IFAC-PapersOnLine, Volume 53, Issue 3, 2020, pp. 251-256
DOI:10.1016/j.ifacol.2020.11.041

Ontology – based context resolution in internet of things enabled diagnostics

Ali Al-Shdifat, Christos Emmanouilidis, Muhammad Khan, and Andrew G Starr

Cranfield University, School of Aerospace, Transport and Manufacturing, Cranfield, Bedfordshire MK43 0AL, UK
(e-mail: {ali.alshdifat ; christosem; muhammad.a.khan; a.starr}@cranfield.ac.uk)

Abstract: Internet of things (IoT)-generated data from industrial systems are often collected in non-actionable form, thus not directly aiding maintenance actions. Context information management is often seen as an enabler for interoperability and context-based service adaptation, acting as a mechanism for linking data with knowledge to adaptive data and services. Ontology-based approaches for semantic maintenance have been proposed in the past as a data and service mediation mechanism and are adopted here as the starting point employed to develop a context resolution service for industrial diagnostics. The underlying ontology of the context resolution mechanism is relevant to failure analysis of mechanical components. The terminology and relationship between concepts are structured on the basis of relevant standards with a reliability-oriented knowledge grounding. A reasoning mechanism is employed to deliver context resolution and the derived context can add a metadata layer on data or events generated by automated and human-driven means. The approach is applied on a gearbox test rig appropriate for emulating complex misalignment cases met in many manufacturing and aerospace applications.

Keywords: Context Management, Maintenance Ontology, Industrial Diagnostics

1. INTRODUCTION

The introduction of internet of things (IoT) technologies has expanded the ability of industries to generate data with devices that are capable of sensing and communicating in real-time, supporting decision-making processes for monitoring the state of equipment and offering guidance for proactive maintenance (Bousdekis et al., 2015). The explosive growth of IoT-generated and managed data nonetheless requires substantial further effort for the effective management and exploitation of the data. Among the key instruments to tackle such complexity is the concept of context information management (Al-shdifat and Emmanouilidis, 2018). Appropriate maintenance knowledge representations can exploit both standard knowledge as well as generated maintenance data. Ontologies offer appropriate formalisation of knowledge and allow context resolution via traversing scalable semantic graphs (Kamsu-Foguem and Noyes, 2013). Domain-specific ontologies are appropriate for modelling key maintenance concepts and drive such reasoning (Karray et al., 2011; Matsokis and Kiritsis, 2012).

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 overall system and operating environment aspects (Emmanouilidis et al., 2019). The resolution of asset context is needed to analyse mechanical systems and logically connect measurements, observed behaviour and intended function, with machinery operating condition and faults. To this end, Fault Modes Effects and Criticality Analysis (FMECA) or simply Fault Modes and Effects Analysis (FMEA) offer appropriate grounding for the baseline of the knowledge mapping (IEC60812, 2018) for several reasons. Firstly, its qualitative part makes it appropriate for abstracting maintenance reliability–oriented knowledge. Secondly, its quantitative part enables prioritisation of maintenance actions based on metrics appropriate for a risk-based approach. Lastly, its bottom up nature enables failure assessment from the base level of production systems, namely data from machinery components, all the way to system-level analysis. According to ISO 17359 (2011), failure mode analysis based on FMECA is recommended to ensure that maintenance activities are consistent with established fundamental practice-oriented knowledge. Therefore, such fundamental knowledge pertaining to mechanical component failure at a sufficiently abstract and descriptive level can be employed as a sound knowledge basis for diagnostics (Du et al., 2013; Yuan et al., 2013; Zhou et al., 2015; Guillén et al., 2016).

This paper presents the development of a context resolution service mechanism for industrial diagnostics, based on the design of a maintenance ontology focused on modelling failure analysis of mechanical components. Section 2 briefly places the present work in the context of the broader body of relevant literature. Section 3 presents the ontology development process, based on established practice and maintenance vocabulary standards. An instantiation of the developed ontology is implemented for testing on an industrially relevant test rig and is presented in section 4. The concluding section offers a discussion on the evaluation of the approach and summarises the paper’s contribution and potential future research pathways.

2. RESEARCH BACKGROUND

It is beyond the purpose of the present paper to review past research on FMEA/FMECA-based ontological modelling but instead there is a specific interest in determining how such a knowledge construct can be used to resolve context resolution requests in order to drive maintenance services

(Karray et al., 2014). In this regard, various different ontological modelling approaches have been pursued in the fields of production, maintenance, and asset management over the years. Using ontologies to model domain knowledge is a valid approach and therefore several research efforts utilising or recommending ontologies in the domain of asset and maintenance management are reported in the literature. Some of them seek to adopt relevant standards as the baseline for the ontology concepts. For example, an asset management ontology based on the PAS55 recommendation, which was later subsumed by the ISO5000 standard has been reported (Frolov et al., 2010), but the scope is broader compared to maintenance and while it serves well asset management purposes, it does not specifically target maintenance. When considering maintenance within the manufacturing domain, it is of interest to capture the functional impact of the asset integrity level on the actual production process. The anticipated integrity level would then require a predictive approach (Cao et al., 2019). Although such an approach can be highly relevant, prediction based only on historical data without accounting for predicted future operating aspects is appropriate only insofar as historical data align also with future expectations, which is often not the case. However, if the intended use of an ontology is to serve maintenance action determination, planning and scheduling, then operational semantics need to be included in the modelling.

An appropriate knowledge construct that links assets, their function, and their faults, with potential impact is FMEA or FMECA analysis (Nuñez & Borsato, 2018). However, an FMECA approach would still be limited in that while it associates assets and component faults with detectability, it does not include explicit information about measurement methods per asset and fault type or specific measured parameters for the measurement methods. While this is appropriate for the original intended purposes of an FMECA study, it falls short of the requirements for a knowledge formalism that would serve operational purposes.

A more promising approach is to extend FMECA by including in the ontology concepts that link failure modes with more detailed diagnostic information and associate recommended actions to resources that would be needed for implementing the actions, such as spare parts and human resources (Jin et al., 2009). Such an extension can look into the recommendations of relevant standards (ISO 13374:1, 2003)(ISO 17359, 2011) that link monitoring parameters and fault indicators to failure modes and recommended (D'Elia et al., 2010). Overall, a maintenance ontology may comprise multiple layers: an upper-level ontology to abstract the key domain concepts; and a lower-level ontology contextualised around specific operational factors (Koukias et al., 2013).

A knowledge construct can be used to resolve context resolution requests in order to drive maintenance services. Such resolution can be achieved by ontological reasoning based on semantic similarity, determined through ontological distance metrics or other appropriate methods (Teoh and Case, 2004). This bears relevance to similarity based reasoning, such as typically employed in Case-Based-Reasoning (CBR) systems, which have been employed in the past in the maintenance domain (Cândea et al., 2014). However, modelling and reasoning capabilities in ontologies go beyond CBR similarity. For OWL2-based the formulation of queries can be done via SPARQL queries in RDF documents. Additionally, depending on the complexity of a given ontology model, the process of semantic matching can be served using the Semantic Web Rule Language (SWRL). Overall, there is a need to further develop ontological-based modelling and inference to drive maintenance services by extending currently employed ontologies concepts to include key additional and operational ones typically included in relevant standards but less so in relevant literature.

3. MAINTENANCE ONTOLOGY DEVELOPMENT

Ontology development can follow one of many processes cited in the literature, including Uschold and King, Grüninger and Fox, Methodology, Ontology Development 101 (OD1) and KACTUS. The process can be aided by using ontology development tools, such as TopBraid and OntoStudio (commercial) or OntoEdit, HOZO and Protégé (open).

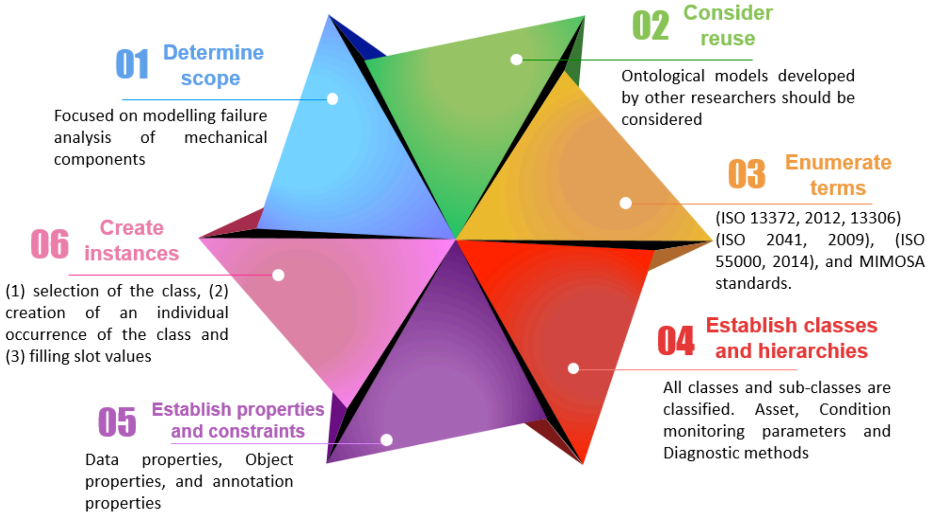

Fig. 1: The ontology development stages

The OD1 is adopted here as it is widely used (Gong and Janssen, 2013; Lau et al., 2014), has been shown to be well-suited for maintenance modelling and is well documented for implementation in Protégé environment (Ren et al., 2019). Protégé was selected here as beyond it's support for XML and RDF schema and OWL, it also provides graphic taxonomy, queries in SPARQL, rules in SWRL language, and a reasoner (Pellet). OD1 involves 6 phases (Noy and McGuinness, 2001) and the way it has been applied here is shown in Fig.1. These steps are outlined next.

3.1 Determine Scope

The initial stage in the methodology is to determine the scope. It requires to define what the ontology will cover, how it will be utilised, and the types of supported questions. The responses to such questions generally evolve throughout the process of constructing the ontology. In this work the focus of the maintenance ontology is on modelling failure analysis of mechanical components to answer queries regarding how faults manifest themselves and how they can be prevented or addressed, so as to adapt relevant diagnostics or maintenance actions in a Condition-Based Maintenance setting.

3.2 Consider Reuse

The evaluation of the degree to which ontologies can be reused or expanded is a significant factor to consider. While other maintenance ontologies exist, the specific interest here is on application-specific, operational, and diagnosability concepts, thus existing ontologies adoption would not apply.

3.3 Enumerate terms

The terminology considered for the present ontology is associated with predictive maintenance. Therefore, the main terminology and the associated definitions are based on consolidated academic literature and mostly on established international standards, such as condition monitoring, diagnostics and maintenance (ISO 13372, 2012, 13306), vibration analysis (ISO 2041, 2009), asset management (ISO 55000, 2014), and MIMOSA (www.mimosa.org) standards.

3.4 Define classes and hierarchies

The techniques used to define class hierarchies (Uschold and Gruninger, 1996) are Top-Down; Bottom-Up; and Mixed. In this work, the Top-Down method was employed, in which general classes are added first, followed by the sub-classes, a process well aligned with asset hierarchies. The process starts with a super-class, i.e. the asset type, and diagnostic methods and condition monitoring parameters. Then classes are divided in sub-classes: for example second tier sub-classes include: types of components, FMECA data, data collection parameters, and measurement methods. An example of class hierarchy is shown in Fig. 2. A more detailed view of the first, second, and third-level classes hierarchy is shown in Fig. 3, using the OntoGraf plug-in.

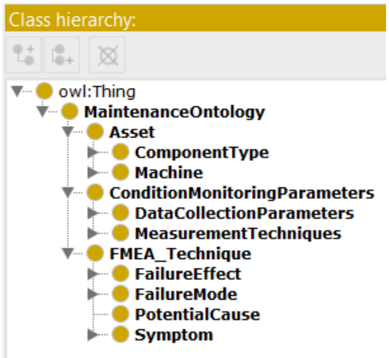

Fig. 2: Ontology classes

3.5 Define properties and constraints

Class hierarchies alone are insufficient to represent knowledge. They need to be accompanied by three distinct types of properties: data properties, object properties, and annotation properties. The object attribute explains the associations among distinct classes. The data property explains the properties of certain occurrences both quantitatively and quantitatively. The annotation property is frequently employed in the description or explanation of particular occurrences. Table 1 shows the aforementioned properties with their relevant Domain and Ranges.

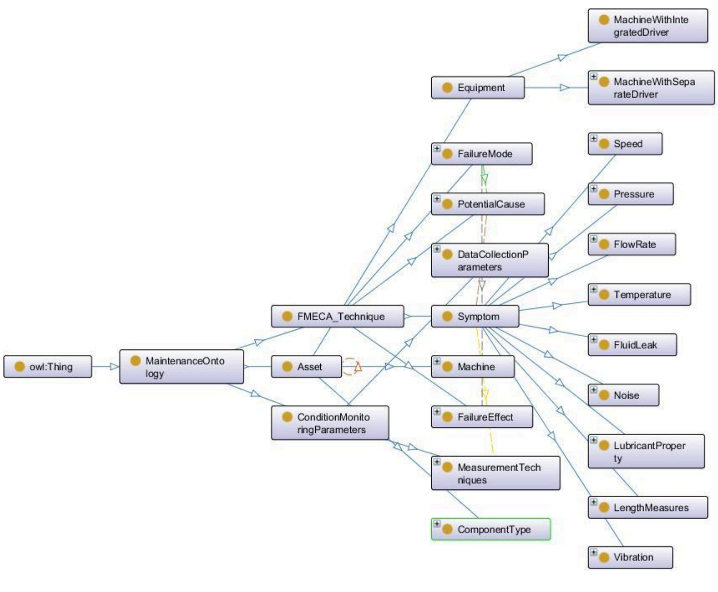

Fig. 3: Hierarchy of level 1, 2 and 3 classes

Table 1: Object Properties

Object Property	Domain	Range
HasFailureCause	FailureMode	PotentialCause
HasFailureEffect	FailureMode	FailureEffect
HasMeasurement	Measurement Techniques	Measurement Location
UseCollector	Measurement Location	CollectorType
UseMagnitude	CollectorType	Magnitude
IsPartOf	TestRigItem	TestRigItem
HasFailureMode	Component	FailureMode

3.6 Create instances

The creation of individual class instances involves: (1) selection of the class, (2) creation of an individual occurrence of the class and (3) filling slot values. These instances are used in the representation of particular elements. A class is selected for every instance in a way that binds the properties of the object, data and/or annotations.

4. IMPLEMENTATION

To test the applicability of the ontology a physical gearbox test rig available at Cranfield University laboratories was employed. This has been designed for emulating complex cases of misalignment, relevant to manufacturing and aerospace engineering assets (Fig. 4). Digital twinning of the gearbox is implemented in a local cloud-based deployment of an IoT platform (Thingsboard). The rig is instrumented with industry-grade sensing, data acquisition and networking with both edge and cloud-based computing support for a complete data process workflow. Data acquisition employs a data acquisition panel with 16 channels (ICP, ac, dc), including eight 24 bit ones, supporting 51.2kHz sampling rate, and anti-aliasing filters, with PLC interfaces and WiFi, LAN and 4G connectivity. The employed sensors are of ICP industry-grade type vibration sensors with 10kHz sampling frequency.

The intended use of the ontology at the next stage after the research presented in this paper is to serve the needs for deploying edge-driven and cloud-based monitoring services for this test rig. The current process involves the determination of measurement location points with a view to selecting the ones which are likely to provide informative data for detecting and quantifying various types of misalignment. Following this, the necessary functions of each component that enable the machine to operate correctly are determined. The FMEA technique is used to map failure modes, causes, effects, symptoms and measuring techniques appropriate for the given components and failure modes.

A typical usage scenario is that during the undertaking of condition monitoring, queries may be raised to resolve the context of the monitoring service. For example, this may involve the determination of possible failure modes for a component, the functional impact of the faults on the rig's operation, the measurement options appropriate for given faults and components, as well as measurement parameters associated with them. For the purposes of this implementation SPARQL queries were built to resolve such queries. SPARQL enables also federated queries over various data sources. Linking other relevant sources was not considered in the present work but is an option for potential extensions. Through the following query, we can find "what are the main components of a given mechanical machine?"

```
SELECT  ?ComponentType
WHERE {?ComponentType rdfs:subClassOf
as:ComponentType }
ORDER BY ?ComponentType
```

Fig. 5 shows the results of a query to identify the main components of an asset type. These components are bearing, coupling, lubricant, rotor, seals, and shaft. The present implementation allows a query in the maintenance ontology to resolve key analysis characteristics, such as components function, failure modes, causes, effects, occurrence, severity, detection and applicable measurement technique.

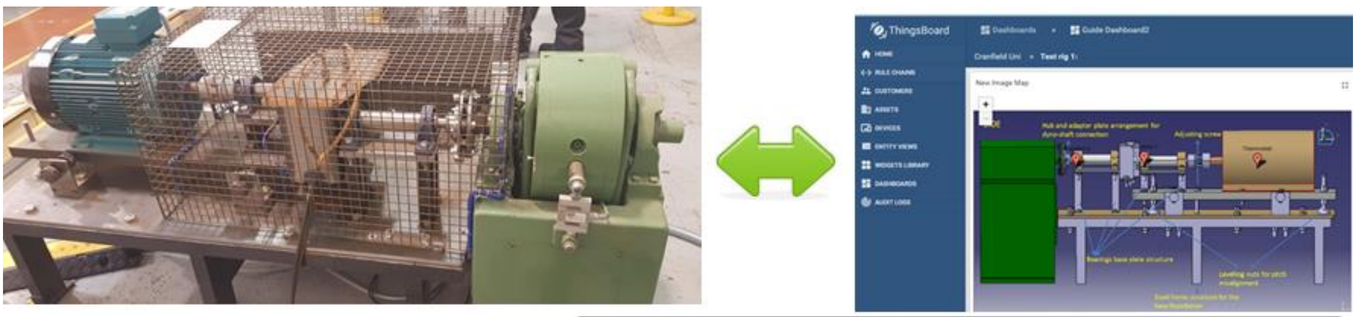

Fig. 4: The physical gearbox transmission test rig for emulating misalignment cases

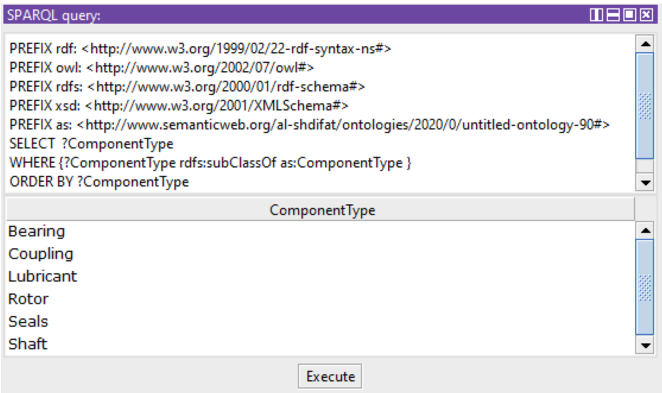

Fig. 5: Query result to identify the main components.

Table 2: Query result to identify the components functions

Component	function
Shaft	It transfers torsional power with the help of transmission components.
Bearing	Supports shaft and reduces friction
Motor	Convert energy into mechanical energy.
Dynamometer	Controls available torsional load on dynamometer.
Coupling	Its function is to connect two shafts.
Gears	To transmit shaft power on predetermined or designed angular velocities.
Lubrication	lubricating the teeth and bearing
Cooling system	To fill the engine's cooling system, to act as a heat exchange fluid.

Table 2 shows the results of a query to identify the functions of the main components of the test rig. A component that has high significance in failure analysis is the bearing (Table 3). The most critical failure mode is fracture and the typical failure mechanism for this is fatigue. Another query can be applied to determine failure modes, failure causes, failure effects, symptoms, and fault severity (SEV), but also determine the faults with highest diagnostic potential (DGN). SEV and DGN scale from 1 to 10, with the higher number representing the higher seriousness or risk.

Table 3: Query outcome for failure mode with highest DGN

Component Type	Failure Mode	Failure Cause	Failure Effect	SEV	Symptom	DGN	Technique
RollingBearing1	Fracture	Fatigue	Premature failure of contact surfaces	4	Vibration1	5	VibrationAnalysis

However, misalignment faults can be the primary causes of fatigue and in turn of bearing faults for this test rig and therefore applying condition monitoring for misalignment will be a key target for the next steps of the research.

5. EVALUATION AND RESULTS DISCUSSION

Several ontology evaluations have been proposed, which can take an implementation or a design viewpoint (Degbelo, 2017) (Kumar & Baliyan, 2018). The scope of the present case study was exploratory, i.e. the aim was to present the development of a context resolution service mechanism for industrial diagnostics, based on the design of a maintenance ontology focused on modelling failure analysis of mechanical components. Therefore, it was considered appropriate to focus on a subset of evaluation criteria, namely effectiveness, internal consistency, and applicability, within the viewpoint of the targeted application case study.

To assess the model functionality a number of queries have been constructed in SPARQL and tested on the ontology model. The process was considered satisfactory when all tests were shown to produce satisfactory responses for the given operation scenario. To assess the reasoning consistency the Pellet reasoned was employed. This verified the structure of the ontology's properties and that classes were implemented as specified. The results of the queries provide evidence for the effectiveness of the ontology in representing key concepts and the relationship between them in the employed test case. This verified the lack of conflicts or inconsistencies between the ontology properties, classes, and instances.

Although the OD1 procedure was implemented, other approaches can be applicable. A comprehensive ontology validation would require a thorough set of query test cases and the application of the ontology to other more complex and operational assets. The Pellet reasoner was applied in this work because it can detect inconsistencies and can verify the class hierarchies, range, domain and conflicting disjoint assertions. The Protegé editor provides a warning with a red triangular alert symbol when a consistency error occurs. The Pellet reasoner is subsequently activated in order to detect any inconsistency and it was employed here until making sure that no inconsistencies were present.

6. CONCLUSION

This paper presented a study for the development of an approach to developing a context resolution service for IoT-enabled industrial diagnostics. It has followed an established ontology development process but its design differs from other approaches in that it expands FMEA/FMECA – based ontology constructs with additional concepts adopted from available standards in the field that link the key reliability-based concepts of the knowledge constructs with asset level and fault – specific relevant diagnosability concepts. The ontology development was further applied on a physical mechanical transmission test rig and it intended to be used in the next phase of the research as the applied context resolution mechanism in condition monitoring. Context resolution is determined through a reasoning mechanism and the next aim is to apply this mechanism to enable a metadata producing mechanism to annotate events generated by automated and human-driven means. While the application focus is quite specific, the ontology abstraction level is actually such that it could also be implemented on other application cases, as it offers a sound baseline for further customisation or extensions. Consequently, further research should be carried out to link the current ontology implementation with a live condition monitoring service, as well as to apply it to real industrial environments as an enabler of more efficient IoT-enabled monitoring services.

REFERENCES

Al-shdifat, A. and Emmanouilidis, C. (2018) 'Development of a Context-aware framework for the Integration of Internet of Things and Cloud Computing for Remote Monitoring Services', Procedia Manufacturing. Elsevier B.V., 00, pp. 31–38.

Bousdekis, A., Magoutas, B., Apostolou, D. and Mentzas, G. (2015), "A proactive decision making framework for condition-based maintenance", Industrial Management & Data Systems, Vol. 115 No. 7, pp. 1225-1250.

Cândea, G., Kifor, S. and Constantinescu, C. (2014), "Usage of Case-based Reasoning in FMEA-driven Software", Procedia CIRP, Vol. 25 No. 0, pp. 93-99.

Cao, Q. Samet, A. Zanni-Merk, C. De Beuvron, F. Reich, C. (2019) 'An Ontology-based Approach for Failure Classification in Predictive Maintenance Using Fuzzy C-means and SWRL Rules', Procedia Computer Science, 159, pp. 630–639.

Degbelo, A. (2017). A Snapshot of Ontology Evaluation Criteria and Strategies. Semantics2017. (pp. 1–8).

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

Du, J., Wang, S. and Zhang, H. (2013) 'Layered clustering multi-fault diagnosis for hydraulic piston pump', Mechanical Systems and Signal Processing. Elsevier, 36(2), pp. 487–504.

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.

Frolov V., Mengel D., Bandara W., Sun Y., Ma L. (2010) Building an ontology and process architecture for engineering asset management. In: Kiritsis D., Emmanouilidis C., Koronios A., Mathew J. (eds) Engineering Asset Lifecycle Management. Springer, London

Gong, Y. Janssen, M., (2013) "An interoperable architecture and principles for implementing strategy and policy in operational processes", Comput. Ind. 64 (8) 912–924.

Guillén, A. J., Crespo, A., Gómez, J. F. and Sanz, M. D. (2016) 'A framework for effective management of condition based maintenance programs in the context of industrial development of E-Maintenance strategies', Computers in Industry, 82, pp. 170–185.

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.

ISO 55000. (2014). Asset management — Overview, principles and terminology

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.

Kamsu-Foguem, B. and Noyes, D. (2013), "Graph-based reasoning in collaborative knowledge management for industrial maintenance", Computers in Industry, Vol. 64 No. 8, pp. 998-1013.

Karray, M., Chebel-Morello, B., Lang, C. and Zerhouni, N. (2011), A component based system for s-maintenance, 9th IEEE International Conference on Industrial Informatics (INDIN), 26-29 July 2011, pp. 519-526.

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.

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.

Kumar, S., & Baliyan, N. (2018). Quality evaluation of ontologies. Semantic Web-Based Systems: Quality Assessment Models (pp. 19–50).

Lau, R. Li, C. Liao, S., (2014) "Social analytics: learning fuzzy product ontologies for aspect-oriented sentiment analysis, Decis". Support Syst. 65 80–94.

Matsokis, A. and Kiritsis, D. (2012), "Ontology-Based Implementation of an Advanced Method for Time Treatment in Asset Lifecycle Management", Mathew, J., Ma, L., Tan, A., Weijnen, M. & Lee, J. (Eds.) Engineering Asset Management and Infrastructure Sustainability, Springer London, pp. 647-662.

Noy, N.F., McGuinness, D.L., (2001) "Ontology development 101: a guide to creating your first ontology", Stanford Medical Informatics Technical Report.

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.

Ren, G., Ding, R. and Li, H. (2019) 'Building an ontological knowledgebase for bridge maintenance', Advances in Engineering Software. Elsevier, 130(March), pp. 24–40.

Teoh, P. C. and Case, K. (2004), "Failure modes and effects analysis through knowledge modelling", Journal of Materials Processing Technology, Vol. 153–154, pp. 253-260.

Uschold M, Gruninger M., (1996) "Ontologies: principles, methods and applications". Knowl Eng Rev; 11:93–136.

Yuan, Y., Jiang, X. and Liu, X. (2013) 'Predictive maintenance of shield tunnels', Tunnelling and Underground Space Technology. Elsevier Ltd, 38, pp. 69–86.

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.

2020-12-18

Ontology – based context resolution in internet of things enabled diagnostics

Al-Shdifat, Ali

Elsevier

Al-Shdifat A, Emmanouilidis C, Khan M, Starr AG. (2020) Ontology – based context resolution in internet of things enabled diagnostics. IFAC-PapersOnLine, Volume 53, Issue 3, 2020, pp. 251-256
https://doi.org/10.1016/j.ifacol.2020.11.041