Offshore Wind Data Integration

Trinh Hoang Nguyen

Offshore Wind Data Integration

Doctoral Dissertation

Submitted for the Partial Fulfillment of the Requirements for the Degree of

Philosophiae Doctor (PhD)

in Information and Communication Technology

University of Agder
Faculty of Engineering and Science
2014

Doctoral Dissertation by the University of Agder 94
ISBN: 978-82-7117-779-9
ISSN: 1504-9272

Printed by the Printing Office, University of Agder
Kristiansand

To my parents & grandparents

Abstract

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 due to large areas of wind resources. However, offshore wind is encountering big challenges such as effective use of wind power plants, reduced cost of installation as well as operation and maintenance (O&M).

Improved O&M is likely to reduce the hazard exposure of the employees, increase income, and support offshore activities more efficiently. In order to optimize the O&M, the importance of data exchange and knowledge sharing within the offshore wind industry must be realized. With more data available and accessible, it is possible to make better decisions, and thereby improve the recovery rates and reduce the operational costs.

This dissertation proposes a holistic way of improving remote operations for offshore wind farms by using data integration. Particularly, semantics and integration aspects of data integration are investigated. The research looks at both theoretical foundations and practical implementations.

As the outcome of the research, a framework for data integration of offshore wind farms has been developed. The framework consists of three main components: the semantic model, the data source handling, and the information provisioning. In particular, an offshore wind ontology has been proposed to explore the semantics of wind data and enable knowledge sharing and data exchange. The ontology is aligned with semantic sensor network ontology to support management of metadata in smart grids. That is to say, the ontology-based approach has been proven to be useful in managing data and metadata in the offshore wind and in smart grids. A quality-based approach is proposed to manage, select, and provide the most suitable data source for users based upon their quality requirements and an approach to formally describing derived data in ontologies is investigated.

Preface

This thesis is submitted to the University of Agder (UiA) for partial fulfilment of the requirements for the degree of philosophiae doctor.

This doctoral work has been performed at the Department of Information and Communication Technology, UiA, Grimstad, with Prof. Dr. Andreas Prinz, UiA as main supervisor and with co-supervisors Dr. Trond Friisø, Origo Solutions AS and Prof. Dr. Rolf Tomas Nossum, UiA.

During my PhD study, I was a visiting scholar at Georgia Institute of Technology (GT), USA from August to December 2012. The host professor was Prof. Leo Mark. From March to May 2014, I was a visiting scholar at Sapienza University of Rome. The host professor was Prof. Maurizio Lenzerini.

This work has been (partially) funded by the Norwegian Centre for Offshore Wind Energy (NORCOWE) under grant 193821/S60 from the Research Council of Norway (RCN). NORCOWE is a consortium with partners from industry and science, hosted by Christian Michelsen Research. My visit to GT was supported by the EU 7th Framework Programme FP7-PEOPLE-IRSES under grant agreement no. 247083, project acronym S2EuNet. My visit to Sapienza University of Rome was partially supported by the Erasmus Staff Mobility program at UiA.

Acknowledgments

I would like to thank my principal supervisor, Prof. Dr. Andreas Prinz, without whom I would not have completed my work. I thank him for all his efforts he has offered me. I also thank him for being a wonderful manager who let me have a flexible working schedule in order to obtain other skills rather than doing research.

I thank my first co-supervisor, Dr. Trond Friisø for his helps, especially, the first year in my PhD program. The welcomeness of him and his family made me feel great when I started a new life in Norway. I thank my second co-supervisor, Prof. Dr. Rolf Nossum for his support and advice. He was the one who encouraged me to submit my first journal paper. I also thank him for being quick to respond to all my requests.

I thank Prof. Frank Li for giving me a research exchange opportunity, Prof. Leo Mark for hosting my visit to Georgia Tech, and Prof. Maurizio Lenzerini for hosting my visit to Sapienza University of Rome. I also thank all my colleagues at ICT department for happy moments we have shared together.

I am grateful to Statkraft AS and Agder Energi for providing me operational wind data.

I would like to thank my Vietnamese friends in Grimstad for accepting me as a member of a big family and for helping me whenever I needed.

Last but not least, I extend my deepest thanks to my family for believing me and supporting me.

Trinh Hoang Nguyen

Grimstad, Norway, July 22, 2014

List of Publications

As the outcome of the PhD study, the author of this dissertation is the main contributor and the first author of all the papers listed below. Papers A-E are selected and presented in Part II of this dissertation as the author's main research achievements. Papers 6-13 are not included in this dissertation.

Papers included in the dissertation

Paper A

Trinh Hoang Nguyen, Andreas Prinz, Trond Friisø, Rolf Nossum, and Ilya Tyapin. A framework for data integration of offshore wind farms. *Renewable Energy*, 60:150-161, 2013.

Paper B

Trinh Hoang Nguyen, Rocky Dunlap, Leo Mark, Andreas Prinz, Bjørn Mo Østgren, and Trond Friisø. Offshore wind metadata management. *Int. J. of Metadata, Semantics and Ontologies*. 9:333-349, 2014.

Paper C

Trinh Hoang Nguyen, Kamyar Rasta, Dafferianto Trinugroho, and Andreas Prinz. A semantic-enhanced quality-based approach to handling data sources in enterprise service bus. *Int. J. of Computer Science and Applications*. In Press.

Paper D

Trinh Hoang Nguyen, Vimala Nunavath, and Andreas Prinz. Big Data metadata management in Smart Grids. *Big Data and Internet of Things: A Roadmap for Smart Environments*. N. Bessis and C. Dobre (eds.), Big Data and Internet of Things: A Roadmap for Smart Environments, Studies in Computational Intelligence 546, Springer International Publishing Switzerland 2014.

Paper E

Trinh Hoang Nguyen, Andreas Prinz, and Josef Noll. An approach to supporting maintenance of offshore wind turbine blades. *Int. J. of Intelligent Computing*, 4:176-191, 2013.

Papers not included in the dissertation

Paper 6

Trinh Hoang Nguyen, Andreas Prinz, Josef Noll. Proactive maintenance of offshore wind turbine blades using knowledge-based force analysis. INTECH

2013, 29-31 Aug, London, UK.

Paper 7

Trinh Hoang Nguyen, Andreas Prinz, Trond Friisø, and Rolf Tomas Nossum. Smart Grid for offshore wind farms: Towards an information model based on the IEC 61400-25 standard. IEEE PES Innovative Smart Grid Technologies Conference 2012. IEEE conference proceedings 2012 ISBN 978-1-4577-2159-5.

Paper 8

Trinh Hoang Nguyen and Andreas Prinz. Using semantics to facilitate data integration of offshore wind farms. Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean, proceedings. IEEE conference proceedings 2012 ISBN 978-1-4673-0782-6. p. 430-433.

Paper 9

Trinh Hoang Nguyen, Kamyar Rasta, Dafferianto Trinugroho, and Andreas Prinz. Using enterprise service bus for offshore wind farm data handling. Proceedings of the IADIS International Conference Applied Computing 2012. IADIS Press 2012 ISBN 978-989-8533-14-2. p. 83-90.

Paper 10

Trinh Hoang Nguyen, Andreas Prinz, Trond Friisø, and Rolf Tomas Nossum. A semantic model for data integration of offshore wind farms. NIK: Norsk Informatikkonferanse 2011 p. 235-238.

Paper 11

Trinh Hoang Nguyen, Andreas Prinz, Trond Friisø, and Rolf Tomas Nossum. Offshore wind information provisioning using the IEC 61400-25 standard and RESTful web services. EWEA Offshore 2011.

Paper 12

Trinh Hoang Nguyen, Andreas Prinz, and Trond Friisø. Reference architecture for remote operations of offshore wind farms. Proceedings of the 24th International Congress on Condition Monitoring and Diagnostics Engineering Management (COMADEM2011). COMADEM International 2011 ISBN 0-9541307-2-3. p. 1051-1059.

Contents

List of Tables

List of Figures

Abbreviations

API	Application Programming Interface
CBM	Condition-based Maintenance
CIM	Common Information Model
CMS	Condition-based Monitoring System
CWA	Closed World Assumption
DL	Description Logic
EDA	Event-Driven Architecture
ESB	Enterprise Service Bus
EU	European Union
EWEA	European Wind Energy Association
HTTP	HyperText Transfer Protocol
ICT	Information and Communication Technology
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IO	Integrated Operations
IoT	Internet-of-Things
IT	Information Technology
KB	Knowledge Base
NIST	National Institute of Standards and Technology
O&M	Operation and Maintenance
OGC	Open Geospatial Consortium

OLF	Norwegian Oil and Gas Association
OWA	Open World Assumption
OWL	Web Ontology Language
OWO	Offshore Wind Ontology
R2RML	RDB to RDF Mapping Language
RDB	Relational DataBase
RDF	Resource Description Framework
REST	REpresentational State Transfer
ROA	Resource-Oriented Architecture
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SQWRL	Semantic Query-enhance Web Rule Language
SSN	Semantic Sensor Network
SWE	Sensor Web Enablement
SWRL	Semantic Web Rule Language
TPWind	European Wind Energy Technology Platform
UML	Unified Modeling Language
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WADL	Web Application Description Language
WPP	Wind Power Plant
WSDL	Web Service Description Language
XML	eXtensible Markup Language

Part I

Chapter 1

Introduction

The steadily increasing global energy consumption, rising of sea levels, acidifying of oceans, melting of ice caps, and the limited availability of fossil fuels create a need for sustainable energy sources. With high potential in harvesting power from the wind, offshore wind energy is being addressed as one of the most promising renewable energy sources. However, the costs of offshore wind energy production are still high due to a number of factors. One of the solutions to reduce the costs is to implement Information and Communication Technology (ICT) into optimizing operation and maintenance (O&M). This chapter presents the context, the problems we want to tackle, the research methodology, highlights of our contributions to knowledge, and the structure of this dissertation.

1.1 The Context

Among renewable energy sources, wind power has high potential in producing electrical energy. The vision of the European Wind Energy Technology Platform (TP-Wind) for wind energy production in 2030 is about 25% coverage of the electricity consumption within Europe, with a total installed capacity of 300 gigawatts (GW) [147, 98].

The concept of harvesting power from the wind was introduced a long time ago. But the first automatically operated wind turbine was introduced in 1887 by Charles F. Brush [114], and the first offshore wind power plant was built in 1991 in Denmark [142]. After decades of development, wind power plants are getting bigger and bigger in size (length of blades, tower, and rotor) as well as amount of energy production (multi-megawatt).

Wind turbines have been installed both onshore and offshore. Onshore wind farms have been built in many countries, such as Denmark, Norway, the UK, Italy, and the US. To utilize the more stable wind resources, wind farms have been moved off the shore, first to shallow waters near shore and now extending to deeper waters and far off the shore. Deep-water offshore solutions give access to large areas with high wind, less sensitivity to noise and visual impacts and size. Offshore wind technology is recognized by the Norwegian government and the European Union (EU) as an area with an enormous energy potential, in both a long-term and climate-friendly perspective [98]. According to the European Wind Energy Association (EWEA), only 1080 MW out of the 56.5 GW wind power installed in the EU at the end of 2007 was offshore [147]. The number is predicted to be 40 GW, and 150 GW for 2020 and 2030, respectively [41].

Moving wind farms from onshore to offshore is facing many problems such as costs of operation and maintenance, installation, accessibility, and corrosion. The cost of developing and operating an offshore wind farm is many times larger than the onshore one [38]. Indeed, it is difficult to access wind turbines due to harsh weather condition, especially in winter. Besides, installation of offshore wind turbines involves heavy transportation (cranes, vessels, etc.). Therefore, seeking ways to reduce the cost is very important [4, 54]. There are several possible ways to reduce the costs such as development of new installation methods [21], wind farm layout optimization [105], improvement of weather forecast and power output prediction [107], remote operation optimization [28], and wind turbine/farm control optimization [80].

Offshore wind turbines are normally unmanned, hence any operation of them is automatic or remote. In each wind farm, there should be communication between wind turbines, between wind farms, and between wind turbines and maintenance personnel with the purpose of production optimization. For example, according to the wind direction and wind speed, every turbine can have a different configuration so that the wind turbine can generate energy most cost effectively. An example of basic daily operations of offshore wind industry is illustrated in Fig. 1.1.

In the operations center, experts with different backgrounds are able to perform a proper analysis based on measurements received from offshore wind farms to arrange unscheduled works and retune the controllers if necessary. The center handles various operations such as data analysis, data visualization, and information collection from weather forecast. Based on the collected information, the center controls

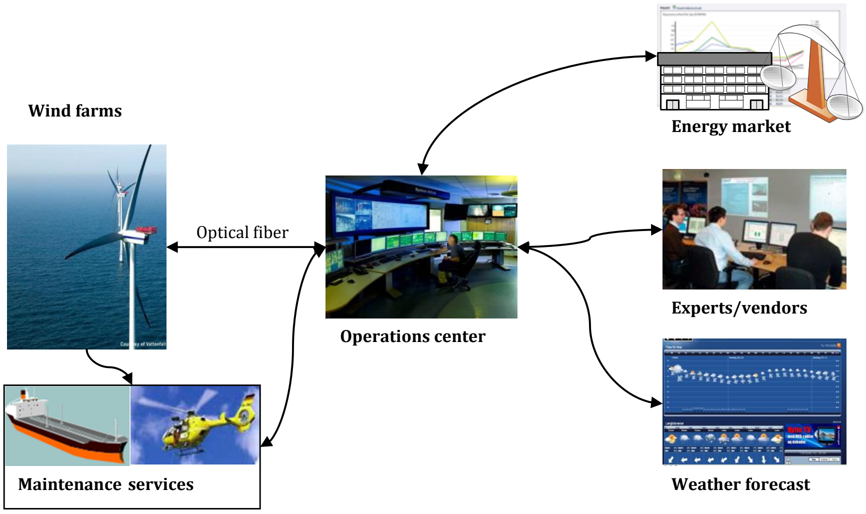


Figure 1.1: Offshore wind daily operations

the velocity of wind turbines and optimizes the wind parks in order to maximize the energy production. In addition, the use of the operations center will allow reliability engineers to better plan and organize maintenance actions related to offshore wind turbines. If there is a case where human participation is needed, the center sends a request to experts to get advice and suggestions from them. Operators or decision-makers at the center will then take the final decision and give it to on-board maintenance services in order to have correct and timely operations. The information from the operations center is also necessary for the energy market (grid/consumers), and vice-versa. Communication between wind farms and operations center probably will be transferred by using optical fibers, whereas communication between wind turbines or wind farms can be exchanged by broadband wireless communication.

1.2 Problem Outline

The main question of O&M is how to optimize the yield from an offshore wind farm over its entire operational period. Although a 25% decrease in O&M costs is expected to give less than 3% reduction in levelized costs, the relationship between improved O&M and optimized availability is important [4]. Improved O&M is also likely to reduce the hazard exposure of the employees, increase income, and support offshore activities more efficiently. One improvement of O&M is implementation of remote operations [73] which is one of integrated asset management aspects as discussed by El-Thalji and Liyanage [36].

Optimal O&M of offshore wind farms depends on a number of factors, such as optimal operation and maintenance planning, instantaneous control for optimal energy production, and remote assistance to on-site workers. These factors rely on availability, accessibility, and quality of information from various sources. For instance, the condition of the various parts of each turbine, wind conditions, price of electricity, accessibility with respect to weather conditions, price and availability of support vessels since the information are input to applications on wind speed prediction, power optimization, fault detection, etc. [79, 60, 146]. Besides, such information is valuable for research projects, political decision making, and commercial applications [36]. Unavailability of data and poor data quality can affect the results of the applications [17, 112]. That said, the more reliable information available and accessible, the better results of optimizations could be achieved. Therefore, *better ways to make reliable offshore wind data available and accessible should be developed.* With more data available, it is possible to make better decisions, and thereby improve the recovery rates and reduce the operational costs. For example, failure data can be used to analyze reliability of wind turbines by applying statistical methods.

Currently, there are a number of difficulties in making offshore wind data available and accessible. Some of them are listed as follows.

- Components of a wind power plant (WPP) are produced by different vendors. Each component has its own software and perhaps its own database. As a result, a software environment of WPP consists of multiple applications with incompatible interfaces and data formats. This leads to the inability to communicate with each other.

- Many partners and systems (e.g., wind turbine modeling, wind turbine control and monitoring) use their own applications and data formats. In addition, the process of agreement on data exchange only happens at the end of the development when the partners encounter integration problems with others. This process is time-consuming.

- Many manufacturers are reluctant to share data about their equipment, or to let third parties collect such data.

- There exist some databases provided by companies that have been collecting wind data for years. These databases are valuable sources for improving the quality of wind turbine components. The data sources are autonomous, distributed and heterogeneous systems so data reside in many incompatible formats and cannot be systematically managed, integrated and unified. Consequently, semantic inconsistency has become an even greater problem for the explicit information or knowledge sharing among users or applications.

- Many sensors are deployed on a wind turbine to measure different physical phenomena and provide data to users and applications by means of services. As sensors are prone to failures, their results might be inaccurate, incomplete, and inconsistent [125]. However, data are typically provided without any quality description attached to them. It is not clear what the accuracy or completeness of the data is. There are also cases where data quality is available but the service that makes data accessible does not provide any method to access the information.

The main research challenge addressed in this work is to propose a holistic way of improving remote operations for offshore wind farms by using data integration. Particularly, **semantics** and **integration** aspects of data integration will be investigated. The research looks at both the theoretical foundations as well as practical implementation. Although information and communication security plays an important role in designing any information system, we have not taken security into account due to time limits of the project. It is assumed that all communications are handled using secure channels.

We try to look at the main research challenge from different aspects by dividing it into research questions as follows.

- **RQ1**: How to solve the semantic inconsistency which has become an important problem of knowledge sharing and data exchange among users or applications?

- **RQ2**: Data sources are considered as autonomous, distributed and heterogeneous systems. How to manage, integrate, and unify them systematically?

- **RQ3**: How to select the most suitable data source? How to provide data for users if the requested data source is not available?

- **RQ4**: How to provide data with quality descriptions for users based on users' requirements?

- **RQ5**: Data quality dimensions are metadata. How to manage this kind of metadata?

- **RQ6**: Missing data can be caused by network disconnection, device faults, and software bugs. Is there any way to fill in the missing data? If yes, how to describe the solution formally?

1.3 Research Method

We approach our research by using an appropriate scientific method. Among three major paradigms (i.e., theory, abstraction, and design) [29], we select the design paradigm based on the characteristics of our work. The four steps: (1) state requirements, (2) state specifications, (3) design and implement the system, and (4) test the system constitute the paradigm. We adapt the paradigm to our situation.

1.3.1 Design Method

Step 1: State Requirements Requirements are defined in Chap. 1 and Chap. 3.

Step 2: State Specifications In this step, theoretical studies are conducted in order to get a comprehensive understanding about offshore wind power, reference architecture, semantic web, web services, and data integration. These studies are presented in Chap. 2.

Step 3: Design and Implementation Having taken requirements and specifications into consideration, design and implementation are carried out and described in Chap. 3.

Step 4: Evaluation The proposed solutions are evaluated against the requirements. The evaluation is presented in Chap. 4. The research is evaluated against research questions posed in Sect. 1.2 and the framework is evaluated against the requirements posed in Sect. 3.1.2.

1.3.2 Research Approach

Three main studies, i.e., data source handling, concept modeling, and information provisioning are defined based on the research questions. Fig. 1.2 depicts the connection between the studies and research questions. Based on the requirements and research questions, possible system architectures and approaches might be developed.

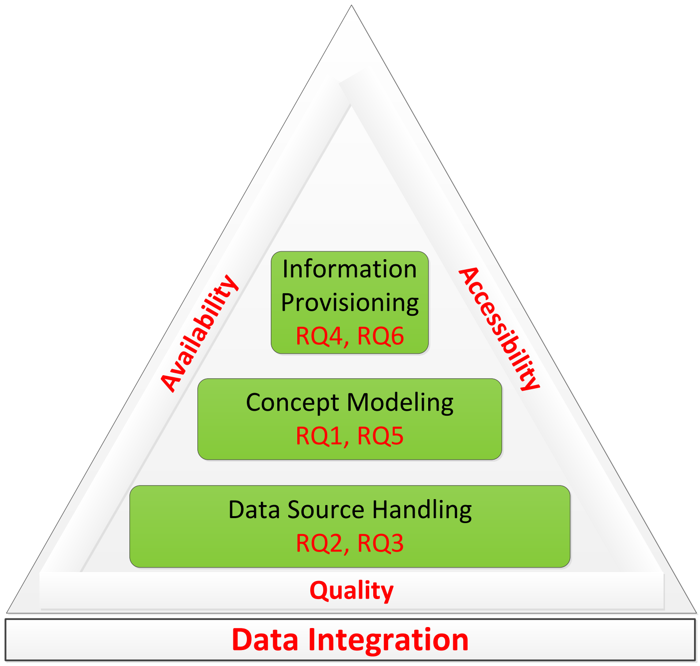


Figure 1.2: Research design

We think that the answers to *RQ1&2* can be found by taking into consideration lessons learned from other industries such as oil & gas where data integration has been studied intensively. The well-known ISO 15926 standard [2], which describes the relations between the oil & gas concepts, would be a great source for us to learn how to describe and connect offshore wind concepts. In addition, wind energy and power system standards would be great sources for getting domain concepts as well as for understanding wind turbines.

The problem described in *RQ3* is faced by many other industries, for instance, oil & gas [132], maritime [94], eHealth [71]. It is therefore worth considering solutions from others, e.g., the operational model proposed by the Norwegian Oil Industry Association (OLF), known as Integrated Operations (IO) [101].

RQ4 poses an important problem in data integration. It is data and information quality. The research on data and information quality began to get attention from research community in the late 1980s [149] and it is still an ongoing research.

RQ5&6 describes metadata issues to which solutions have not been reported yet. Fortunately, research on data quality and metadata have been carried out for the last few decades. Our solutions should be based on the available research and advanced technologies.

1.4 Contributions

Fig. 1.3 illustrates the high level design of our solution and the allocation of our contributions. The main contributions of the dissertation are summarized in the following.

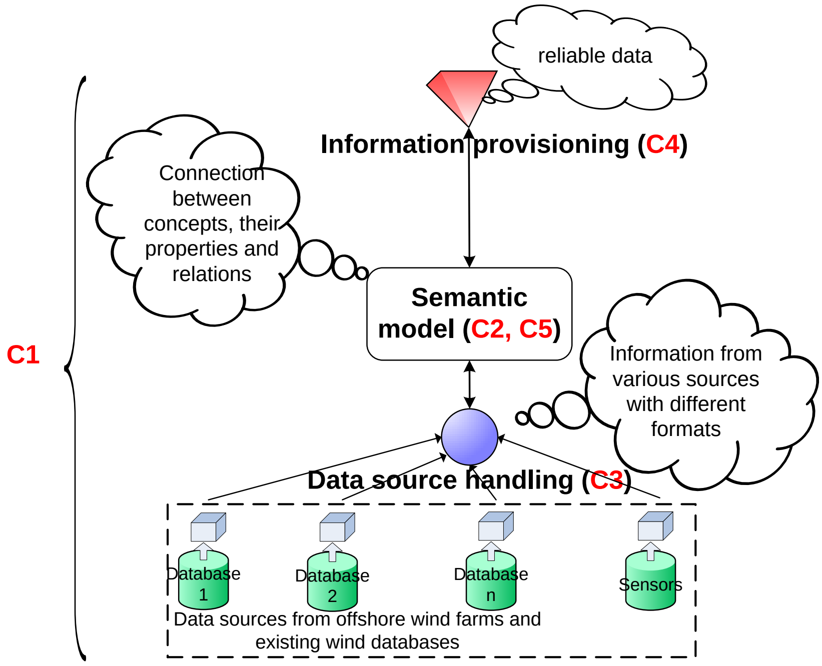


Figure 1.3: An overview of our contributions

- **C1**: The dissertation presents an information technology (IT) framework for data integration of offshore wind farms. The framework aims to optimize remote operations in order to reduce the costs of O&M. The framework allows integrating different data sources within the offshore wind industry into a unified system and providing data to interested partners.

- **C2**: The dissertation presents an offshore wind ontology based on the existing standard IEC 61400-25 to manage offshore wind metadata in a semantic way. The use of the ontology is also investigated.

- **C3**: A semantic-enhanced quality-based approach to handling data sources is proposed. This introduces a new level of abstraction that can improve the process of data quality handling with the help of semantic technologies.

- **C4**: The possibility of deriving new data sources by combining data sources of the same concepts is presented in this dissertation. A formal way to create new data sources by formulating concept dependencies is also investigated.
- **C5**: The dissertation discusses a way to manage metadata when offshore wind energy is integrated into future smart grids. We argue that semantic technologies are a solution to managing metadata effectively.

1.5 Dissertation Structure

The dissertation consists of two parts. Part I introduces the context of our research and problems we try to solve. Part I also gives an overview of our achievements which are presented in detail in Part II. Part I contains five chapters: introduction, state of the art, offshore wind data integration, evaluation & discussion, and conclusions.

Chapter 1 introduces the context, the problem statement, and research method of tackling the problem.

Chapter 2 introduces basic concepts and terminologies that are used through out the dissertation. The chapter then presents the two state-of-the-art aspects of data integration: semantics and integration.

Chapter 3 presents our main achievements. A framework for data integration of offshore wind farms is presented. The framework consists of three components, i.e., the semantic model, the data source handling, and the information provisioning. The details of the three components are discussed.

Chapter 4 presents some related work and discusses our contributions to knowledge. Evaluation of our contributions against the research questions is given.

Chapter 5 briefly summarizes the contributions we have presented in this dissertation. Concluding remarks and future outlook are then presented.

Part II consists of five appendixes A-E which present our selected publications.

Chapter 2

State of the Art

Data integration is not only about integrating different data sources resided in different locations in order to enhance knowledge base, but it is also about exploiting semantics of data stored in the knowledge base. This chapter first presents definitions of concepts that are used in this dissertation. It then describes state-of-the-art data integration with a particular focus on semantics and integration.

2.1 Concepts

A concept can be interpreted and understood differently according to contexts. We therefore briefly describe all the important concepts that are used in this dissertation in order to avoid any ambiguity.

i) Data - Information - Knowledge - Wisdom

Data are a collection of raw and unorganized symbols that represent real world states. *Information* is the processed, organized, and structured data according to a given context [6, 144]. Data are transformed to information only if the data are used for particular purposes, e.g., modeling, documentation. Part of the information will become *knowledge* (i.e., what can be derived from the information) in terms of abstraction and perception. Eventually, only part of the knowledge will be transformed to *wisdom* if the knowledge is used to serve some actionable intelligence [121].

ii) Data integration

Data integration is a process of combining data residing in different sources and providing users with a unified view of these data [67, 85].

iii) Semantics

Semantics is the study of meaning [62]. In this dissertation, semantics of data is being exploited so that data can be understood better by systems and more information can be extracted from the data.

iv) Ontology

In philosophy, ontology is a branch of metaphysics relating to the nature and relations of being [81]; or ontology is the study of the kinds of things that exist [25]. In computer science, the term ontology refers to conceptualizations of terms in a domain vocabulary [25]. In this dissertation, we use the commonly used definition proposed by Gruber [56].

"An ontology is an explicit specification of a conceptualization". In ontologies, concepts, properties, relations, functions, constraints, and axioms of a particular domain are explicitly defined.

2.2 Semantics

This section discusses the semantic aspect of data integration. In particular, ways to represent knowledge are described. Besides, a standard that describes wind data exchange is presented.

2.2.1 Knowledge Representation

We are facing an unforeseen growth of the complexity of knowledge. Every moment there is a huge amount of data produced on the Internet, e.g., from social networks or data collected from oil platforms, operational wind farms, or from healthcare services. All these data are presented in different forms.

It is not difficult for a human to 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. Knowledge representation is the study of the formalizations of knowledge and its processing within machines [55].

2.2.1.1 Forms of Knowledge Representation

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 [55]. The latter one provides a precise semantic interpretation utilizing both forms of the former two.

A semantic network is a graph which involves nodes and links between nodes [81]. Each node represents a domain concept while a link denotes a relation between two domain concepts. Fig. 2.1 represents a semantic network for a wind turbine rotor (WROT).

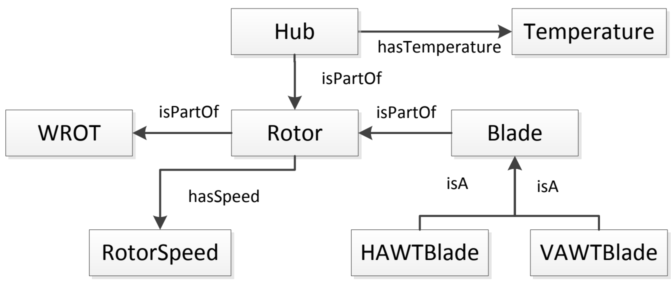


Figure 2.1: A semantic network representation of WROT

WROT, Rotor, RotorSpeed, etc. represent the concepts of the wind domain while *isA*, *isPartOf*, *hasSpeed* 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 a semantic network to represent knowledge. For example, it is not clear how many blades are part of a rotor according to the semantic network presented in Fig. 2.1.

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

(a) IF $Blade\_BL101$ is a $HAWTBlade$ THEN $Blade\_BL101$ is a $Blade$
(b) IF $Blade\_BL101$ is a $HAWTBlade$
AND $wrot$ is a $WROT$
AND $rotor$ is a $WROTRotor$
AND $Blade\_BL101$ is a part of $rotor$
AND $rotor$ is a part of $wrot$
THEN $Blade\_BL101$ is a part of $wrot$

Rule-based knowledge representation systems are especially suitable for reasoning about concrete instance data [55], for example, in the second rule example, *Blade_BL101*, *rotor*, *wrot* are concrete instance data. However, if the rules get more complicated, the logical interactions within the rules become opaque and the performance of rule-based systems decreases.

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. DL is a family of formal knowledge presentation languages [34]. DLs are used to provide a logical formalism for ontologies [65]. DLs are subsets of first-order logic. In DL, domain notions are described by concept descriptions [9]; an axiom is a logical statement relating roles and/or concepts [53].

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. Given that $HAWTBlade$, $Blade$, $Rotor$ are concepts and $hasBlade$ is a role in an ontology and they belong to TBox, and $Blade\_BL101$ belongs to ABox, we can express their relationships as follows.

(1) "$HAWTBlade$ is a kind of $Blade$" is represented as $HAWTBlade \sqsubseteq Blade$
(2) "$Blade\_BL101$ is an instance of $Blade$" is expressed as $Blade\_BL101 \in Blade$

(3) "$Rotor$ has exactly three blades" is expressed as $Rotor \sqcap = 3.hasBlade$.

2.2.1.2 Knowledge Representation with Ontologies

In general, an ontology is needed to make an abstract model of some phenomenon by identifying the relevant concepts of that phenomenon [129]. It facilitates integration of processes within and across business domains, creation of autonomous solutions, and storage of data over time. It is also a key instrument in developing the semantic web which is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation [16]. Ontologies have been extensively used in data integration systems [31].

Well-known ontology languages are SHOE, OIL, DAML-ONT, DAML+OIL, and

OWL [86]. Among them Web Ontology Language (OWL), a language proposed by the World Wide Web Consortium (W3C) Web Ontology Working Group, maintains the compatibility with the existing ones [64].

OWL is a description logic based ontology language [63]. The design of OWL was influenced by its predecessors DAML+OIL, the frames paradigm and Resource Description Framework (RDF) and RDF Schema (RDFS) [64]. OWL takes RDFS basic features and introduces some more features. Basically, OWL is an extension of RDFS, in the sense that OWL uses the RDF meaning of classes and properties [64, 15, 7]. For example, in RDF schema, cardinality restrictions, disjointness of classes, and logical combinations (intersections, unions or complements) are missing. Let us examine some concrete cases within the wind energy domain. The first case is that in RDF, we cannot state that *HydraulicSystem* and *HeatingSystem* are disjoint classes. The second case concerns the lack of cardinality restrictions, e.g., the fact that a WPP can have more than one wind turbine converter component (WCNV) cannot be expressed in RDF, but it can be done in OWL using the following axiom $WPP \sqsubseteq (\geq 1\ hasWPPComponent.WCNV)$.

In OWL, Owl:Thing is a built-in most general class and is the class of all individuals. It is a superclass of all OWL classes. Classes are defined using owl:Class. A class defines a group of individuals that belong together. Individuals can be referred to as being instances of classes. Note that the word concept is sometimes used in place of class. 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. There are two types of properties in OWL ontology. They are object property and data type property. Properties in OWL are also known as roles in DLs and as relations in Unified Modeling Language (UML). An object property relates individuals to other individuals (e.g., *hasWPPComponent* relates *WPP* to *WPP components*). An object property is defined as an instance of the built-in OWL class owl:ObjectProperty. A data type property relates individuals to data type values (e.g., *hasOilPressure*, *hasWindSpeed*). A datatype property is defined as an instance of the built-in OWL class owl:DatatypeProperty. A property in OWL can be transitive, functional, symmetric, or inverse.

There are two versions of OWL: OWL 1 [104] and OWL 2 [139]. OWL 1 has three profiles: OWL Lite, OWL DL, and OWL Full. OWL Lite was designed to support simple class hierarchy and simple constraints. OWL DL was developed to support existing DL and to provide possibility of working with reasoning systems. The OWL DL semantics is very similar to the $\mathcal{SHOIN}^{(\mathcal{D})}$ Description Logic. It provides maximum expressiveness and it is decidable [64]. OWL DL abstract syntax and semantics can be found in [104]. OWL Full is the most expressive language in the OWL family and it is undecidable [64].

OWL 2 has a very similar overall structure to OWL 1. OWL 2 adds some new functionality with respect to OWL 1 such as qualified cardinality restrictions, asymmetric, reflexive, and disjoint properties. OWL 2 takes RDF/XML as a primary syntax to store OWL 2 ontologies, whereas the meaning of OWL 2 ontologies is provided by either the Direct Semantics or the RDF-based semantics. OWL 2 also has three profiles OWL 2 EL, OWL 2 QL, and OWL 2 RL. These profiles are defined by placing restrions on the Functional-Style Syntax of OWL 2 [53].

Ontologies can be developed using ontology editors such as Protégé[1] [50], NeOn Toolkit[2] [57], OWLGrEd[3] [12]. In addition, ontology methodology can also be used for developing and maintaining ontologies [74]. Well-known methods are METHONTOLOGY [40], On-to-knowledge [131], and Neon methodologies [130]. To ensure the quality of content and methodology while developing ontologies, the evaluation of ontologies is needed. Examples of ontology evaluation tools are OOPS! (OntOlogy Pitfall Scanner!)[4] [108] and oQual [47].

2.2.1.3 Ontology Reasoning

A reasoner is a piece of software that is able to infer logical consequences from a set of asserted facts or axioms. It is used to ensure the quality of ontologies. It can be used to test whether concepts are non-contradictory and to derive implied relations. Reasoning with inconsistent ontologies may lead to erroneous conclusions [10]. There are some existing DL reasoners such as FaCT, FaCT++, RACER, DLP and Pellet. A reasoner has the following features: consistency checking, concept satisfiability, classification, and realization checking [124]. Given an assertional box $\mathcal{A}$ (ABox contains assertions about individuals), we can reason w.r.t a terminological box $\mathcal{T}$ (TBox contains axioms about classes) about the following:

- consistency checking: ensures that an ontology does not contain any contradictory facts. An ABox $\mathcal{A}$ is consistent with respect to $\mathcal{T}$ if there is an interpretation $\mathcal{I}$ which is a model of both $\mathcal{A}$ and $\mathcal{T}$;

[1] http://protege.stanford.edu/
[2] http://neon-toolkit.org/wiki/
[3] http://owlgred.lumii.lv/
[4] http://oeg-lia3.dia.fi.upm.es/oops/index-content.jsp

- concept satisfiability: checks if it is possible for a class to have any instances. Given a concept C and an instance a, check whether a belongs to C. $\mathcal{A} \models C(a)$ if every interpretation that satisfies $\mathcal{A}$ also satisfies $C(a)$;
- classification: computes the subclass relations between all named classes to create the complete class hierarchy. Given a concept C, retrieve all the instances a which satisfy C;
- realization: computes the direct types for each of the individuals. Given a set of concepts and an individual a, find the most specific concept(s) C (w.r.t. subsumption ordering) such that $\mathcal{A} \models C(a)$.

2.2.1.4 Ontology Querying

For a relational database (RDB), Structured Query Language (SQL) is the query language of choice, whereas for ontologies, SPARQL and SQWRL (Semantic Query-Enhanced Web Rule Language) [100] are used to build queries. SPARQL is an RDF query language and SQWRL is a SWRL-based (Semantic Web Rule Language) language for querying OWL ontologies. SPARQL extensions such as SPARQL-DL [123], and SPARQL-OWL [77] can be used as an OWL query language in many applications. But SPARQL cannot directly query entailments made using OWL constructs since it has no native understanding of OWL [100]. For example, if we want to retrieve all wind farms and number of WPP in them, we can have an SQWRL query as shown below.

$WF(?p) \wedge hasWFName(?p, ?name) \wedge hasTotalNumberWPP(?p, ?number)$
$\rightarrow sqwrl : select(?p, ?name, ?number)$

2.2.2 The IEC 61400-25 Standard

A typical problem for data exchange is mismatch of terms and formats between sender and receiver. Approved standards are necessary in order to make the data exchange process clear. The International Electrotechnical Commission (IEC) has developed the IEC 61400-25 standard in order to provide a uniform communications basis for monitoring and control of wind power plants [3]. The IEC 61400-25 standard is an adaptation of the IEC 61850 standard series for substation automation. The IEC 61400-25 standard reuses the terms and definitions defined in the IEC 61850 and extends the IEC 61850 with unique information models that are only applied to WPP.

The standard contains six parts that cover informational exchange model, mapping to communication profiles, and conformance testing. The introduction to the standard and basic definitions are given in Part 1. Part 2 and 3 define the data structure of WPP and the model for information exchange, while part 4 specifies the format of messages to exchange for SOAP-based (Simple Object Access Protocol) web services. Part 5 describes conformance testing and part 6 defines additional information models for use in condition monitoring system.

According to the IEC 61400-25 standard, the highest level is called the logical device which stands for a WPP, which is decomposed into logical nodes (LN). Fig. 2.2 shows the WPP information broken down into LNs. The structure of all LNs is specified in [3] and [1].

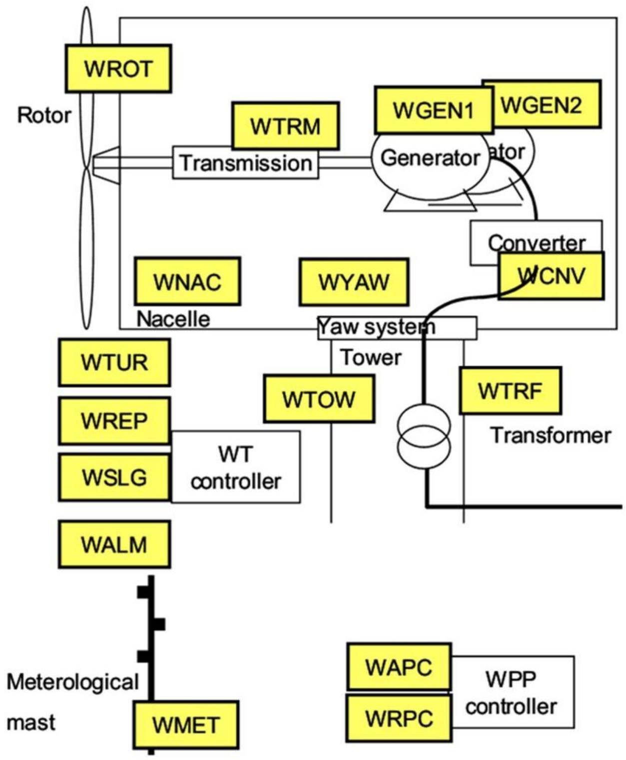


Figure 2.2: Logical nodes defined in the IEC 61400-25 [3]

WROT stands for wind turbine rotor information and WTUR stands for wind turbine general information. More details can be found in [3]. All the LNs inherit their structure from the abstract LN class defined in the IEC 61850-7-2. A LN consists of a collection of related data, called data classes (DC). Each data class inherits a collection of properties, as defined by a so-called common data class (CDC) to which it is assigned. A CDC consists of a collection of data records.

The IEC 61400-25 standard represents a consensus on core information technology for the future transition of the electric distribution grid towards a smart grid [115]. The IEC 61400-25 defines information models and information exchange models for monitoring and control of WPPs. The modeling approach of IEC 61400-25-2 and IEC 61400-25-3 uses abstract definitions of classes and services such that the specifications are independent of specific communication protocol stacks, implementations, and operating systems [118]. Successful implementation of the IEC 61400-25 standard has been reported in [76, 102].

2.3 Integration

This section first discusses approaches to data integration. It then presents state-of-the-art technologies that are available for data integration that we use in our work, i.e., Enterprise Service Bus (ESB) and Representational State Transfer (REST). ESB is a Service-Oriented Architecture (SOA) based middleware and REST is a Resource-Oriented Architecture (ROA) based style.

2.3.1 Data Integration

The purpose of data integration is to provide users with a unified overview over data sources that are heterogeneous, and distributed (i.e., residing in different physical/virtual locations). As a result, the data availability and accessibility can be improved. Data integration is not a new issue. Indeed, it appeared shortly after database systems were first introduced [150]. Even though there are many proposed solutions to the issue, it is still a challenging task. With new technologies, more and more solutions to data integration are introduced; existing solutions are improved, and new solutions are proposed.

Data integration approaches can be classified based upon architecture style, e.g., data warehouse [143], and data mediation [70]. The data warehouse architecture differs from the data mediation approach in terms of data location. Data warehouses contain a large amount of integrated data from different data sources. Data are loaded into a data warehouse by using tools such as ETL (extraction, transformation, and loading) tools. Data warehouses are designed to support decision-making purposes [37]. Examples of data integration tools that use data warehousing approach are Talend Data Integration[5], Pentaho Data Integration[6], and Microsoft SQL Server Integration Services[7]. As opposed to data warehouses, data mediation, also known as virtual database approach, uses a mediated schema over data sources to answer users' queries. The schema interfaces with data sources via wrappers [143] or mappings [85]. An advantage of this approach is that data need not to be moved to a common data storage, thus costs of building and maintaining a common database are saved and data can be delivered in real time. Examples of the data mediation approach are Teiid Data Integration[8], IBM InfoSphere Federation Server[9], and MASTRO[10].

The data mediation approach relies on a mapping between the mediated schema and data sources. The mappings can be classified into three ways according to query answering settings. They are global-as-view (GAV) [27], local-as-view (LAV) [24], and global/local-as-view (GLAV) [44]. In the GAV approach, every element of the mediated schema (global schema) is associated with a view over a data source and mappings are written with respect to the global schema, whereas every data source is defined as a view over the mediated schema and mappings are written with respect to data sources in the LAV approach [85]. Another difference between the GAV and LAV approaches is that query processing in GAV can be based on unfolding techniques while query processing in LAV requires the involvement of reasoning. The GLAV approach is a combined approach which both GAV and LAV assertions are allowed in the mapping [24].

In order to solve the semantic heterogeneity issue in data integration, ontologies are employed. Ontology-based data integration becomes one of the approaches to integrating data from multiple sources. In this approach, ontologies are considered as mediators between users' queries and data sources. The ontology-based approach offers three variations, i.e., single ontology, multiple ontology, and hybrid ontology approaches [128]. In the single ontology approach, all sources are viewed through a global shared ontology. An advantage of this approach is that any changes of a data source will not affect other data sources. However, the global ontology must be

[5]http://www.talend.com/products/data-integration
[6]http://www.pentaho.com/product/data-integration
[7]http://technet.microsoft.com/en-us/library/ms141026.aspx
[8]http://teiid.jboss.org/
[9]http://www-03.ibm.com/software/products/no/ibminfofedeserv
[10]http://www.dis.uniroma1.it/ mastro/

changed to adapt to the change of the data source. In order to avoid such an issue, the global ontology should be built based upon the knowledge of domain experts and concepts from existing data sources. In the multiple ontology approach, each source is exposed through a local ontology. Data exchange is then processed by mapping local ontologies. Any change of a local ontology might lead to changes of other ontologies that are connected to it. In the hybrid ontology approach, sources are also exposed through local ontologies which are connected to a global shared ontology. An advantage of this approach is to allow a new source to connect to the global shared ontology without modifying the existing mappings.

In the ontology-based data integration approach, mapping plays a significant role since it connects data sources to the global schema. The W3C has a RDB2RDF working group[11] whose target is to standardize languages for mapping relational database schemas into RDF and OWL. R2RML (RDB to RDF Mapping Language)[12] and DM (Direct Mapping)[13] are two examples of RDB to RDF mapping languages.

Besides the problems posed in Chap. 1, data integration encounters a number of other issues such as query evaluation optimization, and rules for automatically mapping data items in different sources [85]. Furthermore, committing operations such as delete and update data items in a data integration system is also a challenging problem.

2.3.2 Enterprise Service Bus

Typical communication between applications is to establish a direct connection between them. It means that one application can have many separate connection channels with other applications. This kind of connection is called point-to-point integration. It is easy to establish and handle. Fig. 2.3a shows the point-to-point integration of offshore wind applications. For example, in order to make a model with high accuracy or to predict the power output of a wind farm, the wind farm modeling & control application needs to have input from weather forecast (wind speed, wind direction) and equipment information (the configuration of a wind power plant or set of its component might be different from others).

However, there are some drawbacks with this integration style. Once the number of applications increases, the complexity of the whole system will increase. Any

[11] http://www.w3.org/2001/sw/rdb2rdf/
[12] http://www.w3.org/TR/r2rml/
[13] http://www.w3.org/TR/rdb-direct-mapping/

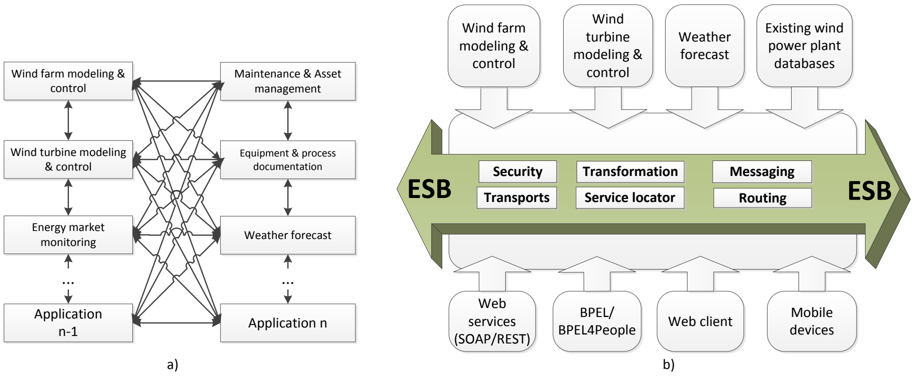


Figure 2.3: Point-to-point and ESB integration

change in an application will make the whole system change; new connections between that application and others need to be reestablished. This can result in high maintenance costs and a lack of flexibility when it becomes necessary to make changes. To overcome such issues, the ESB concept was introduced.

ESB is a software architecture for middleware. It is based on SOA which provides dynamically integrating loosely-couple services in heterogeneous distributed environments, using a standard-based software component technology [148]. SOA topologies include peer-to-peer network, hub & spoke, pipeline, and enterprise service bus [33]. An implementation of ESB is a combination of SOA and Event-Driven Architecture (EDA) [88]. EDA means that any notable thing that happens inside or outside a business disseminates immediately to all interested parties. An example of this could be using a publish/subscribe mechanism in an EDA to enable real-time monitoring of offshore wind farms; whenever new data arrives, it is immediately published to a channel and notifications about it will be sent to subscribers instead of letting subscribers query for information every once in a while. EDA is extremely loosely coupled and highly distributed [89]. EDA is used to complement SOA.

ESB provides a mechanism for managing access to applications and services via a simple and consistent interface. Fig. 2.3b shows ESB integration within the offshore wind industry. ESB is considered a communication backbone between applications. It provides transports, events, and mediation services to facilitate the integration of large-scale heterogeneous applications [88]. The idea of the ESB concept is that every application needs to be connected to a bus and then all applications can share information by producing or consuming information on the bus [26]. ESB provides the following features:

- loosely coupled architecture - whatever changes in an application will not affect other applications in the whole system;
- increased flexibility - easier to change as requirements change;
- standardized platform for integration;
- sharing common services such as security, error management, and reporting;
- more configuration rather than integration coding;
- a mediation to expose REST-based or SOAP-based services;
- service orchestration and automation.

There exist both open source and commercial ESBs. Examples of open source ESB frameworks are Mule ESB, Fuse ESB, PEtALS ESB, ServiceMix, Open ESB [48].

2.3.3 RESTful Web Services

REST is an architecture style which was introduced by Roy Fielding in 2000 [43]. RESTful provides services over the Internet through Web browsers by using the four CRUD (Create, Read, Update, Delete) operations associated with the four methods of the Hypertext Transfer Protocol (HTTP): GET, POST, PUT, DELETE. The GET method is used for retrieving information from a resource or for listing all authorized resources. This method does not change the state of the resource. The POST method is used for creating a resource on a server, while the PUT method is used for updating or modifying the state of a resource on the server. The DELETE method is used for removing a resource from the server.

The most important REST principle is to expose the resources in a RESTful service as unique URIs [42]. A REST architecture must meet the following main constraints: client-server, uniform interface, stateless, cacheable, and layered system. The client-server constraint means that REST uses the client-server architectural style. The uniform interface constraint means that each resource has an identification and resources are exposed to client through different types of representation. Besides, metadata need to be attached to each representation in order to make the message exchange self-descriptive. 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. The cacheble constraint can help to improve network efficiency by allowing clients to cache responses. The layered system constraint allows a system to be composed of hierarchical layers and therefore it is possible to prevent a layer to interact with the other layers that it is not allowed to.

The 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 since both of them use URIs as resource identifiers. Furthermore, common operations on the Semantic Web such as data fetch, insertion, and deletion are the fundamental operations in a REST-based system [13]. Resources in a RESTful web service are both identified by and resolved with a URI that generally has the form as follows.

ResourceURL ::= Protocol://Host/ApplicationPath/ResourceType/ResourceID

For SOAP-based web services, the Web Service Description Language (WSDL) is employed to describe the services, but there is no standard specification for describing REST-based web services yet. Even though the Web Application Description Language (WADL) is claimed to be a description language for REST-based web services, it has not been widely accepted yet.

An example of an open source REST framework is Restlet[14]. It is a high-level Application Programming Interface (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 [45].

2.4 Chapter Summary

In this chapter, we have discussed the two aspects of data integration: *semantics* and *integration*. In the semantics part, we have discussed knowledge representation techniques which are important for exploiting the semantics of data. We have presented the IEC 61400-25 standard which shows the current description of wind energy concepts. The standard describes only the basic structure of message exchange within the wind energy, it however does not focus specifically on either onshore or offshore wind. In the integration part, we have briefly discussed existing data integration approaches. In this work, we have selected the ontology-based data integration approach to solve the data integration issue in the offshore wind industry. The state-of-the-art SOA and ROA have been presented. The combination of SOA and ROA could solve the data source handling issue in data integration.

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

Chapter 3

Offshore Wind Data Integration

This chapter presents a data integration framework for offshore wind farms. The chapter starts with an introduction to the offshore wind stakeholders and how offshore wind data are used within the offshore wind industry. The requirements for the framework are then defined. The proposed framework, its components, and framework implementation are presented afterwards. The chapter concludes with a summary of our main achievements.

3.1 Offshore Wind Data

This section first presents an introduction to offshore wind stakeholders who actively participate in offshore wind daily operations. Then the use of data within the offshore wind industry is discussed. Based on the interests of stakeholders and the use of data, requirements for the proposed data integration framework are defined.

3.1.1 Offshore Wind Stakeholders

Stakeholders are actors that can influence or be affected by a certain problem or action. Their concerns reflect the essentials of the problem domain and determine selection of the requirements for the framework [116]. Therefore, identification of stakeholders is a significant step in building an IT framework. Desirable problem definitions can be found by careful analysis of stakeholders [23].

The offshore wind industry involves many stakeholders from different domains such as authorities, service providers, owners. By analyzing the functions and roles of stakeholders in the offshore wind industry, it is easier to define data exchange formats and what kind of data to exchange. Table 3.1 shows the different categories of stakeholders and their concerns in the offshore wind industry. Note that a stakeholder can be involved in several domains. More information about the offshore wind stakeholders can be found in Paper A.

Different activities are involved in making offshore wind turbines operate effectively. These activities include operation and maintenance, transmission, grid integration, and market energy. For instance, carrying out maintenance and inspection on-time can reduce the equipment failure rates and downtime rates, and hence increase energy production. The energy market regulates the energy price based upon the energy generated in a certain period. The energy market allows investors to bid on energy price in advance. All these activities need accessible and reliable information from various data sources in the offshore wind energy. In order to show the use of data in different activities we have conducted a review as shown in Table 3.2. More details can be found in Paper B.

3.1.2 Framework Requirements

Based on Tables 3.1 and 3.2, we have defined some general requirements in order to make sure that the proposed framework is applicable. We classify the requirements into two categories: functional and non-functional requirements. The functional requirements describe what the framework shall do and they are listed as follows.

- Table 3.1 shows the need for exchanging data within the offshore wind energy. Semantics should be clearly exploited in order to share the common understanding and avoid unambiguity in exchanging data between the stakeholders. The framework shall provide:
  - **R1**: a common understanding on domain concepts and possibility to resolve semantic ambiguity concerning wind data;
  - **R2**: a common platform to exchange data between offshore wind stakeholders and the possibility of integrating data sources, including existing data sources and new developed ones.
- In order to optimize the output of the application areas presented in Table 3.2, the quality of data exchange should be considered. The framework shall also provide:

Table 3.1: Offshore wind domains and stakeholders

Stakeholder role	Domain	Stakeholder	Function
Authorities	Wind farms	Government	Collaboration with owner/operators. Supervision and control of license obligations, environmental impact etc.
	Engineering companies	Supporter on boat/ship	Perform maintenance tasks and marine operations.
Service provider	distribution, transmission	Installation companies	Provide equipments, construction drawings and other documentation.
	maintenance service	Electricity carriers	Carry bulk electricity over long distances. Supply of electricity to end-users.
Vendors/experts	Engineering companies	Equipment vendors	Support equipment, such as turbines, blades. Provide detailed condition monitoring and analysis of specific equipment.
		Domain experts	Give technical support, advice for making urgent decision.
License partners	Wind farms	Research institutions, companies	Cooperation in larger investment and strategic decisions.
Owner/Operator	Wind farms, operations center	Decision-makers, operators	Monitor and control the wind farms' operations.
	distribution, transmission	Electricity producer	Perform overall wind farm control, production planning maintenance decisions, technical and operational analysis, coordination and support of maintenance activities.
Power market	Energy market, customer,	Operators and participants in electricity markets, electricity companies	Impact on amount of production energy. Sell electricity to customers.
	engineering companies	Residential, commercial, industrial, end-users	Consume electricity.
Weather forecast	Weather forecast	Weather forecaster, meteorologist	Storm tracks, short- and long-term predictions of wind and other weather conditions, like wave height, forecast on climate change.

Table 3.2: The use of offshore wind data

Activity	**Target**	**Input data**	**Reference**
Operations	Power optimization	Wind speed, wind direction, generator speed, yaw angle, blade pitch angle, power output.	[79]
	WT control	Wind speed, blade pitch angle, generator torque, power output, rotor speed	[78]
	Fault detection	Active power output, anemometer-measured wind speed, nacelle temp, gearbox bearing temp, gearbox lubricant oil temp, generator winding, power factor, reactive power, phase currents	[146, 60, 117]
Maintenance	Condition based maintenance	Number of wind turbines in a wind farm, number of critical components considered in a wind turbine, failure probability, age of a component since the last maintenance/inspection, cost of replacement a component, crack initiation rate, crack time to failure	[18, 134]
Energy market	Optimal day-ahead bids	Weather prediction models, local meteorological measurements,active power, wind power plant characteristic, nearby terrain and topography, recent energy price	[20]
	Energy price prediction	Measured wind power, predicted wind power, wind speed and wind direction	[19]
Transmission	Maximize utilization of wind resources	Power out, demand, voltage angle, maximum capacity of line, network topology, length of transmission line	[92]
Grid integration	Power System Transient Stability Studies	Active power, reactive power, voltage, frequency, wind speed, rotor speed, generator speed, pitch angle, wind direction	[87]

  - **R3**: the possibility of using data quality dimensions as constraints and criteria for data selection;
  - **R4**: the possibility of creating new data sources by combining data sources of the same concept.

- Data from applications or stakeholders should be made available in terms of services such that other stakeholders can easily subscribe to the services and request for the data. That is to say, the framework shall provide:
  - **R5**: the possibility of creating new data sources by formulating concept dependencies;
  - **R6**: the publish/subscribe feature and the possibility of visualizing requested information.

Different from functional requirements, non-functional requirements describe what properties the framework shall possess. The non-functional requirements are listed as follows. The framework shall be:

- **R7**: platform independent;
- **R8**: flexible enough for further extension.

3.2 The Framework

We develop a framework for remote operations of offshore wind farms. The framework fulfills the requirements posed in Sect. 3.1.2. It is based on the IO architecture from the offshore oil & gas industry. The details of the framework formation are presented in Paper A. Fig. 3.1 depicts the framework integrated in the offshore wind energy context.

Part(1) illustrates all applications from offshore wind partners, such as wind turbine control and modeling system, weather forecast monitoring system, engineering systems, asset management, wind farm modeling and control. *Part*(3) shows two upstream domains, namely production management and operation and maintenance, where new solutions are being developed. *Part*(1) and *Part*(3) are connected by *Part*(2) which presents three components of the proposed framework (*information provisioning*, *semantic model*, and *data source handling*).

The *data source handling* is used to collect data from offshore wind partners, such as weather forecast information, wind turbine modeling information, and existing

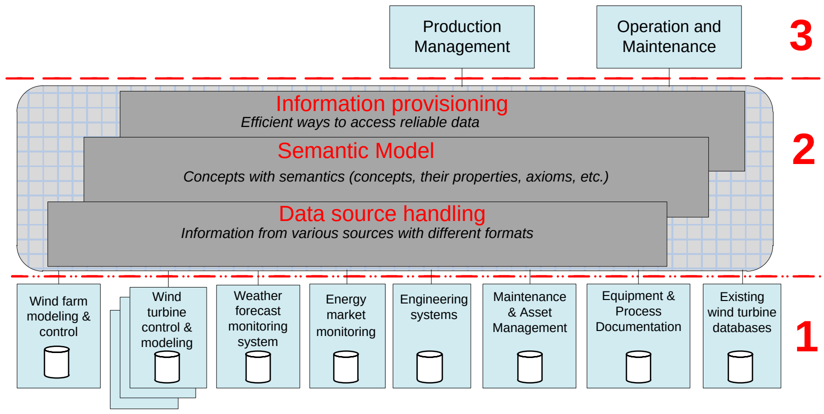


Figure 3.1: The proposed framework in the offshore wind context

wind databases. The data from this component will be processed and passed to the *semantic model*.

The *semantic model* covers the key concepts in the offshore wind domain and their semantic relationships. It is considered a core for data integration. An instance of the model, for example a virtual database, can be developed to store data.

The *information provisioning* for different purposes such as visualization, documentation, and analysis. The acquisition of relevant data underpins the lifecycle of offshore wind farms, through the project phases of feasibility analysis, development, engineering, construction, operation and maintenance, decommissioning, and post decommissioning.

The architectures of the three framework components are illustrated in Fig. 3.2. Sects. 3.3, 3.4, and 3.5 discuss these framework components in detail.

3.3 The Semantic Model

A semantic model shares the common understanding of domain concepts. It provides technologies that document the exchange protocol between offshore wind partners on data exchange, in particular, what data exchange formats to use and what kinds of data to exchange. It is obvious that terminologies used in the business domains are the most stable elements since only some parts of the data from information technology systems are used for decades and become main assets [132]. The idea of creating an *Offshore Wind Ontology* (OWO) from the terminologies is important as it can be used to share, reuse knowledge, and reason about behaviors

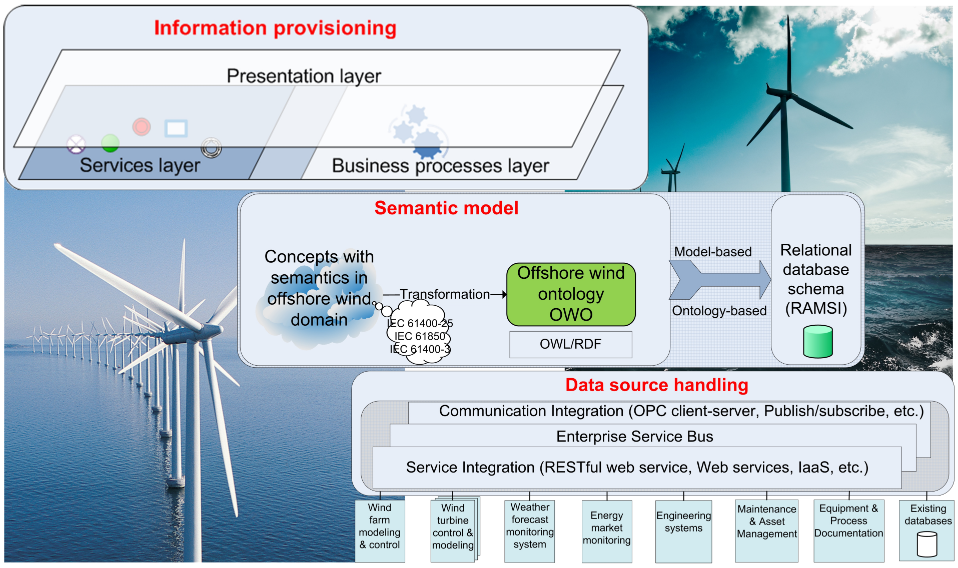


Figure 3.2: The framework components

across domains and tasks.

This section presents the core element of the *semantic model*, the offshore wind ontology. The development of OWO and its use are presented. The proposed *semantic model* fulfills the requirement *R1* presented in Sect. 3.1.2.

The *semantic model* provides the following features:

- an offshore wind ontology to describe the relations between offshore wind concepts;
- a mechanism to query the ontology;
- algorithms to generate relational database scheme from OWO;
- an information model derived from the *semantic model* is used to agree on data exchange formats, and share concepts between offshore wind partners;
- code generation for web service development.

3.3.1 OWO - an Offshore Wind Ontology

In order to develop OWO, domain knowledge and concepts of the offshore wind industry are necessary. Thus, IEC 61400-25 [3], IEC 61850 [1], and IEC 61400-3 [5] are good reference standards for the OWO development. The IEC 61400-25 standard, as discussed in Sect. 2.2.2, serves as a source of domain concepts and a backbone for OWO.

OWO is built manually by defining an ontology for each wind turbine component (ontology for WT generator, WT rotor, WT tower, etc.). The development of the ontology starts with basic terms that are described in the IEC 61400-25, and then specifying and generalizing them as required. In the development of each ontology component, we follow the METHONTOLOGY methodology [52] and Neon methodologies for building ontologies by reusing existing terminologies [138, 130, 137]. The detailed description of OWO is presented in Paper B.

3.3.1.1 OWO Development

As mentioned in Sect. 2.2.1.2, OWL Lite is not expressive enough while OWL Full is too expressive and it is undecidable [64, 124]. OWL DL is decidable and there are tools available for working with OWL DL ontologies, such as the editor Protégé, the reasoner Pellet, and the Jess rule engine. We therefore select OWL DL as the language to build our ontology model.

Each logical node defined in the IEC 61400-25 is represented by a class in OWO. The main classes of OWO are depicted in Fig. 3.3, where "1:1" means "must be exactly one", "1:1..*" means "at least one" and "1:0..*" means "may be zero or more".

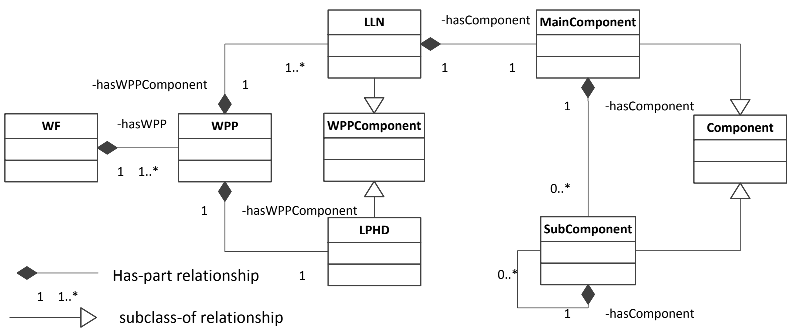


Figure 3.3: OWO main classes

WF stands for wind farm, *WPP* stands for wind power plant, *LLN* stands for logical node, and *LPHD* represents physical device information. A *WF* consists of one or more *WPP*. Each *WPP* consists of one or more *LLN* which is a subclass of the *WPPComponent* class. Each *LLN* consists of one *MainComponent* (e.g., wind turbine generator, wind turbine rotor, and wind turbine yaw). Each *MainComponent* can have many *SubComponent* (e.g., blade, control system, yaw bearing, and yaw brake). Both *MainComponent* and *SubComponent* are subclasses of the *Component* class.

The relations between classes in OWO are described by object properties such as *hasWPP*, *hasWPPComponent*, and *hasComponent*. Data properties are grouped according to categories of information exchange (e.g., equipment info, descriptive info, and discrete state info).

We used cardinality constraints *some* ($\exists$), *min* ($\geq$), *max* ($\leq$), *and exact* ($=$) to express the relations between objects. *Some* is used to create existential restriction which describes a class of individuals that have at least one relationship along a specified property to an individual that is a member of a specified class. For example, $WPP \sqsubseteq \exists\ hasWPPComponent.WGEN$ expresses the fact that a wind power plant must have at least one *WGEN* component where WGEN stands for wind turbine generator information. *Min*, *max*, and *exact* are used to express the fact that an individual is connected by an object property to at least, at most, and exactly a given number of instances of a specified class expression [93]. For instance, $WPP \sqsubseteq (=1 hasWPPComponent.WYAW)$ implies that a wind power plant has exactly one wind turbine yawing information component where WYAW stands for wind turbine yaw information.

3.3.1.2 OWO Consistency Checking

Knowledge modeling consists of creating concepts and organizing them into taxonomy. Consistency checking ensures that classes and instances in an ontology have attributes which conform to the knowledge model [145]. For instance, we can check for consistency of disjoint classes. Two classes are disjoint if and only if they do not share any instance, meaning that an instance of one class cannot be instance of the other class.

3.3.2 From OWO to RDB

RDB and ontologies are similar because both of them are used to maintain models of some universe of discourse. However, there is a difference between them. While ontologies are very useful for knowledge representation, RDB are capable of efficiently managing large amounts of structured data [75]. An advantage of ontologies is that they provide more support for inference in terms of finding answers about the model which had not been explicitly defined [8]. On the other hand, RDB have more mature technologies for storing and managing data. There is another difference between RDB and ontology based on basic assumptions [119]; RDB is based on a closed world assumption (CWA), whereas an ontology such as OWL is based on an open world assumption (OWA). For example, if the following axiom $WPP \sqsubseteq (=1 hasWPPComponent.WYAW)$ is not specified in the OWO ontology then the answer on the question whether or not *WPP* is subclass of $(= 1 hasWPPComponent.WYAW)$ would be *no* in CWA and *unknown* in OWA. The reason is that in OWA it is assumed that more knowledge can be added to the knowledge base later and the answer might be changed.

Due to the specific characteristics of wind energy, such as dealing with data from different sources, with high frequency data to support continuous condition-based monitoring, and the need for storing large amounts of data, both ontology and RDB are needed in our work. Here, RDB can be virtual or physical. If the virtual approach is selected, data federation techniques should be used, otherwise data warehousing techniques should be employed.

It is common that some data about WT components can be obtained only from vendors and the data are allowed to use for decision making but not for storing in the WT owners' systems. However, there are also data that WT owners can store and use for their future purposes. Thus it is a good idea to have a local data storage to store allowable operational data from WTs. We therefore have decided to use the data warehousing solution to handle such an issue.

We use the developed ontology to generate a global RDB schema. While the ontology contains TBox, the RDB will be used to store ABox. Fig. 3.4 illustrates the relation between ABox, TBox, and the derived RDB. Taking into consideration the following: (1) the RDB schema is generated from OWO, it is easy to user the wrapper solution; (2) creating R2RML rules manually is a time consuming process [106]; we therefore have decided to use the wrapper solution instead of the RDB2RDF approach in our system.

In the transformation, we focus only on three elements: part-of relationships, is-a relationships, and data properties. Is-a and part-of relationships in OWO are translated into 1-1 (one-to-one) and 1-n (one-to-many) relationships in RDB, respectively. We also use annotations to specify which classes must be translated into tables. We then propose an algorithm to generate RDB schema from OWO based on the ideas described in [46]. The generated RDB schema can be extended to a

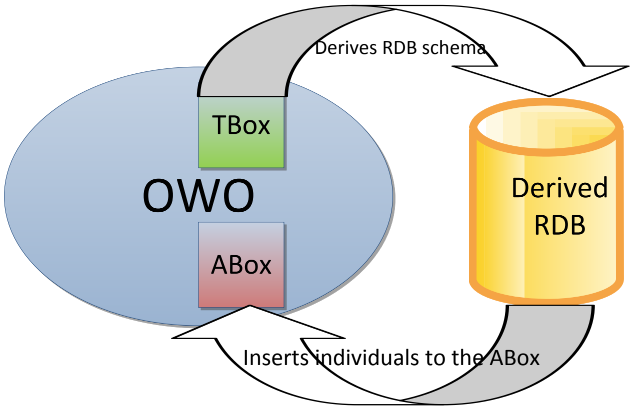


Figure 3.4: RDB schema generation

RAMSI database (reliability, availability, maintainability, safety, and inspectability [135]) which enables decreased downtime and stable production and ensures the reliability of offshore wind turbines. More details of the possibility of generating a RAMSI database can be found in Paper A and the transformation algorithm can be found in Paper B.

3.3.3 OWO Alignment

As the number of devices and appliances grows, the number of sensors embedded in such devices will also grow. Ontologies are an adequate way to model sensors and their capabilities [97]. Sensor metadata are used for selecting sensor sources and for integrating with other data sources [83]. Thus sensor metadata are important and need to be exploited.

The W3C semantic sensor network incubator group has introduced a semantic sensor network (SSN) ontology[1] to describe sensors, observations, and measurements. The ontology describes sensors and their properties such as accuracy, precision, resolution, measurement range, and capabilities. The ontology includes models for describing changes or states in an environment that a sensor can detect, and the resulting observation [30]. The semantic sensor network (SSN) was introduced based on Sensor Web Enablement (SWE) standards proposed by the Open Geospatial Consortium (OGC) [122] and the Stimulus-Sensor-Observation ontology design pattern [72]. Based on the SSN ontology, a number of developments have been reported in several work such as [99, 22]. However, sensor data quality dimensions have not got enough attention in these work.

[1] http://www.w3.org/2005/Incubator/ssn/ssnx/ssn

Even though accuracy has been mentioned in the SSN ontology, data quality is not only about accuracy. Many work shows that data quality should be defined beyond accuracy [66]. We therefore have developed an ontology based on SSN to describe sensors and their quality dimensions.

The developed ontology contains spatial attributes (i.e. information about sensors' location), temporal attributes (i.e. information about timestamp), thematic attributes (i.e. information about sensor type, measurement, units), and quality attributes (e.g. accuracy, timeliness, completeness).

We align the SSN ontology to OWO as shown in Fig. D.4. OWO can be connected to SSN to share common information such as measurement values from sensors embedded on a wind power plant. At the same time, OWO can still guarantee the complete description of a wind power plant data model. These two ontologies are maintained separately. Paper D gives more details of the ontology alignment.

3.4 The Data Source Handling

This section presents the *data source handling* component which fulfills the requirements *R2, R3* and *R4* presented in Sect. 3.1.2. The *data source handling* component provides the following functionalities:

- integrate data sources coming from offshore wind applications or existing wind databases;
- select a data source from available data sources;
- combine several data sources in order to improve the data quality.

The *communication integration* provides various communication approaches, for instance publish/subscribe allowing to access information through message exchange. ESB is employed to adjust data messages between applications. RESTful Web services are used to support interoperable machine-to-machine interaction over a network. The *information as a service* capability will be beneficial to enable more flexible cross-facility information access. It enables loose coupling to data stores and data model [32]. It also enables business processes and users to work with up-to-date data in critical applications.

A new data source is integrated into the framework by two steps: (1) the connection between the platform where the data source is being deployed and ESB is handled by a connector; (2) a wrapper is developed in order to transfer data from one data format to another. Here, the wrapper solution is selected since the mappings are between a data source (could be a file, a relational database) and the generated RDB. It is noted that the generated RDB is the one generated from the OWO ontology that is described in Sect. 3.3.2.

3.4.1 Data Quality Description

A widespread issue in data integration is the management of data with insufficient quality. It is normal that data are provided without any quality description attached to them. Consequently, it is not clear what the quality (e.g., accuracy or completeness) of the data is. There are also cases where data quality is available but the service that makes data accessible does not provide any method to access the information.

In offshore wind energy, a number of sensors are deployed on a wind turbine and they frequently measure and deliver the data to the users and applications by means of services. As sensors are prone to failures, their results might be inaccurate, incomplete, and inconsistent [125]. Wrong decisions can be made because of poor quality data [127, 66]. Therefore, a way to compute data quality and make it available is important.

Data quality describes the characteristics of data and hence gives users a better view on data they want to request. Data quality has several dimensions which are considered as criteria for selecting the most suitable data source according to users' requests. There are more than 17 data quality dimensions which have been mentioned in the literature, e.g., accuracy, completeness, timeliness, consistency, access security, data volume, confidence, and understandability [140, 84, 14, 49]. In this work, we look at the three most common quality dimensions in the literature, i.e., *accuracy*, *completeness*, and *timeliness* [111]. The other dimensions such as *confidence*, *value-added*, and *coverage* are only suggested by a couple of studies because these dimensions can be either derived from the other dimensions or they are applicable only in a few domains.

3.4.2 Data Source Selection

Given that data quality information is available, how can a system fulfill the user's requests for data with given constraints on data quality? The user only cares about the requested data and its quality, and he does not care about from which data source the data is selected. Thus, answering the request by giving a list of possible data sources is not a good answer. The aforementioned question can be reformulated as follows: how to select the best suited data source among available data sources based upon user's defined quality criteria? Although ESB handles communication between applications, it does not support a way to select the most suitable data source among several available ones.

We propose an approach to handling data sources in ESB based on data quality and semantic technology. This introduces a new level of abstraction that can improve the process of data quality handling with the help of semantic technologies. Based on a user request, the selection process requires a set of quality constraints and a selection dimension. The mandatory constraints describe conditions to be met by the data sources, and the optional selection dimension describes which dimension to use for finding the best data source. The detailed description of the proposed approach can be found in Paper C.

3.4.3 Data Source Combination

Another issue is that sometimes none of the available data sources has the required quality. In this case, it might be possible to improve the quality of data to meet the user's requirement by combining existing data sources. A virtual data source is the result of the combination. Let us now consider three simple combination methods of getting a virtual data source from two existing data sources $D1$ and $D2$.

- D1 (A) D2: taking a conventional average of the data sources D1 and D2.
- D1 $\bigoplus$ D2: use data points from data source D1 if available, otherwise use D2.
- D1 (E) D2: pick up the earliest received data point from either D1 or D2.

Virtual data sources are derived using these combination methods. How are the quality attributes of these virtual sources? Let us consider a case of *Timeliness*.

Timeliness is the average time difference between the moment a data point has been successfully received and the moment it is produced. The timeliness of data source D is calculated using Eq. (3.1):

$$Time(D) = \frac{\sum_{i=1}^{N_D}(t(d_i) - t(r_i))}{N_D} \tag{3.1}$$

where N_D denotes the total number of data points in data source$D$, $t(r_i)$ is the moment when the data point i is produced, and $t(d_i)$ denotes the moment when the data point i is received.

Assume that the timeliness $Time(D1)$ and $Time(D2)$ of $D1$ and $D2$, respectively are two independent exponentially distributed random variables as shown in Eq. (3.2). That said, $Time(D1) = \frac{1}{\lambda_1}$ and $Time(D2) = \frac{1}{\lambda_2}$. If $Time(D1) < Time(D2)$, data from $D1$ arrives earlier than data from $D2$.

$$f(t, \lambda) = \begin{cases} \lambda e^{-\lambda t}, & \text{if } t \geq 0 \\ 0, & \text{if } t < 0 \end{cases} \tag{3.2}$$

Table 3.3 shows the *Timeliness* of the virtual data sources for the three combination methods.

Table 3.3: The timeliness of the virtual data source

Method	**Timeliness**
D1 (A) D2	$\approx \frac{3}{2} Time(D1)$
D1 $\bigoplus$ D2	$\frac{P(D1)*Time(D1)+\overline{P(D1)}*P(D2)*Time(D2)}{P(D1)+\overline{P(D1)}*P(D2)}$
D1 (E) D2	$\frac{Time(D1)*Time(D2)}{Time(D1)+Time(D2)}$

The combination results show that by applying the (E) method, the virtual data source can get a better timeliness in comparison with the timeliness of $D1$ and $D2$. However, the (A) method makes the timeliness of the virtual data source worse than the timeliness of $D1$ and $D2$. For the $\bigoplus$ method, it varies from case to case. More details of the data source combination can be found in Paper C.

3.5 The Information Provisioning

The third component of the proposed framework presented in Fig. 3.2 is the *information provisioning*. It is used to provide efficient ways to access the reliable data. The *service layer* provides consumers with sufficient detail to invoke the business functions exposed by a provider of the service. All business workflows are handled by the *business process layer*. The *presentation layer* provides user-friendly human machine interfaces to end-users.

In this section, we look into two important issues related to information provisioning of data integration: how data quality is provided and presented to users and how to formally model derived data. This *information provisioning* component fulfills the requirements *R5* and *R6* presented in Sect. 3.1.2.

The *information provisioning* component provides the following functionalities:

- access to wind data using REST architecture style;
- creating new data sources by formulating concept dependencies.

3.5.1 Data Accessibility & Information Presentation

Data from offshore wind partners are made available through web services. Making information available through web applications increases the availability of information due to platform independence and easy access. We use REST-based web services which are presented in Sect. 2.3.3 to enable accessibility to wind data. Table 3.4 shows an example of identifying wind data resources as unique URIs.

Table 3.4: An example of resource definition for WTUR

URI	Method	Description
/LDs	GET	List IDs of all windmills
/LDid/LNid	GET	Get information of a wind turbine component whose ID is LNid and the component belongs to a particular windmill whose ID is LDid
/LDid/LNid	DELETE	Delete information of a wind turbine component whose ID is LNid
/LDid/WTUR/TotWh	GET	Get total active energy production of a windmill whose ID is LDid
/LDid/WTUR/TurSt	PUT	Update status of a particular windmill whose ID is LDid

Wind data are provided to interested partners in different forms. The data can also be displayed in graphs in order to provide better visualization to operators who are sitting in the operations center. Online monitoring can be combined together with intelligent condition-based applications to provide the operators with better support on making decisions regarding performance of wind turbine components. More details of monitoring wind turbine components can be found in Paper E.

3.5.2 Data Derivation

Missing data can be caused by network disconnection, device faults, and software bugs. In some cases, where monitoring of devices or components is extremely important, a single missing value of a data point could lead to wrong predictions or damage of components. In many domains such as wind energy, many applications related to prediction and monitoring are employed. The power output and weather can be predicted. The performance of a wind turbine blade can be monitored by applications that consume real-time data. The performance of these applications relies very much on data collected from the wind turbines. Missing a single data item in the set of input data to these applications can make the applications produce wrong output or no output at all. In this case, the missing data item needs to be derived from other available data items. Derivation of data also plays a significant role in decision support systems [110]. For instance, in time-series data analysis, missing data that are located in the middle of a time-series have a high influence on the efficiency of algorithms that are used to reveal hidden temporal patterns such as vector autoregression and exponential smoothing [151]. In some cases the missing data can be derived from possible relations between the concepts. This section presents an extension of the work that is described in Sect. 3.4.3. Here, we investigate the possibility of formally describing derived data from the relations between concepts in ontologies from a user interface perspective.

3.5.2.1 Derived Data Definition

Data are classified into two categories: base data and derived data [58]. Base data are those data obtained from data sources. Derived data are those data obtained by combining or computing from base data. Basically, the combination and computation of base data are based on relations between domain concepts. Derived data are described by derived classes and derived attributes. A derived attribute is an attribute that is derived from other attributes in the same class or from different classes that have relationships with the class that contains the attribute. If all attributes of a class are derived, the class is called derived class [11].

Derived data gives advantage for storing data since there is no need to store derived data in a database because such data can be derived from other data that are already stored in the database. Another advantage is that the structure of the data storage is undisclosed to users, derived attributes are accessed via user interface.

3.5.2.2 Derived Data Modeling

Guaranteeing the correctness of derived data is an important task because applications that use the data might produce wrong results. Therefore, derived data need to be handled in such a way that its correctness is ensured. Formally modeling of derived data can help us to figure out different aspects of handling the data, and hence guaranteeing the correctness.

Basically, derived data can be described in OWL using annotation fields for classes and properties, for example, the work reported in [68] describes an approach to attaching formulas directly to properties in ontologies. However, this approach does not give users the full control and the possibility to detect bugs when the descriptions get complicated. We therefore decide to use UML and Object Constraint Language (OCL) to describe derived data. UML can model domain specific concepts, in particular the wind energy [51]. OCL is a complement of UML. It is used to express constraints in UML models [141] in order to makes the models precise, consistent, and complete.

Let us consider an offshore wind farm scenario where many sensors are located on a wind turbine to capture information. What if one of them loses the connection? Information related to that one will be lost. How can we utilize other devices to derive that information so that the monitoring of the wind turbine is still ensured? Fig. 3.5 shows how to make use of derived data from the two wind domain concepts: wind speed and power output. Wind speed is a part of the WMET (wind turbine meteorological information) class and power output is a part of the WTUR (the wind turbine general information) class. Both wind speed and power output can be measured using sensors or derived from each other.

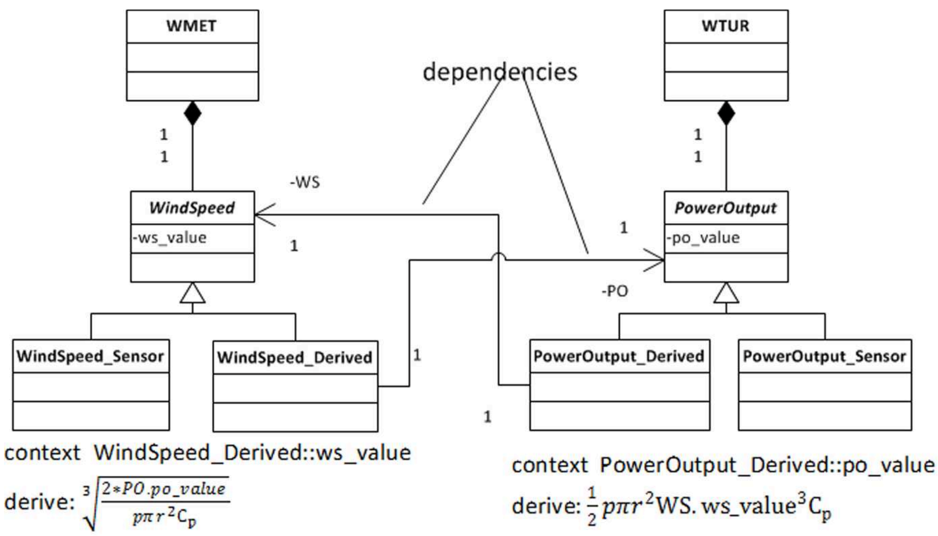


Figure 3.5: Derived data between two concepts

The equations used the OCL constraints shown in Fig. 3.5 are derived from basic mathematical relation between wind speed and power output. The relation is expressed in Eq. (3.3) [95].

$$P_{avail} = \frac{1}{2}\rho\pi r^2 v^3 C_p \tag{3.3}$$

where P_{avail} denotes the available power output (W), ρ denotes air density (kg/m^3), r denotes blade length (m), v is the wind speed (m/s), and C_p denotes the power coefficient. Please note that the power coefficient is not constant; it depends on other factors such as rotational speed of the turbine, pitch angle, and angle of attack [96]. The derived data have the same quality attributes as presented in Sect. 3.4.1. An approach that is similar to the combination methods presented in Sect. 3.4.3 can be developed to obtain the quality of the derived data. More details of data derivation can be found in Paper D.

3.6 Framework Implementation

We prove the applicability of the proposed framework by developing some prototypes for important parts of the framework. We present different prototypes in Papers A, B, C, and E. The prototypes described in Papers A and B are used to support our proposal of the *semantic model* presented in Sect. 3.3. The prototypes described in Papers C & E are used to support our solutions concerning the *data source handling* presented in Sect. 3.4. Eventually, our proposal of the *information provisioning* component, which is presented in Sect. 3.5, is proved by the prototype described in Paper E.

This section first presents technologies that are used in the prototypes. An implementation of the framework is then described.

3.6.1 Technology Selection

Among ESB open source frameworks such as such as PEtALS ESB, Mule ESB, ServiceMix, Open ESB [136], we select Mule ESB[2], a lightweight integration framework to handle data subscription from third parties.

Mule ESB is not based on Java Business Integration (JBI), but it provides seamless support for JBI containers [109]. Hence, it allows components of other ESBs such

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

as ServiceMix, which are based on the JBI model, to be used alongside Mule ESB. Besides, Mule ESB is provided together with an IDE (Integrated Development Environment), namely Mule studio, which makes the process of flow design much easier. Mule ESB also provides many transports, for instance REST, SOAP, JMS (Java Message Service).

Restlet[3] is a high-level API based on the HTTP servlet technique [113]. It provides an abstraction of REST applications, resources, and data representations. Applications developed using Restlet can run on any Servlet engine [45]. Tomcat 7.0[4] is used to provide HTTP web server and servlet container, and JDK 1.6 (Java Development Kit) for supporting Java application. We use WSDL to describe SOAP-based web services. In addition, the WADL (Web Application Definition Language) extension provided in the Restlet framework is also implemented in the prototypes in order to provide definitions of available services.

AJAX technology was used to handle requests and responses between server and client. Flot[5], a JavaScript plotting library is used to produces graphical plots on a web browser. We also use Microsoft SQL Server and MySQL to manage relational databases in different prototypes.

3.6.2 Sample Test Cases

For the prototypes, the following two test cases are considered. The first one concerns integrating existing data sources and migrate data to a local data storage. The second one is to request real-time data that are made available through web services.

Test case 1

Description: There are two wind databases from Statkraft and Agder Energi. A user requests for a wind turbine information from the Hitra wind farm. The Statkraft database stores data from the Hitra wind farm.

Result: the data from the Statkraft database is returned to the user.

Test case 2

Description: There are three wind speed sensors that measure wind speed at a particular area. The measured data are made available through web services and our

[3] http://restlet.org/
[4] http://tomcat.apache.org/index.html
[5] http://www.flotcharts.org/

system is aware of the web service addresses. A user requests for real-time wind speed measured at that area. The user also gives some constraints on the requested data, e.g., the completeness of the data must be more than 75% is chosen for the restriction process and timeliness is selected as the criteria for data source selection.

Result: first, the restriction process is executed, data sources 2 and 3 are selected. Then the selection based on timeliness is executed. As the result, data source 3 is selected since it has better timeliness compared to data source 2.

We setup a prototype system as shown in Fig. 3.6 to run the test cases. The horizontal dashed lines divide the figure into three parts that are marked with numbers (1), (2), and (3). These parts correspond to Part(1), Part(2), and Part(3) presented in Fig. 3.1.

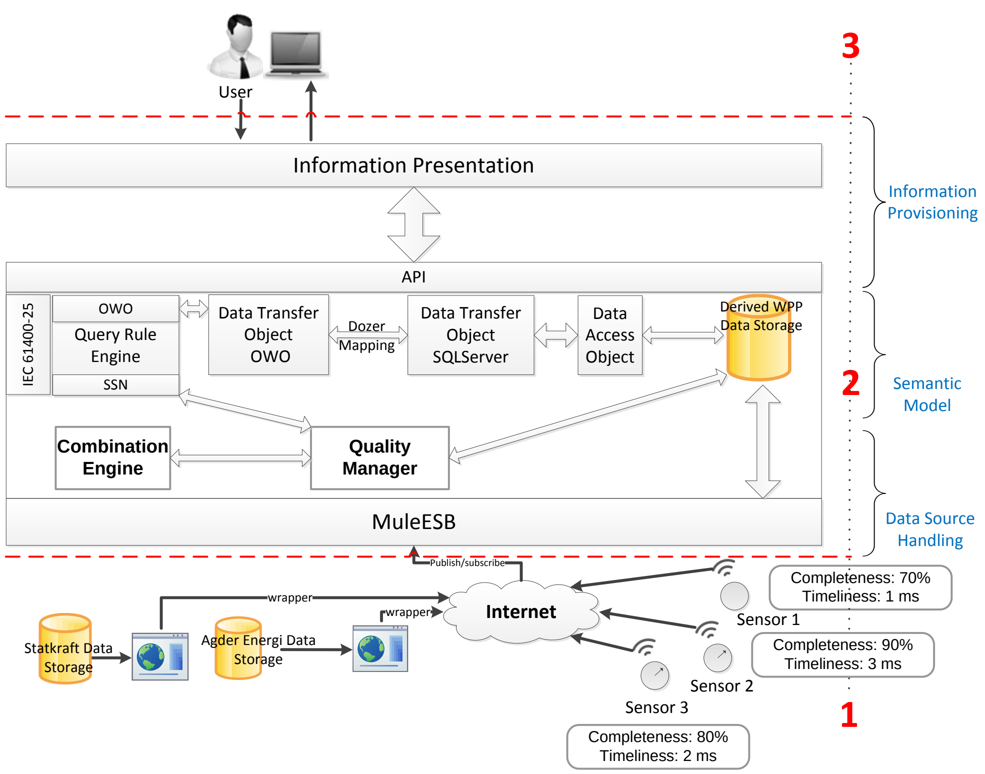


Figure 3.6: A data integration prototype

Wind data are made available through web services and the data are pushed to our system over the Internet from data providers such as Statkraft and Agder Energi. SQWRL, Protégé-OWL API, Jess rule engine, Pellet reasoner are used in order to answer users' queries against OWO. The WPP Data Storage is a Microsoft SQL Server database that has the schema derived from OWO. Data from the database is first loaded to OWO and then is used for answering queries from users. Data Transfer Object OWO (DTO OWO) contains Java classes that are generated from OWO. Data Transfer Object SQLServer (DTO SQLServer) contains Java classes that are generated from the derived RDB. Data Access Object (DAO) handles the access to the database through JDBC (Java Database Connectivity) interfaces. The mappings from DTO OWO to DTO SQLServer and vice versa are handled by Dozer[6], a Java Bean to Java Bean mapper. Wrappers are used to provide access to the Statkraft and Agder Energi databases.

The launching process of data from WPP Data Storage into OWO is described as follows: (1) data in WPP Data Storage are accessed by calling built-in functions provided by DAO; (2) data are then fetched into DTO SQLServer; (3) Dozer converts data in DTO SQLServer to DTO OWO; (4) data are then fetched into OWO using Protégé OWL API. Users can then make requests for data using the SQWRL language against OWO.

Fig. 3.7 shows two tables from a Statkraft wind energy database. The first table displays stationID, timestamp, mean value of pitch control A position, mean value of generator speed (wtc_GenRpm_mean), and mean value of nacelle position. The second table shows timestamp, stationID, and active power (wtc_ActPower).

StationId	TimeStamp	wtc_PitcPosA_mean	wtc_GenRpm_mea	wtc_NacelPos_mea
2,300,249	2012-01-01 00:00:00	-0.680968	1002.45	127.982
2,300,250	2012-01-01 00:00:00	0.498	1501.21	121.763
2,300,251	2012-01-01 00:00:00	0.522061	1501.14	320.471
2,300,252	2012-01-01 00:00:00	0.396181	1501.35	160.459
2,300,274	2012-01-01 00:00:00	0.598759	1501.04	313.628
2,300,275	2012-01-01 00:00:00	-0.531986	1001.79	162.462
2,300,276	2012-01-01 00:00:00	0.471215	1501.24	176.003
2,300,277	2012-01-01 00:00:00	-15.5433	738.791	121.037
2,300,278	2012-01-01 00:00:00	-0.705978	1002.65	313.919
2,300,279	2012-01-01 00:00:00	0.591489	1501.06	113.381

TimeStamp	StationI	wtc_ActPower_n	wtc_VoltPhR_n
2012-01-01 00:00:00	2,300,249	126.095	389.8
2012-01-01 00:00:00	2,300,250	264.843	389
2012-01-01 00:00:00	2,300,251	248.743	390.1
2012-01-01 00:00:00	2,300,252	303.276	389.7
2012-01-01 00:00:00	2,300,274	219.986	390.6
2012-01-01 00:00:00	2,300,275	86.9156	388.5
2012-01-01 00:00:00	2,300,276	277.181	389.4
2012-01-01 00:00:00	2,300,277	74.8694	376
2012-01-01 00:00:00	2,300,278	130.067	389.5
2012-01-01 00:00:00	2,300,279	229.706	389.2

Figure 3.7: Sample of 2 tables from the Statkraft database

The mapping step is necessary since there is a difference in naming between third party's data sources and our system. Table 3.5 shows an example of mappings from the Statkraft database to our generated database schema. In this example, we consider only 1 to 1 mapping. The first column shows data properties in OWO. The second columns shows columns' names that come from different tables in Statkraft database.

The setup allows us to achieve the results as described in the two test cases 1 and 2.

[6]http://dozer.sourceforge.net/

Table 3.5: An example of mapping from Statkraft database to the derived database

OWO property	Column in RDB	Data type
WROT		
hasOilPressure	wtc_HubPresA_mean	real
hasPitchAngleSetPoint	wtc_PitcPosA_mean	real
hasPitchAngleRef	wtc_PitchRef_BladeA _mean	real
WGEN		
hasCurrentPhR	wtc_AmpPhR_mean	real
hasCurrentPhS	wtc_AmpPhS_mean	real
hasCurrentPhT	wtc_AmpPhT_mean	real
hasPh2PhVoltagePhR	wtc_VoltPhR_mean	real
hasPowerFactor	wtc_CosPhi_mean	real

3.6.3 Metadata Management in Smart Grids

As another proof of concept, we have tried to apply our solutions to the management of metadata in smart grids. Smart grids enable consumers to utilize lower tariff charges during off-peak periods and energy producers to react efficiently during peak periods [39]. In smart grids, a huge number of smart meters, sensors, smart appliances, and other smart devices are employed and connected to Internet. This leads to issues in handling and processing vast amounts of data, and integrating these devices in a network so that they can communicate with each other effectively. In order integrate these data sources, metadata management needs to be considered since it is the key to make data integration successful [69]. We consider three main problems concerning the management of metadata in smart grids. They are (1) knowledge sharing and data exchange, (2) derived data from relations between concepts, and (3) data quality as metadata. We apply solutions proposed throughout the development of our framework to the management of metadata. We show that the semantic technologies are mature enough to be used in the developments of smart grids. Paper D gives more details of management of metadata in smart grids.

3.7 Chapter Summary

In this chapter, we have presented our main original contributions to knowledge by proposing a data integration framework for offshore wind farms. The framework consists of three components: the *semantic model*, the *data source handling*, and the *information provisioning*. The *semantic model* is the core of the proposed architecture. It covers the key concepts in the offshore wind domain and their semantic relationships. It defines the basic agreements on data exchange format and what kind of data to exchange. The *data source handling* focuses on data acquisition from different sources, such as data storage, sensors, and existing databases. There are 3 layers, service integration, ESB, and communication integration. Finally, the *information provisioning* provides efficient ways to access to the reliable data. The requirements for the proposed framework have been fulfilled by the framework components.

Chapter 4

Evaluation and Discussion

This chapter presents the evaluations of the proposed ontology, algorithms, and the proposed framework. The contributions in the dissertation are also evaluated against the research questions posed in Sect. 1.2. The chapter ends with the discussion of related work and the contributions.

4.1 Evaluation

In this section we present formal evaluation of the proposed ontology and algorithms in the dissertation. We also evaluate the proposed framework against the requirements presented in Sect. 3.1.2. Eventually, we evaluate our contributions in the dissertation against the research questions posed in Sect. 1.2.

4.1.1 Evaluation of the Proposed Ontology & Algorithms

The following six dimensions of ontology evaluation [108] are considered when evaluating OWO: logical consistency, modeling issues, ontology language specification, real world representation, semantic applications, and human understanding. First, we use Protégé and the Pellet reasoner to check for inconsistencies in the OWO ontology. Since the OWO ontology is based on the IEC 61400-25 standard, its human understanding and real world representation aspects are guaranteed. We have successfully implemented the OWO ontology in some prototypes. Thus, the semantic applications dimension is ensured. Besides, we used OOPS! - an OntOlogy Pitfall Scanner! [108] that covers the aforementioned ontology evaluation dimensions to evaluate OWO. The following pitfalls were detected in the first attempt: missing annotations, missing domain or range in properties, missing inverse relationships, untyped property. No critical pitfall was found. We then corrected the ontology based on the recommendations from the OOPS!.

Apart from the experimental evaluation presented in Sects. 3.6.1 and 3.6.2, we formally evaluate all the proposed algorithms in the dissertation by considering their time complexity. The complexity of the algorithms is shown in Table 4.1.

Table 4.1: Complexity of the proposed algorithms

	Time Complexity	**Note**
OWO to RDB schema	$O(n^3)$	n is the number of concepts
Data Source Selection	$O(n^3)$	n is the number of data sources
Data Source Combination	$O(n)$	n is the number of data items

In this dissertation, we have managed to optimized the algorithm for combining the data sources. We have focused on this algorithm since dealing with virtual data sources is one of the important contributions in our work. The data source selection and OWO to RDB schema algorithms currently work well with small dataset and we think that there are rooms for optimizing these algorithms as well. It is noted that the OWO to RDB schema algorithm is executed only once when the global physical relational database schema is generated for the first time. The algorithm does not depend on the size of instance (i.e., ABox), but it depends on the size of TBox.

4.1.2 Evaluation of the Proposed Framework

The proposed framework is evaluated against the requirements presented in Sect. 3.1.2. Sect. 3.3 presents the *semantic model* component that fulfills requirement *R1*, Sect. 3.4 describes the *data source handling* component that fulfills requirements *R2, R3* and *R4*, and Sect. 3.5 discusses the *information provisioning* component that fulfills requirements *R5* and *R6*. The non-functional requirements *R7* and *R8* are fulfilled since the framework is platform independent and the proposed OWO is extensible. Besides, the framework can be applied to other industries such as maritime. The proposed framework is proved to be applicable as many prototypes of the framework components have been developed throughout the research. The framework does not cover the information and communication security part due to time limits of the project. Scalability is not considered in this work since it depends on the software components that are used to build the framework.

4.1.3 Evaluation against the Research Questions

We evaluate our work against the research questions posed in Sect. 1.2.

RQ1: *How to solve semantic inconsistency which has become an important problem of knowledge sharing and data exchange among users or applications?* We have answered the question by using semantic technologies. In particular, ontology is used to describe wind turbine components. We have developed an offshore wind ontology which is a global schema over data sources. The complete answer to this research question is described in Sect. 3.3.

RQ2: *Data sources are considered as autonomous, distributed and heterogeneous systems. How to manage, integrate, and unify them systematically?* Enterprise service bus plays a middleware role to connect different sources to our system. Wrappers are developed to handle differences in format of data sources. Sect. 3.4 presents the answer.

RQ3: *How to select the most suitable data source? How to provide data for users if the requested data source is not available?* Data quality dimensions for each data source should be associated to the data source. The most suitable data source therefore can be selected according to users' defined requirements. If the requested data source is not available, a combination method can be used to combine available data sources. The answer is also presented in Sect. 3.4.

RQ4: *How to provide data with quality descriptions for users based on users' requirements?* Data quality dimensions can be computed based on either historical data or a reference data source. The data quality descriptions are then attached to the data source and provided through services. Sect. 3.4.1 describes the answer.

RQ5: *Data quality dimensions are metadata. How to manage this kind of metadata?* Metadata can be managed efficiently by using semantic technologies. We have extended the SSN ontology by more adding data quality attributes. In addition, we have aligned the OWO to the extended SSN. Sect. 3.3 highlights the main points of the answer.

RQ6: *Missing data can be caused by network disconnection, device faults, and software bugs. Is there any way to fill in the missing data? If yes, how to describe* *the solution formally?* It is possible to fill in the missing data by deriving data based on the relations between the concepts. We use UML and OCL to formally describe the derived data. The detail answer can be found in Sect. 3.5.2.

The dissertation has completely described the answers to the research questions posed in the beginning of the research. The novelty of this work includes the development of a holistic framework for data integration in a complex and new developing domain - the offshore wind energy domain, the development of an offshore wind ontology that is an important foundation for data exchange and knowledge sharing, and other novel contributions that have been described above. One of the targets of this work is to utilize as much as possible available technologies for building the framework. Indeed, the following technologies and standards have been used in the dissertation: the IEC 61400-25 standards, the IEC 61850 standards, the Oil & Gas Integrated Operations, the METHONTOLOGY and Neon methodologies for ontology development, the web ontology language - OWL, the Pellet reasoner, the Jess rule engine, REST architecture style, SOA, and OCL.

4.2 Related Work & Discussion

Data integration has been addressed in other industries such as oil & gas, maritime where common understanding and agreement on data exchange and knowledge sharing between interested partners/actors need to be considered upon any development and implementation. This section briefly discusses our contributions with respect to related work within the wind energy and oil & gas industry.

4.2.1 Wind Data Integration

Both atomistic and holistic views of our contributions are discussed in this section. As the proposed framework is based on different technologies, related work to some parts of the framework are selected to discuss.

- The authors of [103] propose an ontology model for wind turbines' condition monitoring. The ontology model is used to describe wind turbine components and their associated faults and symptoms. Another work reported in [90] proposes a holistic condition monitoring system where an ontology is considered as a central repository of all information available in wind turbines. Our proposed ontology model is based on the IEC 61400-25 which is a specific standard for monitoring and control of wind power plants.

- A schematic presentation of a RAMS database was introduced by the authors of [61]. A shortcoming of this schema is that the authors did not take the semantics of data into consideration. Hence, the potential of data cannot be completely exploited. Our RAMSI database is derived from OWO and therefore it can bridge the semantic gap between the ontology and the relational database.

- There are several ways of selecting data sources such as content-based filtering [35], social information filtering [120], agent-based selection [126], and quality-based selection [91]. Content-based filtering is a traditional and static way of selecting a data source out of a list of available data sources. This approach filters data sources based on users' keywords. When users send requests for data, the content-based filtering statically selects the data source with more relevant description. This approach might not solve the selection problem if there are data sources with the same descriptions. Another approach is the social information filtering. It refers to a sort of techniques to provide personalized recommendations for users according to the similarities of their interests. This approach is common in sites, e.g., Amazon and LinkedIn. The third category of selection methods is the agent-based approach. Agents evaluate data sources by communicating, cooperating, and rating each other. Each agent can make decision and work autonomously as well. The quality-based approach takes into consideration the importance of data quality. A data source is selected based on data quality dimensions given in users' defined requirements. Our work is based on this approach since data quality is one of important aspects of data integration. We also use semantic technology to solve inconsistency issues and enable semantic description for sensor networks.

- Besides selecting data sources, there have been studies concerning combination of data sources in order to fulfill users' requests on data. Most of the studies are in the field of multi-sensor data fusion which employs different techniques to generate a better virtual sensor, for instance, artificial intelligence, pattern recognition, statistical estimation are used to fuse multi-sensor data [59, 82]. Different from these work, we propose an approach to combining data sources based on data quality dimensions.

Having looked at our work as a whole, we find out that BazeField Wind is related to our work. It is a commercial wind farm management system developed by Baze

Technology[1]. BazeField is used to monitor Sheringham Shoal offshore wind farm which consists of 88 Siemens 3.6MW wind turbine. The system is based on international standards such as the IEC 61400-25. Main modules of BazeField Wind include monitoring, analysis, and operation management. The monitoring module contains basic functionality to monitor the performance of wind turbines and wind parks. The analysis module provides weather analysis, power analysis, turbine alarm, etc. The operation management module supports performance management, production planning, task management, reporting, etc. Since this system is a commercial, we do not know the technologies on which the system is based. But in general, the system looks similar to our work from the data integration aspect. It is not clear to us whether the semantic inconsistency issue of wind data has been taken into consideration in the development of BazeField Wind.

4.2.2 Oil & Gas Integrated Operations

Integrated Operations is a project proposed by the Norwegian Oil Industry Association with the purpose of improving operational decision support for offshore installations from onshore operations centers by implementing ICT. An estimation from the OLF shows that the economic potential of implementing IO is expected to be more than 250 billion Norwegian kroner in net present value [101, 132].

The IO project has taken into consideration the advantages of semantic web technologies and ISO standards. The ISO 15926 standard [2], namely "Integration of life-cycle data for process plants including oil and gas production facilities", is developed to provide a methodology for data integration across disciplines (e.g., health, safety, environment, drilling, and reservoir & production) within the oil & gas industry. Based on the methodology of the ISO 15926, an oil & gas ontology has been developed by the POSC Caesar Association (PCA) in collaboration with the Norwegian offshore industry [133].

A big difference between the offshore wind and oil & gas industries is that there is personnel onboard oil platforms while in wind farms human presence is not normally necessary, the system operates automatically and can be controlled by the operations center. Another difference is the number of wind turbines; there can be hundreds in a wind farm (an asset), whereas a single oil platform is often regarded as an asset on its own. Despite the fact that there are some differences between these two industries, there are still some common characteristics, for instance, there

[1] http://www.bazetechnology.com/

is a need of an operations center to support offshore activities. Data integration challenges are similar in the two industries where data come from different sources in different formats. We took into consideration the IT architecture for IO when we designed our framework, thus our framework has the similar design pattern with the IO architecture. Our work differs from the IO in some aspects such as (1) our work is designed for the wind energy domains; (2) we focus on the semantics and integration aspects; (3) we look into the possibility of bridging the semantic gap between OWO and RBD; and (4) we investigate approaches to deriving new data sources.

Chapter 5

Conclusions

This chapter summarizes our contributions and gives some concluding remarks. The future outlook is presented at the end of the chapter.

5.1 Summary of Contributions

This dissertation explores the possibility of improving availability and accessibility of offshore wind data by solving some data integration issues. Our original contributions to knowledge are summarized as follows.

- **C1**: We have proposed a data integration framework for offshore wind farms. The framework provides a holistic system view over the wind domain. The framework is presented in Sect. 3.2. Details of the framework can be found in Paper A.
- **C2**: We have manually developed an offshore wind ontology (OWO) to explore the semantics of wind data and enable knowledge sharing and data exchange. The ontology is based on the approved international standard IEC 61400-25 and it is open for future extension. In addition, the relation between OWO and relational database has been investigated. More details can be found in Sect. 3.3 and Paper B.
- **C3**: We have presented a quality-based approach to managing, selecting, and providing the most suitable data source for users based upon their quality requirements. The approach gives users the possibility of getting the most suitable data source from the available ones. It also increases the chance to

find the requested data source by combining multiple data sources so that the users' quality requirements are met. We have also proved that in some cases, the quality of combined data source can be better than individual data sources' quality. This contribution is presented in Sect. 3.4.2, Sect. 3.4.3 and Paper C.

- **C4**: We have presented a way to formally describe derived data based on the relations between concepts. Sect. 3.5.2 and Paper D give more details of this contribution.

- **C5**: We have solved some issues regarding the management of big data metadata in Smart Grids with a focus on offshore wind as an energy generator. In particular, we have used ontologies to manage offshore wind metadata and sensors' metadata. Details of this contribution can be found in Sect. 3.6.3 and Paper D.

The layout of the included papers is shown in Fig. 5.1.

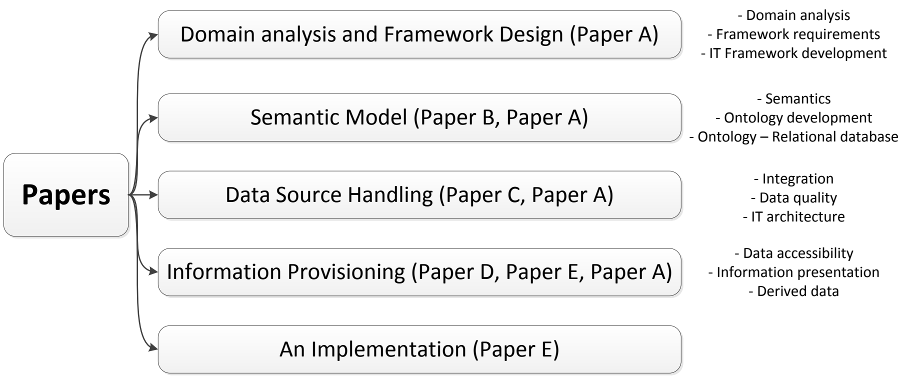


Figure 5.1: Paper layout

5.2 Concluding Remarks

We have followed the design paradigm which consists of four steps (i.e., state requirements, state specifications, design and implement the system, and test the system) to carry out our research. The same paradigm has been also used to tackle each of the research questions.

The proposed framework is equally applicable to problems of other industries which are compatible with the system view, for example, the maritime industry, where common understanding and agreement between participants are handled upon operations. The architecture may be applied to other industries with small changes.

More and more data integration issues are being tackled and solutions to the issues can be improved as new technologies are introduced. We have shown that semantic technologies can be used to solve the inconsistency issue in data integration.

Our work provides a holistic view of improving the accessibility and availability of reliable offshore wind data.

5.3 Future Outlook

In this dissertation so far, information and communication security were not taken into account. It is assumed that all communications are handled using secure channels. It is apparent that the control and monitoring of offshore wind farms can be handled easily at an operations center. However, there are many security vulnerabilities related to data exchange and communications between wind power plants. As future work, we plan to carry out research on security requirements for remote operations of offshore wind farms and security mechanism for making offshore wind farm communication secure. The proposals in the IEC 62351 standard are considered as a starting point.

We want to extend the current work by solving more problems related to metadata management. Metadata provide information about data that are stored in a database without having accessed it. A way of managing metadata has been investigated in this work. However, we have not discussed the quality of metadata. Metadata with high quality can guarantee that proper sensing resources and data sources are found and data are used properly. The quality of metadata definitely affects the use of data and decisions that are based upon the data. We plan to extend the work on data quality to metadata quality research.

As a part of future power grids, offshore wind plants will be integrated in grids in order to make the grids smarter and smarter. A comprehensive data integration framework for smart grids could be developed based on our framework.

The Oil & Gas industry has been in business for a long time and many solutions have been proposed to tackle the data interoperability issue with the industry. The standard ISO 15926 initiated by the POSC Ceasar Association is one of the solutions. Currently, many Oil & Gas companies are expanding their business to the offshore wind domain. As presented in this dissertation, data interoperability is also one of the issues within the offshore wind industry. In order to help these companies reuse their solutions, it is a good idea to extend the ISO 15926 by taking into consideration the proposed offshore wind ontology - OWO.

References

[1] IEC 61850 communication networks and systems in substations - part 7-3: Basic communication structure for substation and feeder equipment - common data classes (2003)

[2] ISO 15926 industrial automation systems and integration - integration of life-cycle data for process plants including oil and gas production facilities (2003)

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

[4] Offshore Wind Power: Big Challenge, Big Opportunity: Maximising the Environmental, Economic and Security Benefits. Carbon Trust (2008)

[5] IEC 61400 wind turbines - part 3: Design requirements for offshore wind turbines (2009)

[6] Ackoff, R.L.: From data to wisdom. Journal of Applied Systems Analysis **16**, 3–9 (2010)

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

[8] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: The description logic handbook: Theory, implementation and applications. Cambridge university press (2003)

[9] Baader, F., Horrocks, I., Sattler, U.: Description logics as ontology languages for the semantic web. In: Mechanizing Mathematical Reasoning, pp. 228–248. Springer (2005)

[10] Baclawski, K., Kokar, M., Waldinger, R., Kogut, P.: Consistency checking of semantic web ontologies. The Semantic Web - ISWC 2002 pp. 454–459 (2002)

[11] Balsters, H.: Modelling database views with derived classes in the UML/OCL-framework. In: UML 2003-The Unified Modeling Language. Modeling Languages and Applications, pp. 295–309. Springer (2003)

[12] Barzdins, J., Barzdins, G., Cerans, K., Liepins, R., Sprogis, A.: OWLGrEd: a UML style graphical notation and editor for OWL 2. In: OWLED (2010)

[13] Battle, R., Benson, E.: 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 (2008)

[14] Baumgartner, N., Gottesheim, W., Mitsch, S., Retschitzegger, W., Schwinger, W.: Improving situation awareness in traffic management. In: Proc. Intl. Conf. on Very Large Data Bases (2010)

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

[16] Berners-Lee, T., Fischetti, M.: Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor. HarperInformation (2000)

[17] Bertino, E.: Data security. Data & Knowledge Engineering **25**(1–2), 199 – 216 (1998)

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

[19] Bitar, E., Rajagopal, R., Khargonekar, P., Poolla, K., Varaiya, P.: Bringing wind energy to market. Power Systems, IEEE Transactions on **27**(3), 1225–1235 (2012)

[20] Botterud, A., Wang, J., Bessa, R., Keko, H., Miranda, V.: Risk management and optimal bidding for a wind power producer. In: Proceedings of the IEEE PES General Meeting 2010 (2010)

[21] Breton, S.P., Moe, G.: Status, plans and technologies for offshore wind turbines in Europe and North America. Renewable Energy **34**(3), 646 – 654 (2009)

[22] Bröring, A., Maué, P., Janowicz, K., Nüst, D., Malewski, C.: Semantically-enabled sensor plug & play for the sensor web. Sensors **11**(8), 7568–7605 (2011)

[23] Bryson, J.M.: What to do when stakeholders matter: stakeholder identification and analysis techniques. Public Management Review **6**(1), 21–53 (2004)

[24] Calì, A., Calvanese, D., De Giacomo, G., Lenzerini, M.: On the expressive power of data integration systems. In: Conceptual Modeling - ER 2002, pp. 338–350. Springer (2003)

[25] Chandrasekaran, B., Josephson, J.R., Benjamins, V.R.: What are ontologies, and why do we need them? Intelligent Systems and Their Applications, IEEE **14**(1), 20–26 (1999)

[26] Chappell, D.A.: Enterprise Service Bus. O'Reilly media (2009)

[27] Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J., Widom, J.: The TSIMMIS project: Integration of heterogenous information sources. Information Processing Society of Japan (1994)

[28] Christiansen, P., Jørgensen, K., Sørensen, A.G.: Grid connection and remote control for the horns rev 150 mw offshore wind farm in denmark. In: Proceedings of the 2nd International Workshop on Transmission Networks for Offshore Wind Farms, pp. 29–30 (2000)

[29] Comer, D.E., Gries, D., Mulder, M.C., Tucker, A., Turner, A.J., Young, P.R., Denning, P.J.: Computing as a discipline. Communications of the ACM **32**(1), 9–23 (1989)

[30] Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., et al.: The SSN ontology of the W3C semantic sensor network incubator group. Web Semantics: Science, Services and Agents on the World Wide Web **17**(0), 25 – 32 (2012)

[31] Cruz, I.F., Xiao, H.: The role of ontologies in data integration. Engineering intelligent systems for electrical engineering and communications **13**(4), 245 (2005)

[32] Dan, A., Johnson, R., Arsanjani, A.: Information as a service: Modeling and realization. In: Systems Development in SOA Environments, 2007. SDSOA'07: ICSE Workshops 2007. International Workshop on. IEEE (2007)

[33] Delia, P., Borg, A.: Mule 2: A Developers Guide. Apress (2008)

[34] Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A.: Reasoning in description logics. Principles of knowledge representation **1**, 191–236 (1996)

[35] Dumais, S.T., Furnas, G.W., Landauer, T.K., Deerwester, S., Harshman, R.: Using latent semantic analysis to improve access to textual information. In:

Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 281–285. ACM (1988)

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

[37] Elmasri, R., Navathe, S.B.: Fundamentals Of Database Systems, 5 edn. Addison-Wesley (2007)

[38] Esteban, M., Diez, J., López, J., Negro, V.: Why offshore wind energy? Renewable Energy **36**(2), 444 – 450 (2011)

[39] ETP: Smart Grids - Strategic Deployment Document for Europe's Electricity Networks of the Future (2010)

[40] Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering. AAAI-97 Spring Symposium Serie (1997)

[41] Fichaux, N., Wilkes, J.: Oceans of Opportunity: Harnessing Europe's Largest Domestic Energy Resource. EWEA (2009)

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

[43] Fielding, R.T.: Architectural styles and the design of network-based software architectures. Ph.D. thesis, University of California (2000)

[44] Friedman, M., Levy, A., Millstein, T.: Navigational plans for data integration. AAAI/IAAI **1999**, 67–73 (1999)

[45] Fu, C., Belqasmi, F., Glitho, R.: RESTful web services for bridging presence service across technologies and domains: an early feasibility prototype. Communications Magazine, IEEE **48**(12), 92–100 (2010)

[46] Gali, A., Chen, C., Claypool, K., Uceda-Sosa, R.: From ontology to relational databases. In: Conceptual Modeling for Advanced Application Domains, *Lecture Notes in Computer Science*, vol. 3289, pp. 278–289. Springer Berlin / Heidelberg (2004)

[47] Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J.: Modelling ontology evaluation and validation. Springer (2006)

[48] García-Jiménez, F., Martínez-Carreras, M., Gómez-Skarmeta, A.: Evaluating open source enterprise service bus. In: e-Business Engineering (ICEBE), 2010 IEEE 7th International Conference on, pp. 284–291. IEEE (2010)

[49] Geisler, S., Weber, S., Quix, C.: Ontology-based data quality framework for data stream applications. In: 16th International Conference on Information Quality. Adelaide, AUS (2011)

[50] Gennari, J.H., Musen, M.A., Fergerson, R.W., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F., Tu, S.W.: The evolution of Protégé: an environment for knowledge-based systems development. International Journal of Human-computer studies **58**(1), 89–123 (2003)

[51] Ghazel, M., Toguyéni, A., Bigand, M.: An UML approach for the metamodelling of automated production systems for monitoring purpose. Computers in Industry **55**(3), 283–299 (2004)

[52] Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer Verlag (2004)

[53] Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. Web Semantics: Science, Services and Agents on the World Wide Web **6**(4), 309–322 (2008)

[54] Greenacre, P., Gross, R., Heptonstall, P.: Great expectations: The cost of offshore wind in UK waters. UK Energy Research Centre, London (2010)

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

[56] Gruber, T.R., et al.: A translation approach to portable ontology specifications. Knowledge Acquisition **5**(2), 199–220 (1993)

[57] Haase, P., Lewen, H., Studer, R., Tran, D.T., Erdmann, M., d'Aquin, M., Motta, E.: The NeOn ontology engineering toolkit. WWW (2008)

[58] Hachem, N.I., Qiu, K., Serrao, N., Gennert, M.A.: GaeaPN: A Petri Net model for the management of data and metadata derivations in scientific experiments. Worcester Polytechnic Institute Computer Science Department Technical Report WPI-CS-TR-94 **1** (1994)

[59] Hall, D.L., Llinas, J.: An introduction to multisensor data fusion. Proceedings of the IEEE **85**(1), 6–23 (1997)

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

[61] Hameed, Z., Vatn, J., Heggset, J.: Challenges in the reliability and maintainability data collection for offshore wind turbines. Renewable Energy **36**(8), 2154–2165 (2011)

[62] Harel, D., Rumpe, B.: Meaningful modeling: what's the semantics of "semantics"? Computer **37**(10), 64–72 (2004)

[63] Horrocks, I., Patel-Schneider, P.F., McGuinness, D.L., Welty, C.A.: OWL: a description logic based ontology language for the semantic web. In: F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P.F. Patel-Schneider (eds.) The Description Logic Handbook: Theory, Implementation, and Applications (2nd Edition), chap. 14. Cambridge University Press (2007)

[64] Horrocks, I., Patel-Schneider, P.F., Van Harmelen, F.: 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 (2003)

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

[66] Huang, K.T., Lee, Y.W., Wang, R.Y.: Quality Information and Knowledge. Prentice Hall PTR (1998)

[67] Hull, R.: Managing semantic heterogeneity in databases: a theoretical prospective. In: Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, pp. 51–61. ACM (1997)

[68] Iannone, L., Rector, A.L.: Calculations in OWL. In: OWLED (2008)

[69] Informatica Corporation: Metadata Management for Holistic Data Governance (2013). White Paper

[70] Inmon, W.H.: Building the data warehouse. Wiley (2005)

[71] Jacq, N., et al.: Data integration in ehealth: a domain/disease specific roadmap. In: From Genes to Personalized Healthcare: Grid Solutions for the Life Sciences, Proceedings of HealthGrid, pp. 144–150 (2007)

[72] Janowicz, K., Compton, M.: The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology. In: 3rd International Workshop on Semantic Sensor Networks (2010)

[73] Johnsson, A., Højholt, L.E.: Use of IEC 61400-25 to secure access to key O&M data. In: European Offshore Wind conference 2007. Berlin, Germany (2007)

[74] Jones, D., Bench-Capon, T., Visser, P.: Methodologies for ontology development. In: Proc. IT&KNOWS Conference of the 15th IFIP World Computer Congress, pp. 20–35. Citeseer (1998)

[75] Khan, S., Qadir, M.: A generic framework for evaluation of ontology to relational database transformation process. In: Information and Emerging Technologies (ICIET), 2010 International Conference on, pp. 1–5. IEEE (2010)

[76] Køhrsen, T.: Deployment of IEC 61400-25-4 - mapping to web services on an industrial PLC platform. Master's thesis, Technical University of Denmark (2008)

[77] Kollia, I., Glimm, B., Horrocks, I.: SPARQL query answering over OWL ontologies. In: The Semantic Web: Research and Applications, pp. 382–396. Springer (2011)

[78] Kusiak, A., Li, W., Song, Z.: Dynamic control of wind turbines. Renewable Energy **35**(2), 456–463 (2010)

[79] Kusiak, A., Zheng, H., Song, Z.: Power optimization of wind turbines with data mining and evolutionary computation. Renewable Energy **35**(3), 695–702 (2010)

[80] Lackner, M.A., Rotea, M.A.: Structural control of floating wind turbines. Mechatronics **21**(4), 704 – 719 (2011)

[81] Lassila, O., McGuinness, D.: The role of frame-based representation on the semantic web. Linköping Electronic Articles in Computer and Information Science **6**(5), 2001 (2001)

[82] Le-Phuoc, D., Hauswirth, M.: Linked open data in sensor data mashups. In: Proc. Semantic Sensor Networks. CEUR-WS (2009)

[83] Le-Phuoc, D., Nguyen-Mau, H.Q., Parreira, J.X., Hauswirth, M.: A middleware framework for scalable management of linked streams. Web Semantics: Science, Services and Agents on the World Wide Web **16**, 42–51 (2012)

[84] Lee, Y.W., Strong, D.M., Kahn, B.K., Wang, R.Y.: AIMQ: A methodology for information quality assessment. Information and Management **40**(2), 133–146 (2002)

[85] Lenzerini, M.: Data integration: A theoretical perspective. In: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 233–246. ACM (2002)

[86] Lenzerini, M., Milano, D., Poggi, A.: Ontology representation & reasoning. Sapienza University of Rome, Roma, Italy, Tech. Rep. NoE InterOp (IST-508011) (2004)

[87] Liu, H., Chen, Z.: Aggregated modelling for wind farms for power system transient stability studies. In: Power and Energy Engineering Conference (APPEEC), 2012 Asia-Pacific, pp. 1–6. IEEE (2012)

[88] Maréchaux, J.L.: Combining Service-Oriented Architecture and Event-Driven Architecture using an Enterprise Service Bus. IBM Developer Works pp. 1269–1275 (2006)

[89] Michelson, B.M.: Event-driven architecture overview. Patricia Seybold Group **2** (2006)

[90] Miguelánez, E., Lane, D.: Predictive diagnosis for offshore wind turbines using holistic condition monitoring. In: OCEANS 2010, pp. 1–7. IEEE (2010)

[91] Mihaila, G.A., Raschid, L., Vidal, M.E.: Using quality of data metadata for source selection and ranking. In: Proceedings of the Third International Workshop on the Web and Databases, WebDB, pp. 93–98 (2000)

[92] Morales, J.M., Pinson, P., Madsen, H.: A transmission-cost-based model to estimate the amount of market-integrable wind resources. Power Systems, IEEE Transactions on **27**(2), 1060–1069 (2012)

[93] Motik, B., Patel-Schneider, P., Parsia, B., Bock, C., Fokoue, A., Haase, P., Hoekstra, R., Horrocks, I., Ruttenberg, A., Sattler, U., et al.: OWL 2 web ontology language: Structural specification and functional-style syntax. W3C recommendation **27**, 17 (2009)

[94] Mu, L., Kuang, L., Prinz, A.: Maritime data integration using standard ISO 15926. In: Proceedings of the 20th International Offshore and Polar Engineering Conference (2010)

[95] Muljadi, E., Pierce, K., Migliore, P.: Control strategy for variable-speed, stall-regulated wind turbines. In: American Control Conference, 1998. Proceedings of the 1998, vol. 3, pp. 1710–1714. IEEE (1998)

[96] Muyeen, S., Tamura, J., Murata, T.: Stability augmentation of a grid-connected wind farm. Springer (2009)

[97] Neuhaus, H., Compton, M.: The semantic sensor network ontology. In: AGILE Workshop on Challenges in Geospatial Data Harmonisation, Hannover, Germany, pp. 1–33 (2009)

[98] NORCOWE: Norwegian Centre for Offshore Wind Energy (NORCOWE) (2008). URL `http://www.norcowe.no/doc/PDF%20files/CMR-08-F90801-PF-01-Rev11-NORCOWE_Project_Description.pdf`. Accessed December 18 2013

[99] O'Byrne, D., Brennan, R., O'Sullivan, D.: Implementing the draft W3C semantic sensor network ontology. In: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on, pp. 196–201 (2010)

[100] O'Connor, M., Das, A.: SQWRL: a query language for OWL. In: Proc. of 6th OWL: Experiences and Directions Workshop (OWLED2009) (2009)

[101] OLF The Norwegian Oil Industry Association: Reference architecture of IT systems for OLF's IO G2. Tech. rep. (2008)

[102] Olsen, A., Osdil, B., Poulsen, B., Pedersen, K., Johansen, K.: Prototype of generic server for wind power plants using IEC 61400-25 standard. International Journal of Distributed Energy Resources **3**(4), 273–290 (2007)

[103] Papadopoulos, P., Cipcigan, L.: Wind turbines' condition monitoring: an ontology model. In: Sustainable Power Generation and Supply, 2009. SUPERGEN'09. International Conference on, pp. 1–4. IEEE (2009)

[104] Patel-Schneider, P.F., Hayes, P., Horrocks, I., et al.: OWL web ontology language semantics and abstract syntax. W3C recommendation **10** (2004)

[105] Pérez, B., Mínguez, R., Guanche, R.: Offshore wind farm layout optimization using mathematical programming techniques. Renewable Energy **53**(0), 389 – 399 (2013)

[106] Pinkel, C., Binnig, C., Haase, P., Martin, C., Sengupta, K., Trame, J.: How to best find a partner? an evaluation of editing approaches to construct r2rml

mappings. In: The Semantic Web: Trends and Challenges, pp. 675–690. Springer (2014)

[107] Pinson, P., Madsen, H.: Adaptive modelling and forecasting of offshore wind power fluctuations with markov-switching autoregressive models. Journal of Forecasting **31**(4), 281–313 (2012)

[108] Poveda-Villalón, M., Suárez-Figueroa, M.C., Gómez-Pérez, A.: Validating ontologies with oops! In: Knowledge Engineering and Knowledge Management, pp. 267–281. Springer (2012)

[109] Rademakers, T., Dirksen, J.: Open-source ESBs in action. Manning (2008)

[110] Ramirez, R.G., Kulkarni, U.R., Moser, K.A.: Derived data for decision support systems. Decision Support Systems **17**(2), 119–140 (1996)

[111] Rasta, K., Nguyen, T.H., Prinz, A.: A framework for data quality handling in enterprise service bus. In: Innovative Computing Technology (INTECH), 2013 Third International Conference on, pp. 491–497 (2013)

[112] Redman, T.C.: The impact of poor data quality on the typical enterprise. Communications of the ACM **41**(2), 79–82 (1998)

[113] Richardson, L., Ruby, S.: RESTful web services. O'Reilly (2008)

[114] Righter, R.: Wind energy in America: A history. University of Oklahoma Press (1996)

[115] Rohjans, S., Uslar, M., Bleiker, R., González, J., Specht, M., Suding, T., Weidelt, T.: Survey of smart grid standardization studies and recommendations. In: Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on, pp. 583–588. IEEE (2010)

[116] Sarkar, P.K., Cybulski, J.: Aligning system requirements with stakeholder concerns: use of case studies and patterns to capture domain experience. In: Seventh Australian Workshop on Requirements Engineering: proceedings, pp. 67–82. Deakin University (2002)

[117] Schlechtingen, M., Ferreira Santos, I.: Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. Mechanical Systems and Signal Processing **25**(5), 1849–1875 (2011)

[118] Schwarz, K.: Monitoring and control of power systems and communication infrastructures based on IEC 61850 and IEC 61400-25. In: 2010 DistribuTech conference & Exposition (2010)

[119] Sequeda, J., Tirmizi, S., Corcho, O., Miranker, D.: Survey of directly mapping SQL databases to the semantic web. The Knowledge Engineering Review **26**(04), 445–486 (2011)

[120] Shardanand, U., Maes, P.: Social information filtering: algorithms for automating "word of mouth". In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 210–217. ACM Press/Addison-Wesley Publishing Co. (1995)

[121] Sheth, A., Anantharam, P., Henson, C.: Physical-cyber-social computing: An early 21st century approach. Intelligent Systems, IEEE **28**(1), 78–82 (2013)

[122] Sheth, A., Henson, C., Sahoo, S.S.: Semantic sensor web. Internet Computing, IEEE **12**(4), 78–83 (2008)

[123] Sirin, E., Parsia, B.: SPARQL-DL: SPARQL query for OWL-DL. In: OWLED, vol. 258 (2007)

[124] Sirin, E., Parsia, B., Grau, B., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. Web Semantics: Science, Services and Agents on the World Wide Web **5**(2), 51–53 (2007)

[125] Snyder, B., Kaiser, M.J.: Ecological and economic cost-benefit analysis of offshore wind energy. Renewable Energy **34**(6), 1567–1578 (2009)

[126] Sreenath, R.M., Singh, M.P.: Agent-based service selection. Web Semantics: Science, Services and Agents on the World Wide Web **1**(3), 261 – 279 (2004)

[127] Strong, D.M., Lee, Y.W., Wang, R.Y.: Data duality in context. Communications of the ACM **40**(5), 103–110 (1997)

[128] Stuckenschmidt, H., Van Harmelen, F.: Information Sharing on the Semantic Web. Springer (2005)

[129] Studer, R., Benjamins, V., Fensel, D.: Knowledge engineering: principles and methods. Data & Knowledge Engineering **25**(1-2), 161–197 (1998)

[130] Suárez-Figueroa, M.C., Gómez-Pérez, A., Fernández-López, M.: The neon methodology for ontology engineering. In: Ontology engineering in a networked world, pp. 9–34. Springer (2012)

[131] Sure, Y., Staab, S., Studer, R., et al.: On-to-knowledge methodology (OTKM). In: Handbook on Ontologies, International Handbooks on Information Systems. Citeseer (2003)

[132] Thorsen, K., Rong, C.: Data integration in oil and gas at Norwegian continental shelf. In: Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on, pp. 1597–1602. IEEE (2008)

[133] Thorsen, K.A.H., Rong, C.: Towards data integration from WITSML to ISO 15926. In: Ubiquitous Intelligence and Computing, pp. 626–635. Springer (2008)

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

[135] Tiusanen, R., Jannes, J., Reunanen, M., Liyanage, J.: RAMSI management - from single analyses to systematic approach. In: COMADEM 2011, pp. 1588–1596. COMADEM, Stavanger, Norway (2011)

[136] Ueno, K., Tatsubori, M.: Early capacity testing of an enterprise service bus. In: Web Services, 2006. ICWS'06. International Conference on, pp. 709–716. IEEE (2006)

[137] Villazón-Terrazas, B., Gómez-Pérez, A.: Reusing and re-engineering non-ontological resources for building ontologies. In: Ontology Engineering in a Networked World, pp. 107–145. Springer (2012)

[138] Villazón-Terrazas, B., Ramírez, J., Suárez-Figueroa, M.C., Gómez-Pérez, A.: A network of ontology networks for building e-employment advanced systems. Expert Systems with Applications **38**(11), 13,612–13,624 (2011)

[139] W3C OWL Working Group: OWL 2 web ontology language document overview. W3C Recommendation (2012)

[140] Wang, R.Y., Strong, D.M.: Beyond accuracy: What data quality means to data consumers. Journal of Management Information Systems **12**, 5–33 (1996)

[141] Warmer, J., Kleppe, A.: The object constraint language: getting your models ready for MDA. Addison-Wesley Longman Publishing Co., Inc. (2003)

[142] 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. Schlömer , C. von Stechow (eds)

[143] Wöhrer, A., Brezany, P., Tjoa, A.M.: Novel mediator architectures for grid information systems. Future Generation Computer Systems **21**(1), 107 – 114 (2005)

[144] Xu, H.: Critical success factors for accounting information systems data quality. Ph.D. thesis, University of Southern Queensland (2009)

[145] Yeh, I., Karp, P., Noy, N., Altman, R.: Knowledge acquisition, consistency checking and concurrency control for gene ontology (GO). Bioinformatics **19**(2), 241–248 (2003)

[146] Zaher, A., McArthur, S., Infield, D., Patel, Y.: Online wind turbine fault detection through automated SCADA data analysis. Wind Energy **12**(6), 574–593 (2009)

[147] Zervos, A., Kjaer, C.: Pure power: wind energy scenarios up to 2030. European Wind Energy Association (2008)

[148] Zhang, Y.: Dependable ESB routing in hybrid service execution environment. AISS: Advances in Information Sciences and Service Sciences **2**(1), 83–93 (2010)

[149] Zhu, H., Madnick, S.E., Lee, Y.W., Wang, R.Y.: Computing Handbook, chap. Data and Information Quality Research: Its Evolution and Future. Chapman and Hall/CRC (2014)

[150] Ziegler, P., Dittrich, K.R.: Three decades of data integration-all problems solved? In: IFIP congress topical sessions, pp. 3–12. Springer (2004)

[151] Zubcoff, J., Pardillo, J., Trujillo, J.: A UML profile for the conceptual modelling of data-mining with time-series in data warehouses. Information and Software Technology **51**(6), 977–992 (2009)

Part II

Appendix A

Paper A

Title	A Framework for Data Integration of Offshore Wind Farms
Authors	Trinh Hoang Nguyen[a], Andreas Prinz[a], Trond Friisø[b], Rolf Nossum[c], Ilya Tyapin[d]
Affiliation(s)	[a] Department of ICT, University of Agder, Grimstad, Norway
	[b] Orgio Solutions AS, Arendal, Norway
	[c] Department of Mathematical Sciences, University of Agder, Kristiansand, Norway
	[d] Department of Engineering, University of Agder, Grimstad, Norway
Journal	Renewable Energy
Copyright	©Elsevier

A Framework for Data Integration of Offshore Wind Farms

Trinh Hoang Nguyen[a,*], Andreas Prinz[a,*], Trond Friisø[b,**], Rolf Nossum[c,*], Ilya Tyapin[d,*]
[a] Department of ICT, University of Agder, Grimstad, Norway
[b] Orgio Solutions AS, Arendal, Norway
[c] Department of Mathematical Sciences, University of Agder, Kristiansand, Norway
[d] Department of Engineering, University of Agder, Grimstad, Norway
E-mails: *{trinh.h.nguyen, andreas.prinz, rolf.nossum, ilya.tyapin}@uia.no
** trond.friiso@origo-solutions.com

Operation and maintenance play an important role in maximizing the yield and minimizing the downtime of wind turbines, especially offshore wind farms where access can be difficult due to harsh weather conditions for long periods. It contributes up to 25-30% to the cost of energy generation. Improved operation and maintenance (O&M) practices are likely to reduce the cost of wind energy and increase safety. In order to optimize the O&M, the importance of data exchange and knowledge sharing within the offshore wind industry must be realized. With more data available, it is possible to make better decisions, and thereby improve the recovery rates and reduce the operational costs. This article describes the development of a framework for data integration to optimize remote operations of offshore wind farms.

Keywords: Offshore wind farms; Data integration; Remote operations; Operation and maintenance; Wind energy.

1 Introduction

The currently accelerating sea level rise, ocean acidification and ice cap melting have prompted the European Union (EU) and other industrialized regions to propose drastic reductions of greenhouse gas emissions. The EU's renewable energy policy aims for 34% of the EU's total electricity consumption coming from renewable energy sources in 2020 and 100% renewables by 2050. Wind energy alone could cover up to 50% of Europe's electricity by then. The offshore wind power production in the EU from 2005 to 2020 is expected to increase from 2 TWh to 140 TWh[12].

It is apparent that moving wind farms offshore brings huge benefits for wind energy production due to highly stable wind, use of large scale wind power plants (WPP), larger wind farms, and fewer societal concerns. It also reduces environmental impact as well as noise and visual disturbance to people [11]. However, the offshore wind industry is facing challenges with high installation and maintenance costs and potentially longer time out of operation at failures. According to [48], failure rates and downtime of WPP components provided by the Landwirtschaftskammer Schleswig-Holstein (LWK, Germany) and the Scientific Measurement and Evaluation Programme (WMEP, Germany) show that some components have low failure rates but very high downtime. For example, the gearbox failure rate is low compared to other components; however downtime is relatively high because the replacement of a failed gearbox requires the use of a crane, barge/ship, etc., and thus the cost associated with replacing it is significant. The amount of downtime caused by malfunctions depends on the repair work required as well as the availability of parts, and the personnel capacity of service teams. In the past, repairs to the generator, drive train, hub, gearbox, and blades have often caused standstill periods of several weeks [17]. The logistical complexity of maintenance and repair procedures increases greatly if they depend on a window of uninterrupted. In case of maintenance and repair procedures, there are two groups of activities which do and do not require a use of a weather window. The first group of activities may be interrupted during its execution in case of unacceptable weather conditions and may be recommenced when fair conditions apply. The second one may not be interrupted during its execution. In this case, the collaboration between the project managers, fault analysis (condition based monitoring) team, maintenance services, and weather forecast is necessary to maintain the turbine on time or make a repair as soon as possible to decrease the energy loss caused by the fault or delayed maintenance.

Operation and maintenance (O&M) costs of offshore wind farms contribute significantly (up to 25-30%) to the cost of energy generation [33, 47, 14]. Although a 25% decrease in O&M costs is expected to give less than 3% reduction in levelized costs [41], the relationship between improved O&M and optimized availability is still important. Improved O&M is also likely to reduce the hazard exposure of the employees, increase income, and support offshore activities more efficiently. One key improvement is the implementation of remote operations [27]. An onshore support system and their sub-systems are performing a multi-function for service technology and management development, such as fault analysis, optimized maintenance activities and visits, 24/7 alarm handling and analysis [10].

In order to minimize the production cost, some optimization opportunities might be applied:

- concerted control of the entire wind farm.
- use of a system-level design and analysis approach for individual turbines and entire wind power plants to optimize wind turbine technology, power plant installation and O&M procedures [49].
- control of wear such that risk/reliability based operation/maintenance can be scheduled for periods where the turbines are accessible.
- control of power production such that the economic outcome is maximized.

To obtain these kinds of optimizations, various kinds of data are needed, and different expertise has to be combined, e.g., turbine control, meteorology, energy trading, marine operations, condition determination and prediction. Availability and accessibility of information are needed for O&M planning and improving the quality of WPP components. Even though there is some wind farms' data available, it is still difficult to handle it. The need of a way of integrating and handling data from different sources with different formats via a unified system is essential. Moreover, an estimation from the Norwegian Oil and Gas Association (OLF) shows that the economic potential of implementing integrated operations is expected to be more than 250 billion Norwegian kroner in net present value [35].

This article focuses on proposal of a data integration framework for offshore wind farms in order to support the offshore wind activities more effectively and enable data exchange and knowledge sharing between the offshore wind partners. The rest of the article is organized as follows: section 2 introduces challenges in data integration of offshore wind farms. Section 3 presents a proposed framework as a solution to overcome the challenges, and section 4 introduces some preliminary results of implementing and development such as an offshore wind ontology model. The results are discussed in Sect. 5. Finally, section 6 contains a summary and conclusions.

2 Data integration within the offshore wind industry

This section discusses challenges that data integration of the offshore wind industry is facing today. The section starts with introduction to offshore wind daily operations and information exchange categories. After that, challenges are discussed.

This work focuses on the operational phase of a wind farm. Installation, construction, and decommissioning are not considered.

2.1 Offshore wind operations

Based on the Smart Grid Interoperability Panel promoted by the National Institute of Standards and Technology [34], the conceptual model for the offshore wind communication is formulated as shown in Fig. A.1. In this conceptual model, each domain is a high-level grouping of organizations, individuals, and systems of the offshore wind industry. Communication between stakeholders in the same domain may have similar characteristics and requirements. The communication flows are bidirectional.

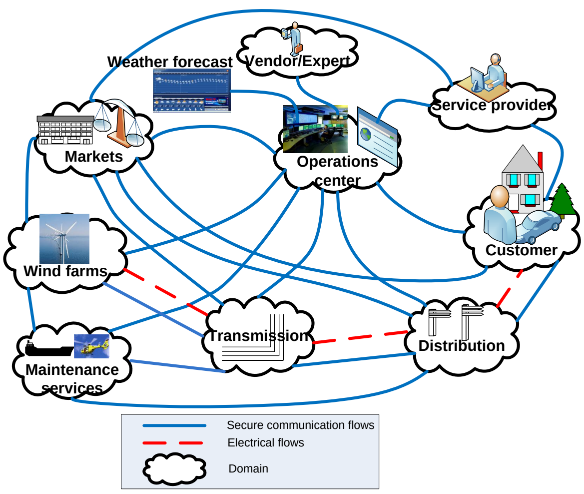


Figure A.1: A conceptual model for offshore wind communications

In each wind farm, there has to be communication between wind turbines, between operations center and maintenance personnel, and communication to other wind farms with the purpose of production optimization. According to wind direction and speed, each turbine can have a different configuration, for example blade pitch angles which are changed by an individual pitch control system, so that the wind can generate energy most cost effectively.

A wind turbine has many components. Hence, it is difficult to monitor the performance of each particular component and possibly detect the faults. For example, the major drawback of a hydraulic pitch control system is the presence of external and internal leakage, and the latter is difficult to detect [6]. According to the severity of a leakage fault, the performance of pitch control systems is degraded and eventually a too high severity fault in leakage will lead the wind turbine to failure. To reduce the cost of maintenance and to prevent such systems from failure, fault detection for leakage should be considered. However, due to non-linearity in hydraulic systems and the large uncertainties in their parameters, fault detection is difficult to implement on site using real-time techniques. Therefore, the important issues are proper communications between the operations center and the wind turbine, off-site data collection and analysis. In the operations center, experts with different backgrounds are able to perform a proper analysis based on measurements received from offshore wind farms to arrange unscheduled works and retune the controllers if necessary. The center handles various operations such as data analysis, data visualization, and information collection from weather forecast. Based on the collected information, the center controls the velocity of wind turbines and optimizes the wind parks in order to maximize the yield over its lifetime. In addition, the use of the operational center will allow reliability engineers to better plan and organize maintenance actions related to offshore wind turbines. If there is a case where human participation is needed, the center sends a request to experts to get advice and suggestions from them. After that, operators or decision-makers at the center will take a final decision to on-board maintenance services in order to have correct and timely operations. The information from the operations center is also necessary for the energy market (grid/consumers), and vice-versa.

Let us take a look at the data exchange and knowledge sharing from the economic perspective where many stakeholders want to have information from their partners. Table A.1 shows the different categories of stakeholders and their concerns in the offshore wind industry. Note that a stakeholder can be involved in several domains. By analyzing the functions and roles of stakeholders in the offshore wind industry, it is easier to define data exchange format and what kind of data to exchange.

In general, data must be collected in real-time at sufficiently high sampling rates and stored in a database to enable both online and offline evaluations. Online evaluation and detection of leakage, for example, would require computational processing speeds of model-based algorithms at $1\ ms$ sampling rate or higher. From a practical point of view, the information technology infrastructure is probably one of the most significant limiting factors to the introduction of advanced level algorithms. Whereas communication between wind farms and the operations center probably will be transferred using optical fibers, communication between wind turbines or wind farms can be exchanged by broadband wireless communication.

Table A.1: Domains, stakeholders, and their functions in the offshore wind industry

Stakeholder role	**Domain**	**Stakeholder**	**Function**
Authorities	Wind farms	Government	Collaboration with owner/operators. Supervision and control of license obligations, environmental impact etc.
Service provider	Engineering companies distribution, transmission maintenance service	Supporter on boat/ship	Perform maintenance tasks and marine operations.
		Installation companies	Provide equipments, construction drawings and other documentation.
		Electricity carriers	Carry bulk electricity over long distances. Supply of electricity to end-users.
Vendors/experts	Engineering companies	Equipment vendors	Support equipment, such as turbines, blades. Provide detailed condition monitoring and analysis of specific equipment.
		Domain experts	Give technical support, advice for making urgent decision.
License partners	Wind farms	Research institutions, companies	Cooperation in larger investment and strategic decisions.
Owner/Operator	Wind farms, operations center	Decision-makers, operators	Monitor and control the wind farms' operations.
	distribution, transmission	Electricity producer	Perform overall wind farm control, production planning maintenance decisions, technical and operational analysis, coordination and support of maintenance activities.
Power market	Energy market, customer,	Operators and participants in electricity markets, electricity companies	Impact on amount of production energy. Sell electricity to customers.
	engineering companies	Residential, commercial, industrial, end-users	Consume electricity.
Weather forecast	Weather forecast	Weather forecaster, meteorologist	Storm tracks, short- and long-term predictions of wind and other weather conditions, like wave height, forecast on climate change.

2.2 Categories of information exchange

Information is the content of the communication that takes place within the framework of monitoring and control of offshore wind turbines [24]. The following information is essential in order to achieve effective monitoring and control.

- *Equipment information*: It contains information about manufacturers, design characteristics (default values of the equipment, allowable values for configuration), operating modes, etc.

- *Discrete state information:* Discrete information concerning the current condition or behavior of a component or a system. It might consist of status (condition of a component or system), alarm and event information.

- *Analog state information*: Continuous information concerning the current condition or behavior of a component or a system. This type of information consists of value of process quantity, measured value, which has been processed and measured, for example, value of a three phase electric power quantity.

- *Control information*: Control information is required for the control of wind power plants, such as access profiles, set points, parameters and commands. This information will first be communicated to wind power plants by certain operators. Wind power plants store control information and provide it for further communication to sub-processes.

- *Historical information*: It might be possible to track the operational trends in logs and reports. Historical information is divided into three categories:

  – Log is a chronological list of events for a specific period of time.

  – Transient log is a short-term chronological list of events and data with high resolution.

  – Report is a periodical notification comprising the information that represent the state and data requested in the report control block.

- *Descriptive information (meta-information)*: It gives the type and the accuracy of the information, as well as the time and the data description. For instance, total time duration of a specific state, properties of the observed data (max, min, average, etc.).

2.3 Challenges in offshore wind data integration

With more data available, it is possible to make better decisions, and thereby improve the recovery rates and reduce the operational costs. Design of new components can get benefits from available data of used components. Capturing failure data can help to analyze reliability of wind turbine using statistical methods [19]. However, the offshore wind data integration is facing some challenges.

Typically, any wind power plant component, which needs to exchange information with other components and operations center, is equipped with a so-called intelligent electronic device (IED), which can send data to external receivers and receive data from external senders [24]. A key point is the capability of the grid to collect and convey information from IEDs on the network to the information systems of the different actors (mainly generating system operator, market system operator, operations center) [4]. With the replacement of electromagnetic devices by IED, the automation of distribution systems has made a significant step forward. IEDs are capable of protection, local monitoring and control, etc., and therefore, communication between IEDs and central controller plays an important role for maintaining, controlling and monitoring a distributed system. However, most components of WPP are produced by different vendors or companies. Each component has its own software and perhaps its own database. As a result, a software environment of a WPP consists of multiple applications having incompatible interfaces and data formats and not being able to communicate with each other.

Agreement on data exchange is time-consuming. Unfortunately, it happens only at the end of the development when the partners encounter integration problems with other partners, for instance, when some terminologies are interpreted differently by different partners. Besides, many actors are reluctant to share data about their equipment or to let third parties collect such data. Therefore, better ways to make data available and accessible would be desirable.

Another problem faced by the data integration is that the traditional point-to-point (P2P) integration makes the collaboration even harder. Fig. A.2 demonstrates the P2P integration of different applications. The big disadvantage of the P2P integration topology is that each application has to establish a separate connection channel with other applications. Once an application changes its data exchange structure, all related applications will be affected. With today's technologies, P2P integration between applications is no longer recommended due to the difficulty in reconstructing the system and the costs of implementing changes. Data collection and usage

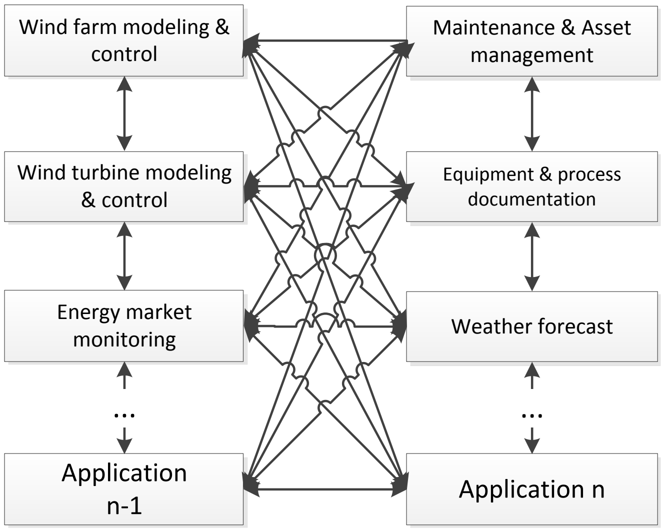


Figure A.2: Point to point integration

is the most challenging problem faced by the offshore wind industry today. Some questions are listed below:

- How to agree on concepts, their properties, and relations in the offshore wind domain?
- How can wind farm operators manage the fact that data comes from different sources with different formats?
- Data sources are considered as autonomous, distributed and heterogeneous systems. How to manage, integrate, and unify them systematically?
- How to solve semantic inconsistency which has become a problem for the explicit information or knowledge sharing among users or applications?

3 A proposed data integration framework

Based on the proposed reference architecture for remote operations of offshore wind farms [31] a data integration framework is proposed as shown in Fig. A.3. *Part*(1) illustrates all applications from offshore wind partners, such as wind turbine control and modeling system, weather forecast monitoring system, engineering systems, asset management, wind farm modeling and control. *Part*(3) shows two upstream domains, namely production management and operation and maintenance where

A

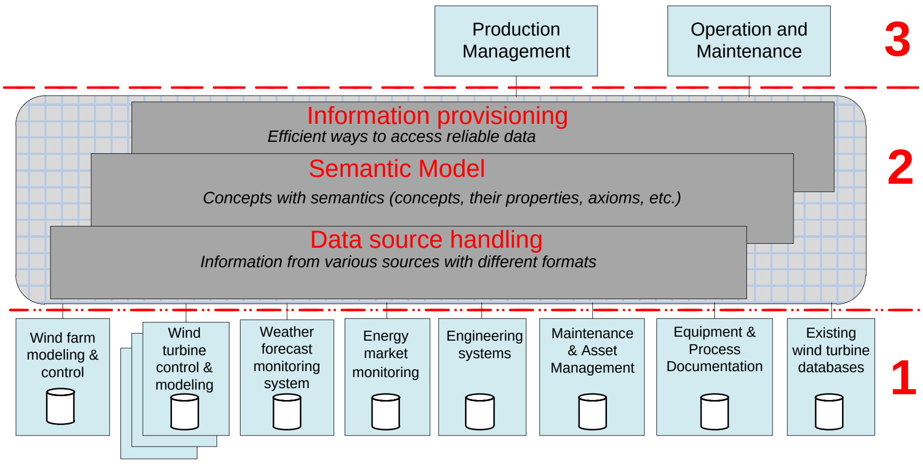


Figure A.3: An overview of the proposed data integration framework

new solutions are being developed. *Part*(1) and *Part*(3) are connected by *Part*(2) which presents three patterns of the proposed framework (*information provisioning*, *semantic model*, and *data source handling*).

The *semantic model* covers the key concepts in the offshore wind domain and their semantic relationships. It is considered a core for data integration. An instance of the model, for example a virtual database (DB), can be developed. Input data for the database is provided by the *data source handling* which collects data from offshore wind partners. The output data from the database is provided to the other partners through the *information provisioning* for different purposes such as visualization, documentation and analysis. The acquisition of relevant data underpins the lifecycle of offshore wind farms, through the project phases of feasibility analysis, development, engineering, construction, operation and maintenance, decommissioning and post decommissioning.

3.1 The semantic model

A semantic model shares the common understanding of domain concepts. Additionally, it documents the exchange protocol between offshore wind partners on data exchange, in particular, what data exchange format and what kinds of data to exchange. Fig. A.4 shows the proposed semantic model.

A typical problem of data exchange is a misunderstanding between sender and receiver. Approved standards are recommended to use in order to make the data exchange unambiguous. There are some relevant standards developed by the Inter-

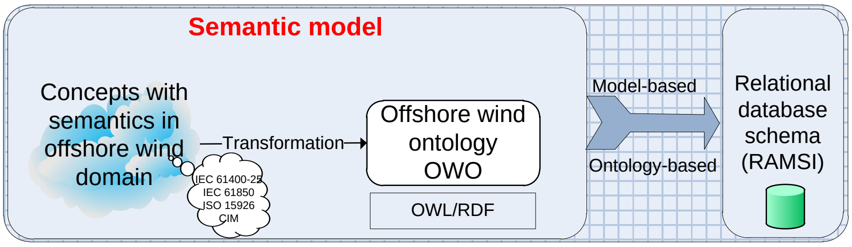


Figure A.4: The proposed semantic model

national Electrotechnical Commission (IEC), for example the Common Information Model (CIM) [25], IEC 61850 [21], IEC 61400-25 [24]. Unfortunately, those standards have been developed by different working groups and therefore lack some harmonization [45]. Additionally, the semantic techniques imposed by the CIM are not properly used [46]. Resolving semantic heterogeneity not only gives users a unified access to distributed data but also facilitates monitoring processes, hence the performance of a WPP might be improved.

Organizational units and information technology systems rarely last more than a few years. Only some parts of the data from these systems are used for decades and become main assets. It is obvious that terminologies used in the business domains are the most stable elements. The idea of creating an *Offshore Wind Ontology* (OWO) from the terminologies in order to share, reuse knowledge, and reason about behaviors across domains and tasks, is important. In general, an ontology is needed to make an abstract model of some phenomenon by identifying the relevant concepts of that phenomenon [40]. It facilitates integration of processes within and across business domains, creation of autonomous solutions, and storage of data over time. It is also a key instrument in developing the semantic web which is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation [3]. In order to develop the OWO, domain knowledge and concepts of the offshore wind industry are necessary. Thus, IEC 61400-25, IEC 61850, and CIM could be good reference standards for the OWO development. In addition, the International Organization for Standardization (ISO) 15926 standard [23] provides an ontology for oil & gas. Building an ontology based on ISO 15926 not only brings benefits for the offshore wind ontology development but also makes it easier for the oil & gas industry to enter the wind energy business.

Most of the properties of the OWO can be adequately represented in the Resource Description Framework (RDF)/RDFS (RDF schema) [8]. However, a number of desirable features are missing in RDF, such as local scope of properties, localized range and domain constraints, boolean combinations of classes, and cardinality restrictions. The Web Ontology Language (OWL) is an extension of RDF schema, in the sense that OWL uses the RDF meaning of classes and properties [8, 20, 1]. In this work, OWL is used to represent the proposed ontology.

3.2 The data source handling

As mentioned in Sect. 2, input data for an instance of the *semantic model* may come from different applications of offshore wind partners. The question is how to get the data from various sources into a single operational model? How to establish collaboration and interoperability between services, applications and parties? The *data source handling* architecture, shown in Fig. A.5, is used to answer these questions.

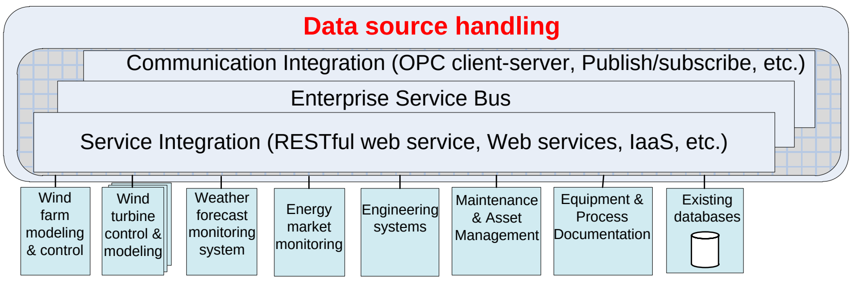


Figure A.5: Data source handling architecture

In order to assist a collaboration between different applications, the *communication integration* provides for various communication approaches, for instance publish/-subscribe allowing to access information through message exchange. Another example could be real-time data exchange between PC-based clients using Microsoft operating systems; in this case the Object Linking Embedded (OLE) for process control (OPC) protocol is essential. The OPC server is a software program that converts the hardware communication protocol used by a programmable logic controller into the OPC protocol, and the OPC client software is a set of programs that connects to the hardware, for example, a human machine interface. The OPC client uses the OPC server to get data from or send commands to the hardware. Web services are used to support interoperable machine-to-machine interaction over a network. Using Representational State Transfer (RESTful) web services and information as a service is a starting point. REST is an architecture style which was introduced by Roy Fielding in 2000. REST provides services over the Internet through the Web browser by using the four CRUD (create, read, update, delete) operations associated with four methods of the Hyper Text Transfer Protocol (HTTP):

GET, POST, PUT, DELETE [13]. The idea of REST consists in embracing a stateless (e.g., interaction can survive a restart of the server) client server architecture in which the web services are viewed as resources and can be identified by their Uniform Resource Identifiers (URI). Besides, REST supports cacheable (a server needs to respond to a request only for the first time and the response is cached in a proxy server and used for similar requests to the same resource without addressing the server) that reduces the load for servers and hence, improves scalability and performance of client server interactions.

Adjustments in data messages between applications can be handled via *Enterprise Service Bus* (ESB). An ESB is a software architecture for middleware. The idea of the ESB concept is that every application needs to be connected to a bus and then all applications can share information by producing or consuming information on the bus. It supports integration of services and applications for complex systems. Moreover, ESB can provide required mediation to expose REST-based or SOAP-based (Simple Object Access Protocol) services for web clients, process services for business processes orchestration and automation, and information services in order to manage diverse data and content in a unified manner.

3.3 The information provisioning

The *information provisioning* part provides efficient ways to access the reliable data, mechanisms to reuse existing applications when introducing new ones, and ensures that new applications are also reusable. For example, a condition monitoring application must be able to integrate with existing vibration monitoring services. Fig. A.6 shows three layers (*business processes*, *services*, and *presentation*) of this part.

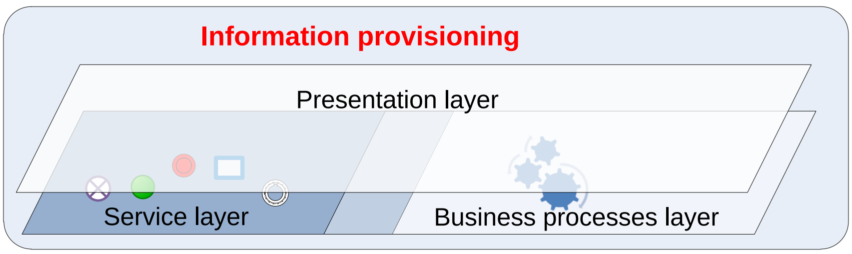


Figure A.6: Information provisioning

In order to enable reusability of a service, it is recommended to build a corresponding service component for the service. The *service layer* contains software components, each of which provides the implementation or realization for a service, or operation on a service. Service components reflect the definition of the service they represent, both in its functionality and its quality of service. The *service layer* provides consumers with sufficient detail to invoke the business functions exposed by a provider of the service. The new consuming applications and solutions will focus on their own functional capabilities, and not on the detail of the external component architecture. All business workflows are handled by the *business process layer*. The *presentation layer* provides user-friendly human machine interfaces to end-users. The *information as a service* capability will be beneficial to enable more flexible cross-facility information access. It enables loose coupling to data stores and data model [9]. It also enables business processes and users to work with up to date data in critical applications.

4 Development and implementation

This section presents some preliminary results of the framework development. An offshore wind ontology is considered as the core of the semantic model.

4.1 The information model

An information model represents the knowledge concerning specific domain communication. In particular, the purpose of creating an offshore wind information model is to facilitate the process of agreement on data exchange as well as collaborations among offshore wind partners. The information model will provide a common basis for understanding the general behavior of offshore wind communications. It is also the foundation of the offshore wind ontology. The information model is a part of the work which has been presented in [32].

4.1.1 The IEC 61400-25 standard

The IEC 61850 standard series is essentially a standard for power substation automation. The IEC 61400-25 standard is an adaptation of the IEC 61850 standard series, with special concern for controlling and monitoring WPPs. Therefore, the IEC 61400-25 standard series is not merely a replica of IEC 61850, but reuses the terms and definitions that will apply to all substations. Furthermore, the IEC 61400-25 standard series extends the IEC 61850 with unique information models which only apply to WPPs. Such unique features include rotor speed, turbine or other

Table A.2: IEC 61400-25: Wind turbines - Communications for monitoring and control of WPPs

Part	Title	Description
1	Overall description of principles and models	Introductory orientation, overview of crucial requirements and basic principles, and a modeling guide.
2	Information model	Specify the compatible logical nodes names, and data names for communication between WPP components.
3	Information exchange model	Define services of the model of the information exchange of intelligent electronic devices in WPPs.
4	Mapping to communication profiles	Mappings for Web services, MMS, OPC XML DA, IEC 60870-5-104 and DNP3.
5	Conformance testing	Define methods and abstract test cases for conformance testing of devices used in WPPs.
6	LN classes and data classes for condition monitoring	Defines additional information models for use in condition monitoring system.

vendor specific components. Table A.2 provides an overview of the IEC 61400-25 series.

The IEC 61400-25 standard represents a consensus on core information technology for the future transition of the electric distribution grid towards a smart grid [38]. The IEC 61400-25 defines information models and information exchange models for monitoring and control of WPPs. The modeling approach of IEC 61400-25-2 and IEC 61400-25-3 uses abstract definitions of classes and services such that the specifications are independent of specific communication protocol stacks, implementations, and operating systems [39]. Successful implementation of the IEC 61400-25 standard has been reported in [36, 29].

According to the IEC 61400-25 standard, the highest level is called the logical device which stands for a WPP, which is decomposed into logical nodes (LN). Fig. A.7 shows WPP information broken down into LNs. The structure of all LNs is specified in [24] and [21]. WROT stands for wind turbine rotor information, and WTUR stands for wind turbine general information. More details are available in [24].

All the LNs used in the WPP modeling inherit their structure from the abstract LN class defined in the IEC 61850-7-2. A LN consists of a collection of related data, called data classes (DC). Each data class inherits a collection of properties, as

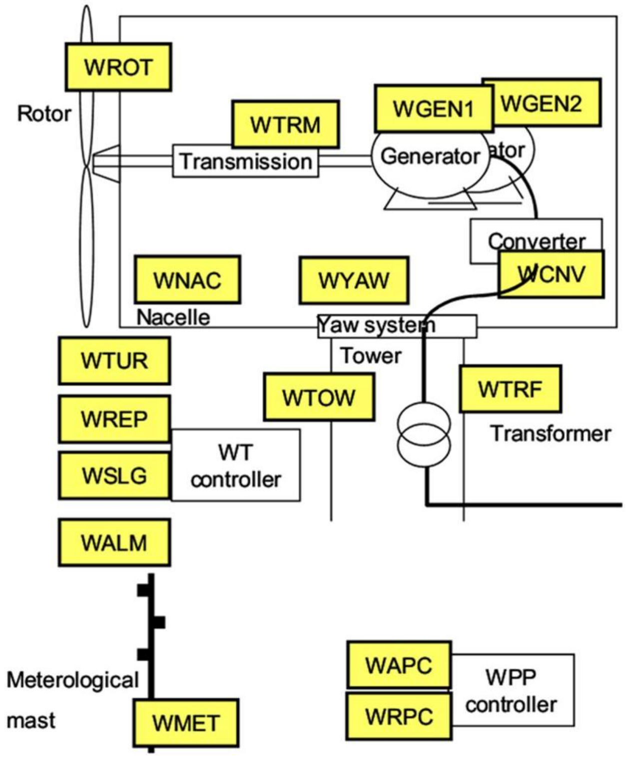


Figure A.7: Wind turbine logical nodes

defined by a so-called common data class (CDC) to which it is assigned. A CDC consists of a collection of data records.

4.1.2 An extension of the IEC 61400-25

One of the differences between onshore and offshore wind turbines is the design of the foundation. In case of an onshore wind turbine, the foundation is fixed and the vendor does not need to spend much effort on it after installation. For an offshore wind turbine, the four most common fixed foundations are monopile, gravity base, tripod, and jacket [42]. But for a deep water offshore wind turbine, it is hard to make the foundation fixed, therefore it is preferably made floating [43]. The technology of floating wind turbines has only recently been developed. In order to support the weight of the turbine and to constrain pitch, roll and heave motions within acceptable limits, a floating structure must provide enough buoyancy. Depending on the topology, wave, sea ice, and seabed conditions, floating platform configurations may vary. The most common platform types use ballast, mooring lines, and buoyancy for stability [5]. The stability of the foundation is of vital importance for floating wind turbines, hence information related to the foundation is critical for analysis, monitoring and control of the wind turbine as well as for further development and improvement. In floating platforms it is conceivable that control is used to limit the response of the entire turbine or platform system to stochastic wave loading [24]. For example, pitch motion can easily be limited by an intelligent collective pitch control strategy and through monitoring in the operations center decision makers can interfere in case of the failure of a built-in intelligent controller.

Depending on the type of floating platform, the foundation information will vary. For instance, for a floating wind turbine that achieves stability through the use of mooring line tension, information related to mooring lines needs to be collected. A new logical node for wind turbines is proposed, namely one for wind turbine foundations (WFOU). The name of the new node is given as WFOU according to the name space concept defined in [21]. As an extension of Fig. A.7, Fig. A.8 depicts the new wind turbine logical node WFOU. The node contains basic attributes for floating offshore wind turbines. The list of attributes is shown in Table A.3. “M” indicates mandatory and “O” indicates optional.

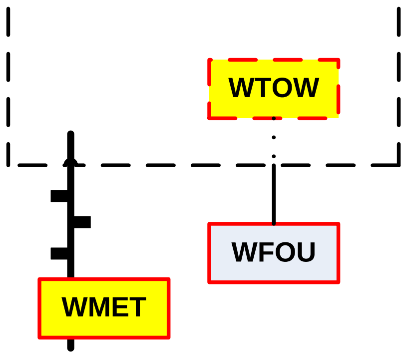


Figure A.8: The proposed wind turbine logical node, WFOU

Table A.3: WFOU class

WFOU class			
Attribute name	**Attribute type**	**Explanation**	**M**
Pt	MV	pitch	O
Rl	MV	roll	O
Yw	MV	yaw	O
Hv	MV	heave	O
Sg	MV	surge	O
Sy	MV	sway	O

In addition, the WPP meteorological (WMET) class is also extended by adding relevant attributes to offshore conditions based on the IEC 61400-3 [26]. Some attributes are shown in Table A.4.

A

Table A.4: WMET class extension

WMET class extension			
Attribute name	**Attribute type**	**Explanation**	**M**
Inherit all data from WMET class provided in the IEC 61400-25-2			
WavAmp	MV	wave amplitude	O
WavHz	MV	wave frequency	O
WavLgt	MV	wave length	O
WavDir	MV	wave direction	O
SeaCurrVel	MV	sea current velocity	O
SeaCurrDir	MV	sea current direction	O
WatLev	MV	water level	O
WatDens	MV	water density	O
WatTmp	MV	water temperature	O
AirDns	MV	air density	O
SolarRadInt	MV	solar radiation intensity	O

MV - measured value is a data class which contains measured attributes such as measured result, a quality attribute that contains information on the quality of the information from the server and timestamps. For more details, see [22].

Even though RDF and OWL schemas provide a clearer semantics for handling changes than do XML (eXtensible Markup Language) schemas, XML schemas still a good candidate for building information models for data exchange than other formats. Keeping the information model in XML schemas gives the following benefits:

- *Flexibility*: It is easy to transform an XML schema to different formats, such as OWL, RDF or SQL or DTD.
- *Understandability*: It takes less time to understand an XML schema than OWL or other formats.
- *Usability*: From XML schemas, Java or .Net classes can be generated using existing tools, such as JAXB or XSD.

Fig. A.9 shows partly the developed information model which is based on the IEC 61400-25, IEC 61850-7 and the proposed extensions for the IEC 61400-25 [1]. In this figure, the dotted line means that a logical node or an attribute is optional.

At the moment, the information model is presented only in XML format. It contains 18 classes and more than 200 attributes.

[1] Send an email to the corresponding author for the complete schema.

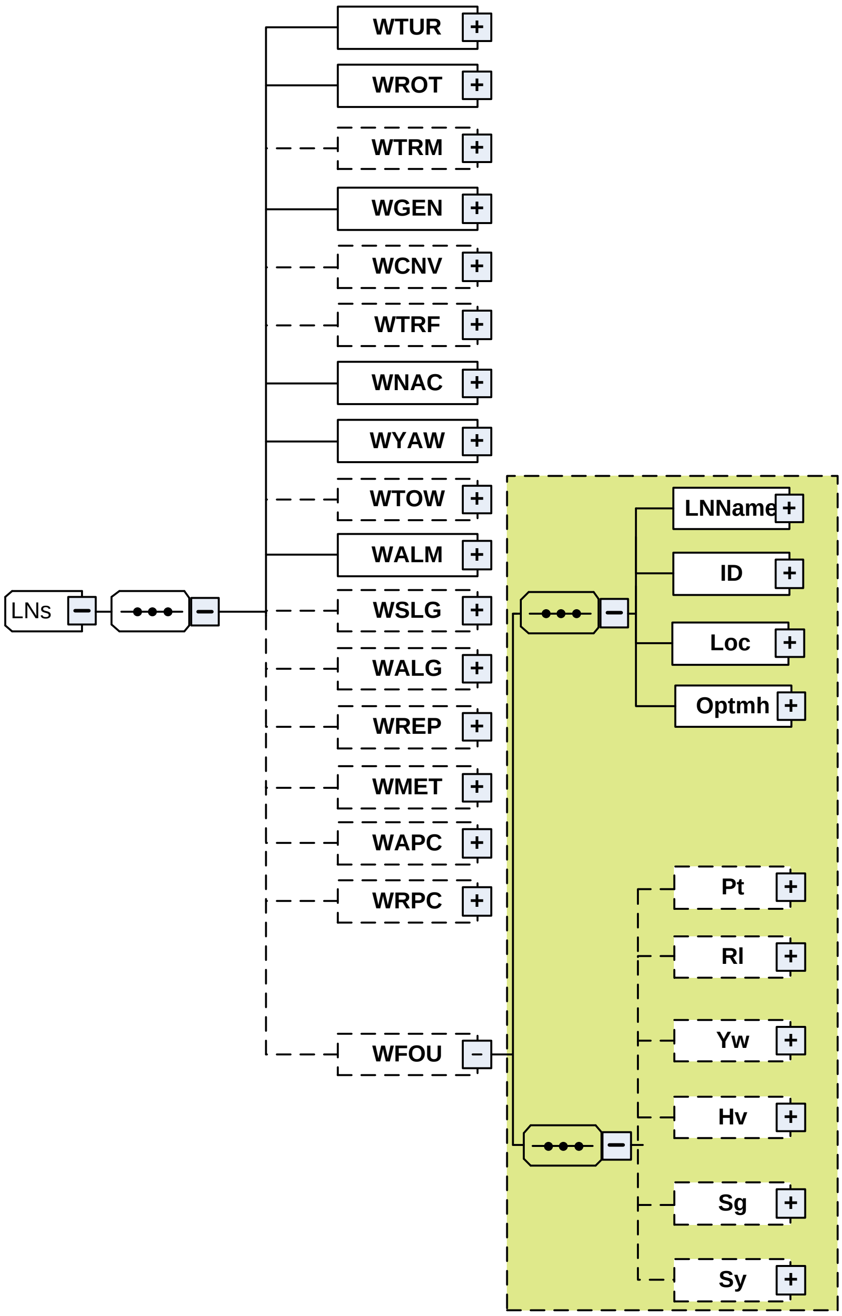


Figure A.9: XML schema of the information model (partly expanded)

4.2 The offshore wind ontology

In order to create the OWO, three elements are needed: strategy, methodology, and tools. A strategy for developing the OWO is shown in Fig. A.10. The strategy has been previously described in authors' work [30].

- Step 1: Select a WPP component, for example WPP rotor.

A

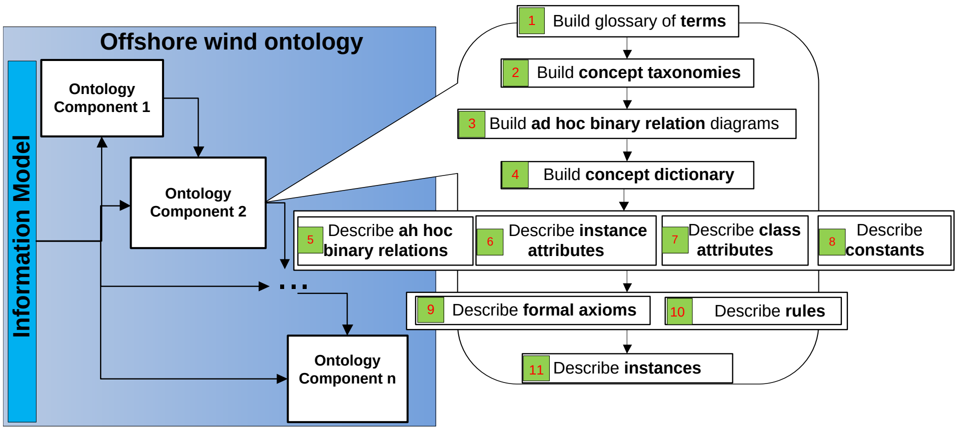


Figure A.10: Ontology development

- Step 2: Use the developed information model (based on the IEC 61400-25 and other standards) to extract concepts, attributes, and their relations to the selected WPP component.
- Step 3: Apply the methodology "METHONTOLOGY" [16] which contains 11 tasks to describe all relations in the ontology component.
- Step 4: Use Protégé 4.1 [2] to build the ontology and check its consistency.
- Step 5: Go back to step 1 and select another WPP component. Ontology component 1 is developed first and ontology component 2 is based on the first ontology. Ontology component n is built based on ontology component $n-1$.

An example of ontology component for WT rotor is shown in Fig. A.11. The WROT component consists of information regarding rotor, blades, pitch angle for blades, rotor speed, etc. In Fig. A.11a, the information structure of WROT based

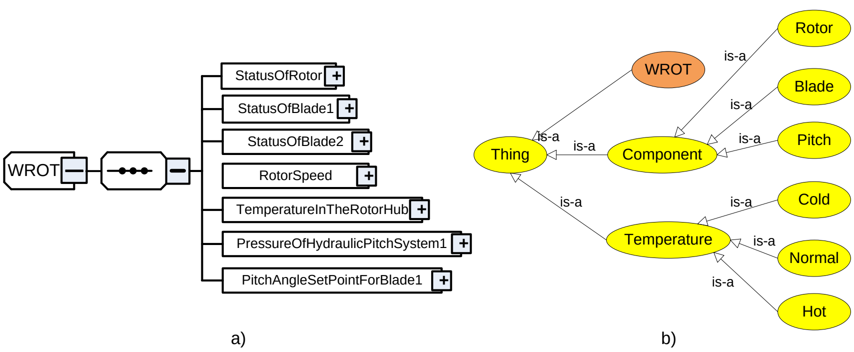


Figure A.11: WROT representation in an XML schema and an ontology

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

on the IEC 61400-25 standard is shown. Although the status of the rotor and the status of blade 1 can have different values, they still have the same semantics. Unfortunately, it is impossible to express this semantics in the information structure. In contrast, an ontology allows to describe the semantics using data properties (hasSpeed, hasStatus, hasPitchAngle) and object properties (hasBlade, hasComponent, hasTemperature) of an ontology. Fig. A.11b shows a graph visualization of the WROT using Protégé 4.1. Note that the relation between "WROT" and "Component" is not shown in Fig. A.11b. The temperature in the rotor hub is represented in the "Temperature" class which has three states (hot, normal, cold) defined by system operators. This class is also used to represent the temperature of the oil in the hydraulic pitch system of a blade.

4.3 Use of the ontology model

There are several ways to use the proposed ontology model. Three of them are shown in Fig. A.12 and described below.

- The information model derived from the semantic model is used to agree on data exchange formats, and share concepts between offshore wind partners.
- Input data for an instance of the semantic model (e.g., a virtual DB) is taken from both existing databases and new data (operational data, maintenance data, failure data). All entering data must either go through the instance which could be an "umbrella" over existing databases or a real database with a schema matching the model.
- Direct data and derived data (e.g., average energy production within an interval, temperature in different units) are exported from the model's instance. Data then can be used for decision support systems or forecasting systems, where artificial neural networks, fuzzy logic are used to predict and give support to decision-makers.

4.3.1 A RAMSI database

Currently, there is a database for offshore reliability data within the oil & gas industry, namely OREDA [10]. However, there is no such a database for offshore wind data. A *RAMSI* database (reliability, availability, maintainability, safety, and inspectability [44]) is proposed based on the developed information model. It is a

A

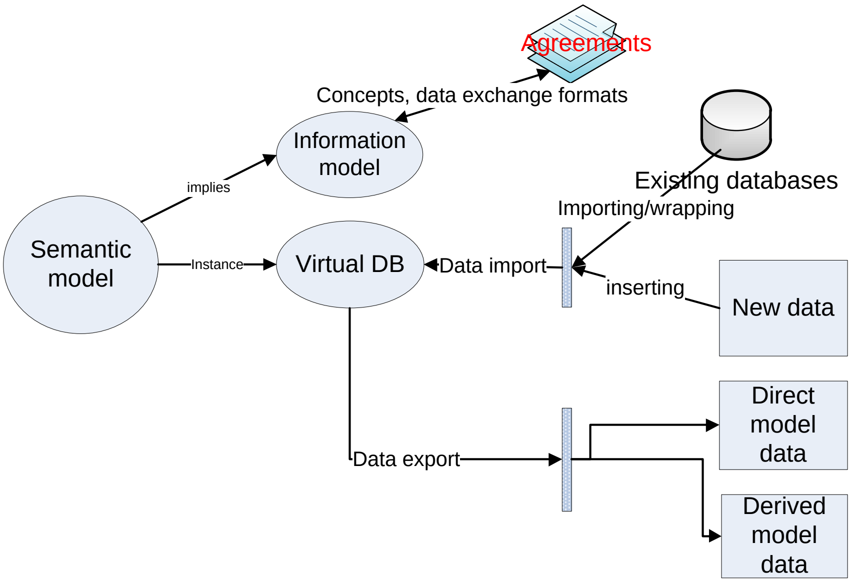


Figure A.12: Use of the semantic model

virtual database which is derived from the *semantic model*, as shown in Fig. A.12. The *RAMSI* contains all the necessary design elements to enable decreased downtime and stable production [18], and hence ensuring the reliability of offshore wind turbines. Currently, there is no such a RAMSI database for the offshore wind energy.

An ontology ideally is a well-defined, semantically enabled data structure for efficient representation of web data, whereas a relational database offers efficient and persistent storage and retrieval, and hence a good solution for ontology storage [28]. Constructing the *RAMSI* based on the *semantic model* bridges the semantic gap between ontology and relational database, and preserves the strengths of both semantic technology and *RAMSI* database. There are many articles dealing with the transformation of an ontology to a relational database, e.g., [2, 15]. Fig. A.13 presents a connection between the RAMSI database and the OWO. One advantage of this approach is that there is no need to pass new data through the OWO because the RAMSI database is already built based on the OWO, i.e. the logical structure of the RAMSI database and the OWO is the same.

4.3.2 Code generation for Web service development

Another use of the model is to use the schema of an information model derived from the semantic model to generate code. For example, using JAXB (Java XML

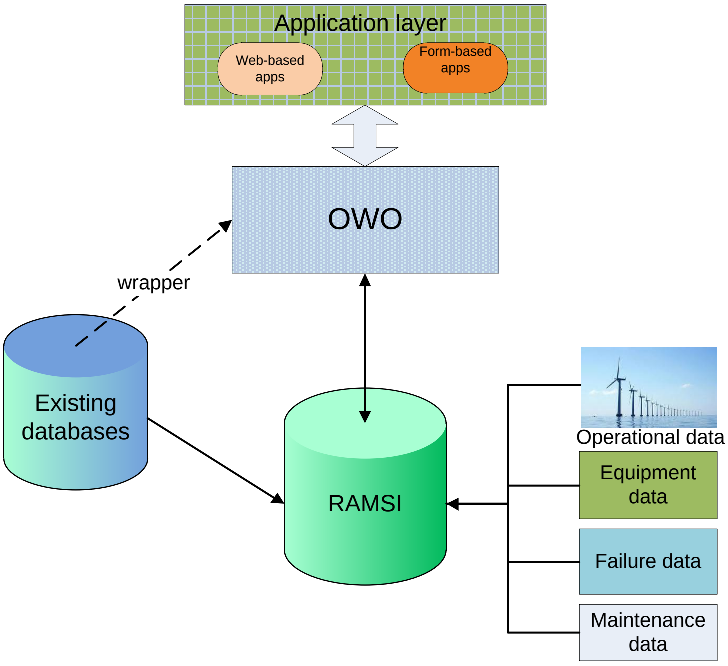


Figure A.13: RAMSI

binding) tools such as JAXB-XJC to generate Java classes, which are considered as Data Transfer Object (DTO), in order to express which information has to be secured. There are some benefits of code generation using the developed schema. First, this can save time in coding. Second, it can obtain the harmony between the web services and the semantic model, as well as to the RAMSI data storage.

4.3.3 A prototype system

A prototype system which proves the concepts presented in Sects. 3.1 and 3.2 has been developed. The system architecture is shown in Fig. A.14. Two data sources are used to provide data for the system. A database schema for WPP is derived from OWO. Data coming from data sources will be used for real-time analysis and it will be stored in the database as historical data. As mentioned in Sect. 4.3.1, data from the database will be loaded to OWO only when needed. The information exchange between OWO and the database is handled by different layers such as data transfer object and data access object. Mule ESB, an open source ESB framework is employed to coordinate the communications and message exchange between the sources and the system.

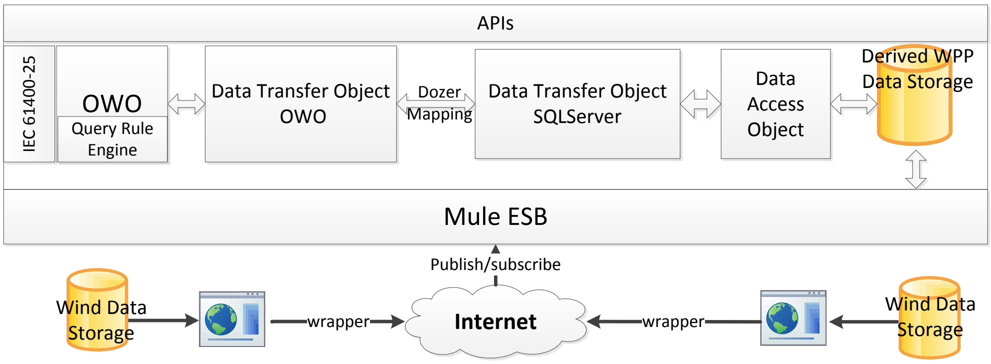


Figure A.14: A prototype data integration for offshore wind energy

5 Discussion

In this work, the system view technique is used to address the problem of operation and maintenance, including accessibility of data, connection and common understanding between offshore wind partners. Data integration could be a way to solve the problem and it has been pointed out in the work. In this connection, a system model based on existing technologies and agreement upon the integration (an approved standard could be a good example) is proposed.

To the authors' knowledge, in the offshore wind industry, no comparable system model has been reported. There are several work related to some part of the system architecture. For example:

- The authors of [37] proposed an ontology model for wind turbines' condition monitoring. The difference between this ontology and the OWO ontology is that OWO is built based on approved international standards such as IEC 61400-25, ISO 15926. Besides, OWO is specified for offshore wind turbines.
- A schematic presentation of a RAMS database was introduced by the authors of [18]. A shortcoming of this schema is that the authors did not take the semantics of data into consideration. Hence, the potential of data cannot be completely exploited.

In the oil & gas industry, there is a reference architecture for integrated operation proposed by the Norwegian Oil Industry Association [35]. It was a foundation and inspiration to propose a data integration framework for offshore wind farms. The proposed framework is equally applicable to problems of other industries which are compatible with the system view, for example, the maritime industry, where common understanding and agreement between participants are handled upon operations. The architecture may be applied to other industries with small changes.

Real data from operational wind power plants is available for verifying the information model and testing the system which was presented in the previous section. In this work so far, information and communication security were not taken into account. It is assumed that all communications are handled using secure channels.

6 Conclusions

Making offshore wind farm information available and accessible is important, since remote operations can be optimized and operation and maintenance hence improved. Consequently, the cost of O&M for offshore wind farms can be reduced. In this work, the offshore wind industry was analyzed and the challenges faced by the data integration of the industry was discussed. A proposal for solving the data integration problem in the form of a novel data integration framework was presented. The framework consists of the semantic model, the data source handling, and the information provisioning. Some preliminary results of the framework development were mentioned. The offshore wind ontology, OWO was taken as an example. The framework is based on state-of-the-art technologies, such as established ISO standards, RESTful web services, Enterprise Service Bus, and the Semantic Web.

Even though the idea of using semantic technologies to facilitate data exchange and improve operation and maintenance is not new in the oil and gas industry, it is quite new in the offshore wind industry, since this domain is still in the early stage of research. Furthermore, the suggested semantic model has applicability in other industries than offshore wind farms. For example, in the maritime industry, where a semantic model also plays an important role in supporting data exchange between ships efficiently.

Acknowledgment

This work has been funded by the Norwegian Centre for Offshore Wind Energy (NORCOWE) under grant 193821/S60 from the Research Council of Norway (RCN). NORCOWE is a consortium with partners from industry and science, hosted by Christian Michelsen Research.

A

References

[1] Antoniou, G., Harmelen, F.v.: Web ontology language: OWL. In: Handbook on Ontologies, International Handbooks on Information Systems, pp. 91–110. Springer Berlin Heidelberg (2009)

[2] Astrova, I., Korda, N., Kalja, A.: Storing OWL ontologies in SQL relational databases. International Journal of Electrical, Computer and Systems Engineering **1**(4), 242–247 (2007)

[3] Berners-Lee, T., Fischetti, M.: Weaving the Web: the original design and ultimate destiny of the World Wide Web by its inventor. HarperInformation (2000)

[4] Brédillet, P., Lambert, E., Schultz, E.: CIM, 61850, COSEM standards used in a model driven integration approach to build the smart grid service oriented architecture. In: Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference, pp. 467 –471 (2010)

[5] Butterfield, C.P., Musial, W., Jonkman, J., Sclavounos, P., Wayman, L.: Engineering challenges for floating offshore wind turbines. National Renewable Energy Laboratory (2007)

[6] Choux, M., Tyapin, I., G.Hovland: Leakage-detection in blade pitch control systems for wind turbines. In: Annual Reliability and Maintainability Symposium (RAMS). Nevada, USA (2012)

[7] Consortium, W.W.W., et al.: OWL Web Ontology Language Reference. W3C recommendation **10** (2004)

[8] Consortium, W.W.W., et al.: Resource description framework (RDF) (2004)

[9] Dan, A., Johnson, R., Arsanjani, A.: Information as a service: Modeling and realization. In: Systems Development in SOA Environments, 2007. SDSOA'07: ICSE Workshops 2007. International Workshop on. IEEE (2007)

[10] El-Thaljia, I., Liyanageb, J.: Integrated asset management practices for offshore wind power industry: A Critical review and a road map to the future. The Proceedings of The Twentieth (2010) International offshore and polar engineering conference (2010)

[11] Esteban, M.D., Diez, J.J., López, J.S., Negro, V.: Why offshore wind energy? Renewable Energy **36**(2), 444 – 450 (2011).

[12] EWEA: EU Energy Policy to 2050 (2011)

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

[14] Fischer, K., Besnard, F., Bertling, L.: Reliability-centered maintenance for wind turbines based on statistical analysis and practical experience. Energy Conversion, IEEE Transactions on (99), 1–12 (2012)

[15] Gali, A., Chen, C., Claypool, K., Uceda-Sosa, R.: From ontology to relational databases. In: Conceptual Modeling for Advanced Application Domains, *Lecture Notes in Computer Science*, vol. 3289, pp. 278–289. Springer Berlin / Heidelberg (2004)

[16] Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer Verlag (2004)

[17] Hahn, B.: Reliability assessment of wind turbines in Germany. In: 1999 European Wind Energy Conference, Nice, France, March, pp. 1–5 (1999)

[18] Hameed, Z., Vatn, J., Heggset, J.: Challenges in the reliability and maintainability data collection for offshore wind turbines. Renewable Energy **36**(8), 2154–2165 (2011)

[19] Herbert, G.J., Iniyan, S., Sreevalsan, E., Rajapandian, S.: A review of wind energy technologies. Renewable and Sustainable Energy Reviews **11**(6), 1117 – 1145 (2007).

[20] Horrocks, I., Patel-Schneider, P., Van Harmelen, F.: 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 (2003)

[21] IEC 61850 communication networks and systems in substations - part 7-1: Basic communication structure for substation and feeder equipment - principles and models (2003)

[22] IEC 61850 communication networks and systems in substations - part 7-3: Basic communication structure for substation and feeder equipment - common data classes (2003)

[23] ISO 15926 Industrial automation systems and integration - Integration of life-cycle data for process plants including oil and gas production facilities (2003)

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

A

[25] IEC 61970-301 Ed. 1: Energy management system application program interface (EMS-API) - Part 301: Common information model (CIM) base (2007)

[26] IEC 61400 wind turbines - part 3: Design requirements for offshore wind turbines (2009)

[27] Johnsson, A., Højholt, L.E.: Use of IEC 61400-25 to secure access to key O&M data. In: European Offshore Wind conference 2007. Berlin, Germany (2007)

[28] Khan, S., Qadir, M.: A generic framework for evaluation of ontology to relational database transformation process. In: Information and Emerging Technologies (ICIET), 2010 International Conference on, pp. 1–5. IEEE (2010)

[29] Køhrsen, T.: Deployment of IEC61400-25-4 - Mapping to Web services on an industrial PLC platform. Master's thesis, Technical University of Denmark (2008)

[30] Nguyen, T., Prinz, A.: Using semantics to facilitate data integration of offshore wind farms. In: Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean, pp. 430–433. IEEE (2012)

[31] Nguyen, T., Prinz, A., Friisø, T.: Reference architecture for remote operations of offshore wind farms. pp. 1051–1059. COMADEM, Stavanger, Norway (2011)

[32] Nguyen, T., Prinz, A., Friisø, T., Nossum, R.: Smart grid for offshore wind farms: towards an information model based on the IEC 61400-25 standard. In: Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES, pp. 1–6. IEEE (2012)

[33] Nielsen, J., Sørensen, J.: Bayesian networks as a decision tool for O&M of offshore wind turbines. ASRANet Ltd. (2010)

[34] NIST: NIST framework and roadmap for smart grid interoperability standards, NIST Special Publication 1108 edn. (2010)

[35] OLF The Norwegian Oil Industry Association: Reference architecture of IT systems for OLF's IO G2. Tech. rep. (2008)

[36] Olsen, A., Osdil, B., Poulsen, B., Pedersen, K., Johansen, K.: Prototype of generic server for wind power plants using IEC 61400-25 standard. International journal of distributed energy resources **3**(4), 273–290 (2007)

[37] Papadopoulos, P., Cipcigan, L.: Wind turbines' condition monitoring: an ontology model. In: Sustainable Power Generation and Supply, 2009. SUPERGEN'09. International Conference on, pp. 1–4. IEEE (2009)

[38] Rohjans, S., Uslar, M., Bleiker, R., González, J., Specht, M., Suding, T., Weidelt, T.: Survey of smart grid standardization studies and recommendations. In: Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on, pp. 583–588. IEEE (2010)

[39] Schwarz, K.: Monitoring and control of power systems and communication infrastructures based on IEC 61850 and IEC 61400-25. In: 2010 DistribuTech conference & Exposition (2010)

[40] Studer, R., Benjamins, V., Fensel, D.: Knowledge engineering: principles and methods. Data & knowledge engineering **25**(1-2), 161–197 (1998)

[41] The Carbon Trust. Offshore wind power: big challenge, big opportunity. Maximising the environmental, economic and security benefits. Tech. rep. (2008)

[42] Thomsen, K.E.: What Is an Offshore Wind Farm? In: Offshore Wind, pp. 1 – 7. Elsevier, Boston (2012).

[43] Tiusanen, R., Jännes, J., Liyanage, J.: Framework to assess system risks associated with design and deployment of large scale offshore wind farms in northern context. In: The 21st International Offshore (Ocean) and Polar Engineering Conference & Exhibition, pp. 540–547. ISOPE, Maui, Hawaii (2011)

[44] Tiusanen, R., Jannes, J., Reunanen, M., Liyanage, J.: RAMSI management - from single analyses to systematic approach. pp. 1588–1596. COMADEM, Stavanger, Norway (2011)

[45] Uslar, M.: The Common Information model for utilities: An Introduction and Outlook on Future Applications. Proceedings der XMLTage pp. 135–148 (2006)

[46] Uslar, M., Grüning, F., Rohjans, S.: A Use Case for Ontology Evolution and Interoperability: The IEC Utility Standards Reference Framework 62357, chap. 9, pp. 187–209. IGI Global (2010)

[47] Van de Pieterman, R., Braam, H., Obdam, T., Rademakers, L., van der Zee, T.: Optimisation of maintenance strategies for offshore wind farms (2011)

[48] Whittle, M., Shin, W., Trevelyan, J., Wu, J.: A Parametric Study of the Effect of Generator Misalignment on Bearing Fatigue Life in Wind Turbine. In: EWEA 2011, Scientific Proceedings of, pp. 24–27. EWEA, Brussels (2011)

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

Appendix B

Paper B

Title	Offshore Wind Metadata Management
Authors	Trinh Hoang Nguyen[a], Rocky Dunlap[b], Leo Mark[b], Andreas Prinz[a], Bjørn Mo Østgren[c], and Trond Friisø[d]
Affiliation(s)	[a] Department of ICT, University of Agder, Grimstad, Norway
	[b] School of Computer Science, Georgia Tech, Atlanta, USA
	[c] Statkraft AS, Oslo, Norway
	[d] Orgio Solutions AS, Arendal, Norway
Journal	International Journal of Metadata, Semantics and Ontologies
Copyright	©Inderscience Enterprises Ltd

B

Offshore Wind Metadata Management

Trinh Hoang Nguyen[a,*], Rocky Dunlap[b,**], Leo Mark[b,**], Andreas Prinz[a,*], Bjørn Mo Østgren[c,+], and Trond Friisø[d,++]
[a] Department of ICT, University of Agder, Grimstad, Norway
[b] School of Computer Science, Georgia Tech, Atlanta, USA
[c] Statkraft AS, Oslo, Norway
[d] Orgio Solutions AS, Arendal, Norway
E-mails: *{trinh.h.nguyen, andreas.prinz}@uia.no
** {rocky, leomark}@cc.gatech.edu
+ BjornMo.Ostgren@statkraft.com ; ++ trond.friiso@origo-s.no

Offshore wind energy is gaining more and more attention from industry and research community due to its high potential in producing green energy and lowering price on electricity consumption. However, offshore wind is facing many challenges, and hence it is still expensive to install in large scale. It therefore needs to be considered from different aspects of technologies in order to overcome these challenges. One of the problems of the offshore wind is that information comes from different sources with diversity in types and format. Besides, there are existing wind databases that should be utilized in order to enrich the knowledge base of the wind domain. This paper describes an approach to managing offshore wind metadata effectively using semantic technologies. An offshore wind ontology has been developed. The semantic gap between the developed ontology and the relational database is investigated. A prototype system has been developed to demonstrate the use of the ontology.

Keywords: offshore wind metadata; metadata management; data integration; offshore wind ontology

1 Introduction

Offshore wind energy is gaining more and more attention from industries and community. Several offshore wind farms have been developed such as Sheringham Shoal, Horns Rev II. The IPCC (Intergovernmental Panel on Climate Change) special report on renewable energy sources and climate change mitigation states that:

"Estimates of global technical potential range from a low of 70 EJ/yr (19,400 TWh/yr) (onshore only) to a high of 450 EJ/yr (125,000 TWh/yr) (onshore and near-shore) among those studies that consider relatively more development constraints. Estimates of the technical potential for offshore wind energy alone range from 15 EJ/yr to 130 EJ/yr (4,000-37,000 TWh/yr) when only considering relatively shallower and near-shore applications; greater technical potential is available if also considering deeper water applications that might rely on floating wind turbine designs" [52].

In Europe, the EU's renewable energy policy is expecting 34% renewable electricity in 2020 and 100% renewables by 2050. By then wind energy alone could provide 50% of Europe's electricity [13]. It is apparent that moving wind farms from onshore to offshore is likely to bring significant benefits for wind energy production due to the high-quality wind resources and the large scale of wind turbines. However, the offshore wind industry is facing some challenges, such as high costs of wind turbine installation, transmission access, operational integration, and operation & maintenance. The offshore wind energy is therefore still an immature technology [12]. One way of overcoming these challenges is to implement advanced information technology. Data mining, fault detection techniques, condition monitoring, decision support systems are being developed in order to optimize the power output and increase life-cycle of equipments. Wind speed, wind direction, generator speed, yaw angle, and blade pitch angle are used for optimizing power output [23]. Some applications use wind speed and power output as input for detecting faults in wind turbine systems [54, 39].

Data are one of main assets because some of it is used for decades [49]. With more reliable data available and accessible, it is possible to make better decisions and predictions, and thereby increase the life of equipment and reduce operational costs. Indeed, the results of wind energy applications such as power optimization, fault detection rely on the availability and quality of data [23, 16, 55]. However, more data does not always mean better results if the data are organized in an ambiguous manner. It is therefore important to make data more useful by exploiting the semantics of the data.

It is noted that in the offshore wind energy, many partners will have their own applications and use their own data formats. Besides, there are many valuable data sources from existing onshore wind farms. The question is how to utilize these data sources and manage them in a way that enables knowledge sharing and common understanding on the domain concepts.

Ontologies are used to facilitate integration within and across business domains, creation of autonomous solutions, and ability to store data over time. An ontology is needed to make an abstract model of a phenomenon by identifying the relevant concepts of that phenomenon [46]. Ontologies are the main element of the ontology-based data integration approach. In this approach, ontologies are considered as mediators between users' queries and data sources. The ontology-based approach offers three variations, i.e., single ontology, multiple ontology, and hybrid ontology approaches [45]. In industries, the ontology-based data integration approach has been used intensively. Indeed, ontologies have been developed in different domains such as the ISO 15926 - "Industrial automation systems and integration - Integration of life-cycle data for process plants including oil and gas production facilities" is considered an ontology for the oil and gas domain [5, 48]. An ontology for multimedia is introduced in [29]. In the health care sector, many ontology models have also been proposed, for example [21, 11]. In the wind energy domain, Papadopoulos and Cipcigan (2009) introduce an ontology model for wind turbines' condition monitoring. The drawback of the proposed model is that it is not based on any wind power plant (WPP) standards and therefore it is hard to integrate the work into other applications or services. Zhu et al. (2008) present in their work an ontology model based on the IEC 61400-25. However, there is no report on the implementation of their ontology model. The IEC 61400-25 is a standard proposed by the International Electrotechnical Commission (IEC). The standard is entitled "Wind turbine - Communications for monitoring and control of wind power plants".

In our work, we introduce an approach to managing offshore wind metadata using semantic technologies. We use the IEC 61400-25 standard and existing wind energy databases as sources of wind domain concepts. We prove that semantic technologies can resolve semantic inconsistency issues in the offshore wind domain effectively. The rest of the paper is organised as follows. Section 2 presents an introduction to wind energy system and the use of data in the offshore wind energy. The section ends with remarks on challenges of offshore wind metadata management and a solution to overcome these challenges. Section 3 introduces a prototype of our proposed ontology model and how to bridge the gap between ontologies and relational databases. Section 4 describes a prototype system where we show a use of our ontology model and how to integrate an existing database into our system. Discussion and future work are presented in section 5. Finally, section 6 contains a summary of the paper.

2 Offshore wind energy

The concept of harvesting power from the wind has been introduced a long time ago. The first automatically operated wind turbine was introduced in 1887 by Charles F.

Brush [38], and the first offshore wind power plant was built in 1991 in Denmark [52]. After decades of development, wind power plants are getting bigger and bigger in size (length of blades, tower, rotor) as well as amount of energy production (multi-megawatt). Wind energy becomes one of the most promising renewable energy sources.

2.1 Wind energy system

A wind power conversion system consists of several components as shown in Fig. B.1. The main components of a wind power plant are rotor, generator, converter, and transformer.

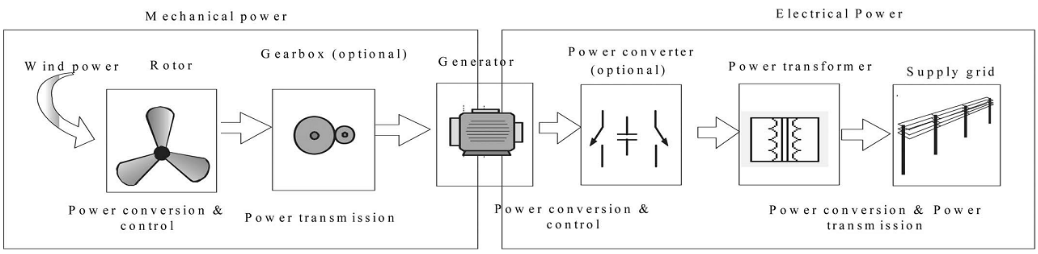


Figure B.1: Wind power conversion system [10]

Wind power is captured by blades and is transformed to mechanical power by a gearbox. The mechanical power is then converted into electrical power by a generator. The electricity is transferred to substations in high voltage in order to lower the loss of energy. It is then integrated in an electricity grid and distributed to consumers.

Offshore wind turbines are normally unmanned, hence any operation of them is automatic or remote. Wind turbine automatic operations are support by several control loops such as pitch control, generator torque control, and yaw control. For instance, a closed loop control is used to adjust the motor variables to be as closed as possible to the turbine curves [26].

Closed feedback loops are used in single wind turbine control. However, when it comes to a wind farm or wind park control, it is better to use open loop controller rather than a closed loop controller to avoid the conflict between the frequency control in the individual wind turbines and the wind farm power control [47]. A Wind farm needs more overview data in order to achieve a global control and optimize the production of the whole wind farm. The importance of global control becomes even more for offshore wind farms where accessibility to wind turbines is an issue.

2.2 The use of data within the offshore wind industry

Data are used in different activities within the offshore wind industry such as operation and maintenance, transmission, grid integration, market energy. In order to show the use of data in different activities we have conducted a brief review as shown in Table B.1.

Operation and maintenance activities in a wind farm include monitoring, predicting wind farm power, inspection, diagnosing and detecting faults. Carrying maintenance and inspection on-time can reduce the risk of using equipments and downtime rates, and hence increasing energy production. The energy market regulates the energy price based upon the energy generated in a certain period. Energy market allows investors to bid on energy price in advance. All these activities need accessible and reliable information from various data sources in the offshore wind energy. Besides, the development of smart grids has already posed challenges to the onshore wind energy, and now it becomes even more challenging when the offshore wind energy is about to join the grids.

2.3 Challenges in managing offshore wind metadata & solution

Components of WPP are produced by different vendors. Each component has its own software and perhaps its own database. As a result, a software environment of WPP consists of multiple applications with incompatible interfaces and data formats and the inability to communicate with each other. In addition, there exist some databases provided by companies that have been collecting wind data for years. These databases are valuable sources for improving the quality of wind turbine components. The data sources are autonomous, distributed and heterogeneous systems so data reside in many incompatible formats and cannot be systematically managed, integrated and unified. Consequently, semantic inconsistency has become an even greater problem for the explicit information or knowledge sharing among users or applications. Therefore, the integration and utilization of information resources has become one of the challenging problems faced in offshore wind communication today [32]. These inconsistencies can be solved by effectively managing metadata.

Fig. B.2 shows two tables from a Statkraft wind energy database. The first table displays stationID, timestamp, mean value of pitch control A position, mean value of generator speed (wtc_GenRpm_mean), and mean value of nacelle position (wtc_NacelPos_mean). The second table shows timestamp, stationID, and active

Table B.1: A review of data usage within the offshore wind research

Activity	**Target**	**Input data**	**Ref.**
Operations	Power optimization	Wind speed, wind direction, generator speed, yaw angle, blade pitch angle, power output.	[23]
	WT control	Wind speed, blade pitch angle, generator torque, power output, rotor speed	[22]
	Fault detection	Active power output, anemometer-measured wind speed, nacelle temp, gearbox bearing temp, gearbox lubricant oil temp, generator winding, power factor, reactive power, phase currents	[55, 16, 39]
Maintenance	Condition based maintenance	Number of wind turbines in a wind farm, number of critical components considered in a wind turbine, failure probability, age of a component since the last maintenance/inspection, cost of replacement a component, crack initiation rate, crack time to failure	[7, 50]
Energy market	Optimal day-ahead bids	Weather prediction models, local meteorological measurements,active power, wind power plant characteristic, nearby terrain and topography, recent energy price	[9]
	Energy price prediction	Measured wind power, predicted wind power, wind speed and wind direction	[8]
Transmission	Maximize utilization of wind resources	Power out, demand, voltage angle, maximum capacity of line, network topology, length of transmission line	[27]
Grid integration	Power System Transient Stability Studies	Active power, reactive power, voltage, frequency, wind speed, rotor speed, generator speed, pitch angle, wind direction	[25]

B

power (wtc_ActPower).

StationId	TimeStamp	wtc_PitcPosA_mean	wtc_GenRpm_mea	wtc_NacelPos_mea	TimeStamp	StationI	wtc_ActPower_n	wtc_VoltPhR_n
2,300,249	2012-01-01 00:00:00	-0.680968	1002.45	127.982	2012-01-01 00:00:00	2,300,249	126.095	389.8
2,300,250	2012-01-01 00:00:00	0.498	1501.21	121.763	2012-01-01 00:00:00	2,300,250	264.843	389
2,300,251	2012-01-01 00:00:00	0.522061	1501.14	320.471	2012-01-01 00:00:00	2,300,251	248.743	390.1
2,300,252	2012-01-01 00:00:00	0.396181	1501.35	160.459	2012-01-01 00:00:00	2,300,252	303.276	389.7
2,300,274	2012-01-01 00:00:00	0.598759	1501.04	313.628	2012-01-01 00:00:00	2,300,274	219.986	390.6
2,300,275	2012-01-01 00:00:00	-0.531986	1001.79	162.462	2012-01-01 00:00:00	2,300,275	86.9156	388.5
2,300,276	2012-01-01 00:00:00	0.471215	1501.24	176.003	2012-01-01 00:00:00	2,300,276	277.181	389.4
2,300,277	2012-01-01 00:00:00	-15.5433	738.791	121.037	2012-01-01 00:00:00	2,300,277	74.8694	376
2,300,278	2012-01-01 00:00:00	-0.705978	1002.65	313.919	2012-01-01 00:00:00	2,300,278	130.067	389.5
2,300,279	2012-01-01 00:00:00	0.591489	1501.06	113.381	2012-01-01 00:00:00	2,300,279	229.706	389.2

Figure B.2: Sample of 2 tables from Statkraft database

Queries against the database can be made by using Structured Query Language (SQL). For example, if we want to retrieve generator speed and active power of a wind turbine which has an identification number is “2300249” for a period from the 1st of January 2012 to the 2nd of January 2012, we can make a query as follows:

SELECT $tblA.[TimeStamp]$, $tblA.StationId$,$tblA.wtc\_GenRpm\_mean$ **AS** GeneratorSpeed,

$tblB.wtc\_ActPower\_mean$ **AS** ActivePower **FROM** $tblA$ **INNER JOIN** $tblB$ **ON**

$tblA.[TimeStamp]$ = $tblB.[TimeStamp]$ **AND** $tblA.StationId$ = $tblB.StationId$ **WHERE**

$tblA.StationId$ = 2300249 **AND** $tblA.[TimeStamp]$ >’2012-01-01T00:00:00’ **AND**

$tblA.[TimeStamp]$ $\leq$’2012-01-02T00:00:00’

Another example of an existing database is shown in Fig. B.3. Different from the previous database, the database uses an additional table for storing names of columns. For example, the 1st column stores timestamp, 2nd column stores status code, 3rd column stores wind speed, 7th column stores nacelle direction.

1	2	3	4	5	6	7
2002-09-09 00:09:52	256	6.6331481933593	4.4197492599487	8.8346118927002	0.1293126046657	79.799926
2002-09-09 00:19:49	256	6.0921554565429	4.1610517501831	7.5616807937622	0.1055947542190	81.369812
2002-09-09 00:29:55	256	6.1096229553222	4.5567989349365	8.0747776031494	0.1346896588802	81.369812
2002-09-09 00:39:52	256	5.9545402526855	4.3056459426879	7.4666337966919	0.1217719763517	81.369812
2002-09-09 00:49:49	256	6.2579078674316	4.0648803710937	8.6401348114013	0.1546563208103	81.369812
2002-09-09 00:59:56	256	6.2300033569335	4.0093669891357	7.8787212371826	0.1374699473381	81.369812
2002-09-09 01:09:53	256	6.4166011810302	4.9183311462402	8.3585872650146	0.1254754364490	81.369812

Figure B.3: Sample of another existing database

Both presented databases are simple and good enough to handle queries against the databases. But when it comes to integrate these databases into different systems or vice versa, there will be problems with mapping different database structures due to mismatch in naming domain concepts. For example, wind speed in one database can be named as “wtc_WindSpeed”, but in another database it can be named as “WindSpeed” or “WSpd”. It is obvious that these concepts have the same semantics - information about wind speed. But syntactically they are different. It is easy for a human being to recognize the similarity, but it is not an easy task for a machine. Resolving semantic heterogeneity not only helps machines understand the domain concepts, but also gives users a unified way to access distributed data. The semantics of data can be exploited and hence improving WPP operations. Thus the idea of creating an offshore wind ontology from the terminologies in order to share, reuse knowledge, and reason behaviours across domain and tasks, is important. An ontology should be developed in order to cover key concepts in the offshore wind domain and their semantic relationships.

Even though the idea of having an ontology to tackle data integration issues is well-known and there are tools to support the development of an ontology, creating an ontology is still not an easy task. Indeed, it involves not only ontology engineers, domain experts but also the concepts from standards and existing databases within the wind industry. The proposed OWO will be an effort to overcome the challenge by providing offshore wind partners a common way to talk when it comes to agreement on data exchange and application integration in a common platform. OWO is developed based on the IEC 61400-25 and concepts from existing wind databases (e.g., Statkraft). Thus, it also covers concepts within the onshore wind energy. Fig. B.4 illustrates an overview of OWO development and its relation to relational database. TBox stands for terminology box which describes the concepts and relations between them, whereas ABox stands for assertion box which describes instances.

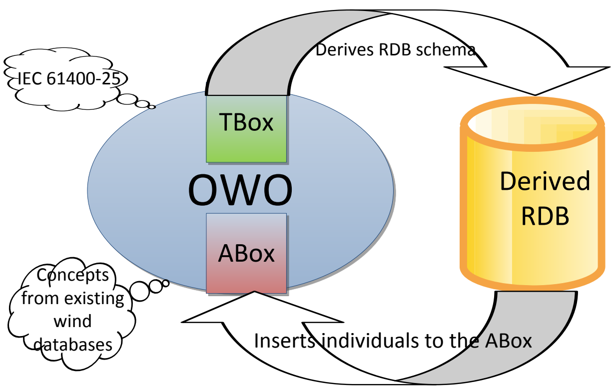


Figure B.4: Relation between OWO and the derived RDB

2.4 The IEC 61400-25 standard

A typical problem for data exchange is mismatch of terms and formats between sender and receiver. Approved standards are necessary in order to make the data exchange process clear. The IEC 61400-25 standard proposes an approach which separates WPP into different components. The standard contains six parts that cover informational exchange model, mapping to communication profiles, and conformance testing. The purpose of the standard is stated as follows: “The IEC 61400-25 series has been developed in order to provide a uniform communications basis for the monitoring and control of wind power plants” [18]. The standard has been reported in several work, e.g. [40, 19, 34]. Even though an information model has been proposed in the standard, it is specially designed for easing data exchange only and the semantics of data has not been considered. In this work, the IEC 61400-25 standard serves as a backbone for the proposed ontology model and it is also the source of domain concepts [30, 32]. The standard has been used together with domain concepts from existing databases as well as domain experts in order to build the proposed ontology.

3 A proposed ontology model for wind power plant

Semantic technologies have been developed to extract semantics of data and organize data in a semantic way. Semantic technologies can be used for representing data and knowledge, integration, and interpretation [42]. In order to make knowledge and information understandable by machines, ontologies are used. Ontologies represent relevant concepts and relationships in a domain. In ontologies, concepts, properties, relations, functions, constraints, and axioms of a particular domain are explicitly defined [15]. There are some advantages of ontologies: (1) sharing common understanding of the structure of information among people or software agents; (2) enabling reuse of domain knowledge; (3) making domain assumptions explicit; (4) analyzing domain knowledge; (5) separating domain knowledge from the operational knowledge. This section discusses semantic technologies and our proposed ontology model.

3.1 Ontology representation

There are several ontology languages such as OIL, DAML-ONT, DAML+OIL, and OWL [24]. Web Ontology Language (OWL), a language proposed by World Wide Web Consortium (W3C) Web Ontology Working Group, is being used intensively by research communities as well as industries. Basically, ontologies can be represented by using RDFS (RDF Schema). However, a number of other features are missing in RDFS such as disjointness of classes (e.g., in RDFS, we cannot state that *HydraulicSystem* and *HeatingSystem* are disjoint classes), boolean combinations of classes (sometimes we need to build new classes by combining other classes using union, intersection, and complement, e.g., *WPP* is the disjoint union of the classes *WROT* and *WGEN*), cardinality restrictions (we cannot say that *WPP* has at least one *WCNV* - wind turbine converter component, or *WPP* has exactly one *WROT* - wind turbine rotor component). OWL is an extension of RDFS, in the sense that OWL uses the RDF meaning of classes and properties [17, 6, 1].

There are three variants of OWL. OWL Lite was designed to support simple class hierarchy and simple constraints. OWL DL (DL stands for “Description Logic”) was developed to support existing DL and to provide possibility to work with reasoning systems. The third one is OWL Full which is the most expressive language in OWL family. OWL Lite is not rich enough to cover the offshore wind domain. OWL Full is too expressive and it is not decidable [44]. A complex ontology can cause big problem since a reasoner might take long time to do reasoning over the knowledge. We therefore select OWL DL as an ontology language to develop our proposed ontology since it is decidable [17] and there are tools to work with it.

3.2 Ontology reasoning and querying

A reasoner is a piece of software that is able to infer logical consequences from a set of asserted facts or axioms. It is used to ensure the quality of ontologies. It can be used to test whether concepts are non-contradictory and to derive implied relations. There are some existing DL reasoners such as FaCT, FaCT++, RACER, DLP and Pellet. A reasoner has following features: satisfiability, consistency, classification, and realization checking [44].

For RDB, SQL is the query language of choice. But for ontologies, SPARQL and SQWRL (Semantic Query-Enhanced Web Rule Language) [33] are used to build queries. SPARQL is an RDF query language and SQWRL is a SWRL-based language for querying OWL ontologies. SPARQL extensions such as SPARQL-DL [43], SPARQL-OWL [20] can be used as an OWL query language in many applications. However, “SPARQL has no native understanding of OWL. It operates only on its RDF serialization and has no knowledge of the language constructs that those serializations represent. As a result, it cannot directly query entailments made using those constructs” [33]. SQWRL employs Pellet as a reasoner and Jess engine as rule engine to parse SWRL rules. For example, if we want to retrieve all wind farms and number of WPP in them, we can have an SQWRL query as shown below.

$WF(?p) \wedge hasWFName(?p, ?name) \wedge hasTotalNumberWPP(?p, ?number)$
$\rightarrow sqwrl : select(?p, ?name, ?number)$

3.3 OWO - an offshore wind ontology

We use Protégé 4.1 to build OWO since it is a platform-independent and open source ontology editor. Besides, it supports various plug-ins to visualize and document ontologies. In order to make sure that our OWO ontology is consistent, we use the Pellet reasoner to check every time when a new class, property, or axiom inserted. We also follow the strategy and methodology development described in [32]. OWO is built by defining an ontology for each wind turbine component (ontology for WT generator, WT rotor, WT tower, etc.). The ontology development will start with basic terms that are described in the IEC 61400-25, and then continue specifying and generalizing them as required.

3.3.1 OWO development

A general view of OWO is shown in Fig. B.5. The dashed lines denote object properties that relate a class to another class (e.g., *hasWPPComponent* relates *WPP* to *WPPComponent*) and the solid lines imply subclass-of relationships. The dot lines denote equivalent class relationships. Each rectangle stands for a class in OWO ontology. The plus “+” on the top-left of a rectangle indicates that the class contains subclasses. *Thing* is the top-most class in OWO.

Each logical node defined in the IEC 61400-25 is represented by a class in OWO. The main classes of OWO are depicted in Fig. B.6, where “1:1” means “must be

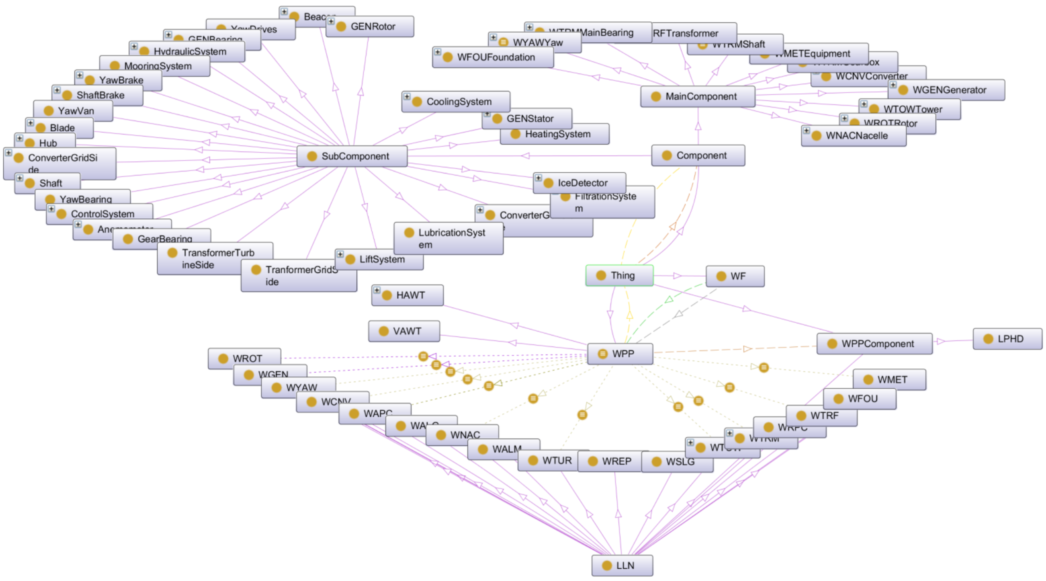


Figure B.5: An overview of OWO (better view in color)

exactly one", "1:1..*" means "at least one" and "1:0..*" means "may be zero or more".

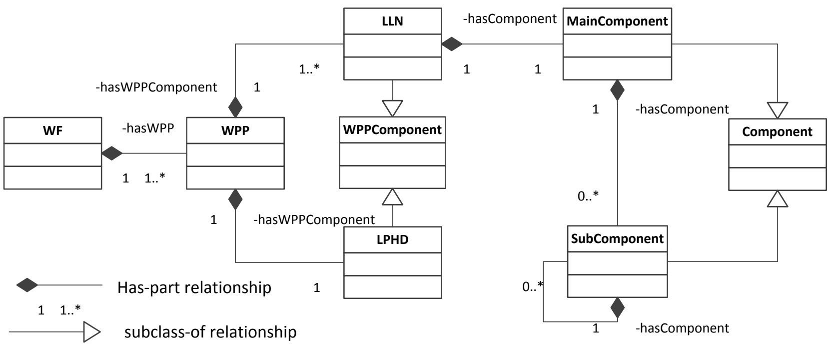


Figure B.6: OWO main classes and their relations

Object properties

In order to specify relations between objects (classes) in OWL, we have defined the object properties as shown in Table B.2. The main object properties are *hasWPP*, *hasWPPComponent*, and *hasComponent*.

We used the following cardinality constraints to express the relations between objects: *some*($\exists$), *min*($\geq$), *max*($\leq$), *exact*($=$). *Some* is used to create existential restriction which describes a class of individuals that have at least one relationship along a specified property to an individual that is a member of a specified class.

Table B.2: Object properties used to define relations in OWO

Object property	Domain	Range	Function
hasWPP	WF	WPP	relates a wind farm to wind power plants
hasWPPComponent	WPP	WPPComponent	relates a wind power plant to WPP components
hasComponent		Component	relates a WPP component to another WPP components

For example, *WPP* $\sqsubseteq \exists$ *hasWPPComponent.WGEN* expresses the fact that a wind power plant must have at least one *WGEN* component. *Min*, *max*, and *exact* are used to express the fact that an individual is connected by an object property to at least, at most, and exactly a given number of instances of a specified class expression [28]. For instance, *WPP* $\sqsubseteq$*(=1hasWPPComponent.WYAW)* implies that a wind power plant has exactly one wind turbine yawing information component.

Data properties

In the offshore wind energy, information is divided into different categories: equipment information, discrete state information, analog state information, and control information [30]. Similarly, we classify data properties defined in OWO into several groups as shown in Table B.3.

Table B.3: An example of data properties defined in OWO

Group	**Data Property**	**Domain**	**Range**
Equipment Info			
	hasBladeLength	*Blade*	Integer
	hasRotorDiameter	*WPP*	Integer
Descriptive Info			
	hasTotalEnergyProduction	*WTUR*	Float
	hasYawingOperationHours	*WYAWYaw*	Float
Discrete State Info			
	hasStatus	*WPP*	Integer
Analog State Info			
	hasOilPressure	*HydraulicSystem*	Float
	hasCurrent	*GENStator*	Float

Wind power plants are divided into two disjoint sets, horizontal axis wind turbines and vertical axis wind turbines. Horizontal axis wind turbines are then further divided into downwind and upwind wind turbines [52] [35]. Due to limitation of space, we present the ontology development of three WPP components.

WROT - Wind turbine rotor information

A wind turbine rotor converts kinetic energy from the wind into mechanical energy. In OWO, wind turbine rotor information is defined under the name *WROT*. Fig. B.7

shows parts of *WROT* in a class diagram.

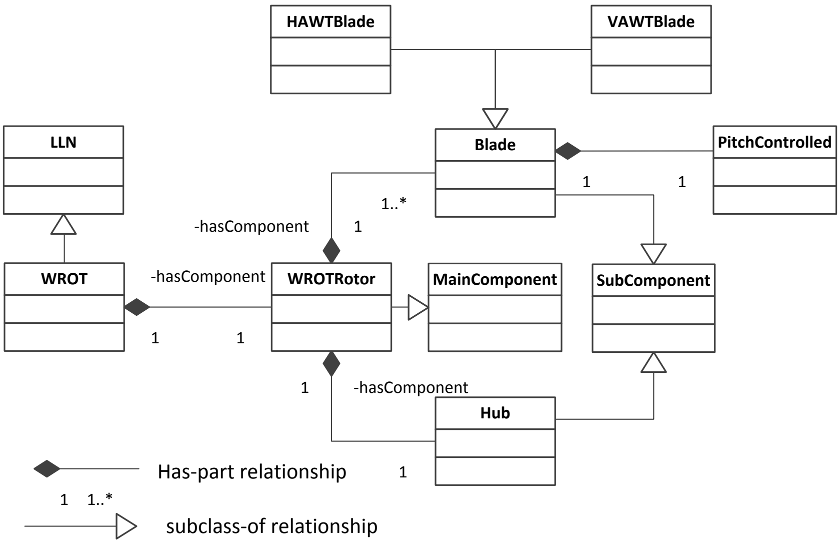


Figure B.7: *WROT* class diagram

WROT component consists of a turbine hub and blades. Blades are classified in two categories: horizontal axis wind turbine blades (single bladed, double bladed, three bladed, multiple bladed), and vertical axis wind turbine blades (straight blade and curved blade). *WROTRotor* is part of *WROT* and object property *hasComponent* is used to describe their relationship. Fig. B.8 depicts how to specify the relationship between *WROTRotor* class and its components in Protégé.

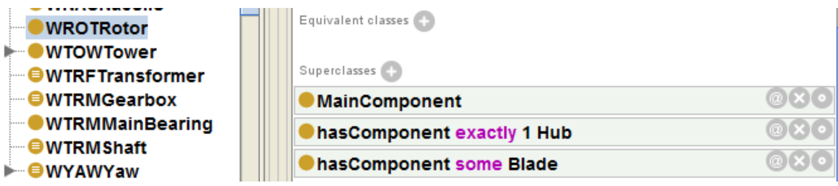


Figure B.8: Part-of relationship of *WROTRotor* class

WROTRotor class is a subclass of a class which is-a *MainComponent* class and has *Blade* and *Hub* as components. Each blade can have a pitch control system to adjust the pitch angle when the wind speed is higher than rated one. A pitch control system is a subclass of control systems in OWO. Besides the pitch control system, there are passive stall power control and active stall power control systems for wind turbines [37].

WTRM - Wind turbine transmission information

A wind turbine transmission consists of the main bearing, high-speed shaft, gearbox, and low-speed shaft. The ratio of the gearbox determines the rotation division and the rotation speed that the generator sees. For example, if the ratio of the gearbox is N to 1, then the generator sees the rotor speed divided by N. This rotation is finally sent to the generator for converting mechanical to electrical power. Recently, a gearless concept has been introduced, meaning that a wind power plant can operate with direct drive [36]. In order to express the constraint about a *WRTM* can have at minimum zero gearbox in OWO, we use cardinality constraint *min (hasComponent min 0 WTRMGearbox)*. Fig. B.9 shows how to define the relationship of *WTRM* and its components in Protégé. A class diagram of WTRM is depicted in Fig. B.10.

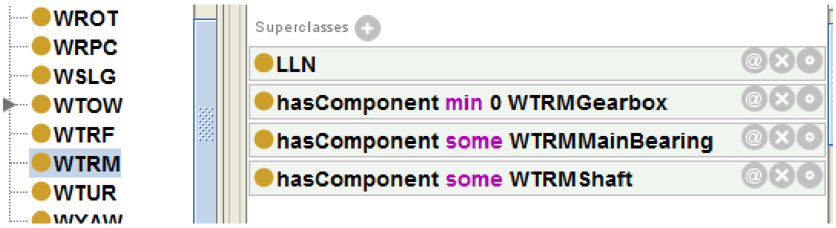


Figure B.9: Relationship between WTRM and its components

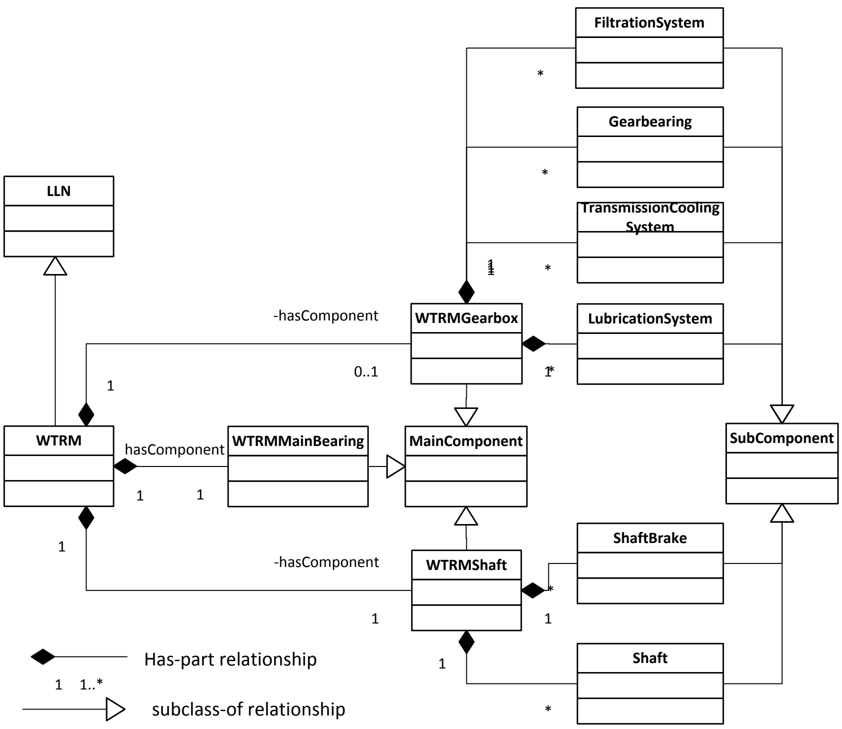


Figure B.10: *WTRM* class diagram

WGEN - Wind turbine generator information

B

At the moment, OWO covers three main types of wind turbine generator: doubly fed induction generator, synchronous generator, and induction generator. The generator classification is given in [10]. The *WGEN* is modeled as shown in Fig. B.11.

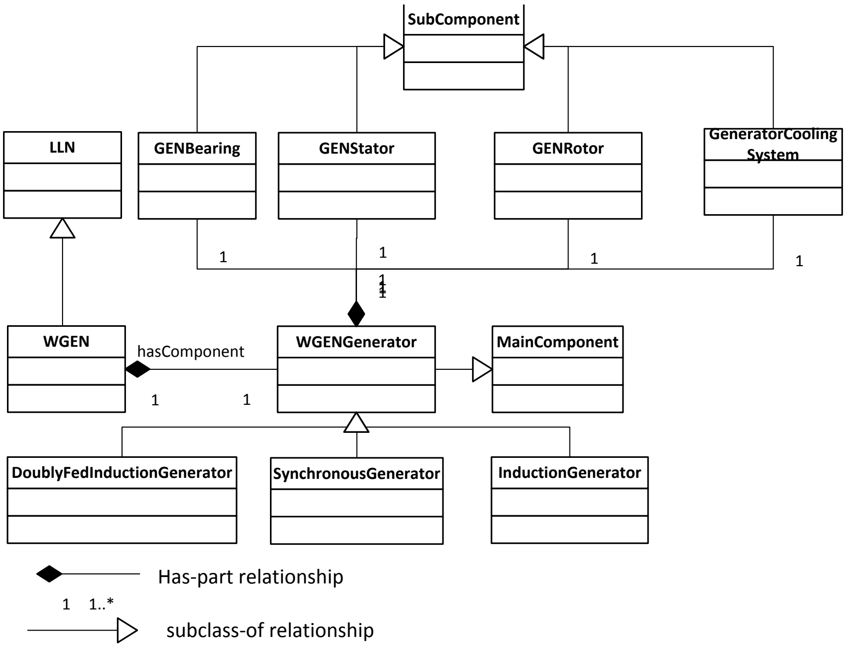


Figure B.11: *WGEN* class diagram

WGEN consists of one component called *WGENGenerator*. *WGENGenerator* consists of the generator cooling system, generator rotor, generator stator, and generator bearing. *WGENGenerator* is related to these components by an object property *hasComponent* as shown in Fig. B.12.

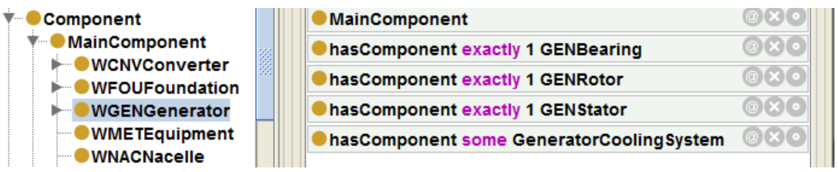


Figure B.12: *WGENGenerator* and its components

OWO individuals

Actual WPP data is an instance of OWO. Fig. B.13 shows two instances of wind farm class (WF_Greater_Gabbard and WF_Thorntonbank). There are two types of WPP in the WF_Thorntonbank wind farm. They are REpower 5M and REpower 6M. The wind farm contains 36 wind turbines in total.

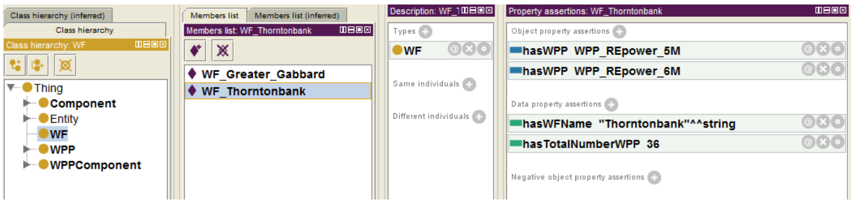


Figure B.13: An example of OWO individuals

3.3.2 OWO consistency checking

Knowledge modeling consists of creating concepts and organizing them into taxonomy. Consistency checking ensures that classes and instances in the knowledge base (KB - equivalent an ontology) have attributes which conform to the knowledge model [53]. For instance, we can check for consistency of disjoint classes. Two classes are disjoint if and only if they do not share any instance, meaning that an instance of a class cannot be instance of the other class. In OWO, if we want to check for the disjunction of two classes *HAWT* and *VAWT* we need to add an individual *(HAWT_Individual1)* for *HAWT*, after that add type *VAWT* for the individual. The result of adding the individual is shown in Fig. B.14. If *HAWT* and *VAWT* are disjoint classes, we will get an error message saying that the ontology is inconsistent.

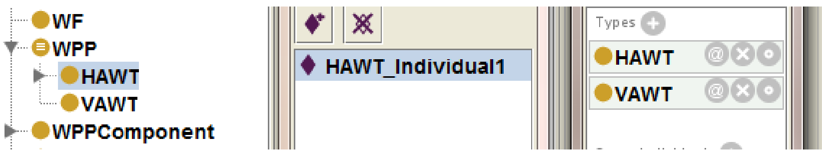


Figure B.14: Adding an individual for two disjoint classes HAWT and VAWT

3.4 Bridging the gap between ontologies and relational database

Relational databases and ontologies are similar because both of them are used to maintain models of some universe of discourse. However, there is a difference between them. While ontologies are very useful for knowledge representation, RDB are capable of efficiently managing large amounts of structured data. An advantage of ontologies is that they provide more support for inference in terms of finding answers about the model which had not been explicitly defined [3]. On the other hand, RDB have more mature technologies for storing and managing data. There is another difference between RDB and ontology based on basic assumptions [41].

Indeed, the RDB is based on a closed world assumption (CWA), whereas an ontology such as OWL is based on an open world assumption (OWA). For example, if the following axiom *WPP* $\sqsubseteq$*(=1hasWPPComponent.WYAW)* is not specified in the OWO ontology then the answer on the question whether or not *WPP* is subclass of $(= 1 hasWPPComponent.WYAW)$ would be *no* in CWA and *unknown* in OWA. The reason is that in OWA it is assumed that later more knowledge can be added to the knowledge base and the answer might be changed.

Owing to the specific characteristics of wind energy, such as dealing with data from different sources, with high frequency data to support continuous condition-based monitoring, and the need for storing large amounts of data, both ontology and RDB are needed in our work. Furthermore, there are many existing wind databases that need to be integrated in new developed systems in order to enhance the knowledge. Examples of existing databases are the one from Statkraft and Agder Energi that we have mentioned in Sect. 2.3. In order to make the developed OWO flexible, the gap between OWO and RDB as well as existing wind databases must be bridged.

3.4.1 Approaches to bridging the gap

There are three approaches to bridging the gap between RDB and ontologies. The first approach is to transform from existing RDB to ontologies. Work on this approach has been reported in [2, 4]. The second approach is to map between existing ontologies and existing RDB. In these approaches, transformations and mappings can be obtained either automatically or manually. The third approach is to transform from existing ontologies to RDB. Examples of implementing this approach are presented in [56, 51, 14]. However, there are shortcomings in these approaches. OWL is so expressive and some of its concepts cannot be represented in the relational model. For instance, it is not easy to manage the number of tables which are generated corresponding to classes or object properties in an ontology. Besides, a slight change of the ontology, for example, adding a new concept can lead to changing the structure of the RDB schema. In our case, the first and second approaches are excluded because we do not have a common RDB schema for different existing databases and the offshore wind domain. We then select the third approach. Our requirements for RDB transformation are:

1. The derived RDB must be simple enough, but it must cover all WPP component concepts (i.e. *WROT*, *WGEN*, and *WTRM*).
2. There is no need for an ontology metadata model since we do not have any

intention to translate the derived RDB back to OWO. An ontology metadata model is a model that contains information about the ontology such as disjoint classes, axioms, constraints.

We have tried to use existing work from [56, 51, 14] to derive a RDB schema from OWO. However, none of these work fulfils the aforementioned requirements. Ontology metadata tables are generated in the work [56] and [51]. For instance, we used the work in [51] to generate RDB schema. As the result, we got 32 ontology metadata tables along with tables that describe WPP. We do not have any intention to use those generated metadata tables. Gali et al. (2004) describe mapping techniques in their work, but there is no software available for using. We therefore develop a tool to transform from OWO to RDB. The transformation is described in the next sub section and followed the ideas described in [14].

3.4.2 Relational database transformation

In the transformation, we focus only on three elements: part-of relationships, is-a relationships, and data properties. Is-a and part-of relationships in OWO are translated into 1-1 (one-to-one) and 1-n (one-to-many) relationships in RDB, respectively. We also use annotations to specify which classes must be translated into tables. Our main algorithms for generating the RDB schema are the following algorithm 1, 2, and 3, where H stands for HashMap, L is a List, S is a Set, and NS is a NodeSet - a defined data type in OWL API. We use the OWL API to parse the ontology. Some built-in methods in the OWL API such as getDataPropertiesInSignature, getSubClasses, getDomains, getRanges are involved in our algorithm.

Algorithm for generating database schema

The algorithm 1 is used to add columns to a table. The algorithm takes an ontology, a reasoner, and name of the table (which is also a class in the ontology) as input. The algorithm first gets all the properties of the given class and save them to the given list. Then it finds all the related classes of the given class. Here, related classes of a given class are classes that have subclass-of or part-of relationships with the given class. For each related class, get all properties of the class and insert them into the list. Finally, the algorithm call a function to alter the given table by adding all the properties as columns.

The algorithm 2 is used to get all data properties of a given class. It takes an ontology, a class, and a list as input. All the properties will be inserted into the list which is the output of the algorithm. The algorithm first gets all data properties in

Algorithm 1 add Columns To A Table

Input: Ontology, an OWL reasoner, and a tablename

$H_{prop} \leftarrow \emptyset$
getAllPropertiesOfaClass(ontology,tblName,H_{prop})
$L_{re} \leftarrow \emptyset$
getAllRelatedClasses(L_{re})
for $i = 0$ to $L_{re}.length$ **do**
    getAllPropertiesOfaClass(ontology,$L_{re}[i]$,H_{prop})
end for
alterTable(tblName,H_{prop})

the ontology. For each data property, the algorithm retrieves all the domains and ranges of the data property. The algorithm inserts all the pairs of domain and range that satisfy the condition where the domain equals to the given class and there exists a range corresponding to the domain.

Algorithm 2 get All Properties Of a Class

Input: Ontology, a given class clz, list of properties H_{prop}

Output: List of all properties of the given class

$S_{prop} \leftarrow ontology.getDataPropertiesInSignature()$
for each $prop \in S_{prop}$ **do**
    $S_{clz} \leftarrow prop.getDomains(ontology)$
    for each $clzExp \in S_{clz}$ **do**
        if $clzExp = clz$ **then**
            $S_{range} \leftarrow prop.getRanges(ontology)$
            for each $range \in S_{range}$ **do**
                Add $(prop, range)$ to H_{prop}
            end for
        end if
    end for
end for

The algorithm 3 is used to get all related classes of a given class. It takes an ontology, a reasoner, a given class, and a list of classes as input. Output of the algorithm will be the list of classes.

The algorithm is executed through 3 steps:

1. Get all subclasses of the given class and store them in a list L.
2. Get all classes that have superclass relationships with the given class and insert them in the list L.
3. For each class in the list L, add the class to a list and apply the algorithm to

Algorithm 3 Get All Related Class Of a Class

Input: Ontology, a reasoner, a given class clz, a list of class L_{re}
Output: List of classes that are related to the given class L_{re}

$S_{ax} \leftarrow ontology$
$.getSubClassAxiomsForSubClass(clz)$
$L_{clz} \leftarrow \emptyset$
$S_{clz} \leftarrow reasoner.getSubClasses(clz, true)$
for each $clzz \in S_{ax}$ **do**
    if $clzz \neq OWLNothing()$ **then**
        Add $clzz$ to L_{clz}
    end if
end for
for each $ax \in S_{ax}$ **do**
    $clzExp \leftarrow ax.getSuperClass()$
    if $clzExpinstanceofOWLClassExpression$ **then**
        $S_{clz} \leftarrow clzExp.getClassesInSignature()$
        for each $class$ in S_{clz} **do**
            if $class \neq OWLNothing()$ **then**
                Add $class$ to L_{clz}
            end if
        end for
    end if
end for
for $i = 0$ to $L_{clz}.size$ **do**
    if $L_{clz}[i] \notin L_{re}$ **then**
        Add $L_{clz}[i]$ to L_{re}
    end if
    getAllRelatedClassOfaClass(ontology,reasoner, $L_{clz}[i], L_{re}$)
end for

the class.

3.4.3 Example of a generated database

As a result of implementing the algorithm on OWO, we get a derived RDB schema which is shown in Fig. B.15. There are two types of tables in the derived RDB. Tables that contains static information which is likely unchanged during the life time of equipments and only inserted when new equipments are purchases, for example blade length, rated wind speed. Tables that contains dynamic information such as analog state value information, descriptive information. These types of information are changing regularly, e.g. every 5 seconds.

B

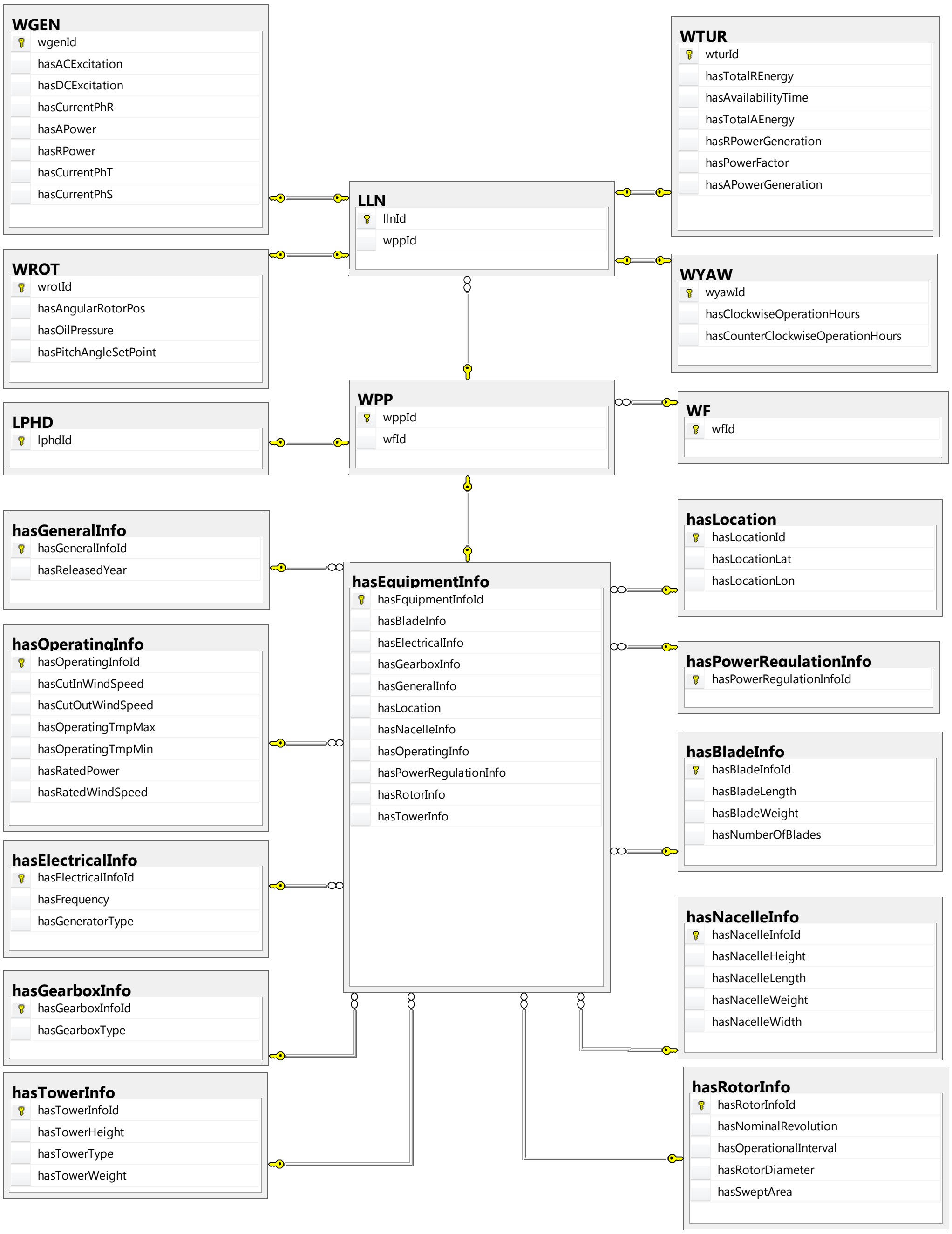


Figure B.15: An example diagram of the derived relational database

4 Implementation

This section shows how to use the developed OWO ontology in handling data integration. We develop a prototype which is used to integrate existing databases from Statkraft and Agder Energi that have been mentioned in Sect. 2.3.

4.1 System description

The prototype system is shown in Fig. B.16. We assume that wind data are made available through web services and it is pushed to our system over the Internet from data providers such as Statkraft and Agder Energi. MuleESB, an open source lightweight integration framework is used to handle data subscription from third parties. Details of implementing MuleESB are describe in [31]. SQWRL, Protégé-OWL API, Jess rule engine, pellet reasoner are used in order to answer users' queries against OWO. The WPP Data Storage is derived from OWO and has the structure shown in Fig. B.15. Data from the database is first loaded to OWO and then is used for answering queries from users. Data Transfer Object OWO (DTO OWO) contains Java classes that are generated from OWO. Data Transfer Object SQLServer (DTO SQLServer) contains Java classes that are generated from the derived RDB. Data Access Object (DAO) handles the access to the database through JDBC (Java Database Connectivity) interfaces. The mappings from DTO OWO to DTO SQLServer and vice versa are handled by Dozer. Wrappers are used to provide access to the Statkraft and Agder Energi databases.

The launching process of data from WPP Data Storage into OWO is described as follows:

1. Data in WPP Data Storage are accessed by calling built-in functions provided by DAO.

2. Data are then fetched into DTO SQLServer.

3. Dozer mapper converts data in DTO SQLServer to DTO OWO

4. Data are then fetched into OWO using Protégé OWL API.

The difference of our approach from other approaches is that we use the developed ontology to generate a RDB schema. While the ontology contains TBox, the RDB will be used to store ABox.

The mapping step is necessary since there is a difference in naming between third party's data sources and our system. The next subsection shows how to map data source from a third party to our derived RDB.

B

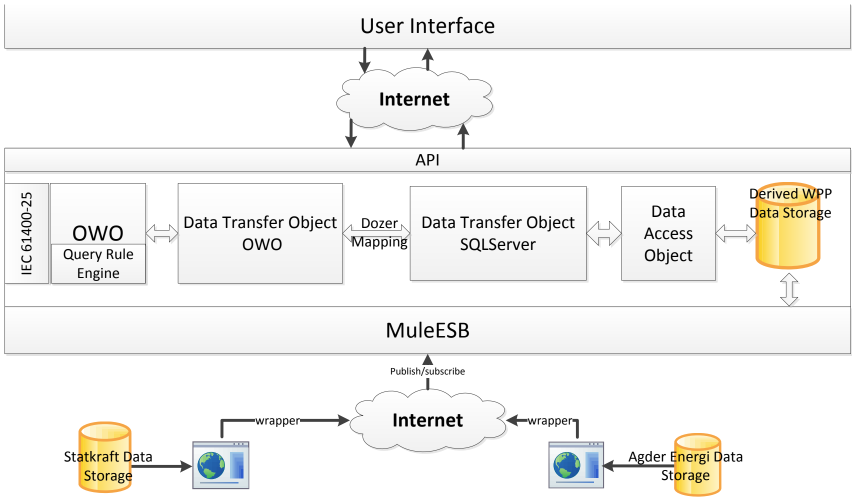


Figure B.16: A data integration prototype for offshore wind energy

4.2 Third parties' data mapping

We show a mapping from Statkraft database which has been shown in Fig. B.2 to the derived database presented in Fig. B.15. Table B.4 shows the mapping. In this work, we consider only 1 to 1 mapping. The first column shows data properties in OWO. The second columns shows columns' names that come from different tables in Statkraft database.

Table B.4: An example of mapping from Statkraft database to the derived database

OWO property	**Column in RDB**	**Data type**
WROT		
hasOilPressure	wtc_HubPresA_mean	real
hasPitchAngleSetPoint	wtc_PitcPosA_mean	real
hasPitchAngleRef	wtc_PitchRef_BladeA _mean	real
WGEN		
hasCurrentPhR	wtc_AmpPhR_mean	real
hasCurrentPhS	wtc_AmpPhS_mean	real
hasCurrentPhT	wtc_AmpPhT_mean	real
hasPh2PhVoltagePhR	wtc_VoltPhR_mean	real
hasPowerFactor	wtc_CosPhi_mean	real
WYAW		
hasOperationSpeed	wtc_GenRpm_mean	real

4.3 Querying OWO

As mentioned before, OWL ontology can be queried using SQWRL. Here, we present some example of SQWRL queries against OWO.

Example 1: A query about the list all wind farms with their name and total number of wind power plants can be present in SQWRL as follows:

B

$$WF(?p) \wedge hasTotalNumberWPP(?p,?number) \wedge hasWFName(?p,?name) \rightarrow sqwrl : select(?p,?name,?number) \wedge sqwrl : orderBy(?number)$$

Example 2: Get the oil pressure and pitch angle set point of the wind power plant which has ID is “2300249”.

$$WF(?p) \wedge hasID(?p,''2300249'') \wedge hasWPPComponent(?p,?comp) \wedge hasOilPressure(?comp,?pres) \wedge hasPitchAngleSetPoint(?comp,?pitchAngle) \rightarrow sqwrl : select(?p,?pres,?pitchAngle)$$

Example 3: Get all the wind power plants that have the latest inspection date is “Feb-01-2012”

$$WF(?p) \wedge hasName(?p,?name) \wedge hasWPP(?p,?wpp) \wedge hasID(?p,?id) \wedge hasLastInspectionDate(?wpp,''132805440'') \rightarrow sqwrl : select(?p,?name,?id)$$

Example 4: SQL query described in Sect. 2.3 can be rewritten in SQWRL as follows:

$$WF(?p) \wedge hasStationId(?p,''2300249'') \wedge hasAPower(?p,?APower) \wedge hasGeneratorSpeed(?p,?GenSpeed) \wedge hasTimeStamp(?p,''2012-01-02T00:00:00'') \rightarrow sqwrl : select(?p,?APower,?GenSpeed)$$

5 Discussion and future work

The main challenge we faced during the development of OWO is to collect the domain concepts from existing wind databases that are in used by different companies. The reason is that those companies are very reluctant to share their data or give very limited access to the data. We realize that the IEC 61400-25 standard is a good source for domain concepts. The idea that separates the data structure of WPP model proposed in the standard makes data model of WPP clear and easy to follow. However, we encountered some problems while implementing the standard such as the standard does not cover all the concepts of the wind domain in general and the offshore wind domain in particular. For instance, foundation monitoring of an offshore WPP is important since offshore WPP is installed in places far away from shore and perhaps in deep water in the future. Besides, data persistent is out of scope of the standard. We therefore have tried to overcome these drawbacks by introducing an ontology model based on the standard and described a way to bridging the gaps between ontologies and relational databases. For each existing database connected to our system, we need to develop a specific wrapper.

The proposed ontology can be extended for describing faults and their symptoms. A RAMSI (reliability, availability, safety, and inspectability) database can be derived from the ontology using our proposed algorithms. The purpose of having a RAMSI database is to provide access to reliability data, including operation, maintenance, failure, installation data.

At the moment, data from RDB is fully loaded into OWO no matter whether part of the data or all the data will be used for answering queries. It is therefore meaningful to optimize the system by the following approaches:

1. Only load data which is needed for reasoning over the ontology. This approach involves a work on SQWRL query optimization and part of SQWRL to SQL transformation.

2. The second approach is to use the ontology only for building SQWRL queries. These queries are then translated into SQL queries and applied against RDB. The transformation from SQWRL to SQL can be made using an intermediate layer such as domain calculus.

6 Conclusions

In this work, we introduced OWO, an offshore wind ontology model based on the IEC 61400-25 standard. OWO provides a semantic way to manage offshore wind metadata. OWO can be used to support data integration in term of facilitating knowledge sharing and data exchange between offshore wind partners, and hence improved offshore wind operations. We also showed how to bridge the semantic gaps between the developed ontology and RDB by deriving a RDB schema from OWO. The RDB compensates the shortcoming of ontology in storing data. We also developed a prototype system to demonstrate how to make query against OWO and how to integrate a third party's data source into our system.

Besides the results we have achieved, our work shows that the semantic technologies are mature enough to use in managing offshore wind metadata. It provides possibility of organizing and extracting semantics of data, and querying over the knowledge base. The developed ontology can be extended and connected to semantic sensor network ontologies. Thus integration of offshore wind into smart grid becomes easier.

Acknowledgment

This work has been funded in part by the Norwegian Centre for Offshore Wind Energy (NORCOWE) under grant 193821/S60 from the Research Council of Norway (RCN). NORCOWE is a consortium with partners from industry and science, hosted by the Christian Michelsen Research. This research is also supported by the EU 7th Framework Programme FP7-PEOPLE-IRSES under grant agreement no. 247083, project acronym S2EuNet. Authors thank to Statkraft and Agder Energi for providing data. Work had done while visiting Georgia Institute of Technology.

References

[1] Antoniou, G., Harmelen, F.: Web ontology language: OWL. Handbook on ontologies pp. 91–110 (2009)

[2] Astrova, I.: Rules for mapping SQL relational databases to owl ontologies. Metadata and Semantics pp. 415–424 (2009)

[3] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: The description logic handbook: theory, implementation and applications. Cambridge university press (2003)

[4] Barrasa, J., Corcho, Ó., Gómez-Pérez, A.: R2O, an extensible and semantically based database-to-ontology mapping language (2004)

[5] Batres, R., West, M., Leal, D., Price, D., Masaki, K., Shimada, Y., Fuchino, T., Naka, Y.: An upper ontology based on ISO 15926. Computers & Chemical Engineering **31**(5), 519–534 (2007)

B

[6] Bechhofer, S., Van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P., Stein, L., et al.: OWL web ontology language reference. W3C recommendation **10**, 2006–01 (2004)

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

[8] Bitar, E., Rajagopal, R., Khargonekar, P., Poolla, K., Varaiya, P.: Bringing wind energy to market. Power Systems, IEEE Transactions on **27**(3), 1225–1235 (2012)

[9] Botterud, A., Wang, J., Bessa, R., Keko, H., Miranda, V.: Risk management and optimal bidding for a wind power producer. In: Proceedings of the IEEE PES General Meeting 2010 (2010)

[10] Chen, Z., Guerrero, J., Blaabjerg, F.: A review of the state of the art of power electronics for wind turbines. Power Electronics, IEEE Transactions on **24**(8), 1859–1875 (2009)

[11] Dieng-Kuntz, R., Minier, D., Ruzicka, M., Corby, F., Corby, O., Alamarguy, L.: Building and using a medical ontology for knowledge management and cooperative work in a health care network. Computers in Biology and Medicine **36**(7), 871–892 (2006)

[12] El-Thalji, I., Liyanage, J.P.: Managing offshore wind energy assets: On the systematic development of an integrated architecture. In: Engineering Asset Management and Infrastructure Sustainability, pp. 181–193. Springer (2012)

[13] EWEA: EU Energy Policy to 2050 (2011)

[14] Gali, A., Chen, C., Claypool, K., Uceda-Sosa, R.: From ontology to relational databases. Conceptual Modeling for Advanced Application Domains pp. 278–289 (2004)

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

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

[17] Horrocks, I., Patel-Schneider, P., Van Harmelen, F.: 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 (2003)

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

[19] Kohrsen, T.: Deployment of IEC61400-25-4 - Mapping to Web services on an industrial PLC platform. Master's thesis, Technical University of Denmark (2008)

[20] Kollia, I., Glimm, B., Horrocks, I.: SPARQL query answering over OWL ontologies. In: The Semantic Web: Research and Applications, pp. 382–396. Springer (2011)

[21] Kumar, A., Smith, B., Pisanelli, D., Gangemi, A., Stefanelli, M., et al.: An ontological framework for the implementation of clinical guidelines in health care organizations. Studies in health technology and informatics pp. 95–107 (2004)

[22] Kusiak, A., Li, W., Song, Z.: Dynamic control of wind turbines. Renewable energy **35**(2), 456–463 (2010)

[23] Kusiak, A., Zheng, H., Song, Z.: Power optimization of wind turbines with data mining and evolutionary computation. Renewable Energy **35**(3), 695–702 (2010)

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

[25] Liu, H., Chen, Z.: Aggregated modelling for wind farms for power system transient stability studies. In: Power and Energy Engineering Conference (APPEEC), 2012 Asia-Pacific, pp. 1–6. IEEE (2012)

[26] Martinez, F., Herrero, L.C., de Pablo, S.: Open loop wind turbine emulator, *Renewable Energy*, **63**, pp.212–221.

[27] Morales, J.M., Pinson, P., Madsen, H.: A transmission-cost-based model to estimate the amount of market-integrable wind resources. Power Systems, IEEE Transactions on **27**(2), 1060–1069 (2012)

[28] Motik, B., Patel-Schneider, P., Parsia, B., Bock, C., Fokoue, A., Haase, P., Hoekstra, R., Horrocks, I., Ruttenberg, A., Sattler, U., et al.: OWL 2 web on-

tology language: Structural specification and functional-style syntax. W3C recommendation **27**, 17 (2009)

[29] Naphade, M., Smith, J., Tesic, J., Chang, S., Hsu, W., Kennedy, L., Hauptmann, A., Curtis, J.: Large-scale concept ontology for multimedia. Multimedia, IEEE **13**(3), 86–91 (2006)

[30] Nguyen, T., Prinz, A., Friisø, T., Nossum, R.: Smart grid for offshore wind farms: Towards an information model based on the IEC 61400-25 standard. In: Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES, pp. 1–6. IEEE (2012)

[31] Nguyen, T., Rasta, K., Trinugroho, D., Prinz, A.: Using enterprise service bus for offshore wind farm data handling. In: IADIS Applied Computing (2012)

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

[33] O'Connor, M., Das, A.: SQWRL: a Query Language for OWL. In: Proc. of 6th OWL: Experiences and Directions Workshop (OWLED2009) (2009)

[34] Olsen, A., Ösdil, B., Poulsen, B., Ole, K., Pedersen, H., et al.: Prototype of generic server for wind power plants using IEC 61400-25 standard (2006)

[35] Papadopoulos, P., Cipcigan, L.: Wind turbines' condition monitoring: an ontology model. In: Sustainable Power Generation and Supply, 2009. SUPERGEN'09. International Conference on, pp. 1–4. IEEE (2009)

[36] Ragheb, A., Ragheb, M.: Wind turbine gearbox technologies. In: Nuclear & Renewable Energy Conference (INREC), 2010 1st International, pp. 1–8. IEEE (2010)

[37] Ragheb, M.: Control of Wind Turbines. URL: https://netfiles. uiuc. edu/mragheb/www/NPRE **20475** (2008)

[38] Righter, R.: Wind energy in America: A history. University of Oklahoma Press (1996)

[39] Schlechtingen, M., Ferreira Santos, I.: Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. Mechanical systems and signal processing **25**(5), 1849–1875 (2011)

[40] Schwarz, K.: Monitoring and control of power systems and communication infrastructures based on IEC 61850 and IEC 61400-25. In: 2010 DistribuTech conference & Exposition (2010)

[41] Sequeda, J., Tirmizi, S., Corcho, O., Miranker, D.: Survey of directly mapping SQL databases to the semantic web. The Knowledge Engineering Review **26**(04), 445–486 (2011)

[42] Sheth, A., Henson, C., Sahoo, S.S.: Semantic sensor web. Internet Computing, IEEE **12**(4), 78–83 (2008)

[43] Sirin, E., Parsia, B.: SPARQL-DL: SPARQL Query for OWL-DL. In: OWLED, vol. 258 (2007)

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

[45] Stuckenschmidt, H., and Van Harmelen, F.:Information sharing on the semantic web, Heidelberg: Springer.

[46] Studer, R., Benjamins, V., Fensel, D.: Knowledge engineering: principles and methods. Data & knowledge engineering **25**(1), 161–197 (1998)

[47] Sørensen, P.E., Hansen, A.D., Iov, F., Blaabjerg, F., Donovan, M.: Wind farm models and control strategies, Risø, Tech. Rep.

[48] Thorsen, K.A.H.: Semantic data interoperability and its applications, *Ph.D. thesis*, University of Stavanger.

[49] Thorsen, K.A.H, Rong, C.: Data Integration in Oil and Gas at Norwegian Continental Shelf, *Advanced Information Networking and Applications-Workshops, AINAW 2008, 22nd International Conference on*, pp.1597–1602. IEEE.

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

[51] Vysniauskas, E., Nemuraite, L.: Transforming ontology representation from OWL to relational database. Information Technology and Control **35**(3A), 333–343 (2006)

[52] 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

B

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. Schlömer , C. von Stechow (eds)

[53] Yeh, I., Karp, P., Noy, N., Altman, R.: Knowledge acquisition, consistency checking and concurrency control for Gene Ontology (GO). Bioinformatics **19**(2), 241–248 (2003)

[54] Zaher, A., McArthur, S.: A multi-agent fault detection system for wind turbine defect recognition and diagnosis. In: Power Tech, 2007 IEEE Lausanne, pp. 22–27. IEEE (2007)

[55] Zaher, A., McArthur, S., Infield, D., Patel, Y.: Online wind turbine fault detection through automated SCADA data analysis. Wind Energy **12**(6), 574–593 (2009)

[56] Zhang, H., Wang, Z., Gao, Z., Li, W.: Design and implementation of mapping rules from OWL to relational database. In: Computer Science and Information Engineering, 2009 WRI World Congress on, vol. 4, pp. 71–75. IEEE (2009)

[57] Zhu, Y.L., ying Wang, X., ling Cheng, D.: Ontology-based research on wind power plant information interaction. In: Knowledge Discovery and Data Mining, 2008. WKDD 2008. First International Workshop on, pp. 169–172 (2008).

Appendix C

Paper C

Title	A Semantic-enhanced Quality-based Approach to Handling Data Sources in Enterprise Service Bus
Authors	Trinh Hoang Nguyen[a], Kamyar Rasta[a], Dafferianto Trinugroho[a], and Andreas Prinz[a]
Affiliation(s)	[a] Department of ICT, University of Agder, Grimstad, Norway
Journal	International Journal of Computer Science & Applications
Status	In Press, Accepted: March 20 2014
Copyright	©Technomathematics Research Foundation

A Semantic-Enhanced Quality-based Approach to Handling Data Sources in Enterprise Service Bus[1]

Trinh Hoang Nguyen[a,*], Kamyar Rasta[a,**], Dafferianto Trinugroho[a,*], Andreas Prinz[a,*]
[a] Department of ICT, University of Agder, Grimstad, Norway
E-mails: *{trinh.h.nguyen, dafferianto.trinugroho, andreas.prinz}@uia.no
** kamyar11@student.uia.no

C

Data quality plays an important role in success of organizations. Poor data quality might significantly affect organizations' businesses since wrong decisions can be made based on data with poor quality. It is therefore necessary to make data quality information available to data users and allow them to select data sources based on their given requirements. Enterprise Service Bus (ESB) can be used to tackle data integration issues. However, data sources are maintained out of the ESB's control. This leads to a problem faced by users when it comes to selecting the most suitable data source among available ones. In this article, we present an approach to handling data sources in ESB based on data-quality and semantic technology. This introduces a new level of abstraction that can improve the process of data quality handling with the help of semantic technologies. We evaluate our work using three different scenarios within the wind energy domain.

Keywords: Quality-based; semantic-enhanced; data source handling.

1 Introduction

Service-oriented architecture (SOA) has changed the way of designing software. SOA provides the possibility of integrating and distributing services in a loosely-coupled manner [46, 4]. There are several SOA topologies such as peer-to-peer network, hub & spoke, pipeline, and enterprise service bus [8]. Among them, Enterprise Service Bus (ESB) is gaining more and more attention from industries. The idea of the ESB concept is that all the applications that are connected to a bus can share, produce, and consume information on the bus [7].

ESB is an emerging technology derived from the combination of SOA and event-driven architecture (EDA) [25]. EDA means that any event that happens inside or outside a business disseminates immediately to all interested parties that subscribe

[1]This article is a revised and expanded version of a paper entitled, A Framework for Data Quality Handling in Enterprise Service Bus, presented at INTECH 2013 conference, 29 - 31 August, London, UK. ISBN: 978-1-4799-0047-3

to the event [26]. An example of this could be using a publish/subscribe mechanism in EDA to enable real-time monitoring of offshore wind farms. Whenever new data arrives, it is immediately published to a channel and notifications about it will be sent to subscribers instead of letting subscribers query for information every once in a while.

ESB provides a number of benefits such as loosely coupled architecture (i.e. whatever changed in an application will not affect other applications in the whole system), increased flexibility (i.e. easier to change as requirements change, standardized platform for integration), sharing common services (security, error management, reporting, etc.), and supporting a large number of communication patterns over different transport protocols [34]. ESB has been used in different areas such as power systems [30], eHealth [41], and oil & gas [32].

Although ESB handles communication between applications, it does not support a way to select the most suitable data source among several available ones. Assume that a user requests for wind speed in the North Sea for the last few months. The user wants to have data with accuracy and completeness in a certain interval. There are probably several available sources of wind speed data in the North Sea. How can we provide the most suitable data source to the user based on the data quality criteria? How can a system respond to the user's request if the requested data source is not available or the user's requirements are not met? Is there any way to combine the available data sources to produce an improved data source? Fig. C.1 depicts an overview of the problem that we are targeting in this work.

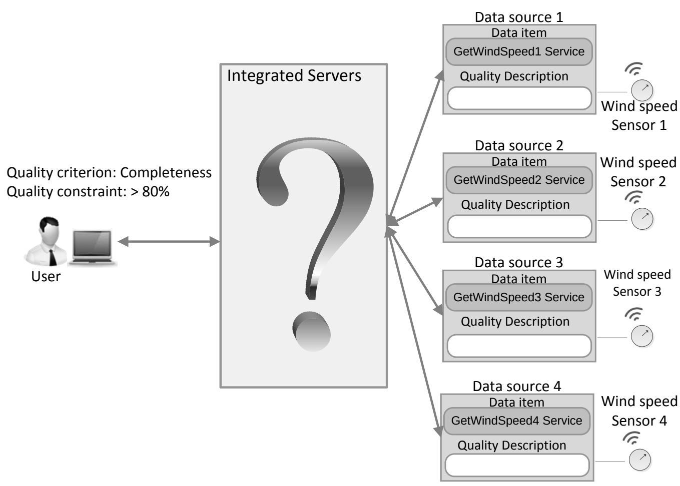


Figure C.1: An abstract view of the problem

This article answers the stated questions by proposing a quality-based approach to handling resources in ESB with semantic technologies. The rest of the article is organized as follows. Section 2 presents the main concepts of data sources and challenges in data source handling. Section 3 discusses data quality and its dimensions. Section 4 describes our proposed approach to tackle challenges in handling data sources. The implementation of the proposed approach is presented in Sect. 5 along with the details of the system components and how they perform the desired functionalities. Related work and discussion are presented in Sect. 6. Finally, section 7 concludes the paper.

2 Data source

This section explains the definition of data source in the context of our work. Challenges in handling data sources are discussed afterwards.

2.1 Data source definition

Data sources are sensor sources, web services that provide data generated by sensors or other appliances, and existing databases. A data source consists of three components: data stream, data source description, and quality description. Fig. C.2 illustrates the components of a data source.

- *Data stream* is a stream of signals coming from a source over time, while a data set is a set of data points coming from a static source, e.g., existing relational databases. Each signal in the stream is a data point, for instance, 6 (m/s) for wind speed at a specific date and time. An example of data stream is a set of 4000 wind speed records from the 1^{st} of May to the 1^{st} of June. Data streams are provided by data providers or by having direct connection to sensors. Data providers can make the data accessible via services.
- *Data source description* contains information about measurement type, measurement unit, data exchange format. For instance, the description of a data source could be: (1) measurement type: wind speed, (2) unit: m/s, and (3) data exchange format: XML.
- *Quality description* is a description of the quality of the data source. For example, the quality description of a data source states that the completeness is 80% and the timeliness is $1ms$.

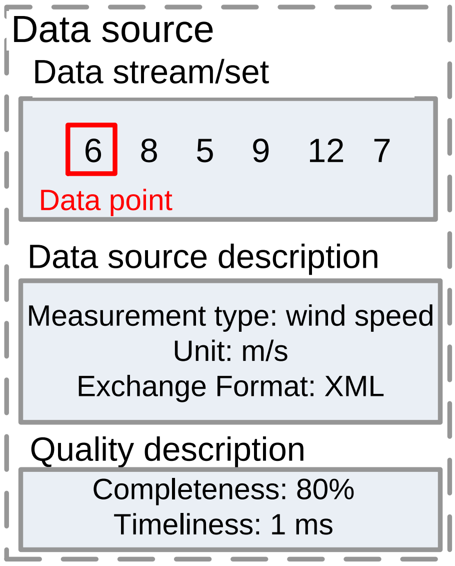


Figure C.2: Components of a data source

2.2 Type of data sources

In this work, data sources are classified into *real* and *virtual* data sources. *Real* data sources are sensors or services that have access to sensors' measurements. Data quality of these data sources can be obtained either from devices' manufacturers or by computation using reference data sources. A *reference* (also known as benchmark in some literature [14]) is a data source that can be either a previous measurement of the same data source or a mathematical model [24, 9] of that data. Reference data source are considered as real data sources.

In contrast, *virtual* data sources have no direct connection to sensors. These data sources are computed by combining more than one *real* data source. It is possible to combine data sources which measure different physical phenomena provided that there is a mathematical relationship between these physical phenomena. If the data sources provide data in different units (Fahrenheit and Celsius), a unit converter will be employed before combining.

Since virtual data sources are generated by combining real data sources, they also consist of three parts: data stream, data source description, and quality description. The virtual data stream is described by a formula, e.g., average of data streams from the real data sources. The quality description of the virtual data source is computed by combining the quality descriptions of the real data sources.

2.3 Challenges in data source handling

There are many challenges in handling data sources in ESB related to data quality, service selection, and data source integration. Here, we address those challenges that we try to overcome in this work.

2.3.1 Availability of data quality description

Typically, data are provided without any quality description attached to them. It is not clear what the accuracy or completeness of the data is. There are also cases where data quality is available but the service that makes data accessible does not provide any method to access the information. A way to compute data quality and make it available is therefore important.

A widespread issue in data integration is the management of data with insufficient quality. For example in offshore wind energy, a couple of sensors are deployed on a windmill and they frequently measure and deliver the data to the users and applications by means of services. As sensors are prone to failures, their results might be inaccurate, incomplete, and inconsistent [37]. Therefore, the data quality issues should be handled in such a way that users and applications can specify the desired quality level of the data. Only when the data source has the requested quality descriptions it would be used for further processing.

2.3.2 Data source selection based upon users' requests

Assume a user can access data sources with data quality information available. How can a system fulfill the user's requests for data with given constraints on data quality? The user only cares about the requested data and its quality and does not care from which data source the data is selected. Thus, answering the request by giving a list of possible data sources is not a good answer. The aforementioned question can be formulated as follows: how to select the best suited data source among available data sources based upon user's defined quality criteria?

2.3.3 Data source combination

Another issue is that sometimes none of the available data sources has the required quality. In this case, it might be possible to improve the quality of data to meet the user's requirement by combining existing data sources. A virtual data source is the result of the combination.

2.3.4 Semantic inconsistency of data sources in sensor networks

Sensors are becoming one of the main data sources since they are used intensively in many areas such as oil & gas, eHealth, smart grid, and smart cities. However, sensor data sources are described differently by their manufacturers. This leads to a semantic inconsistency issue when it comes to integrating different sensors into a system. It therefore makes the process of data source selection and combination more difficult.

2.4 Approaches to selecting data sources

There are several ways of selecting data sources such as content-based filtering [11], social information filtering [39], agent-based selection [40], and quality-based selection [29].

Content-based filtering is a traditional and static way of selecting a data source out of a list of available data sources. This approach filters data sources based on users' keywords. When users send requests for data, the content-based filtering statically selects the data source with more relevant description. This approach might not solve the selection problem if there are data sources with the same descriptions.

Another approach is the social information filtering. It refers to a sort of techniques to provide personalized recommendations for users according to the similarities of their interests. This approach is common in sites, e.g., Amazon and LinkedIn.

The third category of selection methods is the agent-based approach. Agents evaluate data sources by communicating, cooperating, and rating each other. Each agent can make decision and work autonomously as well.

The quality-based approach takes into consideration the importance of data quality. A data source is selected based on data quality dimensions given in users' defined requirements. Our work is based on this approach. We also use semantic technology to solve inconsistency issues and enable semantic description for sensor networks.

3 Data quality

Data quality plays an important role in the success of organizations [44]. Poor data quality might significantly affect organizations' businesses since wrong decisions can be made based on data with poor quality [38, 17, 14].

In this section we define the basic terms and concepts in the article. Although there is no consensus on the definition of data quality [20], the data quality concept is defined as the extent to which attributes of data are suitable for their use. It is usually evaluated from a consumer point of view [17].

3.1 Data quality dimensions

Data quality is a multidimensional notion. There are more than 17 data quality dimensions which have been mentioned in the literature, e.g., accuracy, consistency, confidence, interpretability, completeness, relevancy, timeliness. Some of the studies such as [43, 23, 12] support several quality dimensions. These work often address data quality frameworks that encompass a rich set of data quality dimensions. On the other hand, studies like [13, 2, 15] only cover a small number of dimensions because they are applying the data quality concept to a specific application domain.

3.2 Selected data quality dimensions

In this work, we select the most commonly used quality dimensions in the literature, i.e. *accuracy*, *completeness*, and *timeliness* to demonstrate the proposed approach. These data quality dimensions are defined differently in the literature. Here we define the terminologies based on the existing definitions and our understanding. Table C.1 shows the notation that we use in the following definitions.

Accuracy is defined as how close the observed data are to reality. According to ISO 5725 standard [18], accuracy consists of precision and trueness.

- *Precision* is the closeness of agreement within individual results.
- *Trueness* is defined as the mean value of the difference of data source to the reality.

We assume that the sensors are calibrated, meaning that the trueness is very close to zero. Therefore, we only consider precision as the accuracy in our system. A

C

Table C.1: Table of notation

Symbol	**Explanation**
D	Data source
R	Reference data source (reality)
N_D	total number of data points in D
N_R	total number of data points in R
$N_{D_{cons}}$	total number of consistent data points in D
d_i	a single data point in D
r_i	real value corresponding to d_i
x_i	d_i - r_i
$t(r_i)$	the moment when the data point i is due
$t(d_i)$	the moment when the data point i is available

statistical measure of the precision for a series of repetitive measurements is the standard deviation.

Let x_i denote the difference between the measured data (d_i) and the reality (r_i) and μ denote the trueness. Thus, the accuracy of data source D can be obtained using Eq. (C.1).

$$Acc(D) = \sqrt{\frac{1}{N_D}\sum_{i=1}^{N_D}(x_i - \mu)^2} \tag{C.1}$$

Given $\mu = 0$ and $x_i = d_i - r_i$, Eq. (C.1) can be rewritten as follows.

$$Acc(D) = \sqrt{\frac{1}{N_D}\sum_{i=1}^{N_D}(d_i - r_i)^2} \tag{C.2}$$

Completeness is defined as the ratio of the number of successful received data points to the number of expected data points. The completeness of the data source D can be calculated using Eq. (C.3):

$$Compl(D) = \frac{N_D}{N_R} \tag{C.3}$$

For example, if a user requests for wind speed from a data source and he expects to get 60 data points in 1 hour. If the user has received only 40 data points in 1 hour, the completeness of the wind data source is 67%.

Timeliness is the average time difference between the moment a data point has been successfully received and the moment it is expected to be received. The timeliness of data source D is calculated using Eq. (C.4):

$$Time(D) = \frac{\sum_{i=1}^{N_D}(t(d_i) - t(r_i))}{N_D} \tag{C.4}$$

4 Semantic-enhanced quality-based data source handling

This section describes the proposed approach to handling data sources. By handling we mean that the approach offers ways to manage data sources, to insert a new data source, and to provide the best suited data source to users. An overview of our approach is illustrated in Fig. C.3.

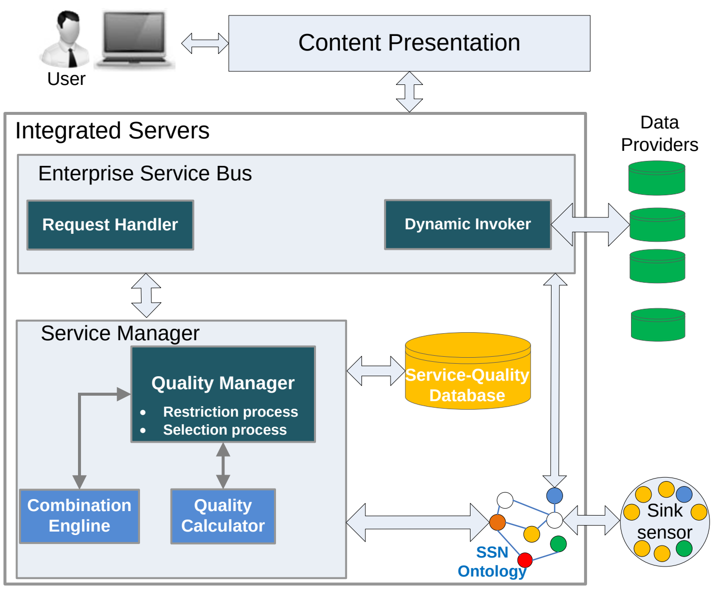


Figure C.3: Overview of the proposed approach

The *Request Handler* is an ESB-dependent module. If the ESB platform is changed, this module needs to be changed. It is used to parse and analyze requests from users. The *Dynamic Invoker* is responsible for invoking web services that describe data sources.

The *Service Manager* is in charge of finding proper data sources. It consists of three sub modules: quality manager, combination engine and quality calculator.

- The *Quality Manager* is used to select the most appropriate data sources in terms of data quality dimensions and constraints from the *Service-quality database*. The *Quality Manager* uses two processes to find the best data source for the user: restriction process and selection process. The purpose of the restriction process is to detect and filter out outliers from the results according to the given quality dimensions and constraints. The restriction process prepares a list of available *good* data sources and hands in this list to the selection process. The basic idea of selection is to find the *best* data source among *good* data sources according to selection quality dimension. If there are more than one data sources that meets users' quality requirements, the *Quality Manager* selects one of the *good* data sources randomly.

- The *Combination Engine* is used to combine existing data sources in order to generate virtual data sources. The module is called when there is no data source that meets the users' defined quality requirements. The combination process includes combination of data points, data source descriptions, and quality descriptions. Section 5.2.2 describes an example of using the *Combination Engine*.

- The *Quality Calculator* is used in case the data quality description of a data source is not available. The computation is done using equations mentioned in Sect. 3.2. After computing the quality dimension the quality calculator can store the information in the *Service-quality Database*. An example of using the *Quality Calculator* is shown in Sect. 5.2.3.

The *Quality Manager* generates a list of suitable data sources. The list contains information such as the data source address, parameters, and values. In case the *Combination Engine* is involved, the list will also contain information about a combination method. *Service Manager* passes the list to *Dynamic Invoker*. *Dynamic Invoker* then access data sources by invoking web services dynamically.

The *Service-quality Database* stores metadata of all available services and the *Semantic Sensor Network Ontology* specifies metadata of sensors such as location, type of sensor, unit, and sensors' quality. Details of the database and the ontology are presented in the next section.

4.1 Integration & management of data sources

Data can come directly from sensors or from services made available by service providers. Hence, data sources can be sensors or services. Metadata of (web) services are stored in a xs while metadata of sensors are described and stored in a semantic sensor network ontology. Metadata of real-time data can be accessed through SSN and metadata of historical data can be accessed through relational database.

4.1.1 The Service-quality Database

The *Service-quality Database* is a relational database that stores metadata of all available services. It does not store measurement data. The database is divided into two parts. The first part contains information about services and their descriptions. The second part stores information about data quality of the services. Based on input from users, the *Quality Manager* uses the database to select the most suitable data source. The result of a query is typically a data source or a list of available data sources. The result is then delivered to the *Dynamic Invoker* service for subsequent binding invocation.

4.1.2 A semantic sensor network ontology

Sensors are being used intensively in different systems. This creates vast amount of data. However, a big portion of data cannot be transformed to knowledge due to the lack of integration and communication between sensor networks. In order to tackle this issue, the W3C Semantic Sensor Network Incubator group took the initiative in enabling SSN by proposing an SSN ontology [6]. The semantic sensor network (SSN) was introduced based on Sensor Web Enablement (SWE) standards proposed by the Open Geospatial Consortium (OGC) [36] and the Stimulus-Sensor-Observation ontology design pattern [19]. Based on the SSN ontology, a number of developments have been reported in several work such as [31, 5]. In this section, we use SSN ontology to overcome the challenge posed in Sect. 2.3.4.

Even though accuracy has been mentioned in the SSN ontology, data quality is not only about accuracy. Many work have reported that data quality should be defined beyond accuracy [17]. We therefore have extended the SSN ontology by adding some quality dimensions that are described in Sect. 3.2.

The developed ontology contains spatial attributes (i.e. information about sensors' location), temporal attributes (i.e. information about timestamp), thematic attributes (i.e. information about sensor type, measurement, units), and quality attributes (e.g. accuracy, timeliness, completeness). Fig. C.4 depicts a partial view of the developed ontology. The ontology describes two type of sensors: temperature sensor and wind speed sensor. Each sensor's value is classified in different level, e.g., high or low. The quality of measurements is defined by the *MeasurementProperty* class.

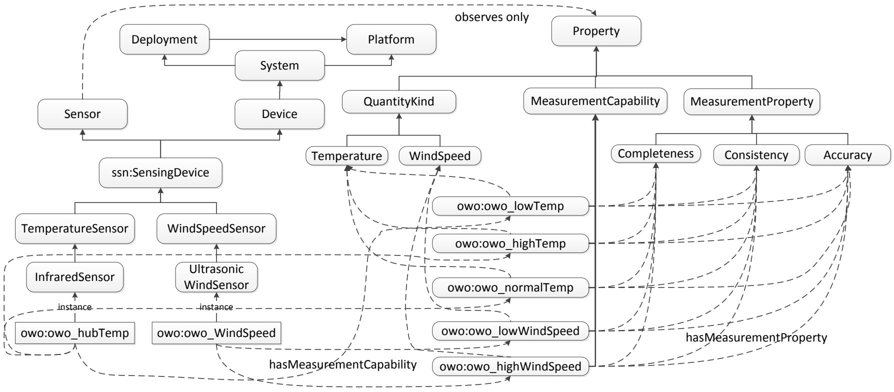


Figure C.4: A partial view of the quality-enhanced SSN ontology

4.2 Data source selection

The data source selection answers the question of how to select the most suitable data source from available data sources. This section presents our solution to overcome the challenge posed in Sect 2.3.2. Based on a user request, the selection process requires a set of quality constraints and a selection dimension. The mandatory constraints describe conditions to be met by the data sources, and the optional selection dimension describes which dimension to use for finding the best data source. A scenario described in Sect. 5.2.3 explains our point.

What if the requested data source is not available? The next section discusses the combination methods that are used to tackle the problem.

4.3 Data source combination

In order to improve the data quality to meet users' requirements, it is possible to combine different data sources. This section presents a solution to the challenge mentioned in Sect. 2.3.3. For simplicity, we consider only those data sources that measure the same physical phenomena. Let us consider a case where we need to combine two data sources D1 and D2. There are several methods to combine data sources, but looking deeply into combination methods is out of scope of this work. Let us examine three simple combination methods as follows.

- D1 (A) D2: taking a conventional average of the data sources D1 and D2.
- D1 $\bigoplus$ D2: use data points from data source D1 if available, otherwise use D2.
- D1 (E) D2: pick up the earliest received data point from either D1 or D2.

Assume that data points in data sources $D1$ and $D2$ are independent normally distributed random variables. Let $Acc(D1)$ and $Acc(D2)$ be the accuracy (precision) of $D1$ and $D2$ respectively. Let P(D1) denote the probability of the event *"D1 having data available"* and P(D2) denote the probability of the event *"D2 having data available"*. These two events are independent. Let $\overline{P(X)} = 1 - P(X)$ denote complementary of the event *"data source X having data available"*.

We also assume that the timeliness $Time(D1)$ and $Time(D2)$ of $D1$ and $D2$ are two independent exponentially distributed random variables as shown in Eq. C.5. That said, $Time(D1) = \frac{1}{\lambda_1}$ and $Time(D2) = \frac{1}{\lambda_2}$.

$$f(t, \lambda) = \begin{cases} \lambda e^{-\lambda t}, & \text{if } t \geq 0 \\ 0, & \text{if } t < 0 \end{cases} \tag{C.5}$$

where λ is a rate parameter.

4.3.1 D1 (A) D2

Completeness: The completeness of the virtual data source is calculated by Eq. (C.6). This is also the result for the cases D1 $\bigoplus$ D2 and D1 (E) D2. In three cases, the value of the completeness of the virtual data source is improved.

$$Compl(D1\ (A)\ D2) = \overline{\overline{P(D1)}\ .\ \overline{P(D2)}} \tag{C.6}$$

Accuracy: Since Acc(D1) and Acc(D2) are normally distributed, sum of them is also normally distributed. As we mentioned in Sect. 3.2, the sensors are calibrated and thus the trueness is equal to zero. The accuracy is then equal to the precision which can be obtained by Eq. (C.7):

$$Acc(D1\ (A)\ D2) = \sqrt{\frac{Acc(D1)^2 + Acc(D2)^2}{4}} \tag{C.7}$$

Timeliness: The timeliness of the virtual data source is the maximum value of the set which consists of $Time(D1)$ and $Time(D2)$, i.e. $max(Time(D1), Time(D2))$ which is not exponential. However, if $Time(D1)$ and $Time(D2)$ are identical and have rate parameter λ, we can easily obtain the probability density function (PDF) of the maximum as follows:

$$f_{tmax}(t; \lambda) = 2\lambda e^{-\lambda t}(1 - e^{-\lambda t}),\ t \geq 0 \tag{C.8}$$

The expected value is obtained via Eq. (C.9).

$$Time[D1\ (A)\ D2] = E[tmax] = \int_0^\infty t f_{tmax}(t; \lambda) dt = \frac{3}{2\lambda} \approx \frac{3}{2} Time(D1) \tag{C.9}$$

4.3.2 D1 $\bigoplus$ D2

Accuracy: The accuracy of virtual data source is calculated using weighted average as shown in Eq. (C.10):

$$Acc(D1 \bigoplus D2) = \frac{P(D1) * Acc(D1) + \overline{P(D1)} * P(D2) * Acc(D2)}{P(D1) + \overline{P(D1)} * P(D2)} \tag{C.10}$$

Timeliness: The timeliness of the virtual data source is calculated using Eq. (C.11):

$$Time(D1 \bigoplus D2) = \frac{P(D1) * Time(D1) + \overline{P(D1)} * P(D2) * Time(D2)}{P(D1) + \overline{P(D1)} * P(D2)} \tag{C.11}$$

4.3.3 D1 (E) D2 method

Accuracy: Let α be the probability of the event a data point $D1_i$ arrives before a data point $D2_i$. In other words, it is the probability that t(D1) is smaller than or equal to t(D2). α is obtained via Eq. (C.12)

$$\alpha = 1 - \frac{\lambda_2}{\lambda_1 + \lambda_2} \tag{C.12}$$

The accuracy of virtual data source is then calculated using Eq. (C.13):

$$Acc(D1\ (E)\ D2) = \alpha Acc(D1) + \overline{\alpha} Acc(D2) \tag{C.13}$$

C

Timeliness: In this case, the timeliness of the virtual data source is the minimum value of the set which consists of $t(D1)$ and $t(D2)$, i.e. $min(t(D1), t(D2))$. Given $t \geq 0$, the distribution function of the timeliness $min(t(D1), t(D2))$ is also exponential [27]. It can be obtained by considering complementary cumulative distribution function as shown in Eq. (C.14).

$$f_{tmin}(t; \lambda) = (\lambda_1 + \lambda_2) e^{-(\lambda_1 + \lambda_2)t} \tag{C.14}$$

The timeliness is then obtained via Eq. (C.15)

$$Time[D1\ (E)\ D2] = \int_0^\infty t(\lambda_1 + \lambda_2) e^{-(\lambda_1 + \lambda_2)t} dt = \frac{1}{\lambda_1 + \lambda_2} \tag{C.15}$$

The Eq. (C.15) can be rewritten through $Time(D1)$ and $Time(D2)$ as follows.

$$Time[D1\ (E)\ D2] = \frac{1}{\lambda_1 + \lambda_2} = \frac{Time(D1) * Time(D2)}{Time(D1) + Time(D2)} \tag{C.16}$$

4.3.4 Summary of combination methods

By combining data quality dimensions, we aim to generate a virtual data source with better data quality. The three combination methods have different effects on data quality dimensions. A method can increase or decrease the quality depends on the receiving data. Table C.2 shows the results of applying combination methods on two independent data sources $D1$ and $D2$.

✓: It can be better than both of D1 and D2.
$-$: It is not decidable. It varies from case to case.
$\times$: It is worse than both of D1 and D2.

Table C.2: Quality combination result

Combination method	Completeness	Accuracy	Timeliness
D1 (A) D2	✓	✓	×
D1 $\bigoplus$ D2	✓	–	–
D1 (E) D2	✓	–	✓

According to this table, all three methods can increase the completeness. By using average method, the combined data source would have better accuracy. This method also improves the accuracy. However, it makes the timeliness become worse. For the $\bigoplus$ method, both the accuracy and timeliness of the combined data source varies from case to case.The (E) method helps to increase the completeness and timeliness, but not accuracy. If the timeliness is the critical choice, (E) method is recommended to use.

5 Implementation

As a proof of concept, we have developed a prototype system based on the overview described in Fig. C.3. This section discusses the internal implementation details of the prototype.

5.1 Prototype description

We use the client-server architecture to develop our prototype system. The client is the web-based client application that allows users to make requests. The client is also in charge of data visualization in terms of graphs. The integrated server (IS) is responsible for request handling and communicating with data providers.

Data providers make services available as web services and describe the services using Web Service Description Language (WSDL). Both Restlet and JAX-WS (Java API for XML Web Services) clients are employed to send data using SOAP protocol and REST architecture style.

5.1.1 The client side

The client side consists of a user interface where users can choose the measurement type, the quality dimensions, the constraints, and one quality dimension as selection dimension. We use Ajax (Asynchronous JavaScript and XML) technology to handle the messages from the server and display information on a graph. Flot, a JavaScript plotting library is used to produce graphical plots on a web browser [22].

5.1.2 The integrated servers

The IS consists of following main components:

- MuleESB[2]: is an open source enterprise service bus framework and we use it as the communication backbone of the system. It is easy to use Mule ESB, compared with other open source ESB frameworks such as PEtALS ESB, ServiceMix, and Open ESB [42]. Mule ESB is not based on JBI (Java Business Integration), but it provides seamless support for JBI containers [35]. Hence, it allows components of other ESBs, e.g., ServiceMix, which are based on the JBI model, to be used alongside MuleESB. Mule ESB is provided together with an Integrated Development Environment, Mule studio which makes the process of flow design much easier. Mule ESB also provides many connectors and transports, for instance REST, SOAP, JMS (Java Message Service) [45].
- Service Manager: is a .NET web service that is deployed on Internet Information Services (IIS)[3] web server.
- Dynamic Invoker: is also a Java-based web service that dynamically binds and invokes services for gathering the selected data sources.
- Service-quality Database: is developed using Microsoft SQL Server[4].
- Request handler: is a Java web service deployed on the MuleESB. This service receives the information from web-based client application and forwards it to the *Service Manager*.
- Quality calculator: The quality calculator is .NET web service which has a separate function for each quality dimension. It takes a data source and a reference data source, then computes the quality value.
- Combination Engine: is a .NET web service. It takes a list of data sources and combines them according to the corresponding combination formula, e.g., the

[2] http://www.mulesoft.org
[3] http://www.iis.net
[4] http://www.microsoft.com/en-us/sqlserver/default.aspx

one presented in Sect. C and stores a new virtual data source in the *Service-quality database*.

5.2 Use case scenarios

C

We use three offshore wind scenarios to demonstrate the use of the prototype system. A number of wind sensors (e.g., wind speed sensor) are attached to a wind turbine to monitor and control the wind turbine [1]. Due to the weather conditions, the sensors are subject to the moisture and corrosion [37]. Consequently, the quality of the data produced by them can be negatively influenced [10]. It is therefore needed to have multiple sensors to measure the same physical phenomena.

5.2.1 Scenario 1

Description: There are three real wind speed data sources in the system. Each data source has two quality dimensions available, e.g., completeness and timeliness. Fig. C.5 shows scenario 1 where the data is mainly provided by wind sensors through a number of wind services.

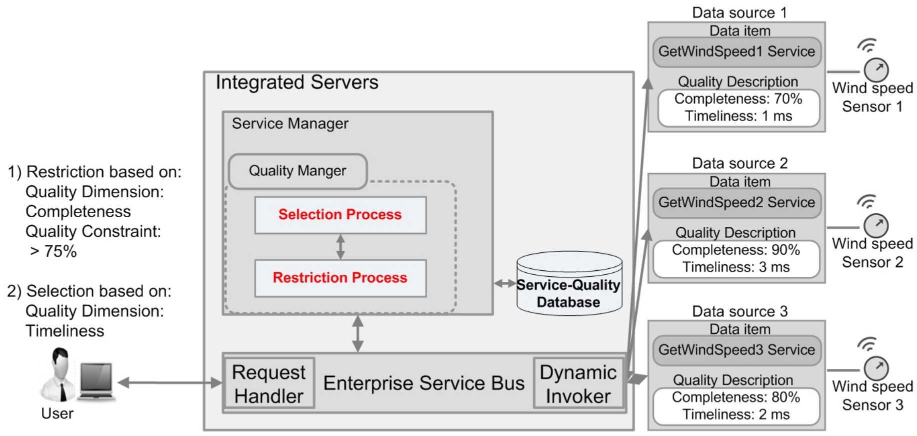


Figure C.5: Scenario 1. Data sources: three real data sources. Quality dimensions: completeness and timeliness. Request: completeness $\geq 75\%$ and selection quality dimension is timeliness.

User's request: The user sends a request for wind speed. Completeness of more than 75% is chosen for the restriction process and timeliness is selected as the criteria for data source selection.

Results: First the restriction process is executed, data sources 2 and 3 are selected. The selection process based on timeliness is then executed. As the result, data source 3 is selected since it has better timeliness compared to data source 2.

5.2.2 Scenario 2

Description: There are two real wind speed data sources. Each data source has one available quality dimension, e.g., completeness. Fig. C.6 shows the scenario where the data is mainly provided by wind sensors through a number of wind services.

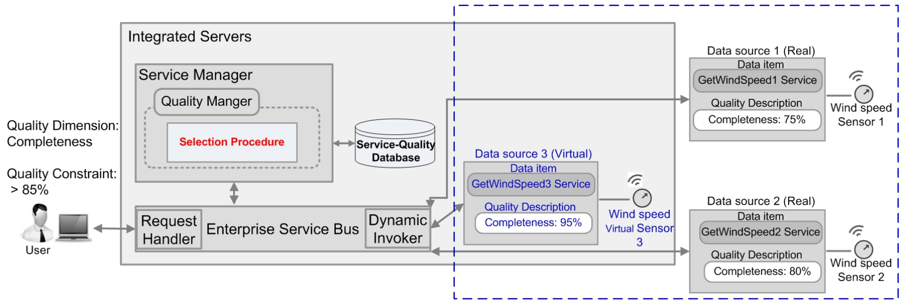


Figure C.6: Scenario 2. Data sources: two real data sources. Quality dimension: completeness. Request: completeness $\geq 85\%$.

User's request: The user issues a request for wind speed with completeness of more than 85%.

Results: The completeness of data sources 1 and 2 are 75%, 80% respectively. It means that none of the given data sources meets the user's requirement. The system therefore to combine these two data sources. As the result, a virtual data source which has the completeness of 95% is produced. The virtual data source that is selected by *Quality Manager* and its graph is shown to the user. The reason is that data sources 1 and 2 have missed some parts of the data but when they are composed, the missing parts are filled.

5.2.3 Computation of data quality scenario

Description: There are three real wind speed data sources in the system. The quality descriptions of data sources 1 and 2 are not available. The reference data source has one available data quality dimension, e.g., completeness. Figure C.7 illustrates the scenario.

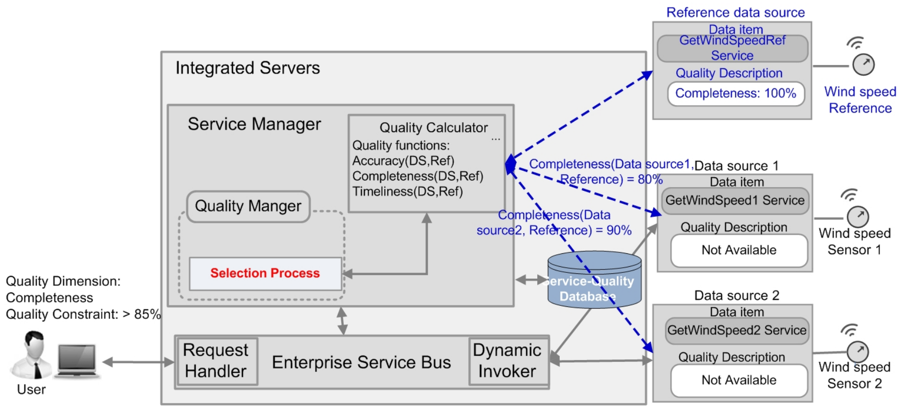


Figure C.7: No data quality descriptions available for data source 1 and 2. The reference data source has completeness available. Request: completeness $\geq 85\%$.

User's request: The user issues a request for wind speed with the completeness of more than 85%.

Results: By using the reference, the *Quality Calculator* computes the completeness of 80% and 90% for data source 1 and 2, respectively. Data source 2 is selected and returned to the user.

6 Related work and discussion

In order to support intelligent handling of data sources in ESB, some efforts have been reported. For example, authors in [45] proposed a multi-layered framework to support content-based intelligent routing path construction and message routing. Their approach facilitates the data source selection based on the message content. Besides handling data sources in ESB, there is an attempt to combine data sources based on data quality as reported in [29]. There has also been some research within the area of multi-sensor data fusion that employed different techniques to generate a better virtual sensor, for instance, artificial intelligence, pattern recognition, statistical estimation are used to fuse multi-sensor data [16, 33].

Different from these work, we have taken into consideration data quality dimensions as criteria for filtering and selecting the best suitable data source. We have also implemented semantic technologies to describe services and sensor networks.

In this work, we have only discussed combination methods for data sources that measure the same physical phenomena, e.g., wind speed. In reality, sometimes we need to combine two or more measurement quantities in order to fulfill users' requests. The combination can be based on mathematical relations between the quantities, e.g., from rotor speed, power output of a wind turbine can be easily derived provided power coefficient of the generator is given and the pitch angle is constant.

7 Conclusions

Data quality is one of data consumers' concerns. It is an important feature of any business organization. Data consumers normally need to be aware of data providers and the quality of data that they get. Indeed, it is complicated for data consumers, data integrators, and even data providers when the number and diversity of data sources increase. ESB is proposed to make the data integration easier. This will reduce the details that data consumers have to know in order to get access to quality data. However, there is a lack of unambiguous ways to manage data sources in current available open source ESB platforms. In this work, we have presented a quality-based approach to managing, selecting, and providing the most suitable data source for users based upon their quality requirements. The approach gives users possibility of getting the most suitable data source from the available ones. It also increases the chance to find the requested data source by combining multiple data sources so that the users' quality requirements are met. We have also enhanced the data quality-based approach by implementing semantic technology to describe sensor networks in an unambiguous manner.

Acknowledgement

This work has been (partially) funded by the Norwegian Centre for Offshore Wind Energy (NORCOWE) under grant 193821/S60 from the Research Council of Norway (RCN). NORCOWE is a consortium with partners from industry and science, hosted by Christian Michelsen Research.

References

[1] Akhmatov, V., & Knudsen, H.: An Aggregate model of a grid-connected, large-scale, offshore wind farm for power stability investigations - importance of

C

windmill mechanical system. International Journal of Electrical Power & Energy Systems **24**, no. 9, 709–717 (2002)

[2] Baumgartner, N., Gottesheim, W., Mitsch, S., Retschitzegger, W., and Schwinger, W.: Improving Situation Awareness In Traffic Management. In Proc. Intl. Conf. on Very Large Data Bases (2010)

[3] Bai, X., Xie, J., Chen, B., and Xiao, S.: Dresr: Dynamic routing in enterprise service bus. In: e-Business Engineering, ICEBE 2007. IEEE International Conference on, pp.528–531, IEEE (2007)

[4] Barry, D. K.:*Web services and service-oriented architectures: the savvy manager's guide*. Morgan Kaufmann (2003)

[5] Bröring, A., Maué, P., Janowicz, K., Nüst, D., & Malewski, C.: Semantically-enabled sensor plug & play for the sensor web. Sensors **11**(8), 7568-7605 (2011)

[6] Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., et al.: The SSN ontology of the W3C semantic sensor network incubator group. Web Semantics: Science, Services and Agents on the World Wide Web **17**, 25–32 (2012)

[7] Chappell, D.A.: *Enterprise Service Bus*. O'Reilly media (2009)

[8] Delia, P., & Borg, A.: *Mule 2: A Developers Guide*. Apress (2008)

[9] de Jesus Rubio, J., Figueroa, M., Pacheco, J., & Jimenez-Lizarraga, M.: Observer design based in the mathematical model of a wind turbine. International Journal of Innovative Computing Information and Control, **7**(12), 6711-6725 (2011)

[10] Desholm, M.: Thermal animal detection system (TADS): development of a method for estimating collision frequency of migrating birds at offshore wind turbines. Tech. report, *National Environmental Research Institute* (2003)

[11] Dumais, S.T., Furnas, G.W., Landauer, T.K., Deerwester, S., and Harshman, R.: Using latent semantic analysis to improve access to textual information. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 281–285, ACM (1988)

[12] DeLone, W.H., and McLean, E.R.: Information Systems Success: The quest for the dependent variable. Information Systems Research **3**, no. 1, 60–95 (1992)

[13] Eppler, M.J.:*Managing Information Quality: Increasing The Value of Information in Knowledge-intensive Products and Processes*. Springer (2006)

[14] Ge, M., and Helfert, M.: A framework to assess decision quality using information a quality dimensions. In Proceedings of the 11th International Conference on Information Quality - ICIQ, v. 6, pp. 10–12 (2006)

[15] Geisler, S., Weber, S., and Quix, C.: An ontology-based data quality framework for Data stream applications. 16th International Conference on Information Quality, pp. 145–159 (2011)

[16] Hall, D.L., and Llinas, J.: An introduction to multisensor data fusion. Proceedings of the IEEE **85**, no. 1, 6–23 (1997)

[17] Huang, K.T., Lee, Y.W., and Wang, R.Y.: Quality Information and Knowledge. *Prentice Hall PTR* (1998)

[18] ISO. ISO 5725-2: 1994: Accuracy (trueness and precision) of measurement methods and results-part 2: methods for the determination of repeatability and reproductibility (1994)

[19] Janowicz, K., and Compton, M.: The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology. 3rd International Workshop on Semantic Sensor Networks (2010)

[20] Klein, B.D.: Data quality in the practice of consumer product management: evidence from the field. Data Quality **4**, no. 1 (1998)

[21] Karastoyanova,D., Wetzstein, B., Lessen, T.V., et al. Semantic service bus: architecture and implementation of a next generation middleware. Data Engineering Workshop, IEEE 23rd International Conference on, pp. 347–354, IEEE (2007)

[22] Laursen, O.: Flot - Attractive Javascript plotting for jQuery - Google Project Hosting. Accessed: 23/05/2013 (2012)

[23] Lee, Y.W., Strong, D.M., Kahn, B.K., and Wang, R.Y.: AIMQ: A methodology for information quality assessment. Information and Management **40**, no. 2, 133–146 (2002)

[24] Manyonge, A. W., Ochieng, R. M., Onyango, F. N., and Shichikha, J. M.: Mathematical modelling of wind turbine in a wind energy conversion system: power coefficient analysis. Applied Mathematical Sciences *6*(91), 4527-4536 (2012)

[25] Maréchaux, J. L.: Combining service-oriented architecture and event-driven architecture using an enterprise service bus. *IBM Developer Works*, 1269–1275 (2006)

[26] Michelson, B. M.: Event-driven architecture overview. *Patricia Seybold Group* (2006)

[27] Marshall, A.W., and Olkin, I.: A multivariate exponential distribution. Journal of the American Statistical Association **62**, no. 317, 30–44 (1967)

[28] Mihaila, G., Raschid, L., and Vidal, M.E.: Querying quality of data metadata. In Third IEEE META-DATA Conference (1999)

[29] Mihaila, G., Raschid, L., and Vidal, M.E.: Using quality of data metadata for source selection and ranking. Proceedings of the Third International Workshop on the Web and Databases (WebDB), pp. 93–98 (2000)

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

[31] O'Byrne, D., Brennan, R., and O'Sullivan, D.: Implementing the draft W3C semantic sensor network ontology. Pervasive Computing and Communications Workshops (PERCOM Workshops), 8th IEEE International Conference on, pp. 196–201 (2010)

[32] The Norwegian Oil Industry Association: Reference architecture of IT systems for OLF's IO G2 (2008). Technical report (2008)

[33] Le P.D, and Manfred Hauswirth, M.: Linked open data in sensor data mashups. Proceedings of the 2nd International Workshop on Semantic Sensor Networks (2009)

[34] Papazoglou, M. P., and Van Den Heuvel, W. J.: Service oriented architectures: approaches, technologies and research issues. The VLDB Journal **16**(3), 389–415 (2007)

[35] Rademakers, T., and Dirksen, J.: *Open-source ESBs in action*, Manning (2008)

[36] Sheth, A., Henson, C., and Sahoo, S.S.: Semantic sensor web. Internet Computing **12**, no. 4, 78–83, IEEE (2008)

[37] Snyder, B., and Kaiser, M.J.: Ecological and economic cost-benefit analysis of offshore wind energy. Renewable Energy **34**, no. 6, 1567–1578 (2009)

[38] Strong, D.M., Lee, Y.W., and Wang, R.Y.: Data Quality in Context. Communications of the ACM **40**, no. 5, 103–110 (1997)

[39] Shardanand, U., and Maes, P.: Social information filtering: algorithms for automating word of mouth. Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 210–217, ACM Press/Addison-Wesley Publishing Co. (1995)

[40] Sreenath, R.M., and Singh, M.P.: Agent-based service selection. Web Semantics: Science, Services and Agents on the World Wide Web **1**, no. 3, 261–279 (2004)

[41] Trinugroho, Y.B.D., Rasta, K., Nguyen T.H., Fensli, R.W., and Reichert, F.: A real-time Web-based health monitoring system based on enterprise service bus. In: Proceedings of 2012 11th IADIS International Conference on WWW/Internet, pp. 165–172 (2012)

[42] Ueno, K., and Tatsubori, M.: Early capacity testing of an enterprise service bus. Web Services, ICWS'06. International Conference on, pp. 709–716, IEEE (2006)

[43] Wang, R.Y., and Strong, D.M.: Beyond Accuracy: What data quality means to data consumers. Journal of Management Information Systems **12**(4), 5–33 (1996)

[44] Wang, R. Y.: A product perspective on total data quality management. Communications of the ACM **41**(2), 58-65 (1998)

[45] Ziyaeva, G., Choi, E., and Min, D.: Content-Based Intelligent Routing and Message Processing in Enterprise Service Bus. Convergence and Hybrid Information Technology, 2008. ICHIT'08. International Conference on, pp. 245–249, IEEE (2008)

[46] Zhang, Y.: Dependable ESB Routing in Hybrid Service Execution Environment. Advances in Information Sciences and Service Sciences (AISS) **2**(1), 83–93 (2010)

Appendix D

Paper D

Title	Big Data Metadata Management in Smart Grids
Authors	Trinh Hoang Nguyen[a], Vimala Nunavath[a], and Andreas Prinz[a]
Affiliation(s)	[a] Department of ICT, University of Agder, Grimstad, Norway
In Book	Big Data and Internet of Things: A Roadmap for Smart Environments
Copyright	©Springer-Verlag

Big Data Metadata Management in Smart Grids

Trinh Hoang Nguyen, Vimala Nunavath, and Andreas Prinz
Department of ICT, University of Agder, Grimstad, Norway
E-mails: {trinh.h.nguyen, vimala.nunavath, andreas.prinz}@uia.no

Smart home, smart grids, smart museum, smart cities, etc. are making the vision for living in smart environments come true. These smart environments are built based upon the Internet of Things paradigm where many devices and applications are involved. In these environments, data are collected from various sources in diverse formats. The data are then processed by different intelligent systems with the purpose of providing efficient system planning, power delivery, and customer operations. Even though there are known technologies for most of these smart environments, putting them together to make intelligent and context-aware systems is not an easy task. The reason is that there are semantic inconsistencies between applications and systems. These inconsistencies can be solved by using metadata. This chapter presents management of big data metadata in smart grids. Three important issues in managing and solutions to overcome them are discussed. As a part of future grids, some concrete examples from the offshore wind energy are used to demonstrate the solutions.

1 Introduction

Advanced technologies are making the vision for living in smart environments become realistic. Recently, several concepts within the smart environments have been introduced, such as smart home, smart transport, smart grids, smart museum, and smart cities. These smart environments are built based upon the Internet of Things (IoT) paradigm where lots of devices, sensors, appliances are connected through the Internet. These devices produce vast amounts of data, thus making the management of data a highly challenging task. Another common feature and an important problem of these smart environments is that each of them involves data modeling, information analysis, integration and optimization of large amounts of data coming from various smart appliances in diverse formats. The data are then processed by different intelligent systems with the purpose of providing efficient system planning, power delivery, and customer operations. Even though there are known technologies for developing most of these smart environments, putting them together to make intelligent and context-aware systems is not an easy task. The reason is that there are semantic inconsistencies between applications and systems. These inconsistencies can be solved by using metadata.

Typically, data are a collection of raw and unorganized symbols that represent real-world states. The information is the processed, organized, and structured data according to a given context [1, 144]. The context of related data and processes will decide the role as information of the captured data. Principally, information is the structured data with semantics. For example, if data are used for documentation or analysis, the data become information. Without metadata, the data cannot easily become information and incomplete or inaccurate metadata or too much metadata can cause misinterpretation of data [55]. Metadata should be therefore managed in a way that data can be easily interpreted and transformed to information.

Metadata management is a key to make data integration successful [25]. It has to be taken into consideration in the development of systems since it helps in making the systems scalable. For formal metadata management, semantic technologies have been developed. Ontology, which is a part of semantic technologies, plays a significant role in managing metadata of a domain. Ontologies can be used to support data integration in terms of facilitating knowledge sharing and data exchange between participants in a domain. In ontologies, concepts, properties, relations, functions, constraints, and axioms of a particular domain are explicitly defined [18]. We use semantic technologies to exploit the semantics of data, and hence ease metadata handling in smart environments.

In this chapter, we discuss how to manage big data metadata in smart grids with a particular focus on (1) knowledge sharing and data exchange, (2) derived data from relations between concepts, and (3) data quality as metadata. We will present a developed ontology model for offshore wind energy metadata management as an example of domain concept descriptions. IEEE P2030 points out that ontology might be a good option to create formal representation of real-world systems or objects composing these systems within smart grids [24]. As the number of devices is increasing tremendously, and many of them will be used in smart environments, it is important to make sure that any future system is scalable enough to keep pace with the technologies. Metadata models, as a backbone of any system, also need to be considered thoughtfully. The models need to be developed so that the following requirements are fulfilled.

- The models need to be compatible with existing data resources and future applications.
- Minimum effort is used to modify the models when integrating new devices.
- New devices' metadata are described in a way that discovery and access to

them are easy.

- It must provide a guide to structuring, sharing, storing, and representing the big data in smart grids.
- The semantics of data needs to be exploited and clearly defined.
- Since it is not feasible to attach metadata with individual data, the metadata models must be related to data sources.

The rest of the chapter is organized as follows. Section 2 gives some background information about the areas that we discuss in this work. Section 3 presents some challenges of big data metadata management that we attempt to tackle. Section 4 describes solutions and approaches to overcoming the challenges. Section 5 discusses our solutions and gives some remarks on future work. Finally, section 6 concludes the chapter.

2 Background

This section describes the background of metadata, semantic technologies, IoT and smart grids. The relations between these areas are also highlighted.

2.1 Metadata

The term "metadata" was first introduced in 1968 by Philip R. Bagley to refer to descriptive data that provided information about other data in a database environment [51]. In different contexts, the term metadata is interpreted in different ways, for example, metadata are data about data; or metadata are machine-readable information about electronic resources or other things; or metadata are structured information that describes, explains, locates an information resource [54]. Basically, metadata are descriptors that describe a way of identifying information. Data without metadata result in blind decision making [55]. In other words, without metadata, data have no identifiable meaning. For instance, when a user searches for information, he will receive a list of search results from a search engine. The search engine looks up for requested information from huge amounts of data based on search terms, tagging content, and other metadata associated with data. Metadata provide the necessary documentation for users by answering who, what, when, where, why, and how questions upon the users' requests.

Metadata put data into a context so that the data can be understood by users and become information. Besides the general role as descriptors, metadata can be used for:

- information classification - information is classified into different categories based on content, purpose, location, area, etc;
- information discovery - a large amount of time is used to look for things, and many of them cannot be found due to the lack of descriptions. Metadata therefore enhance information discovery and knowledge sharing;
- information interpretation - a poor description of data may lead to wrong decision making or business loss due to wrong interpretation of the data;
- data integration - when we integrate data from various sources in different formats and platforms, metadata are the only option that can make a foundation for data integration [55];
- device discovery - based on metadata of devices such as location, type, and other features devices can be discovered either semi-automatically or automatically by a system.

2.2 Big data metadata management

Big data is characterized with volume, variety and velocity [61]. Volume is considered as a huge amount of data which can hold terabytes to petabytes of data which come from different devices, applications, and systems. Velocity is the speed at which the data comes in, and variety means many data types and data formats. Structured, semi-structured and unstructured data are involved in big data [14]. Data often come from machines, sensors, social networks such as Facebook, Tweets, smart phones and other cell phones, GPS devices and other sources making it complex to manage [45]. According to a report from McKinsey Global Institute, every year, over 30 billion original documents with data are created. 85% of the data will never be retrieved, 50% of the data is duplicates, and 60% of stored documents are obsolete. $1 and $10 are the costs to create a document and to manage it, respectively [31]. As the amount of data increases, the cost of management also increases. It is important to describe and manage metadata so that only important and necessary data are stored and provided to users when requested. Since data are used for making decisions by different applications and systems, the quality of data is one of concerns.

Not all of the data captured from sensors or devices are useful, only a part of the data is. Data are transformed to information only if the data are used for particular purposes, e.g., modeling, documentation. Part of the information will become knowledge in terms of abstraction and perception. Users are not interested in information (numbers), they are interested in knowledge, i.e., what can be derived from the information. For example, if a user wants to know about the temperature in a wind turbine hub, he will probably not expect to get a number or set of numbers as a response, but he will probably want to get either “Normal”, “Cold”, or “Hot”. Eventually, only part of the knowledge will be transformed to wisdom if the knowledge is used to serve some actionable intelligence [46]. Every step of the transformation involves management of data, information, and knowledge. Management of big data metadata concerns a way to manage big data metadata such that metadata are good enough to enable knowledge extraction from big data.

2.3 Smart grids and Internet of Things

Smart grids are the future generation of power grids where the energy is managed in a way that both consumers and energy producers will get more benefits from the grid in terms of reduction of expenditure on energy and reduction of carbon emissions. Indeed, it enables consumers to utilize lower tariff charges during off-peak periods and energy producers to react efficiently during peak periods. Smart grids are also used to effectively response to the fluctuations of renewable sources such as wind and solar when they are integrated in a power grid.

A smart grid is an electricity network that efficiently delivers sustainable, economic, and secure electricity supplies by intelligently integrating the actions of all users connected to it, including generators, consumers and those that do both [15]. On the consumer side, smart grids involve many smart meters and smart appliances, for example, smart washing machines, and dishwashers. The number of smart appliances is increasing dramatically. These devices are connected directly to the Internet. A large amount of sensors are used in these devices to make sure that every single change can be detected, managed and controlled. On the energy provider side, intelligent applications are used to maintain balance between demand and supply. Smart grids will bring the decision making gradually from a centralized level to local and finally to automatic.

In order to make a grid become smart, different technologies and applications are involved, e.g., advanced metering infrastructure (AMI), distribution management

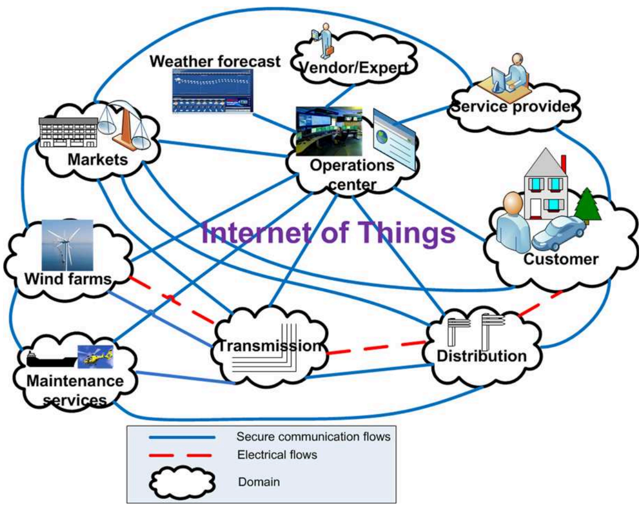


Figure D.1: An example of conceptual model for smart grid communication

system (DMS), geographic information system (GIS), outage management systems (OMSs), intelligent electronics devices (IEDs), wide-area measurement systems (WAMS), and energy management systems (EMSs) [11]. These systems are driven effectively by IoT [56].

In IoT, things are connected in such a way that machines and applications can understand our surrounding environments better and therefore make intelligent decisions and respond to the dynamics of the environments effectively [5]. These things communicate to each other over the Internet. Advantages of IoT will contribute a lot to the effort of making smart grids in terms of real-time monitoring and control. Smart grid applications require quick response time no matter how big the data are. One example of such a system is an energy trading system which allows energy consumers or third parties to bid for energy prices in advance [12].

Due to characteristics of smart grids, a number of challenges are encompassed with the development of smart grids such as support heterogeneous participants, flexible data schema (e.g., add new or remove old appliances), complex event processing, privacy and security [57]. Thus, data from IoT alone are not enough. The data must be used together with the domain knowledge, machine interpretable metadata, services, etc. to become useful.

Figure D.1 illustrates a conceptual model for smart grid communication with a focus on offshore wind as an energy generator. The model is based on the *Smart Grid Interoperability Panel* promoted by the National Institute of Standards and Technology (NIST) [38].

Each domain is a high-level grouping of organizations, individuals, and systems of the offshore wind industry. Communication between stakeholders in the same domain may have similar characteristics and requirements. The communication flows are bidirectional. In this model, smart meters, smart appliances are installed at households, sensors are embedded on wind turbines, and intelligent programs are used at operations center.

Metadata are significant in the smart grid context. It is needed for organizing and interpreting data coming from energy market, service providers, customers, power grid, and power generators. Managing metadata in such a varied environment is a challenging task.

2.4 Semantic technologies

Semantic technologies have been developed to make metadata understandable by a machine. Ontology is a part of semantic technologies that plays a significant role in managing metadata of a domain. There are several ontology languages such as SHOE, OIL, DAML-ONT, DAML+OIL, and OWL [30, 64]. Web Ontology Language (OWL), a language proposed by World Wide Web Consortium (W3C) Web Ontology Working Group, is being used intensively by research communities as well as industries. Ontologies can be represented by using Resource Description Framework (RDF)/RDFS (RDF Schema). However, a number of other features are missing in RDFS such as cardinality restrictions, logical combinations (intersections, unions or complements), and disjointness of classes. Let us examine some concrete cases within the offshore wind energy. The first case is that in RDF, we cannot state that *HydraulicSystem* and *HeatingSystem* are disjoint classes. The second case concerns the lack of cardinality restrictions, e.g., the fact that a wind power plant (WPP) can have more than one wind turbine converter component (WCNV) cannot be expressed in RDF, but it can be done in OWL using the following axiom $WPP \sqsubseteq (\geq 1\ hasWPPComponent.WCNV)$. OWL is an extension of RDFS, in the sense that OWL uses the RDF meaning of classes and properties [20, 15, 7]. The design of OWL was influenced by its predecessors DAML+OIL, the frames paradigm and RDF [20].

In OWL, Owl:Thing is a built-in most general class and is the class of all individuals. It is a superclass of all OWL classes. Classes are defined using owl:Class. A class defines a group of individuals that belong together. Individuals are also known as instances. Individuals can be referred to as being instances of classes. Note that the word concept is sometimes used in place of class. 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. There are two types of properties in OWL ontology, they are object property and data type property. Properties in OWL are also known as roles in description logics and relations in Unified Modeling Language (UML). An object property relates individuals to other individuals (e.g., *hasWPPComponent* relates *WPP* to *WPP components*). An object property is defined as an instance of the built-in OWL class owl:ObjectProperty. A data type property relates individuals to data type values (e.g., *hasOilPressure*, *hasWindSpeed*). A datatype property is defined as an instance of the built-in OWL class owl:DatatypeProperty. A property in OWL can be transitive, functional, symmetric, or inverse.

OWL DL (DL stands for "Description Logic") is a variant of OWL. It was developed to support existing DL and to provide a possibility of working with reasoning systems. In this work, OWL DL is used to develop ontologies. The OWL DL semantics is very similar to the $\mathcal{SHOIN}^{(\mathcal{D})}$ Description Logic. It provides maximum expressiveness and it is decidable [20]. OWL DL abstract syntax and semantics can be found in [41].

2.5 Ontology reasoning and querying

A reasoner is a piece of software that is able to infer logical consequences from a set of asserted facts or axioms. It is used to ensure the quality of ontologies. It can be used to test whether concepts are non-contradictory and to derive implied relations. Reasoning with inconsistent ontologies may lead to erroneous conclusions [3]. There are some existing DL reasoners such as FaCT, FaCT++, RACER, DLP and Pellet. A reasoner has the following features: satisfiability, consistency, classification, and realization checking [49]. Given an assertional box $\mathcal{A}$ (ABox contains assertions about individuals), we can reason w.r.t a terminological box $\mathcal{T}$ (TBox contains axioms about classes) about the following:

- Consistency checking: ensures that an ontology does not contain any contradictory facts. An ABox $\mathcal{A}$ is consistent with respect to $\mathcal{T}$ if there is an interpretation $\mathcal{I}$ which is a model of both $\mathcal{A}$ and $\mathcal{T}$.

- Concept satisfiability: checks if it is possible for a class to have any instances. Given a concept C and an instance a, check whether a belongs to C. $\mathcal{A} \models C(a)$ if every interpretation that satisfies $\mathcal{A}$ also satisfies $C(a)$.

- Classification: computes the subclass relations between all named classes to create the complete class hierarchy. Given a concept C, retrieve all the instances a which satisfy C.
- Realization: computes the direct types for each of the individuals. Given a set of concepts and an individual a, find the most specific concept(s) C (w.r.t. subsumption ordering) such that $\mathcal{A} \models C(a)$.

For relational database (RDB), Structured Query Language (SQL) is the query language of choice. But for ontologies, SPARQL and SQWRL (Semantic Query-Enhanced Web Rule Language) [39] are used to build queries. SPARQL is an RDF query language and SQWRL is a SWRL-based language for querying OWL ontologies. SPARQL extensions such as SPARQL-DL [48] and SPARQL-OWL [27] can be used as OWL query languages in many applications. But SPARQL cannot directly query entailments made using OWL constructs since it has no native understanding of OWL [39].

3 Challenges in managing big data metadata in smart grids

There are a number of challenges associated with management of big data metadata such as metadata quality, metadata provenance, semantics, and metadata alignment. In this section, we attempt to tackle three challenges in managing smart grids' big data metadata.

3.1 Knowledge sharing and information exchange

In a diverse environment such as smart grids, meters, appliances, and applications are developed by different companies and vendors. Many of them use their own proprietary data formats, protocols, and platforms, thus data exchange is impeded. Using approved standards would contribute to solving such problems since they can make the data exchange unambiguous. The standards can be seen as a means of interoperability, a dictionary of data that can be used to manage, simplify, and optimize data models [9]. However, there are some problematic issues related to existing international standards for data exchange. For instance, it takes some years to approve a standard internationally, but it seems that new technologies are proposed every year. As a result, novel concepts and terms are introduced, but they are not immediately described in these international standards.

The lack of widely accepted standards prevents the interoperability between smart devices, applications, smart meters, and renewable sources [47]. The Institute of Electrical and Electronics Engineers (IEEE), and NIST have recommended a list of standards that should be considered while developing smart grids [24, 38]. These standards have been developed by different working groups, leading to a lack of harmonizations. Although these standards describe different parts of smart grids, they share a common feature, i.e., the smart grid concepts. The question here is how to structure the domain concepts such that semantics is exploited effectively, knowledge sharing and data exchange are eased, and new concepts are updated in knowledge bases timely.

3.2 Relations between concepts

Ontologies can be used to support data integration in terms of facilitating knowledge sharing and data exchange between participants in a domain. Ontologies describe the relations between concepts and their properties. These relations are metadata since relations can lead to computability of derived data. This opens several possible paths for calculation and gives users the possibility of selecting the most suitable one. However, there is a lack of a formal description of such relations in ontologies. One important question in managing metadata in ontologies is how to handle relations so that the selection of data (independent of type of data: base or derived data) can be done at runtime depending on the actual situation.

3.3 Data quality

It is normal to use more than one sensor to measure, e.g., pressure or temperature at a particular point. The quality of each sensor is different from the others and depends on the conditions. In offshore wind energy, a couple of sensors are deployed on a windmill and they frequently measure and deliver the data to the users and applications by means of services. As sensors are prone to failures their results might be inaccurate, incomplete, and inconsistent [50]. Therefore, the data quality should be handled in such a way that users and applications can specify the desired quality level of the data. Only when the data source has the requested quality descriptions it would be used for further processing. One of the issues related to data quality is the handling of data quality at user level in enterprise applications where there is a potentially large number of data sources with quality information. Another issue is that sometimes none of the available data sources has the required quality information. In this case, how a system should respond to such a request should be considered and a way to provide requested data to users should be investigated.

4 Solutions

This section presents three solutions and approaches to overcoming the challenges described in Sect. 3. Smart grids involve vast amounts of data from consumers, generators, billing, and management. Here, we use a case in which offshore wind energy plays a role as a renewable energy source generator to demonstrate our points.

4.1 Semantic-enhanced concept modeling

This section discusses the solution to the challenge described in Sect. 3.1. We look into how the semantic technologies can help us to solve the challenges with taking into consideration the requirements for the developed metadata models presented in Sect. 1.

4.1.1 The information model

An information model plays an important role in building a smart grid. It not only provides a common basis for understanding the general behavior of smart grid communication, but also facilitates the collaboration process between smart grid stakeholders due to shared concepts with a common semantics. An example of sharing common concepts between partners of offshore wind energy is illustrated in Fig. D.2.

Availability and reliability of data are significant for any systems and partners. Offshore wind partners can efficiently perform their work using the available data. For example, wind speed information is the input to a wind speed prediction program. The output from the program can be used with the generator speed to predict the availability of wind power in the next few hours. In order to optimize wind farm efficiency, wind farm operations information regarding wind direction, active power,

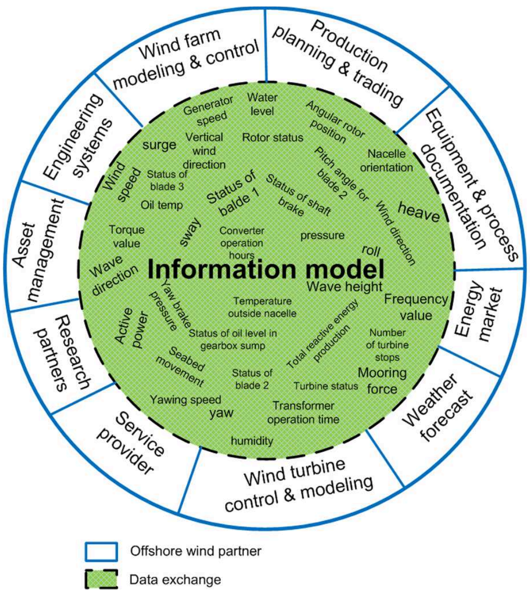


Figure D.2: Data pie chart for the offshore wind industry [36]

status of blades, etc. is needed. The weather forecast and energy market information is used to manage wind power production as well as maintenance for wind turbines (e.g., a wind turbine can be stopped when consumer demand is low). An information model is developed based on the IEC 61400-25 standard [23] to keep pace with the continual introduction of new technologies. More details about the information model can be found in [37].

4.1.2 An offshore wind ontology

An information model represents the knowledge concerning specific domain communication. In particular, the purpose of creating an offshore wind information model is to facilitate the process of agreement on data exchange models as well as collaborations among offshore wind partners. We use the developed information model to build an offshore wind ontology (OWO) as depicted in Fig. D.3. The idea of creating OWO from the terminologies is to share, reuse knowledge, and reason about behaviors across a domain and task. It is also a key instrument in developing the semantic web in which information is given well-defined meaning, better enabling computers and people to work in cooperation [8]. An ontology helps to make an abstract model of a phenomenon by identifying the relevant concepts of that phenomenon [53].

Suppose several different sources/data storages contain wind turbine information.

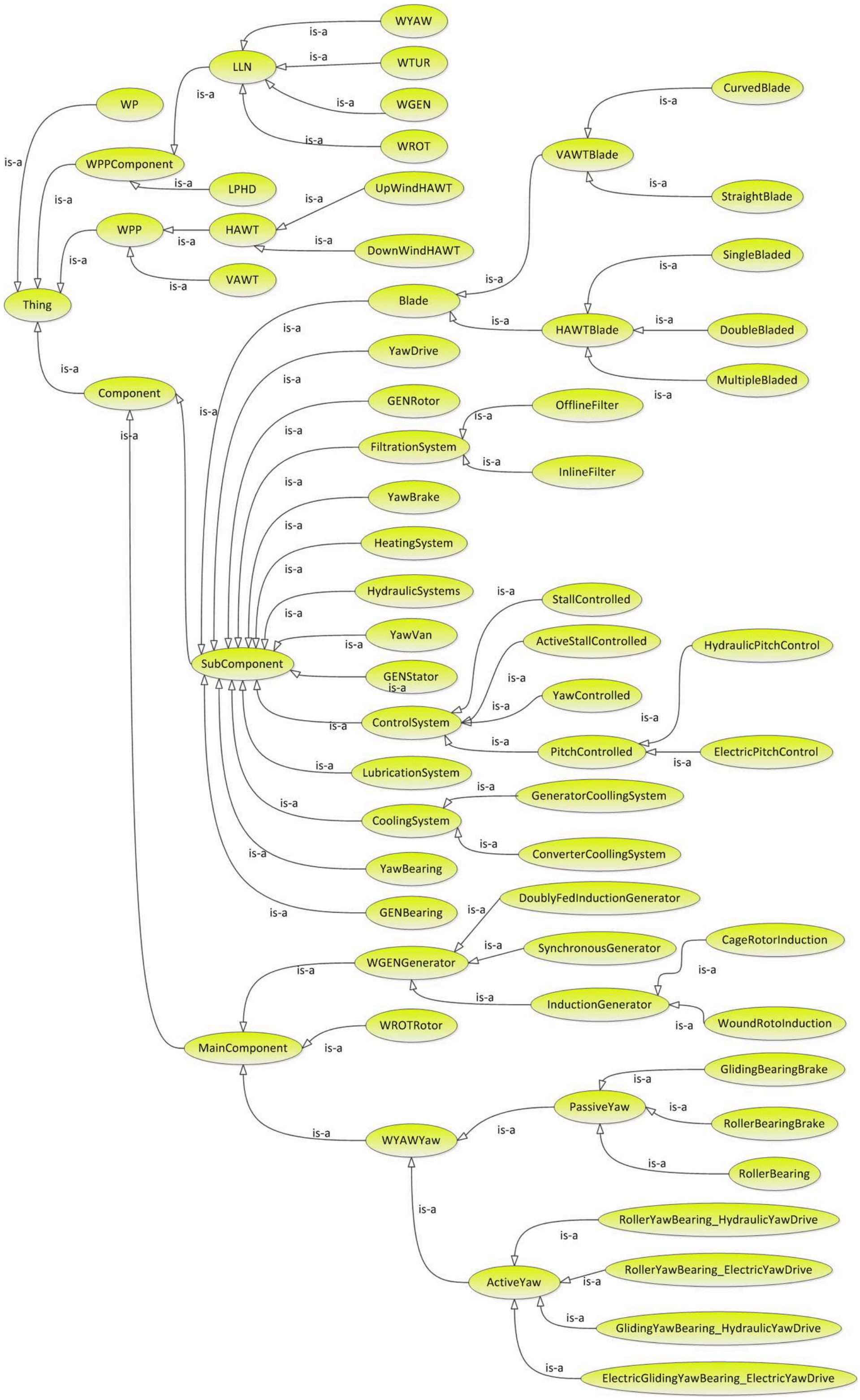


Figure D.3: OWO visualization

D

If these sources share and publish the same underlying ontology of the terms they all use, then computer agents can extract and aggregate information from these different sources. The agents can use this aggregated information to answer user queries or to provide input data to other applications. For example, a SQWRL query over OWO that is used to get oil pressure and pitch angle set point of the wind power plant which has ID is "2300249", is expressed as follows:

$WF(?p)$ ^ $hasID(?p,''2300249'')$ ^ $hasWPPComponent(?p, ?comp)$ ^
$hasOilPressure(?comp, ?pres)$ ^ $hasPitchAngleSetPoint(?comp, ?pitchAngle)$
->$sqwrl : select(?p, ?pres, ?pitchAngle)$

4.1.3 Semantic sensor network ontology

As the number of devices and appliances grows, the number of sensors embedded in such devices will also grow. Ontologies are an adequate way to model sensors and their capabilities [35]. Sensor metadata are used for selecting sensor sources and for integrating with other data sources [28]. Thus sensor metadata are important and needs to be exploited. However, sensor metadata alone cannot make a grid become smart. These metadata must be associated with metadata from devices and appliances that are participated in the grid.

The W3C semantic sensor network incubator group has introduced a semantic sensor network (SSN) ontology[1] to describe sensors, observations, and measurements. The ontology describes sensors and their properties such as accuracy, precision, resolution, measurement range, and capabilities. The ontology includes models for describing changes or states in an environment that a sensor can detect and the resulting observation [13]. An example of the alignment of the SSN ontology to the developed OWO is depicted in Fig. D.4.

The developed OWO can be connected to SSN to share common information such as measurement values from sensors embedded on a wind power plant. At the same time, OWO can still guarantee the complete description of a wind power plant data model. These two ontologies should be maintained separately since the number of concepts in these ontologies might grow as new technologies are introduced.

[1]http://www.w3.org/2005/Incubator/ssn/ssnx/ssn

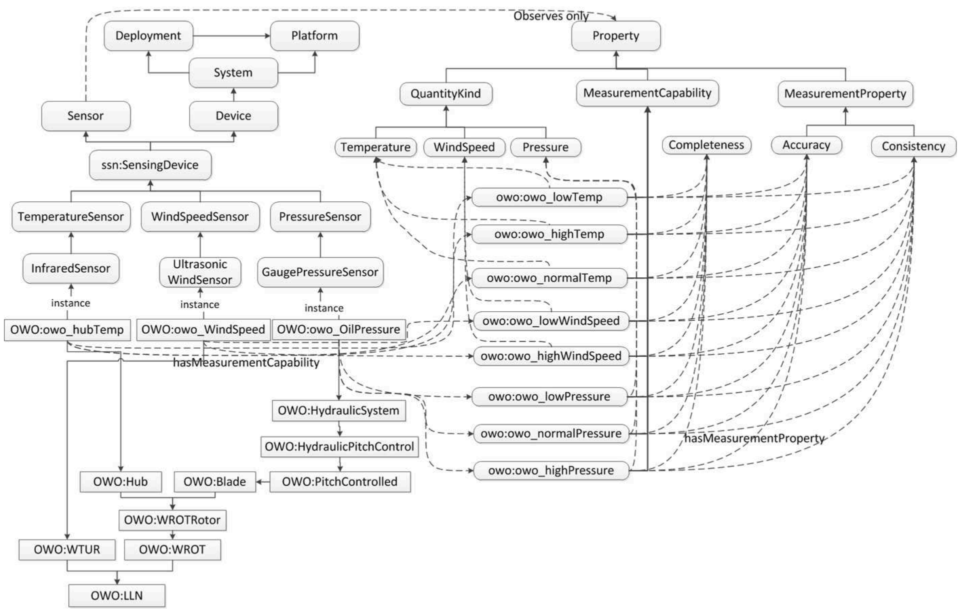


Figure D.4: An example of the alignment of the SSN ontology to OWO

4.2 Relations between concepts

Missing data can be caused by network disconnection, device faults, and software bugs. In some cases, where monitoring of devices or components is extremely important, a single missing value of a data point could lead to wrong predictions or damage of components. In the wind energy domain, many prediction and monitoring applications are employed, for example, power output prediction, wind turbine blade monitoring. The performance of these applications relies very much on data collected from the wind turbines. Missing of a single data item in the set of input data to these applications can make the applications produce wrong output or no output at all. In this case, the missing data item needs to be derived from other available data items. Derivation of data also plays a significant role in decision support systems [43]. For instance, in time-series data analysis, missing data that are located in the middle of a time-series have a high influence on the efficiency of algorithms that are used to reveal hidden temporal patterns such as vector autoregression and exponential smoothing [62]. This section describes a way to model possible paths to deriving missing data from relations between the concepts.

4.2.1 Derived data modeling

Data are classified into two categories: base data and derived data [19]. Base data are those data obtained from data sources. Derived data are those data obtained by combining or computing from base data. The combination and computation of base data are based on relations between domain concepts.

Derived data are described by derived classes and derived attributes. A derived attribute is an attribute that is derived from other attributes in the same class or from different classes that have relationships with the class that contains the attribute. If all attributes of a class are derived, the class is called derived class [4].

Derived data give an advantage of storing data since there is no need to store derived data in a database. Another advantage is that the structure of the data storage is undisclosed to users, derived attributes are accessed via user interface.

Guaranteeing the correctness of derived data is an important task because applications that use the data might produce wrong results if they receive insufficient input. Therefore, derived data need to be handled in such a way that its correctness is ensured. Formally modeling of derived data can help us to figure out different aspects of handling the data, and hence guaranteeing the correctness.

We use UML [17] to model the concepts in the wind domain. UML is based on object-oriented design concepts and is independent of any specific programming language. We also use Object Constraint Language (OCL) to express constraints in UML models [59]. OCL is a complement of UML. It makes models precise, consistent, and complete. In this work, we add OCL constraints to our models to tackle the derived issue mentioned in Sect. 3.2. We analyze two wind energy related cases where derived data play an significant role. We use the ontology introduced in Sect. 4.1.2 to demonstrate the cases.

4.2.2 Derived data within one concept

Temperature measurement can be presented in different units such as Fahrenheit (F) or Celsius (C). The relation between F and C is as follows.

$$F = \frac{9}{5} * C + 32 \tag{D.1}$$

or

$$C = \frac{5}{9} * (F - 32) \tag{D.2}$$

The derivation can be obtained during execution time, for example, the authors of [10] use SWRL to define the transformation between temperature measurement units. However, such an approach will limit the possibility of expressing complex equations. A better approach is to attach formulas directly to properties in ontologies such as [22]. Let us consider a simple ontology describing the wind turbine generator (WGEN) concept and temperature as one of its properties. Figure D.5 illustrates a formal model of temperature conversion using UML and OCL.

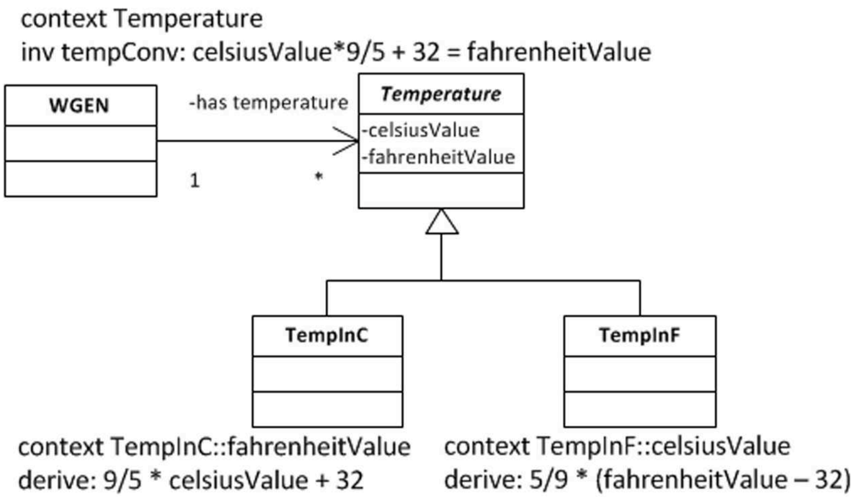


Figure D.5: Temperature conversion

$WGEN$ denotes the wind turbine generator class as described in [23]. $Temperature$ is an abstract class that contains two attributes: the $celsiusValue$ and $fahrenheitValue$. The two classes $TempInC$ and $TempInF$ contain rules to convert temperature unit from C to F and from F to C, respectively.

4.2.3 Derived data between two concepts

Let us consider an offshore wind farm scenario where many sensors are located on a wind turbine to capture information. What if one of them loses the connection? Information related to that one will be lost. How can we utilize other devices to derive that information so that the monitoring of the wind turbine is still ensured? Figure D.6 shows how to make use of derived data from two parameters within the wind domain. The basic mathematical relation between wind speed and power output is expressed in Eq. (D.3) [33].

$$P_{avail} = \frac{1}{2}\rho\pi r^2 v^3 C_p \tag{D.3}$$

where P_{avail} denotes the available power output (W), ρ denotes air density (kg/m^3), r denotes blade length (m), v is the wind speed (m/s), and C_p denotes the power coefficient. Please note that the power coefficient is not constant; it depends on other factors such as rotational speed of the turbine, pitch angle, and angle of attack [34].

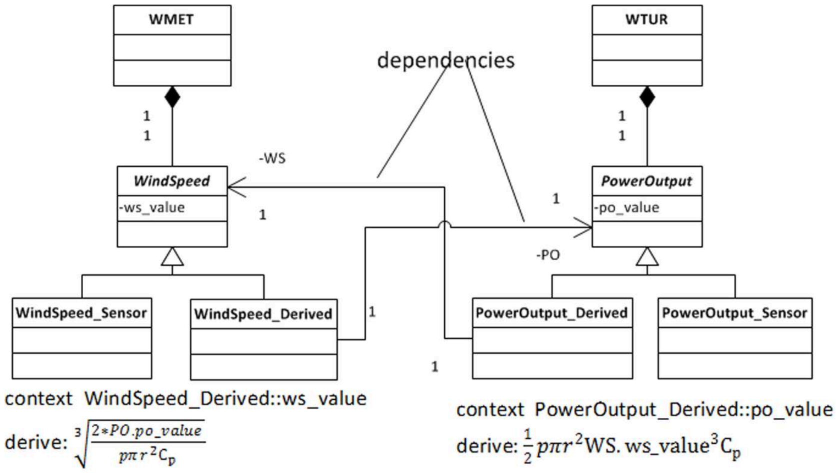


Figure D.6: Derivative relationships between two concepts

4.2.4 Derived data with more than two concepts

What happens if one more parameter is added to the system? As an extension of the two concept model, we can have a model for three parameters as shown in Fig. D.7.

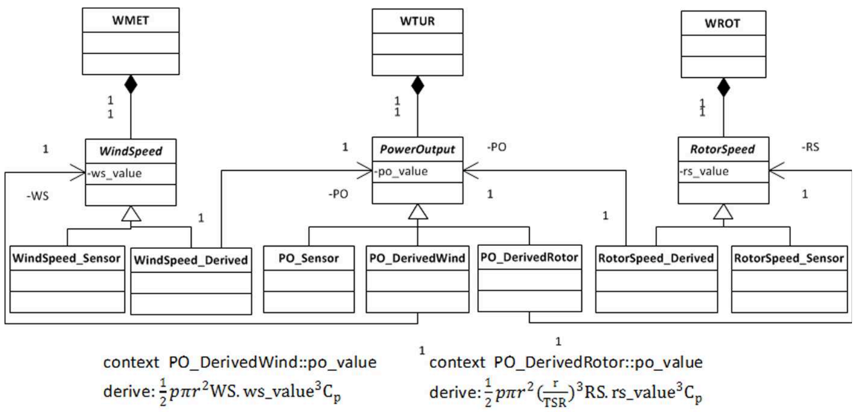


Figure D.7: Derivative relationships between three concepts

Equation (D.3) can be rewritten as follows:

$$P_{avail} = \frac{1}{2}\rho\pi r^2 C_p(\frac{r}{TSR})^3\omega^3 \tag{D.4}$$

where TSR is tip speed ratio, ω (rpm) is the rotational speed of the blade. The TSR value can be obtained from the blade manufacturer, otherwise let TSR equal 7 since it is the most widely reported value in three bladed wind turbines [42]. We can then easily obtain $PO\_DerivedRotor$ as shown in Fig. D.7.

A simple path, which is extracted from the model described in Fig. D.7, is shown in Fig. D.8a where $WindSpeed\_Derived$ can be derived from $PO\_DerivedRotor$ which can be derived from $RotorSpeed\_Sensor$.

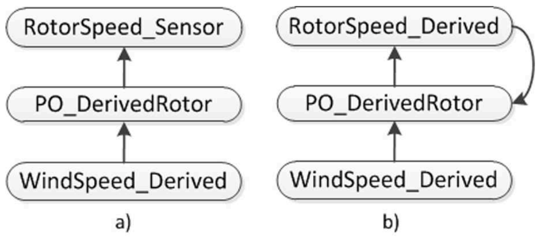


Figure D.8: WindSpeed is derived from PowerOutput and RotorSpeed

If we choose $RotorSpeed\_Derived$ instead of $RotorSpeed\_Sensor$, this leads to a cyclic dependency as shown in Fig. D.8b. Cyclic dependencies have to be avoided, as they cannot be computed.

Fig. D.9 depicts a model which is the extension of the model illustrated in Fig D.7. In order to solve the derivation cycle issue, the transitive closure of the dependency $dependsOn$ should not be reflexive.

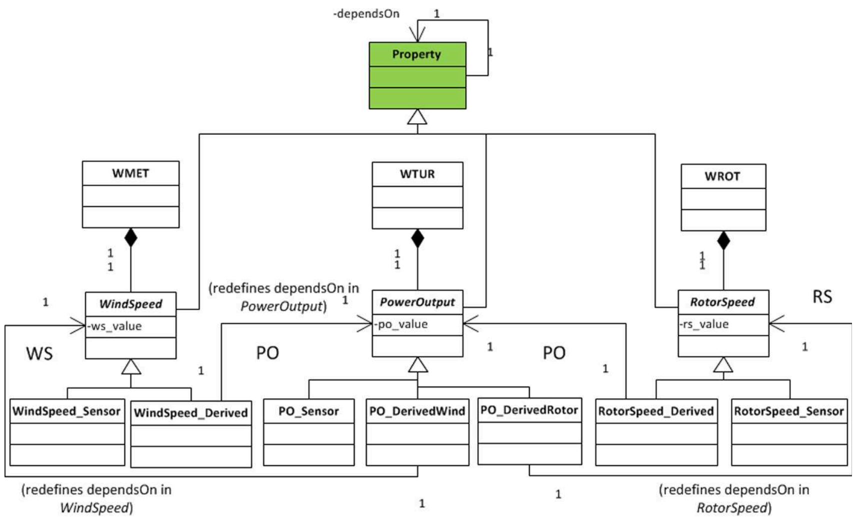


Figure D.9: Solving the cyclic derivation issue in derivative relationships between three parameters

The transitive closure of $dependsOn$ is expressed in OCL as follows:

$context Property$

$inv\ cycleRestriction$: **not** $self.dependsOn.closure()->include(self)$

4.3 Data quality

Data quality can influence the decisions made by organizations. Indeed, wrong decisions can be made because of poor quality data [52, 66]. Data quality describes the characteristics of data and hence gives users a better view on data they want to request for. We consider data quality as metadata. Data quality has several dimensions which are criteria for selecting the most suitable data source according to users' requests. This section presents a solution to the challenge posed in Sect. 3.3.

4.3.1 Data quality dimensions

There are more than 17 data quality dimensions which have been mentioned in literature, e.g., accuracy, completeness, timeliness, consistency, access security, data volume, confidence, and understandability [58, 84, 14, 49]. The most commonly used quality dimensions are *accuracy*, *completeness*, and *timeliness* [44]. The other dimensions such as *confidence*, *value-added*, and *coverage* are only suggested by a couple of studies because these dimensions can be either derived from the other dimensions or applicable only in a few domains. There is no unique definition for each data quality dimension, so we describe the dimensions based on existing definitions and our understanding. Table D.1 shows the notation that we use in our definitions.

Accuracy is defined as how close the observed data are to reality. According to the ISO 5725 standard [26], accuracy consists of precision and trueness.

We assume that the sensors are calibrated, meaning that the trueness is very close to zero. Therefore, we only consider precision as the accuracy in our system. A statistical measure of the precision for a series of repetitive measurements is the standard deviation. Let μ denote the trueness ($\mu = 0$). Thus, the accuracy of data

Table D.1: Table of notation

Symbol	Explanation
D	Data source
R	Reference data source (reality)
N_D	total number of data points in D
N_R	total number of data points in R
d_i	a single data point in D
r_i	real value corresponding to d_i
x_i	d_i - r_i
$t(r_i)$	the moment when the data point i is due
$t(d_i)$	the moment when the data point i is available

source D can be obtained using Eq. (D.5).

$$Acc(D) = \sqrt{\frac{1}{N_D}\sum_{i=1}^{N_D}(x_i - \mu)^2} = \sqrt{\frac{1}{N_D}\sum_{i=1}^{N_D}(d_i - r_i)^2} \tag{D.5}$$

Completeness is defined as the ratio of the number of successful received data points to the number of expected data points. The completeness of the data source D can be calculated using Eq. (D.6).

$$Compl(D) = \frac{N_D}{N_R} \tag{D.6}$$

Timeliness is the average time difference between the moment a data point has been successfully received and the moment it is produced. The timeliness of data source D is calculated using Eq. (D.7).

$$Time(D) = \frac{\sum_{i=0}^{N_D}(t(d_i) - t(r_i))}{N_D} \tag{D.7}$$

4.3.2 Combination and computation of data quality

By combining existing data sources, it is possible to improve the quality of data to meet the user defined requirement. The combination of data sources is defined as the process of constructing a data source from existing data sources. We present three simple methods to combine data quality: D1 (E) D2, D1 $\bigoplus$ D2, and D1 (A) D2.

- D1 (A) D2: taking a conventional average of the data sources D1 and D2.

- D1 $\bigoplus$ D2: use data points from data source D1 if available, otherwise use D2.
- D1 (E) D2: pick up the earliest received data point from either D1 or D2.

Table D.2 gives an overview of all combination methods with data quality dimensions. These methods are used to generate the virtual data source from the real data sources. P(D1) denotes the probability of the event *D1 having data available* and P(D2) denotes the probability of the event *D2 having data available*. $Acc(D1)$ and $Acc(D2)$ are the accuracy (precision) of $D1$ and $D2$, respectively. α is the probability of the event a data point $D1_i$ arrives before a data point $D2_i$.

The following assumptions are made in order to obtain Table D.2. (1) Data sources $D1$ and $D2$ are independent and normally distributed; (2) timeliness $Time(D1)$ and $Time(D2)$ of $D1$ and $D2$ are two independent distributed exponential random variables.

The combination methods have different effects on the data quality dimensions. A quality dimension can increase or decrease depending on a combination method. Table D.3 shows relation the between the combination operations and the data quality dimensions, where (✓) indicates that it can be better than both of D1 and D2, ($-$) means it varies from case to case, and ($\times$) means it is worse than both of D1 and D2.

According to this table, all three methods can increase the completeness. By using the average method, the combined data source would have better accuracy. However, it makes the timeliness become worse. For the $\bigoplus$ method, both the accuracy and timeliness of the combined data source varies from case to case. The (E) method helps to increase the completeness and timeliness, but not the accuracy. If the timeliness is the critical choice, the (E) method is recommended to use.

4.3.3 A data quality-based framework for data source selection

We have developed a framework for data source selection based on data quality dimensions. An overview of the framework is shown in Fig. D.10. The framework offers ways to manage data sources, to insert a new data source, and to provide the best suited data source to users. Due to limitation of space, we cannot describe the prototype in detail. More information about the prototype implementation can be found in [44].

The prototype contains three main parts: web-based client application, integrated

Table D.2: Combination results

Method	Completeness	Accuracy	Timeliness
D1 (A) D2	$\overline{\overline{P(D1)} \, . \, \overline{P(D2)}}$	$\sqrt{\frac{Acc(D1)^2+Acc(D2)^2}{4}}$	$\approx \frac{3}{2} Time(D1)$
D1 $\bigoplus$ D2	$\overline{\overline{P(D1)} \, . \, \overline{P(D2)}}$	$\frac{P(D1)*Acc(D1)+\overline{P(D1)}*P(D2)*Acc(D2)}{P(D1)+\overline{P(D1)}*P(D2)}$	$\frac{Compl(D1)*Time(D1)+\overline{P(D1)}*P(D2)*Time(D2)}{P(D1)+\overline{P(D1)}*P(D2)}$
D1 (E) D2	$\overline{\overline{P(D1)} \, . \, \overline{P(D2)}}$	$\alpha Acc(D1) + \overline{\alpha} Acc(D2)$	$\frac{Time(D1)*Time(D2)}{Time(D1)+Time(D2)}$

Table D.3: Quality combination relations

Combination method	Completeness	Accuracy	Timeliness
D1 (A) D2	✓	✓	×
D1 $\bigoplus$ D2	✓	–	–
D1 (E) D2	✓	–	✓

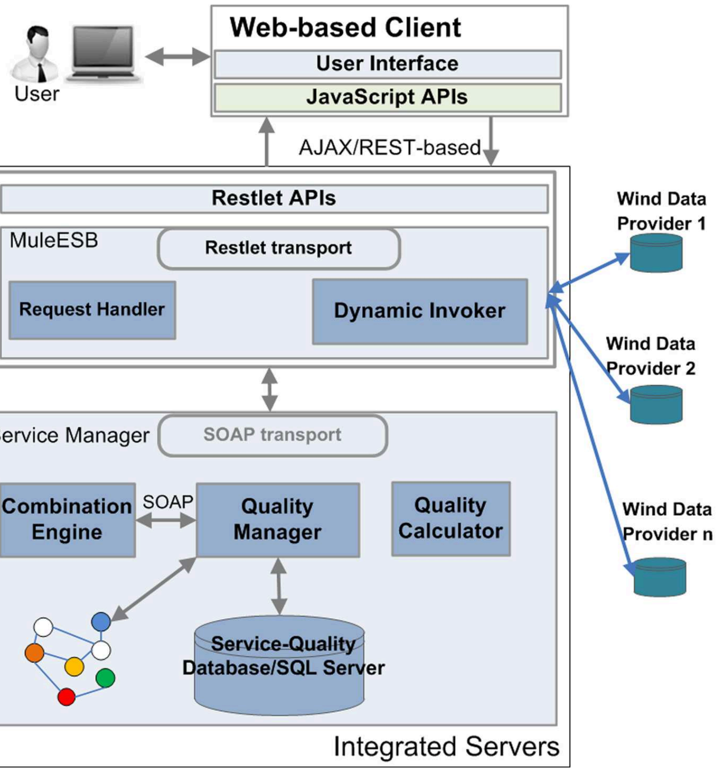


Figure D.10: An overview of a quality-based data source handling framework

servers (IS), and data provider services. The web-based client application receives requests from users and forwards them to the IS. The client is in charge of data visualization in terms of graphs. The IS is responsible for data quality handling and communicating with data providers. The data providers store the data and provide addresses to access those data. The IS consists of an open source enterprise service bus, MuleESB and the *Service Manager* which contains the *Combination Engine*, the *Quality Manager*, the *Quality Calculator*, and the *Service-quality Database*.

5 Discussion and future directions

One reason of having ontologies is to share an understanding of domain concepts between partners who are working in different domains. We have proven the usefulness of having ontologies in smart grids where energy generator, energy providers, consumers need to share the common view on domain concepts.

Many technologies (smart meter, semantic technologies, etc.) are mature enough to be used in building smart grids. But bringing these technologies together to enable smart grids is still a challenging task.

Information and communication security always has a significant role in any information systems and it is not an exception for smart grid systems. The power industry needs to manage not only the power system infrastructure, but also the information infrastructure. The reason is that the power industry increasingly relies on information to operate power systems and many manual operations are being replaced by automation. It is obvious that better decisions can be made by humans or intelligent systems based on available information. However, information needs to be made accessible in a secure way. One way of doing it is to lower the risk by granting access to metadata to only trusted partners.

Metadata provide information about data that are stored in a database without having accessed it [32]. Quality of metadata guarantees that proper sensing resources and data sources are found and data are used properly. The quality of metadata definitely affects the use of data and decisions that are based upon the data. There are several metadata quality criteria that must be taken into consideration such as correctness, completeness, accuracy, consistency, value-added, interpretability. Among these criteria accuracy, completeness, and consistency are the most common criteria for measuring metadata quality in literature [40]. The challenge is among those metadata quality dimensions which ones are the most important and how to check their quality. Another challenge that has not been addressed in this work is tracking provenance of metadata when it comes to metadata combination and enhancement. Besides management of metadata, agreement on the definition of concepts is also an important task since without understanding the definitions, metadata may be misinterpreted or misused. We plan to tackle these challenges in future work.

6 Conclusions

Technologies bring us closer to our vision for living in smart environments. Even though there are available technologies for us, it is still not an easy task to bring all the technologies together. A smart grid is an example of a smart environment. In smart grids, a huge number of smart meters, sensors, smart appliances, and other smart devices are employed and connected to Internet. This leads to issues in handling and processing vast amounts of data, and integrating these devices in a network so that they can communicate with each other through intelligent systems and applications. In this chapter, we have discussed issues related to management of big data metadata in smart grids. Three problems were addressed: concept modeling for knowledge sharing and data exchange, formal description of derived data from concept relations, and data quality handling. We have also proposed solutions and approaches to solving these problems. Some concrete examples within the offshore wind energy were used to demonstrate our points.

This work shows that the semantic technologies are mature enough to be used in the development of smart grids in particular and smart environments in general. The work also proves that data quality can be improved in some cases by combining different data sources that provide measurements about the same physical phenomenon. Relations between concepts not only describe real-world objects/phenomena, but also open several possible paths for calculation and give users the possibility of selecting the most suitable one.

Acknowledgment

This work has been (partially) funded by the Norwegian Centre for Offshore Wind Energy (NORCOWE) under grant 193821/S60 from the Research Council of Norway (RCN). NORCOWE is a consortium with partners from industry and science, hosted by Christian Michelsen Research.

References

[1] Ackoff, R.L.: From data to wisdom. Journal of Applied Systems Analysis **16**, 3–9 (2010)

[2] Antoniou, G., Harmelen, F.v.: Web ontology language: OWL. In: Handbook on Ontologies, International Handbooks on Information Systems, pp. 91–110. Springer Berlin Heidelberg (2009)

[3] Baclawski, K., Kokar, M., Waldinger, R., Kogut, P.: Consistency checking of semantic web ontologies. The Semantic Web - ISWC 2002, pp. 454 – 459 (2002)

[4] Balsters, H.: Modelling database views with derived classes in the UML/OCL-framework. In: UML 2003-The Unified Modeling Language. Modeling Languages and Applications, pp. 295–309. Springer (2003)

[5] Barnaghi, P., Wang, W., Henson, C., Taylor, K.: Semantics for the Internet of things: early progress and back to the future. International Journal on Semantic Web and Information Systems (IJSWIS) **8**(1), 1–21 (2012)

[6] Baumgartner, N., Gottesheim, W., Mitsch, S., Retschitzegger, W., Schwinger, W.: Improving situation awareness in traffic management. In: Proc. Intl. Conf. on Very Large Data Bases (2010)

[7] Bechhofer, S., Van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P., Stein, L., et al.: OWL web ontology language reference. W3C recommendation **10** (2004)

[8] Berners-Lee, T., Fischetti, M.: Weaving the Web: the original design and ultimate destiny of the World Wide Web by its inventor. HarperInformation (2000)

[9] Brédillet, P., Lambert, E., Schultz, E.: CIM, 61850, COSEM standards used in a model driven integration approach to build the smart grid service oriented architecture. In: Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference, pp. 467 –471 (2010)

[10] Bröring, A., Maué, P., Janowicz, K., Nüst, D., Malewski, C.: Semantically-enabled sensor plug & play for the sensor web. Sensors **11**(8), 7568–7605 (2011)

[11] Camacho, E.F., Samad, T., Garcia-Sanz, M., Hiskens, I.: Control for renewable energy and smart grids. The Impact of Control Technology, Control Systems Society pp. 69–88 (2011)

[12] Chen, J., Chen, Y., Du, X., Li, C., Lu, J., Zhao, S., Zhou, X.: Big data challenge: a data management perspective. Frontiers of Computer Science **7**(2), 157–164 (2013)

[13] Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., et al.: The SSN ontology of the W3C semantic sensor network incubator group. Web Seman-

D

tics: Science, Services and Agents on the World Wide Web **17**(0), 25 – 32 (2012)

[14] Datastax Corporation: Big Data: Beyond the Hype (2013). White paper

[15] ETP: Smart Grids - Strategic Deployment Document for Europe's Electricity Networks of the Future (2010)

[16] Geisler, S., Weber, S., Quix, C.: An ontology-based data quality framework for data stream applications. In: 16th International Conference on Information Quality, pp. 145–159 (2011)

[17] Ghazel, M., Toguyéni, A., Bigand, M.: An UML approach for the metamodelling of automated production systems for monitoring purpose. Computers in Industry **55**(3), 283–299 (2004)

[18] Gruber, T.R., et al.: Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human Computer Studies **43**(5), 907–928 (1995)

[19] Hachem, N.I., Qiu, K., Serrao, N., Gennert, M.A.: GaeaPN: A Petri Net model for the management of data and metadata derivations in scientific experiments. Worcester Polytechnic Institute Computer Science Department Technical Report WPI-CS-TR-94 **1** (1994)

[20] Horrocks, I., Patel-Schneider, P., Van Harmelen, F.: 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 (2003)

[21] Huang, K.T., Lee, Y.W., Wang, R.Y.: Quality Information and Knowledge. Prentice Hall PTR (1998)

[22] Iannone, L., Rector, A.L.: Calculations in OWL. In: OWLED (2008)

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

[24] IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications, and Loads. IEEE Std 2030-2011 pp. 1–126 (2011)

[25] Informatica Corporation: Metadata Management for Holistic Data Governance (2013). White Paper

[26] ISO: ISO 5725-2: 1994: Accuracy (trueness and precision) of measurement methods and results-Part 2: methods for the determination of repeatability and reproductibility. International Organization for Standardization (1994)

[27] Kollia, I., Glimm, B., Horrocks, I.: SPARQL query answering over OWL ontologies. In: The Semantic Web: Research and Applications, pp. 382–396. Springer (2011)

[28] Le-Phuoc, D., Nguyen-Mau, H.Q., Parreira, J.X., Hauswirth, M.: A middleware framework for scalable management of linked streams. Web Semantics: Science, Services and Agents on the World Wide Web **16**, 42–51 (2012)

[29] Lee, Y.W., Strong, D.M., Kahn, B.K., Wang, R.Y.: AIMQ: A methodology for information quality assessment. Information and Management **40**(2), 133–146 (2002)

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

[31] Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H.: Big data: The next frontier for innovation, competition, and productivity. Tech. rep., McKinsey Global Institute (2011)

[32] Margaritopoulos, T., Margaritopoulos, M., Mavridis, I., Manitsaris, A.: A conceptual framework for metadata quality assessment. Universitätsverlag Göttingen p. 104 (2008)

[33] Muljadi, E., Pierce, K., Migliore, P.: Control strategy for variable-speed, stall-regulated wind turbines. In: American Control Conference, 1998. Proceedings of the 1998, vol. 3, pp. 1710–1714. IEEE (1998)

[34] Muyeen, S., Tamura, J., Murata, T.: Wind turbine modeling. Stability Augmentation of a Grid-connected Wind Farm pp. 23–65 (2009)

[35] Neuhaus, H., Compton, M.: The semantic sensor network ontology. In: AGILE Workshop on Challenges in Geospatial Data Harmonisation, Hannover, Germany, pp. 1–33 (2009)

[36] Nguyen, T.H., Prinz, A., Friisø, T., Nossum, R.: Smart grid for offshore wind farms: Towards an information model based on the IEC 61400-25 standard. In: Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES, pp. 1–6 (2012).

D

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

[38] NIST: NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 2.0, NIST Special Publication 1108R2 edn. (2012)

[39] O'Connor, M., Das, A.: SQWRL: a query language for OWL. In: Proc. of 6th OWL: Experiences and Directions Workshop OWLED2009 (2009)

[40] Park, J.R.: Metadata quality in digital repositories: A survey of the current state of the art. Cataloging & Classification Quarterly **47**(3-4), 213–228 (2009)

[41] Patel-Schneider, P.F., Hayes, P., Horrocks, I., et al.: OWL web ontology language semantics and abstract syntax. W3C recommendation **10** (2004)

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

[43] Ramirez, R.G., Kulkarni, U.R., Moser, K.A.: Derived data for decision support systems. Decision Support Systems **17**(2), 119–140 (1996)

[44] Rasta, K., Nguyen, T.H., Prinz, A.: A framework for data quality handling in enterprise service bus. In: Innovative Computing Technology (INTECH), 2013 Third International Conference on, pp. 491–497 (2013)

[45] Rossouw, L., Re, G.: Big data–big opportunities. RISK **16**(2) (2012)

[46] Sheth, A., Anantharam, P., Henson, C.: Physical-cyber-social computing: An early 21st century approach. Intelligent Systems, IEEE **28**(1), 78–82 (2013).

[47] Singh, A.: Standards for Smart Grid. International Journal of Engineering Research and Applications (IJERA) (2012)

[48] Sirin, E., Parsia, B.: SPARQL-DL: SPARQL query for OWL-DL. In: OWLED, vol. 258 (2007)

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

[50] Snyder, B., Kaiser, M.J.: Ecological and economic cost-benefit analysis of offshore wind energy. Renewable Energy **34**(6), 1567–1578 (2009)

[51] Solntseff, N., Yezerski, A.: A survey of extensible programming languages. Annual Review in Automatic Programming **7**, 267–307 (1974)

[52] Strong, D.M., Lee, Y.W., Wang, R.Y.: Data duality in context. Communications of the ACM **40**(5), 103–110 (1997)

[53] Studer, R., Benjamins, V., Fensel, D.: Knowledge engineering: principles and methods. Data & Knowledge Engineering **25**(1-2), 161–197 (1998)

[54] Tambouris, E., Manouselis, N., Costopoulou, C.: Metadata for digital collections of e-government resources. Electronic Library, The **25**(2), 176–192 (2007)

[55] Tannenbaum, A.: Metadata solutions: using metamodels, repositories, XML, and enterprise portals to generate information on demand. Addison-Wesley Longman Publishing Co., Inc. (2001)

[56] Vermesan, O., Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., Jubert, I.S., Mazura, M., Harrison, M., Eisenhauer, M., et al.: Internet of things strategic research roadmap. O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, et al., Internet of Things: Global Technological and Societal Trends pp. 9–52 (2011)

[57] Wagner, A., Speiser, S., Harth, A.: Semantic web technologies for a smart energy grid: Requirements and challenges. In: ISWC Posters & Demos (2010)

[58] Wang, R.Y., Strong, D.M.: Beyond accuracy: What data quality means to data consumers. Journal of Management Information Systems **12**(4), pp. 5–33 (1996)

[59] Warmer, J., Kleppe, A.: The object constraint language: getting your models ready for MDA. Addison-Wesley Longman Publishing Co., Inc. (2003)

[60] Xu, H.: Critical success factors for accounting information systems data quality. Ph.D. thesis, University of Southern Queensland (2009)

[61] Zikopoulos, P.C., Eaton, C., DeRoos, D., Deutsch, T., Lapis, G.: Understanding big data. The McGraw-Hill Companies (2012)

[62] Zubcoff, J., Pardillo, J., Trujillo, J.: A UML profile for the conceptual modelling of data-mining with time-series in data warehouses. Information and Software Technology **51**(6), 977–992 (2009)

Appendix E

Paper E

Title	An Approach to Supporting Maintenance of Offshore Wind Turbine Blades
Authors	Trinh Hoang Nguyen[a], Andreas Prinz[a], and Josef Noll[b]
Affiliation(s)	[a] Department of ICT, University of Agder, Grimstad, Norway
	[b] University of Oslo/UNIK, Norway
Journal	Journal of Intelligent Computing
Copyright	©DLINE

An Approach to Supporting Maintenance of Offshore Wind Turbine Blades

Trinh Hoang Nguyen[a,*], Andreas Prinz[a,*], and Josef Noll[b,**]
[a] Department of ICT, University of Agder, Grimstad, Norway
[b] ConnectedLife Group, University Graduate Center, Norway
E-mails: *{trinh.h.nguyen, andreas.prinz}@uia.no
** josef@unik.no

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 reliability-centered 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.

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 [11]. 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 over-maintenance [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 [17, 13]. 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 [3].

This article[1] introduces an approach, namely knowledge-based 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

[1]This article is a revised and expanded version of a paper entitled, "Proactive Maintenance of Offshore Wind Turbine Blades Using Knowledge-based Force Analysis" presented at INTECH 2013 conference, 29 - 31 August, London, UK. ISBN: 978-1-4799-0047-3

2 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 3. Section 5 describes our proposed approach. Section 6 presents a working system that is used to prove the concepts. Section 7 discusses related work. Finally, section 8 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 [16]. 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 [6]. 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 [8]. Mass imbalance is a major source of vibration since it is proportional with wind speed in square (v^2) [7]. 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 [9], i.e. $m_1 * r_1 = m_2 * r_2 = m_3 * r_3$ or $F_{c1} = F_{c2} = F_{c3}$ thus

$$\vec{F}_{c1} + \vec{F}_{c2} + \vec{F}_{c3} = 0 \tag{E.1}$$

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.

E

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 [21]. 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 E.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.

Table E.1: An example of wind turbine rotor information [21]

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

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 [14]. 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. Fig. E.1 represents a semantic network for wind turbine rotor (WROT). WROT, Rotor, RotorSpeed, etc. represent the concepts of the wind domain while *isA*, *isPartOf*, *hasSpeed* 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.

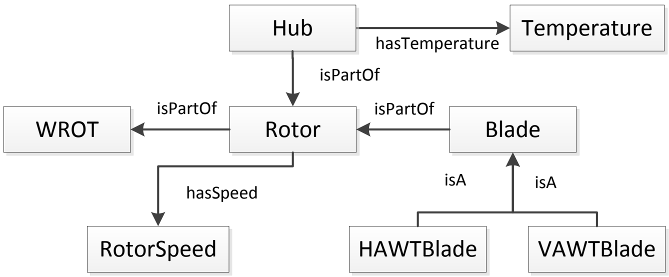


Figure E.1: A semantic network representation of WROT.

(a) IF $Blade\_BL101$ is a $HAWTBlade$ THEN $Blade\_BL101$ is a $Blade$

(b) IF $Blade\_BL101$ is a $HAWTBlade$
AND $wrot$ is a $WROT$
AND $rotor$ is a $WROTRotor$
AND $Blade\_BL101$ is a part of $rotor$
AND $rotor$ is a part of $wrot$
THEN $Blade\_BL101$ is a part of $wrot$

Rule-based knowledge representation systems are especially suitable for reasoning about concrete instance data [14], for example, in the second rule example, *Blade_BL101*, *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 [20]. 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) $Blade\_BL101$ is a blade ($Blade\_BL101 \in Blade$) belongs in ABox, provided that $Blade$ is a concept of an ontology.

(3) $Rotor$ has exactly three blades ($Rotor \sqcap = 3.hasBlade$), 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 [15]. 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 [19, 4, 1]. In OWL, Owl:Thing is a built-in most general class 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., $hasStatus$, $hasPitchAngleSetPoint$. 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 $\mathcal{SHOIN}^{(\mathcal{D})}$ [35]. $\mathcal{SHOIN}^{(D)}$ 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 (WS-I) 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 [10] 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 [2].

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 Fig. E.2.

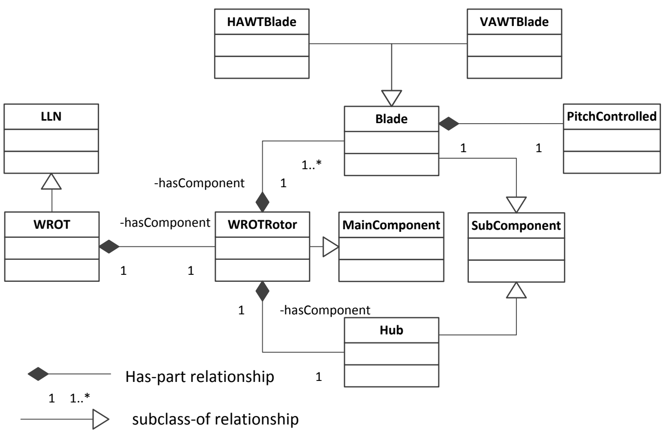


Figure E.2: UML class diagram of WROT.

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 can be classified into passive stall power control and active stall power control systems [32].

We use Protégé[2] to build the WROT ontology. Protégé is an open source ontology editor. It allows users to work with both RDF and OWL. Fig. E.3 depicts an example

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

of how to define the relationships between *WROTRotor* class and its components in Protégé. *WROTRotor* is part of *WROT* and object property *hasComponent* is used to describe their relationships.

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

$hasComponent$ 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 (E.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.

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. Fig. E.4 and Fig. E.5 illustrate our point.

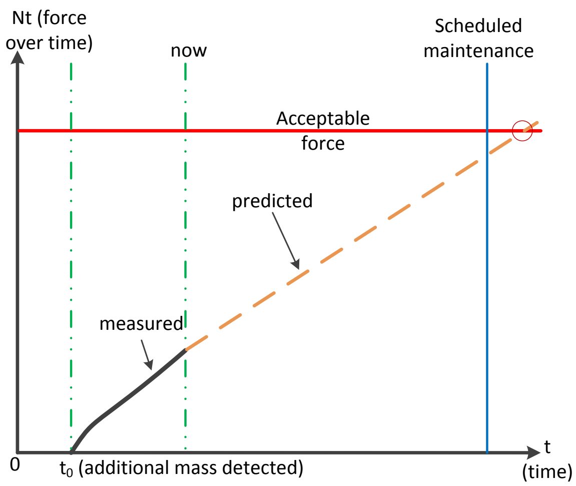


Figure E.4: The wind turbine can continue operating and the maintenance can be carried out as scheduled.

Fig. E.4 shows a normal case where the predicted accumulated force exceeds the normal force after the scheduled maintenance. Fig. E.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:

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

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 (E.3).

$$v_{tip} = \frac{RPM * \pi * D}{60} \tag{E.3}$$

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

E

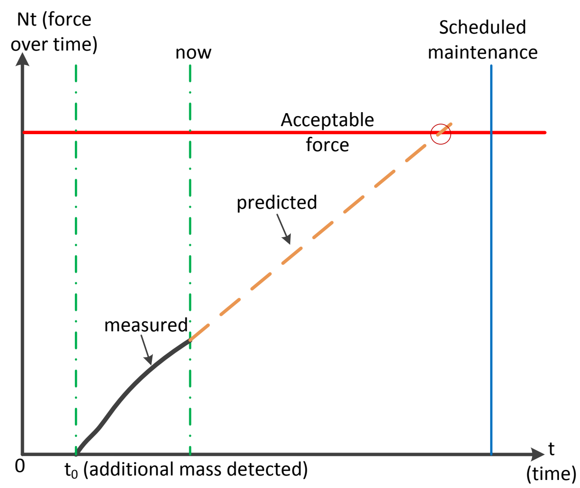


Figure E.5: The maintenance needs to be rescheduled.

E

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

$$v_{tip} = \lambda * v_{wind} \tag{E.4}$$

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. (E.5a) describes the calculation of the force.

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

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

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

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

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

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

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

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

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

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

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

Therefore (E.5a) can be rewritten as follows:

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

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

Fig. E.6 illustrates the architecture of the developed prototype.

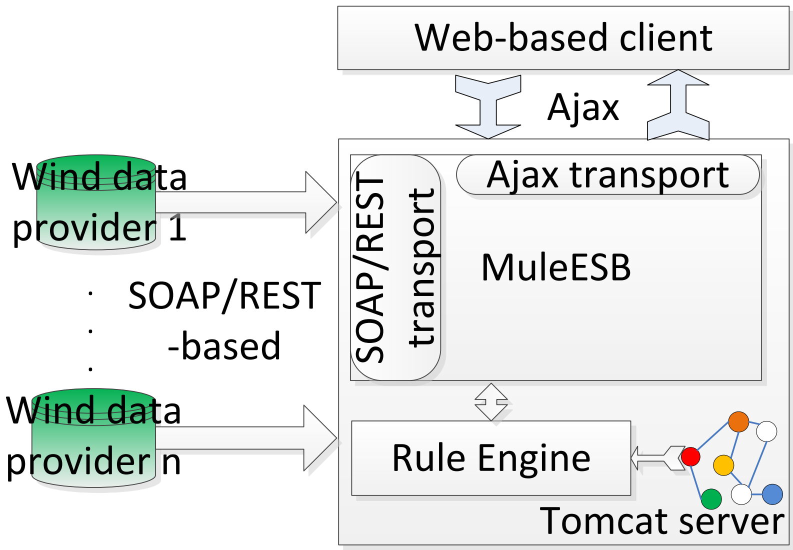


Figure E.6: The architecture of the prototype.

6.1.1 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

[3]http://www.mulesoft.org
[4]http://www.restlet.org/downloads/
[5]http://herzberg.ca.sandia.gov/
[6]http://clarkparsia.com/pellet/

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 Fig. E.7.

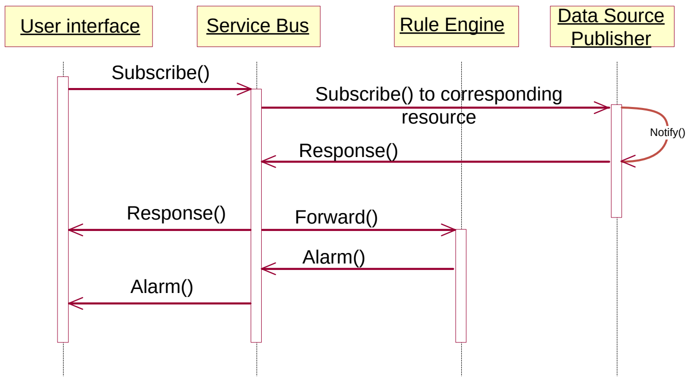


Figure E.7: UML sequence diagram of the prototype.

6.1.2 Rule design

Semantic Web Rule Language (SWRL) is developed based on OWL and Rule Markup Language (RuleML) [18]. 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.

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

E

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

$\rightarrow shutdownWPP(true)$

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.

$WROT(?wrot) \wedge hasComponent(?wrot, ?r) \wedge hasBlade(?r, ?b) \wedge hasAccumulatedForce(?b, ?f)$

$\wedge swrlb : greaterThan(?f, 30000) \rightarrow hasTriggerAlarm(?wrot, true)$

E

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]?

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

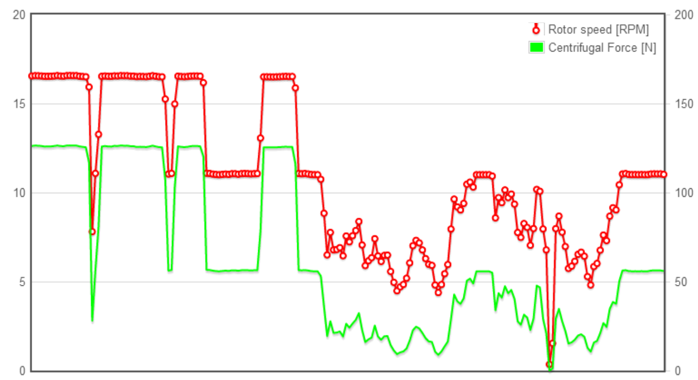


Figure E.8: Rotor speed and centrifugal force.

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

6.2.1 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 Fig. E.9. Generator speed data of two different wind power plants is used in this scenario.

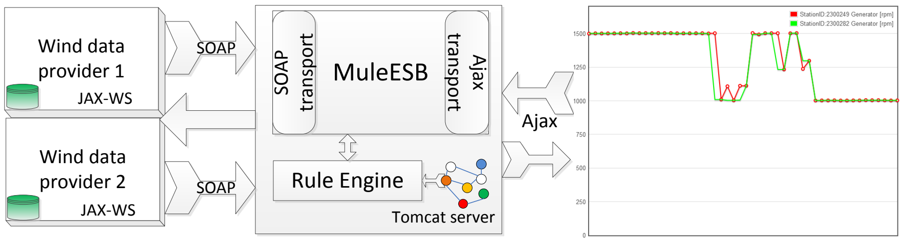
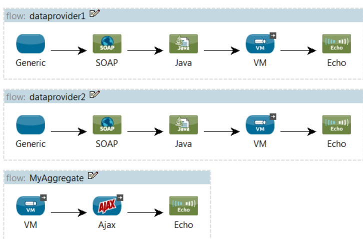


Figure E.9: Real-time monitoring on generator speed of two wind power plants.

The configuration of the flows is shown in Fig. E.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 inbound-endpoint *VM* and sent to the client who subscribes to the data.




```
<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 E.10: Mule configuration for two data sources.

6.2.2 Scenario 2

Fig. E.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.

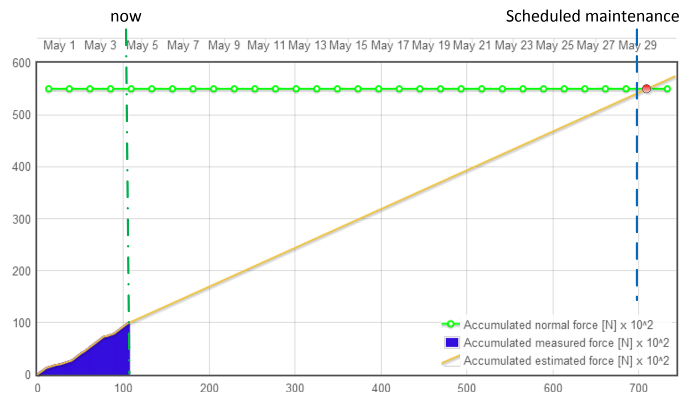


Figure E.11: Scheduled maintenance is on May 29^{th}. The accumulated predicted force will meet the accumulated acceptable one after the scheduled maintenance date.

6.2.3 Scenario 3

Let us take a look at another case where the accumulated predicted force exceeds the accumulated acceptable one before the scheduled maintenance as shown in Fig. E.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. [12] introduced an intelligent system

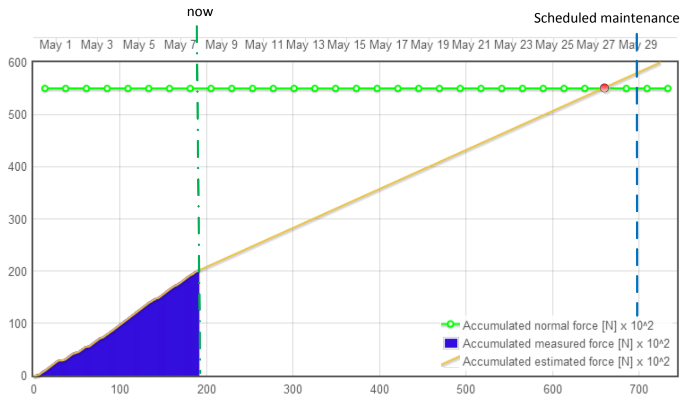


Figure E.12: Scheduled maintenance is on May 29^{th}. The accumulated predicted force will exceed the accumulated acceptable one on May 27^{th}.

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 [5] 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 real-time 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.

Acknowledgement

This work has been (partially) funded by the Norwegian Centre for Offshore Wind Energy (NORCOWE) under grant 193821/S60 from the Research Council of Norway (RCN). NORCOWE is a consortium with partners from industry and science, hosted by Christian Michelsen Research.

References

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

[2] Battle, R., Benson, E.: 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 (2008)

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

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

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

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

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

[8] El-Thalji, I., Liyanage, J.P.: 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, pp. 934–941 (2010)

[9] European Commission: 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. (2005)

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

[11] Fischer, K., Besnard, F., Bertling, L.: Reliability-centered maintenance for wind turbines based on statistical analysis and practical experience. Energy Conversion, IEEE Transactions on **27**(1), 184–195 (2012)

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

[13] García Márquez, F.P., Tobias, A.M., Pinar Pérez, J.M., Papaelias, M.: Condition monitoring of wind turbines: Techniques and methods. Renewable Energy **46**, 169–178(2012)

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

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

[16] Hahn, B.: Reliability assessment of wind turbines in Germany. In: 1999 European Wind Energy Conference, Nice, France, March, pp. 1–5 (1999)

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

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

[19] Horrocks, I., Patel-Schneider, P.F., Van Harmelen, F.: 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 (2003)

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

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

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

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

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

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

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

[27] Maréchaux, J.L.: Combining service-oriented architecture and event-driven architecture using an enterprise service bus. IBM Developer Works pp. 1269–1275 (2006)

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

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

[30] Nilsson, J., Bertling, U.: Maintenance management of wind power systems using condition monitoring systems - life cycle cost analysis for two case studies. Energy Conversion, IEEE Transactions on, **22**(1), 223–229 (2007)

[31] O'connor, M., Knublauch, H., Tu, S., Musen, M.: Writing rules for the semantic web using SWRL and Jess. Protégé With Rules WS, Madrid (2005)

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

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

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

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

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

[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. Schlömer , C. von Stechow (eds)