1.Service description

The threshold-based maintenance triggering service is designed to monitor the health status of SEW production modules every *n* days and give inspection and replacement suggestions for the sub elements that compose a certain module (for more contextual details please refer to Deliverable 5.2). For further versions, this service could be automated to run periodically and provide maintenance recommendations when needed. In the current version, the service is launched manually by the maintenance manager every n days to retrieve maintenance suggestions.

2.Inputs

The service requires two inputs:

* The module ID (appellation) for which the manager wishes to check the health status and/or receive maintenance suggestions. This is a user-defined input.
* A json object containing failure and maintenance data for lines, modules, and sub-elements. This information should be requested from the knowledge data base of MODAPTO and updated by maintenance operators. It follows the following format:

[

{"stage": "",

"cell": "",

"module": "",

"component": "",

"failure type (electrical\/mechanical)": "",

"failure description": "",

"maintenance action performed": "",

"component replacement (yes\/no)": "",

"name of worker":"",

"event date":""},

{

},…

{

}

]

In this version, the module ID and sub-element IDs are predefined as internal parameters, since the approach has only been tuned and implemented for the “Press” module, due to data availability. This means that only the second input (the json object) is currently required. Future versions may incorporate the module ID as a user input, allowing the algorithm to dynamically determine the relevant sub-elements based on a predefined hierarchical dictionary of SEW components.

The input json object should contain the following data types: the value of all keys are of type str. The 'module' and 'component' keys store the IDs of the corresponding module and component; these can be either str or int.

The algorithm is currently working with an adapted version of the database provided by SEW, renamed here as “CORIM\_tool\_test.json”.

3.Output

The output of the algorithm is a .json file named maintenance\_recommendations.json containing:

* Maintenance actions to be scheduled
* Contextual information about the maintenance recommendation

The json output looks like this when no recommendation is given:

{

"recommendation": "non",

"details": "The system is under control"

}

The json output looks like this when a recommendation is given:

{

"recommendation": "schedule a replacement of sub element 100019",

"details": "It failed 16 times in the last 270 days"

}

The last output scenario should then be converted to notifications, which can be used to schedule maintenance actions and to update production schedules accordingly.

4.Tests

* To simulate replacement suggestions, please set the parameter replacement\_threshold = 7 and today = pd.to\_datetime("21/08/2024", dayfirst=True).
* To simulate an inspection suggestion, please set replacement\_threshold = 8 and today = pd.to\_datetime("21/08/2024", dayfirst=True)