RE4DY
MANUFACTURING DATA NETWORKS

RE4DY TOOLKIT

Name of the Tool	**CERTH XAI and Active Learning Platform for Defect Detection (READY Platform for Defect Detection)**
Tool Owner	Industry Commons Foundation
Version	1.0
Date	Nov 2025
Version	V1.0

Table of contents

1. Component Description

The platform offers AI-driven defect detection by analyzing images of manufacturing assets, specifically tailored for the hard metal industry. The defect detection and localization platform is enhanced with AI explainability and human-AI collaborative features. Designed using a micro-service architecture, the platform is adaptable and extensible, catering to a diverse set of users. This includes data scientists who develop AI models and maintainers who monitor conditions with these pre-trained models. At its heart, the platform employs advanced Machine Learning and Deep Learning techniques, ensuring high precision in both defect recognition and localization.

The XAI and Active Learning Core Engine is a central component of the XAI and Active Learning platform, ensuring that key functionalities related to reasoning are seamlessly integrated and executed. It consists of three components briefly described below:

1. **Reasoning/Recommender Module:**
   - **Purpose:** This module stands as the primary component responsible for offering the inference and reasoning services of the platform.
   - **Technical Approach:** To cater to a diverse set of use-cases and ensure robustness in its inference capabilities, this module incorporates a multitude of techniques. Notably, it employs rule-based and graph-based reasoning methodologies. Additionally, fuzzy logic is integrated, enhancing the flexibility and adaptability of the reasoning processes. This ensures that the module can cater to both well-defined and ambiguous data scenarios, thereby broadening its applicability.
2. **Data Management Module:**
   - **Purpose:** At its core, this module's responsibility is to orchestrate the collection and subsequent management of data.
   - **Data Sources:** Data ingested by this module can stem from a variety of origins. One significant source is the simulated data generated by the Digital Twin. This serves to provide a virtual representation of the data scenarios. Additionally, this module integrates historical data and real-time data garnered from other WP3 components, ensuring a holistic data perspective.
3. **Model Management Module:**
   - **Purpose:** This module is dedicated to the meticulous management of AI models. Given the dynamic nature of artificial intelligence and the continuous evolution of models, a dedicated module for this purpose ensures that the most optimized and updated models are always in operation.
   - **Functionality:** Beyond mere storage, this module facilitates version control, monitoring, and updates for AI models, ensuring that they remain relevant and accurate over time.

4. **XAI Module:**

Purpose: This module is dedicated to interpret AI results and decisions.

Functionality: Visual representation (graphs etc.) regarding XAI is provided. For example, graphs explaining the importance of various features during training phase.

2. Input

Main inputs for the XAI and Active Learning Core Engine are:

- **Manufacturing Asset Images** that the system analyzes. Given the specificity for the hard metal industry, these images might reveal intricate details, patterns, and potential anomalies or defects.
- **Model Parameters and Configurations** that guide how the image analysis and defect detection models should operate. For example, training configurations or specific thresholds for defect identification.
- **Feedback Data for Active Learning** collected during defects identification and validation (i.e., either confirmed or corrected defects). This feedback can be looped back into the system to refine and improve the AI models continually.
- **Operational and Historical Data:** Information about the manufacturing processes, historical defect rates, types of defects commonly encountered, and other related data can be valuable inputs. This data can provide context, aiding in the interpretation of results and in making informed recommendations.

3. Output

Main outputs of the XAI and Active Learning Core Engine are:

- **Defect Detection Reports/Graphs** about the type, size, location, and severity of each defect identified in the analyzed images.
- **Explanations and Justification:** For each defect detected, the system offers a human-readable explanation using Explainable AI (XAI) techniques, supplemented by visual aids where appropriate.
- **Recommendations:** Based on the identified defects, the engine generates actionable insights or recommendations, suggesting maintenance, repair, or additional inspections, and may provide predictive insights about potential future issues.
- **Active Learning Feedback Loops:** Leveraging active learning, the engine flags areas of uncertainty in its analyses and seeks human verification, while also reflecting improvements made from previous feedback.

- **Data Summaries:** Users are presented with summaries that provide overviews of the collected data, displaying trends in defect types and frequencies, as well as other relevant data usage statistics.
- **Model Management Summaries:** An overview of all AI models deployed, detailing their versions, update histories, and training data.
- **Alerts and Notifications:** Users are kept informed through real-time alerts and notifications about critical defects, anomalies, system health, and data collection status.

4. Information Flow

AI-Driven Defect Detection

The use case "AI-Driven Defect Detection" outlines the process through which a User (Data Scientist, Maintainer) interacts with the components of the “XAI and Active Learning Platform” (AI System) to analyze manufacturing asset images for potential defects. The procedure encompasses the provision of necessary input data (External Sources), processing of images using pre-trained AI models, generation of detailed defect detection reports, and the retrieval of these results by the User. Optionally, the User can contribute feedback for active learning to continually refine the AI models, enhancing the accuracy of defect detection over time. Through this structured interaction, the User can initiate a defect detection process, review the findings, and optionally participate in a feedback loop to improve future defect detection accuracy, thereby leveraging AI capabilities to maintain high standards of quality control in manufacturing assets.

Primary Actor: User (Data Scientist, Maintainer)

Secondary Actors: External Data Sources, AI System

Preconditions:

- The XAI and Active Learning Platform (AI System) is properly configured and operational.
- Relevant AI models for defect detection are available and properly trained.
- Manufacturing asset images along with any necessary model parameters and configurations are accessible.

Main flow:

- **Input Data Provision** - The User or External Data Sources provide manufacturing asset images, model parameters, and configurations to the AI System.
- **Image Processing** - The AI System processes the provided images using pre-trained machine learning and deep learning models to identify potential defects.

- **Defect Detection Report Generation** - Upon analyzing the images, the AI System generates defect detection reports. These reports detail the type, size, location, and severity of each detected defect.

- **Results Retrieval** - The User accesses the defect detection results, reports, and any associated data visualizations through the Graphical User Interface (GUI) or the REST API.

Optional steps:

- **Active Learning Feedback** - The User can provide feedback data for active learning, which is utilized to refine and improve the AI models continually, enhancing the accuracy of defect detection over time.

Exception paths:

- The User accesses the defect detection results, reports, and any associated data visualizations through the Graphical User Interface (GUI) (or a REST API).

Post-conditions:

- Defect detection reports are generated and accessible to the User.
- Active learning feedback loop may be initiated to enhance future defect detection accuracy.

Trigger:

The User initiates the defect detection process by providing the necessary input data or by instructing the system to utilize the data from External Data Sources.

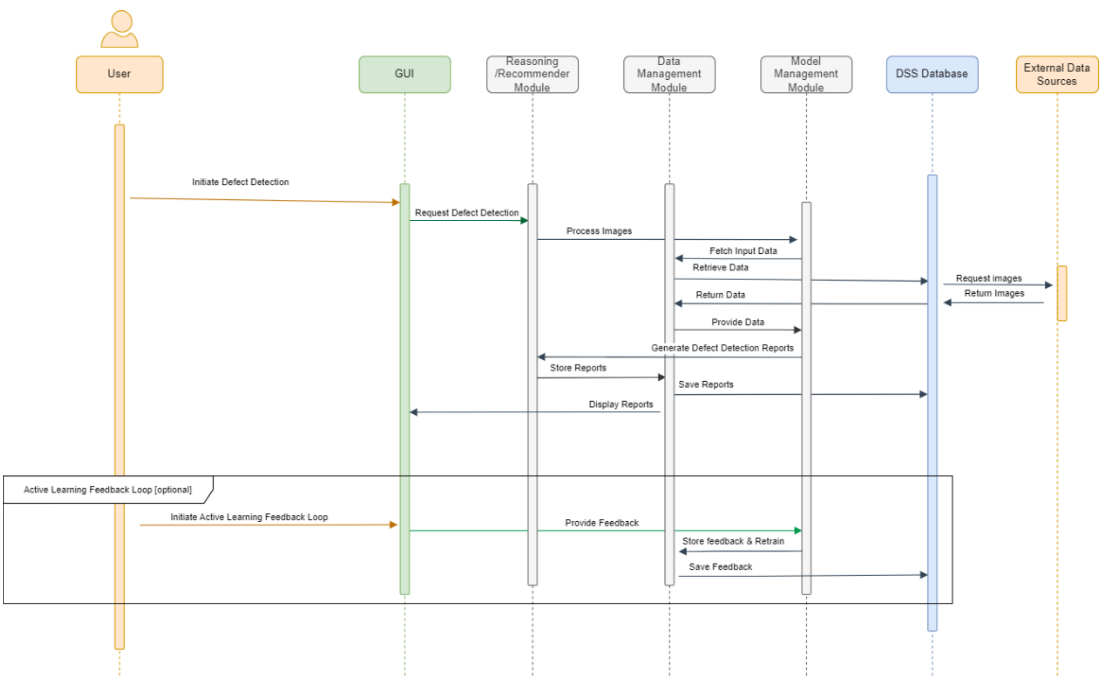

Figure 1 Data Flow of AI Driven Defect Detection

AI Explanation Retrieval

The "AI Explanation Request" use case facilitates a deeper understanding for the User (Data Scientist, Maintainer) by providing insights into the AI decisions and results, particularly pertaining to the identified defects in analyzed images, through the XAI (Explainable AI) Module of the system.

Primary Actor: User (Data Scientist, Maintainer)

Secondary Actors: AI System

Preconditions:

- AI System is properly configured and operational.
- Relevant AI models for defect explanation are available and properly trained.
- Images have been analyzed, and defects have already been identified by the AI System.
- The User has access to the system through the Graphical User Interface (GUI) or the REST API.

Main Flow:

- The User initiates a request for an explanation of the AI decisions and results through the GUI (or a REST API).
- The AI System processes the request through the XAI (Explainable AI) Module to generate explanations and visual representations (e.g., feature importance graphs) regarding the AI decisions.
- The explanations and visual representations are made accessible to the User through the GUI (or a REST API).

Post-conditions:

The User gains insights into the AI decisions through the explanations and visual representations provided by the XAI Module.

Trigger:

The User initiates the process by requesting an AI explanation.

Exception Paths:

In the event of any anomalies or issues during the AI explanation generation, an error notification is generated by the AI System and relayed to the User or System Administrator for resolution.

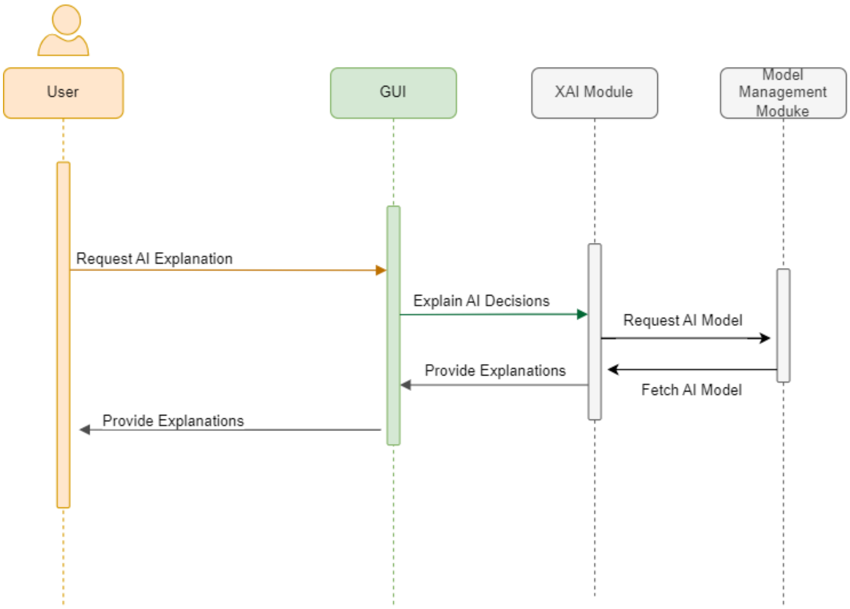

Figure 2: Data Flow of XAI

AI Model Management

The use case "Model Management", delineates the interaction between a User (Data Scientist, Maintainer) and the AI System for effectively managing the available AI models dedicated to defect detection. Through the Model Management Module, accessed via the Graphical User Interface (GUI), the User engages in various model management tasks such as version control, monitoring, and updating AI models based on evolving requirements, feedback, or new data. These actions are facilitated by the AI System to ensure proper storage, tracking, and updating of the models as per the User's actions, enabling a structured approach to managing the AI models to ensure their optimum performance and relevancy over time. Through this defined flow, any encountered anomalies, or issues trigger error notifications, ensuring the User or System Administrator is informed for prompt resolution, contributing to the effective management and utilization of AI models for defect detection.

Primary Actor: User (Data Scientist, Maintainer, System Administrator)

Secondary Actor: AI System

Preconditions:

- The AI System is properly configured and operational.
- Relevant AI models for defect detection are available and properly trained.
- The User has access to the system through the Graphical User Interface (GUI).

Main Flow:

- The User accesses the Model Management Module through the GUI to view and manage the available AI models.
- The User can perform various model management tasks including:
  - Version control to track and manage different versions of AI models.
  - Monitoring to oversee the performance and usage of AI models.
  - Updating AI models based on evolving requirements, feedback, or new data.
- The AI System facilitates these models' management tasks, ensuring the models are properly stored, tracked, and updated as per the User's actions.

Post-conditions:

AI models are managed effectively through the Model Management Module, with their versions controlled, performance monitored, and updates properly executed.

Trigger:

The User initiates the process by accessing the Model Management Module to manage the AI models.

Exception Paths:

In the event of any anomalies or issues during the model management phases, an error notification is generated by the AI System and relayed to the User or System Administrator for resolution.

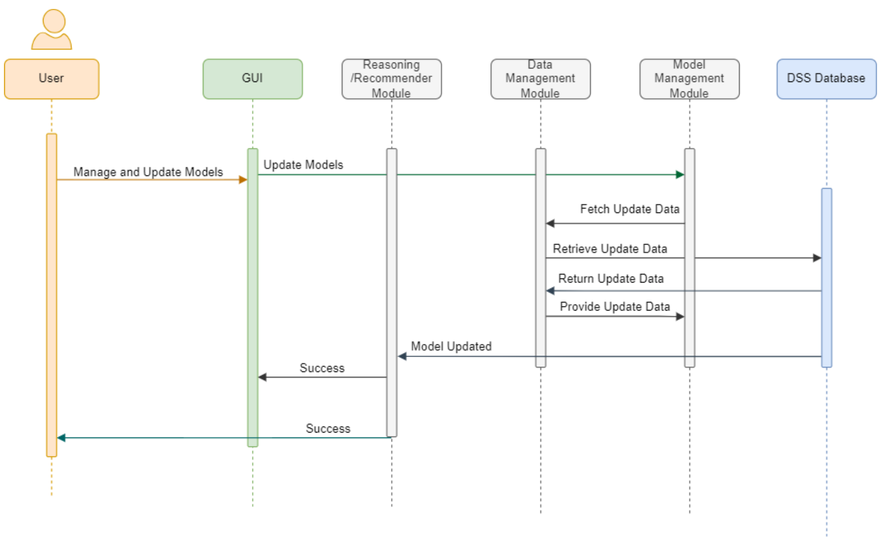

Figure 3 Data Flow for AI Models Management

5. Internal Architecture

The architecture of the platform is structured into three distinct layers: the Presentation Layer, the Service Layer, and the Persistence Layer. The delineation into these three layers embodies a standard architectural design aimed at logically organizing the system's components. This segregation promotes a clear separation of concerns, paving the way for easier maintenance and scalability of the platform.

- The **Presentation Layer** is engineered for user interaction and information presentation.
- The **Service Layer** encapsulates the core business logic, acting as a conduit between the presentation and persistence layers, mediating their interactions.
- The **Persistence Layer** is devoted to data storage and retrieval, providing a robust foundation for the platform's data-centric operations.

Each layer, with its set of dedicated functionalities, interacts cohesively to deliver the comprehensive capabilities of the CERTH Data Transformation Component. This layered design also augments the platform's adaptability, ensuring that modifications or extensions in one layer have minimal ripple effects on the others.

Presentation Layer:

- For RE4DY end-user or other components that will consume data transformation services there are no UIs available. All the calls will be done through Data Space Connectors between data providers and data consumers.
- **In general** User Interface in Apache NiFi serves as the primary interaction point for users, enabling them to design, monitor, and manage their data transformations but this will be used during development phase only. This UI is flexible, and efficient for diverse data integration needs and is going to support a lot the delivery of data transformation services.

Service Layer:

- **Flow Controller:** Is a core component of architecture. Essentially, it is the "brain" behind Data Transformation component operations, orchestrating the processing of data and ensuring that all NiFi and RE4DY custom processors work in harmony.
- **Data Loading sub-component:** Is a custom designed component for data loading, tailored to address unique RE4DY requirements. The component supports functions such as source integration, flexible data ingestion, and error handling.
- **Data Extraction sub-component:** Is a custom designed component for data cleaning and filtering (if needed).

- **Data Transformation sub-component:** Is a custom designed RE4DY data transformation processor, built on top of the Apache NiFi data transformation processors. Some of the envisioned functionalites so far are:
    - **Data Conversion:** Transform data from one format or structure to another. JOLT processors are used as well at this part.
    - **Data Enrichment:** Augmenting data with additional information.
    - **Data Aggregation:** Summarizing or grouping data.
    - **Business Logic Application:** Applying specific logic or rules.
    - **Data Mapping:** Mapping the data to the standardized data models and ontologies selected to be used by the RE4DY use case providers.
- **Data Space Connectors[1]:** They are enabling trusted and sovereign communication between two parties that are involved in a data transformation service. A data consumer that needs a data source with a specific data format will setup a data space connector. This connector will communicate with the relevant data space connector of the data provider. The data from provider data sources will be transformed through the Data Transformation Processors and the result will be transmitted to the consumer.

Persistence Layer:

- **Flow File Repository:** It is responsible for storing the metadata of the Flow Files that are currently being processed by the system. In essence, the Flow File Repository is crucial for ensuring data integrity, system recoverability, and providing a snapshot view of the current state of data.
- **Content Repository:** Is a core component of Apache NiFi that manages the actual content or data associated with the FlowFiles being processed in the system. Unlike the FlowFile Repository, which handles metadata, the Content Repository deals with the data payload.
- **Provenance Repository:** The Provenance Repository is responsible for recording and preserving a comprehensive history of each Flow File that flows through the system. This detailed log enables users to trace the lineage and lifecycle of the data as it is processed.
- **Ontology Repository:** Ontology Repository in the context of data transformation provides a structured, semantic framework that ensures that data is not just transformed in structure but also in meaning. By mapping data to standardized ontologies, it provides a way to ensure data consistency, integration, and meaningful representation across diverse RE4DY use cases.
- **External Data Sources:** The actual data sources from RE4DY partners that is going to be transformed.

[1] https://international-data-spaces-association.github.io/DataspaceConnector/

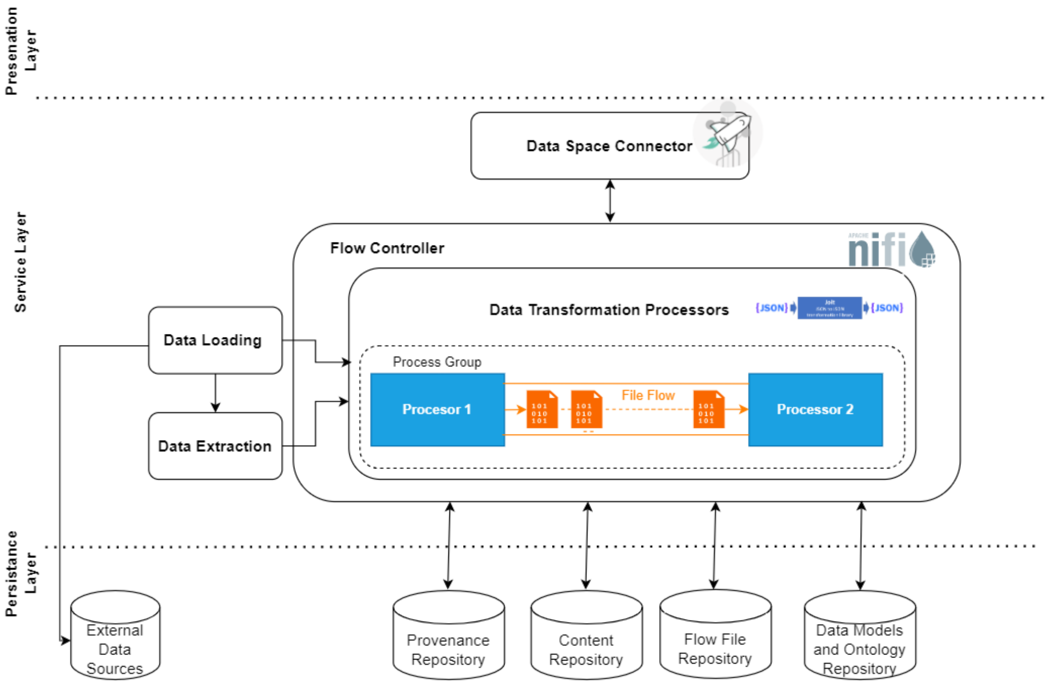

Figure 4: Architecture of RE4DY Data Transformation Services

6. API

TBA in the updated version of the deliverable.

7. Implementation Technology

The combination of IDSA Connectors, Apache NiFi and JOLT for our custom data extraction component ensures that we have both a robust and scalable data extraction framework, as well as specialized capabilities for JSON data transformations. Apache NiFi offers us the infrastructure and environment to manage large-scale data flows, while JOLT provides the specificity and flexibility needed for JSON data manipulations. Together, they form a potent tech stack for our custom data extraction needs.

8. Comments

None.