A Smart Decision Making System for the Optimization of Manufacturing Systems Maintenance using Digital Twins and Ontologies

ABADI Mohammed[1], ABADI Chaimae[2], BEN-AZZA Hussain[4]
Laboratory of Mechanics, Mechatronics and Command
Modeling, Information Processing and Control of Systems (MTICS) Team
Ecole Nationale Supérieure d'Arts et Métiers (ENSAM-Meknès)
Moulay Ismail University, B.P. 4042, 50000
Meknes, Morocco

ABADI Asmae[3]
Euromed Research Center
Euromed University of Fez
Fez,
Morocco

***Abstract*—Now-a-days manufacturing processes are becoming more and more complex which constantly complicate the management of their life cycle. Although, in order to survive and maintain a good position in the competitive industrial context, industrials have understood that they must optimize the whole life cycle of their manufacturing processes. The maintenance constitutes one of the key processes indispensable to ensure the proper functioning and to optimize the lifetime of machines and production lines, and thus to optimize quality and production costs. Therefore, its automation and optimization represent until now a center of interest for researches and manufacturers, especially those related to the integration of artificial intelligence tools in the industry. In this context, several new concepts and technologies have emerged, particularly in the context of industry 4.0. One of these new concepts is digital twins, which has become a promising direction to optimize manufacturing processes lifecycle. However, the implementation of this technology faces several complex problems related to the interoperability between physical entities and their virtual counterparts, as well as to the logical reasoning between the different elements constituting the digital twin. It is in this context that an approach based on digital twins and ontologies is proposed. The originality of this paper lies in two important points: the first is the exploitation of the expressiveness and reasoning capabilities of ontologies to solve cyber-physical interoperability problems at the digital twin level, while the second is the automation of the whole maintenance process and its decision making key points using the inference potentialities of ontologies. The applicability and effectiveness of the proposed approach is validated through an industrial case of study.**

Keywords—Maintenance systems; maintenance policy; digital twin; reasoning; ontologies; automation; cyber-physical interoperability; decision making; artificial intelligence

I. Introduction

Mastering the maintenance process of industrial systems has become a necessity for companies, in order to pursue the continuous growth of competitive markets. The achievement of this goal will allow manufacturers to optimize the productivity and quality of their production systems, and therefore gain in terms of costs, quality and delays. To this end, industrials and researchers have started to automate this key process, but until now a complete automation is not yet achieved [1]. That justifies the first objective of this paper, which is the development of a new global approach for the maintenance process automation (MMSDTO), from the data collection phase to the establishment of maintenance plans. This operation will be based on a very important artificial intelligence tool which is the Digital Twin. In fact, it will allow collecting the necessary information related to the fields, to follow the production process in real time and to locate and predict failures in the machines. However, the use of this technology requires the resolution of the cyber-physical interoperability problem, which has become the focus of many research works. Therefore, this point constitutes the second objective to be achieved through this paper. Consequently, a new concept will be integrated in the approach, namely ontologies. In fact, their expressiveness and reasoning capabilities will be exploited to preserve the semantics of the large quantities of data exchanged between the physical and virtual spaces of the Digital Twin, to overcome the problem of interoperability and also to make it able to do the logical reasoning and generate the desired results.

Thus, the first part of this paper is a literature review containing the different concepts and key points related to the work realized, notably the Digital Twin and ontologies, as well as the limitations of previous research works and the problems that need to be overcome. Then, the operating system of the newly developed MMSDTO approach will be presented and explained. The different steps of the proposed methodology will be detailed in the following sections. Finally, the approach will be applied on an industrial case study to validate its reliability and efficiency.

II. Related Work

The digital twin is a concept that has recently been the focus of several research studies [2], especially in relation to Industry 4.0 [3]. Its appearance dates back to 2003, when Michael Grieves and John Vickers participated in a conference on product life cycle management [4]. At this event, they presented the Digital Twin as a mirror space model that is used to represent physical entities in a virtual space [5]. In fact, the

Digital Twin numerically reproduces the operation and behavior [6] in real time [7] of physical elements. Therefore, several decisions will be taken in order to optimize the production system and its productivity [6]. This justifies the proposition of the digital twin's standard structure given by Michael Grieves and John Vickers. In fact, they proposed to generalize the structure (physical entity, virtual counterpart, connection between physical and virtual spaces) on digital twins [8]. Afterwards, this structure was extended, and thus, two other elements (services and digital twin data model) were added [9] to make the structure more complete and efficient. Then, this structure was projected on the production workshops by [10]. They proposed a conceptual model with four elements [11], namely:

- Physical Shop-floor: it consists of production lines, production materials and tools, products and employees [10].
- Virtual Shop-floor: it is a virtual reproduction of the functioning of the physical shop-floor and the behavior of its elements [11].
- Shop-floor Service System: it is a set of computer tools (information systems, computer aided tools, etc.) that form the services necessary to execute the commands preventing from the physical and virtual spaces [10].
- Shop-floor Digital Twin Data: data collected from physical and virtual spaces, as well as information generated from the methods of modelization, optimization and prediction of the service space are integrated [10].

The digital twins have been used in different domains such as design [12], logistics management [13], production management [14], maintenance [15], etc. Several works have focused on maintenance. The author [16] realized a literature review on the different papers produced on digital twins for maintenance. Some of these works such as [17], [18] and [19], propose the use of digital twins to do specific maintenance tasks. There are also more general approaches to predict the asset state, in order to predict accordingly the corresponding maintenance plan [20], [21], [22], etc.

By analyzing these research works, we can notice that they have two key limitations: the first one is the cyber-physical interoperability problem of digital twins. In fact, all these works propose approaches for maintenance optimization using digital twins, but they do not mention how to establish the connection between the two physical and virtual spaces of the digital twin, which is considered as a major problem to overcome. The second limitation is that none of these proposed approaches address or automate all the essential points of the maintenance process at once. It is necessary that the proposed approach be global and treat these essential points, in particular: automatic data management, real-time failure detection, prediction of future failures, automation of the maintenance policy choice, optimization of corrective and preventive maintenance, establishment of maintenance plans, automation of decision making etc.

III. The Global Proposed MMSDTO Methodology

The principal goal of this paper is to automate and optimize the maintenance management of manufacturing processes. Therefore, a structured methodology based on digital twins (DT) and ontologies is proposed. In fact, these two concepts are merged and integrated in the field of maintenance, which gives birth to MMSDTO (Maintenance Management System based on Digital Twins and Ontologies) methodology.

It is decomposed into two main phases as indicated in Fig. 1, namely: Construction phase and operation phase.

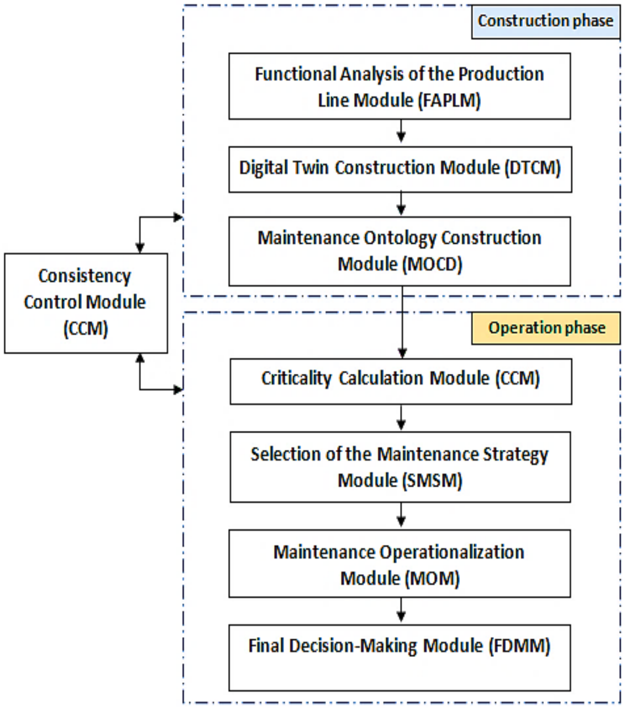

Construction phase
Functional Analysis of the Production Line Module (FAPLM)
Digital Twin Construction Module (DTCM)
Maintenance Ontology Construction Module (MOCD)
Consistency Control Module (CCM)
Operation phase
Criticality Calculation Module (CCM)
Selection of the Maintenance Strategy Module (SMSM)
Maintenance Operationalization Module (MOM)
Final Decision-Making Module (FDMM)

Fig. 1. The Global Proposed MMSDTO Methodology.

First, the construction phase contains three modules:

- Functional Analysis of the Production Line Module (FAPLM): the good knowledge of the production system is an essential factor to have a successful study, for this reason this first module of the construction phase represents one of the pillars of the MMSDTO methodology. Its usefulness lies in the fact that it collects all the necessary information (blocks, components and sub-components of the production lines, production parameters, etc.).
- Digital Twin construction module (DTCM): In this module, the Digital Twin will be constructed with consideration of its standard structure [9]. In fact, a virtual counterpart will be established from the physical entity studied. Moreover, the cyber-physical connection between these two spaces, serving for the transfer of the collected data, the deduced information and the necessary services for the functioning of the system, will be realized.

- Maintenance Ontology Construction Module (MOCD): Through this last module of the construction phase, the expressiveness capacity that ontologies possess will be exploited to build a maintenance ontology (DTM-Onto). The available data will be expressed in a standard language, which will allow converting them to a semantic model that will be used to achieve interoperability between the different elements of the DT.

The operation phase contains four modules:

- Criticality Calculation Module (CCM): The DTM-Onto previously constructed will be enriched by rules of computation and classification of the criticalities of the different elements of the studied system. This will be used to highlight the critical elements of the system to which priority will be given in the maintenance programs.
- Selection of the Maintenance Strategy Module (SMSM): After the analysis of the failure modes of each element as well as the evaluation of the criticality associated to each mode in the previous module, a hierarchization of the different criticality indexes is done in this module, in order to choose the adequate maintenance policy.
- Maintenance Operationalization Module (MOM): The DTMa-Onto will be enriched by other calculation rules to generate the various results and information necessary for the establishment of maintenance plans and planning.
- Final Decision-Making Module (FDMM): Based on the results obtained by the ontologies, maintenance actions must be planned. Thus, maintenance plans and planning will be realized to synthesize the work realized.
- Consistency Control Module (CCM): This module plays a key role in maintaining consistency between the modules of the two main phases of the global methodology.

These modules are executed according to a working process, as shown in Fig. 2.

The first step of the operating system of the proposed MMSDTO methodology is the functional decomposition of the studied machine in several blocks, then in several elements. This will allow to build the Digital Twin of the machine and to feed the ontology with the necessary data and inference rules. In fact, the ontology will help to solve the cyber-physical interoperability problem between the elements of the Digital Twin. In addition, it will generate several computation and classification results: criticalities of the machine elements, TBF, MTBF, Weibull parameters, etc. Finally, all these results will lead to the establishment of global maintenance plans and planning for the production equipment.

All the modules of the methodology will be detailed in the following sections.

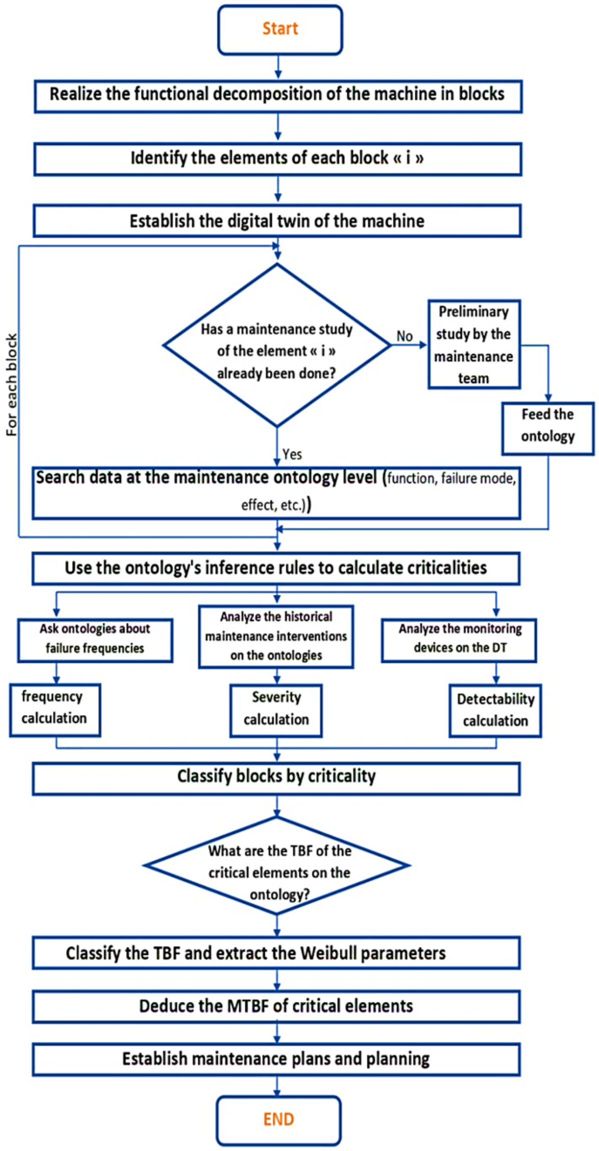

Start
Realize the functional decomposition of the machine in blocks
Identify the elements of each block « i »
Establish the digital twin of the machine
For each block
Has a maintenance study of the element « i » already been done?
No
Preliminary study by the maintenance team
Feed the ontology
Yes
Search data at the maintenance ontology level (function, failure mode, effect, etc.))
Use the ontology's inference rules to calculate criticalities
Ask ontologies about failure frequencies
Analyze the historical maintenance interventions on the ontologies
Analyze the monitoring devices on the DT
frequency calculation
Severity calculation
Detectability calculation
Classify blocks by criticality
What are the TBF of the critical elements on the ontology?
Classify the TBF and extract the Weibull parameters
Deduce the MTBF of critical elements
Establish maintenance plans and planning
END

Fig. 2. The Working Process of the MMSDTO System.

IV. Functional Analysis of the Production Line Module

This module consists in identifying the components of the production system studied as well as the functions they must ensure. Therefore, it is a question of identifying the functions performed by the system and the technological solutions that achieve these functions. To do this, the following elementary phases are implemented:

- A decomposition of the system into blocks, components and sub-components in an exhaustive or limited way, up to the desired level of decomposition according to the needs and the objectives of the study.
- A description of the functions performed by each element.
- A preliminary identification of the dependencies (or cause & effect) between these functions.

V. Digital Twin Construction Module

The main objective of this module is the construction of the Digital Twin which requires the validation of the five elements of its standard structure, namely: Physical entity, virtual model, connection model, services and DT data model: [9].

Firstly, a passage from the functional decomposition of the production line module is crucial, at the level of which the latter is decomposed into several blocks, the blocks into components and the components into sub-components. In fact, this decomposition serves to simplify the virtual reproduction of the functioning of the physical entities, as well as to identify the zones that must be reinforced by sensors to achieve a perfect similarity between the physical and virtual spaces, to increase the detection of failures and to ensure a good follow-up of the production and maintenance. On the other hand, on the virtual level, the components of the manufacturing system are represented by geometrical models in CAD software. In addition, the flows (production flows, logistic flows, etc.) and the behavioral models (fatigue, elasticity, etc.) are simulated respectively on flow and behavioral modeling software.

At this stage, and to ensure a faithful exchange of data and services between the two spaces, a cyber-physical connection must be established. Normally, this connection is achieved using artificial intelligence tools and monitoring information systems [23] [24], but this still has some shortcomings in the industrial context, namely: the difficulty of preserving the semantics of the transmitted data, the inability to transfer a considerable quantity of information between the different actors of the system and the difficulty of logical reasoning. So, to overcome these problems, the concept of ontologies is integrated. This concept has already been integrated in one of our previous paper [23], through the construction of a production ontology that solves the problem of interoperability between physical entities and their virtual counterparts. In this paper, another maintenance ontology (DTMa-Onto) is added. The DTMa-Onto will be enriched with inference rules (maintenance rules, prediction rules, optimization rules, etc.), in order to be able to establish, at the end, the maintenance plans and planning.

In fact, both ontologies will be fed with the necessary data. This data will be stored in the ontologies, processed by the ontology inference processors, and transferred to the virtual space. Afterwards, new information can be generated.

This information will also be stored and processed by the ontologies. The cycle repeats itself in order to control and master the production and maintenance.

The operating system of the Digital Twin is clearly schematized in Fig. 3.

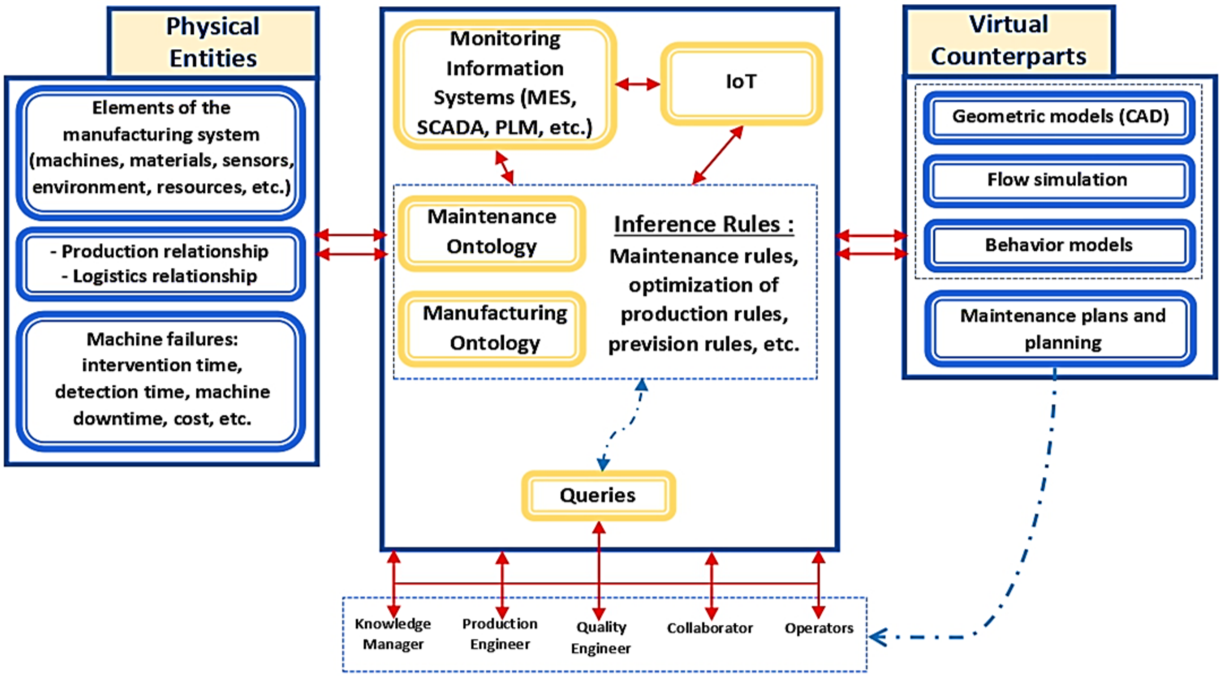

Physical Entities
Elements of the manufacturing system (machines, materials, sensors, environment, resources, etc.)
- Production relationship
- Logistics relationship
Machine failures: intervention time, detection time, machine downtime, cost, etc.
Monitoring Information Systems (MES, SCADA, PLM, etc.)
IoT
Maintenance Ontology
Manufacturing Ontology
Inference Rules : Maintenance rules, optimization of production rules, prevision rules, etc.
Queries
Virtual Counterparts
Geometric models (CAD)
Flow simulation
Behavior models
Maintenance plans and planning
Knowledge Manager
Production Engineer
Quality Engineer
Collaborator
Operators

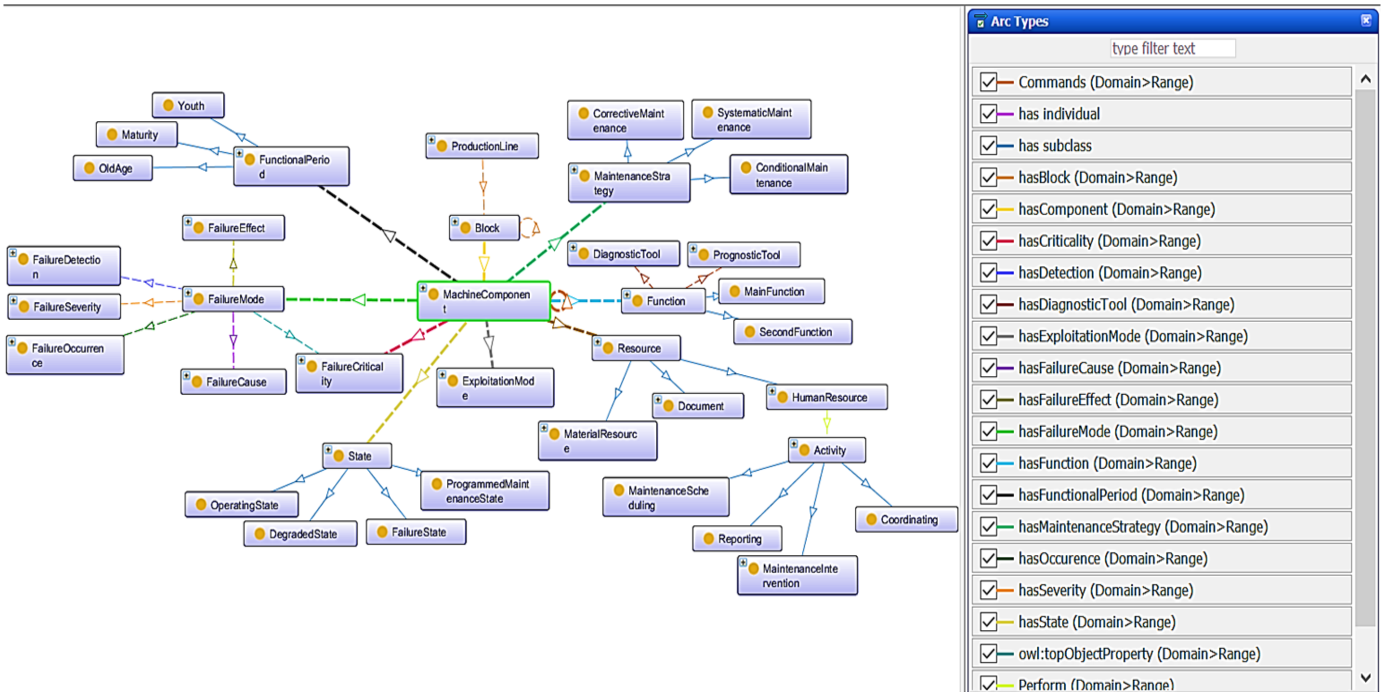
Fig. 3. The Operating System of the Digital Twin is Clearly Schematized in Figure 3.

VI. Maintenance Ontology Construction Module

This module represents the starting point for the implementation of the later phases.

At this level, a maintenance ontology (DTMa-Onto) will be constructed, as shown in Fig. 4, which will solve a large part of the problem of interoperability between the two physical and virtual spaces of the Digital Twin, due to its capacity to exchange a large quantity of data between people and/or machines, to analyze the information exchanged and to reuse it [25].

This will help to reproduce the operation of the production system digitally, as well as detect anomalies and report failures. In fact, the construction of the DTMa-Onto recognizes that it must pass through several main stages and that are realized on the editor of the ontologies Protégé.



☑ Commands (Domain>Range)
☑ has individual
☑ has subclass
☑ hasBlock (Domain>Range)
☑ hasComponent (Domain>Range)
☑ hasCriticality (Domain>Range)
☑ hasDetection (Domain>Range)
☑ hasDiagnosticTool (Domain>Range)
☑ hasExploitationMode (Domain>Range)
☑ hasFailureCause (Domain>Range)
☑ hasFailureEffect (Domain>Range)
☑ hasFailureMode (Domain>Range)
☑ hasFunction (Domain>Range)
☑ hasFunctionalPeriod (Domain>Range)
☑ hasMaintenanceStrategy (Domain>Range)
☑ hasOccurence (Domain>Range)
☑ hasSeverity (Domain>Range)
☑ hasState (Domain>Range)
☑ owl:topObjectProperty (Domain>Range)
☑ Perform (Domain>Range)

Fig. 4. The Constructed DTMa-Onto.

A. Class Definition

This phase consists in the determination of the ensemble of individuals in a specific domain. The domain considered in our case is the maintenance domain. In this way, these classes are divided into three categories:

- Classes related to the production line: they contain the different blocks, components and subcomponents of the machine, the operating mode as well as the human and material resources necessary for the operation of the production line, etc.
- Classes related to the maintenance of the production line: they comprise the failure modes, the human and material resources necessary for diagnosis and maintenance, the operating period, etc.
- Classes related to the choice of maintenance strategies: they include the different types of possible maintenance (corrective maintenance, preventive maintenance, etc.).

The whole set of classes is represented in Fig. 5.

B. Object and Data Properties Definition

First, this step consists of defining the object properties. In other words, the relationships between classes and individuals must be determined, while defining their domains, ranges and inverse properties.

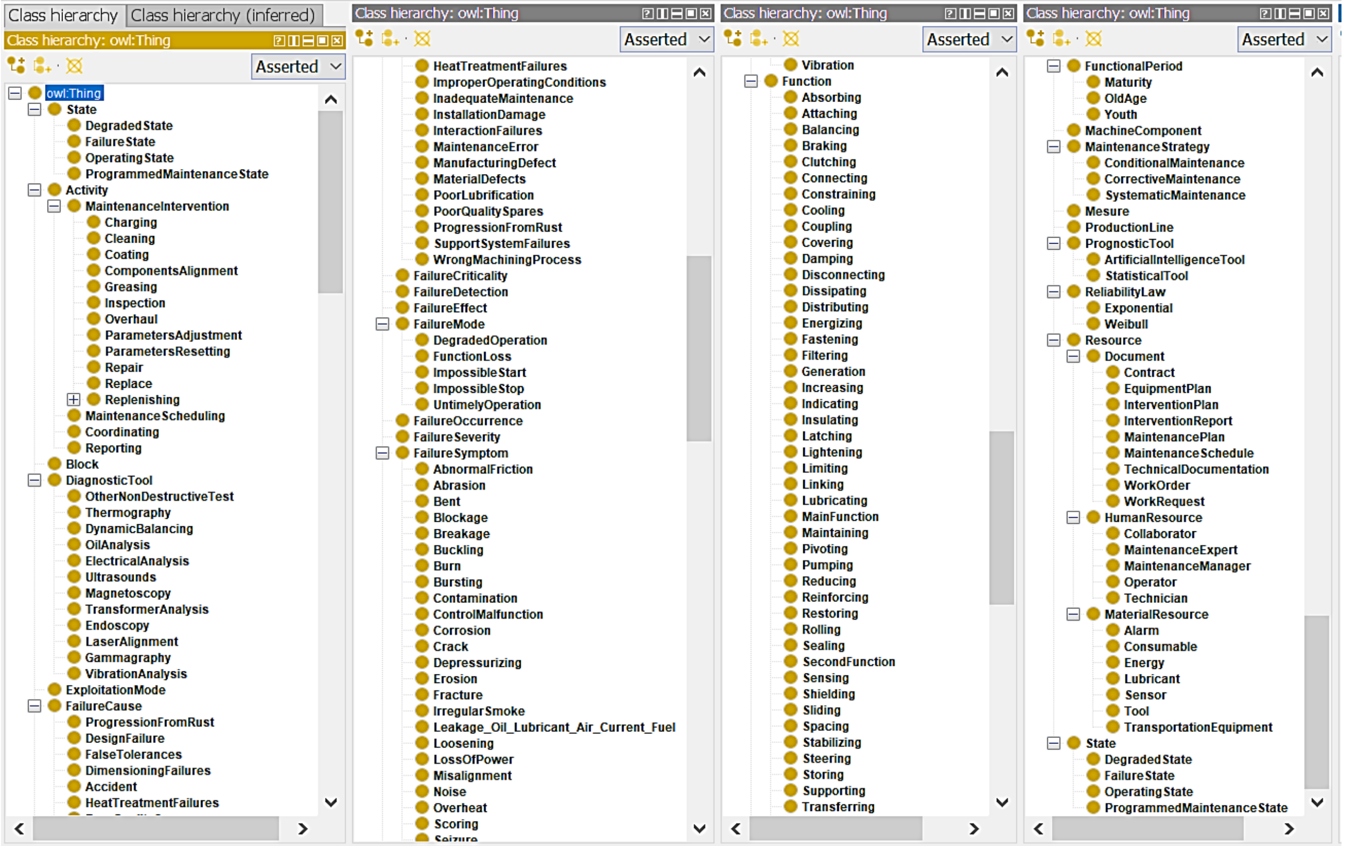
Secondly, we need to specify the data properties. These relate a predicate to a single subject to an attribute data form, which can be a string, an integer, a real number, a date, etc. Fig. 6 recapitulates all the object properties and the data properties of the DTMa-Onto.

It should be noted that the implementation of all these steps leads to a hierarchy of the classes and subclasses of the production line, the maintenance of its different parts and the choice of the appropriate maintenance strategy for each situation.

The particularity of the proposed DTMa-Onto resides in the fact that it is standard and can be adapted to the maintenance of any industrial manufacturing process.

In order to evaluate the consistency of the developed ontology and to execute forward the reasoning rules, we have used the description logic reasoner Pellet [25] which is integrated in the open-source ontology editor Protégé 5.0.

In the next section, the expressiveness of the OWL ontology DTMa-Onto will be enhanced with three different categories of reasoning rules modeled with SWRL Language (Semantic Web Rule Language).



- owl:Thing
  - State
    - DegradedState
    - FailureState
    - OperatingState
    - ProgrammedMaintenanceState
  - Activity
    - MaintenanceIntervention
      - Charging
      - Cleaning
      - Coating
      - ComponentsAlignment
      - Greasing
      - Inspection
      - Overhaul
      - ParametersAdjustment
      - ParametersResetting
      - Repair
      - Replace
      - Replenishing
    - MaintenanceScheduling
    - Coordinating
    - Reporting
  - Block
  - DiagnosticTool
    - OtherNonDestructiveTest
    - Thermography
    - DynamicBalancing
    - OilAnalysis
    - ElectricalAnalysis
    - Ultrasounds
    - Magnetoscopy
    - TransformerAnalysis
    - Endoscopy
    - LaserAlignment
    - Gammagraphy
    - VibrationAnalysis
  - ExploitationMode
  - FailureCause
    - ProgressionFromRust
    - DesignFailure
    - FalseTolerances
    - DimensioningFailures
    - Accident
    - HeatTreatmentFailures

Class hierarchy: owl:Thing

- HeatTreatmentFailures
- ImproperOperatingConditions
- InadequateMaintenance
- InstallationDamage
- InteractionFailures
- MaintenanceError
- ManufacturingDefect
- MaterialDefects
- PoorLubrification
- PoorQualitySpares
- ProgressionFromRust
- SupportSystemFailures
- WrongMachiningProcess
- FailureCriticality
- FailureDetection
- FailureEffect
- FailureMode
  - DegradedOperation
  - FunctionLoss
  - ImpossibleStart
  - ImpossibleStop
  - UntimelyOperation
- FailureOccurrence
- FailureSeverity
- FailureSymptom
  - AbnormalFriction
  - Abrasion
  - Bent
  - Blockage
  - Breakage
  - Buckling
  - Burn
  - Bursting
  - Contamination
  - ControlMalfunction
  - Corrosion
  - Crack
  - Depressurizing
  - Erosion
  - Fracture
  - IrregularSmoke
  - Leakage_Oil_Lubricant_Air_Current_Fuel
  - Loosening
  - LossOfPower
  - Misalignment
  - Noise
  - Overheat
  - Scoring
  - Seizure

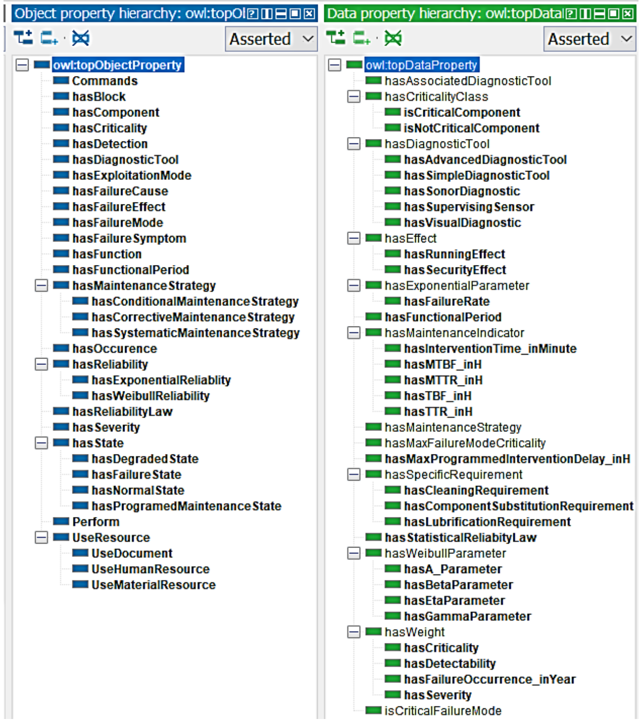
Class hierarchy: owl:Thing

- Vibration
- Function
  - Absorbing
  - Attaching
  - Balancing
  - Braking
  - Clutching
  - Connecting
  - Constraining
  - Cooling
  - Coupling
  - Covering
  - Damping
  - Disconnecting
  - Dissipating
  - Distributing
  - Energizing
  - Fastening
  - Filtering
  - Generation
  - Increasing
  - Indicating
  - Insulating
  - Latching
  - Lightening
  - Limiting
  - Linking
  - Lubricating
  - MainFunction
  - Maintaining
  - Pivoting
  - Pumping
  - Reducing
  - Reinforcing
  - Restoring
  - Rolling
  - Sealing
  - SecondFunction
  - Sensing
  - Shielding
  - Sliding
  - Spacing
  - Stabilizing
  - Steering
  - Storing
  - Supporting
  - Transferring

Class hierarchy: owl:Thing

- FunctionalPeriod
  - Maturity
  - OldAge
  - Youth
- MachineComponent
- MaintenanceStrategy
  - ConditionalMaintenance
  - CorrectiveMaintenance
  - SystematicMaintenance
- Mesure
- ProductionLine
- PrognosticTool
  - ArtificialIntelligenceTool
  - StatisticalTool
- ReliabilityLaw
  - Exponential
  - Weibull
- Resource
  - Document
    - Contract
    - EquipmentPlan
    - InterventionPlan
    - InterventionReport
    - MaintenancePlan
    - MaintenanceSchedule
    - TechnicalDocumentation
    - WorkOrder
    - WorkRequest
  - HumanResource
    - Collaborator
    - MaintenanceExpert
    - MaintenanceManager
    - Operator
    - Technician
  - MaterialResource
    - Alarm
    - Consumable
    - Energy
    - Lubricant
    - Sensor
    - Tool
    - TransportationEquipment
- State
  - DegradedState
  - FailureState
  - OperatingState
  - ProgrammedMaintenanceState

Fig. 5. All Classes of the DTMa-Onto.



Object property hierarchy: owl:topOI
Asserted
Data property hierarchy: owl:topData
Asserted

- owl:topObjectProperty
  - Commands
  - hasBlock
  - hasComponent
  - hasCriticality
  - hasDetection
  - hasDiagnosticTool
  - hasExploitationMode
  - hasFailureCause
  - hasFailureEffect
  - hasFailureMode
  - hasFailureSymptom
  - hasFunction
  - hasFunctionalPeriod
  - hasMaintenanceStrategy
    - hasConditionalMaintenanceStrategy
    - hasCorrectiveMaintenanceStrategy
    - hasSystematicMaintenanceStrategy
  - hasOccurence
  - hasReliability
    - hasExponentialReliablity
    - hasWeibullReliability
  - hasReliabilityLaw
  - hasSeverity
  - hasState
    - hasDegradedState
    - hasFailureState
    - hasNormalState
    - hasProgramedMaintenanceState
  - Perform
  - UseResource
    - UseDocument
    - UseHumanResource
    - UseMaterialResource

- owl:topDataProperty
  - hasAssociatedDiagnosticTool
  - hasCriticalityClass
    - isCriticalComponent
    - isNotCriticalComponent
  - hasDiagnosticTool
    - hasAdvancedDiagnosticTool
    - hasSimpleDiagnosticTool
    - hasSonorDiagnostic
    - hasSupervisingSensor
    - hasVisualDiagnostic
  - hasEffect
    - hasRunningEffect
    - hasSecurityEffect
  - hasExponentialParameter
    - hasFailureRate
  - hasFunctionalPeriod
  - hasMaintenanceIndicator
    - hasInterventionTime_inMinute
    - hasMTBF_inH
    - hasMTTR_inH
    - hasTBF_inH
    - hasTTR_inH
  - hasMaintenanceStrategy
  - hasMaxFailureModeCriticality
  - hasMaxProgrammedInterventionDelay_inH
  - hasSpecificRequirement
    - hasCleaningRequirement
    - hasComponentSubstitutionRequirement
    - hasLubrificationRequirement
  - hasStatisticalReliabilityLaw
  - hasWeibullParameter
    - hasA_Parameter
    - hasBetaParameter
    - hasEtaParameter
    - hasGammaParameter
  - hasWeight
    - hasCriticality
    - hasDetectability
    - hasFailureOccurrence_inYear
    - hasSeverity
  - isCriticalFailureMode

Fig. 6. Object Properties and Data Properties of the DTMa-Onto.

VII. Criticality Calculation Module

After constructing the backbone of the DTMa-Onto through the definition of classes, object properties and data properties, now it needs to be enriched with inference rules in order to perform the computation and the logical reasoning. Therefore, a first category of rules will be executed in this module, namely the rules for calculating and classifying the criticality of components. First of all, the criticality evaluation is based on the calculation and estimation of:

- The Frequency of Occurrence Index (O) (Rules R1-R4): it represents the probability that the cause of failure appears and generates the failure mode considered.
- The Severity Index (S) (Rules S1-S4): it quantifies the severity of the consequences that the failure generates.
- The Non-Detection index (D) (Rules D1-D6): it represents the probability of non-detection of the failure mode.

The criticality index (Rule C1) is calculated by multiplying the three elementary indices for each component:

$$C = O * S * D \tag{1}$$

In addition to that, the evaluation is done taking into account the current or expected state of the system, which allows to prioritize the failure modes and to identify the most critical ones to study in priority.

In addition to that, the evaluation is done taking into account the current or expected state of the system, which allows to prioritize the failure modes and to identify the most critical ones to study in priority.

As Table I shows, the calculation of these indices is formalized in the form of several rules expressed using the SWRL language.

TABLE I. RULES OF CRITICALITIES CALCULATION AND CLASSIFICATION EXPRESSED IN SWRL LANGUAGE

Criticality Rules
R1 : MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^hasFailureOccurrence_inYear(?M, ?o) ^ swrlb:greaterThanOrEqual(?o, 12) -> hasOccurrence(?M, 4)
R2 : MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^hasFailureOccurrence_inYear(?M, ?o) ^ swrlb:greaterThanOrEqual(?o, 4) ^ swrlb:lessThanorEqual(?o, 12) -> hasOccurrence(?M, 3)
R3 : MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^hasFailureOccurrence_inYear (?M, ?o) ^ swrlb:greaterThan(?o, 1) ^ swrlb:lessThanorEqual(?o, 4) -> hasOccurrence(?M, 2)
R4 : MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^hasFailureOccurrence_inYear(?M, ?o) ^ swrlb:lessThanOrEqual(?o, 1) -> hasOccurrence(?M, 1)
S1 : MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^hasInterventionTime_inMinute(?M, ?IT) ^ swrlb:greaterThan(?IT, 60) -> hasSeverity(?M, 4)
S2 : MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^hasInterventionTime_inMinute(?M, ?IT) ^swrlb:greaterThan(?IT, 20) ^ swrlb:lessThan(?IT,60) -> hasSeverity(?M, 3)
S3 : MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^hasInterventionTime_inMinute(?M, ?IT) ^swrlb:greaterThan(?IT, 5) ^ swrlb:lessThan(?IT, 20) -> hasSeverity(?M, 2)
S4 : MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^hasInterventionTime_inMinute(?M, ?IT)^swrlb:lessThan(?IT,5) -> hasSeverity(?M, 1)
D1 : MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^ hasSupervisingSensor(?M, 0) ^ hasVisualDiagnostic(?M, 0)^hasSonorDiagnostic(?M,0)^hasSimpleDiagnosticTool(?M,0) ^hasAdvancedDiagnosticTool(?M,0) -> hasDetectability(?M, 4)
D2 : MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^ hasSupervisingSensor(?M, 0) ^ hasVisualDiagnostic(?M, 0)^hasSonorDiagnostic(?M,0)^hasSimpleDiagnosticTool(?M,0) ^hasAdvancedDiagnosticTool(?M,1) -> hasDetectability(?M, 4)
D3 : MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^ hasSupervisingSensor(?M, 0) ^ hasVisualDiagnostic(?M, 0)^hasSonorDiagnostic(?M,0)^hasSimpleDiagnosticTool(?M,1) ^hasAdvancedDiagnosticTool(?M,0) -> hasDetectability(?M, 3)
D4 : MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^ hasSupervisingSensor(?M, 0) ^ hasVisualDiagnostic(?M, 0)^hasSonorDiagnostic(?M,1)^hasSimpleDiagnosticTool(?M,0) ^hasAdvancedDiagnosticTool(?M,0) -> hasDetectability(?M, 2)
D5 : MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^ hasSupervisingSensor(?M, 0) ^ hasVisualDiagnostic(?M, 1)^hasSonorDiagnostic(?M,0)^hasSimpleDiagnosticTool(?M,0) ^hasAdvancedDiagnosticTool(?M,0) -> hasDetectability(?M, 1)
D6 : MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^ hasSupervisingSensor(?M, 1) ^ hasVisualDiagnostic(?M, 0)^hasSonorDiagnostic(?M,0)^hasSimpleDiagnosticTool(?M,0) ^hasAdvancedDiagnosticTool(?M,0) -> hasDetectability(?M, 1)
C1: MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^ hasOccurrence(?M, ?o) ^ hasSeverity(?M, ?s) ^ hasDetectability(?M, ?d) ^ swrlb:multiply(?q, ?o, ?s, ?d) -> hasCriticality(?M, ?q)
C2: MachineComponent(?C) ^ hasMaxFailureModeCriticality(?C, ?q) ^ swrlb:greaterThanOrEqual(?q, 32) -> hasCriticalityClass(?C, "Critical")
C3: MachineComponent(?C) ^ hasMaxFailureModeCriticality(?C, ?q) ^ swrlb:lessThan(?q, 32) -> hasCriticalityClass(?C, "NotCritical")
Q1: MachineComponent(?C) ^ hasFailureMode(?C, ?M) ^ hasCriticality(?M, ?q) ^ hasCriticalityClass (?C, ?s) -> sqwrl:select (?C , ?M , ?q, ?s) ^ sqwrl:columnNames ("MachineComponent", "FailureMode", "Failure Mode Criticality", "Component Criticality Class")
Q2: MachineComponent(?C) ^ hasCriticalityClass(?C, ?s) -> sqwrl:select(?C, ?s) ^ sqwrl:columnNames("MachineComponent", "Component Criticality Class")

In fact, the Digital Twin collects the necessary data (i.e. downtime, failures appeared, failure frequency, etc.), stores them on the DTMa-Onto. In its turn, the DTMa-Onto treats this collected information, and assigns for each index an adequate value according to a rating scale (from 1 to 4) programmed by the inference rules, and then the criticality of each component is calculated. It should be noted that the values as well as the rating criteria can change from one production process to another, but the principle remains the same.

A deployment of the results of the computation is performed. The final objective is to define and launch all the necessary actions, both corrective and preventive, taking into account the priorities highlighted by the evaluation of the failures' criticality. Indeed, we proceed to a prioritization of all the failure modes according to their criticality indexes (The rules: C2, C3, Q1, Q2). The critical points of the equipment are then identified. They correspond to the failures which have a criticality higher than a threshold predefined by the maintenance team and which takes into account the expected reliability objectives as well as the studied technologies. Priority actions must also be considered for any severity or occurrence index score equal to 4, because, even if the criticality of these failures is lower than the pre-established threshold, they represent a real risk. This constraint has been taken into account in the rules.

VIII. SELECTION OF THE MAINTENANCE STRATEGY MODULE

Fig. 7 summarizes the methodology for choosing the maintenance policy proposed.

The inference rules implemented in the criticality calculation module have led to the establishment of other rules concerning the choice of the maintenance policy.

In fact, to make the analysis of the failure modes of each component and the evaluation of the criticality associated with each mode useful, it is necessary to define for each component the type of maintenance that is appropriate. To do this, the maintenance team determines a minimum criticality threshold above which the failure modes become critical, and then the corrective maintenance is replaced by the preventive one.

However, in some cases, it is difficult to control the equipment even if the criticality index exceeds the threshold set by the working group, and therefore, its maintenance remains corrective. In addition to this, the safety of the personnel and the equipment must be taken into account.

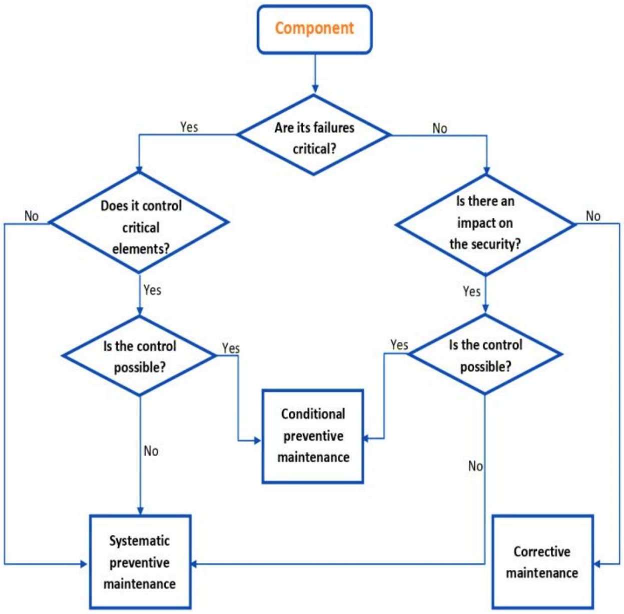

Component
Are its failures critical?
Yes
No
Does it control critical elements?
Is there an impact on the security?
No
No
Yes
Yes
Is the control possible?
Yes
Yes
Is the control possible?
Conditional preventive maintenance
No
No
Systematic preventive maintenance
Corrective maintenance

Fig. 7. The Proposed Methodology for the Maintenance Policy Selection.

This approach for selecting the maintenance strategy is programmed at the DTMa-Onto level using various rules expressed by the SWRL language. Table II groups all these rules.

TABLE II. MAINTENANCE STRATEGY RULES EXPRESSED IN SWRL LANGUAGE

Maintenance Strategy Rules
P1: MachineComponent (?C) ^ hasCriticalityClass(?C, "Critical") ^ Commands(?C, NoneComponent) -> hasMaintenanceStrategy(?C, "SystematicMaintenance")
P2: MachineComponent (?C) ^ MachineComponent (?N) ^ hasCriticalityClass(?C, "Critical") ^ Commands(?C, ?N) ^ hasCriticalityClass(?N, "NotCritical") ^ hasAssociatedDiagnosticTool(?C, 0) -> hasMaintenanceStrategy(?C, "SystematicMaintenance")
P3 : MachineComponent (?C) ^ MachineComponent (?N) ^ hasCriticalityClass(?C, "Critical") ^ Commands(?C, ?N) ^ hasCriticalityClass(?N, "NotCritical") ^ hasAssociatedDiagnosticTool(?C, 1) -> hasMaintenanceStrategy(?C, "ConditionalMaintenance")
P4: MachineComponent (?C) ^ hasCriticalityClass(?C, "NotCritical") ^ hasSecurityEffect (?C, 1) ^ hasAssociatedDiagnosticTool(?C, 1) -> hasMaintenanceStrategy(?C, "ConditionalMaintenance")
P5: MachineComponent (?C) ^ hasCriticalityClass(?C, "NotCritical") ^ hasSecurityEffect (?C, 1) ^ hasAssociatedDiagnosticTool(?C, 0) -> hasMaintenanceStrategy(?C, "SystematicMaintenance")
P6: MachineComponent (?C) ^ hasCriticalityClass(?C, "NotCritical") ^ hasSecurityEffect (?C, 0) -> hasMaintenanceStrategy(?C, "CorrectiveMaintenance")

IX. MAINTENANCE OPERATIONALIZATION MODULE

As presented in Table III, this module covers the third class of inference rules of the DTMa-Onto.

These rules are a formalization of the two statistical laws of component reliability, namely:

- The exponential law: where the failure rate is constant. It is valid in the case of electrical components and components in maturity phase.

$$R(t) = exp(-\lambda t) \tag{2}$$

With:

✓ λ is the failure rate. It represents the proportion of defective parts that we obtain during a very short time interval

- The Weibull law: It is valid in the general case and takes into account the three phases of the life cycle (Growth, maturity and decline)

$$R(t) = \exp(-(\frac{t-\gamma}{\eta})^{\beta} \tag{3}$$

With:

✓ β is the shape parameter

✓ η is the scale parameter

✓ γ is the position parameter

TABLE III. MAINTENANCE OPERATIONALIZATION RULES EXPRESSED IN SWRL LANGUAGE

Maintenance Operationalization Rules
W1: MachineComponent(?C) ^ ReliabilityLaw(?R) ^ hasReliabilityLaw (?C, ?R) ^ hasStatisticalReliabityLaw(?R, "Weibull") ^ hasBetaParameter(?R, ?b) ^ hasGammaParameter(?R, ?g) ^ hasEtaParameter(?R, ?e) ^ hasA_Parameter(?R, ?A) ^ swrlb:multiply(?t, ?A, ?e) ^ swrlb:add(?u, ?t, ?g) -> hasMTBF_inH(?C, ?u) ^ hasMaxProgrammedInterventionDelay_inH(?C, ?u)
W2: MachineComponent(?C) ^ hasFunctionalPeriod(?C, "Maturity") -> hasStatisticalReliabilityLaw(?R, "Exponential")
W3: MachineComponent(?C) ^ hasReliabilityLaw(?C, ?R) ^ hasStatisticalReliabilityLaw(?R, "Exponential") ^ hasFailureRate(?R, ?l) ^ swrlb:divide(?u, 1, ?l) -> hasMTBF_inH(?C, ?u) ^ hasMaxProgrammedInterventionDelay_inH(?C, ?u)

This module works in interaction with the numerical analysis environment and the programming language "Matlab". Indeed, using the TBF (Time between failures) extracted from the failure histories collected by the Digital Twin and stored in the DTMa-Onto, the approach developed by [26] will be applied for the estimation of the exponential and Weibull law parameters of the concerned components. These parameters will constitute the inputs for the SWRL rules of the ontology. These rules model the exponential (rules W2-W3) and Weibull (rules W1) reliability laws, and enable the calculation of MTBF (Mean time between failures) and the prediction of the next component failures, which will allow to schedule future maintenance interventions before the machine components fail. The MTBF formula is as follows:

$$R(t) = \exp(-(\frac{t-\gamma}{\eta})^{\beta}) \quad (4)$$

With:

- ✓ β is the shape parameter
- ✓ η is the scale parameter
- ✓ γ is the position parameter

X. Final Decision-Making Module

From the maintenance Operationalization module, the TBFs and MTBFs of the different critical components are calculated, and the life phase of each element is determined. Based on these calculation results obtained by the DTMa-Onto, the appropriate maintenance actions (lubrication, greasing, etc.) must be planned before the end of the TBFs. All this work gives birth to the maintenance plans and planning, which will be stored and archived at the DTMa-Onto level, in order to benefit from the experience feedback for future maintenance situations. This phase has two objectives, namely the identification and standardization of good practices and methods, as well as the transmission to the design all the experience acquired (means, processes, operating modes, etc.). The obvious goals are to capitalize successful actions and generalize them.

XI. Case of Study

The objective of this part is to validate the applicability of the proposed methodology MMSDTO, as well as its effectiveness for the resolution of maintenance problems of industrial production systems. This validation will be done through a concrete example of a company that operates in the agro alimentary industry. In fact, this company works on dairy products manufacturing (milk, yogurt, etc.), but the case study will only focus on the yogurt manufacturing and packaging process.

The first step of the process is the unrolling of the plastic, which is done by two rollers driven by a gear motor. The plastic roll is introduced in the insertion section, and in order to ensure its pecking, it passes directly into a heating system to heat its edges, due to heating resistances and temperature probes. This operation is done by a chain and serves to transmit the plastic strip during the rest of the process. Before moving to the forming block, the plastic strip passes through two ionizing dust collectors. The first one ionizes the plastic strip to prevent electrostatic sticking and to remove any foreign body. The second is a tunnel of ultraviolet lamps, which ionizes the operculum to prevent electrostatic sticking and to aspirate the existing particles. Once the plastic and the operculum are ionized, the product cups are formed using a mold and punches according to the desired specifications. The cups are then dosed with the products, which have already been prepared in the process section, using a piston dosing unit. Then the operculum is welded onto the dosed cups using a cam press and the product packs are cut out. After the product packs have been cut, they are transported to the case packer by a conveyor system to be packed in empty boxes. The last step of the packaging process is the manual pallet handling of the full boxes, and their storage in cold rooms.

The competition in the market is fierce and the interruption of production generates huge losses for the company (market shares, profits, etc.), that's why the control of the production equipment maintenance is a necessity. To this effect, the MMSDTO methodology is applied.

The first step in the approach is the functional decomposition of the production process into blocks, the blocks into components and the components into sub-components. This decomposition will be used, initially, to locate the areas of the production line that require the installation of sensors for the data transfer between the two physical and virtual spaces of the Digital Twin. Fig. 8 and 9 show, respectively, an example of the installation of the position determination sensors and an extract from the virtual part of the Digital Twin.

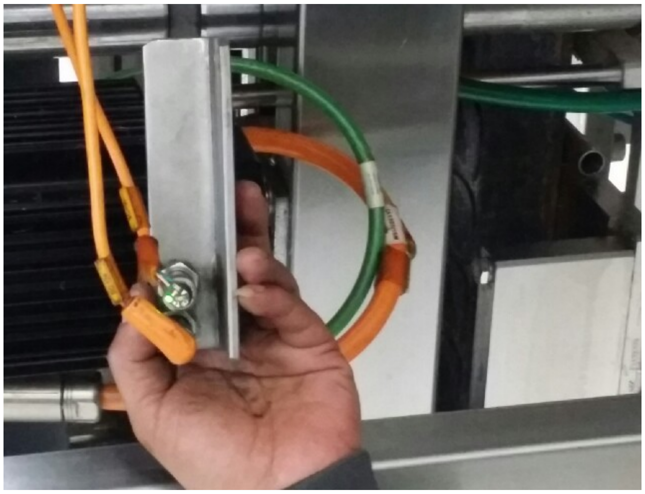
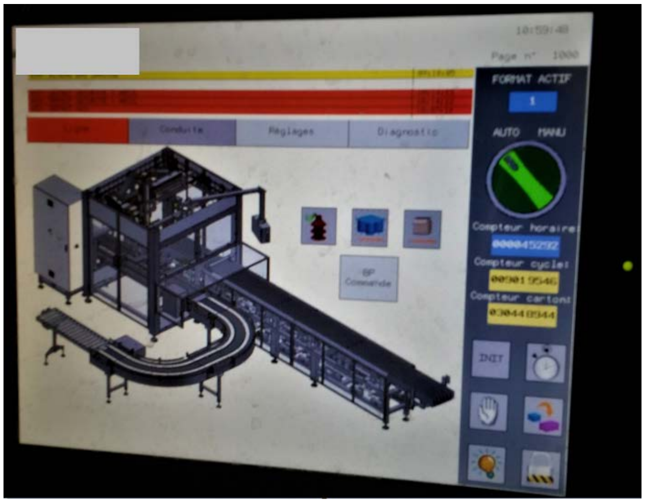


Fig. 8. Example of Sensors Installation Necessary for the Digital Twin Functioning.



FORMAT ACTIF 1

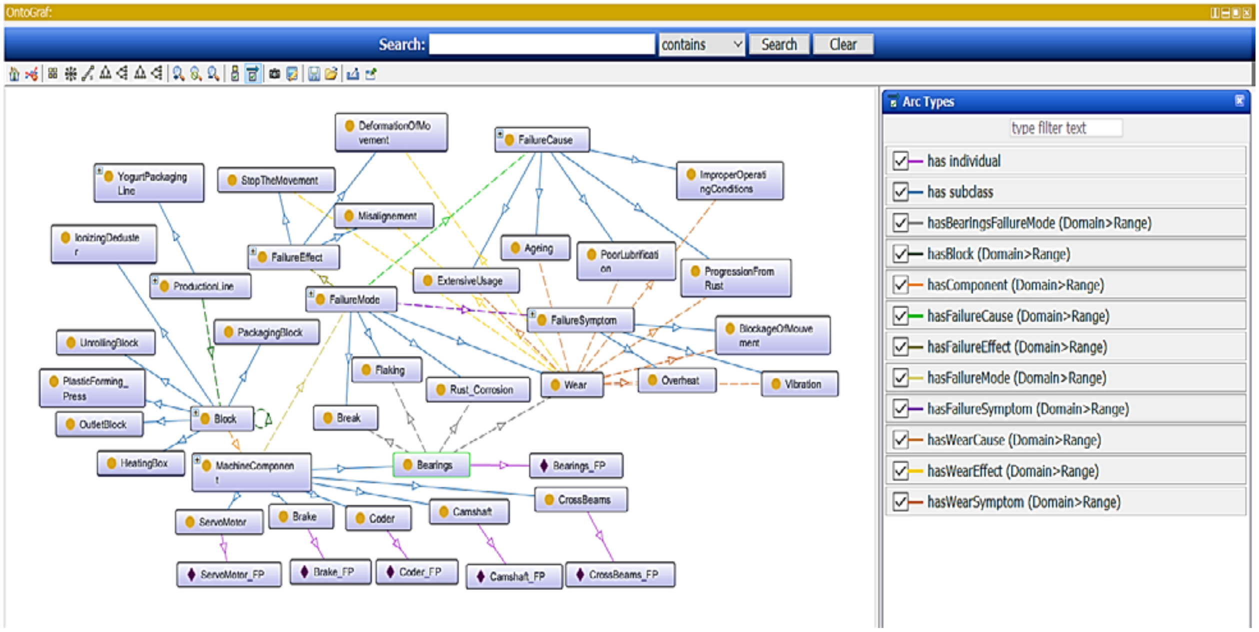
AUTO MANU

Compteur horaire: 000045292

Compteur cycle: 000015546

Compteur carton: 030448944

Fig. 9. An Extract from the Virtual Part of the Digital Twin.



☑ has individual
☑ has subclass
☑ hasBearingsFailureMode (Domain>Range)
☑ hasBlock (Domain>Range)
☑ hasComponent (Domain>Range)
☑ hasFailureCause (Domain>Range)
☑ hasFailureEffect (Domain>Range)
☑ hasFailureMode (Domain>Range)
☑ hasFailureSymptom (Domain>Range)
☑ hasWearCause (Domain>Range)
☑ hasWearEffect (Domain>Range)
☑ hasWearSymptom (Domain>Range)

Fig. 10. The DTMa-Onto related to the Case of Study.

The second objective of this decomposition is the preliminary construction of the DTMa-Onto. Thus, the classes, the object properties and the data properties of the forming press are entered on the ontology (the work will be focused on the forming block to simplify the case of study and make it clearer). In fact, as it is represented on Fig. 10, the DTMa-Onto schematizes the different blocks of the production line, the failure modes of the different components of the forming block (the studied block), their causes, their symptoms and their effects.

Now, based on the collected data from the Digital Twin, in particular the machine downtime tracking, the DTMa-Onto, and due to its information storage and logical reasoning capabilities, assigns for each failure mode a suitable occurrence, severity and detectability value. Then, the DTMa-Onto immediately generates the results of the calculation and classification of the failure modes' criticalities, using the inference rules listed in the Table IV.

The totality of the calculation and criticality classification results of the other failure modes of the forming press components is presented in the Fig. 12.

After the classification of the failure modes by their criticality, the ontology lists all the critical elements based on the criticality threshold set by the maintenance team.

An extract of the obtained results are illustrated in the Fig. 11. These results concern the criticalities of the forming press bearings failure modes.

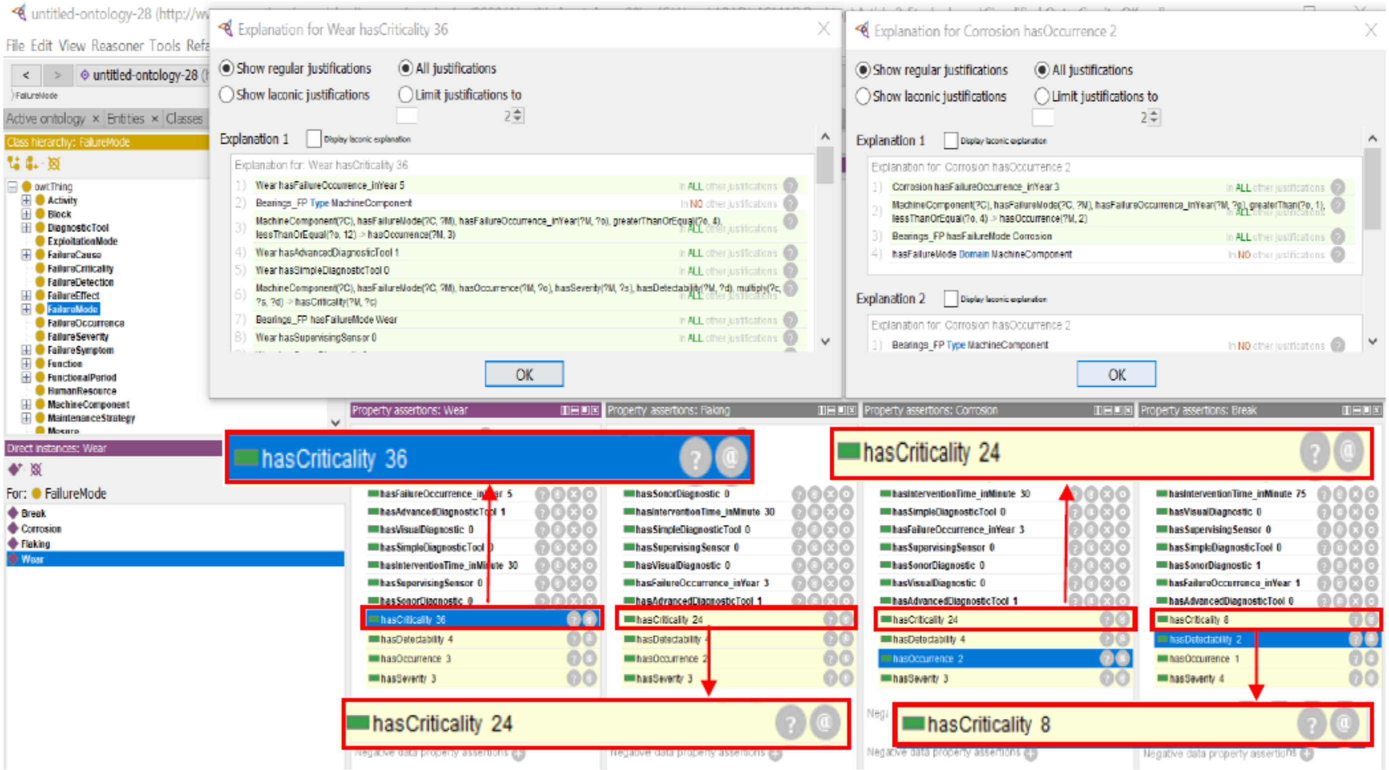
At this stage, the second category of inference rules, namely the rules for the maintenance policy choice, is applied. Thus, the type of maintenance for each component is determined by the DTMa-Onto.

The third category of rules, including maintenance operationalization rules, will be applied on the brake and bearings of the forming press to calculate the proper MTBFs. Fig. 14 shows the results obtained.

The Fig. 13 summarizes the results of the appropriate maintenance policy for each element. The results obtained indicate that the bearings and the brake of the forming press require a systematic preventive maintenance policy.

Finally, from all the results obtained before, it only remains to establish the general maintenance plans and planning, in order to control the maintenance process of the production line, and subsequently improve its performance.

It is necessary to note that the maintenance plans and planning are not presented in order to avoid burdening the study case with too much information and results.



● Show regular justifications ● All justifications
○ Show laconic justifications ○ Limit justifications to 2

Explanation 1 □ Display laconic explanation

Explanation for: Wear hasCriticality 36

1) Wear hasFailureOccurrence_inYear 5 — In ALL other justifications
2) Bearings_FP Type MachineComponent — In NO other justifications
3) MachineComponent(?C), hasFailureMode(?C, ?M), hasFailureOccurrence_inYear(?M, ?o), greaterThanOrEqual(?o, 4), lessThanOrEqual(?o, 12) -> hasOccurrence(?M, 3) — In ALL other justifications
4) Wear hasAdvancedDiagnosticTool 1 — In ALL other justifications
5) Wear hasSimpleDiagnosticTool 0 — In ALL other justifications
6) MachineComponent(?C), hasFailureMode(?C, ?M), hasOccurrence(?M, ?o), hasSeverity(?M, ?s), hasDetectability(?M, ?d), multiply(?c, ?s, ?d) -> hasCriticality(?M, ?c) — In ALL other justifications
7) Bearings_FP hasFailureMode Wear — In ALL other justifications
8) Wear hasSupervisingSensor 0 — In ALL other justifications

● Show regular justifications ● All justifications
○ Show laconic justifications ○ Limit justifications to 2

Explanation 1 □ Display laconic explanation

Explanation for: Corrosion hasOccurrence 2

1) Corrosion hasFailureOccurrence_inYear 3 — In ALL other justifications
2) MachineComponent(?C), hasFailureMode(?C, ?M), hasFailureOccurrence_inYear(?M, ?o), greaterThan(?o, 1), lessThanOrEqual(?o, 4) -> hasOccurrence(?M, 2) — In ALL other justifications
3) Bearings_FP hasFailureMode Corrosion — In ALL other justifications
4) hasFailureMode Domain MachineComponent — In NO other justifications

Explanation 2 □ Display laconic explanation

Explanation for: Corrosion hasOccurrence 2

1) Bearings_FP Type MachineComponent — In NO other justifications

Property assertions: Wear

hasFailureOccurrence_inYear 5
hasAdvancedDiagnosticTool 1
hasVisualDiagnostic 0
hasSimpleDiagnosticTool 0
hasInterventionTime_inMinute 30
hasSupervisingSensor 0
hasSonorDiagnostic 0
hasCriticality 36
hasDetectability 4
hasOccurrence 3
hasSeverity 3

Property assertions: Flaking

hasSonorDiagnostic 0
hasInterventionTime_inMinute 30
hasSimpleDiagnosticTool 0
hasSupervisingSensor 0
hasVisualDiagnostic 0
hasFailureOccurrence_inYear 3
hasAdvancedDiagnosticTool 1
hasCriticality 24
hasDetectability 4
hasOccurrence 2
hasSeverity 3

Property assertions: Corrosion

hasInterventionTime_inMinute 30
hasSimpleDiagnosticTool 0
hasFailureOccurrence_inYear 3
hasSupervisingSensor 0
hasSonorDiagnostic 0
hasVisualDiagnostic 0
hasAdvancedDiagnosticTool 1
hasCriticality 24
hasDetectability 4
hasOccurrence 2
hasSeverity 3

Property assertions: Break

hasInterventionTime_inMinute 75
hasVisualDiagnostic 0
hasSupervisingSensor 0
hasSimpleDiagnosticTool 0
hasSonorDiagnostic 1
hasFailureOccurrence_inYear 1
hasAdvancedDiagnosticTool 0
hasCriticality 8
hasDetectability 2
hasOccurrence 1
hasSeverity 4

hasCriticality 36

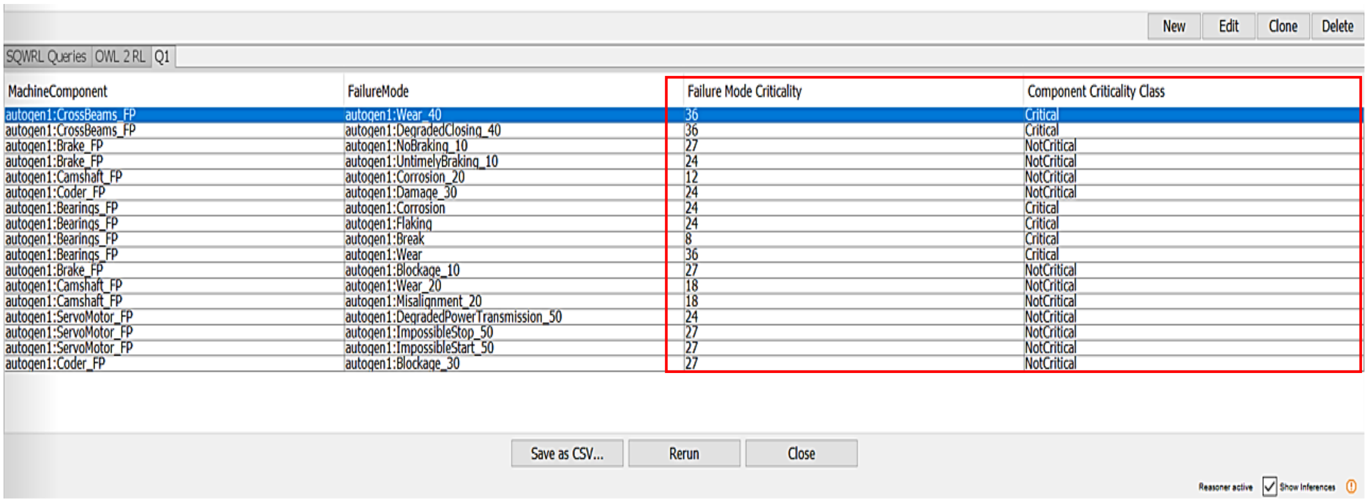
hasCriticality 24

hasCriticality 24

hasCriticality 8

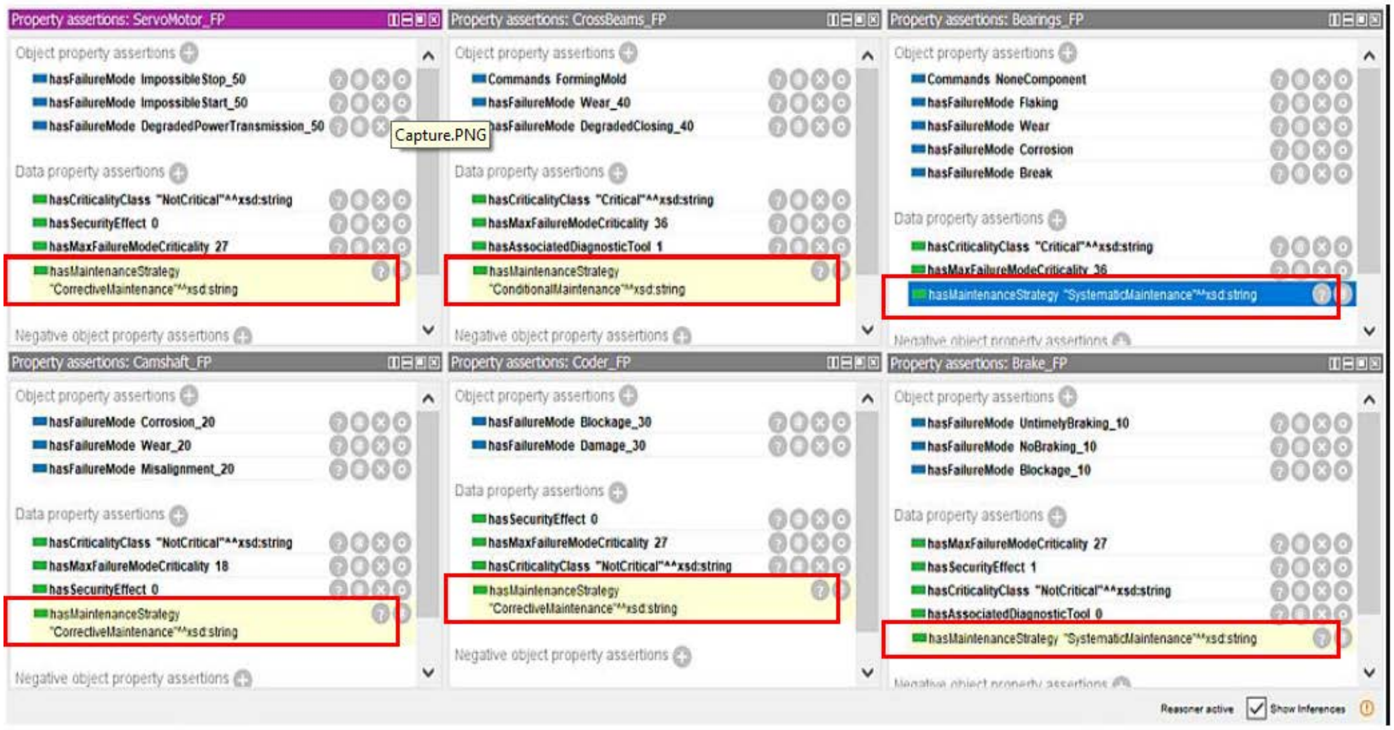
Negative data property assertions

Fig. 11. An Extract of the Criticalities Calculation Results Obtained.



MachineComponent	FailureMode	Failure Mode Criticality	Component Criticality Class
autogen1:CrossBeams_FP	autogen1:Wear_40	36	Critical
autogen1:CrossBeams_FP	autogen1:DegradedClosing_40	36	Critical
autogen1:Brake_FP	autogen1:NoBraking_10	27	NotCritical
autogen1:Brake_FP	autogen1:UntimelyBraking_10	24	NotCritical
autogen1:Camshaft_FP	autogen1:Corrosion_20	12	NotCritical
autogen1:Coder_FP	autogen1:Damage_30	24	NotCritical
autogen1:Bearings_FP	autogen1:Corrosion	24	Critical
autogen1:Bearings_FP	autogen1:Flaking	24	Critical
autogen1:Bearings_FP	autogen1:Break	8	Critical
autogen1:Bearings_FP	autogen1:Wear	36	Critical
autogen1:Brake_FP	autogen1:Blockage_10	27	NotCritical
autogen1:Camshaft_FP	autogen1:Wear_20	18	NotCritical
autogen1:Camshaft_FP	autogen1:Misalignment_20	18	NotCritical
autogen1:ServoMotor_FP	autogen1:DegradedPowerTransmission_50	24	NotCritical
autogen1:ServoMotor_FP	autogen1:ImpossibleStop_50	27	NotCritical
autogen1:ServoMotor_FP	autogen1:ImpossibleStart_50	27	NotCritical
autogen1:Coder_FP	autogen1:Blockage_30	27	NotCritical

Fig. 12. The Calculation and Criticality Classification Results of the other Failure Modes of the Forming Press Components.



Property assertions: ServoMotor_FP

Object property assertions
- hasFailureMode ImpossibleStop_50
- hasFailureMode ImpossibleStart_50
- hasFailureMode DegradedPowerTransmission_50

Data property assertions
- hasCriticalityClass "NotCritical"^^xsd:string
- hasSecurityEffect 0
- hasMaxFailureModeCriticality 27
- hasMaintenanceStrategy "CorrectiveMaintenance"^^xsd:string

Negative object property assertions

Property assertions: CrossBeams_FP

Object property assertions
- Commands FormingMold
- hasFailureMode Wear_40
- hasFailureMode DegradedClosing_40

Data property assertions
- hasCriticalityClass "Critical"^^xsd:string
- hasMaxFailureModeCriticality 36
- hasAssociatedDiagnosticTool 1
- hasMaintenanceStrategy "ConditionalMaintenance"^^xsd:string

Negative object property assertions

Property assertions: Bearings_FP

Object property assertions
- Commands NoneComponent
- hasFailureMode Flaking
- hasFailureMode Wear
- hasFailureMode Corrosion
- hasFailureMode Break

Data property assertions
- hasCriticalityClass "Critical"^^xsd:string
- hasMaxFailureModeCriticality 36
- hasMaintenanceStrategy "SystematicMaintenance"^^xsd:string

Negative object property assertions

Property assertions: Camshaft_FP

Object property assertions
- hasFailureMode Corrosion_20
- hasFailureMode Wear_20
- hasFailureMode Misalignment_20

Data property assertions
- hasCriticalityClass "NotCritical"^^xsd:string
- hasMaxFailureModeCriticality 18
- hasSecurityEffect 0
- hasMaintenanceStrategy "CorrectiveMaintenance"^^xsd:string

Negative object property assertions

Property assertions: Coder_FP

Object property assertions
- hasFailureMode Blockage_30
- hasFailureMode Damage_30

Data property assertions
- hasSecurityEffect 0
- hasMaxFailureModeCriticality 27
- hasCriticalityClass "NotCritical"^^xsd:string
- hasMaintenanceStrategy "CorrectiveMaintenance"^^xsd:string

Negative object property assertions

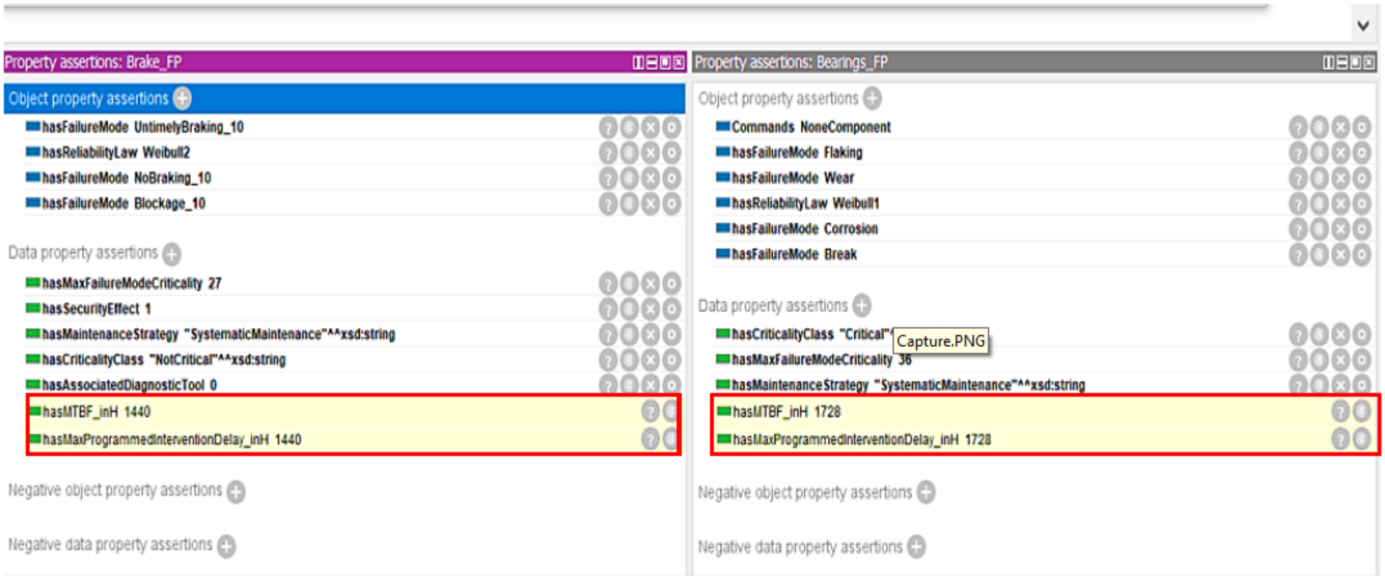
Property assertions: Brake_FP

Object property assertions
- hasFailureMode UntimelyBraking_10
- hasFailureMode NoBraking_10
- hasFailureMode Blockage_10

Data property assertions
- hasMaxFailureModeCriticality 27
- hasSecurityEffect 1
- hasCriticalityClass "NotCritical"^^xsd:string
- hasAssociatedDiagnosticTool 0
- hasMaintenanceStrategy "SystematicMaintenance"^^xsd:string

Negative object property assertions

Fig. 13. The Results of the Appropriate Maintenance Policy for each Element of the Forming Press.



Property assertions: Brake_FP

Object property assertions
- hasFailureMode UntimelyBraking_10
- hasReliabilityLaw Weibull2
- hasFailureMode NoBraking_10
- hasFailureMode Blockage_10

Data property assertions
- hasMaxFailureModeCriticality 27
- hasSecurityEffect 1
- hasMaintenanceStrategy "SystematicMaintenance"^^xsd:string
- hasCriticalityClass "NotCritical"^^xsd:string
- hasAssociatedDiagnosticTool 0
- hasMTBF_inH 1440
- hasMaxProgrammedInterventionDelay_inH 1440

Negative object property assertions

Negative data property assertions

Property assertions: Bearings_FP

Object property assertions
- Commands NoneComponent
- hasFailureMode Flaking
- hasFailureMode Wear
- hasReliabilityLaw Weibull1
- hasFailureMode Corrosion
- hasFailureMode Break

Data property assertions
- hasCriticalityClass "Critical"
- hasMaxFailureModeCriticality 36
- hasMaintenanceStrategy "SystematicMaintenance"^^xsd:string
- hasMTBF_inH 1728
- hasMaxProgrammedInterventionDelay_inH 1728

Negative object property assertions

Negative data property assertions

Fig. 14. The Obtained Results by the Maintenance Operationalization Module Rules.

XII. Conclusion and Perspectives

An intelligent decision making system for the automation of industrial production systems maintenance using digital twins and ontologies is developed in this paper. The integration of these two concepts in the same approach applied to the maintenance process has given birth to a hybrid system that presents new originalities.

On the first hand, the majority of research works related to digital twins propose a variety of approaches in relation to different domains and processes in general, and in relation to the maintenance process in particular, but they do not take into account the dimension of the connection between the physical and virtual spaces, which is essential for the applicability of these approaches. Thus, the resolution of this problem of cyber-physical interoperability of digital twins was the first objective to be attained through this paper. In fact, the concept of ontologies was integrated in the proposed approach. A maintenance ontology (DTMa-Onto) was constructed, and due to the expressivity capacities of ontologies, the DTMa-Onto was employed to ensure in real time a faithful transfer of large quantities of data between the two physical and virtual spaces in order to reproduce the functioning and behavior of physical entities digitally. On the other hand, the second originality of this paper is that it proposes an automation approach for the control of the whole maintenance process, contrary to previous works which only treat some sides of the maintenance process in the same approach and cannot take the decision automatically. Indeed, the DTMa-Onto has been fed by many categories of inference rules that are used to calculate and classify criticalities and downtimes, choose the maintenance policy, propose actions to be taken, predict failures, etc. To summarize, the work realized is in the form of a structured and global methodology for the automation of the entire maintenance process, from data collection to decision making and the methodology was validated by an industrial case study.

As perspectives, it is suggested to integrate other aspects in the proposed approach (the environmental aspect, the financial aspect, etc.). It is also proposed to automate the maintenance process in another way, using other artificial intelligence tools. Another perspective is to propose other approaches for the automation of other processes (logistics, design, etc.).

REFERENCES

[1] VanDerHorn, E., & Mahadevan, S. (2021). Digital Twin: Generalization, characterization and implementation. *Decision Support Systems*, *145*, 113524.

[2] Ghita, M., Siham, B., & Hicham, M. (2020). Digital twins development architectures and deployment technologies: Moroccan use case. *International Journal of Advanced Computer Science and Applications*, *11*(2).

[3] Pang, T. Y., Pelaez Restrepo, J. D., Cheng, C. T., Yasin, A., Lim, H., & Miletic, M. (2021). Developing a digital twin and digital thread framework for an ‘Industry 4.0’Shipyard. *Applied Sciences*, *11*(3), 1097.

[4] Grieves, M. (2014). Digital twin: manufacturing excellence through virtual factory replication. White paper, 1(2014), 1-7.

[5] Grieves, M., & Vickers, J. (2017). Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In Transdisciplinary perspectives on complex systems (pp. 85-113). Springer, Cham.

[6] Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, *51*(11), 1016-1022.F

[7] Rasheed, A., San, O., & Kvamsdal, T. (2020). Digital twin: Values, challenges and enablers from a modeling perspective. *Ieee Access*, *8*, 21980-22012.

[8] Abadi, M., Abadi, C., Abadi, A., & Ben-Azza, H. (2022). Digital Twin-Driven Approach for Smart Industrial Product Design. In *International Conference On Big Data and Internet of Things* (pp. 263-273). Springer, Cham.

[9] Tao, F., Zhang, M., Liu, Y., & Nee, A. Y. (2018). Digital twin driven prognostics and health management for complex equipment. *Cirp Annals*, *67*(1), 169-172.

[10] Tao, F., & Zhang, M. (2017). Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing. *Ieee Access*, *5*, 20418-20427.

[11] Souifi, A., Cherfi, Z., Marc, Z., Barkallah, M., & Haddar, M. (2021, March). Le jumeau numérique dans le pilotage de performance. In *17ème colloque national S-mart AIP-PRIMECA*.

[12] He, B., Li, T. Y., & Xiao, J. L. (2022). Digital Twin-Driven Design for Product Control System. In *Digital Twins for Digital Transformation: Innovation in Industry* (pp. 41-65). Springer, Cham.

[13] Zhao, Z., Zhang, M., Chen, J., Qu, T., & Huang, G. Q. (2022). Digital Twin-enabled Dynamic Spatial-temporal Knowledge Graph for Production Logistics Resource Allocation. *Computers & Industrial Engineering*, 108454.

[14] Nguyen, T., Duong, Q. H., Van Nguyen, T., Zhu, Y., & Zhou, L. (2022). Knowledge mapping of digital twin and physical internet in supply chain management: A systematic literature review. *International Journal of Production Economics*, *244*, 108381.

[15] Fu, Y., Zhu, G., Zhu, M., & Xuan, F. (2022). Digital Twin for Integration of Design-Manufacturing-Maintenance: An Overview. *Chinese Journal of Mechanical Engineering*, *35*(1), 1-20.

[16] Errandonea, I., Beltrán, S., & Arrizabalaga, S. (2020). Digital Twin for maintenance: A literature review. *Computers in Industry*, *123*, 103316.

[17] Pairet, È., Ardón, P., Liu, X., Lopes, J., Hastie, H., & Lohan, K. S. (2019, March). A digital twin for human-robot interaction. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (pp. 372-372). IEEE.

[18] Longo, F., Nicoletti, L., & Padovano, A. (2019). Ubiquitous knowledge empowers the Smart Factory: The impacts of a Service-oriented Digital Twin on enterprises' performance. *Annual Reviews in Control*, *47*, 221-236.

[19] Rødseth, H., Wilson, A., & Schjølberg, P. (2020). Integrated Production and Maintenance Planning for Successful Asset Management Strategy Implementation. In *Engineering Assets and Public Infrastructures in the Age of Digitalization* (pp. 610-617). Springer, Cham.

[20] Eckhart, M., & Ekelhart, A. (2019). Digital twins for cyber-physical systems security: State of the art and outlook. *Security and quality in cyber-physical systems engineering*, 383-412.

[21] Semeraro, C., Lezoche, M., Panetto, H., & Dassisti, M. (2021). Digital twin paradigm: A systematic literature review. *Computers in Industry*, *130*, 103469.

[22] Aivaliotis, P., Georgoulias, K., & Chryssolouris, G. (2019). The use of Digital Twin for predictive maintenance in manufacturing. *International Journal of Computer Integrated Manufacturing*, *32*(11), 1067-1080.

[23] Abadi, M., Abadi, C., Abadi, A., & Ben-Azza, H. (2022). A Smart Decision Making System for the Selection of Production Parameters using Digital Twin and Ontologies. *International Journal of Advanced Computer Science and Applications*, *13*(2).

[24] Meyer, M., Yu, Z., Gulati, P., Delforouzi, A., Roggenbuck, J., & Wolf, K. Ontologies for digital twins in smart manufacturing Whitepaper.

[25] Abadi, A., Sekkat, S., & Ben-Azza, H. (2015, December). Utilisation des ontologies pour supporter la conception simultanée du produit et de sa chaîne logistique. In *Xème Conférence Internationale: Conception et Production Intégrées*.

[26] Mfetoum, I. M., Essiane, S. N., Kamanke, E. F., Ndzie, G. B., & Mbemmo, S. (2016). Estimation des paramètres du modèle de WEIBULL: application à la modélisation de la fiabilité de l’entailleuse-perceuse mécanique de ballasts de chemin de fer de CAMRAIL. *Sciences, Technologies et Développement, Edition spéciale, pp117-121*.