can be used to develop both soft sensors and control applications. Cybernetica CENIT implements both a variant of the Extended Kalman Filter (EKF) ([69],[70]) and a moving horizon estimator (MHE) ([71],[72]) that both can be used for online adjustment of process parameters. For control, a Non-linear Model Predictive Controller (NMPC) [73] is implemented.

Cybernetica CENIT is an industrially proven technology with several hundred soft sensor and controller applications already running on different industrial plants.

7.3.2 Cybernetica ModelFit

Cybernetica ModelFit is presented in more detail in section 18.2. It is a tool used for off-line model development and validation as well as off-line state and parameter estimation with Kalman filters or Moving horizon estimators. The ability to tune the models and Kalman filters off-line is an advantage as it can allow for the consideration of many different control and estimation strategies.

Cybernetica ModelFit is used industrially and is a ready-to-use tool.

7.3.3 Cybernetica OPC UA Server

Cybernetica OPC UA Server, described in section 18.6, can be used to collect and distribute real-time data in a standardized way to any application that implement an OPC UA Data Access client interface. The Cybernetica OPC UA server can easily be extended to collect data from various data sources or proprietary protocols via plugins.
OPC stands for “Open Platform Communication” and UA stands for “Unified Architecture”, which is the newest version. It is a standardized way to exchange data. The OPC UA Data Access specification is maintained by the OPC Foundation[6].

7.3.4 SINTEF Pragmatic framework

The SINTEF framework for "Pragmatism in industrial modelling" in presented in more detail in Annex 6. *Pragmatic framework for development of hybrid models*. This is a proposed systematic work procedure that can be applied to any development of hybrid digital twins. This a dynamic approach where both physical based models and data driver models are applied in the most efficient manner to respond to the demands from the problem owner. A pragmatic model is in most cases using a mix of existing models and software, new development where necessary, data driven models and model corrections from data (extended Kalman filters[7], data assimilation[8]). The "Pragmatic Model" can use information from any source available in order to deliver the requested predictions and at the required speed and accuracy.
A method for applying data to improving non-linear model closures is presented in Annex 5. Machine learning for hybrid models (SINTEF). The proposed methodology is generic and can be applied to any process model where models can be supported and approved by application of data. SINTEF has also experienced that neural network-based machine learning will in some cased not be as efficient as classical methods such as the "least squares method" [9] method, attributed to Carl Friedrich Gauss. This method is an integral part of the Pragmatic framework.
In digital twin application where different software are to be applied in a web of service providers, requesting data and providing data from each other for the overall benefit of the final twin application. Normally, the relevant pieces of software may not be able to exchange data due to differences in input and output formats and units. The solution to this is to introduce interoperability, as explained in *Annex 4. SINTEF Open Framework and Tools (SOFT)*. If the digital twin application is built on SOFT, the data exchange may be done seamlessly as long as plugins have been developed for the software’s data formats and data units.

7.3.5 SINTEF Open Framework and Tools (SOFT)

SOFT is a framework for semantic interoperability of scientific software and is detailed more in *Annex 4. SINTEF Open Framework and Tools (SOFT).* SOFT is a datacentric modelling framework with special focus on information interchange in multi-scale-based applications. However, SOFT can also provide the "glue" in a digital twin that requests data from many different sub applications or sensor data. SOFT is designed to accommodate for a non-homogenous set of in-house open source and/or proprietary simulators, often written in different programming languages, and using different data formats. The complexity and diversity of such a system required SOFT to provide formal schemas and

[6] http://www.opcfoundation.org

[7] https://en.wikipedia.org/wiki/Extended_Kalman_filter

[8] https://en.wikipedia.org/wiki/Data_assimilation

[9] https://en.wikipedia.org/wiki/Least_squares

structures of meta-data that allows for information interpretation regardless of the original storage format, the application that produced the data and the application that processes the data. It has been proposed a standard for data exchange by separately describing meta-data specific to different knowledge domains.

SOFT, via a mechanism of plugins, offers the possibility to utilize different tools for storage of such data and meta-data. Further, SOFT facilitates scientific software development by a clear separation of numerical routines and platform-dependent input, output and analysis routines. Automated testing and simulation data analysis are also achieved in SOFT via external plugins and interfaces to scripted languages such as Python and JavaScript.

7.3.6 TEKNOPAR STEEL4.0

STEEL4.0 of TEKNOPAR aims to support the hybrid twin of Noksel's SWP machinery and related processes. Statistics and ML/AI will be applied together with 3D model visualisations. The utilized technologies include but not limited to Matlab, Pyhton, Java, Flink, Zeppelin, Kafka, FIWARE, Cassandra, JSware, IDOC. More details on STEEL4.0 components and technologies are provided in Annex 3. TEKNOPAR Tool and Components.

7.4 Pilot requirements related to Hybrid Digital Twins

The following sections describe the pilot requirements that are specific for hybrid digital twin, and come in addition to the requirements described in sections 4.4 (Plant digital twins) and 8.4 (Cognitive digital twins).

7.4.1 Hydro case

A hybrid digital twin model will make it possible to predict anomalous behaviour in the fluoride recovery process and compensate appropriately for detected issues. The mechanistic basis is a model of fluoride emissions from electrolysis cells and is grounded in literature ([74],[75]). Data-driven methods contribute to of the hybrid digital twin in two ways: 1) predictions for future weather conditions (temperature and absolute humidity) will come from a data-driven, correlational model, and 2) Kalman filters will be applied to select variables in the mechanistic model. In this way, the hybrid digital twin for fluoride recovery at the Karmøy technology pilot (KTP) will incorporate both of the state-of-the-art data-driven methods for hybrid models of interest in this project: sub-models and state and parameter estimation.

Preliminary data sets for the development and tuning of the mechanistic fluoride emissions model are under preparation and will include composition and temperature measurements of the gas leaving the electrolysis cell. Weather data for the data-driven temperature and humidity model is being collected from Yr.no and the weather station at Karmøy.

7.4.2 Elkem case

A hybrid digital twin enables the possibility to predict the behaviour of the ferrosilicon refining process, and to optimise the amount of recycled materials and slag formers utilised to control temperature and chemical composition. A mechanistic model of the ferrosilicon refining process will be combined with one or more data-driven methods, such as machine vision applied to thermal images, Kalman filter and the moving horizon estimator. This hybrid digital twin of the ferrosilicon refining process at Elkem

Bremanger thus utilises physical knowledge of the process in combination with advanced methodologies for data-driven approaches.

The development of a mechanistic model suited for online use is currently under way and will describe the temperature and composition of the ferrosilicon during refining. Preliminary tests with thermo cameras for real-time temperature measurements have been performed, which will be exploited by the hybrid digital twin to improve accuracy.

7.4.3 Saarstahl case

The Saarstahl case is an example of a hybrid digital twin. Several different aspects (conditions on the production environment) are known or captured during the project. These including geometry and surface texture of the production plant, lighting condition, camera setup (lense system and sensor) and the shape, optical appearance, and movement characteristics of metal bars.

The usage of the different aspects in this use-case is indirect, insofar as the generate synthetic training data that can be used to train neural networks for instance segmentation, which then can be used to implement an object tracking system. The requirements of the pilots are that the generated training data covers the relevant aspects of the production environment relevant to training effective instance segmentation networks.

7.4.4 Sidenor case

The Sidenor case can be a strong example of a hybrid digital twin. Here major elements and the physics and chemistry are known (see report D2.1). A physics-based model can be assembled from existing sub models or built from scratch if needed. In this case there will be elements in the sub-models that are not well known. These elements may be understood by data and ML. By exploiting the physics-based model, and supported by both existing and new data, this will open for a demonstration of a combined physics and data driven model.

7.4.5 Sumitomo case

The Sumitomo case is extremely complex. However, some knowledge exists about the physical relations between composition of fuel, ash components, particles and deposition and fouling (see report D3.1). Similar knowledge exists about the phenomena that limits corrosion. This opens for a combinations of physics-based modelling and data-driven modelling, with applications in monitoring and control, and where the casualties in the system can be explained to a large extent. The development of tools on identification and state estimation is required by both tasks on plant ML/AI and hybrid digital twins.

7.4.6 Noksel case

The Noksel case aims to reduce machine downtimes, decrease energy consumption and increase total equipment performance. For that purpose, a digital twin that collects and analyses multiple sensors' data in real time has been developed, and a smart condition monitoring system for predictive maintenance has been enabled. Data driven models for descriptive, diagnostics and predictive purposes are being developed and will be utilized. For the 3D visualisations of the SWP machine's 3D models, hence model driven twin toolbox elements will be generated. As a result, the developed hybrid digital twin model will join physical and virtual worlds to create a new layer in which intelligent objects interact with each other to virtualize the steel pipe manufacturing process on the SWP machinery. In efforts to develop a cognitive digital twin for the Noksel case, knowledge graphs will be developed and experts' knowledge will be embedded into the hybrid digital twin.

7.5 Initial recommendations/ toolbox integration

The first step is to build a preliminary mechanistic model for the industrial cases where applications in the CENIT platform will be developed. Model tuning will proceed as data is collected from the industrial cases, eventually leading to the identification of model state variables and parameters for estimation that can best take advantage of available measurements.

8 Cognitive Digital Twins

8.1 Introduction

The focus of this will be on an enhancement of the hybrid twin, providing cognitive capabilities in order to support unpredicted behaviour of a system by ensuring close to optimum performances. Using a cognitive twin coupled with hybrid analytics (see Task 5.5) and optimisation is a crucial requirement for refinery of the future. In addition to monitor the production processes and to capture events (anomalies or alerts), such a twin should be capable of: (1) quantify the impact of identified events on the current plan and (2) provide alternatives for handling the events and evaluate these alternatives.
This requires more "cognitive augmentation" of assets for enabling continuous, "on the fly" process improvement. A long-time goal is to develop an industrial standard for development of digital twins.
The goal is to be able not only to maintain a certain behavioural level, but even to improve the behaviour in uncertain, time-variant environments. This requires fusion of the internal knowledge represented by the hybrid digital twin with knowledge about the "external" world, e.g. problem-solving methods.
This vision will be realised by the meta-reasoning (cognition) which covers two aspects:

- Self-reflection, i.e. automated discovery of an unknown situation by monitoring and assessing the environment and its own behaviour. This will be done by using the hybrid digital twin as well as the knowledge about the environment, experience gained by applying model/knowledge, etc.
- Self-adaptation, i.e. ability to learn how to react in unknown situations based on defined own goals. This will be done by finding new ways to solve these goals (including combination of several goals or conflict solving) and by triggering appropriate changes in own system parameters, algorithms, structure, etc.

The goal is to identify what behaviours might be modified to better adapt the system to its environment. The adaptation will be done after self-reflection, where an abstract representation of the desired behaviour is processed. This does not necessarily result in new algorithms but in changing parameters of the engaged algorithms to alter their behaviour.
To augment hybrid twins with cognition, we will introduce a new level of processing, which will:

- interpret the hybrid models and identify important contexts and suitability for use in regulation
- validate the models quantitatively (by testing new data) and qualitatively (evaluated by experts)

- suggest appropriate strategies consisting of one or more control actions
- provide guidelines and planning assistance.

The output of the cognition process will be transformed into actionable information for real-time adjustment of the most significant measures in the process. In some cases, it may be possible to make multiple fast runs of the digital twin model to investigate several scenarios which can be used to decide what would be the best option. The result will be enhanced and updated process information for use by operators or directly in the automatic control system.

8.2 State of the art

8.2.1 Topic 4: Cognitive Control: Data-driven self- adaptive control model

Challenges for the Process Industry

Mathematical and statistical models based on directly measured process data can significantly help improve process control and operation and this requires expertise in sensor technology, analytics and machine learning, skills that are severely lacking in the process industry today. Small systematic deviations in the process data can be detected by sensors, providing the model with essential data resulting in early warnings and data-driven error prevention system. The outcome of such an intelligent system will be a new and autonomous process control resulting in a stable and predictable process with optimal resource and energy utilization. In addition, the industry can also improve on production planning, minimize waste and be able to have a uniform end-product quality.

State of the Art

To understand, monitor and control a physical process, it is essential to gain knowledge of the states of the system. However, physical sensors have several shortcomings. First, certain sensors are expensive and may constitute a large part of the total cost of a control system, in addition to inducing errors such as stochastic noise and biases. Furthermore, certain states and signals are challenging to measure directly and must be estimated. The extended Kalman filter is an example of an estimator for nonlinear systems used to estimate missing states from indirect and noisy measurements. As pointed out in the noise terms in Kalman filters can be very difficult to estimate, as the noise is usually the result of a number of different effects such as model inaccuracies, discretization and the existence of hidden states. In several applications, the noise is assumed to be independent over time, whereas in reality the mentioned effects cause highly correlated noise.

Our approach

Many estimators and controllers are based on the system model which is either inaccurate, non-complete or non-existent for complex systems. Furthermore, it is desirable and cost-effective to be able to estimate and control the processes without installing and/or developing new sensors, but instead utilize years of available data to learn the relevant representations through use of ML. This project will develop data-driven estimators and controllers for the process industry which incorporate physical principles and limitations of the actual process. This will result in new knowledge and methods for estimation of processes characterized as highly nonlinear, with sparse and delayed measurements with a high degree of uncertainty.

Nonlinear Model Predictive Control (NMPC). It is primarily based on mechanistic modelling, but has functionality to combine such models with data driven techniques. Models are developed specifically for the purpose of online control, to attain a numerically efficient "digital twin". CENIT updates model states and parameters online from available measurements, using techniques like Extended Kalman Filters (EKF) and Moving Horizon Estimators (MHE). This allows applications to be adaptive and to a high extent "self-learning". Cybernetica CENIT is industrially proven (TRL 9), through a wide range of applications worldwide.
In this project, a cognitive extension of CENIT is planned – "Cybernetica Cognitive CENIT".
Cognitive techniques will be used to monitor and analyse the process models' prediction error. In this way, one expects to be able to detect abnormal situations where the process model no longer is valid (fault detection), and in some cases determine the cause (fault isolation). As a special case, instrument failure should be detected, and faulty measurements prevented from entering the control system. The controller part of Cybernetica Cognitive CENIT system will be predictive, adaptive, and with learning capabilities. Extensions will be made, to incorporate adequate actions as response to new (cognitive based) information on abnormal situations.

8.2.2 Topic 5 COGNITWIN for Plants optimization

Challenges for the Process Industry

A typical manufacturing process involves successive or parallel steps transforming raw material to a final product. A product is the result of a Bill of Resources (personal, equipment, energy, etc.), Bill of Material (raw materials and consumables) and Product Production rules (recipes used to instruct a manufacturing operation how to produce a product). Process optimisation can occur in several ways:

- Optimisation of recipe specification and raw materials fed into the process
- Improvements to process control, resolution of quality issues, and equipment failure prediction
- Gaining process insight generation through data and analytics

State of the Art
So far, process engineers rely on their experiences and scientific knowledge to address these challenges in their daily operation. There is significant ongoing research in extracting process information from data, with Artificial intelligence (AI) and machine learning (ML) showing much promise. Hybrid systems that aim to combine these new AI/ML methods with more conventional monitoring and control functions are focus for basic research [76].

Our approach
The project intends to develop a data-driven generic tool for quality parameters optimization. Different statistical and machine learning techniques will be evaluated including classification, clustering, regression, reinforcement learning, deep learning etc.
SINTEF has experience also in physical modelling of relevant processes. The performance will be evaluated using data made available by the industrial project partners.
The plantwide control methodology provides a suitable framework for processes optimization by appropriate control structure design. The control structure design is to select the best set of variables

to be controlled to obtain robust operation that ensure close to optimum performance. The realization of optimizing control follows a bottom-up procedure, where the first step is the key objective in this project: To provide reliable and accurate process information (the desired process variables) by combining new sensor data, already available measurements data and validated data and modelling.

8.3 Partner technologies

8.3.1 Cybernetica CENIT (Cognitive extension)

Cybernetica Cognitive CENIT is a planned extension to Cybernetica CENIT and is described in more detail in *Annex 7. Cybernetica Tool components*. Cybernetica CENIT fulfils the self-adaptation requirement of cognition via state and parameter estimation with process data. The scope of the current Extended Kalman Filter (EKF) and Moving Horizon Estimators (MHE) are, however, limited because: 1) they are unable to distinguish between meaningful and non-meaningful deviations hybrid model predictions and process measurements and 2) they are unable to suggest changes to the structure of the ordinary differential equations (ODEs) that form the mechanistic model. The cognitive extension to Cybernetica CENIT will address these two limitations.

Cybernetica Cognitive CENIT will be able to distinguish prediction deviations due to model inaccuracy from prediction deviations due to errors or issues in process data. Being able to attribute prediction deviations to input or model error is important because the two error classes warrant markedly different responses. In the case of input error, the appropriate response is, depending on the feasibility, some combination of correcting the faulty input signal and minimizing the faulty signal's impact on the model-predictive control. These responses can include: 1) using a default signal instead of the faulty signal, 2) ignoring model state variables that are highly correlated with the faulty signal and 3) altogether suspending estimation for the affected data points. In the case of model error, the appropriate response is to try to adapt the model to most accurately reproduce the process data. This is the procedure that Cybernetica CENIT follows today without confidence that the process data being used for model adaptation is valid. An important goal for Cognitive CENIT will then be to distinguish between input and model error based on offline training of a classification algorithm.

Cybernetica Cognitive CENIT will analyse prediction error distributions in order to suggest structural changes to the model itself. The estimators implemented in Cybernetica CENIT assume that the model structure is correct and that prediction error is normally distributed around a mean value, which the estimator tries to centre at zero. In many cases, this assumption is not true, and significant deviation from normally distributed error may imply error in the model structure, as illustrated in Figure 11. The figure shows (left) the normally-distributed prediction error; (center) the skewed prediction error that may indicate incorrect dependence on a model parameter or input variable; and (right) the prediction error with two peaks that may indicate a missing parameter that is necessary for accurately reproducing both sets of data.
Identifying structural issues in mechanistic models and discovering these issues is well-suited for a technology based on Artificial Intelligence and/or Machine Learning techniques.

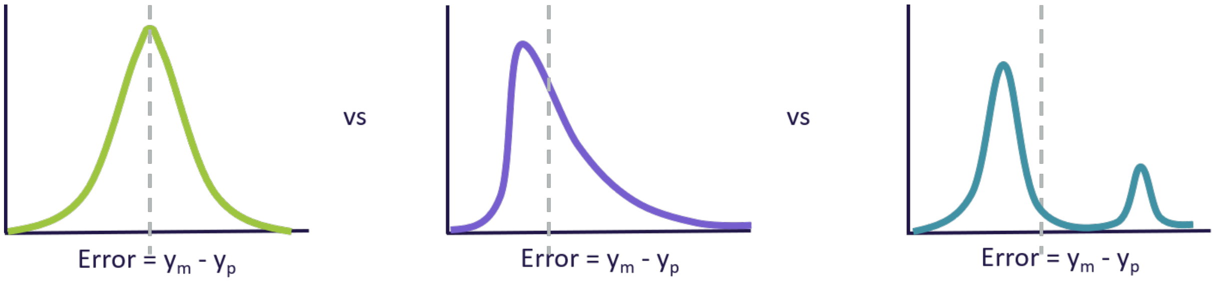


Figure 11: Three examples of model predictive error.

8.4 Pilot requirements related to Cognitive Digital Twins

8.4.1 Hydro case

A cognitive digital twin will use the short-term and long-term predictions of the hybrid model described in section 7.4.1 in order to advise on both the immediate and planned future actions of the fluoride regulation system. Short-term disturbances in fluoride recovery will be handled by proposed changes to the operation of the Gas Treatment Centre (GTC), while long-term disturbances will also pre-emptively suggest that the electrolysis process operators prepare and compensate for the future effects of fluoride losses.

The application developed to regulate fluoride recovery at the Karmøy technology pilot (KTP) will become cognitive by using a customized implementation of the nonlinear model predictive control (NMPC) functionalities in Cybernetica CENIT.

8.4.2 Elkem case

A cognitive digital twin will use the predictions of a hybrid model to advise on both the immediate and planned future actions of the ferrosilicon refining process. The goal is to maximise the utilization of recycled materials while adhering to product specifications by optimising the amount of recycled materials and slag formers added.

The application developed for Elkem Bremanger will become cognitive by using a customized implementation of the nonlinear model predictive control (NMPC) functionalities in Cybernetica CENIT.

8.4.3 Saarstahl case

A cognitive digital twin will use the object tracking system to include temporal coherence (i.e. identify billets in consecutive frames).

8.4.4 Sidenor case

The cognitive twin for the Sidenor case is planned built on the hybrid digital twin. The cognitive twin should be able to tell if the ladle can safely be able to be used one more time, exploiting available information, predictions using the physics-based model and the experiences obtained by the operators (human contributions).

8.4.5 Noksel case

The Noksel case deals with digital twin-based services for smart condition monitoring and predictive maintenance of the SWP machinery. In determining the cases causing failures and defects in the machinery components, experts (operators) knowledge is critical to obtain. In order to provide a cognitive digital twin, expert knowledge will be modelled and embedded into the hybrid digital twin, deep learning algorithms will be used for predictive maintenance.

Noksel-T5.4.1- TEKNOPAR's digital twin will enable self-learning and proactive SWP machinery related to predictive maintenance decisions.

8.4.6 Sumitomo case

The Sumitomo pilot problem examines providing digital twin -based services for monitoring and active management of fouling at the heat exchange surfaces. This requires data-driven models of for fouling monitoring to be developed, potentially including the impacts of direct fouling measurements and estimation of fuel quality to fouling and corrosion. Given this information, guidelines for optimal operation with available fuels are then determined. This poses requirements on tools for process control and prescriptive maintenance to be developed.

8.5 Initial recommendations on AI/Analytics and Cognitive Toolbox integration

The Cybernetica CENIT offer a quite complete set of tools which may at least support two of the pilots. However, several presented partner technologies offer significant additional possibilities. Combining these technologies may prove very efficient.

As the COGNITWIN pilots are, as we write, not fully defined there will be a continuous and mutual development of the pilot UseCases and the partner technologies, and where the partner technologies have to be adapted when necessary. In a majority of the pilots mechanistic models and data/ML will be integrated into hybrid twins. As we know more about all detailed physics-based models available, as well as the model development needs in the different pilots, and the amount and quality of data available, the optimal set of tool have to be picked from the toolbox.

The initial recommendation is to develop Cybernetica CENIT as a base technology, and adapt the other presented partner technologies when necessary to support the pilot cases.

9 Summary of tool box elements and their relations to the pilots

Table 1 The table shown the toolbox elements and their uses or potential uses across the industrial pilots

Pilot type	Pilot	Task	WP5 Toolbox COMPONENTS																					
			DFKI - : generation of photorealistic data	DFKI: Neuroscope	Nissatech: D2Lab-C	SINTEFs BEDROCK	TEKNOPAR Industrial Big Data Analytics (IBVA)	TEKNOPAR Machine Learning Library (TMLL)	SINTEF Open Framework and Tools (SOFT)	Machine learning for hybrid models (SINTEF)	Pragmatic framework for development of hybrid models	Cybernetica CENIT	Cybernetica Cognitive CENIT	Cybernetica ModelFit	Cybernetica RealSim	Cybernetica Viewer	Cybernetica ProXim	Cybernetica OPC UA Server	Bonzai (Scortex)	Como (Scortex)	Keras / tensorflow (Scoretex)	Physics-based model and data for state estimation (OULU)	Associated tools/models for plant modelling and simulation	Finite Markov Chains Matlab toolbox (MCPC) (OULU)
Non-ferrous	Hydro	Plant Digital Twins with ML/AI				(x)								x	(x)	(x)	(x)							
		Multi-variate Sensor analytics,Deep Learning																						
		Deep Learning Performance																						
		Hybrid Digital Twins				(x)			(x)	(x)	(x)	x						x				(x)		
		Cognitive Digital Twins			(x)							x	x											
	Elkem	Plant Digital Twins with ML/AI				(x)								x	(x)	x	(x)		(x)	(x)	(x)			
		Multi-variate Sensor analytics, Deep Learning																	(x)	(x)	(x)			
		Deep Learning Performance																	(x)	(x)	(x)			
		Hybrid Digital Twins				(x)			(x)	(x)	(x)	x						x				(x)		
		Cognitive Digital Twins			(x)							x	x											
Steel	Saarstahl	Plant Digital Twins with ML/AI	x	x															x	x	x			
		Multi-variate Sensor analytics, Deep Learning	x	x															x	x	x			
		Deep Learning Performance	x	x															x	x	x			
		Hybrid Digital Twins	x	x																				
		Cognitive Digital Twins	x	x																				
	Sidenor	Plant Digital Twins with ML/AI				(x)								(x)	(x)	(x)	(x)							
		Multi-variate Sensor analytics, Deep Learning																						
		Deep Learning Performance																						
		Hybrid Digital Twins				(x)			(x)	(x)	x	(x)						(x)				(x)		
		Cognitive Digital Twins			x	(x)						(x)	(x)											
	Noksel	Plant Digital Twins with ML/AI				(x)	x	x																
		Multi-variate Sensor analytics, Deep Learning					x	x																
		Deep Learning Performance																						
		Hybrid Digital Twins				(x)	x	(x)														(x)		
		Cognitive Digital Twins				(x)	(x)	(x)																
Engineering	Sumitomo	Plant Digital Twins with ML/AI												(x)	(x)	(x)	(x)		(x)	(x)	(x)	x	x	
		Multi-variate Sensor analytics, Deep Learning																	(x)	(x)	(x)			
		Deep Learning Performance																	(x)	(x)	(x)			
		Hybrid Digital Twins				(x)	(x)	(x)	(x)	(x)	(x)	(x)						(x)				x		(x)
		Cognitive Digital Twins			(x)	(x)	(x)	(x)				(x)	(x)									x		

In Table 1 we see an overview of the toolbox elements and their relations to the pilots. Toolbox elements which are directly planned used in a pilot is marked with an "x", while identified toolbox elements that currently are not planned used, but has the potential to be used, are marked with "(x)". From the table we can get a good overview over the tools planned to be used in each of the pilots. As the resources for each pilot are limited it is not possible to involve all research partner in each and all of the pilots. Therefore, each pilot team, being a subset of the technology partners, will work with the tools that they know best and which will be the most efficient in order to support a given pilot. The tools with only "(x)" may not be the preferred tool now, due to partner preferences, but may be important contributions in similar future projects.

Looking across the pilots we can see that we have tools related to image processing and modeling (largly the Saarstahl pilot), tools related to machinery optimization and maintenance (largely the Noksel pilot) and tools related to process control and optimization (largely the Hydro, Elkem, Sidenor and Sumitomo pilots). Some of the tools have the potential to be used across this rough classification of the pilots.

9.1 Relations to WP6 "Business impact"

The tools provided by DKFI, Fraunhofer, Nissatech , Oulu and SINTEF are considered as open. This indicates that as much as possible of the tools and methods will be made publicly available and published. However, it is expected that the work in the COGNITWIN project will supply these partners with a competitive edge that may be exploited in future similar projects for the process industries.
The developments of Scortex and Cybernetica will result in proprietary tools which will be protected but made available to the industry at any time. The developments of these tools in COGNITWIN will be a significant contribution to resolve similar pilot challenges in the future.

9.2 Relations to WP7: Communication, Dissemination, Standardisation

The planned work to support the pilots will generate multiple publications to explain the models and methods that have been developed. Such publications are not only important to market COGNITWIN as such, but are critical to advance the use of the developed tools in future projects. The publications will not only market the tools but also the research partners and bring these into new projects and developments.

10 Conclusions

This report has presented the baseline toolbox that will be the starting point for development of the models that will satisfy the needs for the industrial pilots. The available technologies go beyond this report and the totality of tools and methods available is split between this report and the sister COGNITWIN report "D4.1 Baseline Platform, Sensor and Data Interoperability Toolbox". These two reports together give a full picture of the baseline partner technologies in COGNITWIN.
In this report we have pointed out technologies which will be useful to serve all 6 pilots. The partners, together, have several technologies that may serve the industrial pilot cases. As the COGNITWIN project evolves it will become clearer where adaptations of existing technologies, or some new developments, will be needed.

11 References

[1] Ikonen, E., I. Selek, J. Kovacs & M. Neuvonen (2016). Examination of operational optimization at Kemi district heating network. Thermal Science 20 (2), pp 667-678

[2] Ikonen, E. and K. Najim (2001). Non-linear process modelling based on a Wiener approach. Journal of Systems and Control Engineering - Proceedings of the Institution of Mechanical Engineers Part I, 215, pp.15–27.

[3] Ikonen, E. and K. Najim (2002). Advanced Process Identification and Control, Marcel Dekker Inc., New York, U.S.A., 310 p.

[4] Ikonen, E. (1996). Algorithms for process modelling using fuzzy neural networks: A distributed logic processor approach. Acta Universitatis Ouluensis, C95, Oulu, Finland.

[5] Ikonen, E. and K. Najim (2008). On-line optimization of replacement policies using learning automata. International Journal of Systems Science, 39 (3), pp. 237–249.

[6] Najim, K., E. Ikonen and P. Del Moral (2006). Open-loop regulation and tracking control based on a genealogical decision tree. Neural Computing & Applications, 15, no. 3/4, pp. 339-349.

[7] Ikonen, E., J. Kovacs and J. Ritvanen (2013) Circulating fluidized bed hot loop analysis, tuning and state-estimation. International Journal of Innovative Computing, Information and Control, 9 (8) , pp. 3357-3376.

[8] Hultgren, M., E. Ikonen and J. Kovacs (2014) Circulating fluidized bed boiler state estimation with an unscented Kalman filter tool. 2014 IEEE Conference on Control Applications, Antibes, France.

[9] Ikonen, E., I. Selek and K. Najim (2016) Process control using finite Markov chains with iterative clustering. Computers & Chemical Engineering, 93, 293–308.

[10] Ikonen, E. (2017) Active suspension control with state estimation using finite Markov chains. International Journal of Advanced Mechatronic Systems 7 (3) 183–192.

[11] Ikonen, E. and K. Najim (2009). Multiple model based control using finite controlled Markov chains. Cognitive Computation, 1 (3), pp. 234–243.

[12] Ikonen, E, I. Selek and J. Bene (2012). Optimization of pumping schedules using the genealogical decision tree approach. Chemical Product and Process Modeling, 7 (1), 1-25. DOI: 10.1515/1934-2659.1629

[13] Selek, I., J. Bene and E. Ikonen (2013) Utilizing permutational symmetries by dynamic programming – Application for the optimal control of water distribution systems under water demand uncertainties. International Journal of Innovative Computing, Information and Control, 9 (8), pp. 3091-3114.

[14] Ikonen, E. and K. Najim. (1996). Fuzzy neural networks and application to the FBC process. IEE Proceedings - Control Theory and Applications, 143, no 3, pp. 259–269.

[15] Ikonen, E., K. Najim and U. Kortela (2000). Neuro-fuzzy modelling of power plant flue-gas emissions. Engineering Applications of Artificial Intelligence, 13, no. 6, pp. 705–717.

[16] Selek, I (2009) Novel evolutionary methods in engineering optimization – towards robustness and efficiency. Acta Universitatis Ouluensis D1023.

[17] Bene, J., Cs. Hös and I. Selek (2012) Optimal (short-term) pump schedule detection for water distribution systems by neutral evolutionary search. Applied Soft Computing, 12(8), pp. 2336 – 2351.

[18] Hauger et al., Computers & Chemical Engineering, 123: 184-195 (2019).

[19] Zubov et al., Macromolecular Reaction Engineering, 11: 1700014 (2017).
[20] J. Bellatoni and K. Dodge, "A square root formulation of the Kalman-Schmidt filter", AIAA Journal, 5, pp 1309-1314, 1967.
[21] Dan Simon, Optimal State Estimation, pp 400-409, Wiley, 2006.
[22] Rao et al., "Constrained linear state estimation – a moving horizon approach", Automatica, 37, pp 1619-1628, 2001.
[23] Rao et al., "Constrained State Estimation for Non-linear Discrete-Time Systems: Stability and Moving Horizon Approximations", IEEE Transactions on Automatic Control, Vol 48, No 2, pp 246-258, 2003.
[24] Rawlings et al., Model Predictive Control: Theory, Computation, and Design 2nd edition, Nob Hill Publishing, 2017.
[25] Ikonen, E and J. Kovacs (2007). Learning control of fluidized bed combustion processes for power plants. In: S. Kalogirou (Ed.), Artificial Intelligence in Energy and Renewable Energy Systems, Nova Publishers, 12, pp. 395-438.
[26] Hultgren, M., E. Ikonen, J. Kovács, (2014) Oxidant Control and Air-Oxy Switching Concepts for CFB Furnace Operation. Computers & Chemical Engineering, 16, pp 203-219.
[27] Niva, L., E. Ikonen and J. Kovács (2015). Self-optimizing control structure design in oxy-fuel circulating fluidized bed combustion, International Journal of Greenhouse Gas Control, 43, 93-107.
[28] Hultgren, M., E., Ikonen and J. Kovacs (2017) OTU-CFB boiler control design with the dynamic relative gain array and partial relative gain. Industrial & Engineering Chemistry Research, 56 (48), pp. 14290-14303.
[29] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for Simplicity: The All Convolutional Net," in ICLR (workshop track), 2015.
[30] C. Olah, A. Mordvintsev, and L. Schubert, "Feature Visualization," Distill, vol. 2, no. 11, Nov. 2017.
[31] Mahendran and A. Vedaldi, "Visualizing Deep Convolutional Neural Networks Using Natural Pre-images," Int. J. Comput. Vis., vol. 120, no. 3, pp. 233–255, Dec. 2016
[32] Olah, A. Mordvintsev, and L. Schubert, "Feature Visualization," Distill, vol. 2, no. 11, Nov. 2017.
[33] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. abs/1311.2, pp. 818–833, 2014.
[34] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in Proceedings of the IEEE International Conference on Computer Vision, 2017, vol. 2017-Octob, pp. 618–626.
[35] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. abs/1311.2, pp. 818–833, 2014.
[36] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," in Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.
[37] R. Fong and A. Vedaldi, "Interpretable Explanations of Black Boxes by Meaningful Perturbation.," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3429–3437.

[38] Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks,” in 2018 IEEE W. Conf. on App. Computer Vision (WACV), 2018, vol. abs/1710.1, pp. 839–847.

[39] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, “On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation,” PLoS One, vol. 10, no. 7, p. e0130140, Jul. 2015.

[40] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs,” Dec. 2014.

[41] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps,” in ICLR 2014 Workshop, 2013.

[42] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic Attribution for Deep Networks,” in Proceedings of the 34th International Conference on Machine Learning, 2017, vol. 70, pp. 3319–3328.

[43] R. Fong and A. Vedaldi, “Interpretable Explanations of Black Boxes by Meaningful Perturbation.,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3429–3437.

[44] T. Dahmen et al., “Digital reality: a model-based approach to supervised learning from synthetic data,” AI Perspect., vol. 1, no. 1, p. 2, Dec. 2019.

[45] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[46] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[47] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[48] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[49] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).

[50] Chollet, François. "Xception: Deep learning with depthwise separable convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[51] Tan, Mingxing, and Quoc V. Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." arXiv preprint arXiv:1905.11946 (2019).

[52] He, Tong, et al. "Bag of tricks for image classification with convolutional neural networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019.

[53] Lee, Jungkyu, Taeryun Won, and Kiho Hong. "Compounding the Performance Improvements of Assembled Techniques in a Convolutional Neural Network." arXiv preprint arXiv:2001.06268 (2020).

[54] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[55] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).

[56] Lin, Tsung-Yi, et al. "Focal loss for dense object detection." Proceedings of the IEEE international conference on computer vision. 2017.

[57] Yu, Changqian, et al. "Bisenet: Bilateral segmentation network for real-time semantic segmentation." Proceedings of the European conference on computer vision (ECCV). 2018.

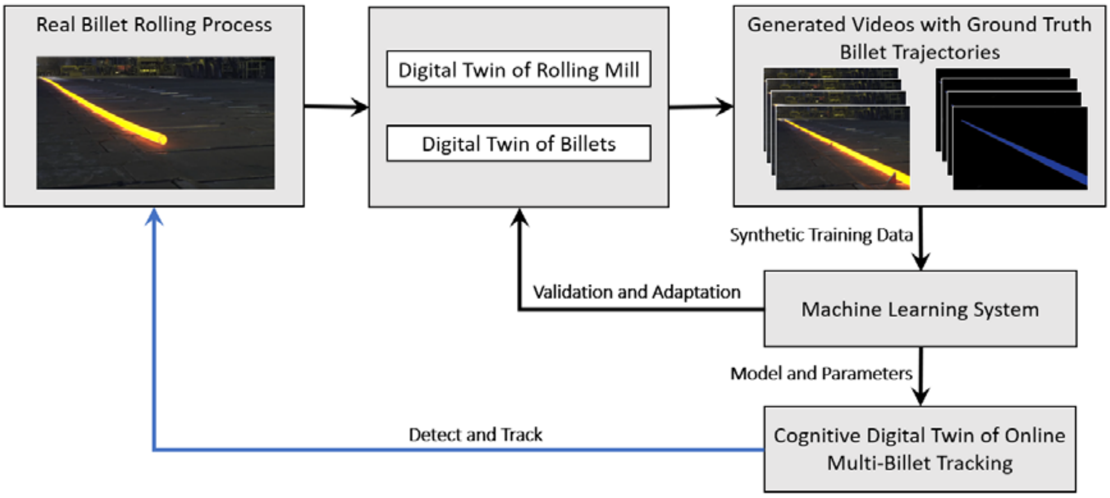
[58] Poudel, Rudra PK, Stephan Liwicki, and Roberto Cipolla. "Fast-scnn: fast semantic segmentation network." arXiv preprint arXiv:1902.04502 (2019).

[59] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531 (2015).
[60] Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." arXiv preprint arXiv:1510.00149 (2015).
[61] Frankle, Jonathan, and Michael Carbin. "The lottery ticket hypothesis: Finding sparse, trainable neural networks." arXiv preprint arXiv:1803.03635 (2018).
[62] Krishnamoorthi, Raghuraman. "Quantizing deep convolutional networks for efficient inference: A whitepaper." arXiv preprint arXiv:1806.08342 (2018).
[63] Hubara, Itay, et al. "Quantized neural networks: Training neural networks with low precision weights and activations." The Journal of Machine Learning Research 18.1 (2017): 6869-6898.
[64] Moons, Bert, et al. "Minimum energy quantized neural networks." 2017 51st Asilomar Conference on Signals, Systems, and Computers. IEEE, 2017.
[65] Miyashita, Daisuke, Edward H. Lee, and Boris Murmann. "Convolutional neural networks using logarithmic data representation." arXiv preprint arXiv:1603.01025 (2016).
[66] Jacob, Benoit, et al. "Quantization and training of neural networks for efficient integer-arithmetic-only inference." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
[67] Hauger et al., Computers & Chemical Engineering, 123: 184-195 (2019).
[68] Zubov et al., Macromolecular Reaction Engineering, 11: 1700014 (2017).
[69] J. Bellatoni and K. Dodge, “A square root formulation of the Kalman-Schmidt filter”, AIAA Journal, 5, pp 1309-1314, 1967.
[70] Dan Simon, Optimal State Estimation, pp 400-409, Wiley, 2006.
[71] Rao et al., “Constrained linear state estimation – a moving horizon approach”, Automatica, 37, pp 1619-1628, 2001.
[72] Rao et al., “Constrained State Estimation for Non-linear Discrete-Time Systems: Stability and Moving Horizon Approximations”, IEEE Transactions on Automatic Control, Vol 48, No 2, pp 246-258, 2003.
[73] Rawlings et al., Model Predictive Control: Theory, Computation, and Design 2nd edition, Nob Hill Publishing, 2017.
[74] Haupin and Kvande. TMS Light Metals, 903-909 (1993).
[75] Yang et al. TMS Light Metals, 641-646 (2014)
[76] S. Moe, A. M. Rustad, and K. G. Hanssen, “Machine Learning in Control Systems: An Overview of the State of the Art,” in Artificial Intelligence XXXV, vol. 11311, M. Bramer and M. Petridis, Eds. Cham: Springer International Publishing, 2018, pp. 250–265.

12 Annex 1. DFKI-Tool-Component

Template for tool/method/component descriptions in WP4 and WP5 baseline

Component/Tool description
Component/Tool/Method/Framework/Service Name
Method: generation of photorealistic data of the billet rolling process at Saarstahl AG
Short Description – incl. Purpose
The problem of Multi-Object-Tracking (MOT) consists in following a trajectory of different objects in a frame sequence. Recently, more and more MOT algorithms have started exploiting the representational power of deep learning. The strength of deep learning networks lies in their ability to learn rich representations as well as to extract complex and abstract features from their input. Researchers have had great success with supervised deep learning on labelled data. However, the availability of training data is the main problem of deep learning methods. In case of multi-object tracking, labelling the training set can constitute a tremendous effort. High-fidelity simulations make it possible to train and test DL algorithms more effectively, leading to more robust and adaptive networks. Models can gain much more experience in the photorealistic virtual world than in the real environment. We can simulate rare events that pose challenging situations, e.g. appearance of abnormalities in industrial processes. We can also generate broadly distributed variations in a data set, enabling the model to better generalize in cases of unseen data.
Function – suitable for which process steps (ICT/Data process) ***Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation***
The method is suitable for generating training data in cases where only a small amount of real-world data is available or annotating available data sets constitutes a tremendous effort.
Examples of usage / illustrations
We want to generate video sequences of the billet rolling process. Later, the videos will be used as a ground truth training data for AI-based multi-billet-tracking system. The sketch of the method:



Images below show examples of simulated billets.

Overall architecture / pipeline / workflow (incl. figure – elements according to BDVA)

A 3D model of the rolling mill will be computed by aerial close-range photogrammetry, in which numerous real-world photos are captured by a drone camera and stitched together. Parametric modelling of billets as well as simulation of the rolling mill's dynamic behaviour with further video rendering will be conducted in 3D modelling software, e.g. Cinema 4D and Maxwell Render.

Interfaces (in/out) – system/user

Subordinates/parts – any platform dependencies

Data (in/out)

Standards (any standards being used)

Licenses, etc. (free for use in the project)

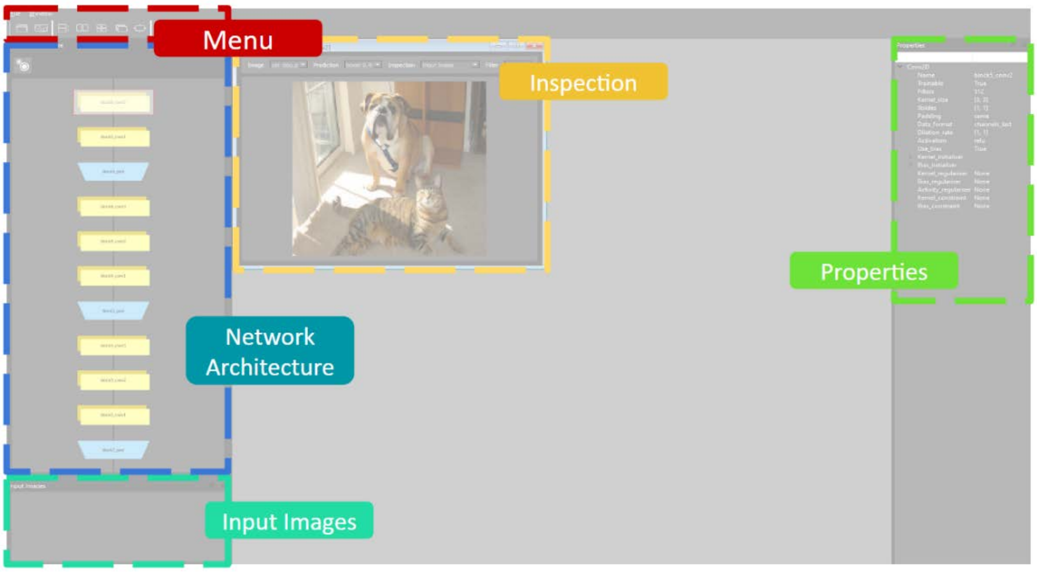
Agisoft Metashape (photogrammetry), Maxon Cinema 4D R21 (3D modelling), Maxwell Render (physically-based render)

TRL for overall component/tool and any parts/subordinates

References – incl. web etc.

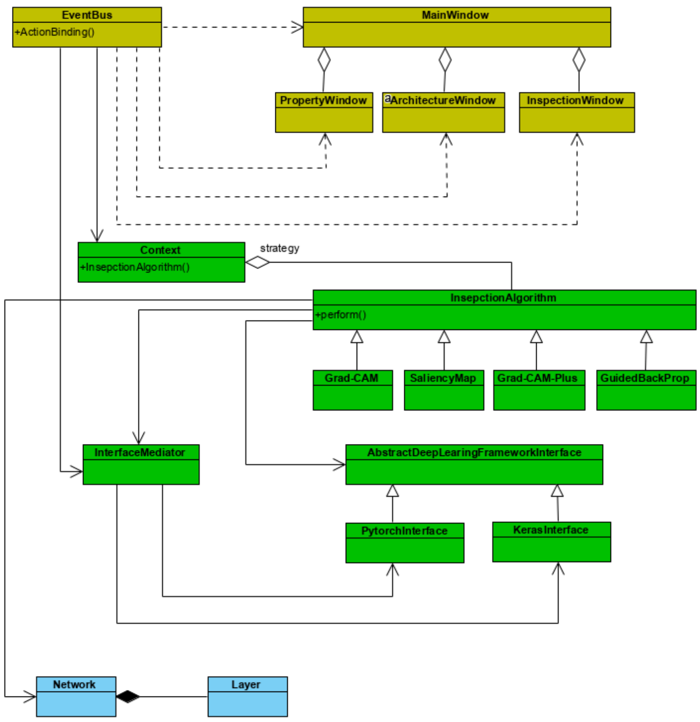
To be considered in particular for the following COGNITWIN pilots

Saarstahl

Component/Tool description
Component/Tool/Method/Framework/Service Name
Component: Neuroscope
Short Description – incl. Purpose
Neuroscope is an interactive software for the visual debugging of artificial neural networks in Deep Learning. While extremely powerful in terms of detection accuracy and generalisation capabilities, the use of Deep Learning networks close to production still suffers from severe shortcomings from an operational perspective. A key shortcoming is a lack of reliable methods to understand the root cause of errors in the case of unwanted behaviour. One approach to this issue is network visualisation. In this approach, one generates visual representations of internal states of the network that allow an expert to identify, understand and solve situations of incorrect network output. Neuroscope implements several such methods including class activation maps, gradient guided class activation maps, and saliency maps in an interactive user interface and with backend connections to the frameworks Tensor Flow and PyTorch.
Function – suitable for which process steps (ICT/Data process) ***Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation***
The software is a versatile software that can be used in any situation where a convolutional neural networks exhibits unwanted or incorrect behaviour. In the specific situation of the COGNITWIN project, Neuroscope will be used to facilitate the tracking of billets in the rolling mill using Deep Learning methods.
Examples of usage / illustrations
In the following, we explain the usage of the software. 

As an interactive tool, Neuroscope allows different processes and usage scenarios. A typical workflow consists of opening a network architecture with pre-trained weights definition (file – open – network architecture) and several input images (file – open – input image). One can then create one or more inspection windows (window – new inspection) and configuring them to compare a correct situation with the incorrect. A typical operation is to open a correctly handled image in one inspection window and an incorrect one on a second window. By interactively selecting different network layers in the network architecture view (left), the correct and incorrect behaviour can be tracked through the network layers.

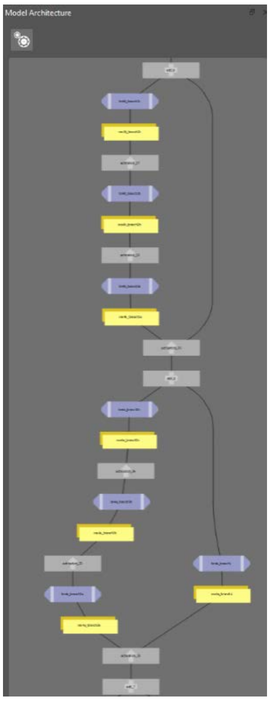
Overall architecture / pipeline / workflow (incl. figure – elements according to BDVA)



The system consists of a Python software developed against the PyQT user interface library. A core element of the architecture is the abstraction of the Deep Learning framework by an interface layer. This approach allows to use the software with either PyTorch or Tensor flow (via Keras) as a backend.

Interfaces (in/out) – system/user

Neuroscope presents the user with an interactive user interface in the sense of a desktop application. Hereby, it is important to keep in mind that the intended user of the Neuroscope application is an machine learning expert who is currently debugging a Deep Learning system. Consequently, the terminology of the user interface follows the established terminology of the Machine Learning field and as such is quite technical.



The example on the left shows the Model Architecture window. This graph-based representation is generated automatically when loading a neural network model. It is noteworthy that this automatic overview works not only for static-network frameworks like Tensor Flow but also for dynamic-network frameworks like PyTorch.

Contrary to other network architecture visualisation packages like tensor board, the graph view is not is not purely for visualisation purposes, but serves as an interactive user interface component. By scrolling and clicking through the architecture view, a user can select different network layers for the inspection and property windows. The usage reminds strongly on the use of a callstack window in a conventional debugger.

Subordinates/parts – any platform dependencies

The system depends on deep learning frameworks as computational backends. It supports Tensor Flow (via Keras) and PyTorch. Because of the use of the Python language, the software is operating system independent.

Data (in/out)

Input of the system are (1) network architecture description files (2) network weights and (3) optional input images for some of the visualisations.

Standards (any standards being used)

Network IO operations are delegated to the Deep Learning backend, so the system supports any file format supported by Tensor Flow or PyTorch, respectively.

Licenses, etc. (free for use in the project)

TRL for overall component/tool and any parts/subordinates

The system is currently in TRL 3.

References – incl. web etc.

To be considered in particular for the following COGNITWIN pilots

Saarstahl

13 Annex 2. SINTEFs BEDROCK

Component/Tool description
Component/Tool/Method/Framework/Service Name
BEDROCK Software framework and underlying component for several SINTEF activities on advanced process control, digital twin, pilot plant operation and research data management for process plants.
Short Description – incl. Purpose

The BEDROCK framework is a flexible, lightweight easily deployable software bundle of modules developed at SINTEF Industry applied as the foundation for digital twinning R&D activities and process control. The purpose of the project to enable a framework for building various process plant applications. High-level architecture is shown in the figure below:

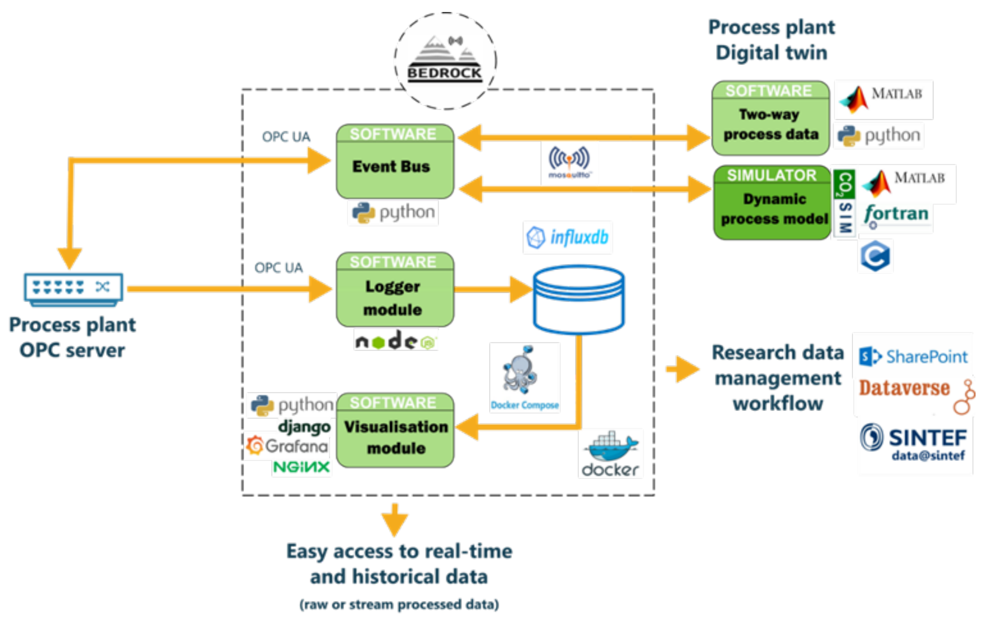


Figure 1: High-level software architecture of the SINTEF BEDROCK framework.

The software enables bi-directional interfacing between process plant OPC server and dynamic process models in order to perform advanced process control based on the OPC UA and MQTT protocols. The framework includes additional modules for two-way interaction with process parameters by Matlab or Python. The framework is currently deployed in production at SINTEF Industry's pilot facility at Tiller, Trondheim, where it enables easy access to real-time and historical process data and stream processing for researchers and customers. BEDROCK is also central part of the workflow for research data management of pilot test-campaign process data. Similar applications of the framework on SINTEF laboratory test rigs is under planning. As the framework name implies, the software bundle is an important foundation for a range of digitalization activities at the Department of Process Technology at SINTEF Industry. Key to the development of the framework, has been

SINTEF's easy access to an in-house industrial grade process plant; The Tiller pilot facility is used as a testing ground for application and development of BEDROCK.

Function – suitable for which process steps (ICT/Data process)
Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation

Data collection
Collection of process data from a process plant OPC server is an important functionality of the BEDROCK framework. It enables separate database storage of separate customer pilot plant test campaigns allowing for isolation of confidential datasets. (The OPC server historian enables access to previous test campaigns independent of ownership to the data.)

Core component of the framework is a logger module written in NodeJS, which is well suited for asynchronous applications. Utility functions enables subscriptions to process tag lists provided during configuration of the logger. All data is logged to an InfluxDB database. Influx has a database engine purpose-built for handling large streaming time series datasets. Both the logger module and the InfluxDB database server runs dockerized in the BEDROCK docker-compose bundle. The location of the logging database can either be persistent storage on the BEDROCK host machine, or on remote location.

Automated procedures for research data management and secure storage of process datasets have been developed in another SINTEF framework named BUNKER. Framework interaction is illustrated in Figure 2. This workflow ensures continuous backup of the BEDROCK database with data transfer from the process plant server Linux domain to the Microsoft SharePoint environment. Data collection workflow applies SharePoint Workspace (connected to the ongoing project) as staging area of datasets prior to permanent transfer and storage of datasets with metadata to a higher-level data management archive. The current framework has an API for transfer of BEDROCK data to the REST-API of the Dataverse research data repository software. This open API enables compatibility with SINTEF's coming research data management framework "data@sintef" currently in the acquisition phase at SINTEF.

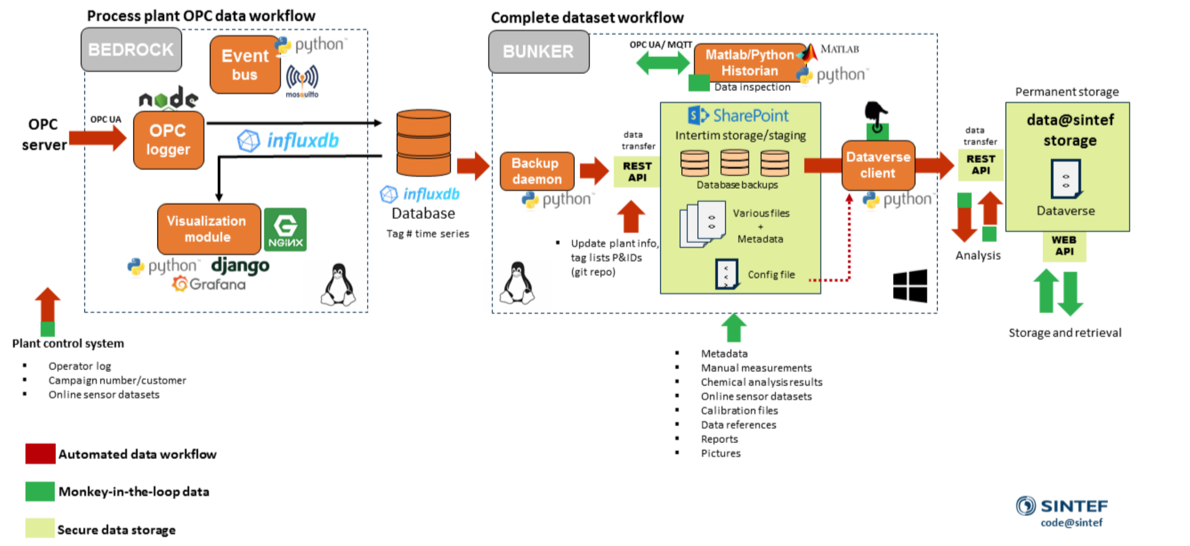


Figure 2: Research data management workflow for pilot plant test campaign datasets at SINTEF.

Visualisation

The framework enables easy remote access to real-time or historical process parameters for SINTEF researchers and customers during and after process plant operation. The visualisation module allows for SINTEF access control and flexible dashboard layouts. The user interface can be accessed remotely over web. Example screenshot from the web app is depicted in Figure 3 below.

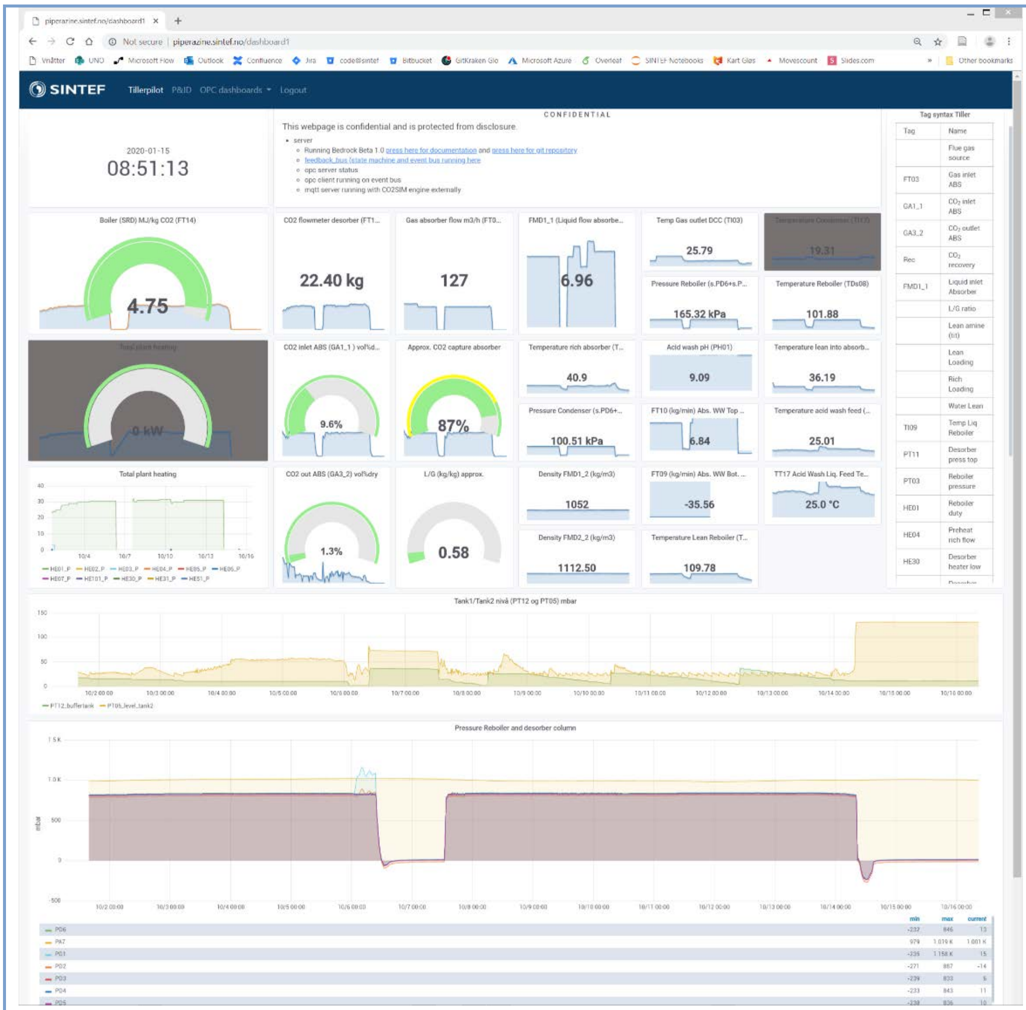


Figure 3: Screenshot from the BEDROCK visualization module web application.

The web application is built on Python Django with embedded flexible dashboards from a Grafana server running dockerized in the framework. As for the rest of the BEDROCK framework, the visualization module is easy deployable on any machine.

Data processing

Compute modules can be added for stream processing of real-time data to the visualisation module. This is currently integrated for some KPIs and calculated process parameters at the Tiller facility. The event bus module enables bi-directional linking of OPC server and back-end compute modules for data analytics, stream processing or model-predicted control loops.

Automated data sanitation and post-processing of process data enables testing of different data processing strategies and comparing the resulting calculations and estimates. Auto-generation of reports for process plant operation (daily, weekly etc.) is a time saving feature for SINTEF operators/researchers.

Sharing and access

BEDROCK is deployed in production on a server at SINTEFs Tiller pilot plant facility connected to the in-house network. Modules deployed on SINTEFs network can only be accessed by SINTEF users. Due to the full dockerization of the framework's software bundle, full-stack or various components of BEDROCK can easily be deployed on any machine in approximately 5 minutes time. Deployment of customized BEDROCK bundle has been applied on Azure cloud for sharing of online and historical process data with clients outside of SINTEF as shown in Figure 3. Encrypted Process data (TLS) is pushed to the cloud via a data diode for safeguarding of SINTEFs sensitive network.

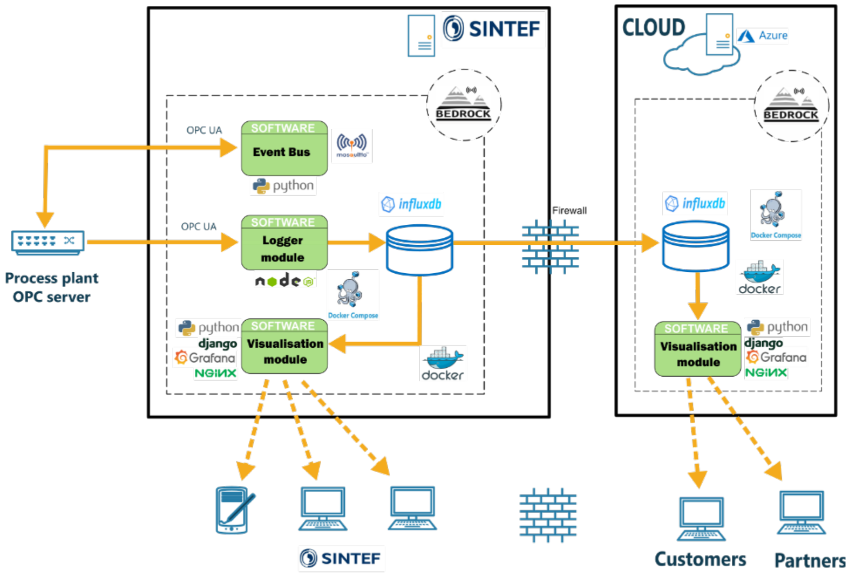


Figure 3: Cloud deployment of BEDROCK modules for sharing and external access.

For such deployments, SINTEF access control to the application is ensured by Azure Active Directory (SINTEF AD) OAuth 2.0 for authorization according to SINTEF standards.

The BEDROCK Event Bus can function as a notification module in the framework for subscribing and publishing notifications or reports to users and operators over email or SMS.

Analytics

Modules for real time interaction with process data is available both for Matlab and Python. The current Matlab module communicates over MQTT with the BEDROCK event bus module enabling two-way communication of tags (process measurements and set-points). The current version of the Python module is read-only for process tags and set points. Easy access to real-time and historical process data directly into Matlab or Python ensures good options of data analytics.

Decision support

Demonstration of process decision support has been performed with application of SINTEF's in-house dynamic process simulator CO2sim Dynamics. Digital twinning with the CO2 capture pilot at Tiller, Trondheim has shown good results with the simulator replicating the physical process plant P&ID and tag-list. As the lowest level of decision support, the digital twin can operate isolated, but in parallel with the physical plant, enabling mirrored dashboards in the user interface. Testing of new setpoints can be performed on the simulator to obtain estimates on resulting process operation prior to changing the setpoint of the real plant.

Script based process control

Activities on script-based process control of the Tiller pilot plant is under planning. This will allow for time-saving and better reproducibility of results by reducing the need for human operators for currently manually performed procedures like plant start-up, shut-down and time-optimized targeting of process steady-state setpoints. Script-based control is also a bi-directional interaction between the process plant OPC server and the scripting module, including a feed-back loop for monitoring of setpoint and manipulated variables.

Advanced process control

The framework enables bi-directional coupling of SINTEF in-house dynamic process models and a real process plant. A simulator replicating the physical plant process flow diagram and essential tag list runs in parallel with the plant. Running the simulator faster than real-time enables insight into predictions of future operation, which is applied for process control in a continuous feed-back loop between simulator and process plant. The BEDROCK framework enables the setup and testing of such digital twins. The flexibility of the system also allows for replacing the physical plant with a simulator – running digital twin control strategy testing with two simulators and no physical plant in the loop.

Examples of usage

Examples of BEDROCK application *user stories* below:

Advanced process control

The framework is applied for testing of digital twins in post-combustion CO_2 capture plants, aiming at achieving CO_2 long-term capture targets at a minimized cost. A dynamic simulator allows for pilot plant operation with a model predicted control loop, optimizing operation based on external input parameters such as electricity price and utility cost based on continuous predictions into future plant operation.

Process design

Important inputs to the design of a new industrial process plant is achieved by testing of dynamic process scenarios applying the BEDROCK framework for coupling two simulators for testing of control strategies. Modification of the process flow diagram is tested on a simulator, saving time and money on costly physical pilot plant activities.

Easy access to process data

The BEDROCK framework enables close participation in pilot test campaign operation without participants being present at the Tiller site. Real-time process data can be analysed in the office as they are produced. Researcher hours are saved on the production of auto-generated daily reports to the customer. The customer on the pilot test project experiences that SINTEF provides access to important result, long before the final project report is written. Despite being in the US, the customer can follow the testing at Tiller in real-time over the web application.

Script-based control

Time is saved during pilot testing by script-based process control. The operators order automatic start-up of the plant, enabling the pilot plant to be at steady state at 08:00 when the researchers arrive for manual measurements. campaign is completed. More steady state data points are collected during a week due to faster tuning in of the plant to the desired setpoints.

Monitoring of laboratory test rigs

An existing laboratory test rig running National Instruments LabView is retrofitted with the LabView software OPC module. By deploying the BEDROCK framework on the computer in the lab, the researcher can monitor the results produced in the lab from the office. Two-way control enables performing remote adjustments to the ongoing test. Automated intelligent experimental test planning, allows for automated assessments of a produced result giving inputs to optimal settings in the next test run (which starts automatically).

Research data management

The BEDROCK database structure ensures no cross-contamination between different confidential industry projects performed in the same pilot plant. Researchers working in each project only have access to process data from the project database, and not from the complete OPC historian.

The research data management workflow implemented allows for cross linking and precomputation of process data and chemical analysis data present in the SINTEF central data repository after the test campaign is completed.

Overall architecture / pipeline / workflow
The current overall framework software architecture is based on a Docker container bundle orchestrated by Docker Compose. The core modules are Docker containers of NodeJS OPC UA logger, Python Event Bus, InfluxDB server, Python Django web application and Grafana server. All components interplay in a Docker network – BEDROCK is configured and ready for deployment in 5 minutes after cloning the repository on a computer running Docker. Minimum configuration inputs are OPC UA server url and the process tag list of interest. Separate deployments also need a running MQTT broker available for application of two-way control.
Interfaces (in/out) – system/user

- Web application - with flexible dashboards as part of a customizable website framework.
- MQTT – access to subscriptions and publication to Event Bus topics.
- REST-API
  - InfluxDB access
  - Dataverse or higher RDM system (a high-level Windows app with GUI is built as part of the framework)
- OPC UA – direct client or server interface available in Python
- Matlab interface – two-way real-time or historical process data from OPC server to Matlab
- Python terminal / notebooks – one-way monitoring of real-time or historical data
- Linux command line – BEDROCK docker bundle configuration and CLI interface of modules
- SharePoint – access to data pushed from BEDROCK

Subordinates/parts – any platform dependencies

Core modules

- Python
- NodeJS
- InfluxDB
- Grafana
- Moquitto (or similar MQTT broker)

Utility modules

- Python
- Matlab

Data (in/out)

In:

- Process plant OPC UA client tag structures
- MQTT messages over Event Bus topics
- Metadata describing the datasets
- BEDROCK configuration parameters (in condfg files)
- Tag list of interest to be implemented in the OPC workflow

Out:

- Process plant OPC UA server tag structures
- Time series tag structures in database
- MQTT messages over Event Bus topics
- Processed data from tag time series raw data
- Dataset metadata

Standards

OPC UA protocol is applied for bi-directional communication between OPC server and the BEDROCK Logger module and Event bus. The Python OPC-UA / IEC 62541 Client and server library enables both client and server function of the modules. The API offers an interface to send and receive all UA defined structures and high-level classes. MQTT is applied for bi-directional communication with the BEDROCK Event Bus module. Influxdb, a purpose-built time series database engine is applied as database infrastructure. BEDROCK visualization module can also process data from other database sources, such as MySQL, PostgreSQL, Google Stackdriver and Azure Monitor.
Licenses
The BEDROCK framework is based on open-source software. Application of sub-modules involving Matlab are the only current licence requirements (Python-based versions of similar modules have been developed). The current version of the framework is intended as a generic tool for application in SINTEF R&D projects for digitalization in the process industry.
TRL for overall component/tool and any parts/subordinates
To be discussed.
References – incl. web etc.
The BEDROCK source code repository lives under git version control in SINTEF private cloud code@sintef as part of the DigtalTwin project (DIG) [https://stash.code.sintef.no/projects/DIG]. The BEDROCK repository contains the latest production branch deployed at the Tiller facility in addition to various development branches. The framework aims at further refinement and development as new projects comes in. Please contact Aslak Einbu (aslak.einbu@sintef.no) for further questions or discussion regarding collaboration or contribution.
To be considered in particular for the following COGNITWIN pilots
The BEDROCK development team at SINTEF Industry is looking for partners inside/outside SINTEF for application and further joint development of the software.

14 Annex 3. TEKNOPAR Tool Components

Component/Tool description
Component/Tool/Method/Framework/Service Name
COMPONENT: STEEL 4.0 TEKNOPAR Industrial Big Data Analytics (IBDA)
Short Description – incl. Purpose
Being one of the components of STEEL4.0, IBDA aims to generate meaningful information to support digital twin for state estimation and process control. By linking the real sensors data, provided by IIoTP, to the SPW plant model, IDBA searches the effective list of parameters needed to simulate SWP. Information generated by IDBA is used for NOKSEL's maintenance and management processes. IDBA supports plant's digital twin by processing sensor data retrieved by IIoTP, and by proving the processed data to TMML as input. The big data analytics performed by IDBA includes time series, sensor, IoT stream data analytics. Feature extractions are performed by IDBA. Common to the other components of STEEL 4.0, IBDA is easily adaptive to changes in needs. It is easy to use on the cloud and in built-in servers. It is scalable against changes in size and speed. IDBA performs real-time data processing with low latency on the scale of seconds.
Function – suitable for which process steps (ICT/Data process) ***Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation***
Analytics IDBA conducts descriptive, diagnostic, predictive and hybrid analytics on the big data provided by IIoTP of STEEL4.0. Big data analytics conducted by IDBA is used as input for TMML.
Examples of usage
Based on multivariate sensor's data collected, IBDA conducts analytics for feature extractions needed for predictive maintenance of the SWP machinery. TMML uses the features extracted by IBDA.
Overall architecture / pipeline / workflow (incl. figure – elements according to BDVA)
The main flow of system is presented by the below System Sequence Diagram of STEEL4.0.

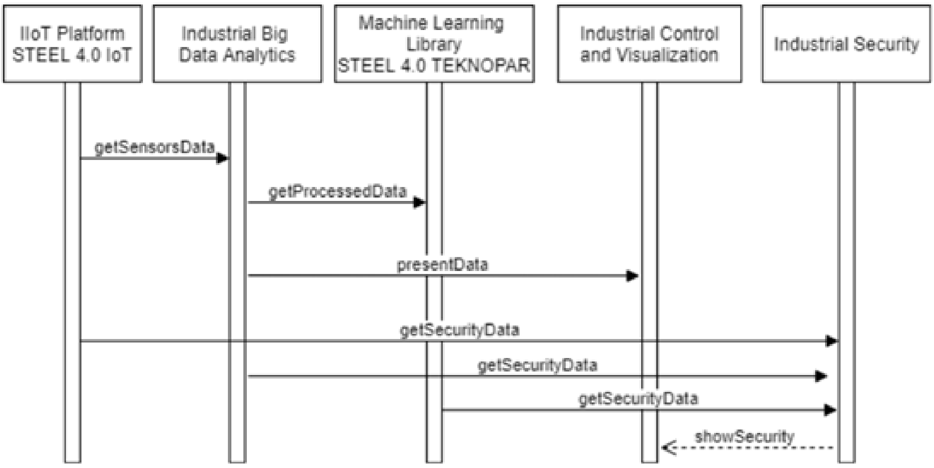
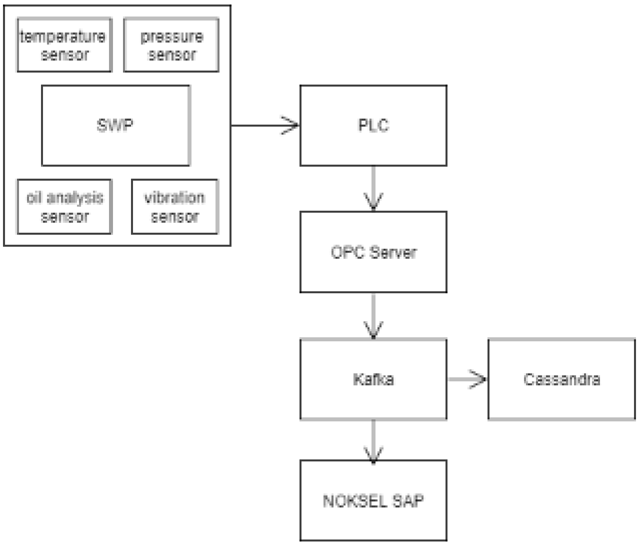


Figure 2. System Sequence Diagram of STEEL 4.0



Interfaces (in/out) – system/user
IN: IIoTP Platform STEEL 4.0 IoT OUT: TMML (TEKNOPAR's Machine Learning Library), Industrial Security (IDS), Industrial Control Panel and Visualisation (ICPV)
Subordinates/parts – any platform dependencies
IIoTP, TMML and IDS
Data (in/out)
IN: Raw sensor Data from IIoTP OUT: Processed meaningful data and features extracted to TMML and security data to IDS
Standards

JSON, OPC UA, TCP/IP
Licenses
Apache License 2.0 (for Kafka, Cassandra, Flink, NoSQL) [1, 2] Oracle Technology Network License (for Java) [3] Pyhton [4], Matlab[5]
TRL for overall component/tool and any parts/subordinates
6
References – incl. web etc.
1.http://www.apache.org/licenses/, 30 Nov 2019 2. http://www.apache.org/licenses/LICENSE-2.0, 30 Nov 2019 3. https://www.oracle.com/downloads/licenses/standard-license.html, 3 Dec 2019 4. https://www.python.org/, 14 Feb 2020 5. https://www.mathworks.com/products/matlab.html, 14 Feb 2020
To be considered in particular for the following COGNITWIN pilots
Pilot 6: Noksel – (COGNITIVE) DIGITAL TWIN POWERED CONDITION MONITORING (and Control) IN STEEL PIPE MANUFACTURING INDUSTRY

Component/Tool description
Component/Tool/Method/Framework/Service Name
COMPONENT: STEEL 4.0 TEKNOPAR Machine Learning Library (TMML)
Short Description – incl. Purpose
Being one of the components of STEEL4.0, TMML aims to build a cognitive digital twin for the production processes of the SWP machinery. TMML includes effective and distributed machine learning algorithms for industrial data analysis and predictive maintenance to be used for SWP. The ML algorithms developed within and provided by TMML enables SWP digital twin for state estimation and process control.

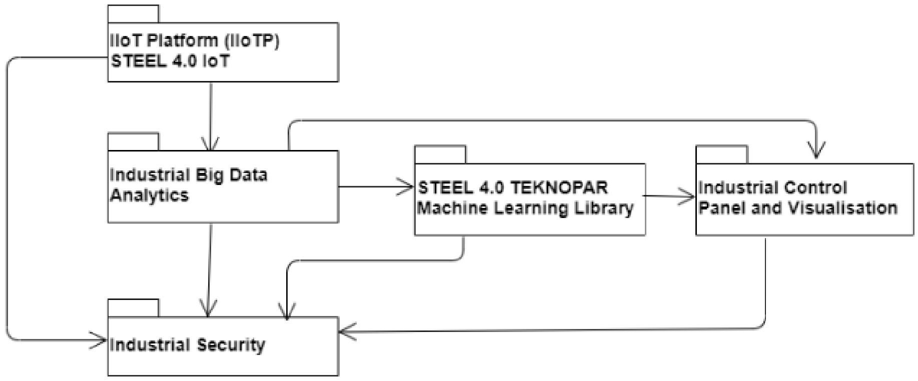


Figure 2. STEEL4.0 UML Package Diagram

TMML aims to have high performance and low delay, and performs real-time data processing. TMML has a modular design, and enables Flink compatible easy integration. TMML, which is easily adapted to changing needs, enables data scalability for changes in size and changes in speed.

Function – suitable for which process steps (ICT/Data process)
Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation

Processing
Multivariate sensor data collected by IIoTP and pre-processed by Industrial Big Data Analytics (IBDA) is used in TMML for predictive maintenance and anomaly detection.

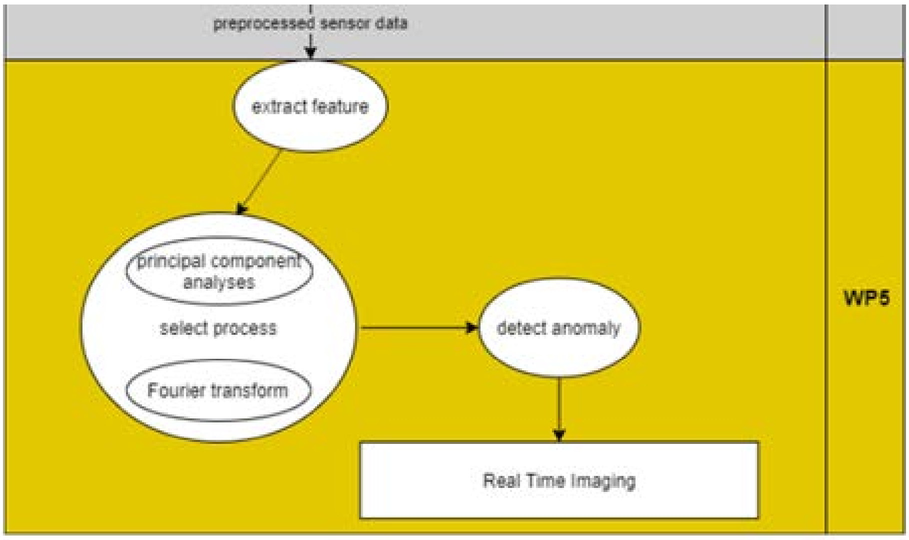


Figure 2. STEEL4.0 WP5 Related Data Flow Diagram

Decision Support
ML/DL algorithms will support decision regarding predictive maintenance of the SWP machinery at the pilot's plant.

Visualisation

TMML provides input on predictive maintenance of SWP machinery to ICPV component.

Examples of usage

IDBA conducts big data analytics on multivariate sensor data collected by IIoTP, and the output provided by IDBA is used by TMML. TMML provides and enables the application of machine learning algorithms and deep learning needed to perform smart predictive maintenance for the SWP machinery. Thus, TMML helps to build a cognitive digital twin. TMML utilizes supervised, unsupervised, multidimensional scaling and reinforcement learning algoırithms as needed.

The output generated by TMML is sent to ICPV for real time visualisation, as well as to the maintenance and management system of NOKSEL.

Overall architecture / pipeline / workflow (incl. figure – elements according to BDVA)

TMML component is related to the data analysis and processing functions of STEEL 4.0.

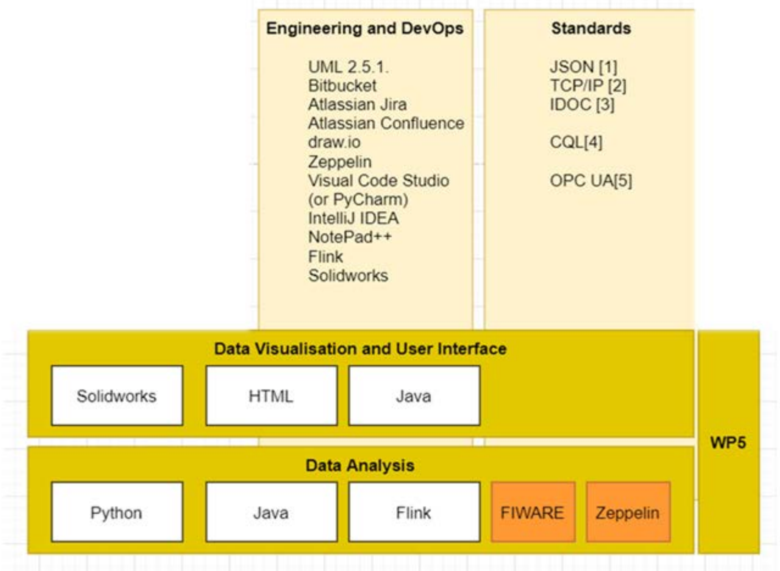


Figure 2. BDVA RM for STTEL4.0 to be used for WP5

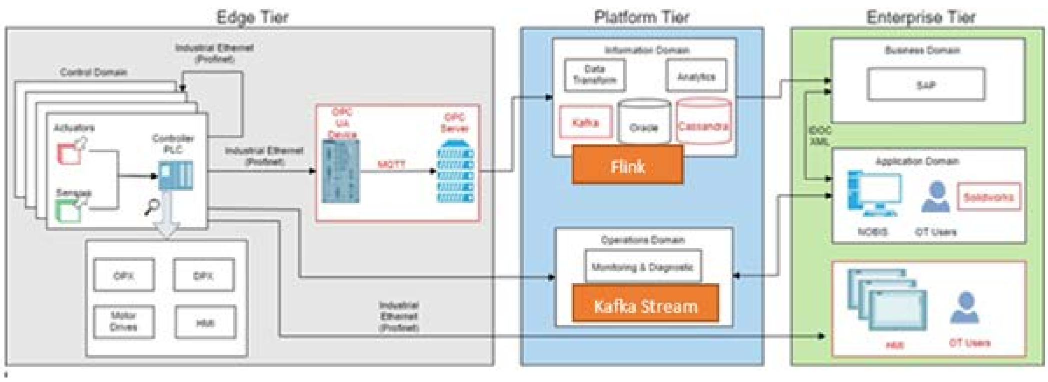


Figure 2. Future Architecture at NOKSEL

Interfaces (in/out) – system/user
IN: Industrial Big Data Analytics (IDBA) OUT: Industrial Data Security (IDS), Industrial Control Panel and Visualisation (ICPV)
Subordinates/parts – any platform dependencies
Data analysis: Matlab, Pyton, Java, Flink, FIWARE, Zeppelin, Kafka, Cassandra
Data (in/out)
IN: Preprocessed sensor data, originally collected from multi-variate sensors, from IDBA component OUT: Security data required by IDS component, and Maintenance and Defect Anomaly data related to predictive maintenance for ICPV and NOKSEL's existing system
Standards
JSON, TCP/IP, IDOC, CQL, OPC UA
Licenses
Apache License 2.0 (for Flink, NoSQL, Kafka, Cassandra) **Oracle Technology Network License (for Java) [3]** **Pyhton [4], Matlab[5]**
TRL for overall component/tool and any parts/subordinates
6
References – incl. web etc.
1. http://www.apache.org/licenses/ , 30 Nov 2019 2. http://www.apache.org/licenses/LICENSE-2.0, 30 Nov 2019 3. https://www.oracle.com/downloads/licenses/standard-license.html, 3 Dec 2019 4. https://www.python.org/, 14 Feb 2020 5. https://www.mathworks.com/products/matlab.html, 14 Feb 2020
To be considered in particular for the following COGNITWIN pilots
Pilot 6: Noksel – (COGNITIVE) DIGITAL TWIN POWERED CONDITION MONITORING (and Control) IN STEEL PIPE MANUFACTURING INDUSTRY

Component/Tool description
Component/Tool/Method/Framework/Service Name
COMPONENT: STEEL 4.0 Industrial Control Panel and Visualisation (ICPV)
Short Description – incl. Purpose
Being a component of STEEL4.0, ICPV aims to visualise historical data, trend graphs, status monitoring, status reports, maintenance modules. ICPV enables users to generate custom

reports, tables, and graphs. ICPV displays various graphics and tables in user applications in real time via Fiware. ICPV's real-time status display has maximum delay of 100miliseconds. It has a modular structure suitable for addition of new visualisation tools and capabilities. ICPV shall retrieve data from Cassandra and Flink to user applications via JSware in JSON format.
Function – suitable for which process steps (ICT/Data process) ***Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation***
Visualisation ICPV supports the cognitive digital twin by means of visual components generated by data retrieved from other components of STEEL4.0. **Decision Support** Decisions are supported via visualisation of data displayed and reports generated.
Examples of usage
Real time and calculated information for condition monitoring and predictive maintenance of SWP are visualised by ICPV.
Overall architecture / pipeline / workflow (incl. figure – elements according to BDVA)
Interfaces (in/out) – system/user
IN: Data from IBDA and TMML OUT: Security data to IDS
Subordinates/parts – any platform dependencies
Jsware, JSON, HTML, Java, Solidworks
Data (in/out)
ICPV retrieves user and facility information data from the relational database to user applications via JSware in JSON format.
Standards
JSON[1], JSware[2]
Licenses
NA
TRL for overall component/tool and any parts/subordinates
6
References – incl. web etc.
1-https://www.json.org/json-en.html, Feb 15 2020

2-https://www.jsware.net/, Feb 15 2020
To be considered in particular for the following COGNITWIN pilots
Pilot 6: Noksel – (COGNITIVE) DIGITAL TWIN POWERED CONDITION MONITORING (and Control) IN STEEL PIPE MANUFACTURING INDUSTRY

15 Annex 4. SINTEF Open Framework and Tools (SOFT)

Component/Tool description

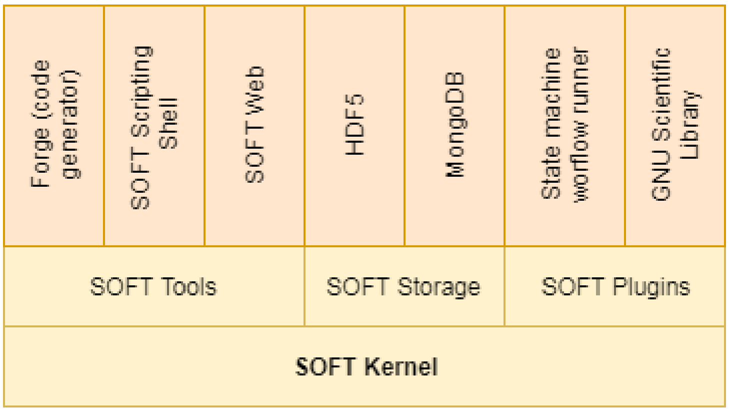
Component/Tool/Method/Framework/Service Name

Name SINTEF open frameworks and tools (SOFT)

Short Description – incl. Purpose

SOFT is an acronym for SINTEF Open Framework and Tools. SOFT5 is a set of libraries and tools to support scientific software development.

The development of SOFT5 was motivated by many years of experience with developing scientific software, where it was observed that a lot of efforts went into developing parts that had little to do with the domain. A significant part of the development process was spent on different software engineering tasks, such as code design, the handling of I/O, correct memory handling of the program state and writing import and export filters in order to use data from different sources. In addition comes the code maintenance with support of legacy formats and the introduction of new features and changes to internal data state in the scientific software. With SOFT5 it is possible to utilize reusable software components that handle all this, or develop new reusable software components that can be used by others in the same framework.



SOFT5 contains a core library with plugin support. The library also comes with set of interfaces (API) to create extensions and custom plugins. The core library is used to connect a software application with the framework.

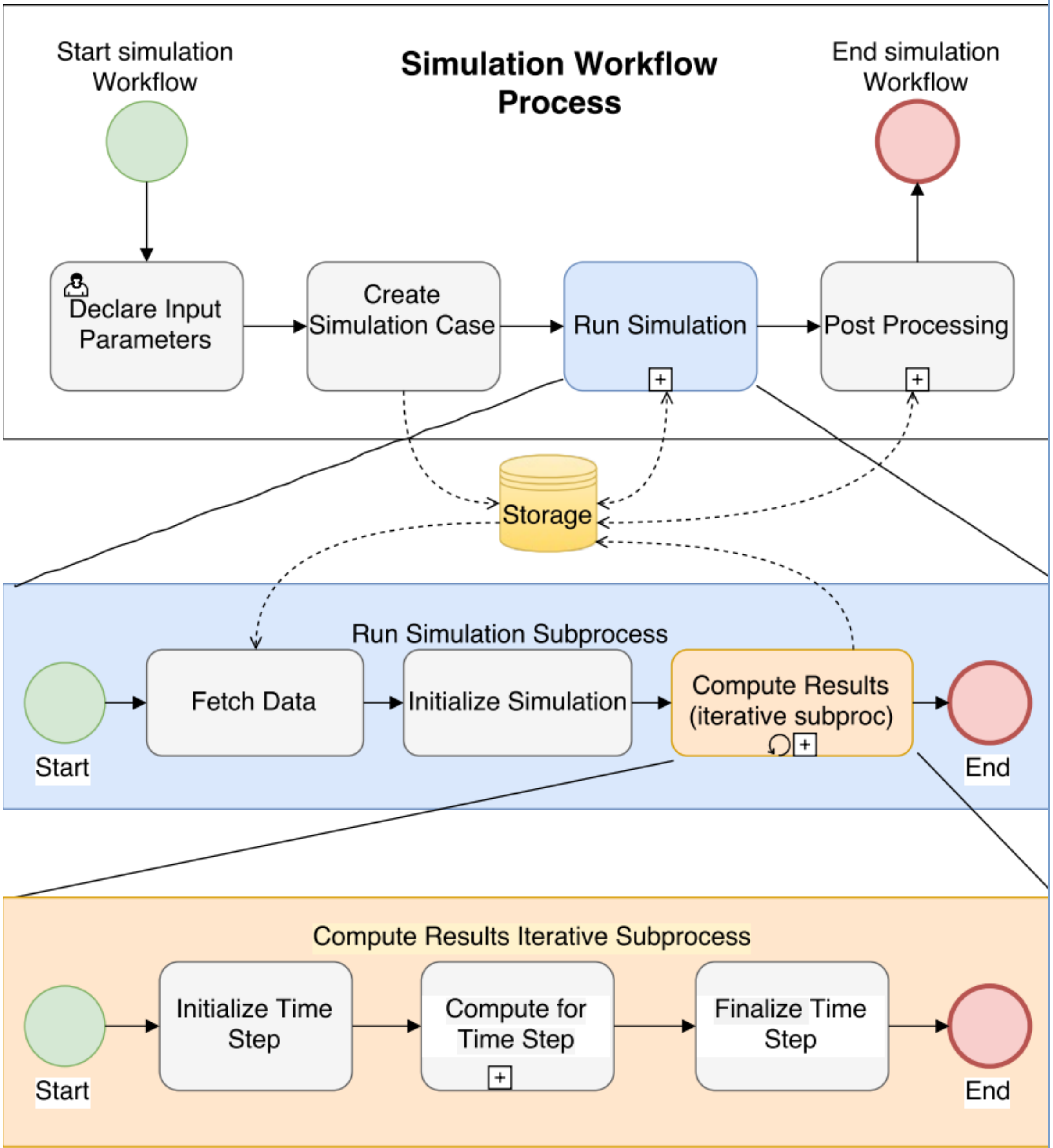
There are currently two supported storage options for storing with SOFT5, namely HDF5 and MongoDB. Local data stored in HDF5 files is suitable for managing local data

The main approach to developing software with SOFT5 is to incrementally describe the domain of the software using entities (see below). The entities can represent different elements of the software, and be used in handling I/O as well as in code generation and documentation. Entities can also be used for annotating data and data sets. This might be useful in cases where for instance the origin of the data, license and ownership are of importance.

Since any complex software will have many entities and often multiple instances of the same entity, SOFT5 allows for creating collections of entities with defined relationships. These entity collections are called 'collections' (see below).

One idea of SOFT5 is that software may be written is such way that business logic is handled by the codebase, while I/O, file-formats, version handling, data import/export and

interoperability can be handled by reusable components in the SOFT5-framework, thus reducing risk and development time.
Function – suitable for which process steps (ICT/Data process) ***Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation***
Anywhere where there is data-exchange
Examples of usage / illustrations
An example of a simulation workflow is shown in the figure below. All communication between each part and the storage is taken care of by SOFT. The proposed workflow is created in the SOFT workflow runner used to orchestrate the simulation.



Overall architecture / pipeline / workflow (incl. figure – elements according to BDVA)

This software is an interoperability layer, and can be adopted to be used for coupling different softwares in a given workflow. It can also be used to build and run workflows.
Interfaces (in/out) – system/user
C, C++APIs. By the use of built in code generation, Domain specific APIs are generated for C/C++, Fortran or Python. The domain specific APIs are most relevant, and expansion to additional languages is possible.
Subordinates/parts – any platform dependencies
C/C++ compiler CMake Boost Qt5 Hdf5 library MongoDB There are two implementations; SOFT5 is currently not Windows compatible, while DLite is platform independent
Data (in/out)
Any, a meta data schema must be created for the data involved
Standards (any standards being used)
HDF5 JSON
Licenses, etc. (free for use in the project)
LGPL
TRL for overall component/tool and any parts/subordinates
6
References – incl. web etc.
https://github.com/SINTEF/dlite https://github.com/NanoSim/SOFT5 Hagelien,Thomas F., Chesnokov, Andrey, Johansen, Stein Tore, Ernst A. Meese, and Løvfall, Bjørn Tore, “SOFT: A FRAMEWORK FOR SEMANTIC INTEROPERABILITY OF SCIENTIFIC SOFTWARE,” in SINTEF PROCEEDINGS: Progress in Applied CFD – CFD2017, vol. 2, Trondheim: SINTEF Academic Press, 2017, pp. 273–278. URL: https://www.sintefbok.no/book/download/1119
To be considered in particular for the following COGNITWIN pilots
Sumitomo, Sidenor, Hydro, Elkem

16 Annex 5. Machine learning for hybrid models (SINTEF)

Component/Tool description

Component/Tool/Method/Framework/Service Name

Method: Integration of physics-based model and data-based model, for hybrid models

Short Description – incl. Purpose

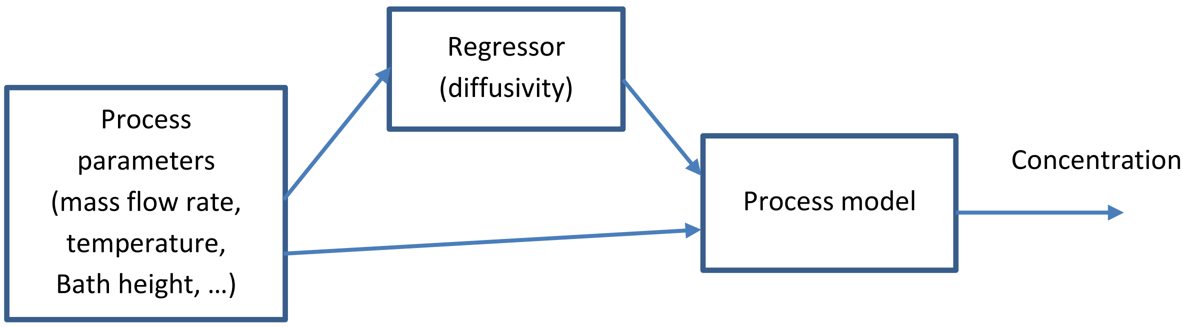
Pure data-based models require an enormous amount of measured data to be trained. The data is often not available in industrial processes, due to the complexity of the measurement and/or to the cost of installing sensors and collecting data. The use of physics-based models within the data-based model framework reduces the dimensionality of the training problem. This makes an optimal use of the existing measured data, since it does not consume data to fit phenomena that can be modelled.
The method described here is about integrating any differentiable phenomenon or process model in a data-based regression algorithm.

Function – suitable for which process steps (ICT/Data process)
Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation

The method is suitable where machine learning techniques are used. It can take advantage of the well-known models for some parts of the process, while relying on data-based models for other parts of the process for which modelling is challenging.

Examples of usage / illustrations

As an example, we want a model for the concentration of a dissolved specie in a liquid after a gas is bubbled through it. Many physical parameters will influence the concentration, amongst which the bubble size, residence time of bubbles or the species diffusivity. In turn, process parameters will affect the value, for example the bubble flow rate, the height of the bath or the temperature. Some sub-phenomena can be practically modelled, like the bubble size given the flow rate and other relevant parameters (though with a degree of uncertainty), while other are very little understood. For the latter, regression to measured data will give an estimate of the physical parameters as a function of the process parameters.
Basic sketch of the method:



The regressor predicts a species diffusivity based on the known process parameters. It is a pure data-based model. The physics-based process model takes the process parameters and the predicted diffusivity and predicts a final concentration. The integration allows training the data-based model *through* the physics-based model in a seamless manner.
Overall architecture / pipeline / workflow (incl. figure – elements according to BDVA)
The method takes data from a database and produces a predictive model. It can be trained on static data (historic data), but can also be trained continuously on dynamic data to follow the evolution of the process.
Interfaces (in/out) – system/user
The method is implemented in python.
Subordinates/parts – any platform dependencies
Data (in/out)
The method predicts numerical data. It takes numerical input data, but could be adapted to take discrete input data. The data needs to be pre-processed to ensure their quality.
Standards (any standards being used)
Licenses, etc. (free for use in the project)
Only open source software is used
TRL for overall component/tool and any parts/subordinates
References – incl. web etc.
To be considered in particular for the following COGNITWIN pilots
Hydro, Elkem, Sidenor

17 Annex 6. Pragmatic framework for development of hybrid models

In order to work with hybrid models the first step is to develop new or adapt existing models and software in a manner that will satisfy the needs for a given Digital Twin development. In order to support the work process, from understanding the challenge, writing Use Case, establishing a team of system architects, arriving at the concepts and methods, and performing the actual work, a systematic approach has been proposed [1]. The pragmatic model developments steps can be organized as follows:

- Problem and Context Identification
- Analytical Strategy and Plan
- Architecture of the Analytical Framework
- Execution (Orchestration of Analyses, Simulations and Experiments)
- Evaluation of the Solution
- Conclusion and Communication

The pragmatic model development steps are illustrated in Figure 1. Key elements in the pragmatic model developments is that the developments team has a strong foundations in physics, chemistry, thermodynamics, mathematics and statistics. In addition, a deep understanding of the processes in question, as well as the available modeling concepts, selection of the best team and understanding of the available data, are all factors that are critical for obtaining the best possible result.

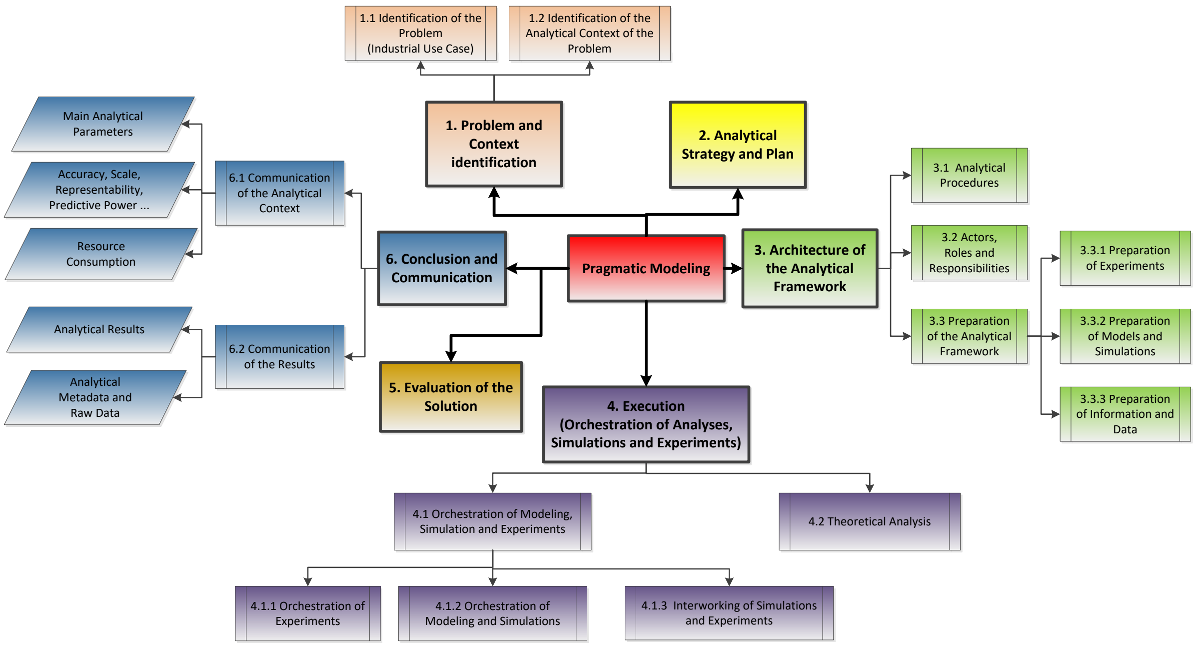


Figure 1 Some of the important phases, processes and results in a typical pragmatic analysis [1].

In "Pragmatic Modeling" we take advantage of all the physics and chemistry that is known and user the available data to tune the model, to explore deficiencies in the model, and to improve weak model elements. In some cases, it may be both useful and necessary to develop sub models that are 100% data based. This approach is what we call hybrid as it exploits the combination of physics and data-based models which has the better accuracy and prediction power.

By basing the pragmatic model on known physics and chemistry the models will have the possibility to predict outside the domain of existing data. Such models have therefore very useful when we want to investigate new ways to run a process. This is contrary to a data-based model which can only be relied if we operate inside the domain of experience.

The methods of "pragmatism in industrial modeling" are used by many teams worldwide and with great success. Some papers have explicitly discussed extensions and applications of the pragmatic framework. In [2] a combination of laboratory data and a physics based model for particle sedimentation in a flowing non-Newtonian fluid was explored and discussed. In [3] the concept of "pragmatism in industrial modeling" was applied to an oil&gas field drilling Use Case, and it was demonstrated how a very powerful model framework could be developed in a very short time. Another application of the framework was related to metal yield in a Ferro Chromium tapping process [4]. In this case a physics-based model was developed that helped to cut the loss of metal to slag by a significant amount. The model suggested a minor rebuild of the furnaces tapping area and this rebuild resulted in the improvement as predicted. In this case the only available data was visual observations and long-term mass balances.

[1] Zoric, Josip, Johansen, Stein Tore, Einarsrud, Kristian Etienne, and Solheim, Asbjørn, “ON PRAGMATISM IN INDUSTRIAL MODELING,” in *Progress in Applied CFD ; Selected papers from 10th International Conference on Computational Fluid Dynamics in the Oil & Gas, Metallurgical and Process Industries*, Trondheim, 2015, vol. 1, pp. 9–24.

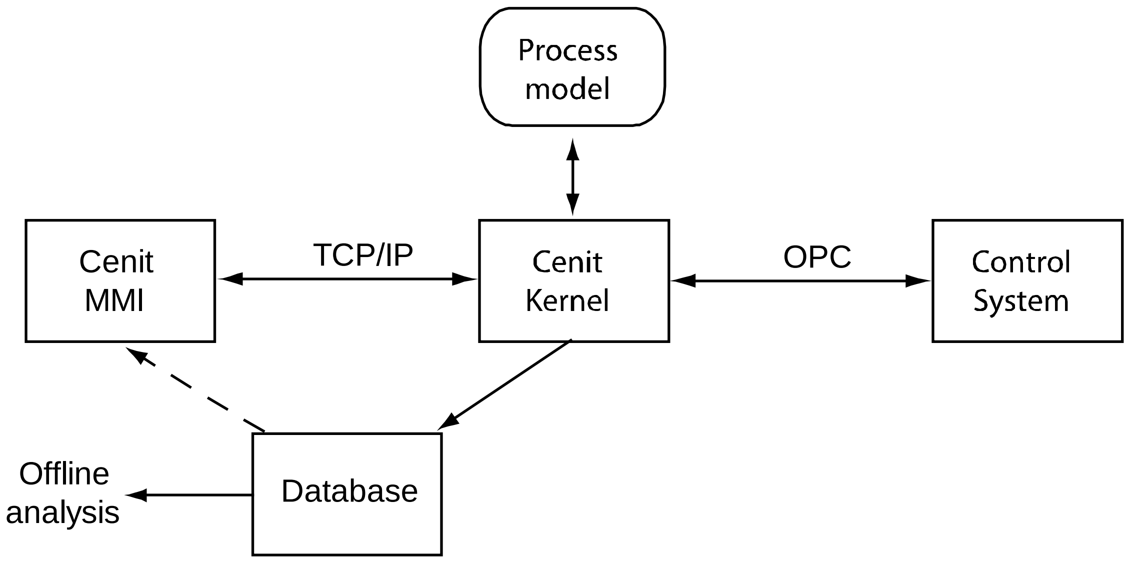
[2] J. Zoric *et al.*, “On Pragmatism in industrial modeling - Part II: Workflows and associated data and metadata,” presented at the The 11th International Conference on CFD in the Minerals and Process Industries, Melbourne, Australia, 7-9 December, 2015, 2015, p. 7 pages.

[3] Johansen, Stein Tore, Meese, Ernst A., Zoric, Josip, Islam, Aminul, and Martins, Dwayne w., “ON PRAGMATISM IN INDUSTRIAL MODELING PART III: APPLICATION TO OPERATIONAL DRILLING,” in *Progress in Applied CFD – CFD2017 Proceedings of the 12th International Conference on Computational Fluid Dynamics in the Oil & Gas, Metallurgical and Process Industries*, vol. 2, 2 vols., Trondheim, Norway: SINTEF Proceedings, 2017.

[4] S. T. Johansen and E. Ringdalen, “Reduced metal loss to slag in HC FeCr production - by redesign based on mathematical modelling,” in *Furnace Tapping 2018 Conference, Edited by J.D. Steenkamp & A. Cowey, Kruger National Park, 14-17 October 2018*, Kruger National Park, South Africa, vol. Symposium Series S98, pp. 29–38.

18 Annex 7. Cybernetica Tool components

18.1 Cybernetica CENIT

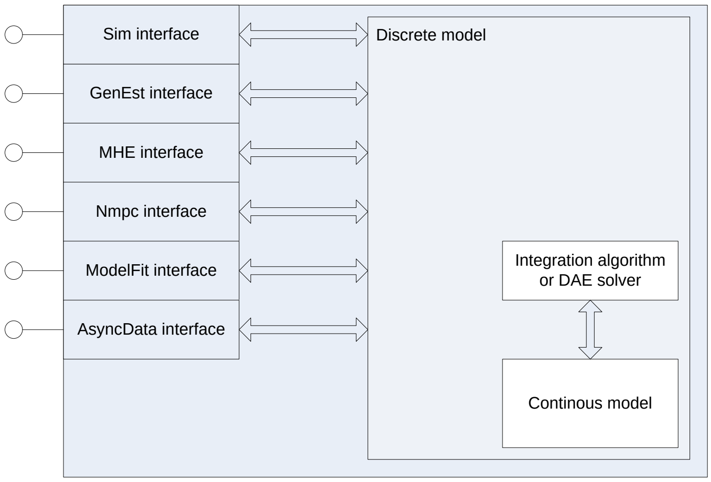
Component/Tool description
Component/Tool/Method/Framework/Service Name
Cybernetica CENIT
Short Description – incl. Purpose
Cybernetica CENIT is a tool for online estimation and nonlinear model predictive control. It can be used as both a soft sensing application and a control application. Model Predictive control is an advanced control method where a mathematical model of the process is used to predict future behavior. The predictions from the model are used in a mathematical optimization algorithm that calculates the optimal process inputs in order to achieve optimal future behavior of selected variables in the process. Constraints and setpoints may be imposed both on the manipulated process inputs variables and the controlled process output variables. Model predictive control also has the advantage that couplings between variables in the process are taken into account.
Function – suitable for which process steps (ICT/Data process) ***Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation***
Data collection, control, visualisation.
Examples of usage / illustrations
Overall architecture / pipeline / workflow
Main components of Cybernetica CENIT: 

Cybernetica CENIT consists of a generic part and an application-specific part, namely the process model. A Cybernetica CENIT application is defined as Cybernetica CENIT and some process model together.
The following table describes the main components of a Cybernetica CENIT application:

Component	Purpose
CenitKernel	This is the main component of Cybernetica CENIT. It implements communication with the process control system and the calculation algorithms (estimator and nonlinear model predictive controller).
CenitMMI	This is an engineering interface used to configure and supervise CenitKernel, mainly during the engineering phase of the project. The operators interface is normally integrated in the existing DCS interface.
Process model	This is the application-specific part of a Cybernetica CENIT application. It implements a mathematical representation of the process that is controlled.
Database	An optional database for logging parameters and calculated data from CenitKernel. The data is used both by CenitMMI and for offline data analysis, and can be used to trend inputs, states and other calculated values.
Control system	This is the process control system (DCS/ PLC), which handles the low-level communication with the process. This system is not a part of Cybernetica CENIT and should implement an OPC server on a standard form to handle the communication with CenitKernel. Both OPC Classic and OPC UA interfaces are supported by Cenit. The communication includes process measurements, manipulated variables and possibly other variables as well.

The model component is implemented as a Microsoft Windows dynamic link library (DLL). One or more model interfaces can be implemented in such a DLL, depending on which calculation modules shall be used. It is not necessary to implement unused interfaces.

The interfaces do not depend on each other, and it is possible to implement different models for each interface, i.e., a complex model for the simulator interface and a simpler model for the controller. However, it is quite common to implement the same model for all the interfaces. The figure below shows how to do this. In this figure, there is a common inner model code base for all the interfaces:



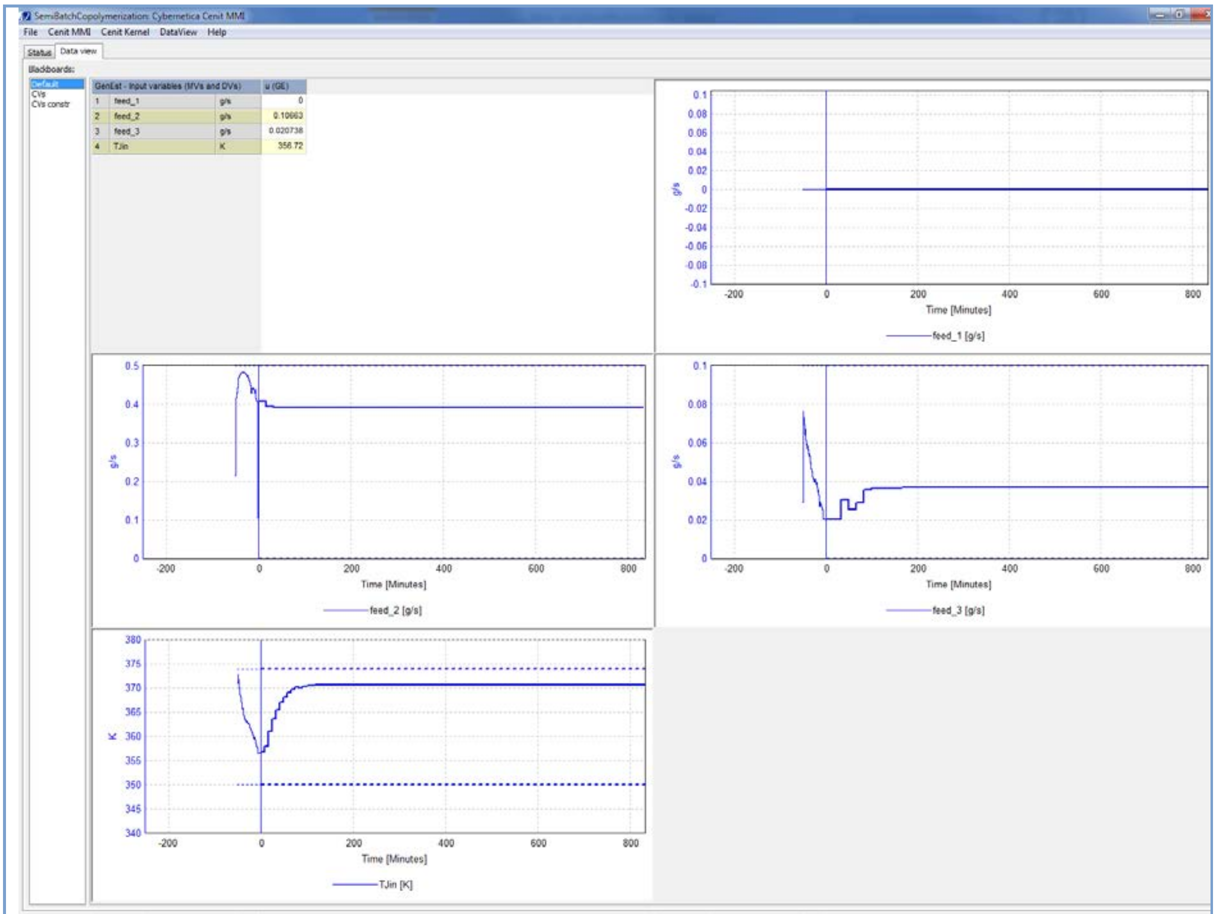
The available interfaces are:

- Sim interface: Used to simulate the process.
- GenEst interface: Used by the Kalman Filter.
- MHE interface: Used by the Moving Horizon Estimator.
- Nmpc interface: Used by the non-linear predictive controller
- ModelFit interface: Used by Cybernetica ModelFit.
- AsyncData interface: Used by Cybernetica Cenit to handle input data that requires special handling; e.g. registration of process event data.

Interfaces (in/out) – system/user

Data can be presented to the user by using Cybernetica CenitMMI, or extracted from the database using the included tool getdbdata.

Example of CenitMMI displaying some historical trend and prediction plots for some manipulated variables:



Subordinates/parts – any platform dependencies

May use PostgreSQL database.

Data (in/out)

In: Process measurements and other data via OPC (classic or UA).
Out: Estimated process values (soft sensor) and manipulated values (non linear model predictive control).

Standards (any standards being used)

OPC Classic DA, OPC Unified Architecture DA

Licenses, etc. (free for use in the project)

Cybernetica Cenit licenses are provided free of charge for the duration of the COGNITWIN-project for project partners who need such license to execute their work in the project. Should the project result be taken into permanent use after the end of the project, licenses are provided on fair and reasonable terms as stated in the Grant Agreement.

TRL for overall component/tool and any parts/subordinates

9 - Commercial product.

References – incl. web etc.

http://cybernetica.no/technology/model-predictive-control/

To be considered in particular for the following COGNITWIN pilots
Hydro, Elkem

Component/Tool description
Component/Tool/Method/Framework/Service Name
Cybernetica Cognitive CENIT
Short Description – incl. Purpose
This is a planned extension of the existing Cybernetica CENIT that will add cognition to the application. The goals of the extension are to: - Combine mechanistic modelling of physical processes with machine learning/ AI - Exploit big data sets from the process to improve the model Both generic functionality and application specific in the form of a new model interface will be added. The cognitive extension may either extend the current estimator (digital twin) or it may replace it entirely. Cybernetica Cenit already implements adaption in the form of parameter estimation. In addition we would like develop and implement methods for real-time and offline analysis of the estimator (digital twin) performance related to process data. In this way it should be possible to automatically classify types of errors: sensor failure, input error or model error. Ultimately, the goal will be to suggest model improvements based on this analysis.
Function – suitable for which process steps (ICT/Data process) ***Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation***
All
Examples of usage / illustrations
Example 1: Error classification: Estimators are generally unable to distinguish between prediction deviations resulting from the following errors: • Faulty input data (requires correction or scepticism) • Faulty model (suggest adaption) Being able to distinguish between these errors is important because the required response is very different:

In the case of input error, the appropriate response is some combination of correcting the faulty input signal and minimizing the faulty signal's impact on the model-predictive control. This can include:

-Using a default signal instead of the faulty signal,
-Ignoring model state variables that are highly correlated with the faulty signal, and
-Altogether turning off estimation for the affected data points.

In the case of model error, the appropriate response is to try to adapt the model to most accurately reproduce the process data.

An important goal for Cognitive CENIT will be to distinguish between these cases based on an offline training of a classification algorithm.

Example 2:
Situations where the model structure is incomplete or wrong may be identified using an automated analysis of the prediction error distributions. Currently Cybernetica CENIT estimators assume that the model structure is correct, and that the prediction error is normally distributed around a mean value, which the estimator tries to center at zero. In many cases this is not true, and significant deviation from normally distributed error may imply error in the model structure. Identifying this error is non-trivial and may be a well-suited task for an AI extension.

18.2 Cybernetica ModelFit

Component/Tool description

Component/Tool/Method/Framework/Service Name

Cybernetica ModelFit

Short Description – incl. Purpose

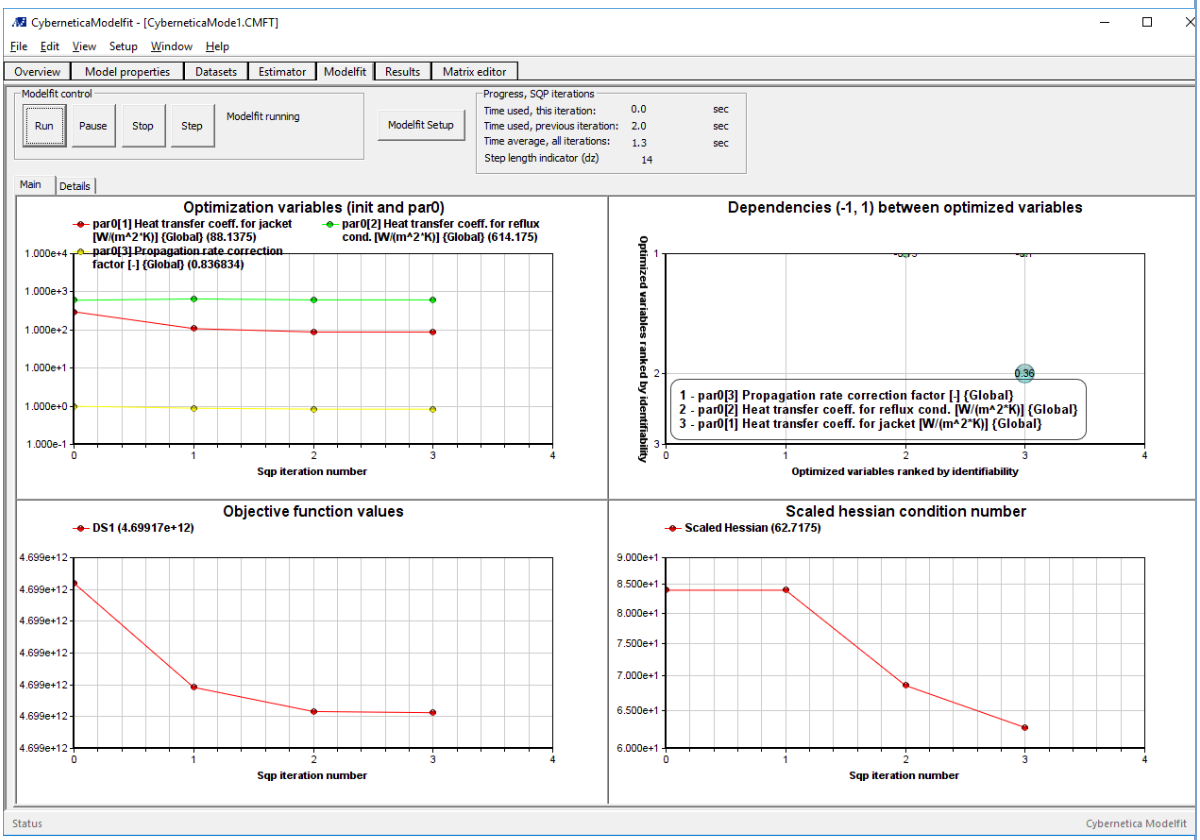
Cybernetica ModelFit is a tool used for off-line estimation of model states and parameters, for model validation, and for design of the on-line estimation part of Cybernetica CENIT applications. ModelFit is used to decide which model parameters should be estimated on-line, to design the on-line estimators, and to estimate the parameters that are considered constant. ModelFit interfaces to Cybernetica Model and Application Components, and it supports the same model formats as CENIT.

Function – suitable for which process steps (ICT/Data process)
Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation

Data processing, analytics, visualisation.

Examples of usage / illustrations

Cybernetica ModelFit user interface:



The features of Cybernetica ModelFit include:

• Design and tuning of on-line estimators in CENIT applications. • Estimation of constant or time varying model parameters. • Estimation of initial states. • Simultaneous use of multiple data sets. • Parameter identifiability analysis. Cybernetica ModelFit is flexible with respect to configuration of the parameter estimation. Parameters can be time varying or constant. Multiple data sets from different operating conditions may be used to find the best parameter fit taken all data sets into account**.**
Overall architecture / pipeline / workflow (incl. figure – elements according to BDVA)
Interfaces (in/out) – system/user
Subordinates/parts – any platform dependencies
Data (in/out)
In: Files with logged data. Cybernetica CENIT database. Out: Result files.
Standards (any standards being used)
N/A
Licenses, etc. (free for use in the project)
Cybernetica ModelFit licenses are provided free of charge for the duration of the COGNITWIN-project for project partners who need such license to execute their work in the project. Should the project result be taken into permanent use after the end of the project, licenses are provided on fair and reasonable terms as stated in the Grant Agreement.
TRL for overall component/tool and any parts/subordinates
9 - Commercial product.
References – incl. web etc.
http://cybernetica.no/technology/model-predictive-control/
To be considered in particular for the following COGNITWIN pilots
Hydro, Elkem

18.3 Cybernetica RealSim

Component/Tool description

Component/Tool/Method/Framework/Service Name

Cybernetica RealSim

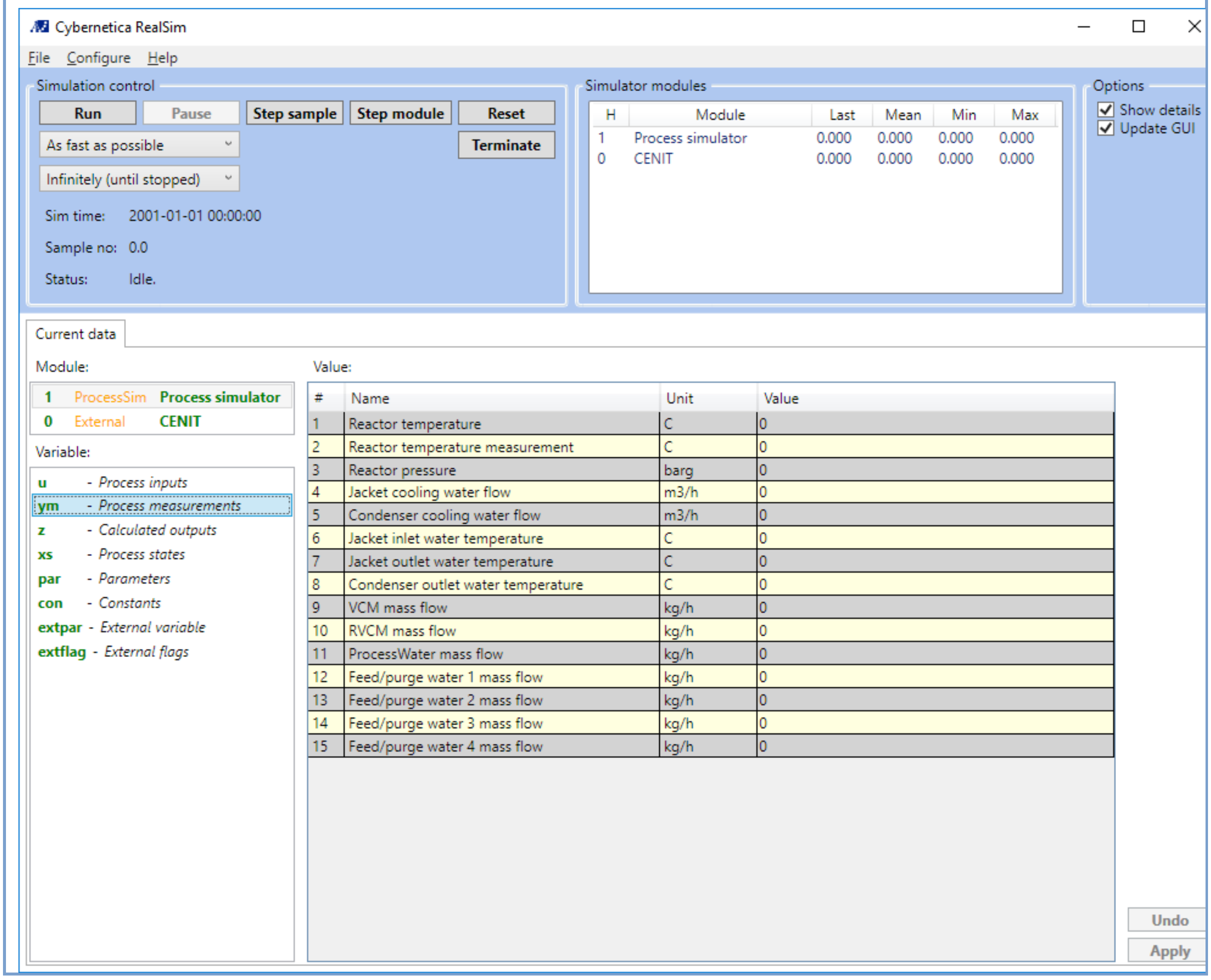
Short Description – incl. Purpose

Cybernetica RealSim is a plant replacement process simulator used for testing of CENIT or other control applications. It communicates over the OPC protocol in order to replicate the interface to the DCS at the plant as closely as possible. It interfaces to Cybernetica Model and Application Components. The plant replacement model might be the same as the model used in CENIT or it might be a different one in order to evaluate how the controller responds to model uncertainty and unknown process disturbances. Cybernetica RealSim is typically used during application development and for factory acceptance tests (FAT).

Function – suitable for which process steps (ICT/Data process)
Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation

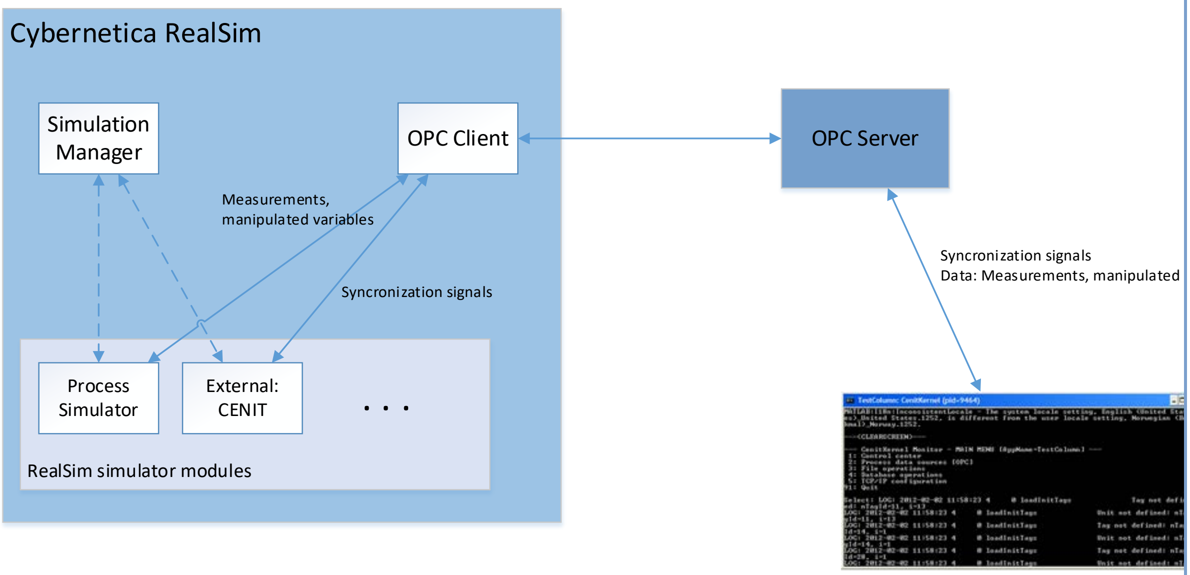
Examples of usage / illustrations

Example of Cybernetica RealSim user interface:



Overall architecture / pipeline / workflow (incl. figure – elements according to BDVA)

The following figure shows how Cybernetica RealSim works as a plant replacement tool for Cybernetica CENIT:



The external module: Cybernetica CENIT

Interfaces (in/out) – system/user

Subordinates/parts – any platform dependencies

Data (in/out)

Standards (any standards being used)

Licenses, etc. (free for use in the project)

Cybernetica RealSim licenses are provided free of charge for the duration of the COGNITWIN-project for project partners who need such license to execute their work in the project. Should the project result be taken into permanent use after the end of the project, licenses are provided on fair and reasonable terms as stated in the Grant Agreement.

TRL for overall component/tool and any parts/subordinates

8

References – incl. web etc.

http://cybernetica.no/technology/model-predictive-control/

To be considered in particular for the following COGNITWIN pilots

Hydro, Elkem

18.4 Cybernetica Viewer

Component/Tool description
Component/Tool/Method/Framework/Service Name
Cybernetica Viewer
Short Description – incl. Purpose
Cybernetica Viewer is a tool for creating user interfaces to display an manipulate data from an OPC server in various ways.
Function – suitable for which process steps (ICT/Data process) ***Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation***
Data visualisation.
Examples of usage / illustrations
Overall architecture / pipeline / workflow (incl. figure – elements according to BDVA)
Interfaces (in/out) – system/user
Subordinates/parts – any platform dependencies
Data (in/out)
Standards (any standards being used)
Licenses, etc. (free for use in the project)
Cybernetica Viewer licenses are provided free of charge for the duration of the COGNITWIN-project for project partners who need such license to execute their work in the project. Should the project result be taken into permanent use after the end of the project, licenses are provided on fair and reasonable terms as stated in the Grant Agreement.
TRL for overall component/tool and any parts/subordinates
9
References – incl. web etc.
To be considered in particular for the following COGNITWIN pilots

18.5 Cybernetica ProXim

Component/Tool description

Component/Tool/Method/Framework/Service Name

Cybernetica ProXim

Short Description – incl. Purpose

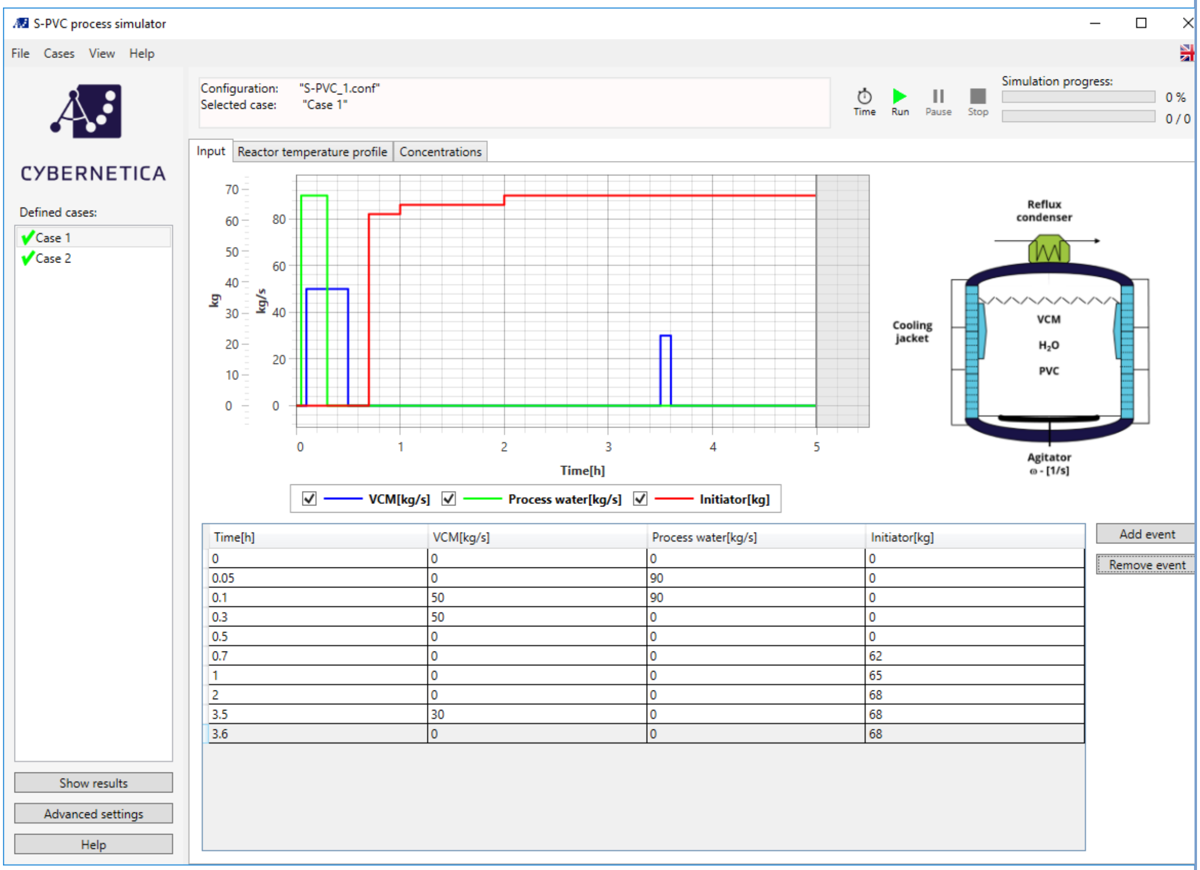
Cybernetica ProXim is a software platform for building tailor-made process simulators using the same kind of process models as Cybernetica CENIT.
The platform includes components for simulation and data visualisation.

Function – suitable for which process steps (ICT/Data process)
Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation

Data analytics, decision support and visualisation.

Examples of usage / illustrations

Example of the user interface of a process simulator:



Overall architecture / pipeline / workflow (incl. figure – elements according to BDVA)

Interfaces (in/out) – system/user

Subordinates/parts – any platform dependencies
Data (in/out)
Standards (any standards being used)
Licenses, etc. (free for use in the project)
TRL for overall component/tool and any parts/subordinates
8
References – incl. web etc.
http://cybernetica.no/technology/dynamic-simulation/
To be considered in particular for the following COGNITWIN pilots
Elkem, Hydro

18.6 Cybernetica OPC UA DA Server

Component/Tool description

Component/Tool/Method/Framework/Service Name

Cybernetica OPC UA Server

Short Description – incl. Purpose

The Cybernetica OPC UA Server is a general purpose OPC UA server supporting the Data Access (DA) interface. It can be used as a hub for exchanging real-time data from processes with other clients that support OPC UA.

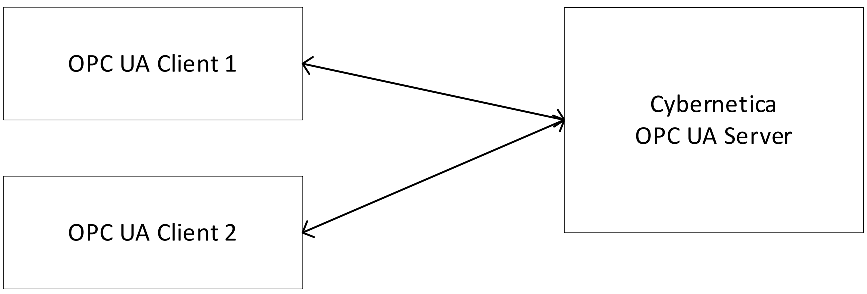
The OPC UA server has a plugin API that allows specialized plugins to be developed. These can be used to collect and distribute data from other data sources (like databases, process control systems or simulators).

Function – suitable for which process steps (ICT/Data process)
Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation

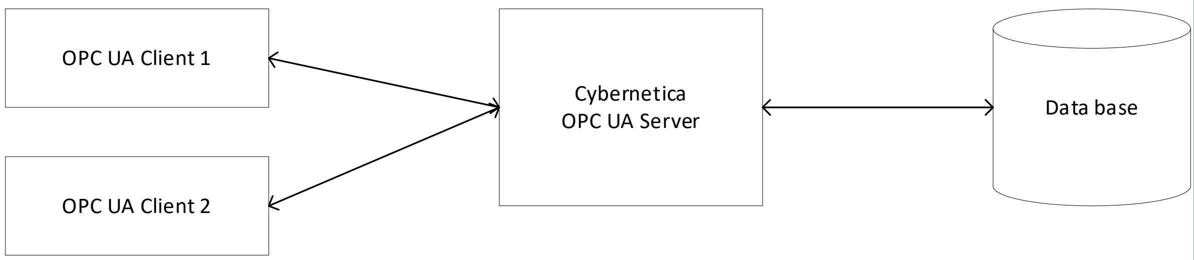
Data collection, integration, sharing.

Examples of usage / illustrations

Example 1: Real-time data exchange



Example 2: Distributing data from a database (or DCS or some other source)



Overall architecture / pipeline / workflow (incl. figure – elements according to BDVA)

Interfaces (in/out) – system/user

OPC UA Data Access (DA).

Subordinates/parts – any platform dependencies

Data (in/out)
Standards (any standards being used)
Licenses, etc. (free for use in the project)
Cybernetica OPC UA Server licenses are provided free of charge for the duration of the COGNITWIN-project for project partners who need such license to execute their work in the project. Should the project result be taken into permanent use after the end of the project, licenses are provided on fair and reasonable terms as stated in the Grant Agreement.
TRL for overall component/tool and any parts/subordinates
8
References – incl. web etc.
To be considered in particular for the following COGNITWIN pilots

19 Annex 8. Scortex Tool components

Component/Tool description
Component/Tool/Method/Framework/Service Name
Bonzai
Short Description – incl. Purpose
Bonzai is the Scortex machine learning library. Built on top of keras, it enables the company to train and evaluate deep learning models.
Function – suitable for which process steps (ICT/Data process) ***Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation***
Data analysis. Dataset curation. Machine learning training. Machine learning evaluation.
Examples of usage / illustrations
During a project: Scortex will use the library to: • Load and curate datasets • Launch keras / tensorflow trainings • Evaluate the performances of the trainings • Logs dataset, trainings, results in database
Overall architecture / pipeline / workflow (incl. figure – elements according to BDVA)
Interfaces (in/out) – system/user
Subordinates/parts – any platform dependencies
Mostly keras and tensorflow
Data (in/out)
In: Azure blobs images & Datasets in Scortex format. Out: keras models. Logged trainings information & results.
Standards (any standards being used)
PEP8
Licenses, etc. (free for use in the project)
Proprietary. In development, remains the property of Scortex. Will be used by Scortex exclusively.
TRL for overall component/tool and any parts/subordinates
TRL 7
References – incl. web etc.
To be considered in particular for the following COGNITWIN pilots
This library will be used to train networks. Additionally all new features to accelerate models will be developed in this tool Light architectures Folding Quantized training and inference

Component/Tool description
Component/Tool/Method/Framework/Service Name
Como
Short Description – incl. Purpose
Scortex annotation tool
Function – suitable for which process steps (ICT/Data process) ***Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation***
Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation Data visualisation, annotation.
Examples of usage / illustrations
Scortex uses como
Overall architecture / pipeline / workflow (incl. figure – elements according to BDVA)
Interfaces (in/out) – system/user
Web interface. • Input are information from mongodb and images stored in azure • Output are annotation saved in a mongodb database
Subordinates/parts – any platform dependencies
MongoDb. Scortex storage system.
Data (in/out)
Annotations saved in mongodb
Standards (any standards being used)
Licenses, etc. (free for use in the project)
Proprietary. In development, remains the property of Scortex. Will be used by Scortex exclusively.
TRL for overall component/tool and any parts/subordinates
TRL 8
References – incl. web etc.
To be considered in particular for the following COGNITWIN pilots
Is used by Scortex to annotate data on which we train.

Component/Tool description
Component/Tool/Method/Framework/Service Name
Keras / tensorflow
Short Description – incl. Purpose
Tensorflow and keras are open source machine learning / deep learning libraries
Function – suitable for which process steps (ICT/Data process)   ***Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation***
Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation
Examples of usage / illustrations
During a project: Scortex uses keras and tensorflow via their bonzai library in order to: • Launch keras / tensorflow trainings   • Evaluate the performances of the trainings   • Visualize the results and understand model behaviour   • Predict in real time on the production line (when using GPU)
Data (in/out)
In: Azure blobs images & Datasets in Scortex format.   Out: keras models. Logged trainings information & results. Predictions.
Standards (any standards being used)
PEP8
Licenses, etc. (free for use in the project)
Open source
TRL for overall component/tool and any parts/subordinates
TRL 9
References – incl. web etc.
Keras documentation: https://keras.io/layers/convolutional/   Keras github: https://github.com/keras-team/keras   Tensorflow documentation: https://github.com/keras-team/keras   Tensorflow github: https://github.com/tensorflow/tensorflow
To be considered in particular for the following COGNITWIN pilots

20 Annex 9. UOULU tools

Component/Tool description
Component/Tool/Method/Framework/Service Name
Method: Integration of physics-based model and data for state estimation Method II: Associated tools/models for plant modelling and simulation. Tool III: Finite Markov Chains Matlab toolbox (MCPC)
Short Description – incl. Purpose
First principle models need to be adjusted using real data, due to unmodelled phenomena, model simplifications, unforeseen changes in process, etc. It is common that important plant characterizations/states cannot be directly and/or reliably measured, or that such measurement is too expensive. State estimation fuses both process models and data to provide better knowledge on the unknown states. Tools include both deterministic and stochastic/Bayesian approaches and can be based on simulations (of plant models) or embedded with models using the model internal structure. The MCPC-Matlab toolbox implements state estimation and control design tools based on finite Markov chain mappings. As such, the methods depend heavily on the availability and type of plant models. The methods find immediate applications in process monitoring and control.
Function – suitable for which process steps (ICT/Data process) ***Data collection, curation, integration, sharing, access, processing, analytics, decision support, control, visualisation***
State estimation methods are necessary when on-line monitoring and control of dynamic processes is developed and applied.
Examples of usage / illustrations
In boilers, the heat value of the incoming fuel may vary a lot, due to changes in fuel quality, moisture, molar feed rate, etc. Given a simplified model where the heat value appears as a dynamic parameter or state, and a plant model linking the (measured) plant inputs to (measurable) plant outputs, an estimate of the heat value or its distribution can be constructed on-line. The approach becomes notably more complex when the number of unknown terms increases, and when the measurable quantities are outcomes further in the process flow.
Overall architecture / pipeline / workflow (incl. figure – elements according to BDVA)
The method takes data on-line and from a database and provides on-line a state estimate of unknown quantities. In most cases, the problems have a dynamic nature.
Interfaces (in/out) – system/user
Algorithm development in Matlab. We aim at max compatibility with free software Octave.
Subordinates/parts – any platform dependencies
Data (in/out)

The method predicts numerical data. It takes numerical input data, but could be adapted to take discrete input data. The data needs to be pre-processed to ensure their quality.
Standards (any standards being used)
Licenses, etc. (free for use in the project)
Matlab is used in development work. Open source software (Octave) for final algorithm tools.
To be considered in particular for the following COGNITWIN pilots
Sumitomo SHI FW Energia Oy