An Approach to Supporting Maintenance of Offshore Wind Turbine Blades

Trinh Hoang Nguyen[1], Andreas Prinz[1], Josef Noll[2]
[1]Department of ICT, University of Agder
Grimstad, Norway
[2]ConnectedLife Group, University Graduate Center
Oslo, Norway
{trinh.h.nguyen, andreas.prinz}@uia.no, josef@unik.no

ABSTRACT: *Offshore wind turbine blades suffer from various faults such as blade angle asymmetry, icing, and bends. Replacement of rotor blades normally involves heavy transportation (e.g., vessel and crane), and dependency on weather conditions. As a result, wind turbines come to a long standstill and costs on energy production increase. One approach to reducing the downtime and costs is to apply a proper maintenance strategy, e.g., corrective, time-based, failure-based, condition-based, or reliabilitycentered maintenance. This article introduces an approach, which combines the knowledge-based approach with the force analysis technique, to supporting maintenance of offshore wind turbine blades. The approach solves the semantic ambiguity of offshore wind information and provides the possibility of monitoring the performance of wind turbine blades in real-time, leading to advanced alarms when needed. The approach can be adapted and applied to other wind turbine components.*

Keywords: Wind Turbine Blades, Knowledge-based Approach, Semantic Ambiguity, Offshore Wind Blades

Received: 19 July 2013, Revised 2 September 2013, Accepted 12 September 2013

1. Introduction

Using renewable energy to meet the future electricity consumption and to reduce environmental impact is a significant target of many countries around the world. Wind power is one of the most promising renewable energy technologies. In particular, the development of offshore wind power is increasing rapidly since there is a high potential in harvesting the wind at sea. Offshore wind energy continues to gain momentum. Indeed, for offshore wind power plants built from 2007 to 2009, the capacity rates from 2 to 5 MW [37]. Offshore wind turbines can access higher-quality wind resources and are larger than onshore ones in terms of size and capacity. The cost of offshore wind energy is still high due to involvement of heavy transportation (e.g., crane and vessels) for installing and repairing wind turbines, replacement of equipment, and operation and maintenance (O&M). The O&M cost alone constitutes up to 30% in offshore installations [12]. The main purpose of O&M is to lower the cost on repair and replacement of equipment by preventing malfunctions of equipment, providing early detection of equipment failures, and minimizing system downtime. In order to reduce the O&M cost, appropriate maintenance strategies need to be applied.

In general, maintenance strategies are classified into two main groups: proactive maintenance and corrective maintenance [22]. The difference between these two strategies is that proactive maintenance is carried out before failures occur, while corrective maintenance is carried out after failures occur [22], [30]. Proactive maintenance can be divided into preventive maintenance (or scheduled maintenance) and predictive maintenance which is referred to as condition-based maintenance (CBM). Preventive maintenance is carried out according to an established time schedule. Condition-based maintenance is carried out when maintenance tasks are initiated in response to a specific system condition. The CBM has a dominant position in maintenance strategies

There are some challenging issues which are involved in preventive maintenance such as under-maintenance and overmaintenance [36]. Over-maintenance happens when, for example, a replacement of an equipment is not necessary, but the replacement is carried out anyway according to a maintenance schedule. Under-maintenance occurs, for example, when a wind turbine component is about to be damaged, but operators are not aware of it due to the lack of a component monitoring system and it is not the time to carry on the maintenance yet. These issues increase the costs of maintenance while associated failures do not decrease.

On the other hand, CBM utilizes condition-based monitoring systems (CMS) for equipment. Basic CMS techniques are vibration analysis, acoustic emission, strain measurement, and oil analysis [18], [14]. CMS is a part of the reliability maintenance for a wind turbine. It continuously monitors the performance of wind turbine components and helps determine the optimal time for specific maintenance, hence improving maintenance management and increasing reliability [30]. Besides, CMS can predict when components of a wind turbine are likely to fail months in advance. Maintenance teams therefore can make an optimal schedule for maintenance [4].

This article[1] introduces an approach, namely knowledgebased force analysis, to supporting maintenance for offshore wind farm blades. The approach will solve the semantic ambiguity of offshore wind information and provide the possibility of monitoring the performance of wind turbine blades in real time and generate advanced alarms when needed. The rest of the article is organized as follows. Section II describes the structure of a wind turbine rotor and its associated failures focusing on wind turbine blades. An information model of a wind turbine rotor and how to represent wind turbine rotor knowledge are presented in section III. Section V describes our proposed approach. Section VI presents a working system that is used to prove the concepts. Section VII discusses related work. Finally, section VIII concludes the article with some remarks on future work.

2. Offshore Wind Turbine Rotors

Wind turbine blades capture kinetic energy of the wind and convert it into mechanical energy. The wind turbine rotor is turned by a lift that is generated when the wind passes over the blades. The rotor is connected to the main shaft that drives a generator to produce electric power. The wind turbine rotor consists of a turbine hub and blades. Modern wind turbines fall into two primary designs: horizontal axis wind turbine blades (single bladed, double bladed, three bladed, multiple bladed), and vertical axis wind turbine blades (straight blade and curved blade). Blades can have separate pitch control systems or centralized pitch control system or neither of them. The purpose of having a pitch control system is to optimize the power output that can be produced.

Common fault modes of a wind turbine include gearbox oil over-temperature, blade angle asymmetry, pitch thyristor fault, and yaw runaway [24]. Repairs to the generator, drive train, hub, gearbox, and blades often caused standstill periods of several weeks [17]. This is caused by immature repair technology as well as transportation issues involved in the repair. Even though there are many technologies available today, the replacement of wind turbine rotors is still a challenge, especially for offshore wind turbine components since the replacement involves heavy transportation (vessel, crane) and dependency on weather conditions.

Typical faults of wind turbine rotors are blade surface roughness, damages of a blade’s surface painting, icing, cracks, breakups, and bends [7]. The reasons that cause these faults are various, for example, dirt, dead bird/insects, blowholes, drain holes, turbulent wind, out-of-control rotation, lightning, and production defects. These faults negatively affect the wind power output due to reduction of aerodynamic performance of the blade [9]. Mass imbalance is a major source of vibration since it is proportional with wind speed in square (v^2) [8]. If the masses balance, the absolute value of the centrifugal forces F_{ci} are the same for each blade and the three force vectors will add to zero [10], i.e. $m_1 * r_1 = m_2 * r_2 = m_3 * r_3$ or $F_{c1} = F_{c2} = F_{c3}$ thus

$$\overrightarrow{F}_{c1} + \overrightarrow{F}_{c2} + \overrightarrow{F}_{c3} = 0 \quad (1)$$

[1]This article is a revised and expanded version of a paper entitled, “*Removed for review*” presented at INTECH 2013 conference, 29 - 31 August, London, UK. ISBN: 978-1-4799-0047-3

where m is the mass of the blade, and r is the distance from the blade's center to the rotor axis. The rotor balance is broken if one of the parameters m_i, r_i (i = 1, 2, 3) changes.

Despite the fact that there are many improvements in blade construction, it remains true that every blade in operation is vulnerable to wear and fatigue [28]. Faults of gearboxes and blades are of particular interest to the wind industry due to their cost to repair [23], especially in the offshore wind with big wind turbines and large blades located far away from the shore. The faults often cause a long downtime and result in high cost including expenses on transportation and decrease in power production. Thus the rotor blade is one of the most critical wind turbine subassemblies which should be monitored [26]. It is necessary to have strategies that provide the possibility of monitoring, planning and scheduling for inspection and maintenance, as well as alarming and predicting failures in order to reduce unexpected damages.

3. Wind Turbine Rotor Information Representation

An offshore wind software environment normally involves different applications (applications for monitoring, for analysis, and for predicting) and different sources with a variety of formats. When it comes to collecting, organizing, and sharing information between these applications, some problems will occur such as data availability, and format incompatibility. It is therefore difficult to enable data exchange and knowledge sharing. Moreover, the semantics of the data is not exploited completely.

3.1 Wind turbine rotor information

In order to ease data exchange within the wind energy, the International Electrotechnical Commission (IEC) has proposed the IEC 61400-25 standard [1]. The standard is entitled "*Wind turbine - Communications for monitoring and control of wind power plants*". The standard defines a basic data exchange structure for wind power plant components. Wind turbine rotor information (WROT), as shown in table 1, is one of classes defined in the standard. STV stands for status value and MV stands for measured value. The formats of these values are defined in the IEC 61850-7-3 standard.

WROT CLASS		
Attribute name	**Attribute type**	**Explanation**
Inherit all mandatory data from common class provided in the IEC 61400-25-2		
RotSt	STV	Status of rotor
BlStBl1	STV	Status of blade 1
BlStBl2	STV	Status of blade 2
BlStBl3	STV	Status of blade 3
PtCtlSt	STV	Status of pitch control
RotSpd	MV	Rotor speed
RotPos	MV	Angular rotor position
HubTmp	MV	Temperature in the rotor hub

Table 1. An Example of Wind Turbine Rotor Information [1]

3.2 Knowledge representation

A human being can easily understand the semantics of concepts of a domain, but this is not the case for a machine. Resolving semantic heterogeneity not only helps machines understand the domain concepts, but also gives users a unified way to view the distributed data. There are three well known forms of domain knowledge representation that allow domain knowledge to be expressed in a semantic way. These are semantic network, rules, and logics [15]. The last one provides a precise semantic interpretation utilizing both forms of the former two.

A **semantic network** (e.g., RDF) is a graph which involves nodes and links between nodes. Each node represents a domain concept while a link denotes a relation between two domain concepts. Figure 1 represents a semantic network for wind turbine rotor (WROT). WROT, Rotor, RotorSpeed, etc. represent the concepts of the wind domain while *isA*, *isPartOf*, *has-Speed* represent the relations between the concepts. Semantic networks use structure representations to express statements about a domain of interest. Even though relations between concepts are well defined in the semantic network, there is a problem when using semantic network to represent knowledge. For example, it is not clear how many blades are part of Rotor.

In a **rule-based** approach, IF-THEN constructs are used to express various kinds of statements. An example is shown as follows.

(a) IF *Blade BL*101 is a *HAWTBlade* THEN
*Blade BL*101 is a *Blade*

(b) IF *Blade BL*101 is a *HAWTBlade*
AND *wrot* is a WROT
AND *rotor* is a WROT*Rotor*
AND *Blade BL*101 is a part of *rotor*
AND *r* is a part of *wrot*
THEN *Blade BL*101 is a part of *wrot*

Rule-based knowledge representation systems are especially suitable for reasoning about concrete instance data [15], for example, in the second rule example, *Blade BL*101, *rotor*, *wrot* are concrete instance data. However, it is hard for humans to read when the rules are getting more complicated.

The **logic-based** approach gives more precise semantics since it utilizes both the structure representation and rules. Description logics (DLs) are an example of the logic-based knowledge representation approach. DLs are used to provide high level ontologies [21]. DLs are associated with two components, terminological box (TBox) and assertional box (ABox). TBox contains sentences describing concept hierarchies. In other words, it contains axioms about classes. ABox contains assertions about individuals. The knowledge base (KB) consists of TBox and ABox. Here are some examples of how to expressing concepts and individuals in ABox and TBox.

(1) *HAWTBlade is Blade* (*HAWTBlade* $\sqsubseteq$ *Blade*) belongs in the TBox, provided that *HAWTBlade and Blade* are concepts of an ontology.

(2) *Blade BL*101 is a *blade* (*Blade BL*101 $\in$ *Blade*) belongs in ABox, provided that *Blade* is a concept of an ontology.

(3) *Rotor* has exactly three b*lades (Rotor* $\sqcap$ = 3.*hasBlade*), provided *Rotor* is a concept and *hasBlade* is a role that belongs in TBox.

3.3 Ontology representation

Ontology is defined as a specification of a conceptualization. Concepts, properties, relations, functions, constraints, and axioms of a particular domain are explicitly defined in ontologies [16]. There are several knowledge representation languages such as OIL, DAML-ONT, DAML+OIL, RDF, and OWL [25]. Among them Resource Description Framework (RDF) and Web Ontology Language (OWL) are being used intensively by research communities as well as industry. OWL is an extension of RDFS, in the sense that OWL uses the RDF meaning of classes and properties [20], [5], [2]. In OWL, Owl:Thing is a built-in most general class

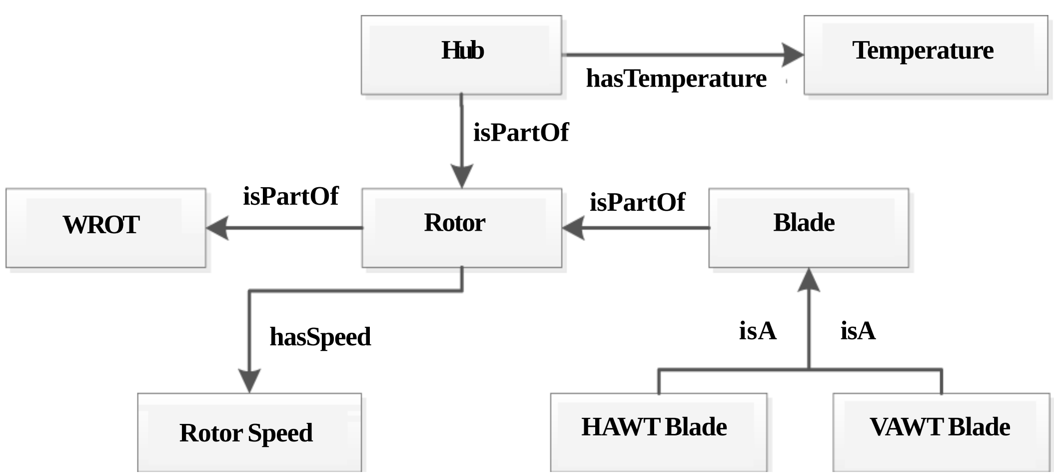


Figure 1. A semantic network representation of WROT

and is the class of all individuals. It is a superclass of all OWL classes. A class defines a group of individuals that belong together. Individuals can be referred to as being instances of classes. Classes are a concrete representation of concepts. Owl:Nothing is a built-in most specific class and is the class that has no instances. It is a subclass of all OWL classes. Properties in OWL are also known as roles in description logics and relations in Unified Modeling Language (UML). OWL has two types of properties: object property and datatype property. An object property relates individuals to other individuals (e.g., *hasComponent* relates *WROT* to *WROTRotor*). A datatype property relates individuals to data type values, e.g., *hasStatus*, *hasP itchAngleSetP oint*. OWL has three sublanguages: OWL Lite, OWL DL, and OWL Full. Among these sublanguages, OWL DL is more expressive than OWL Lite, but less expressive than OWL Full. OWL DL is a syntactic variant of the description logic $SHOIN^{(D)}$ [35]. $SHOIN^{(D)}$ includes concept operators (and $\wedge$, negation $\neg$, existential $\exists$, union $\vee$, universal $\forall$), transitive roles, role hierarchy, inverse properties, nominals, unqualified cardinality restriction, and datatype support. In this work, OWL DL is selected to represent the wind turbine rotor ontology.

4. Real-Time Monitoring

In order to reduce critical failures of wind turbine components, real-time monitoring systems for components need to be employed at operation centers. Monitoring systems involve fast and automatic data exchange between wind power plants and the operations center and accessibility to necessary information in real time, as well as integration of distributed wind data. Making information available through web services increases the accessibility to information due to platform independence and easy access. A web service is defined by the World Wide Web Consortium (W3C) as a software system identified by a URI (Unified Resource Identifier) and it employs XML (eXtensible Markup Language) to describe its public interfaces and bindings.

There are two types of web services, XML-based web services and Representational State Transfer (RESTful) web services. The big web services are based on XML, SOAP, WSDL (Web Service Definition Language), and other technologies specified in the Web Services Interoperability (WSI) basic (Carlin and Abusabal, 2009). The REST architecture was introduced by Roy Fielding in 2000. RESTful provides services over the Internet through the Web browser by using the four CRUD operations (create, read, update, delete) corresponding to four HTTP methods: GET, POST, PUT, DELETE. The most important REST [11] principle is to expose the resources in a RESTful service as unique URIs. Stateless in RESTful services means that no state should be stored on the server between requests from the client. Each request should therefore contain all the information necessary to serve the client. For each request, there is a status code sent along with the response for indicating the result of the response. A REST-based design provides a unified way of organizing and accessing data over many different mediums, enabling mashups. It fits to the Semantic Web scheme which also uses URIs as resource identifiers. Furthermore, all common operations on the Semantic Web with the exception of query - data fetch, insertion, and deletion - are the fundamental operations in a REST-based system [3].

Enterprise Service Bus (ESB) is considered a communication backbone between application. It provides transports, events, and mediation services to facilitate the integration of large-scale heterogeneous applications [27]. ESB is based on loosely couple architecture and offers a common integration platform. Besides, ESB provides a publish/subscribe feature that allows users to subscribe to data sources exposed through web services and get updates from data source providers without sending requests periodically. This work utilizes advantages of both ESB and RESTful web services to support real-time monitoring of wind turbine components at operation centers.

5. A Knowledge-based Force Analysis Approach

This section describes our approach which is based on semantic technologies and force analysis technique. Semantic technologies are used to build a knowledge base for WROT, while force analysis technique is employed to detect abnormality that occurs in wind turbine blades.

5.1 An ontology for wind turbine rotors

Based on the idea of using the IEC 61400-25 standard as a source of wind domain concepts to build an offshore wind ontology and the strategy presented in [29], we introduce an ontology for WROT. A class diagram of the WROT ontology is shown in Figure 2.

MainComponent and SubComponent are complementary classes that classify components of WROT in different levels. *MainComponent* contains all the components that have direct connection to LLN (Logical Node). *SubComponent* contains all the components that have a connection to a component in *MainComponent* or have a connection to other components in *SubComponent*. Each blade can have a pitch control system to adjust the pitch angle to the wind speed. A pitch control system is a subclass of control systems. Pitch control systems system is a subclass of control systems. Pitch control systems can be classified into passive stall power control and active stall power control systems [32].

We use Protege[2] to build the WROT ontology. Protege is an open source ontology editor. It allows users to work with both RDF and OWL. Figure 3 depicts an example of how to define the relationships between *WROTRotor* class and its components in Protege. *WROTRotor* is part of WROT and object property *hasComponent* is used to describe their relationships.

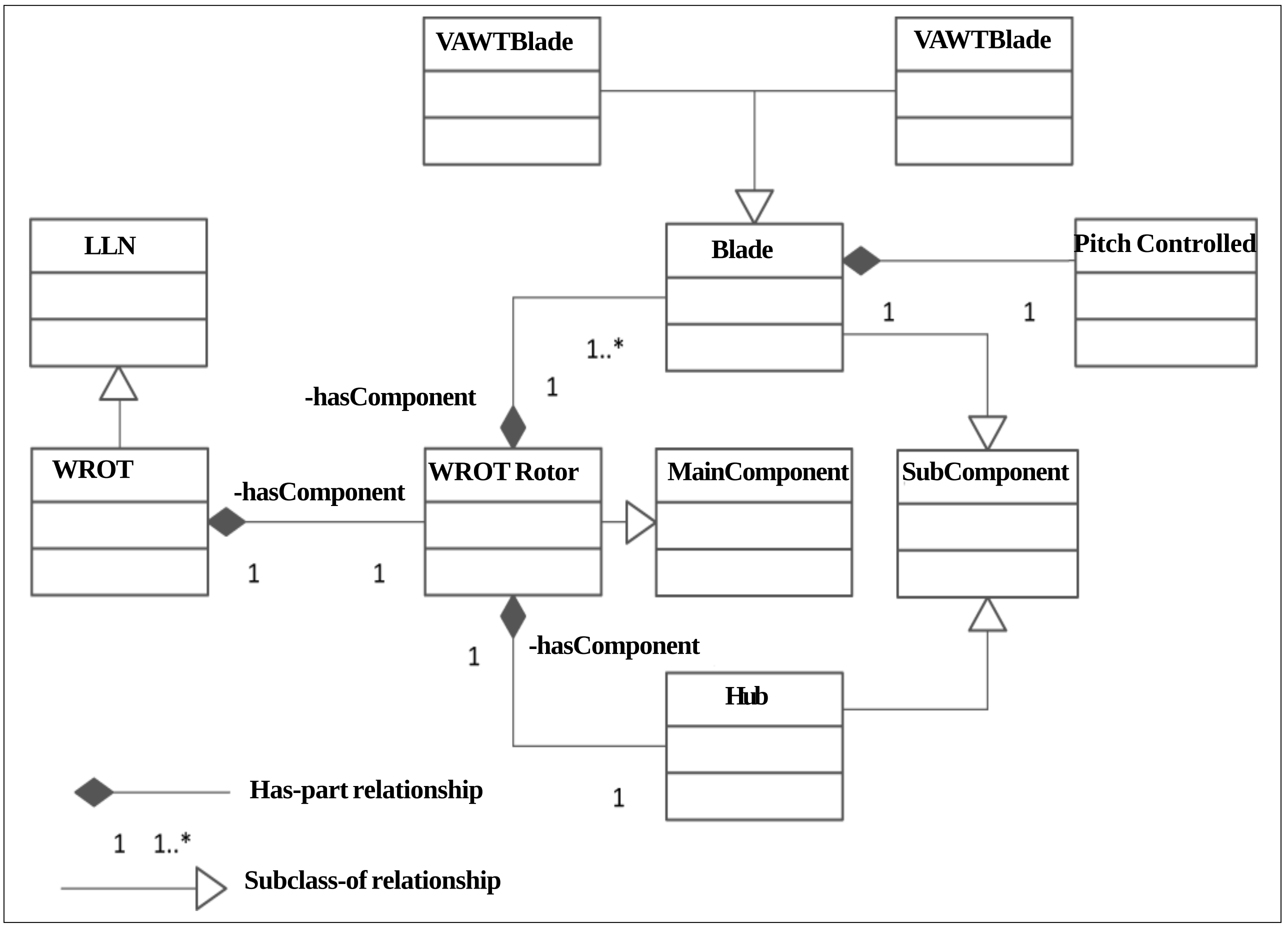


Figure 2. UML class diagram of WROT

hasComponent is an object property that relates *WROTRotor* class to *Hub* class. Since each WROT has only one hub, we use cardinality restriction "*exactly*" to clarify the relationship. A WROT might have more than one blade, therefore we use the constraint "*some*" on hasComponent property.

5.2 Centrifugal force analysis

Typically wind turbine blade fatigues are caused by different forces, including the gravitational force caused by the pull of the Earth on the mass of the blade and the centrifugal force due to the rotation of the rotor blade. The centrifugal force causes blade stretching, bending and torsion. It is one of the main loads on a wind turbine blade [34]. In a normal state, as stated in (1) the three force vectors will add to zero. However, the aerodynamic balance will be broken if there is a change in one of these three force vectors caused by an additional mass *m* on one of the wind turbine blades. The centrifugal force caused by the mass is considered as an additional force. It can be measured by a force sensor embedded on a wind turbine blade. However, in this work, we calculate the centrifugal force based on the available rotor speed.

[2] *http://protege.stanford.edu*

We assume that a schedule for blade maintenance has been made. What should wind turbine operators do if there is an additional mass *m* detected on a tip of the blade? Shall the wind turbine be stopped immediately for cleaning? Is there any possibility for operators at the operations center to check whether or not it is possible to let the wind turbine continue operating and the blade maintenance will be carried out as scheduled or the maintenance plan must be rescheduled?

The idea is that the rotor should be serviced after a certain accumulated force $Acc(F)$. An accumulated force is the force that stresses on a blade over a certain period. If the accumulated predicted force exceeds $Acc(F)$ and this happens before a scheduled maintenance, an alarming signal will be sent to the operators at the wind farm operations center. The signal will indicate when the event likely happens so that maintenance can be timely scheduled or rescheduled. Figure 4 and Figure 5 illustrate our point.

Figure 3. WROTRotor class is a subclass of a class which is a MainComponent class and has Blade and Hub as components

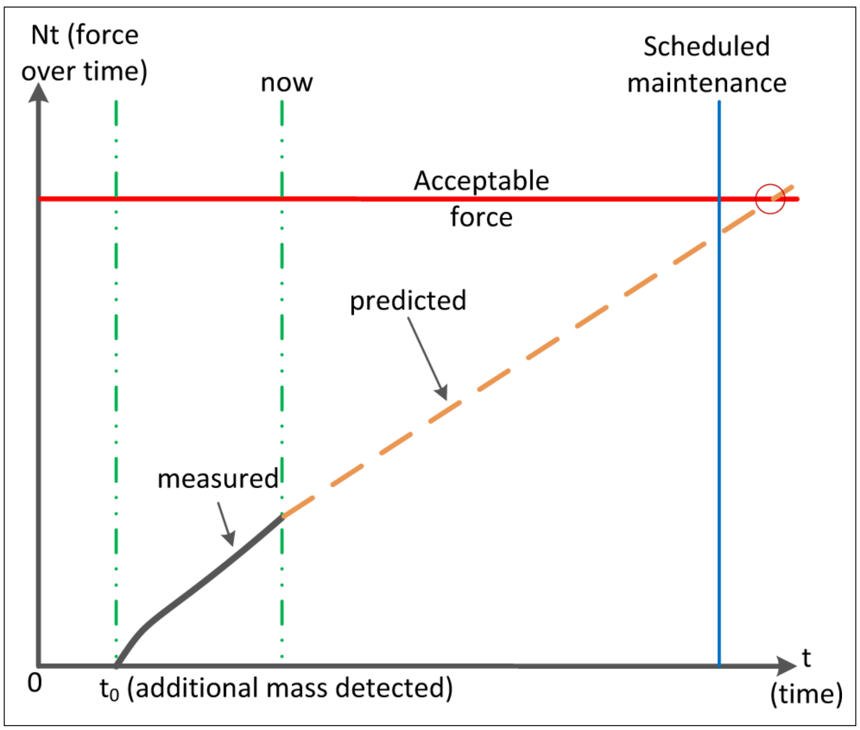


Figure 4. The wind turbine can continue operating and the maintenance can be carried out as scheduled

Figure 4 shows a normal case where the predicted accumulated force exceeds the normal force after the scheduled maintenance. Figure 5 shows a case where the accumulated predicted force exceeds the acceptable force before the scheduled maintenance. It means that the maintenance needs to be rescheduled. In this case, an alarm will be sent to the operations center so that operators can reschedule the maintenance plan.

Theoretically, the centrifugal force can be obtained by the following equation:

$$F_{cf} = \frac{mv^2}{r} \tag{2}$$

where *m* is the mass, *r* is the radius, *v* is the velocity. If the mass is located on the tip of the blade, *v* is the blade tip speed and *r* is the blade length. The blade tip speed can be calculated from the rotational speed and the length of the blade using (3).

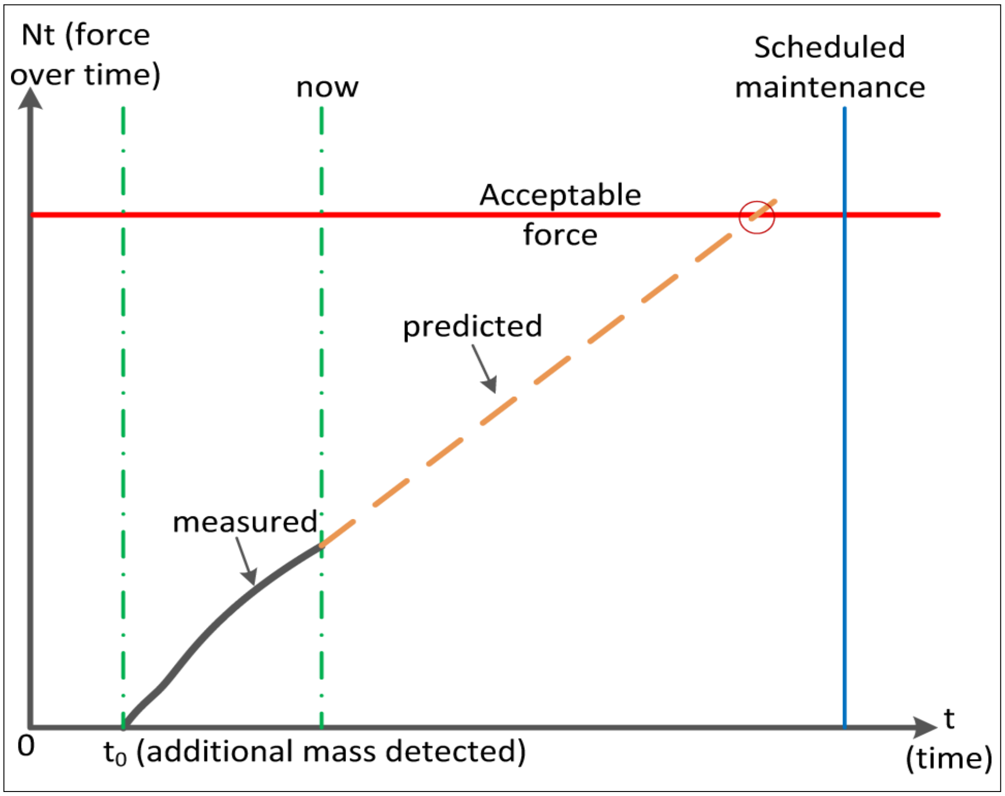


Figure 5. The maintenance needs to be rescheduled

$$v_{tip} = \frac{RPM * \pi * D}{60} \quad (3)$$

where RPM is the rotational speed, and D is the diameter of the turbine; $D = 2 * r$.

The blade tip speed can also be obtained through the tip speed ratio (TSR) as (4), where λ is TSR.

$$v_{tip} = \lambda * v_{wind} \quad (4)$$

Besides the measured force (F_{cf_mes} based on rotational speed), we define two more kinds of centrifugal force. These are acceptable F_{cf_acpt} and predicted centrifugal force F_{cf_pred}. The predicted force is calculated based on the average wind speed prediction.

The acceptable force that stresses on a blade over a period is called accumulated acceptable force $Acc(F_{cf\_acpt})$. We assume that this force is provided in the blade specification by the manufacturer of the blade. If the accumulated centrifugal force stressed on a blade over a period exceeds the accumulated acceptable one, the possibility of having a blade fault is very high.

Based on the measured and predicted forces we calculate the accumulated predicted force $Acc(F_{cf\_pred})$. The calculation of accumulated predicted force consists of two parts. The first part is calculated based on measured data from the beginning to the current point. The second part is based on the predicted force from the current moment to the end of the considered period. (5a) describes the calculation of the force.

$$Acc(F_{cf\_pred}) = \int_{start}^{now} F_{cf\_mes} + \int_{now}^{end} F_{cf\_pred} \quad (5a)$$

$\int_{start}^{now} F_{cf\_mes}$ is the measured accumulated force and can be calculated as follows:

$$\int_{start}^{now} F_{cf\_mes} = \sum_{i=0}^{now} \int_{i}^{i+1} f_i(x)\,dx \quad (5b)$$

where $\int_i^{i+1} f_i(x)\,dx$ can be calculated as follows:

$$\int_i^{i+1} f_i(x)\,dx = \frac{1}{2}(n_i + n_{i+1}) \tag{5c}$$

where n_i, n_{i+1} are the corresponding centrifugal forces at the measurement points i, $i + 1$, respectively and

$$n_i = \frac{m * RPM_i^2 * \pi^2 * D^2}{60^2 * r} \tag{5d}$$

$\int_{now}^{end} F_{cf\_pred}$ can be calculated as follows:

$$\int_{now}^{end} F_{cf\_pred} = (end - now)\, F_{cf\_pred} \tag{5e}$$

where F_{cf_pred} is calculated based on the average wind speed forecast v_{wind} as follows:

$$F_{cf\_pred} = \frac{m * (\lambda * v_{wind})^2}{r} \tag{5f}$$

Therefore (5a) can be rewritten as follows:

$$Acc(F_{cf\_pred}) = \sum_{i=0}^{now} n_i - \frac{1}{2}(n_0 + n_{now})\,(end - now)\, F_{cf\_pred} \tag{6}$$

Through these equations we can compare the accumulated predicted and acceptable additional force caused by a mass on a wind turbine blade. The output of the comparison is an alarm signal to the operations center in case the overflow happens before the scheduled maintenance date.

6. Implementation

As a proof of concept, an intelligent system has been developed. The system allows users to continuously monitor performance of the wind turbine rotor system. It also provides advanced alarms based on the centrifugal force that stresses on a wind turbine blade.

6.1 System architecture

Figure 6 illustrates the architecture of the developed prototype.

1) **System description:** On the server side, we use Mule ESB[3], an open source lightweight integration framework for enterprise services. Mule ESB handles publish/subscribe services to enable real-time monitoring of wind turbine blades. Restlet[4] is a high-level API based on the HTTP servlet technique. It provides an abstraction of REST applications, resources, and data representations. Applications developed using Restlet can run on any Servlet engine. The Jess[5] rule engine and the Pellet[6] reasoner are employed to execute rules and reason over the ontology. The output of the reasoner will be used to decide whether or not an alarm needs to be triggered. Real-time data can be collected directly from sensors embedded on WPP or through a SCADA (Supervisory Control and Data Acquisition) system. The rotor speed data is provided by an onshore wind energy database from Statkraft AS. On the client side, Ajax (Asynchronous JavaScript and XML) technology is used to handle the messages from the server and display information on a graph. Flot[7], a JavaScript plotting library is used to produce graphical plots on a web browser. A JAX-WS (Java API for XML Web Services) client plays the role of a data source (sensors, database, etc.) to send data to our system using SOAP/REST messages. A UML sequence diagram of the prototype is shown in Figure 7.

[3]http://www.mulesoft.org

[4]http://www.restlet.org/downloads/

[5]http://herzberg.ca.sandia.gov/

[6]http://clarkparsia.com/pellet/

[7]https://code.google.com/p/flot/

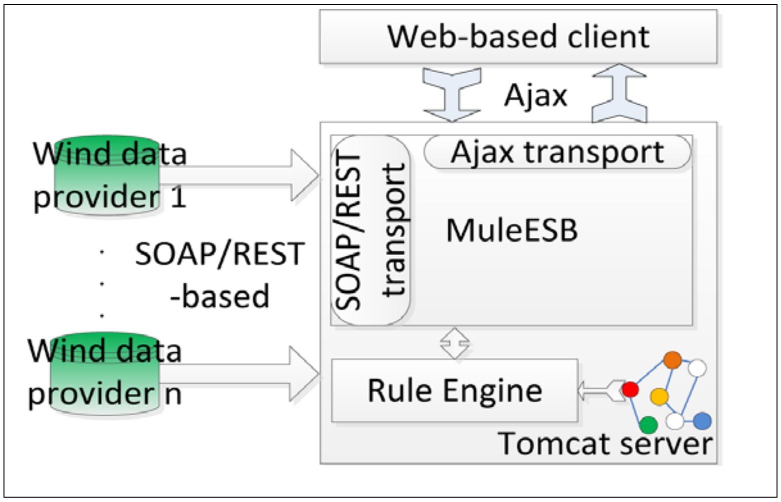


Figure 6. The architecture of the prototype

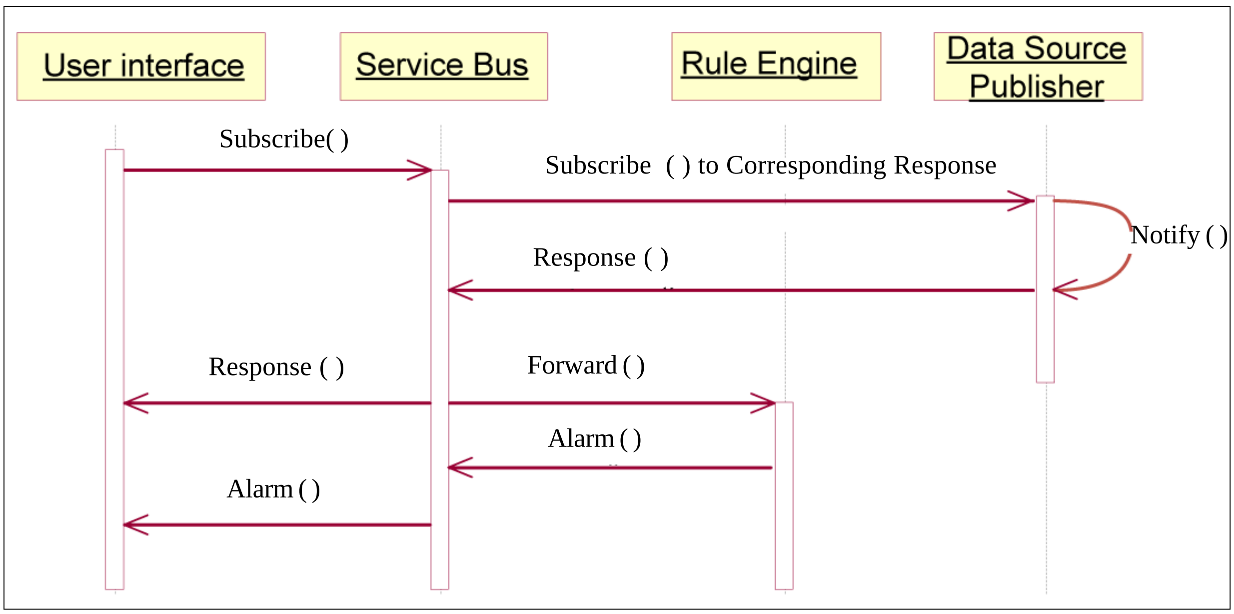


Figure 7. UML sequence diagram of the prototype

2) **Rule design:** Semantic Web Rule Language (SWRL) is developed based on OWL and Rule Markup Language (RuleML) [19]. It is a formal description logic-based extension of OWL. SWRL includes a high-level abstract syntax for Horn-like rules in the both OWL DL and OWL Lite sublanguages. We use SWRL to build semantic rules for the WROT ontology. SWRL rules reason about OWL individuals in terms of OWL classes and properties [31]. In our system, rules are used for executing commands such as shut down the power plant when the wind speed exceeds 25 *m/s* or trigger an alarm to notify operators if there is something wrong with blades. Some examples of SWRL rules are presented as follows.

Example 1: If the wind speed exceeds 25 *m/s*, the wind turbine rotor needs to be stopped.

$$WROT\,(?wrot) \wedge hasComponent\,(?wrot, ?r) \wedge hasSpeed\,(?r, ?sp) \wedge$$
$$swrlb : greaterThan(?sp; 25) \rightarrow shutdownWPP(true)$$

Example 2: If the accumulated force on a wind turbine blade exceeds 30000 *N*, an alarm needs to be triggered to inform operators at the operations center. Value *f* is provided in the blade specification.

$$WROT(?wrot) \wedge hasComponent(?wrot, ?r)$$
$$\wedge\ hasBlade(?r, ?b) \wedge hasAccumulatedForce(?b, ?f)$$
$$\wedge swrlb: greaterThan(?f, 30000) \rightarrow hasTriggerAlarm(?wrot, true)$$

6.2 Experiment and analysis

Given that all blades are identical, then the lift generated on each blade would be the same at a given angle on the hub. Let the rotor blade radius $r = 72\ m$. Let the average wind speed in May be 8 *m/s* and let TSR equal 7 since in general three bladed wind turbines, TSR is between 6 and 8 with 7 being the most widely reported value [33]. What would happen if one of the blade tips has one kg of a death bird[8]?

Figure 8 shows the relation between rotor speed and the centrifugal force. The graph allows users to have visual monitoring of the rotor blade.

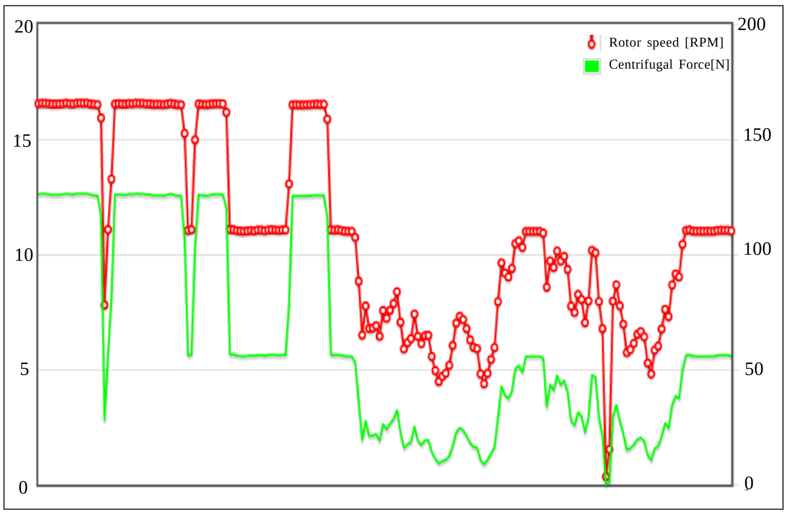


Figure 8. Rotor speed and centrifugal force

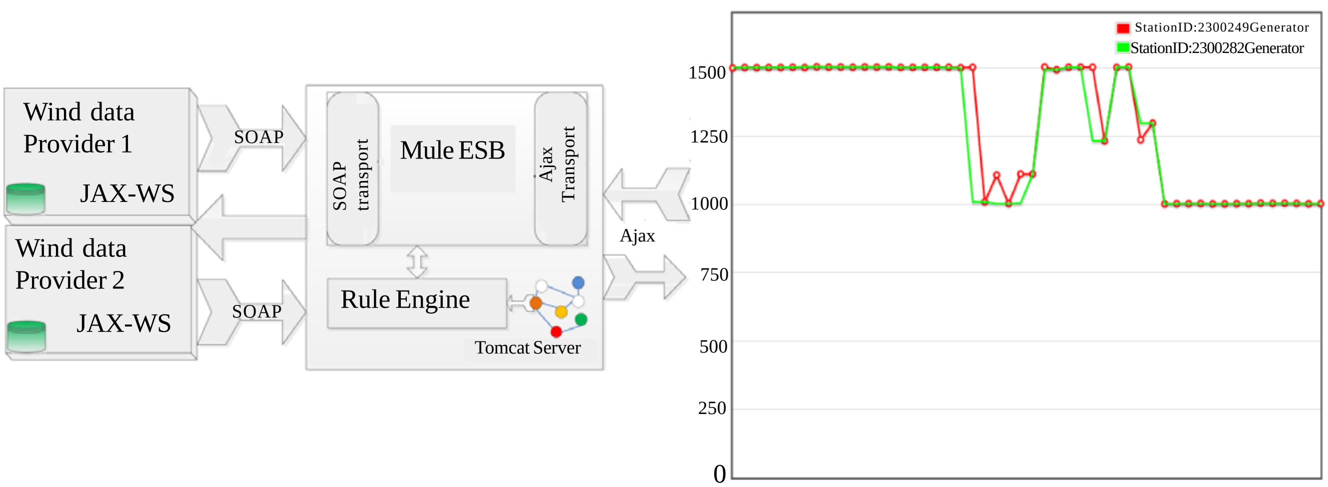
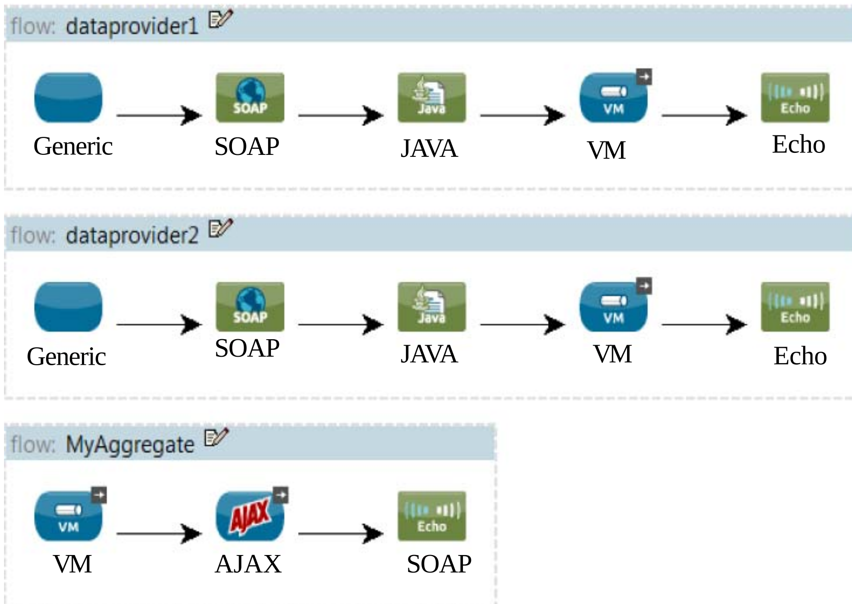


Figure 9. Real-time monitoring on generator speed of two wind power plants

1) **Scenario 1:** This scenario describes a case when one wants to compare performance of two different wind power plants in real time. In this scenario, we assume that data of wind power plants comes from two different sources. A snapshot of the implementation is shown in Figure 9. Generator speed data of two different wind power plants is used in this scenario.

[8]No birds were harmed in the production of this work.




```
<flow name =“dataprovider1" doc:name =“dataprovider1">
<inbound-endpoint address =“http://128.39.201.43:65080/services/
winddata”/>
<cxf:jaxws-service port =“80"
serviceClass =“no.uia.smile.wind.winddataprovider”/>
<component doc:name =“Java”>
<singleton-object class =“no.uia.smile.wind.winddataprovider”/>
</component>
<vm:outbound-endpoint exchange-pattern =“one-way” path =“vm.aggregator”
responseTimeout =“10000"
mimeType =“text/plain” doc:name = “VM”/>
<echo-component doc:name =”Echo”/>
</flow>
<flow name =“dataprovider2" doc:name =“dataprovider2">
<inbound-endpoint address =“http://128.39.201.14:65082/services/
winddata” />
...
</flow>
<flow name =“MyAggregate” doc:name =“MyAggregate”>
<vm:inbound-endpoint exchange-pattern =“one-way” path =“vm.aggregator”/>
<ajax:outbound-endpoint channel =“/services/WindEnergyChannel”
connector-ref =“AjaxConnector” />
</flow>
```

Figure 10. Mule configuration for two data sources

The configuration of the flows is shown in Figure 10. There are 3 flows in this scenario. The first and second ones show data coming from sources via SOAP messages. After that, data is pushed in a queue VM. VM is the in-memory transport that can be used for communication between Mule flows. In these flows, VM acts as an outbound-endpoint. VM in the third flow acts as an inbound-endpoint. Two outbound-endpoints VM in the first two flows and one inbound-endpoint VM in the third flow share the same queue. Data is taken from the inboundendpoint VM and sent to the client who subscribes to the data.

2) **Scenario 2:** Figure 11 shows a case in which the accumulated predicted force exceeds the accumulated acceptable one after the scheduled maintenance. It means that there is no need for changing the maintenance plan.

3) **Scenario 3:** Let us take a look at another case where the accumulated predicted force exceeds the accumulated acceptable one

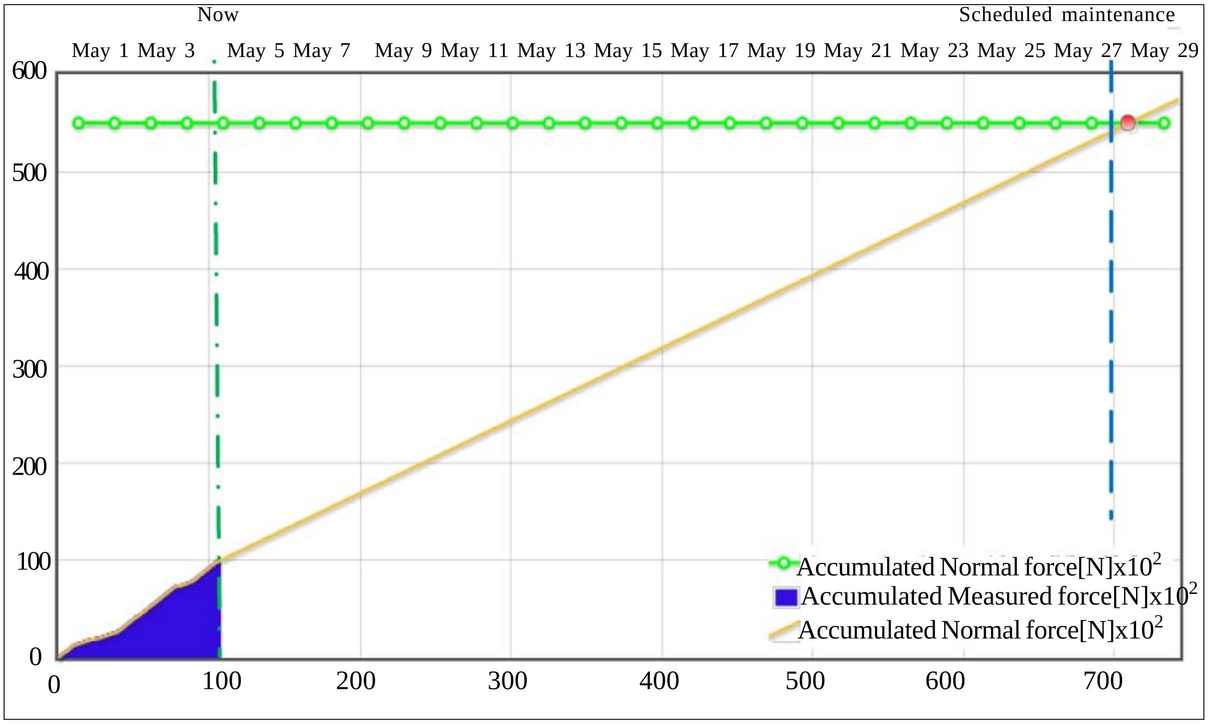


Figure 11. Scheduled maintenance is on May 29th. The accumulated predicted force will meet the accumulated acceptable one after the scheduled maintenance date

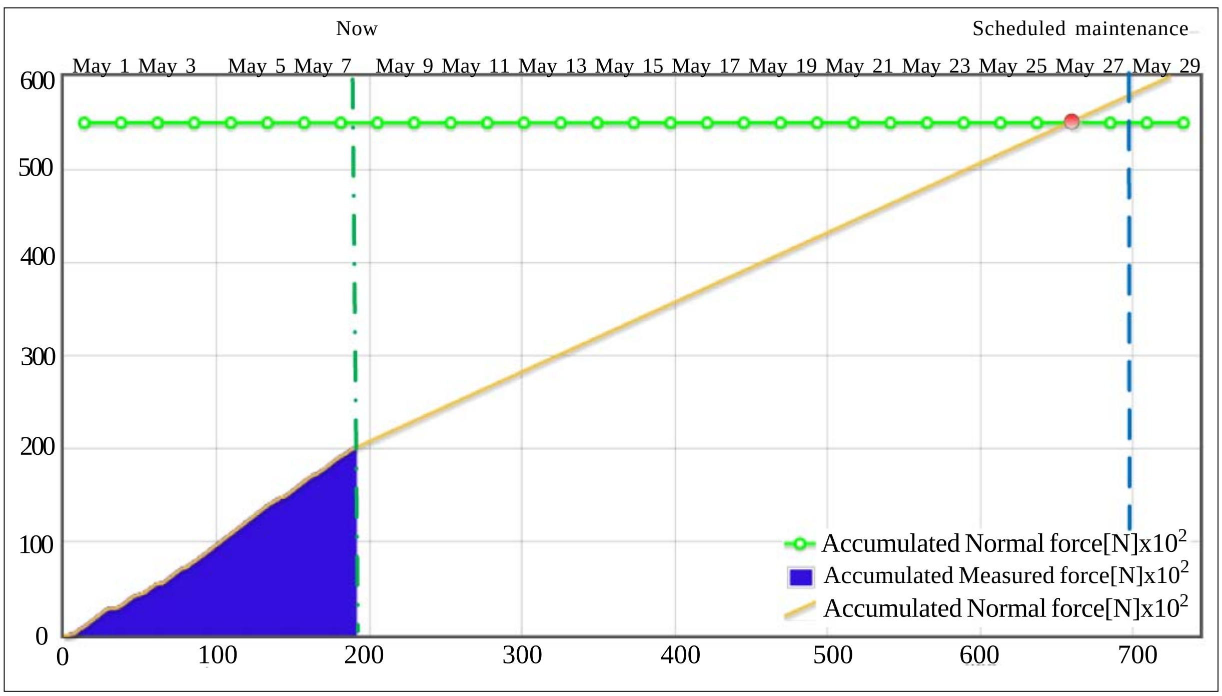


Figure 12. Scheduled maintenance is on May 29th. The accumulated predicted force will exceed the accumulated acceptable one on May 27th

before the scheduled maintenance as shown in Figure 12. Obviously, the maintenance plan must be rescheduled. In this case, the system will issue an alarm to the operators at the operations center in order to reschedule the maintenance plan.

7. Related work & Discussions

There has been lots of work carried out to lower the maintenance costs of wind turbine blades by using artificial intelligence to develop intelligent systems or proposing new maintenance strategies. For example, Kusiak & Verma [24] introduced a data-mining approach to analyzing and predicting faults associated with the blade pitch angle of a wind turbine. Garcia et al. [13] introduced an intelligent system utilizing artificial intelligence and modeling techniques such as neural networks, genetic algorithms, and fuzzy logic for on-line health condition monitoring of a wind turbine gearbox. Besnard & Bertling [6] presented an approach for condition-based maintenance optimization applied to wind turbine blades. Different from these approaches, our approach uses knowledge-based force analysis to providing continuous monitoring and advanced alarming. The approach also solves the semantic ambiguity of offshore wind information by utilizing the benefits of ontology.

We used the average monthly wind speed to calculate the predicted centrifugal force in a month, but the result could be improved if we use weather forecast information in higher resolution, for example, by daily or hourly information. The more precise the forecast information is, the better the result we can achieve. One possible direction to extent the current work is to develop a recommendation system for rescheduling maintenance based on available information about weather windows and transportations.

8. Conclusions

Moving wind turbines from shallow water off the shore and to deeper water is bringing more benefits to the wind energy industry since wind turbines can have better access to high wind resources. However, the costs of O&M and installation are relatively high due to the limitation of accessibility to offshore wind turbines, especially during the winter time. In case of a replacement of a component, heavy transportations such as vessels and cranes must be involved. In order to reduce the breakdown and downtime of wind turbines as much as possible, proper O&M strategies must be implemented. Since blade fault is one of the most common fault modes of a wind turbine, blades should be monitored and maintained appropriately such that the cost of maintenance and replacement of blades can be lowered. In this work, we have introduced an approach which uses knowledge-based force analysis to supporting maintenance for offshore wind turbine blades. We have developed a system that allows to have realtime monitoring of a wind turbine blade and additional force on it. The system also provides the possibility of triggering an advanced alarm when needed. We used data from an onshore wind farm operated by Statkraft AS to test our system. The results showed that the system can predict when the additional force exceeds the acceptable one. Particularly, in our experiment the alarm was issued in advance to warn the operators at the operations center about the need of changing the maintenance plan. In the future, we plan to apply our approach to other wind turbine components.

References

[1] IEC 61400. (2006). Wind turbines - part 25: Communications for monitoring and control of wind power plants.

[2] Antoniou, G., Van Harmelen, F. (2009). Web ontology language: OWL. *In:* Handbook on ontologies, p. 91–110. Springer.

[3] Battle, R., Benson, E. (2008). Bridging the Semantic Web and Web 2.0 with Representational State Transfer (REST). Web Semantics: Science, Services and Agents on the World Wide Web, 6 (1) 61–69.

[4] Bechhoefer, E., Wadham-Gagnon, M., Boucher, B. (2012). Initial condition monitoring experience on a wind turbine. *In*: Annual conference of the prognostics and health management society.

[5] Bechhofer, S., Van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P., Stein, L. et al. (2004). OWL web ontology language reference. *W3C recommendation.*

[6] Besnard, F., Bertling, L. (2010). An approach for condition-based maintenance optimization applied to wind turbine blades. Sustainable Energy, *IEEE Transactions on*, 1 (2) 77–83.

[7] Caselitz, P., Giebhardt, J. (2005). Rotor condition monitoring for improved operational safety of offshore wind energy converters. *Journal of Solar Energy Engineering*, 127 (2) 253–261.

[8] Cibulka, J., Ebbesen, K. (2012). A review on approaches for condition based maintenance in applications with induction machines located offshore. *Modeling, Identification and Control,* 33 (2) 69–86.

[9] El-Thalji, I., Liyanage, J. P. (2010). Integrated asset management practices for offshore wind power industry: A critical review and a road map to the future. *In*: The Proceedings of The Twentieth (2010) International Offshore and Polar Engineering conference, p. 934–941.

[10] European Commission. (2005). Advanced maintenance and repair for offshore wind farms using fault prediction and condition monitoring techniques, Offshore M&R Final Public Report under the FP5 Contract NNE 5/2001/710. Tech. rep.

[11] Fielding, R., Taylor, R. (2002). Principled design of the modern Web architecture. *ACM Transactions on Internet Technology* (TOIT), 2 (2) 115–150.

[12] Fischer, K., Besnard, F., Bertling, L. (2006). Reliability-centered maintenance for wind turbines based on statistical analysis and practical experience. Energy Conversion, *IEEE Transactions on*, 27 (1) 184–195.

[13] Garcia, M. C., Sanz-Bobi, M. A., del Pico, J. (2006). SIMAP: Intelligent system for predictive maintenance: Application to the health condition monitoring of a windturbine gearbox. *Computers in Industry*, 57 (6) 552–568.

[14] Garc1a Marquez, F. P., Tobias, A. M., Pinar Perez, J. M., Papaelias, M. (2012). Condition monitoring of wind turbines: Techniques and methods. *Renewable Energy*.

[15] Grimm, S. (2010). Knowledge representation and ontologies. *In:* Scientific Data Mining and Knowledge Discovery, p. 111–137. Springer.

[16] Gruber, T. R., et al. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International journal of human computer studies*, 43 (5) 907–928.

[17] Hahn, B. (1999). Reliability assessment of wind turbines in germany. *In*: 1999 EuropeanWind Energy Conference, Nice, France, March, p. 1–5.

[18] Hameed, Z., Hong, Y., Cho, Y., Ahn, S., Song, C. (2009). Condition monitoring and fault detection of wind turbines and related algorithms: A review. *Renewable and Sustainable energy reviews,* 13 (1) 1–39.

[19] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., Dean, M., et al. (2004). SWRL: A semantic web rule language combining OWL and RuleML. *W3C Member submission*, 21, 79.

[20] Horrocks, I., Patel-Schneider, P. F., Van Harmelen, F. (2003). From SHIQ and RDF to OWL: The making of a web ontology language. Web semantics: science, services and agents on the World Wide Web, 1 (1) 7–26.

[21] Horrocks, I. R. (1997). Optimising tableaux decision procedures for description logics. Ph.D. thesis, Citeseer.

[22] Karyotakis, A. (2011). On the optimisation of operation and maintenance strategies for offshore wind farms. Ph.D. thesis, University College London.

[23] Kusiak, A., Verma, A. (2011). A data-driven approach for monitoring blade pitch faults in wind turbines. Sustainable Energy, *IEEE Transactions on,* 2 (1) 87–96.

[24] Kusiak, A., Verma, A. (2012). A data-mining approach to monitoring wind turbines. Sustainable Energy, *IEEE Transactions on,* 3 (1) 150–157.

[25] Lenzerini, M., Milano, D., Poggi, A. (2004). Ontology representation & reasoning. Universit di Roma La Sapienza, Roma, Italy, Tech. Rep. NoE InterOp (IST-508011).

[26] Li, Y. (2011). Discussion on the principles of wind turbine condition monitoring system. *In*: Materials for Renewable Energy & Environment ICMREE), 2011 International Conference on, 1 (1) 621–624. *IEEE*.

[27] Mar´echaux, J. L. (2006). Combining service-oriented architecture and eventdriven architecture using an enterprise service bus. IBM Developer Works, p. 1269–1275.

[28] McGuinness, J. (2011). How to plan proactive blade maintenance. Accessed: April-2013

[29] Nguyen, T. H., Prinz, A., Friisø, T., Nossum, R., Tyapin, I. (2013). A framework for data integration of offshore wind farms. Renewable Energy, 60, 150 – 161 .

[30] Nilsson, J., Bertling, U. (2007). Maintenance management of wind power systems using condition monitoring systems. *In*: Power Engineering Society General Meeting. IEEE, p. 1–1.

[31] Oconnor, M., Knublauch, H., Tu, S., Musen, M. (2005). Writing rules for the semantic web using swrl and jess. Prot´eg´e With Rules WS, Madrid.

[32] Ragheb, M. (2008). Control of wind turbines. URL: https://netfiles. uiuc. edu/mragheb/www/NPRE 20475.

[33] Ragheb, M., Ragheb, A. M.(2011). Wind turbines theory-the betz equation and optimal rotor tip speed ratio. R. Carriveau, Fundamental and Advanced Topics in Wind Power, p. 19–37.

[34] Schubel, P. J., Crossley, R. J. (2012). Wind turbine blade design. *Energies*, 5 (9) 3425–3449.

[35] Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., Katz, Y. (2007). Pellet . A practical owl-dl reasoner. Web Semantics: science, services and agents on the World Wide Web, 5 (2), 51–53.

[36] Tian, Z., Jin, T., Wu, B., Ding, F. (2011). Condition based maintenance optimization for wind power generation systems under continuous monitoring. *Renewable Energy*, 36 (5) 1502–1509.

[37] Wiser, R., Yang, Z., Hand, M., Hohmeyer, O., Infield, D., Jensen, P., Nikolaev, V., O'Malley, M., Sinden, G., Zervos, A. A. Z.: Wind energy. Cambridge University Press, Cambridge United Kingdom and New York, NY, USA, In IPCC Special Report on Renewable Energy Sources and Climate Change Mitigation edn. (2011). O. Edenhofer , R. Pichs-Madruga, Y. Sokona, K. Seyboth, P . Matschoss, S. Kadner , T. Zwickel, P . Eickemeier , G. Hansen, S. Schlmer , C. von Stechow (eds).