Towards Explainability in Cyber-Physical Production Systems
(Touching on AI, MLOps, etc. along the way)

Klaus Schmid

Software Systems Engineering
University of Hildesheim
sse.uni-hildesheim.de
Intelligent CPPS

In Industry 4.0 machine intelligence is a basic assumption.

We need:
• Foundations that are able to support AI applications for Industry 4.0 (IIP-Ecosphere ➔ Oktoflow platform)
• Deal with common problems that come together with AI ➔ Explain-project (MLOps, explainability)
Oktoflow-Platform

• Result of the IIP-Ecosphere project
• Model-driven I4.0 platforms
• AI-enabled I4.0 / IIoT
Oktoflow platform

• Vertical AAS integration (from device to application)
• Model-driven approach / Low-code
• Massive code generation (up to 86% in demonstrators)
• Heterogeneous Edge devices
• AI integration (Python, RapidMiner, FLower)
• Standard-based (OPC-UA, AAS, MQTT, ...)
• Open for extensions

Oktoflow Architecture

- SCADA, ES, Visualization, Data exploration
  - Management UI
  - Apps
    - Deployment
    - Configuration
    - Adaption
    - AI-Toolkit
    - Data integration
  - Secure Store
  - Security, data protection, data sharing
  - Transport and Connectors, Service Interfaces / Monitoring
  - Connector
  - Edge, Cloud, Server

- Integrated AI support
- powered by BaSyx

Store
Ano
Coordinated activity + Federated Learning

Cobot 1
(gripping)

Product
identification
(MIP magnetic
sensor)

Lenze linear drive (condition
monitoring on forward path)

Cobot 2
(visual
quality
inspection)
Status of oktoflow

Current work
• Consolidation
• Integration of IDTA AAS standards
• Continued / new collaborations
• Evaluation in further demonstrators

It’s open source: *Use it, extend it and collaborate!*
AI needs Explainability

Foundation of the Explain project

• Machine Learning in the context of CPPS requires a perspective of the whole life-cycle
  ➔ MLOps

• As CPPS systems are typically critical, we need operators and others to understand
  ➔ Explainability
The MLOps Lifecycle

Data Engineering

Data collection → Data analysis → Data preparation

Model engineering

Model building → Model training → Model evaluation → Model selection → Model packaging

Operations

Monitoring ← Model deployment ← CI/CD testing

Supporting activities

Tools
Infra-structure
Versioning
Need and benefits of explainability

Core goal: output explanations = result of deployed models

Explain machine learning outputs:
– why is the current product broken?
– why should modify the operation parameters for the power plant (now)?
Need and benefits of explainability

Additional explanation options:

• During trainings:
  – for identifying potential issues when trying to improve models
  – for review (model acceptance)

• During operations:
  – Additional information on when explanations are wrong
Challenges

MLOps related challenges

• Data-related:
  – Drift (sensor, environment (e.g., lighting)
  – Data annotation (sufficient and ongoing in production)
  – Manufacturer-specific interfaces
  – Data-volume
  ...

• Model-related:
  – Importance of domain knowledge
  – Identifying appropriate models
  – Model update
  ...

• Operations-related:
  – Operations environment (cloud, server, edge)
  – Edge-processing of AI
  ...

Challenges: Explainability

Explainability in MLOps
- Diversity of explanation problems
- Systematic integration in the MLOps life-cycle
- User-interaction for explainability
- Deployment and registries for explainers
Explain MLOps-Architecture
(with explanation support)

Data Administration
- Data Management
  - Samples
  - Metadata
  - Datasets
  - Data ingress
  - Stream datasets
- Data Monitoring
  - Drift metrics
  - Drift detection

Model Training
- ML IDE
  - Programming interface
  - Dependency management
- Model Training
  - Experiment metadata
  - Training job execution
  - Hyperparameter tuning
  - Explainable model training

Model Management
- Model Registry
  - Registered Models
  - Registered Pipelines
- Register models
- Registered Explainers
- Register explainers
- Model Serving
  - Model outputs
  - Explainable model training
  - Trainable models

Model Monitoring
- Performance alarms
- Model comparison
- Trigger model retraining
- Trigger alarms

Feedback
- User Feedback
  - Feedback GUI
  - Prediction feedback
- Explainer feedback
  - Model and explainer feedback

Data
- Datasets
- Metadata
- Samples
- Drift metrics
- Drift detection

Architecture (with explanation support)
Oktoflow-Platform

- Result of the IIP-Ecosphere project
- Model-driven I4.0 platforms
- AI-enabled I4.0 / IIoT

Coordinated activity + Federated Learning

Cobot 1 (gripping)
Cobot 2 (visual quality inspection)
Lenze linear drive (condition monitoring on forward path)

Questions/Comments

The MLOps Lifecycle

Data Engineering
- Data collection
- Data analysis
- Data preparation

Model engineering
- Model building
- Model training
- Model evaluation
- Model selection
- Model packaging

Supporting activities
- tools
- infrastructure
- versioning

Operations
- Monitoring
- Model deployment
- CI/CD testing

Data Administration
- Data Management
- Model Training
- Model Management
- Feedback

Data Observation
- User feedback
- Model and explainer feedback
- Model and version feedback

Product identification (MIP magnetic sensor)
Cobot 1 (gripping)
Cobot 2 (visual quality inspection)
Lenze linear drive (condition monitoring on forward path)

• Result of the IIP-Ecosphere project
• Model-driven I4.0 platforms
• AI-enabled I4.0 / IIoT