Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2017) 55-65

ECCOMAS
Proceedia

UNCECOMP 2017
2nd ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Rhodes Island, Greece, 15–17 June 2017

DATA–DRIVEN IDENTIFICATION, CLASSIFICATION AND UPDATE OF DECISION TREES FOR MONITORING AND DIAGNOSTICS OF WIND TURBINES

Imad Abdallah, Vasilis K. Dertimanis, and Eleni N. Chatzi

Institute of Structural Engineering, ETH Zürich
Stefano-Franscini-Platz 5, 8093 Zürich, Switzerland
e-mail: abdallah,v.derti,chatzi@ibk.baug.ethz.ch

Keywords: Wind turbines, data–driven updating, probabilistic decision trees, quantitative risk analysis, O&M, object–oriented design

Abstract. *This paper describes a conceptual framework for the online monitoring of wind turbines (WTs) relying on an object–oriented (OO), event–and–decision tree–driven platform for information processing and propagation. To this end, a WT is viewed as a multilayered system of objects (e.g. structure, controller, actuator, etc.) that are defined on the basis of abstract superclasses, attributed with specific properties and methods. The former generally provides insight about the current state of the respective object, while the latter communicates state information and determines the interaction among objects and events. The term state refers herein to a set of mutually exclusive "positions", which a specific object may reach (e.g. safe, critical, fail, etc.), while an unknown state is also included in order to take into account possible combinations of events that have not been registered during the design phase and would eventually be identified in the decision tree through the real–time telemetry. The envisioned platform is purely probabilistic, e.g. a set of probabilities is initially assigned to all events and updated accordingly, based on actual information extracted from the WT. This information may either be acquired using sensors (through corresponding sensor objects), or may be estimated using appropriate algorithms (through corresponding methods, such as Kalman-based filters). A paradigm of the proposed conceptual framework focuses on the tower substructure of the WT and indicates the potential of the proposed approach, especially in respect to the design of specialized software for monitoring and diagnostics of both new and existing WT installations.*

Peer-review under responsibility of the organizing committee of UNCECOMP 2017.
doi: 10.7712/120217.5351.17092

1 INTRODUCTION

Despite the fact that wind turbines (WTs) constitute one of the most important, continuously evolving and expensive structures in modern engineering [1, 2], their maintenance and inspection is still succeeded using conventional methods [3], such as visual inspection (planned and unplanned), non-destructive evaluation (i.e., crack detection using ultrasound technologies) and post–processing of SCADA data. Specialized methods are only focused on specific components, as for example the condition monitoring (CM) of elements of the drive train (i.e., gearbox and main bearing) [4], while structural health monitoring (SHM) systems are deployed mostly for research purposes, or only during the certification stage, and are far from forming part of the actual engineering practice [5]. As a result, there exist currently no systematic, quantitative and automated tools for monitoring, detection and diagnostics of WTs, for operation, maintenance (O&M) and decision making within their life-cycle.

In identifying possible reasons for this discrepancy, one may argue that state–of–the–art CM and SHM methods are still not straightforward to adopt, when O&M engineers are in fact seeking robust and reliable decision making tools that compress data in user–friendly output formats. They are also by default "local", in the sense that they are configured for monitoring only a specific WT component (e.g. the blade, the gearbox, the tower etc.), and they generally lack cross communication and interaction capabilities. As a result, in this way, interconnecting links or the monitoring of critical events among the components of a WT is a non-trivial task.

A remedy solution to this problem may be provided via implementation of event and decision trees. Event trees are traditionally used in quantitative risk analysis of engineering systems, laying a path from an initiating event to an end state of a system. An initiating event may correspond to a sudden increase in wind speed (gust) or an earthquake tremor, and the corresponding end states may pertain to critical interference of a wind turbine blade with the tower surface or a crack in the concrete foundation, respectively. For a given initiating event, multiple end states are possible. Intermediate chronological events make up the branches leading from the initiating event to the end state, and each event is associated with a probability of occurrence.

Under this perspective, previous studies have presented general purpose solutions that are based on event and decision trees [6, 7, 8], as well as on Bayesian networks [9]. However, their common feature is that they only correspond to the design phase of the system and, once formulated, they remain static. In this work, we propose real-time updating of decision trees as a decision-making tool for O&M that integrates live data (or estimates) from the individual components of WTs. Decision trees are preferred over Bayesian Networks, as the former can be easily updated from data, they are visually more appealing and simpler to interpret, while it is easier to follow and track an event path, a way that follows the sequence and chronology of how events are interlinked.

On the basis of our proposed supervisory platform, (i) events are classified and event probabilities are updated via real time telemetry from the WT; and (ii) new initiating events and end states are identified. In the first step, a set of condition and/or structural data samples can be trained using decision trees, which are initially provided by engineers. Then, by running new data through the trees, classification and prediction may be carried out. Data that does not fit the existing decision tree structure is used to identify new initiating events and end states. We stipulate that such a framework can be used for real-time monitoring and diagnostics of structures, root cause analysis of future failures and quantitative risk analysis in the context of operation and maintenance scheduling of components. The use of decision trees is motivated by the fact that they tend to be easier to interpret than other quantitative data-driven methods such

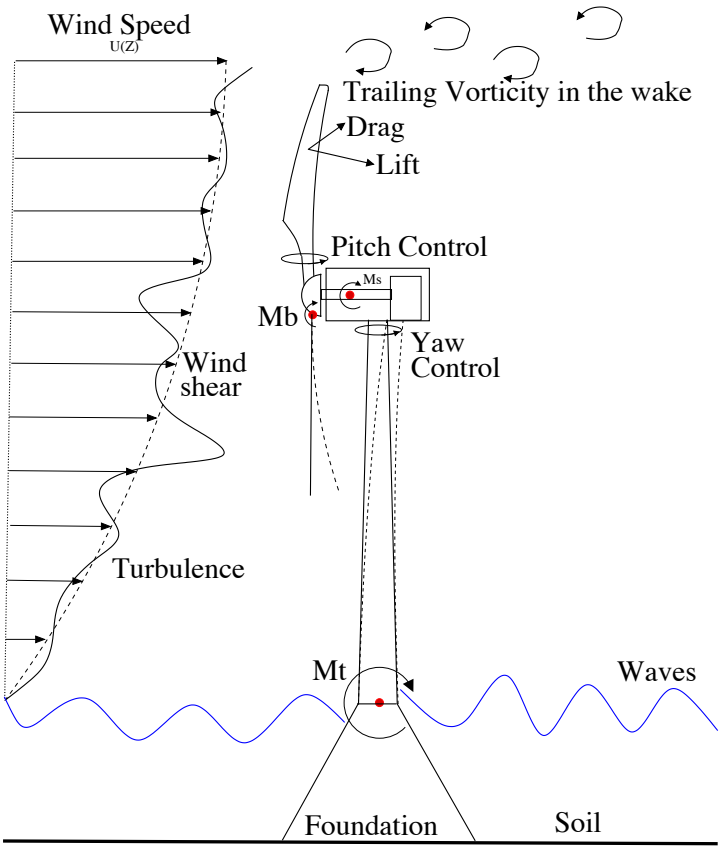

Figure 1: Schematic of a wind turbine and its interaction with the environment.

as Bayesian networks (Bayesian networks may link more variables in complex, direct and indirect ways, rendering interpretation oftentimes problematic) [10]. Furthermore, the learning and updating process of decision trees from real–time data is a far more mature field as compared to the training of Bayesian Networks [11].

Our envisioned approach follows an object–oriented (OO) architecture. This implies that all the aforementioned features are developed as corresponding properties and methods of abstract classes and interfaces, instances of which are used to interpret a multi–layered diagram of WT objects and the interactions among them. Decision trees are embedded locally in a specific object (i.e., the tower, the controllers, the blade, etc), which is essentially a finite state machine. Under this layout, possible CM and SHM tools are also locally registered and they are used to determine the current state of an object and/or notify neighbouring objects for the occurrence of an event or a change in state.

2 PROBLEM FORMULATION

2.1 The wind turbine

Figure 1 sketches a representative wind turbine (WT) facility and its interaction with the environment. The WT is exposed to wind and wave loads, which act primarily on the structural components, i.e., the foundation, the tower and the blades. While the wave motion is a typically uncontrolled (and often unmeasurable) form of structural excitation, the wind speed and direction are essentially responsible for the production of energy. The latter is achieved via the effective cooperation of the pitch and yaw control systems that optimally adjust the angles of the blades and the nacelle.

Component	Function
Rotor blades	Convert aerodynamic forces into rotational motion
Drive train	Converts the rotational motion of the rotor blades into electric power, and consists of main bearing, main shaft, gearbox, brake, coupling and generator
Nacelle and main frame	Houses and supports the drive train and transfers the loads from the rotor to the tower, respectively
Tower	Erects the rotor up in the air, and insures that the loads are transferred from the rotor down to the foundations
Foundation	Ensures load bearing capabilities and support of the wind turbine (resist overturn)
Controllers	Ensure that the turbine operates within the structural design limits, while maximizing power extraction from the wind
Sensors	Measure quantities critical for operation and monitoring (e.g. wind speed & direction, pitch and yaw angles, bearing's acceleration, etc.)
Actuators	Transfer the control commands to the structural components (e.g. nacelle and blade hub)
Safety systems	Override control commands to ensure safety and structural integrity of the wind turbine

Table 1: The main components of a WT and their functionality.

A modern WT is composed of several interrelated subsystems that fulfill specific tasks. The most important of these are listed in Table 1, along with their main functionality. It must be noted that each subsystem consists of a number of subcomponents itself, rendering the WT a complex multi–level engineering system. Indicatively, the drivetrain consists of a frame construction that supports the main bearing, main shaft, the gearbox, a brake and a high–speed coupling, as well as the generator that transforms the mechanical motion into electrical energy. It is thus apparent that the structural and operational integrity of the facility are strongly dependent on the constitutive sub–components of each sub–system.

2.2 The monitoring problem

Under this setting, the problem considered herein pertains to the establishment of a robust, data–driven supervision platform for the monitoring of WTs over their whole life-cycle, which combines three main drivers:

[D1] An effective tool attributed with the ability of tracking, identifying and classifying indi-

vidual events based on real–time telemetry from components and propagating their implications to the global state.

[D2] The integration of a probabilistic framework, aiming at projecting the current state of all individual components to the future, and providing both short and long–term predictions of the local and global safety margins.

[D3] Flexibility of the tool to integrate new components (e.g. new sensors, new controller, etc.) in the context of evolving structures.

Due to the inherent complexity of the monitored system, the envisioned supervision platform should be further attributed with adaptive tools that may create sequences of events that were not predicted during the design stage. The conceptual framework of such a platform is outlined in what follows.

3 THE SUPERVISORY MONITORING PLATFORM

3.1 Description

The proposed monitoring framework is based on the OBEST paradigm [6, 7] and its predecessor [8], in which engineering systems and their constitutive components are represented as objects (e.g. instances of predefined classes) that are interlinked and exchange information with respect to their current state. OBEST is composed of four different views that describe the functionality of a system in various levels. These are the *system structure diagram*, the *interaction diagram*, the *state transition diagram* and the *data flow diagram*. In our implementation we currently focus on the first three, as the latter is strongly dependent on the availability of measured information from the WT, rather than to an explicit scheme that is defined during the design stage of the WT.

The system structure and interaction diagrams describe the constitutive objects of the system and their interaction, respectively. For a WT, a merged view of both is displayed in Fig. 2, where the boxes correspond to the individual WT components and subsystems and the arrows to the interactions among them. In general, a piece of information received by an object results in altering its internal state or a subset of its fundamental characteristics (attributes). This accordingly implies the transmission of another piece of information that communicates its new condition to further objects. Thus, the behaviour or state of each component in the system is entirely encapsulated within the confines of an object and the performance of the entire WT is represented by combining and connecting object models for individual components or subsystems.

Each component/subsystem has an internal representation that is modelled using discrete, *mutually exclusive* states and corresponding state transition events (refer to Sec. 4 for a visual interpretation of such a diagram). The initial state transition trees are herein provided by the component engineer/designer. However, unlike in traditional quantitative risk assessment and system reliability, the initial (a-priori) event-state transition trees are further updated by actual measurements of the WT system; we propose to perform the updating using a decision tree learning approach (ID3, CART, C4.5/5.0, Random forests, Naive Bayes, etc.) [12].

3.2 Data–driven probabilistic updating

A novelty of our framework is the integration of Structural Health Monitoring (SHM) data, such as strains and accelerations, complementary to Condition Monitoring (CM) ones such as

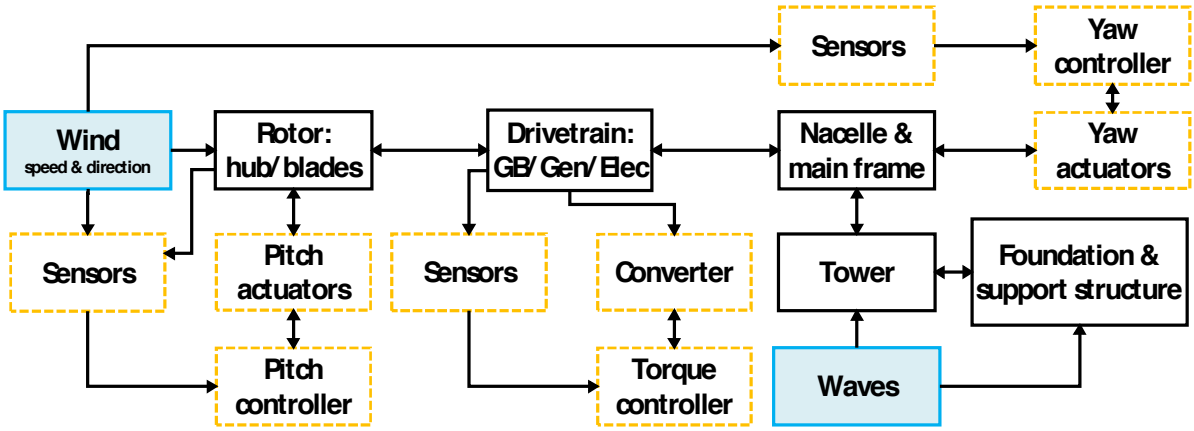

Figure 2: A simplified WT structure and interaction diagram (combined).

control signals, status logs, temperature, etc. when monitoring a structure and diagnosing any faults or failure. This is observed in Fig. 2, where the Sensors object (see below) receives both wind speed and strain measurements as inputs from the blade and subsequently forwards this information to the pitch control algorithm object. Both condition and structural signals are then propagated through the pitch control object internal event tree until an end state is reached and an output message is accordingly transmitted to the pitch actuators object.

The updating aspect of the framework consists in identifying new branches in the event tree and classifying initially unknown end states. Such end states could be, for example, operating modes such as normal, critical, abnormal, fault, failure, and unclassified. As the continuous stream of telemetry is propagated through the event trees of each component, classifications and predictions are continually accomplished. Decision trees are admittedly not the the most competitive means of classifying telemetric data or for assessing conditional probabilities in complex systems, but they offer the benefit of straightforward interpretation by non-expert operators via visualization of the chains of events following an accidental end state. Another advantage of decision trees is related to diagnosis, which is the task of locating the source of a system fault once it is detected by tracing the sequence of events in the tree that lead to fault or failure [13].

Such an object oriented framework together with updating through decision tree learning is appealing in the context of a system that is continuously evolving/ageing in time, which is particularly true for a wind turbine system. For instance, ageing and degradation of the blade composites or the gearbox bearings introduce new and previously unaccounted for states, which could be captured via decision tree learning. Furthermore, wind turbines are operating under significantly diverse environmental conditions, hence the resulting ageing process and the faults appearing may not be identical across similar systems. The decision tree learning offers the possibility to infer new and distinct states from each of operating turbine. Finally, a common scenario involves renewal and replacement of components on wind turbines; for instance ten years into the operation of a wind turbine, the operator may decide to introduce a new pitch control algorithm/features for load reduction, which then introduce new operating modes (events) and states. In the context of the proposed object oriented framework for monitoring and diagnostics, this can be readily achieved and seamlessly integrated with the already existing components (objects).

The framework we propose is probabilistic in nature. The first possibility here is to use the decision tree, where the conditional probability of an end state $EndState_i$ depends on the

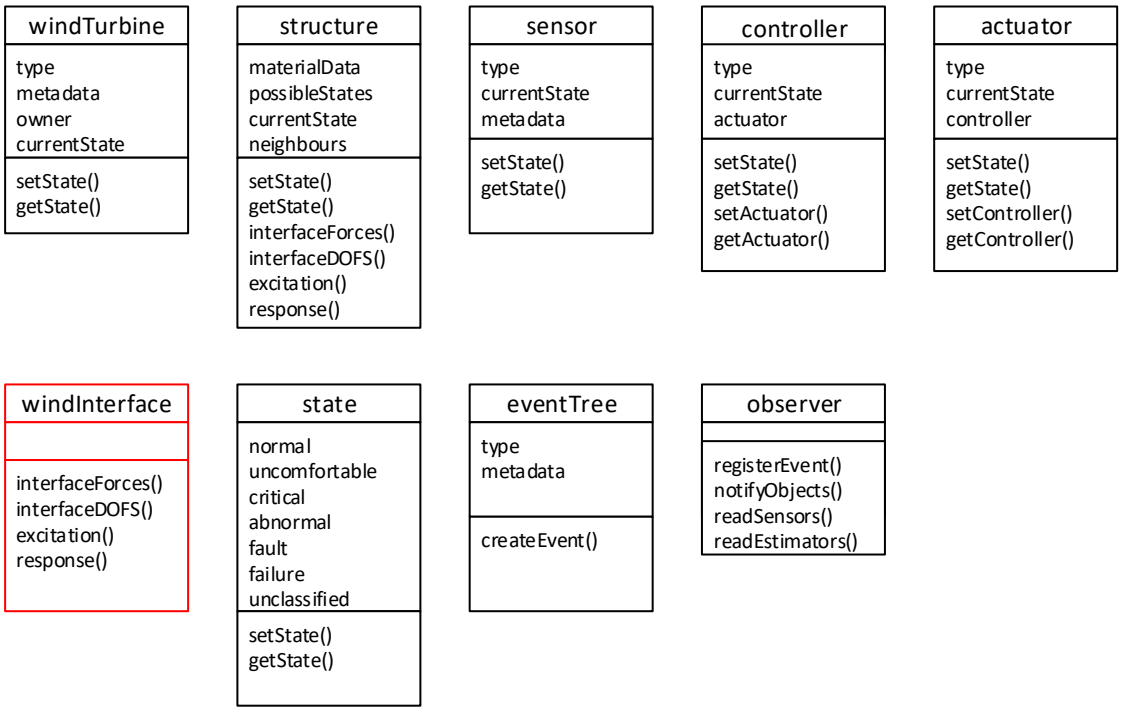

Figure 3: Indicative classes and interfaces (in red) of the supervisory platform.

conditional probabilities of the preceeding events e_i on a path such that:

$$P\left(EndState_i\right) = P\left(e_1\right) \cdot P\left(e_2 \mid e_1\right) \cdot P\left(e_3 \mid e_2, e_1\right) \ldots \tag{1}$$

The limitation appears when the component displays a behaviour with feedback (i.e., after a repair or update in the system) or for evolving systems (e.g. when new sensors are integrated or aging of the system), which implies a need to establish several decision trees based on the possible ordering of the events or based on new initiating events. One way around this is to convert decision tree learners to undirected Markov networks [14] or make use of hidden Markov decision trees [15, 16]. An alternative is to map the decision trees learners into Bayesian Networks for further assessment of the conditional probabilities [17]. The conditional probabilities of the end states are a means to predicting the likelihood of potential fault or failure occurring. Such a probabilistic approach would inherently include uncertainties stemming from measurements.

For systems comprising several elements, the number of different combinations of inspection and maintenance could be large [18]. The possibility to update the conditional probabilities of end states and intermediate events based on real-time telemetry facilitates pre-posterior type of analysis (i.e., assessment of the value of data before they become available), and allows for optimal planning of inspection and maintenance on the structure.

3.3 Objects

An extensive outlining of the individual classes that define the objects of the WT subsystems falls outside the aim of the current study. However, our framework relies on a number of abstract class and interface definitions, which may be easily inherited to other pending subclasses. These are illustrated in Fig. 3 and follow the conventional object oriented design pattern. In the current stage of our conceptual interpretation, these are separated into two broad categories: the first corresponds to classes directly related to the WT and its constituent subsystems (e.g.

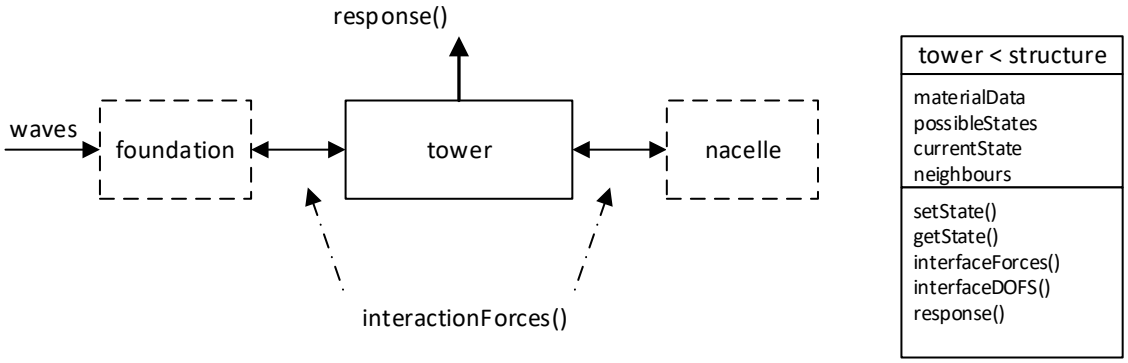

Figure 4: Interaction diagram for the tower substructure. Tower extends class *structure*.

windTurbine class, *structure* class, *controller* class, etc.), while the second category defines the aspects of the supervisory platform that correspond to interactions, events and information flow (e.g. associated interfaces and *eventTree*, *state* and *observer* classes).

The specified properties and methods of the definitions of Fig. 3 are indicative and may include any individual method that is integrated into the monitoring platform. For example, an adopted SHM method that may be used for tracking the remaining fatigue life of s structural element can be placed within the setState() method of the structure class. Similarly, a Kalman filter that may be established for estimating unmeasured states required for the controller class can be registered within the readEstimators() method of the observer class.

4 EXAMPLE: THE TOWER SUBSTRUCTURE

As an indicative example of our framework, we focus on the tower component of the WT. A tower object is an instance of the tower class, which is a child of the structure superclass and inherits its properties and methods. Figure 4 illustrates the adopted interaction diagram. It is assumed that both waves and wind do not contribute in a significant way to the excitation of the tower. Thus, its structural response is only affected by the interaction forces that are developed with its neighbouring components, the foundation and the rotor-nacelle assembly. The latter are also instances of the corresponding foundation and nacelle classes, again children of the structure superclass.

Fig. 5 displays an indicative event-state transition tree for the tower. The properties of the state class of Fig. 3 are clearly indicated, along with events that have been specifically defined for this object. Notice the presence of an *Unclassified* state and *Unknown events* in Fig. 5. The decision tree would, based on the real–time telemetry from the tower, learn those unknown events and the corresponding unclassified states. In a fully probabilistic framework, all the events among the different states must be attributed with corresponding conditional probabilities. The latter are initially assigned during the design stage and updated accordingly with respect to the actual structural behaviour based on the real-time telemetry. The latter could be achieved by combining both sensors, which provide a direct indication of the structural response at the measurement points, and estimators, which can be utilized in order to estimate the structural response in unmeasured spots along the tower [19]. The term estimator could refer to, but is not restricted to, Kalman filters, surrogate models, and reduced time marching simulators which could be run in near real-time.

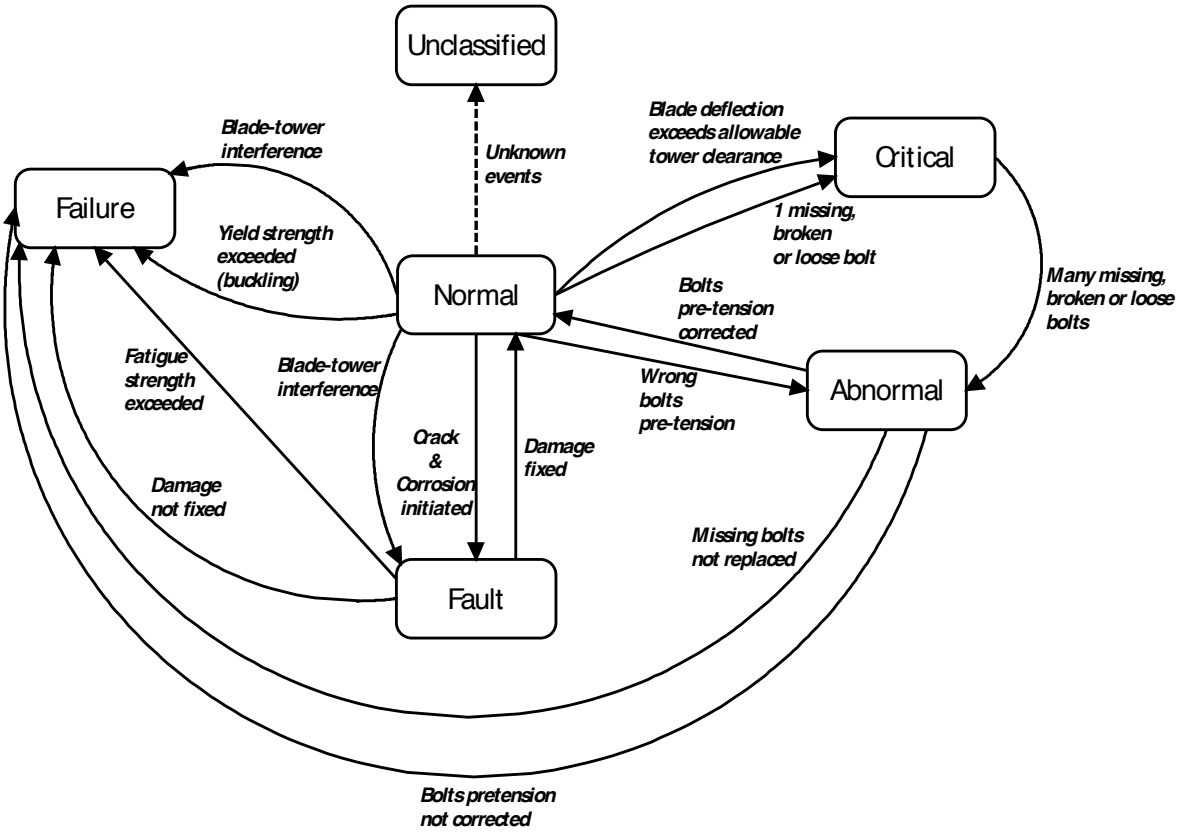

Figure 5: An event–state transition tree of the WT tower.

5 CONCLUSIONS

We provided a conceptual description of a monitoring and diagnostics framework of wind turbines. The framework combines three main drivers:

[D1] tracking, identifying and classifying individual components' decision trees based on real–time telemetry and propagating their implications to the global state

[D2] real-time updating of conditional probabilities of events and end states of each component aiming at providing both short and long–term predictions of the safety margins

[D3] flexibility to integrate new system components (e.g. new sensors, new controller, etc.) in the context of evolving structures

The above drivers are rendered possible by the OO architecture of the framework, where each component (object) is defined as an extension of an abstract superclass. Two broad families of superclasses are defined, the first pertaining to the wind turbine and the second relating to interaction, events and information flow. We illustrate the framework with an example case-study involving the tower object.

ACKNOWLEDGEMENTS

The authors would like to gratefully acknowledge the support of the European Research Council via the ERC Starting Grant WINDMIL (ERC-2015-StG #679843) on the topic of Smart Monitoring, Inspection and Life-Cycle Assessment of Wind Turbines.

REFERENCES

[1] European Commission. Energy road map 2050. Technical report, European Union, Luxembourg, 2011.

[2] Fraunhofer Institute for Wind Energy and Energy System technology. Wind energy report. Technical report, IWES, Germany, 2011.

[3] B. Yang and D. Sun. Testing, inspecting and monitoring technologies for wind turbine blades: A survey. *Renewable and Sustainable Energy Reviews*, 22:515–526, 2013.

[4] Z. Hameed, Y. S. Hong, S. H. Ahn, and C. K. Song. Condition monitoring and fault detection of wind turbines and related algorithms: A review. *Renewable and Sustainable Energy Reviews*, 13:1–39, 2009.

[5] F. Grasse, V. Trappe, S. Thoens, and S. Said. Structural health monitoring of wind turbine blades by strain measurement and vibration analysis. In *Proceedings of the 8th International Conference on Structural Dynamics (EURODYN)*, Leuven, Belgium, 2011.

[6] G. D. Wyss and F. A. Durán. OBEST: The object based event scenario tree methodology. Technical Report SAND2001-0828, Sandia National Laboratories, Albuquerque, CA, USA, 2001.

[7] G. D. Wyss, F. A. Durán, and V. J. Dandini. An object-oriented approach to risk and reliability analysis: Methodology and aviation safety applications. *Simulation*, 80(1):33–43, 2004.

[8] G. D. Wyss, R. L. Craft, and D. R. Funkhouser. The use of object-oriented analysis methods in surety analysis. Technical Report SAND99-1242, Sandia National Laboratories, Albuquerque, CA, USA, 2001.

[9] D. Koller and A. Pfeffer. Object-oriented bayesian networks. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 302–313, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[10] S. Bayat, M. Cuggia, D. Rossille, M. Kessler, and L. Frimat. Comparison of bayesian network and decision tree methods for predicting access to the renal transplant waiting list. *Stud Health Technol Inform*, 150:600–604, 2009.

[11] J. Cheng, R. Greiner, J. Kelly, and D. Bell. Learning bayesian networks from data: An information-theory based approach. *Artificial Intelligence*, 137:43–90, 2002.

[12] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, USA, 2006.

[13] A. X. Zheng, J. Lloyd, and E. Brewer. Failure diagnosis using decision trees. In *Proceedings of the First International Conference on Autonomic Computing, ICAC '04*, pages 36–43, New York, USA, 2004.

[14] D. Lowd and J. Davis. Improving markov network structure learning using decision trees. *Journal of Machine Learning Research*, 15:501–532, 2014.

[15] M. I. Jordan, Z. Ghahramani, and L. K. Saul. *Hidden Markov decision trees*, volume 9 of *Advances in neural information processing systems*, pages 501–507. MIT Press, Cambridge, MA, USA, 1997.

[16] J. L. Trivino-Rodriguez and R. Morales-Bueno. MPSG. a unified view of markov chains and decision trees. Technical report, Department of Languages and Computer Sciences, University of Málaga, Málaga, Spain.

[17] G. Bearfield and W. Marsh. Generalising event trees using bayesian networks with a case study of train derailment. In *International Conference on Computer Safety, Reliability, and Security, SAFECOMP 2005*, pages 52–66, Fredrikstad, Norway, 2005.

[18] D. Straub, V. Malioka, and M.H. Faber. A framework for the asset integrity management of large deteriorating concrete structures. *Structure and Infrastructure Engineering*, 5:199–213, 2009.

[19] S. Eftekhar-Azam, V. Dertimanis, E. Chatzi, and C. Papadimitriou. Output-only schemes for joint input-state-parameter estimation of linear systems. In *UNCECOMP 2015 - 1st ECCOMAS Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering*, pages 497–510, 2015.