Final Report

For the

Wind Turbine Analysis Project

May 2018

**PROJECT 15-11-03**

*Authored by:*

*J. Neil Otte, Rahul Rai, Clare Paul, and Barry Smith*

*With contributions by:*

*Ruoyu Yang, Binbin Zhang, and Munira Binti Mohd Ali*

*University at Buffalo, SUNY*

*12 Capen Hall*

*Buffalo, NY 14260-1660*

*Prepared for:*

DMDII

Contents

[Acronyms and Abbreviations 3](#_Toc519264452)

[1 Introduction 4](#_Toc519264453)

[Ontologies 4](#_Toc519264454)

[1.1 Ontologies 4](#_Toc519264455)

[1.1.1 Tiered Ontologies 4](#_Toc519264456)

[1.1.2 The WTBA Ontology 7](#_Toc519264457)

[1.1.3 Artificial Intelligence 8](#_Toc519264458)

[2 Working with Data: Alignment and Ingestion 12](#_Toc519264459)

[2.1 Requirements 14](#_Toc519264460)

[3 Accomplishments and Future Research 15](#_Toc519264461)

[3.1 Accomplishments 15](#_Toc519264462)

[3.1.1 Education 15](#_Toc519264463)

[3.1.2 Lessons Learned 15](#_Toc519264464)

[3.2 Future Research 16](#_Toc519264465)

[4 References 18](#_Toc519264466)

[Appendix 1: Industrial Ontology as a Stepping Stone to AI and Machine Learning 19](#_Toc519264467)

[Deep Learning and its Limits 19](#_Toc519264468)

[5 Appendix 2: Two Sample SPARQL Returns 22](#_Toc519264469)

[Query 1A Return 22](#_Toc519264470)

[Query 1b Return 24](#_Toc519264471)

# Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Acronyms** | **Terms** |
| BFO | Basic Formal Ontology |
| CCO | Common Core Ontologies |
| CHAMP | Coordinated Holistic Alignment of Manufacturing Processes |
| PLC | The Product Life Cycle Ontologies |
| GUI | Graphical User Interface |
| IOF | Industrial Ontologies Foundry |
| IRI | International Resource Identifier |
| EMAE6 | Early Materials and Environments 2006 |
| JSON | JavaScript Object Notation |
| JSON-API | JSON Abstract Public Interface |
| R2RML | RDB to RDF Mapping Language |
| NLP | Natural Language Processing |
| OWL | Web Ontology Language |
| RDF | Resource Description Framework |
| SPARQL | SPARQL Protocol and RDF Query Language |
| URL | Uniform Resource Locator |
| URN | Uniform Resource Name |

# Introduction

Last year, the University at Buffalo, CUBRC, and Cobham Industries, carried out the Coordinated Holistic Alignment of Manufacturing Processes (CHAMP) project with the objective of enabling manufacturing organizations to overcome some of the issues caused by data heterogeneity. The CHAMP project sought to provide remedies to situations in which disparate data sources cannot be used in combination because of differences in their underlying conceptual schemas. This resulted in a suite of ontologies that are now being actively investigated for use by multiple parties within the community of ontologists working to support digital manufacturing, including the Industrial Ontologies Foundry [[1]](#footnote-1)—a NIST-led project whose goal is the creation of a shared platform for coordinated ontology creation to support interoperability of digital manufacturing software and information systems.

Following CHAMP, DMDII expressed an interest in seeing an application of ontologies within the domain of manufacturing. Out of discussions with DMDII, the members of the CHAMP team at the University at Buffalo, and Clare Paul of the Air Force Research Laboratory committed to the present project, which we have titled: Wind Turbine Analysis (WTBA). WTBA builds upon the work of CHAMP, reusing its ontologies in the building of a single application ontology necessary for the performance of a particular material analysis using a particular data set. This project will be of interest to industry professionals. This includes industry professionals who wish to immediately use the framework we have to select materials for wind turbines using our dataset and procedures, as well as the broader community, who are already using or planning to use semantic technology, and are interested in seeing how it may facilitate the discovery of data that meet a set of pre-defined constraints or parameters.

# Ontologies

## Ontologies

We follow the W3C recommendation that ontologies be published for exchange as OWL files (more precisely: as files using the OWL 2 Web Ontology Language). OWL is the current standard and there is an ecosystem of tools developed for creating ontologies within its language.

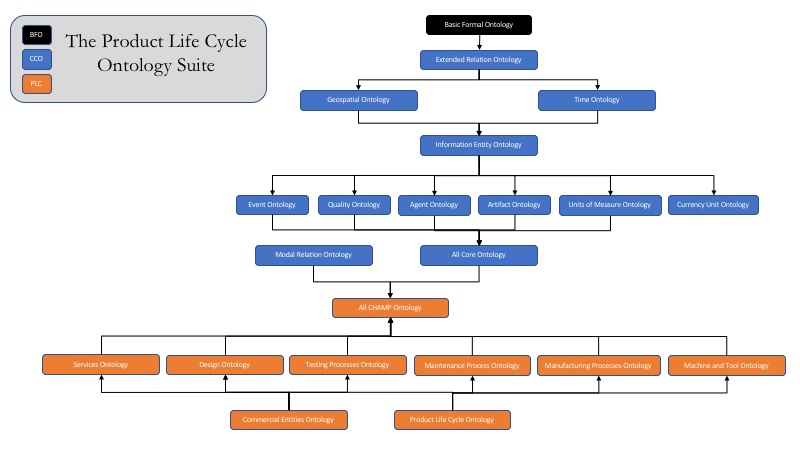
For further discussion of OWL, RDF, JSON, and other programming languages for interest to this domain, see the previously submitted CHAMP report.[[2]](#footnote-2)

### Tiered Ontologies

Contemporary ontology engineering pursues sustainability by working within commonly accepted ontologies that impose downward conformity upon the lower-tiered ontologies that use them. This practice allows ontologies developed independently to achieve a degree of compatibility that —facilitate integration of data which the ontologies are used to describe. This involves a distinction, discussed in detail in the CHAMP report, between Top-Level, Mid-Level, and Domain-Level ontologies.

Briefly, the level of an ontology is determined by the level of generality of the types in reality that it represents. The top-level concerns entities common across all domains, such as objects, times, spaces, processes, and qualities. We use for this purpose Basic Formal Ontology (BFO), which is the most widely used top-level ontology currently under review by the International Standards Organization to form ISO 21838-2. Mid-level ontologies represent those entities that are common across many domains of inquiry. These include currencies, units of measure, treatments of agents and the actions they perform, countries and states, information, and other very large domains. Here, our choice of mid-level ontologies is the suite of Common Core Ontologies (CCO).

Domain level ontologies are more particular still, and often invoke the interests of a particular community. Here, the Product Life-Cycle (PLC) Ontologies produced by the CHAMP project address data representation for industry, providing a presentation of the domains of the PLC including products, material, and data that participate in processes in its successive phases.



*Figure 2 The import structure among the ontologies of the Product Life Cycle suite, the Common Core Ontology suite, and Basic Formal Ontology*

The PLC ontologies are reference ontologies. That is, they are intended for aggressive re-use by multiple other ontologies developed for example by specific enterprises or to support specific software applications. However, user of these ontologies will often be required to extend them by creating application ontologies designed to address specific local needs. This may involve the creation of a small set of subclasses of existing parent classes within the PLC ontologies, or it may involve creating a seperate ontology for a software application that is not represented within the PLC, but which shares some representation with the PLC ontologies.

Application ontologies are the lowest tier of ontology; this is because they frequently contain classes that are particular to a data set, or are constructed in a manner required by some need of a particular end user. For this reason, unlike the PLC ontologies, application ontologies are often intended for one-off support of a particular function or form of analysis and are not intended for general re-use . Where application ontologies are successful, however, then they may serve as use cases for others, but they can do this effectively only if they are developed within a framework of mid-level and top-level ontologies that is widely accepted. Otherwise, when another user takes inspiration from an application ontology developed by some other party, the ontology that user will go on to create will more often than not lack interoperability with the application ontology that inspired it.

The outcomes of the present project should be assessed against this background. It demonstrates how ontologies can be used by a materials analyst to discover materials that will optimize wind turbine blade performance given certain parameters. As such, it represents a steel thread, or execution path through a computer system that can meet the business objectives of a company. In what follows, we detail how the present ontology was used to create data that can be used to perform a material selection procedure, and we also detail how such a tool could be commercially viable. These are immediate benefits of ontologies that demonstrate a clear value and application to DMDII members.

Because this is an application ontology, it is not created to be re-used by extended or adopted for integration by others; rather, it was designed for one specific task and with one specific data source in mind to enable performance of this task. It is also one of the use-cases that the Industrial Ontologies Foundry is presently reviewing as they pursue their long-term vision for ontology engineering coordination and practice.

### The WTBA Ontology

The Wind Turbine Blade Application (WTBA) ontology is a relatively small application ontology consisting of 82 classes. This set of classes is a conservative extension of the 1393 total classes available in the total suite of imported ontologies we are using. These classes are derived from the PLC ontologies, the ontologies of the CCO, as well as BFO and the RO.

The WTBA Ontology was developed to address the following use case:

*As a wind turbine blade engineer, I want to use artificial intelligence (AI) to recommend optimum composite material designs so that I can deliver maximum performance within environmental constraints.*

It includes representations of key parameters, such as:

• Key Performance Parameters – e.g. power efficiency, max deflection, tip velocity

• Environment – e.g. wind loads, gravity induced loads, moisture

• Design Requirements – e.g. sufficient durability, sufficient strength, minimize weight

• Design Approach – e.g. stiffness, strength, durability

• Test Methods – e.g. fatigue testing, monotonic testing, fiber-volume testing

• Engineered Material Product – e.g. carbon fiber, glass fiber, glass matt

• Product Function – e.g. resist mechanical loads, maintain reinforcement orientation

• Product Form – e.g. tow, mat, laminate

• Manufacturing Process – e.g. resin infusion, autoclave cure

### Artificial Intelligence

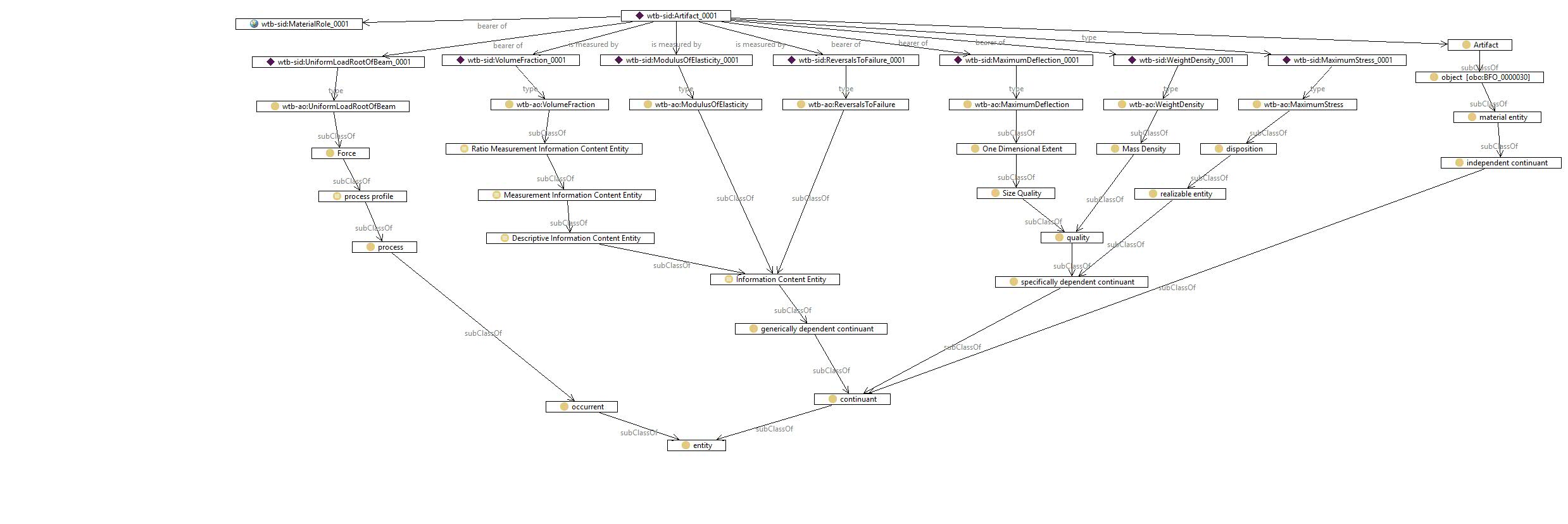
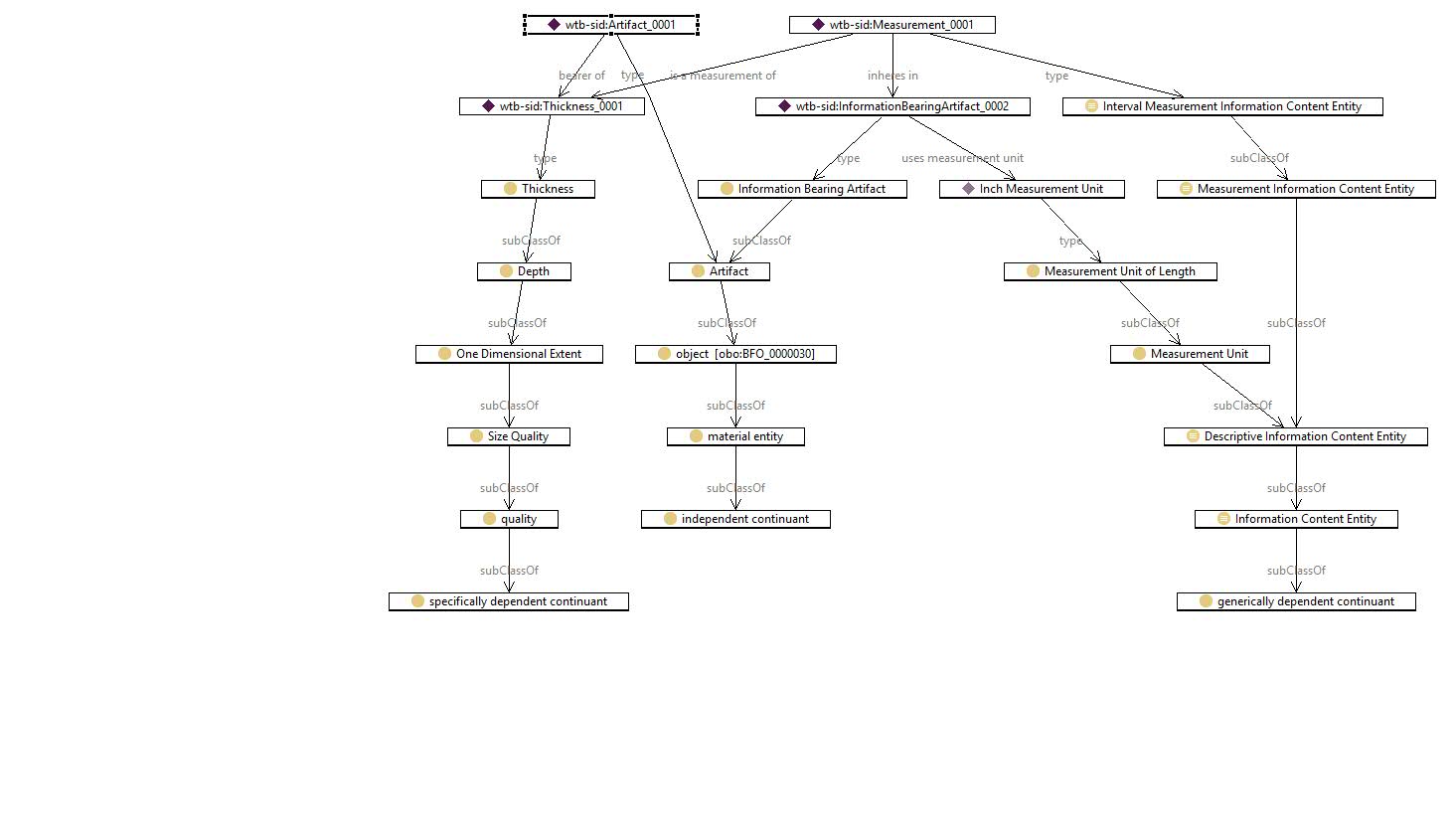
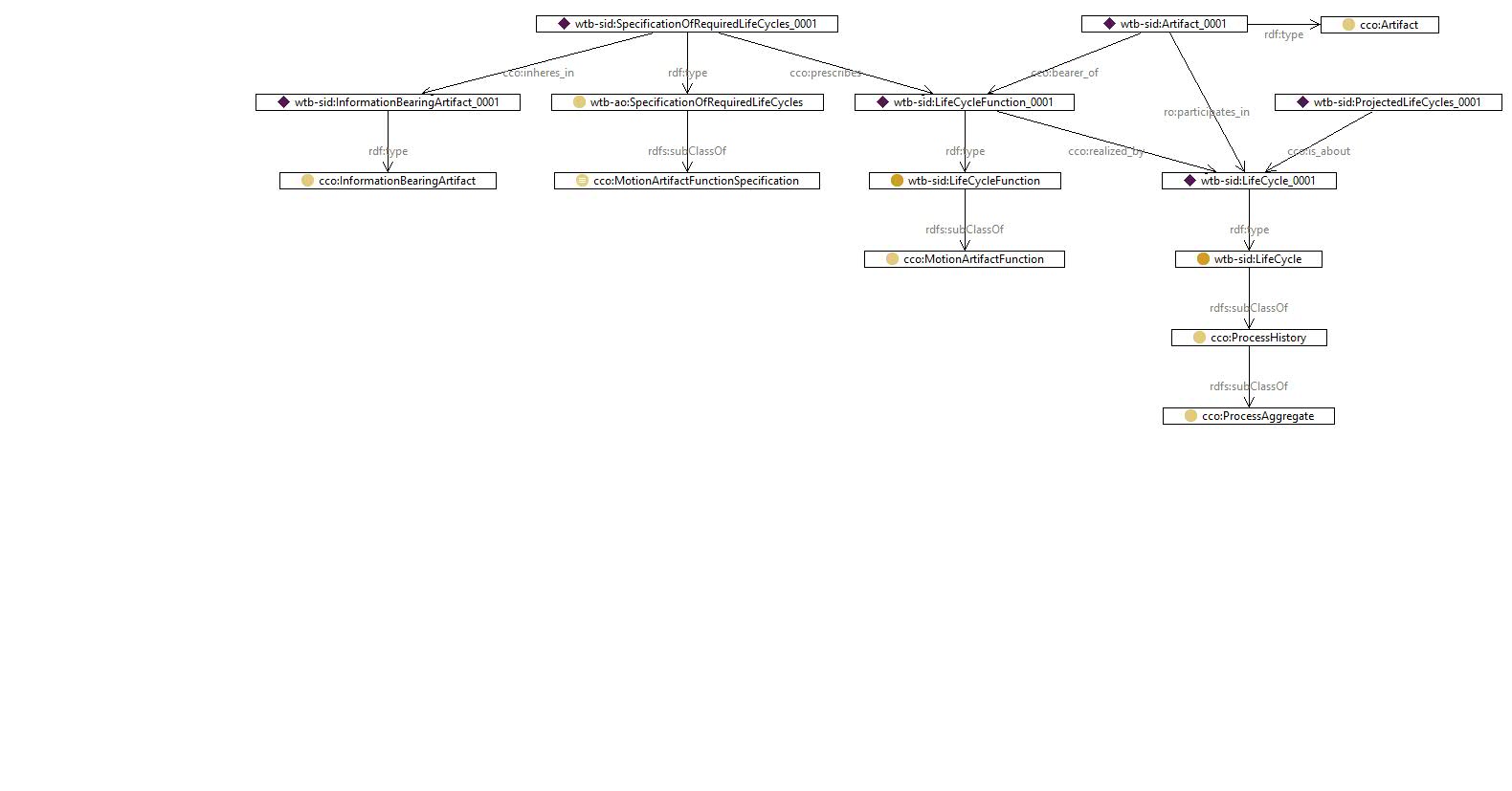
### 

Artificial Intelligence (AI) is an interdisciplinary science whose goal is the creation of technologies that allows computers and machines to function in an intelligent manner where intelligent here is referring to the ability to learn and understand, to solve problems and to make decision []. Simulating intelligence in AI requires intelligence to be broken down into different field such as knowledge representation and reasoning, machine learning, natural language processing, machine perception, computer vision, and robotics.   
  
In this project, we are using knowledge representation (e.g. both the ontology files and the RDF files produced from these files and the dataset) and automated reasoning. Knowledge representation denotes a formally structured representation of reality, while automated reasoning refers to a specific class of programs (e.g. FACT++ and Hermit) , as well as the query language SPARQL, capable of evaluating such structured representations both for consistency, as well as for deductively inferring theorems given asserted axioms within the representation. This includes the ability to infer new instances of classes, as well as new subclasses, based on relations among sets and subsets.

*Form*

Below, we provide sample images that demonstrate the form of the knowledge graph. These images were created using TopBraid Composer—an ontology editor and data management software tool.

In each of the images, the diamonds mark instances of classes, while the circles indicate classes themselves.



# Working with Data: Alignment and Ingestion

*Dataset*

Ontologies consist not of data but rather of classes, axioms, and relations, which are used to tag data to make it discoverable, comparable, and available for certain sorts of retrieval and reasoning. The process of tagging data (or of aligning data to an ontology) can be viewed as a kind of *mapping*. A standard procedure for mapping creates two files, as follows:

1. The mapping itself, which is a software file that specifies that an element in a target data set is an instance of a class in the ontology, and that such an instance bears corresponding relations from the ontology to other instances in the data set. Mapping files may be SPARQL queries, or they may use a language, such as R2RML, which is frequently employed to create mappings between relational and graph databases.
2. The knowledge graph that is produced by the mapping process and which is commonly saved in RDF or JSON. This amounts to a set of triples that conform to the semantics of the ontology. This knowledge graph is then stored in a database where it is queried using SPARQL.

For this project, we used the dataset from an open source database for wind turbine blade materials data at Sandia National Lab at Montana State University.[[3]](#footnote-3) This materials data set includes, in particular, data pertaining to different types of reinforcement and matrix materials. In order to avoid duplicating unnecessary effort, we selected to map only those components necessary for our queries of interest, though the procedures we describe here would enable further uses of the ontology to enable other queries of interest.

The mapping itself was performed by first creating a set of dummy instances of classes in the ontology, where these instances had the form necessary for the mapping. Following this, TopBraid Composer was used to construct a series of SPARQL construct queries, which took data from the Sandia National Lab data set (in excel format) and created an RDF graphical representation of this data.

*SPARQL Queries*

Queries were then developed that conformed to the key performance parameters used by a wind turbine engineer in designing a turbine blade. These queries allowed data that took into account a variety of conditions, including material type, fiber direction, gauge size, the resin used on the material, the maximum stress applied to the material during testing, and many other variables. Such queries draw upon the computational semantics of the ontologies and SPARQL to look for data instances in the graph that satisfy the constraints of the query. This allowed, for instance, the stipulation of particular conditions, and the exploration of data over a number of scenarios involving the use within different environment of different composite materials, ply orientations, reinforcement and matrix materials, and so forth.

For example, to find the variation of the frequency (Hz) and the maximum stress (Mpa) applied to a specimen; according to the material of which it is a type; its fiber direction; and gauge section width and thickness, the following SPARQL query was used:

**Query 1.**

|  |
| --- |
| PREFIX emae06: <http://example.com/emae06#>  PREFIXxsd: <http://www.w3.org/2001/XMLSchema#>  SELECT DISTINCT ?Material ?Resin  ?ReinforcementFiber?ZeroDegreeDirectionFiber\_pct?FortyFiveDegreeDirectionFiber\_pct?NinetyDegreeDirectionFiber\_pct?OtherDegreeDirectionFiber\_pct?GaugeSectionWidth\_mm?GaugeSectionThickness\_mm?MaximumStress\_MPa?FatigueTestFrequency\_Hz  WHERE {  ?a emae06:sshUsesMaterial ?Material .  ?a emae06:sshResin ?Resin .  ?a emae06:sshZeroDegreeReinforcementFabric ?ReinforcementFiber.  ?a emae06:sshPctInZeroDegDirection ?ZeroDegreeDirectionFiber\_pct.  ?a emae06:sshPctInFortyFiveDegreeDirection ?FortyFiveDegreeDirectionFiber\_pct.  ?a emae06:sshPctInNinetyDegreeDirection ?NinetyDegreeDirectionFiber\_pct.  ?a emae06:sshPctInOtherDirection ?OtherDegreeDirectionFiber\_pct.  ?a emae06:sshGaugeSectionThickness ?GaugeSectionThickness\_mm.  ?a emae06:sshGageSectionWidth ?GaugeSectionWidth\_mm.  ?a emae06:sshMaxStress ?MaximumStress\_MPa.  ?a emae06:sshFatigueTestFrequency ?FatigueTestFrequency\_Hz .  }  ORDER BY ?Material ?Resin ?ReinforcementFiber?ZeroDegreeDirectionFiber\_pct?FortyFiveDegreeDirectionFiber\_pct?NinetyDegreeDirectionFiber\_pct?OtherDegreeDirectionFiber\_pct?GaugeSectionWidth\_mm?GaugeSectionThickness\_mm |

Note, the prefix ‘emae06’, short for the name of the MSU data set ‘Early Materials and Environments 2006’, denotes the base URI of the application ontology.

A complete list of SPARQL queries is provided at the github repository[[4]](#footnote-4) where they are available for re-use, along with the original RDF file. We list below in natural language a sample of other queries that we created:

|  |
| --- |
| **Query 1**  According to the material type, its fiber direction, and the specimen’s dimensions, find the variation in the frequency and the maximum stress applied to the specimen.  **Query 1a**: Filter the search by the material’s layer structure.  **Query 1b**: Filter the search by the material’s name[[5]](#footnote-5).  **Query 2**  According to the material type, its fiber direction, and the specimen’s dimensions, find the variation of the maximum strain percentage and the load cycle time applied to the specimen.  **Query 2a**: Filter the search by the material’s layer structure.  **Query 2b**: Filter the search by the material’s name.  **Query 3**  According to the material type, its fiber direction, and the specimen’s dimensions, find the variation of the stress ratio (min stress/max stress) and the load cycle time applied to the specimen.  **Query 3a**: Filter the search by the material’s layer structure.  **Query 3b**: Filter the search by the material’s name. |

## 

## Requirements

The mid-level ontologies in the PLC suite that were used in this project were vetted in three different ways. First, the ontologies were validated by comparing the terms in the ontologies to the terms in the dataset and confirming that there is adequate coverage in the ontology for the dataset. Second, ontology reasoners such as FaCT++ and Hermit were used to confirm the consistency[[6]](#footnote-6) of the ontology classes and axioms and conformance to semantic technology standards. Third, the ontology content was assessed for its conformity to the principles of the BFO framework.

# Accomplishments and Future Research

## Accomplishments

### Education

For one semester, the WTBA project funded three graduate students, two of whom are from the engineering school at the University of Buffalo.

In addition to the opportunities offered to students, work on the WTBA led to collaborations with a number of academic partners, who are now, with the University at Buffalo, involved actively in the Industrial Ontologies Foundry.

These institutions and their lead members include:

* INP-ENIT, University of Toulouse (Hedi Karray)
* Clemson University (Venkat Krovi)
* École polytechnique fédérale de Lausanne (Dimitris Kiritsis)
* Loughborough University, UK (Bob Young)
* National Center for Ontological Research (Kemper Lewis, Rahul Rai, and Barry Smith)
* Penn State (Timothy Simpson)
* Texas State (Farhad Ameri)
* UMass Amherst (Ian Grosse)
* University of Toronto (Michael Grüninger)

### 

### Lessons Learned

For the students attached to the project, their primary focus has been ontology engineering research and the application of design principles in building OWL ontologies using BFO principles. This project presented them with a nice transition into the realm of engineering, where the ontologies they have built are now applied to data within a particular project. This involved becoming significantly more familiar with running servers and writing SPARQL queries.

The application ontology design itself neatly falls out of the work on the PLC ontologies, which the team had spent a good deal of time perfecting. This project, indeed, underscores the value of working with mid-level ontologies like the CCO and the PLC, because working within such a framework makes very clear to the application ontology engineer how to create new classes for particular applications in a manner that may be understood well by others. We were thus able to avoid many of the basic errors that plague ontologies and data models within the realm of industry, such as those detailed by Smith (2006) concerning the ISO standard 15926 (“Lifecycle Integration of Process Plant Data Including Oil and Gas Production Facilities”).

For Clare Paul, who provided the technical expertise necessary to the completion of this project, the project provided an opportunity to become more familiar with the basic representation of measurement data provided by the Common Core Ontologies. This familiarity will be valuable in the future work to occur on the materials property ontology, whose alignment with the PLC and CCO ontologies is an ongoing effort.

## Future Research

DMDII members are clearly concerned with the commercial viability of this project. The primary steps for using the deliverables of this project are relatively simple for most technical experts.

They include:

1. Downloading and installing an RDF triple store, such as Apache Jena
2. Loading RDF files into the server
3. Running SPARQL queries in the server endpoint, or writing other queries of interest.

However, in order to be truly commercially viable, even these steps might prove off-putting to many DMDII members who are unaccustomed to running a local server and understanding SPARQL syntax.

An alternative is to develop a graphical user-interface that presents drop-down options to a user, allowing them to select what they wish to find and the constraints they wish to place on their search. Such forms can then write SPARQL queries and run these queries against a backend server without the user needing to understand SPARQL, RDF, or any of the semantic technologies involved.

One example of such a tool is OWL2TL[[7]](#footnote-7). This open, online tool presents a series of fields to users, whose inputs are used to construct a SPARQL query that queries the annotation properties of ontology files. The SPARQL query is run on the backend of the website. This would be one way of making the sort of system we have built considerably more simple and user-friendly.

# References

[1] Arp, R., Smith, B. and Spear, A., 2015. Building Ontologies with Basic Formal Ontologies.

[2] Frechette, S.P., 2011. Model based enterprise for manufacturing. Duffie, ed., Omnipress, Madison, WI

[3] Otte, J. N., Ruttenberg. A. “BFO: Basic Formal Ontology”. *Applied Ontology* [Forthcoming]

[4] Pettey, C, 2016, 5 Ugly Truths About Postmodern ERP (Gartner)<https://www.gartner.com/smarterwithgartner/5-ugly-truths-about-postmodern-erp/>

[5] Shen, W, et al, 2008, Computer supported collaborative design: Retrospective and perspective, Computers in Industry, 59/9: 855-862.

[6] Smith, B. 2006. “Against Idiosyncrasy in Ontology Development”. B. Bennett and C. Fellbaum (Eds.), *Formal Ontology and Information Systems*, (FOIS 2006), Amsterdam: IOS Press, 2006, 15-26. Preprint version available here: <http://ontology.buffalo.edu/bfo/west.pdf>

[7] Smith, B. and Ceusters, W., 2010. Ontological realism: A methodology for coordinated evolution of scientific ontologies. *Applied ontology*, *5*(3-4), pp.139-188.

[8] Smith, Barry et al. “The OBO Foundry: Coordinated Evolution of Ontologies to Support Biomedical Data Integration.” *Nature biotechnology* 25.11 (2007): 1251. *PMC*. Web. 10 Jan. 2018.

[9] Furini F, Rai R, Smith B, Colombo G, Krovi V. Development of a Manufacturing Ontology for Functionally Graded Materials. ASME. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 1B: 36th Computers and Information in Engineering Conference ():V01BT02A030. doi:10.1115/DETC2016-59964.

[10] Negnevisttsky M., 2011. Artificial Intelligence A Guide to Intelligent System.

# Appendix 1: Industrial Ontology as a Stepping Stone to AI and Machine Learning

By ‘AI application’ we mean a computer program that can create an output in response to an external input in a way that is similar to the ways in which humans characteristically react to corresponding inputs. In this context, the inputs include the data found in the SANDIA database, as well as the specified parameters and goals of a designer: the output is the inference to an optimal design solution. The webform-based application sketched in the aforementioned can be regarded as a limited case of AI — limited in the sense that it can simulate only a narrow set of human cognitive processes, but still intelligent, since it is performing complex operations that yield outputs from the given inputs of the sort which human beings characteristically achieve albeit with much greater investments of time and effort.

The framework is intelligent also in the sense that, in contrast to traditional relational database-driven approaches, it is highly flexible and highly extendible. The same technology can be adjusted in routine ways for a range of different applications and also for a range of different databases. The latter can be combined and extended with a much higher degree of flexibility than is the case with more traditional technology.

## Deep Learning and its Limits

Today’s standard for AI applications rests on an approach called *agnostic deep machine learning*, which requires very large data sets (of the order of 1012 data points per problem) to be made available for analysis. The data points are input-output tuples representing the ways humans react to some specific type of situation, and these tuples are used to train stochastic models that mimic human behavior in situations of identical type.

Such models are ‘agnostic’ in the sense that they do not rely on any prior knowledge about the task at hand. All AI approaches, including agnostic deep machine learning, seek to develop algorithms that will allow the computer to discover solutions well-defined mathematical problems of a quite specific sort. To discover such solutions however, the mathematics involved requires huge amounts of data of quite specific sorts. In most cases, however, such data are unobtainable.

This is for a variety of reasons, including:

Quantities of data of the required magnitudes are unobtainable; the requisite quantities may indeed be obtainable by large enterprises such as Google or Amazon or the NSA; but engineering companies and of engineering research organizations such as SANDI have much more narrowly focused sets of data sources and this restricts the amount of data that is available.

The data cannot be aggregated in the ways the algorithms require because they are represented in heterogeneous ways – for example, because the data has been compiled by different groups using different coding systems, or because the procedures used by a single group have been evolving over time. Problems of this sort arise wherever data is accumulating. One of the most frequent complaints of managers in large corporations is: “We created this huge and expensive data lake and now you are telling me you cannot analyse the data?”

Note that problem 2. can still arise even where data collections of the needed magnitudes are obtainable.

Further problems arise in virtue of the fact that the outputs of machine learning are always approximations. In mission-critical environments, however (for example, in environments involving autonomous vehicles), exact outputs are required. Problems arise also because, even where large quantities of information are available, it may be that the information is for one reason or another skewed. Data relating to human interactions of specific types may be missing, so that the models may yield behaviors for those interactions of sometimes drastically mistaken sorts. (Training sets for autonomous vehicle software, for example, may lack data pertaining to specific sorts of very rare accident-causing situations.)

This leads to a further problem, relating to the fact that the models themselves operate as a black box, which means that:

It is impossible to predict how the models will respond to given inputs, and thus impossible to determine how given outputs were arrived at. This means that it we are typically unable to identify the causes of erroneous behavior.

We can see clear why 3. is a problem by considering cases of liability. Who or what is responsible when an AI application yields an erroneous output? It is in light of such liability concerns in relation to machine learning applications in medicine, engineering, and military contexts that the idea of ‘meaningful AI’ has arisen. A meaningful AI application would be one for which it would be possible determine how outputs are generated.

Lessons learned from the DMDII Wind Turbine Ontology Project

SANDIA is a large database. But it is not yet a case of ‘big data’ of the sort that would allow deep machine learning of the sort described here.

Our SPARQL endpoint case study illustrating reasoning with the SANDIA data is, at the same time, a case study on how a machine learning approach to meaningful AI might be achieved in the engineering domain. Consider again the three problems we identified:

Too little data.

Data cannot be aggregated.

Black box AI creates liability issues.

For each of these problems, the ontology work we have performed as a basis for reasoning with the SANDIA data provides a stepping stone of a type that will be needed to move to a situation in which a machine learning approach to meaningful AI can be achieved in the engineering domain.

As concerns 1. and 2., our Wind Turbine Ontology shows how highly disparate data can be aggregated into large knowledge graphs to which – again, given the necessary quantities of aggregated data – deep learning methodologies can be applied.

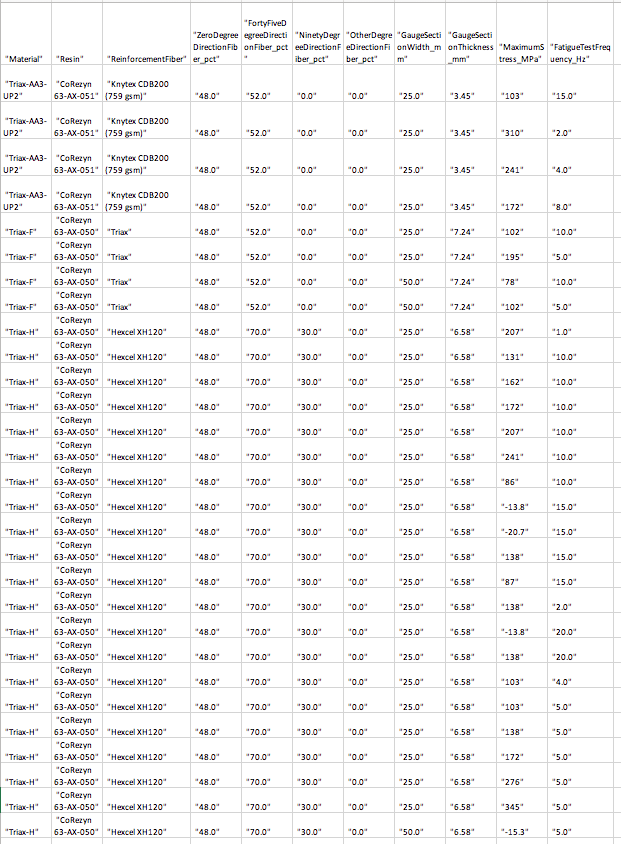
As concerns 3., our ontology approach relies on the systematic tagging of all data points with ontology terms, where the latter are provided with definitions. These definitions exist in two forms – in natural language to make them intelligible to human users, and in formal languages such as the W3C Web Ontology Language (OWL) to enable computing with the data along the lines we demonstrated in our case study. The definitions provide the vehicle through which tagging with ontology terms provides a basis for identifying meaning in the knowledge graphs we produce. The idea of introducing meaning of this sort is already being explored as one potential avenue for the creation of meaningful AI in military contexts.

Ontology tagging can provide an avenue also for embedding prior knowledge – for instance, general knowledge about the physics of materials, about aerodynamics, or about the capabilities of turbine blades of specific shapes or sizes. First, the definitions in the ontologies themselves replicate the definitions of the corresponding terms used in the empirical sciences. Second, to express the knowledge deriving from such sciences in a way that will allow incorporation into an AI application will require the use of these very terms. The ways in which the CCO and PLC ontologies have been developed in such a way as to be interoperable from the start provides a terminological architecture for such incorporation. Moreover, the hierarchy of the ontologies mimics the hierarchical organization of the sciences themselves, where all general knowledge pertaining to general laws of physics or chemistry provides the high-level starting points for scientific subdisciplines at successively lower levels.

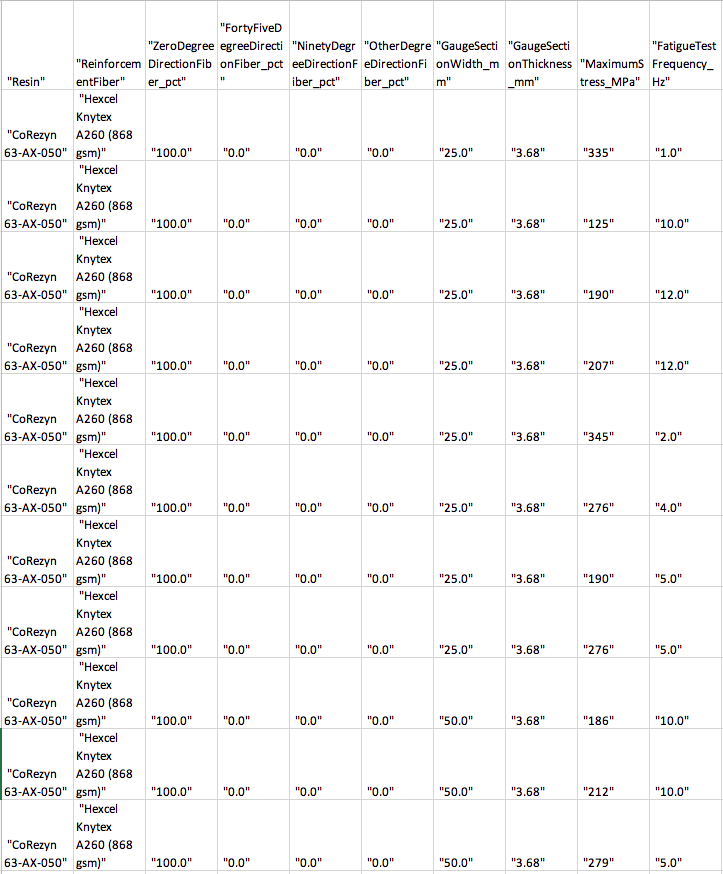
The CCO ontologies are already being used in conjunction with machine learning in an AFRL and DARPA Hallmark Program[[8]](#footnote-8) to develop a platform upon which machine learning applications can function to provide space situational awareness. The ontologies provide the common vocabulary for translating between these various applications so that they can be interoperable.

# Appendix 2: Two Sample SPARQL Returns

## Query 1A Return



## Query 1b Return



1. The website of the IOF may be accessed here: <https://sites.google.com/view/industrialontologies/home> [↑](#footnote-ref-1)
2. The final CHAMP report may be accessed here: https://github.com/NCOR-US/CHAMP/blob/master/CHAMP%20Base%20Year%20Final%20Technical%20Report.docx [↑](#footnote-ref-2)
3. The dataset is described and may be accessed here: http://energy.sandia.gov/energy/renewable-energy/wind-power/blade-reliability/blade-materials-and-substructures-testing/ [↑](#footnote-ref-3)
4. All files related to the Wind Turbine Project are available here: <https://github.com/NCOR-US/WindTurbineProject> [↑](#footnote-ref-4)
5. Note: We include in this report the returned results for queries 1a and 1b (see Appendix). For a full list of all returns, see the project repository on github in fn 4. [↑](#footnote-ref-5)
6. For an explanation of consistency in the context of ontologies, see the final CHAMP report. [↑](#footnote-ref-6)
7. OWL2TL may be accessed here: https://owl2tl.com. [↑](#footnote-ref-7)
8. https://www.darpa.mil/program/hallmark [↑](#footnote-ref-8)