Context Ontology Development for Connected Maintenance Services

Emmanouilidis, C, Gregori, M., and Al-Shdifat, A.

*Cranfield University, School of Aerospace, Transport and Manufacturing, Cranfield, Bedfordshire MK43 0AL, UK.
(e-mail: {christosem; ali.alshdifat }@cranfield.ac.uk, matteo.gregori91@gmail.com)*

Abstract: The opportunity to shift from corrective and preventive to data-driven Predictive Maintenance has received a significant boost with the deeper penetration of Internet of Things (IoT) technologies in industrial environments. Processing IoT generated data nonetheless creates challenges for data management and actionable data processing. One way to handle such complexity is to introduce context information modelling and management, wherein data and service delivery are determined upon resolving the apparent context of a service or data request. In this paper, context information management is considered on the basis of a valid knowledge construct for reliability-oriented maintenance management. The aim is to produce a viable semantic organization of data for maintenance services. It is applied on an industrial case linked to maintenance of a distributed fleet of connected production grade industrial printers. The complexity of translating the data generated by such production assets to actionable information is significant, as the status of a single asset is characterised by several hundreds of failure modes and a multitude of event codes. To assess the viability of the ontology for the targeted application, a qualitative usability evaluation study of the ontology is performed.

Keywords: Internet of Things, Context Management, Ontology, Interoperability, Maintenance

1. INTRODUCTION

Asset and maintenance management are concerned with the management practices, technologies and tools necessary to maximise the value delivered by physical engineering assets. Internet of things (IoT) - generated data are increasingly considered as an asset and the data asset value needs to be maximised too. However, asset-generated data in practice are often collected in non-actionable form. The difficulty does not only lie only with the usual big data challenges, namely volume, velocity, veracity, and variety, but also with the poor grounding of the data to established or evolving domain knowledge and limited understanding of the data context. In order to efficiently manage such data, context information management has emerged as a key concept to enabling complexity and interoperability management in IoT-enabled environments (Perera et al., 2014). In the application domain of asset and maintenance management, context is relevant to the asset and its hierarchy, the user, the production or service business circumstances, as well as to overall system and operating environment aspects (Emmanouilidis et al., 2019). Focusing on the asset context, relevant domain knowledge can be modelled in many forms but of particular interest are knowledge constructs relevant to reliability analysis, such as the well-established Fault Modes and Effects Analysis (FMEA) and its variation Fault Modes Effects and Criticality Analysis (FMECA) (IEC60812, 2018). However, FME(C)A models are mostly employed as a design-stage engineering study. In contrast, maintenance services need to be invoked at operating time and therefore relevant knowledge representations need to be enriched to enable the dynamic inference of context and the composition of contextually relevant services. This can be served via domain-relevant ontological modelling and several relevant maintenance

ontologies have been proposed in the literature. However, such ontologies need to be further developed to drive the adaptation of context-dependent maintenance services.

This paper presents a study of maintenance ontologies from the viewpoint of reliability-oriented context information management and proposes a baseline context information management ontology aligned with the needs of maintenance services for connected production machines. This ontology is applied on an industrial case study relevant to maintenance services for a distributed fleet of connected industrial printers. Results from a qualitative usability evaluation of the developed ontology are then presented. The rest of this paper is organised as follows: Section 2 discusses related work on maintenance ontologies. Section 3 presents the ontology development, while section 4 introduces its implementation on a case study. Section 5 outlines the qualitative evaluation approach and its results. Section 6 is the conclusion.

2. RELATED WORK

While maintenance knowledge can be represented in multiple forms, of particular interest for driving maintenance services are reliability–based knowledge constructs, which relate fault modes with recommended actions. Among those, Fault Modes and Effects Analysis (FMEA) and its variation Fault Modes Effects and Criticality Analysis (FMECA) are often employed as an engineering study at the design stage of a physical asset (IEC60812, 2018). For such knowledge to be exploitable at operating time, the representation needs to shift from tables to a semantically enriched model. Reflecting on needs for dynamic knowledge composition and utilisation, ontologies based on FMEA principles have been proposed (Lee, 2001)(Dittmann et al., 2004). The core maintenance concepts of FMEA models are the failure modes and their causal relationship with phenomena or events that may

trigger the occurrence of a failure, as well as their impact in the form of effects on the state of an asset and its function.

A maintenance ontology can feed into an agent which drives maintenance services, such as a process monitoring agent. Instead of a direct query matching, a reasoning mechanism applied on a maintenance ontology is based on a semantic matching (Pakonen et al., 2007). Upon capturing at an abstract level the key concept of a failure mode, extended FMECA ontologies provide not only relationships between causal phenomena, failure modes, and their effects, but also with recommended actions. Knowledge relevant to actions extends to guidance for required resources, including human resources and spare parts (Jin et al., 2009). Therefore, such a maintenance ontology serves as a semantic formalism which can be employed to drive maintenance services (Karray et al., 2010)(Karray et al., 2014). The mechanism for this is through resolving the context of a service request. Reasoning employing semantic similarity using ontological distance metrics or other relevant means can be used to this end.

Asset maintenance action recommendation nonetheless cannot be considered in isolation from other operational aspects and therefore operational semantics are needed for an ontology of appropriate scope and applicability. Therefore, a maintenance ontology can be extended to a multi-layer modelling construct: an upper level ontology to capture the higher level concepts and entities; and a lower level one with event-level operational and application-specific context (Koukias et al., 2013). Decoupling maintenance semantics into upper and lower abstraction layers allows the effective decentralisation of modelling, which is highly relevant to modern connected and distributed extended production ecosystems (Abele et al., 2014). It enables service adaptation and delivery consistent with wider production and operational factors and can drive both asset–specific and fleet-level services, inclusive of prognostics and health management (PHM) (Medina-Oliva et al., 2014). A maintenance ontology extended with PHM concepts can therefore be valuable in enabling a data and event-driven process workflow, wherein data acquired from assets are processed and translated into maintenance action recommendations (ISO 13374:1, 2003). Such workflows can take into account condition monitoring signals and parameters which are indicative of an asset's condition. Therefore the ontology can include established knowledge about fault modes detectability through monitored parameters (ISO 17359, 2011) to drive reasoning that relates monitoring parameters and indicators to fault modes and then fault modes to actions (D'Elia et al., 2010).

Maintenance ontologies may also be looked upon from the viewpoint of the domain that the maintenance function serves. For example, when considering the manufacturing domain, it is of interest to capture the functional impact of asset integrity level on the actual manufacturing process. Although such impact can be expressed in different ways that link condition monitoring with the manufacturing function (Cao et al., 2019), employing mature knowledge constructs, such as FMECA, offers a sound basis upon which to express the organisational, and functional association between a manufacturing asset hierarchy and its linkage with the functional integrity of the production facility. Ontological approaches to support maintenance management that employ

FME(C)A concepts have been developed for a range of assets, including wind turbines (Zhou et al., 2015), robotised production (Chioreanu et al., 2015), machine tools (Zhou et al., 2017), pumps (Nuñez & Borsato, 2018), smart homes (Ali & Hong, 2018), aeronautics/space (Castet et al., 2018), and transport infrastructure (Ebrahimipour et al., 2010)(Ren et al., 2019). FMECA based ontological modelling can be further enhanced by additional reliability models, such as Fault Tree Analysis (FTA), to allow a deeper causal analysis of failures. FMECA-based ontologies can therefore play an key role in dependability analysis for distributed asset management in Cyber Physical Systems (CPS) (Sanislav et al., 2016). Nonetheless, unless both domain and application specific context is built into the ontology, the offered maintenance service adaptation and delivery mechanisms are unlikely to be effective. While the high level domain context can in many cases be abstracted through generic modelling formalisms, the application – specific context has to be modelled and resolved in application – specific terms.

3. MAINTENANCE ONTOLOGY MODELLING

This work has been motivated by the need to drive the adaptation of the service management process for a fleet of networked production assets, namely heavy duty production printers of a large original equipment manufacturer (OEM). Available methods for ontology development involve a specification phase, before moving to conceptualisation and design, implementation, deployment, and evaluation, while including also ontology maintenance mechanisms. For example, *Menthontology* involves a specification phase, followed by conceptualisation and implementation (Lopez et al., 1999). A similar process is followed in this paper. The first two phases are presented in the remainder of this section, while the implementation is described in section 4.

Specification. This concerns the ontology purpose and scope. The ontology development is motivated by the need to design an approach to context information management relevant to data and event-driven service management of a distributed fleet of critical production assets. The scope was linked to reliability aspects of the networked production assets. It was therefore considered appropriate to adopt the FMECA key concepts to define ontology entities and their relationships.

Conceptualisation. This deals with knowledge acquisition and structure. Activities include gathering relevant terms and relationships between them and is the main design step. Once a decision was made to adopt FMECA as the backbone knowledge for this study, the assembled terminology was based on a subset of the core terminology of relevant standards, primarily (IEC60812, 2018), and (ISO13372, 2012)(ISO13306, 2017)(ISO2041, 2018). At the first cycle of the ontology definition, the decision was not to completely replicate the terminology of the aforementioned standards but to simply adopt a subset of terms, some in adapted form, so as to allow deploying it on the considered case study. Further future work will examine a more generic and holistic terminology usage based on the above standards. Upon enumerating the terms to be utilised in the ontology, the next step was to define, appropriate classes for terms, their hierarchy, and relationships between them. The class hierarchy is shown in *Fig. 1*.
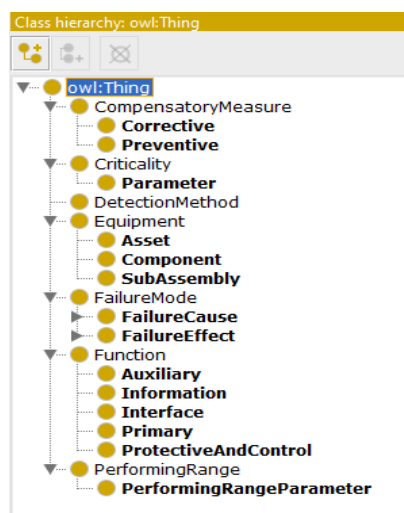
Class hierarchy: owl:Thing

- owl:Thing
  - CompensatoryMeasure
    - **Corrective**
    - **Preventive**
  - Criticality
    - **Parameter**
  - DetectionMethod
  - Equipment
    - **Asset**
    - **Component**
    - **SubAssembly**
  - FailureMode
    - **FailureCause**
    - **FailureEffect**
  - Function
    - **Auxiliary**
    - **Information**
    - **Interface**
    - **Primary**
    - **ProtectiveAndControl**
  - PerformingRange
    - **PerformingRangeParameter**

Fig. 1. Classes and Sub-Classes in the ontology

A more detailed view of the class hierarchy with key relations between them is shown in *Fig. 2*. Having defined the modelled classes, the next OWL 2 syntax elements are the object class properties.
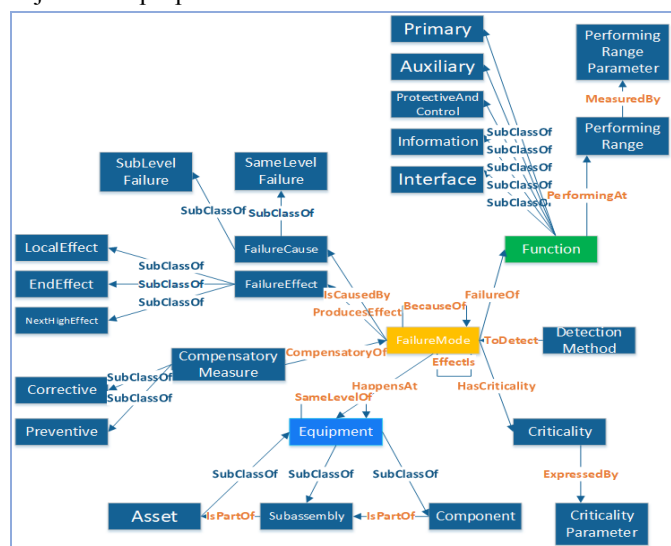
Fig. 2. Representation of the ontology model

Table 1. Object Properties

Object Property	Domain	Range
BecauseOf	FailureMode	FailureMode
CompensationFor	Compensatory Measure	FailureMode
EffectIs	FailureMode	FailureMode
ExpressedBy	Criticality	Parameter
FailureOf	FailureMode	Function
HappensAt	FailureMode	Equipment
HasCriticality	FailureMode	Criticality
IsCausedBy	FailureMode	FailureCause
IsPartOf	Component	Subassembly

These define the relation between classes and data properties, identifying the values used to initialize a given class. Object properties can be seen as predicates that define the relation between a subject and an object. Object properties are listed in Table 1. Data properties represent the parameters a given Class can be initialized with. They define the type of value to be initialized. In this ontology, data properties are instantiated to define code identifiers for Components, Processes (for example detection methods), and Failure Modes (Table 2).

Table 2. Ontology data properties

Data Property	Domain	Range
Asset_ID	Asset	Integer
Asset_name	Asset	String
Component_ID	Component	Integer
Component_name	Component	String
DetectionMethod	DetectionMethod	String
FailureCode	FailureMode	Integer
FailureMode	FailureMode	String
Function_ID	Function	Integer
Function_name	Function	String

4. IMPLEMENTATION

In the implementation step concept instances (also defined as objects, values or individuals) are created, which represent the physical elements that will be the subject of analysis in a case study. A class is selected for every instance such that the attributes of the object, data and/or annotation are bound together. The final stage includes the steps of initialisation and individuals' implementation on the case study.

The designed ontology is applied on the case of the service management process for a large OEM of production printing machines. Leaving commercially sensitive information aside, the study presents only information strictly relevant to the presented research. Production printers are high value assets, costing hundreds of thousands of dollars per unit, while the value associated with their output is many times more. Demand for their output can be created on very short notice and any events that may result in functional failures and unplanned production stoppage are sources of considerable business disruption and costs. The OEM enters into a Service Level Agreement (SLA) with customers, which involves on-site technical support for production disruption resolution within a few hours of a service event or call, when the issue under consideration cannot be resolved promptly otherwise.

The studied family of assets constitutes complex hierarchies of asset components, which have over 800 failure modes associated with them. The connected production machines generate data which include alert and fault codes, operational and historical usage parameters, as well as monitored indicators. This contains information of potential value for detecting, diagnosing, or predicting events of interest. The volume of data produced by such networked production machinery is in the order of several GBs daily (~2TB annually). However, the key difficulty lies with the

complexity of the asset itself and the several hundreds of failure modes from 15 key modules of the production asset. For such data to produce actionable knowledge, the development of a maintenance context ontology can resolve the context of service events and link them with recommended actions. Serving the needs of this case study, instances of objects are created. These instances are the agents that constitute the physical elements that are the subject of analysis for the service management process.

The ontology is populated with concepts and entities containing contain engineering, operational and maintenance management data and information. Although the data sets are specific to the studied company and the product family under consideration, its nature and structure are generic. Definition data includes asset components and their hierarchy, fault modes, names and codes, and severity assessment. Monitoring data relate machinery measurements to asset components, operational information, and event codes. Overall 26 classes, 16 object properties, and 17 data properties were defined. After identifying the semantic organisation of information within the datasets, the next step was to create instances using existing classes for each key element contained in the dataset. To this end 889 instances of individuals were created. To populate individuals' properties, fault names have been used as data keys and for each Failure Mode the following key properties have been initialised:

"HasCriticality" (object property) + Criticality (class)

"IsCausedBy" (object property) + Failure Cause (class)

"HappensAt" (object property) + Equipment (class)

"Failure_code" (data property) + <integer> (type)

The primary use of the ontology is to serve resolving queries linked to maintenance operations, including monitoring, scheduled, or unscheduled maintenance. For example, if a machine shows FAILURE CODE 010-319, by querying the model it is possible to derive enriched information, such as:

Failure CODE 010-319 is associated with the failure "FUSING HOT NOT READY"; "FUSING HOT NOT READY" is a failure of the "FUSING" function; "FUSING HOT NOT READY" happens at the "FUSER"; "FUSING HOT NOT READY" is detected by "TEMPERATURE".

Resolving the context of an alert, event, and maintenance service request or action is required to be able to dynamically compose data process workflows that may link data to monitoring, diagnostic, predictive, or prescriptive knowledge or actions. The next section discusses the model evaluation.

5. EVALUATION

Several ontology evaluation methodologies and metrics have been proposed, which can take a design or an implementation viewpoint (Degbelo, 2017)(Kumar & Baliyan, 2018). Design evaluation criteria include accuracy, adaptability, cognitive adequacy, completeness, conciseness, consistency, clarity, expressiveness, and grounding. Implementation quality is assessed for computational efficiency, congruency, practical usefulness, precision, and recall. The scope of the present case study was exploratory, i.e. the intention was to propose an initial viable ontology prior to final design, implementation, deployment, and validation. Therefore, it was considered appropriate to focus on a subset of evaluation

criteria, namely usability, correctness, and applicability, within the viewpoint of the targeted application case study.

5.1 Usability Evaluation

Model usability was assessed using a System Usability Scale (SUS) test (Brooke, 2013). While SUS is not strictly relevant to ontologies, it is widely adopted as a practitioner's approach for usability testing in similar problem domains, as it originated from the perspective of delivering appropriate information in decision support systems for fault diagnosis. It is therefore highly relevant to the application context of this paper, and has been applied in the past for application ontology evaluation cases (Tan et al., , 2017). SUS tests typically comprise ten statements from a pool of possible evaluation assessments, following a Likert scale and the scores are translated to a range between 0 and 100; however numbers do not denote percentages.

The present study employed ten statements, asking respondents to express their level of agreement (strongly disagree, disagree, neutral, agree, and strongly agree). The questions were as follows:

1) I could contribute to the model presented in this project

2) I found the model unnecessarily complex

3) I find the model easy to understand

4) I need further theoretical support to understand the model

5) I found the concepts in this system well integrated

6) I thought there was too much inconsistency in this model

7) I would imagine that most maintenance experts would understand this model very quickly

8) I find the model very cumbersome to understand

9) I am confident I understand the model conceptualisation

10) I needed to ask a lot of questions before I could understand the conceptualization of the model

The viewpoint was that of qualitative assessment and the sample included a mix of six industrialists and academic researchers. After collecting the results of this survey, the actual SUS test was performed and SUS scores were calculated following the adopted methodology (Brooke, 2013)(Tan et al., 2017). These are shown in Table 3:

Table 3. SUS results

Recipient	1	2	3	4	5	6
Score	75	77.5	70	67.5	92.5	77.5

Results show that individual scores are placed well above the acceptable border of 50.9, with an average of 76,67 and a peak of 92.5. This provides evidence of positive usability perception in qualitative terms, which bodes well with the purpose of the ontology. However, in any consequent phase after the present study, sample sizes should be larger and seek to include especially a significant number of practitioners.

5.2 Correctness Evaluation

Correctness evaluation was performed by contrasting the Protégé model with existing literature for Maintenance and FMECA Standards. This primarily involved (IEC60812, 2018), and secondary (ISO13372, 2012), (ISO13306, 2017),

(ISO2041, 2018). At first, terminology mapping was applied to verify that standard vocabulary would fit into the model.

Table 4. FMECA and ontology vocabulary mapping

Class	Term
CompensatoryMeasure; Preventive; Corrective	Treatment; Preventive; Corrective; Attenuation
Criticality	Criticality; Prognosis; Availability
Parameter	Likelihood; Severity; Testability; Maintainability; Reliability; Alignment; Thermal Growth; Abnormality;
DetectionMethod; PerformingRange; PerformingRangeParameter	Detection Method; Control; Condition Monitoring; Diagnostics; Alarm; Alert; Sign; Symptom; Dynamic Range; Thermography; Time Window; Frequency domain; Time Domain; Waterfall; Critical Speed Map; Failure Rate; Fault Frequency; Frequency Analysis; Pareto Analysis; Risk Assessment; Baseline
Equipment; Asset; Subassembly; Component	System; Item; Element; Equipment; Machine;
FailureMode	Failure Mode; Common Mode Failures; Breakdown; Failure; Fault; Anomaly; Distortion; Fault Progression;
FailureCause	Failure Cause; Failure Mechanism; Human Error; Common Cause Failures; Background Noise; Triboelectric Noise; Noise Floor; Root Cause
SameLevel/SublevelFailure	Hierarchy Level;
FailureEffect	Failure Effect;
Function; Primary; Auxiliary; Information; Interface; ProtectiveAndControl	Redundancy; Machinery Health Management;
<Other>	Process; Scenario;

Table 5. FMECA and ontology syntax

FMECA step	Ontology	Syntax example
Sub-divide item or process into elements	Class: Equipment Sub-Classes: Asset, Sub-assembly, Components	"Engine" (subassembly) is part of "Car" (asset);
Identify functions for each element	Class: Function	"Heating" primary function of "Heater"
Identify failure modes	Class: FailureMode	"HeaterBreak" (FailureMode) failure of "Heating"
Identify detection methods	Class: DetectionMethod	"TempSensor" (DetectionMethod) detects "HeaterBreak"
Identify failure causes	Class: FailureCause	"Overheating" (failureCause) subclass of "Break" (failureMode)
Determine severity of failure final effect	Class: Criticality	"Level1" (Criticality) determined through "Severity" (Parameter)
Estimate likelihood of failure mode	Class: Criticality	"Level1" (Criticality) expressed by "Risk" (Parameter)
Estimate other criticality parameters	Class: Parameter	"Severity" (Parameter)
Identify actions	Class: CompensatoryMeasure	"ComponentChange" (compensatory Measure) Compensatory of "Heater Break"

All relevant terms were checked to assess if they could be expressed through the model or initialised in it. Next, the FMECA terms were compared with the model to verify that each of the steps of a standard process workflow could be performed, and the information was properly labelled and stored. The first step showed a basic overlap of the vocabulary with ontology classes and individuals (Table 4). The second step confirmed adherence to FMECA process steps, validating its correctness (Table 5).

6. CONCLUSION

This paper presented the design of a maintenance ontology linked to reliability-related knowledge, aimed at supporting context management for maintenance services. While the application focus is specific, the ontology abstraction level is such that it could also be implemented on other application cases, as it offers a sound baseline for further customisation or extensions. The semantic structure is particularly relevant for Predictive Maintenance, as it allows to process data in to assess likelihood and criticality of functional disruptions. The presented model satisfies one of the four sections of the context lifecycle, namely context modelling. It requires, therefore, an appropriate context gathering solution as well as proper development in reasoning and dissemination. This is included in the next steps of the work employing the Pellet reasoned included in Protégé. Further work also includes additional applications in the case of IoT-enabled monitoring of complex machinery at laboratory settings to enable context-based adaptation for related maintenance services.

Ontology designs for complex and distributed fleets of diverse types of assets can become cumbersome and hard to handle. Meta-modelling approaches, such as the ADOxx© (https://www.adoxx.org/live/owl) web ontology language can be employed for the systematic integration of modelling elements, such as simulation, analysis, visualisation, etc and become part of systematic ontology development in complex systems (Milicic et al., 2016). Overall, such approaches are well aligned with broader event-driven condition–based maintenance architectures, as they support the resolution of event context, enabling context-based maintenance actions determination (Bousdekis et al., 2018).

REFERENCES

Abele, L., Grimm, S., Zillner, S., & Kleinsteuber, M. (2014). An ontology-based approach for decentralized monitoring and diagnostics. *INDIN 2014*, 706–712. https://doi.org/10.1109/INDIN.2014.6945600

Ali, N., & Hong, J. E. (2018). Failure detection and prevention for cyber-physical systems using ontology-based knowledge base. *Computers*, *7*(4). https://doi.org/10.3390/computers7040068

Bousdekis, A., Papageorgiou, N., Magoutas, B., Apostolou, D., & Mentzas, G. (2018). Enabling condition-based maintenance decisions with proactive event-driven computing. *Computers in Industry*, *100*(May), 173–183. https://doi.org/10.1016/j.compind.2018.04.019

Brooke, J. (2013). SUS: a retrospective. *Journal of Usability Studies*, *8*(2), 29–40.

Cao, Q., Giustozzi, F., Zanni-Merk, C., de Bertrand de Beuvron, F., & Reich, C. (2019). Smart Condition Monitoring for Industry 4.0 Manufacturing Processes: An Ontology-Based Approach. *Cybernetics and Systems*, *50*(2), 82–96. https://doi.org/10.1080/01969722.2019.1565118

Castet, J. F., Bareh, M., Nunes, J., Okon, S., Garner, L., Chacko, E., & Izygon, M. (2018). Failure analysis and products in a model-based environment. *IEEE Aerospace Conference Proceedings*, 1–13. https://doi.org/10.1109/AERO.2018.8396736

Chioreanu, A., Brad, S., Porumb, C., & Porumb, S. (2015). E-maintenance ontology-based approach for heterogeneous distributed robotic production capabilities. *International Journal of Computer Integrated Manufacturing*, 28(2), 200–212. https://doi.org/10.1080/0951192X.2014.880802

D'Elia, A., Roffia, L., Zamagni, G., Vergari, F., Toninelli, A., & Bellavista, P. (2010). Smart applications for the maintenance of large buildings: How to achieve ontology-based interoperability at the information level. *IEEE Symposium on Computers and Communications*, 1077–1082. https://doi.org/10.1109/ISCC.2010.5546633

Degbelo, A. (2017). A Snapshot of Ontology Evaluation Criteria and Strategies. *Semantics2017*. (pp. 1–8). https://doi.org/10.1145/3132218.3132219

Dittmann, L., Rademacher, T., & Zelewski, S. (2004). Performing FMEA Using Ontologies. *18th International Workshop on Qualitative Reasoning*, 209–216. doi=10.1.1.83.4435

Ebrahimipour, V., Rezaie, K., & Shokravi, S. (2010). An ontology approach to support FMEA studies. *Expert Systems with Applications*, 37(1), 671–677. https://doi.org/10.1016/j.eswa.2009.06.033

Emmanouilidis, C., Pistofidis, P., Bertoncelj, L., Katsouros, V., Fournaris, A., Koulamas, C., & Ruiz-Carcel, C. (2019). Enabling the human in the loop: Linked data and knowledge in industrial cyber-physical systems. *Annual Reviews in Control*, 47, 249–265. https://doi.org/10.1016/j.arcontrol.2019.03.004

Hammar, K. (2010). The state of ontology pattern research: a systematic review of ISWC, ESWC and ASWC 2005–2009. *2nd International workshop on ontology patterns – WOP2010. vol VIII. 8 Nov 2010* (pp. 29–37). Shanghai.

IEC60812. (2018). Failure modes and effects analysis (FMEA and FMECA).

ISO 13374:1. (2003). Condition Monitoring and diagnostics of machines - Data processing, communication and presentation - Part 1: general guidelines.

ISO 17359. (2011). Condition monitoring and diagnostics of machines: General guidelines.

ISO13306. (2017). Maintenance - Maintenance Terminology.

ISO13372. (2012). Condition monitoring and diagnostics of machines — Vocabulary.

ISO2041. (2018). Mechanical vibration, schock and condition monitoring - vocabulary.

Jin, G., Xiang, Z., & Lv, F. (2009). Semantic integrated condition monitoring and maintenance of complex system. *IE and EM 2009 - Proceedings 2009 IEEE 16th International Conference on Industrial Engineering and Engineering Management*, 670–674. https://doi.org/10.1109/ICIEEM.2009.5344503

Karray, M.H., Morello, B. C., Lang, C., & Zerhouni, N. (2014). Self-Management Process in S-Maintenance Platform. *Engineering Asset Management 2011, LNME* (Vol. 9, pp. 1–9). London: Springer. https://doi.org/10.1007/978-1-4471-4993-4

Karray, Mohamed Hedi, Chebel-Morello, B., & Zerhouni, N. (2010). A contextual semantic mediator for a distributed cooperative maintenance platform. *INDIN2010*, 181–188. https://doi.org/10.1109/INDIN.2010.5549437

Koukias, A., Nadoveza, D., & Kiritsis, D. (2013). Semantic data model for operation and maintenance of the engineering asset. *IFIP AICT 398*(PART 2), 49–55. https://doi.org/10.1007/978-3-642-40361-3_7

Kumar, S., & Baliyan, N. (2018). Quality evaluation of ontologies. *Semantic Web-Based Systems: Quality Assessment Models* (pp. 19–50). https://doi.org/https://doi.org/10.1007/978-981-10-7700-5

Lee, B. . (2001). Using FMEA models and ontologies to build diagnostic models. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 15(4), 281–293.

Lopez, M. F., Gomez-Perez, A., & Sierra, J. P. (1999). Using Menthontology. *IEEE Intelligent Systems and Applications*, 14(1), 37–46. https://doi.org/10.1109/5254.747904

Medina-Oliva, G., Voisin, A., Monnin, M., & Leger, J. B. (2014). Predictive diagnosis based on a fleet-wide ontology approach. *Knowledge-Based Systems*, 68, 40–57. https://doi.org/10.1016/j.knosys.2013.12.020

Milicic, A., Kiritsis, D., & Efendioglou, N. (2016). From English to RDF - A Meta-Modelling Approach for Predictive Maintenance Knowledge Base Design. *IFIP APMS 488*, pp. 677–686). Springer. https://doi.org/10.1007/978-3-319-51133-7

Nuñez, D. L., & Borsato, M. (2018). OntoProg: An ontology-based model for implementing Prognostics Health Management in mechanical machines. *Advanced Engineering Informatics*, 38, 746–759. https://doi.org/10.1016/j.aei.2018.10.006

Pakonen, A., Tommila, T., Pirttioja, T., & Seilonen, I. (2007). OWL based information agent services for process monitoring. *IEEE, ETFA*, 9–16. https://doi.org/10.1109/EFTA.2007.4416747

Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. *IEEE Communications Surveys and Tutorials*, 16(1), 414–454. https://doi.org/10.1109/SURV.2013.042313.00197

Ren, G., Ding, R., & Li, H. (2019). Building an ontological knowledgebase for bridge maintenance. *Advances in Engineering Software*, 130, 24–40. https://doi.org/10.1016/j.advengsoft.2019.02.001

Sanislav, T., Mois, G., & Miclea, L. (2016). An approach to model dependability of cyber-physical systems. *Microprocessors and Microsystems*, 41, 67–76. https://doi.org/10.1016/j.micpro.2015.11.021

Tan, H., Adlemo, A., Tarasov, V., & Johansson, M. E. (2017). Evaluation of an application ontology. *JOWO 2017. CEUR-WS.org* (Vol. 2050, pp. 1–14).

Zhou, A., Yu, D., & Zhang, W. (2015). A research on intelligent fault diagnosis of wind turbines based on ontology and FMECA. *Advanced Engineering Informatics*, 29(1), 115–125. https://doi.org/10.1016/j.aei.2014.10.001

Zhou, Q., Yan, P., & Xin, Y. (2017). Research on a knowledge modelling methodology for fault diagnosis of machine tools based on formal semantics. *Advanced Engineering Informatics*, 32, 92–112. https://doi.org/10.1016/j.aei.2017.01.002