COORDINATED PRODUCTION
FOR BETTER RESOURCE EFFICIENCY

# D5.6 Prototype of the Model Management Platform

**Udo Enste**
*LeiKon GmbH - Germany*

**April 2020**
www.spire2030.eu/copro

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723575

## Project Details

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

## *THE COPRO PROJECT*

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

## The CoPro Consortium

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

## Document details

| | | |
|---|---|---|
| **DELIVERABLE TYPE** | **DEMO VIDEO** | |
| **DELIVERABLE NO** | **5.6** | |
| **DELIVERABLE TITLE** | Prototype of the Model Management Platform | |
| **NAME OF LEAD PARTNER FOR THIS DELIVERABLE** | **LEIKON** | |
| **VERSION** | **1.3** | |
| **CONTRACTUAL DELIVERY DATE** | **APRIL 30, 2020** | |
| **ACTUAL DELIVERY DATE** | **MAY 15, 2020** | |
| **Dissemination level** | | |
| **PU** | Public | **X** |
| **CO** | Confidential, only for members of the consortium (including the Commission) | |

**Abstract**

Model based applications will increasingly be developed for different kind of applications and projects, by different people, with different model development tools and in in different versions. To handle the increasing number of models, companywide model management and maintenance is necessary. Therefore, LeiKon developed a Model Management Platform for versioning and deployment of model master data, model algorithms and model parameter sets of models that were developed by different tools.
This deliverable is provided as a video which shows the engineering steps of the prototype to import and to put models developed by different modelling environments under version control. The video will be public available at the web page of the CoPro project (https://www.spire2030.eu/copro).

### *REVISION HISTORY*

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

| Revision | Date | Description | Author (Organisation) |
|---|---|---|---|
| **V1.0** | 15.04.2020 | Description of a video of the prototype of the Model Management Platform | Enste, LeiKon |
| **V1.1** | 16.04.2020 | Revision by INEOS | Schiermoch, Rahimi-Adli (INEOS) |
| **V1.2** | 17.04.2020 | Final version | Enste, LeiKon |
| **V1.3** | 15.05.2020 | Final approval | S. Engell (TUDO) |

## Disclaimer

# Table of contents

# 1 Executive summary

Model-based applications play an important role, e.g. in process monitoring, planning and scheduling, and advanced process control. The models differ in the mathematical formulation and in their degree of complexity. During the life cycle of model-based applications testing, deployment and change processes must be supported. Commonly occurring change processes are e.g.: tuning of model parameters, new constraints which must be taken into account or changes of the scope of the applications itself. To ensure the transparency, traceability, and also to enable a continuous improvement process, the history of the evolution of the model, changes of the model parameters and constraints as well as information on master data like revision dates, authors, storage locations and deployment status are needed. Up to now usually all this life cycle information are not tracked but is – at best – hidden in comments.

In order to increase the efficiency of the model development and the deployment process of model-based applications, a prototype of a model management platform was developed during CoPro. The platform helps to manage, to deploy and to reuse model-based real-time applications. For this purpose, the platform stores implementation files and model parameters, provides version control and facilitates the management of formal information about the models, the algorithms and the constraints related to models, and the model parameters. Exports of the model descriptions, done by model development tools as well as header information of executables can be imported into the model management platform, in order to store as much information as possible about the core development systems and support the master data management.

The requirements and the functional design of the Model Management Platform are described in deliverable D5.5. Within CoPro the prototype of the model management platform was installed successfully at the site of INEOS in Köln managing more than 100 surrogate models for the energy management department.

A video provided as deliverable D5.6 shows a typical workflow to manage a model-based application within the platform.

# 2 Model Management Platform

Within CoPro, LeiKon developed a Model Management Platform. In the platform, model-based applications can be assigned to projects. It is possible to track meta data of the projects (e.g. their scope, which plant they refer to) and of the models itself (e.g. authors) as well as the history of changes of a model. For user interaction, a web-based user interface is provided. External tools can be connected using an open service-oriented web API.

The model information is stored using functions, given as single (multiple) file(s) which can be uploaded to and downloaded from the Model Management Platform. By uploading a function from an external tool like Matlab, meta data of the function as e.g. its inputs and outputs, their data types and descriptions can be automatically detected and stored. A model dataset also contains model parameters and constraints for each application, which are stored directly in the model management database and can be viewed and changed from the user interface. Changes of functions or

parameters lead to new versions which are also handled within the platform. A version is a set of changes to the files, functions and parameters of a model, (Fig. 1).

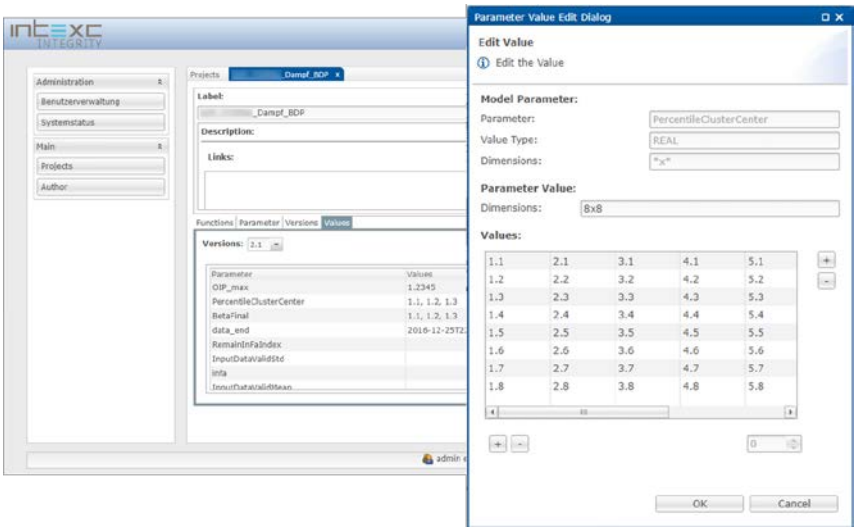


*Figure 1: A parameter matrix versioned by the Model Management Platform*

To support the deployment and the commissioning of model-based applications, the Model Management Platform provides so called “deployment channels”. The use of these channels enables an execution in different environments (e.g. production and testing). A version can be assigned to a channel. In the first phases of the development of a model-based solution, different versions will be executed in a test environment and, after the results have been validated, the final version can be switched to a production environment by changing the channel. If the model modifications are determined to be faulty, the production environment can be switched back to a previous version using the same deployment mechanism. Thus, a seamless and fault-tolerant deployment and commissioning phase that minimize errors is enabled (Fig. 2).

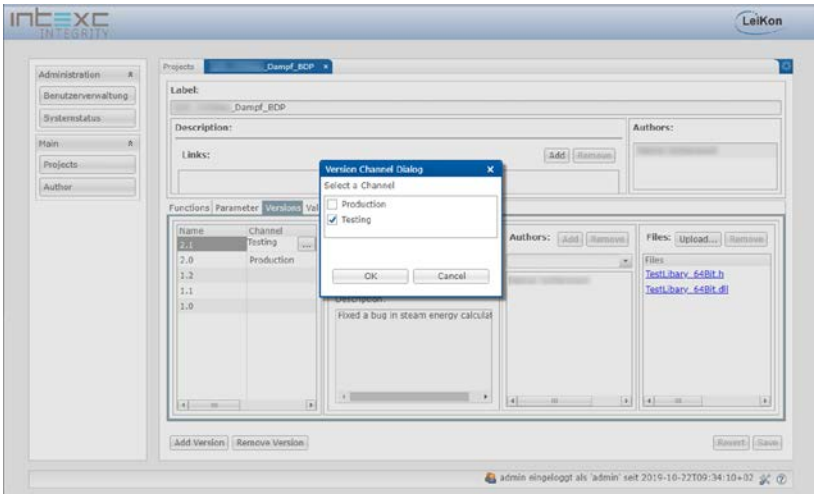


*Figure 2: Deployment of model-based real-time applications*

The model management platform is used within a use case of CoPro at an industrial site to manage more than 100 surrogate models representing Best Demonstrated Practice (BDP) curves of several production plants, which describe reference values for resource efficient operation. The models are developed for whole plants or for different segments of the plants and these are centrally stored in

the model management platform and can be used online in runtime environments to execute the newest released version of the models.

# 3 Description of the contents of the video

The video provided as D5.6 shows a first prototype of the Model Management Platform developed by LeiKon in task 5.3. A use case is presented where a new model was developed by an external tool (here Matlab) in order to realize a model-based real-time application within a specific project. The demonstration follows a typical workflow shown in Figure 3.

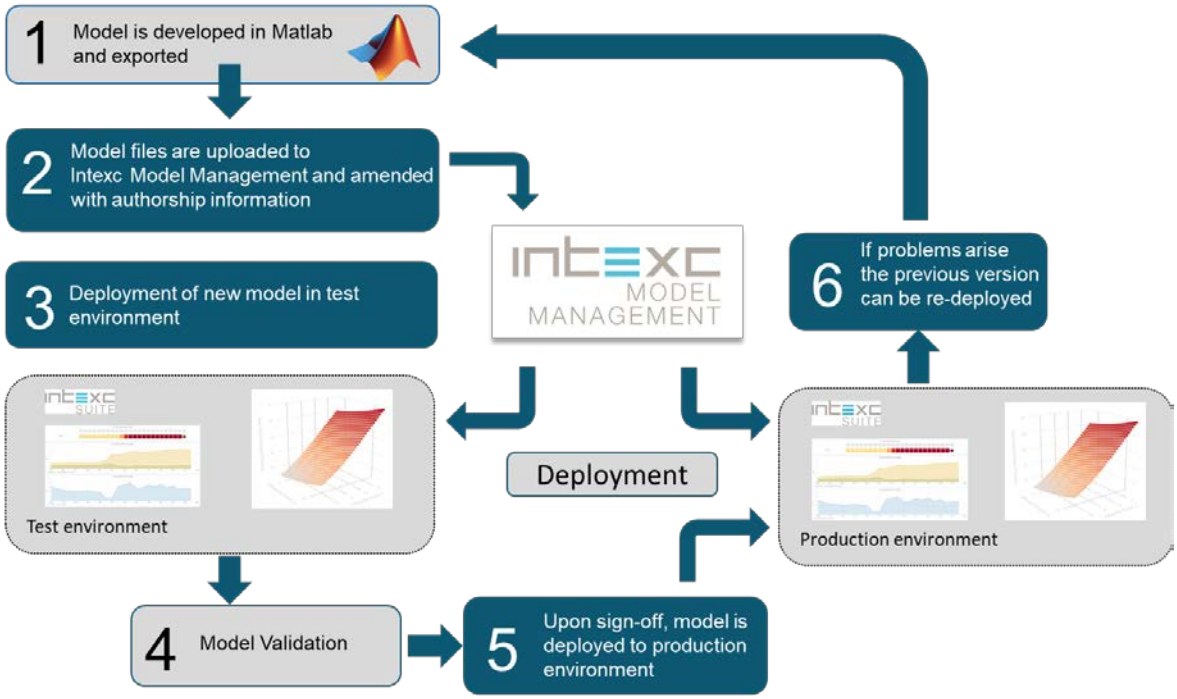


*Figure 3: Workflow of the demo case shown in the video of D5.6*

The model management procedure is organized hierarchically. A model will be assigned to a specific project folder. Within one of these folders an example of a Matlab model will be uploaded. Before the implementation of the model management platform the models are exported as dll files and dedicated header files using the Matlab compiler. During the upload into the model management platform, all available native model information like available model functions and their parameters are extracted from the uploaded files and automatically stored as meta information of the model. Further information can be added manually by a web-based user interface.

The imported model is classified as a test version, triggering deployment into a connected test environment. It can be changed to a productive version after successful testing. The deployment of models within a specific project branch to a runtime-environment like production or testing is limited to a single model at the same time to prevent errors.

As a last step it will be shown how a model update can be performed: An improved pair of dll and header file, which were previously exported from Matlab, are uploaded as a new version. After auxiliary information is added, the new version is first deployed to the test environment. Once the functionality is verified, the new version is deployed to the production environment. In case a problem is identified, the production environment can be rolled back to use the previous version of the model. Afterwards, the model developers can analyse the problem offline, e.g. by using the model development tool.

The video will be provided at:
https://www.spire2030.eu/copro