Enriching Analytics Models with Domain Knowledge for Smart Manufacturing Data Analysis

Heng Zhang[a]*, Utpal Roy[a] and Yung-Tsun Tina Lee[b]

[a] *Department of Mechanical and Aerospace Engineering, Syracuse University, Syracuse, NY, USA;*

[b] *Systems Integration Division, National Institute of Standards and Technology, Gaithersburg, MD, USA*

*Corresponding author. Email: hzhang33@syr.edu

Enriching Analytics Models with Domain Knowledge for Smart Manufacturing Data Analysis

Today, data analytics plays an important role in Smart Manufacturing decision making. Domain knowledge is very important to support the development of analytics models. However, in today's data analytics projects, domain knowledge is only documented, but not properly captured and integrated with analytics models. This raises problems in interoperability and traceability of the relevant domain knowledge that is used to develop analytics models. To address these problems, this paper proposes a methodology to enrich analytics models with domain knowledge. To illustrate the proposed methodology, a case study is introduced to demonstrate the utilization of the enriched analytics model to support the development of a Bayesian Network model. The case study shows that the utilization of an enriched analytics model improves the efficiency in developing the Bayesian Network model.

Keywords: Smart Manufacturing, Data Analytics, Domain Knowledge, Interoperability, Traceability, Bayesian Network

1. Introduction

Due to advances in information technologies and artificial intelligence, the Smart Manufacturing (SM) concept has emerged to lead a new paradigm of manufacturing systems. In such systems, data analytics can play an important role to turn data into valuable insights to assist SM decision making. To successfully perform data analytics, domain knowledge is required to support the development of analytics models. Kopanas et al. (2002) investigated the role of domain knowledge in industrial data analytics projects. They concluded that the use of domain knowledge is crucial in all phases of a data analytics project – problem definition, data understanding, data pre-processing, data mining, evaluation of the analytics model and deploying the developed analytics model. However, in today's data analytics projects, the relevant domain knowledge, which is used in developing an analytics model, is only documented, but not properly captured and integrated with the analytics models.

The lack of properly captured and integrated domain knowledge raises an issue in the interoperability of the domain knowledge for developing an analytics model. Currently, there are two standards that support analytics models' interoperability – PMML (Predictive Model Markup Language) (Data Mining Group, 2016) and PFA (Portable Format for Analytics) (Data Mining Group, 2015). The PMML formally represents analytics models to allow the exchange of analytics models between data analytics applications. The PFA covers the main functionalities of the PMML and focuses on the deployment of analytics models by streamlining the entire scoring flow. However, these standards only capture information that is related to the final stages of data analytics projects. They do not possess the capability to capture the domain knowledge that is used in the early phases of data analytics projects. The information exchange between domain experts and data analysts about the application domain relies solely on vocal discussions and written document exchange. To improve the efficiency of the information exchange in SM environment, this interoperability problem must be addressed.

Another issue that is caused by the lack of properly captured and integrated domain knowledge lies in the traceability of analytics models. Without domain knowledge being properly captured and integrated, no software tools can process and understand the documented natural language-based knowledge. This brings difficulties in carrying out data analytics projects for SM systems. As more data is collected from an SM system (as more sensors to be plugged), an analytics model that has already been developed for the SM system needs to be updated accordingly. This calls for the traceability of the domain knowledge that is used to develop the analytics model because the modification or the re-development of the analytics model needs the previous knowledge. To bring understanding of analytics models to downstream activities, the knowledge traceability problem must be solved.

The contribution of this paper is twofold. First, we propose to formalize domain knowledge to support the development of analytics models. Second, to allow the exchange of the domain knowledge with analytics models, we propose to integrate the formalized domain knowledge with analytics models by semantically connecting them. This paper is organized as follows: Section 2 reviews the related studies about integrating domain knowledge into data analytics. Section 3 describes the general framework to enrich analytics model with domain knowledge. To illustrate the proposed idea in detail, the development of a Bayesian Network (BN) model to predict energy consumption of injection moulding processes is investigated. So, section 4 first briefly describes this data analytics project without using the proposed enriched analytics model. Then, it elaborates the details of the development and utilization of the enriched analytics model for developing the BN model. Finally, section 5 summarizes the paper.

2. Related Work

Researchers have previously tried to integrate domain knowledge and engineering models. For example, Kim et al. (2017) proposed a local model calibration approach and a local model averaging approach to incorporating domain expert knowledge into engineering process models. However, these approaches lack model interoperability which is required by SM. Kusiak (2018) mentioned that data-driven modelling and enterprise interoperability are expected to become the pillars of the future of SM. But the interoperability of analytics models was not discussed in his work. Dotoli et al. (2018) pointed out that big data analysis and semantic models are crucial for factory automation. Similarly, they did not consider the integration between analytics models and semantic models. Palmer et al. (2018) presented a comprehensive manufacturing reference ontology to support manufacturing knowledge's interoperability. However,

computational models for manufacturing decision-making, especially data-driven models, were not included.

Some research was carried out in bridging the semantic gap between data and analytics models. For example, Johnson et al. (2010) proposed using an ontology to capture the domain concepts which were used to represent important variables for learning a decision tree. Although this was a good attempt in using ontologies to capture domain concepts for learning a decision tree, this study did not formalize the rules for data processing in the ontology. It also did not explicitly represent the decision tree and semantically connect the decision tree to the ontology. Trappey et al. (2013) developed a knowledge management approach using ontology-based artificial neural networks to automatically classify and search documents. In their research, the domain ontology was used as a bridge to map the concepts related to the documents to the input nodes of the neural network. However, there was no formal representations of the neural network so that there was no semantic links between the domain ontology and the neural network. Similar research on formalizing domain knowledge to bridge semantic gaps had been performed by Perez-Rey et al. (2006), Sinha and Zhao (2008), Munger et al. (2015), and Arena et al. (2018), etc. They also have problems in semantically integrating formal domain knowledge with analytics models.

There are also studies on using domain knowledge to construct analytics models. Campos and Castellano (2007) proposed learning a BN structure by specifying the structural restrictions from expert knowledge. However, no specific domain knowledge formalization and integration were shown in this research. Lechevalier et al. (2016) introduced a domain-specific modelling approach to integrate a manufacturing system model with data analytics to facilitate effective and efficient data analytics in manufacturing systems. In this research, although the manufacturing domain knowledge

was captured, and the knowledge was used in creating a Neural Network structure, the domain knowledge and the Neural Network model's structure were loosely coupled. There were no mappings between the pieces of knowledge used for creating the structure and the specific structures (e.g., input neurons, hidden neurons, the structure of the Neural Network) that were captured explicitly and formally by the manufacturing meta-model. Again, the semantic links between the manufacturing meta-model and the Neural Network meta-model were missing. Kalet et al. (2017) proposed using a dependency-layered ontology, which was implemented in OWL (Web Ontology Language), to solve the inconsistency and incompatibility between different BN models in the medical domain. However, the BN model was not formally represented. Also, there were no semantic links between the developed BN model and the ontology.

Hartmann et al. (2017) presented a model-driven analytics idea to emphasize the importance of using formalized domain knowledge in data analytics. They proposed using a domain model to explicitly define the semantics of raw data in the form of metadata, domain formula, mathematical models, and learning rules. However, the paper did not specify in what format to capture the metadata, mathematical formulas, and learning rules as well as how to integrate them. Also, the metadata model was not semantically connected to the analytics model.

To sum up, there are research gaps in (1) formalizing domain knowledge for data analytics, especially in explicitly and formally modelling the rules to process the data and constructing the analytics model; and (2) integrating formalized domain knowledge with analytics model. To address these problems, this paper proposes an enriched analytics model to enrich the formally captured analytics models with domain knowledge.

3. Proposed Enriched Analytics model

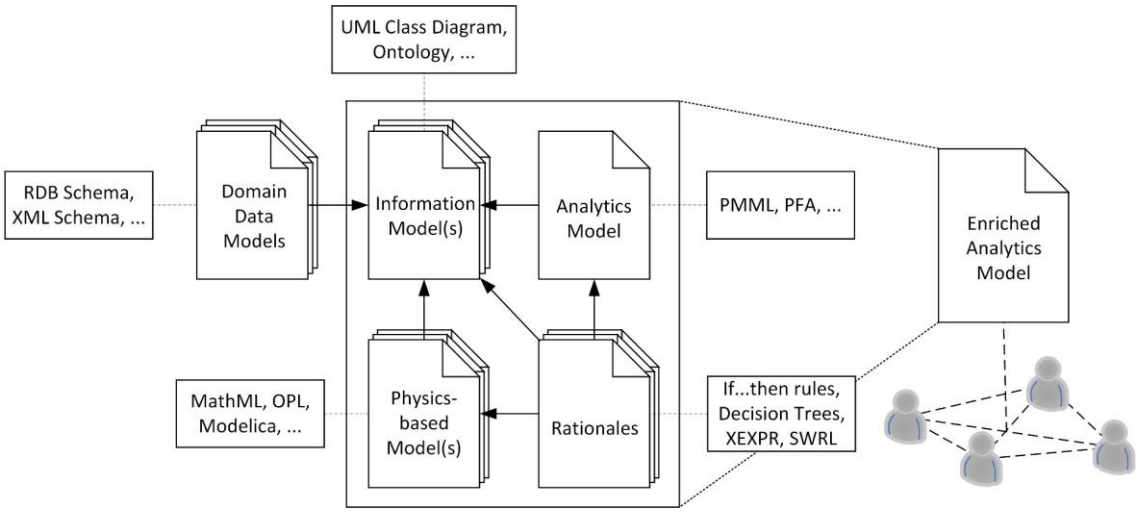

Figure 1. Enriching Analytics model with Domain Knowledge

To support the development of an analytics model, knowledge that is needed from the SM domain can be: (1) the domain meanings of the analytics model's entities (e.g., nodes, arcs, variables, etc.), (2) the physical or behavioural information that provides insights of a certain manufacturing system on which the data analytics project focuses, and (3) reasons or descriptions about why or how a certain structure or a parameter of the analytics model is defined. To incorporate all these types of knowledge into an analytic model, in this paper, they are captured into information model(s), physics-based model(s), and rationales, respectively. Figure 1 depicts the relationships between these models and the analytics model. From the perspective of facilitating the interoperability of the enriched analytics model, all three models along with the analytics model can be modelled using a uniform text-based format like XML (Extensible Markup Language), JSON (JavaScript Object Notation) or OWL, etc. To support the traceability of the enriched analytics model, the entities which are related across models should be semantically connected. The detailed descriptions of the four models are described in the following sections.

3.1 Information Model

In software engineering, an information model is a representation of concepts, relationships, constraints, rules, and operations to specify data semantics for a chosen domain of discourse (Lee, 1999). Here, the information model(s) provides a common terminology for the application domain of a data analytics project. Compared to data models which have implementation-specific details, information models define concepts and relationships in a higher abstract level, and they are protocol neutral (Pras and Schoenwaelder, 2003). In manufacturing domain, for example, the ANSI/ISA-95 (ANSI/ISA, 2010) standard is an information model that defines concepts and relationships to support the interfacing between an enterprise's business system and its manufacturing control system. The B2MML (Business To Manufacturing Markup Language) (Mesa International, 2013) is an XML-based data model which implements ANSI/ISA-95.

To explicitly express the domain meanings of other models (i.e., physics-based model, analytics model, and rationales), entities from other models need to be semantically linked to the corresponding entities in information model(s). Additionally, the information model(s) also needs to be semantically connected to the corresponding data model(s) for understanding the data.

3.2 Physics-based Model

Physics-based models are mathematical, empirical, simulation-based, and AI-based models, etc. that are developed to capture physical mechanics of a phenomenon or behaviours of an SM system. For example, forecasting models are developed to predict customer demand; production scheduling models are used for shop floor management; and cutting force models are developed for modelling material removal processes. Though these models are developed for a certain manufacturing application, the

physics/behaviours captured in these models can provide valuable insights of the manufacturing system for data analytics.

Normally, the physics-based models can only be processed by specific software tools. This is because these physics-based models are normally represented as application-specific languages. For example, the mathematical optimization problems can be modelled by the AMPL (A Mathematical Programming Language) (Fourer et al., 1990) and the OPL (Optimization Programming Language) (Hentenryck, 1999), which are processable in optimization solvers like CPLEX (IBM, 2018). But it is very difficult to process optimization models in these languages outside these application-specific tools.

To enable a universal method to extract information from the physics-based models, physics-based models need to be transformed into text-based formats. The text-based formats are friendly for software tools to parse. Currently, a lot of models have the text-based representation formats. For example, The MathML (Mathematical Markup Language) is an XML-based markup language which can represent both the meaning (i.e., Content ML) and format (i.e., Presentation ML) of mathematical expressions. The PMML, as discussed previously, is developed to represent predictive models in XML. The Ontology for Optimization (Witherell et al., 2007) represents optimization models in the engineering design domain in OWL. It should be noted that, no matter by which text-based format (i.e., XML, JSON, or OWL, etc.) a physics-based model is represented, to incorporate the physics-based model(s) into an analytics model, the physics-based model(s) should be transformed into the same format as the other models (i.e., information model(s), analytics model, and rationales).

3.3 Analytics model

The analytics model captured here is the model for developing a data analytics project for an SM application. To facilitate the enrichment of the SM domain knowledge, models like decision trees, cluster models, regression models, Neural Network models, or BN models, etc. also need to be formally represented. To express the domain meanings, the entities in an analytics model (e.g., nodes in a BN) should be semantically connected to the corresponding domain concepts defined in the information model(s).

Current standards like PMML and PFA that provide formal representations to support interoperability of analytics models can be used to represent the enriched analytics model to formally represent analytics models. Again, the analytics model needs to be converted to the same representation format as other models.

3.4 Rationales

In a data analytics project, the knowledge, which is a set of rules for guiding the development of an analytics model, is represented in rationales. For the knowledge from the application domain, rationales need to have connections to the related information models for obtaining the semantic meaning of the domain concepts. The rationales may also need to be linked to the physics-based models to indicate the part of system behavioural knowledge used in developing the analytics models. Also, the rationales need to connect to the analytics models to specify the links between the analytics model and the knowledge used in model development.

Like other individual models in the enriched analytics model, the rationales are also needed to be formally represented to make them processable and understandable by software tools. Because of the nature of the rationales, the rationales can be represented as rule-like styles. There are some technologies available to formally express rules. For example, the XEXPR scripting language (W3C, 2000) enables the expression of rules in

XML. JsonLogic (Wadhams, 2015) allows the construction of complex rules and serialization of the rules in JSON. In OWL, the SWRL (Semantic Web Rule Language) (W3C, 2004) language can be used to build rules. The selection of the languages should conform to the overall representation technique.

4. Validation of the Proposed Method Using a Case Study

To illustrate the proposed enriched analytics model in detail, this section introduces an example of data analytics project in a previous work. The overall data analytics process in the previous work without using the enriched analytics model is first introduced. Then, the reproduction of the process using the proposed enriched analytics model is elaborated. Finally, the utilization of the enriched analytics model and the benefits of using it are discussed.

4.1 Development of A Bayesian Network for Predicting Energy Consumption of Injection moulding Processes

In a previous study (Li et al., 2017), a BN model was developed to predict the energy consumption of the injection moulding process. The advantages of using a BN to predict energy consumption of injection moulding are: (1) BN is suitable for small data sets. To train a BN model for energy estimation, data from part design, mould design, material, and machine needs to be available. Although injection moulding is one of the mass-production processes, the collected data for different products/parts may be limited. (2) A BN allows efficient use of different sources of knowledge: knowledge provided by domain experts and the knowledge learned from data. The ability to learn a BN structure from data can help the user to identify new relationships between parameters, which in turn can be used for process improvement. (3) A BN can answer queries based on incomplete information. A designer may not possess all the information like the properties of the injection moulding machine that will be used for producing the part. A

BN can provide an estimate for a query considering nearly all possible values for that missing information based on the knowledge learned from data.

To study the role of SM domain knowledge in developing the BN, a BN model was first created by learning its structure and parameters from data using the 'bnlearn' package (Scutari and Denis, 2014) in R without the intervention of the domain knowledge. The BN nodes were selected from the parameters related to product, material, machine, process, and environment, etc. (Table I). The parameters were extracted from Nannapaneni et al. (2016). After the learning process, prediction correctness was tested. It was achieved at 76.8%, which is relatively low for effective prediction. For more information about the definition of prediction correctness, please refer to Li et al. (2017). By carefully studying the structure of the learned BN (Figure 2), we found that the learned structure missed finding important relationships and captured wrong/weak relationships instead. To improve the learned model, expert knowledge was applied to identify the problems in the model. The BN development process is shown in Figure 3.

TABLE I. PARAMETERS FOR MODELLING THE BAYESIAN NETWORK NODES

Category	***Name***	***Unit***	***Description***
Product	V_p	m^3	Volume of the part
	Δ	N/A	Percentage of volume used for gating system
	d	mm	Maximum depth of the part
	n	N/A	Number of cavities
	h_m	mm	Maximum wall thickness
Material	$Material$	N/A	Material type for the injection moulded material
	ρ	kg/m^3	Specific density of polymer
	γ	mm^2/s	Thermal diffusivity of the material
	C_p	$J/kg℃$	Heat capacity of the polymer
	H_f	kJ/kg	Heat of fusion
	ϵ	N/A	Percentage shrinkage rate of the polymer
Machine	$Machine$	N/A	Machine type for the injection moulding machine
	P_b	kW	Power consumption when the machine is idling
	s	mm	Maximum clamp stroke
	t_d	s	Dry cycle time
	P_{inj}	kW	Machine injection power
Process	p_i	MPa	Injection pressure
	T_m	℃	Recommended mould temperature
	T_{inj}	℃	Injection temperature
	T_{ej}	℃	Ejection temperature
Environment	T_{pol}	℃	Initial temperature of the polymer
Others	Q	m^3	Maximum flow rate for injection
	Q_{avg}	m^3	Average flow rate
	P_m	kW	Melting power
	V_s	m^3	Volume of one shot including gating system
	E_m	kJ	Energy consumption in melting
	E_{inj}	kJ	Energy consumption of injection
	t_{inj}	s	Injection time
	E_c	kJ	Energy consumption in cooling
	COP	N/A	Coefficient of performance
	E_r	kJ	Energy consumption in resetting
	t_r	s	Resetting time
	E_{shot}	kJ	Energy consumption of a shot
	η	N/A	Efficiency
	t_{cyc}	s	Cycle time
	E_{part}	kJ	Energy consumption of a part

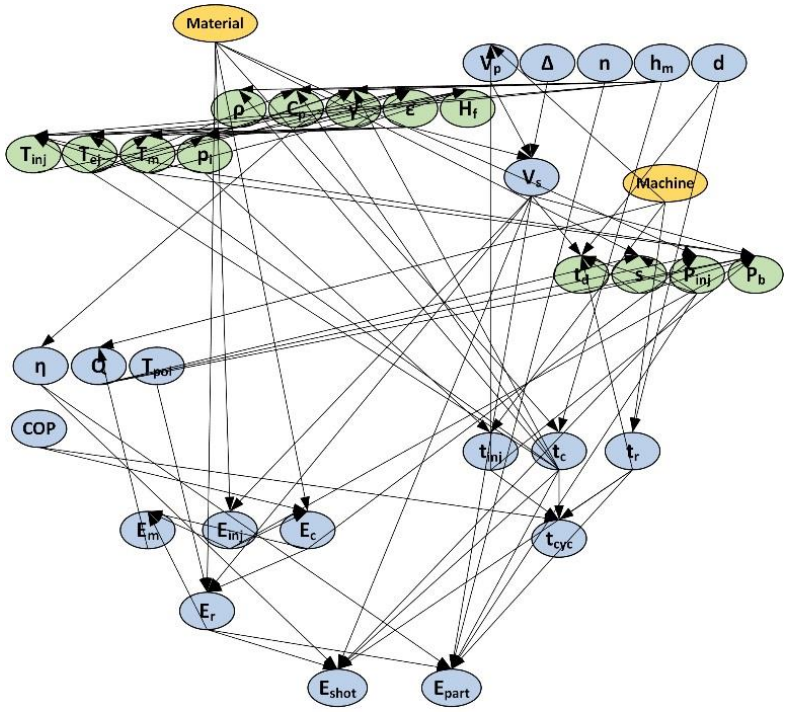

Figure 2. A BN Structure Learned from Data

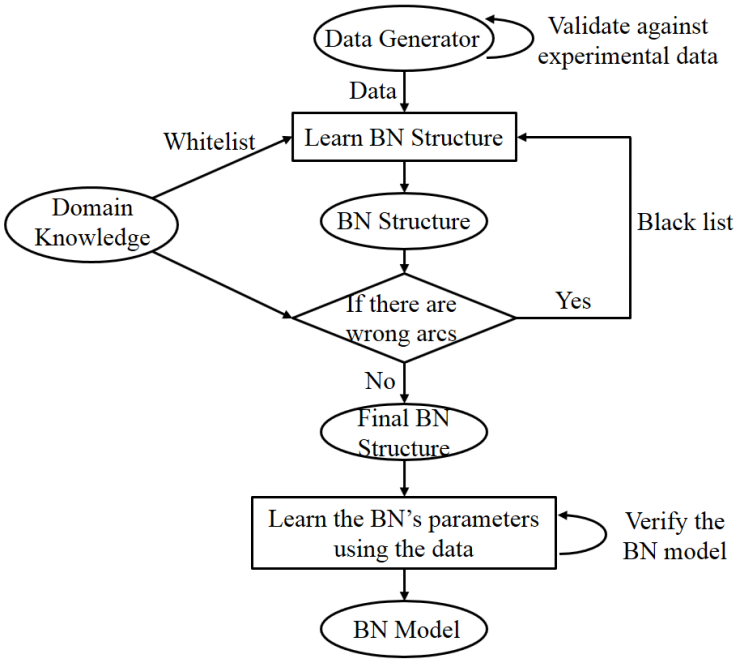

Figure 3. Development Process for the BN

Due to the lack of LCA (Life-cycle assessment) data from real injection moulding processes, a simulation-based data generator had been developed to generate the data. This data generator had been validated against experimental data from the literature (Ribeiro, 2012). Before learning the structure from data, a whitelist which captures

important relationships between nodes was created. A whitelist, which contains arcs that need to be included in the BN, was created based on the knowledge found from mathematical equations (shown in Table II) for calculating the energy consumption of injection moulding processes. The equations are extracted from Madan et al. (2013). An equation can be considered as defining the parent/child relationships for the equation variables. The independent variables (i.e., variables on the right-hand side) of an equation are treated as the parent nodes of the dependent variable (i.e., the variable on the left-hand side).

Additionally, a blacklist, which prevents the BN from creating arcs between nodes, was created from the problems identified in the learned BN structure. Through carefully examining the learned BN structure (Figure 2), four problems were identified: (1) a parameter node (i.e., nodes which represent material, product, or process parameters) from one of the five categories (i.e., product, process, material, machine, and environment) should not have causal relationships with parameter nodes from the other four categories. For example, in Figure 2, material-related parameters like density ρ and heat capacity C_p are found to not dependent on material but are related to a product-related property – maximum wall thickness h_m, which is wrong. Though there are recommendations for the minimum wall thickness according to the injection moulded materials, h_m are normally designed as thin as possible. This is because thinner walls require less material and less cooling time. However, there are no recommendations for h_m according to different materials. (2) The concept nodes like Material and Machine should not be related to the parameters from categories other than Material and Machine, respectively. Figure 2 shows that machine property nodes of maximum clamp stroke s and injection power P_{inj} are found to be dependent on node Material. However, the injection moulding machine is selected based on the shot size and the maximum clamp

stroke, which are dependent on the product not material. (3) Parameter nodes within a category should not have parent-child relationships. It is true that within some categories, like Machine and Material, parameter nodes are related. But mainly material type or machine type determines the properties. (4) The parameter nodes from the five categories should not have any parent nodes other than the concept nodes. It can be observed in Figure 2 that parameter nodes from the five categories like injection temperature T_{inj} and ejection temperature T_{ej} are found to have parent nodes in the Others category like injection t_{inj}. However, there should not have causal relationships between T_{inj} and t_{inj}.

TABLE II. EQUATIONS FOR ESTIMATING ENERGY CONSUMPTION OF INJECTION MOULDING

Stage	***Equations***
Melting	$Q = P_{inj} * 1000/p_i,\ Q_{avg} = 0.5Q$ $P_m = \frac{\rho Q_{avg} C_p (T_{inj} - T_{pol}) + \rho Q_{avg} H_f}{1000}$ $V_s = V_p \left(1 + \frac{\epsilon}{100} + \frac{\Delta}{100}\right) n,\ E_m = (P_m * V_s)/Q$
Injection	$E_{inj} = P_{inj} t_{inj},\ t_{inj} = \frac{2V_s p_i}{P_{inj}}$
Cooling	$E_c = \frac{\rho V_s C_p (T_{inj} - T_{ej}) + \rho V_s H_f}{1000 * COP},\ t_c = \frac{h_m^2}{\pi^2 \gamma} ln \frac{4(T_{inj} - T_m)}{T_{ej} - T_m}$
Resetting	$E_r = 0.25(E_{inj} + E_c + E_m),\ t_r = 1 + 1.75 t_d \sqrt{\frac{2d+5}{s}}$
Whole Process	$E_{shot} = 1.2 \times \left(\frac{0.75E_m + E_{inj}}{\eta_{inj}} + \frac{E_r}{\eta_r} + \frac{E_c}{\eta_c} + \frac{0.25E_m}{\eta_m}\right) + P_b t_{cyc}$ $t_{cyc} = t_{inj} + t_c + t_r,\ E_{part} = \frac{E_{shot}}{n}$

By utilizing the whitelist and the blacklist, an iterative approach to learn the BN structure from data was applied (Figure 3). By using the iterative approach, the wrong arcs can be easily identified and handled during each iteration. With the whitelist and the iteratively updated blacklist, the learning procedure is repeated until no wrong arcs can

be found in the BN structure. After learning the BN parameters (i.e., conditional probability tables for discrete nodes and Gaussian distributions for continuous nodes) and verifying the BN model, the development of the BN was finalized. The prediction correctness of the BN model developed with the domain knowledge was achieved at 85%, which is sufficient and is higher than the learned BN model (Figure 2).

4.2 Development of the Enriched Analytics model

In this section, the enriched analytics model for the BN is developed. The development of each individual model and the integration between the models are introduced. In this paper, OWL 2 (W3C, 2012) is used as the format for implementing all the models.

4.2.1 Information Model

Since the application domain of this case study is targeting at estimating energy consumption of injection moulding processes, the information model used in this paper is selected from a previous work (Zhang et al., 2015). This information model was developed to facilitate the sustainability evaluation in the manufacturing domain. This model was also extended with respect to the injection moulding process. A compact version of the information model, or the Sustainable Manufacturing Ontology (SMO), is shown in Figure 4. A brief explanation of some important concepts in the information model is narrated below:

- **Product**: A *Product* describes an object which is synthesized by a set of parts or subassemblies (each subassembly itself is also a product). The spatial relationships and contact constraints between parts are also defined within the *Product* class.

- **Part**: A *Part* is a single component that is used to construct a *Product*. A *Part* is a minimal functional unit of a product; thereby a part must be formed with a type of material and it has a certain geometrical shape.
- **Material**: A *Material* describes a kind of material associated with a *Part*. A *Material* has a list of properties like mechanical properties, chemical properties, thermal properties, etc. which are captured in the *Parameter* class.
- **Process Plan**: A *ProcessPlan* defines a sequence of manufacturing operations to produce a *Part*. The types of processes, types of equipment and operation parameters are specified in a *ProcessPlan*.
- **Process**: A *Process* describes a series of operations that need to be carried out to produce the final product. A *Process* can be a *ManufacturingProcess* or an *AssemblyProcess*. A *ManufacturingProcess* is a process that transforms a raw material into a finished or a semi-finished *Part*. It can be a machining process, a casting process, a forging process, or a heat treatment process, etc. All the *ManufacturingProcesses* required to be carried out to produce a *Part* construct a *ProcessPlan*.
- **Activity**: An *Activity* is a minimal operational unit of a *Process*. For example, an *Activity* of a typical machining process can be setting up the machine, fastening the workpiece, positioning the cutting tool, injecting the cutting fluid, etc.
- **Environment**: The *Environment* class describes the environment related concepts of an *Activity* or a *Process*. All types of the environmental impacts are defined here, and each type of impacts is represented as a sustainability indicator. The sustainability of a *Part* or a *Product* can be further evaluated by considering the *Process*es that are carried out to produce the *Part* or *Product*.

- **Parameter**: A *Parameter* represents an entity that describes a property of a manufacturing concept. The properties of a *Product*, a *Part*, a *Material*, a *Process*, and an *Activity* are modelled as *Parameter*s.
- **Equipment**: An *Equipment* can be a tool or a machine on the shop floor.

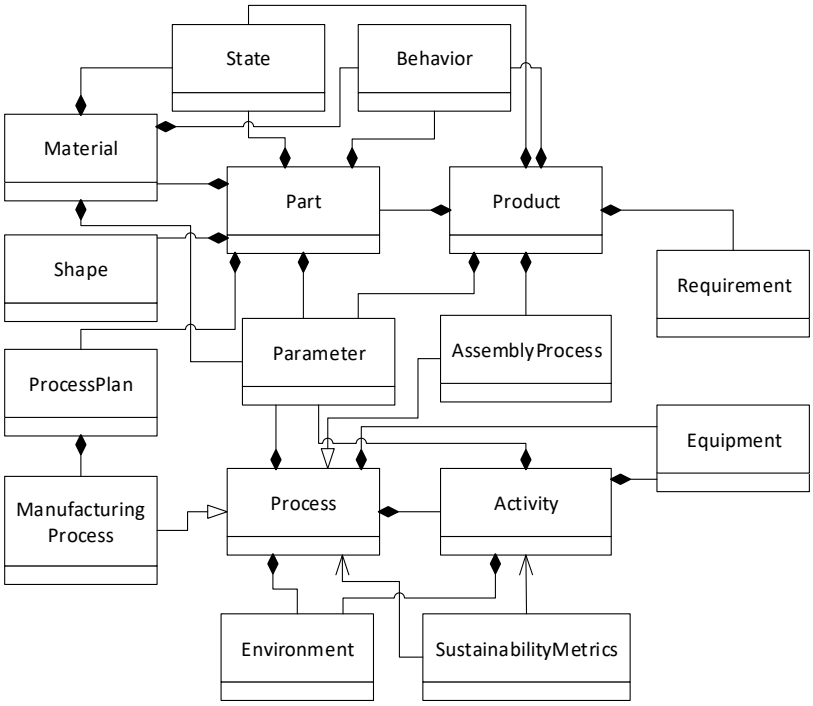

Figure 4. A UML Representation of the Sustainable Manufacturing Ontology (SMO)

4.2.2 Physics-based Model

The physics-based models used in developing the BN are the mathematical equations which estimate the energy consumption of injection moulding (Table II). To represent mathematical equations in OWL, the OntoModel proposed by Suresh et al. (2010) has been used. In OntoModel (Figure 5), other than capturing the assumption, universal constant and dependent variable, etc., an equation can be represented using the Content ML in MathML. In Figure 5, the black boxes represent *owl:class*es; the green boxes are datatypes; the red arrows indicate the *has-a* object properties; the green arrows indicate the data properties. The OntoModel is modified so that it can be connected to the domain

information model (i.e., SMO). The *Variable* class in OntoModel is connected to the *Parameter* class in the SMO, which connects the variables in an equation to their domain meanings.

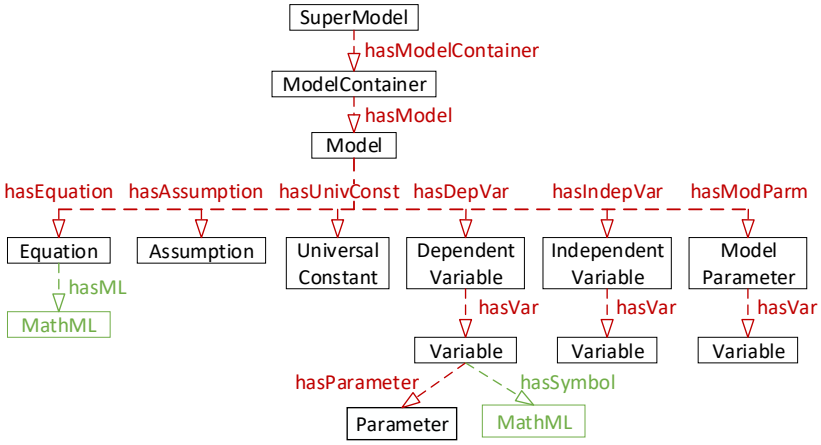

Figure 5. OntoModel and Its Connection to the SMO

4.2.3 Analytics model

To represent a BN (i.e., the analytics model) in OWL, an OWL-based BN model is developed. Figure 6 demonstrates this OWL-based BN model in a tree structure. The class names in this model are borrowed from PMML 4.3 - Bayesian Network Models (Data Mining Group, 2016). The structure of the PMML BN model is slightly modified (e.g., adding *BayesianNetworkNode* class, replacing the 'has-a' relationship between ContinuousDistribution and *NormalDistribution* with the 'hasSubClass' relationship) to better fit the OWL structure. This OWL-based BN model has been verified with the BN example provided on the webpage of the PMML BN model. The verification proves the OWL-based BN model to be capable of fully representing BNs.

All the parameters in Table I are modelled as the *BayesianNetworkNode* instances in the OWL-based BN model. The semantic connection between a *BayesianNetworkNode* and a manufacturing concept in the SMO is achieved by an *isAssociateTo*(*BayesianNetworkNode, domainConcept*) object property.

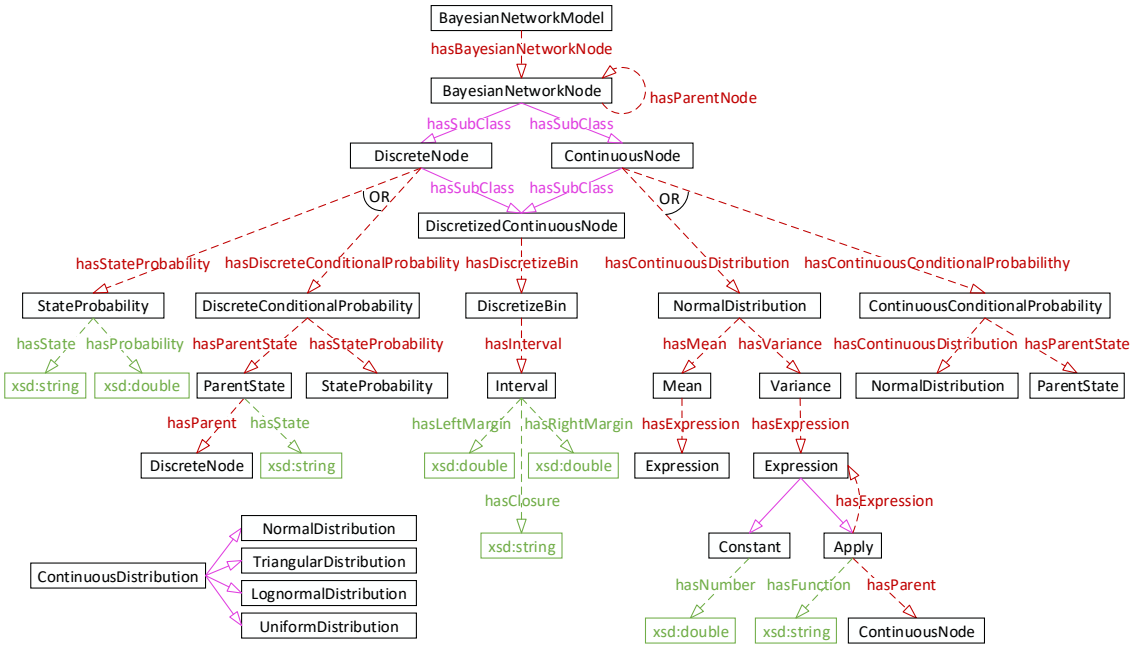

Figure 6. A Tree Presentation of the OWL-based BN Model

4.2.4 Rationales

To improve the BN structure with domain knowledge, the rationales to facilitate the creation of the whitelist and the blacklist have been developed. The whitelist rules/rationales are created to capture the BN node relationships provided from the physics-based models (i.e., equations) and domain rules. Based on the identified four problems of the learned BN structure (section 4.1), blacklist rules/rationales are developed. The blacklist rules can be used to avoid incorrect structures in the BN. All the whitelist and blacklist rules are modelled using SWRL in OWL. Object property *hasParentNode* represent a whitelist relationship. Object property *hasWrongArc* represents a blacklist relationship. To enhance the traceability of the rationales, each rule has its own corresponding numbered object property. For example, object property *hasParentNode* in whitelist rule 1 is *hasParentNode1*. The whitelist rules and blacklist rules are described as follows.

Whitelist Rule 1

This rule is created based on the physics-based models (i.e., equations in Table II). As discussed before, an equation can be considered as defining the parent/child relationships for the equation variables. The independent variables (i.e., variables on the right-hand side of an equation) of an equation are treated as the parents of the dependent variable (i.e., the variable on the left-hand side of an equation). Object property *hasParentNode1* represents a parent/child relationship between two *BayesienNetworkNode*s.

DependentVariable(?dv), IndependentVariable(?iv), MathematicModel(?m), Variable(?v_dv), Variable(?v_iv), Parameter(?p1), Parameter(?p2), BayesianNetworkNode(?n1), BayesianNetworkNode(?n2), hasDependentVariable(?m, ?dv), hasIndependentVariable(?m, ?iv), hasVariable(?dv, ?v_dv), hasVariable(?iv, ?v_iv), isAssociatedWith(?n1, ?p1), isAssociatedWith(?n2, ?p2), isAssociatedWith(?v_dv, ?p1), isAssociatedWith(?v_iv, ?p2) -> hasParentNode1(?n1, ?n2)

Whitelist Rule 2

According to the classification of parameters in Table I, the manufacturing concept nodes (i.e., Machine and Machine) should have parent-child relationships with their related parameter nodes.

ManufacturingConcept(?mc), Parameter(?p), BayesianNetworkNode(?n1), BayesianNetworkNode(?n2), isAssociatedWith(?n1, ?mc), isAssociatedWith(?n2, ?p), hasParameter(?mc, ?p) -> hasParentNode2(?n2, ?n1)

Whitelist Rule 3

Some process parameters in the injection moulding process are selected according to the material. For example, the selection of T_{inj}, T_{ej}, T_m, and p_i are based on the material

type (Boothroyd et al., 2011). So, causal relationships between the *Material* node and these process parameters should be captured.

Material(?m), Parameter(?pp), Process(?p), BayesianNetworkNode(?n_m), BayesianNetworkNode(?n_pp), isAssociatedWith(?n_m, ?m), isAssociatedWith(?n_pp, ?pp), hasParameter(?p, ?pp) -> hasParentNode3(?n_pp, ?n_m)

Blacklist Rule 1

To address the first problem of the learned BN structure, a set of rules to prevent connecting parameter nodes from different categories are created. Here, the rule to prevent parameter nodes from the *Material* and *Product* categories to be connected is demonstrated.

Material(?material), Parameter(?p_material), Parameter(?p_product), Product(?product), BayesianNetworkNode(?n_p_material), BayesianNetworkNode(?n_p_product), isAssociatedWith(?n_p_material, ?p_material), isAssociatedWith(?n_p_product, ?p_product), hasParameter(?material, ?p_material), hasParameter(?product, ?p_product) -> hasWrongArc1(?n_p_material, ?n_p_product), hasWrongArc1(?n_p_product, ?n_p_material)

Blacklist Rule 2

This rule addresses problem # 2. It avoids the *Material* and the *Machine* nodes to be connected to the parameter nodes from other categories. An example rule is shown below to prevent the *Material* node to be connected to the machine-related parameter nodes.

Machine(?machine), Material(?material), Parameter(?p_machine), BayesianNetworkNode(?n_material), BayesianNetworkNode(?n_p_machine), isAssociatedWith(?n_material, ?material), isAssociatedWith(?n_p_machine, ?p_machine), hasParameter(?machine, ?p_machine) -> hasWrongArc2(?n_material, ?n_p_machine), hasWrongArc2(?n_p_machine, ?n_material)

Blacklist Rule 3

Targeting at problem # 3, this blacklist rule avoids the parameter nodes within one category to be connected to each other. The example for the material category is shown below.

Material(?m), Parameter(?p1), Parameter(?p2), BayesianNetworkNode(?n1), BayesianNetworkNode(?n2), isAssociatedWith(?n1, ?p1), isAssociatedWith(?n2, ?p2), hasParameter(?m, ?p1), hasParameter(?m, ?p2), DifferentFrom (?n1, ?n2) -> hasWrongArc3(?n1, ?n2)

Blacklist Rule 4

To prevent the parameter nodes from the 5 categories to have any parent nodes other than the ones defined by the whitelist rules (problem 4), an example rule is demonstrated below for the process category. In this rule, the "*hasTempParentNode*" object property represents an arc learned from data.

Process(?process), Parameter(?pm), hasParameter(?process, ?pm), BayesianNetworkNode(?n_p_m), BayesianNetworkNode(?n_mc), isAssociatedWith(?n_p_m, ?pm), hasTempParentNode(?n_p_m, ?n_mc) -> hasWrongArc4(?n_p_m, ?n_mc)

4.2.5 Representation of the Enriched BN Model in OWL

During the development of the enriched BN model, the individual models for the information model, analytics model, and physics-based model are separately created first. These models are general and could be applied to any applications and do not have any populated instances. After verifying that all the individual OWL-based models can sufficiently represent the models, the enriched analytics model is then created by importing the three OWL-based models into the OWL-based enriched analytics model. Instances of the domain concepts in the SMO, the *BayesianNetworkNodes* in the BN model, and the equations represented by the OntoModel are populated. The whitelist and blacklist rules are then created using the SWRL rules. All these models and rules are implemented in protégé 5.2, which is an open-source ontology editor. The screenshot of the enriched BN model in protégé is demonstrated in Figure 7.

Figure 8 shows a screenshot of reasoning the rationales (i.e., whitelist and blacklist rules). The object property assertions highlighted in light yellow are inferred relationships from reasoning the rationales. It can be observed that the rationale used to create or eliminate an arc can be easily tracked by using the numbered object properties.

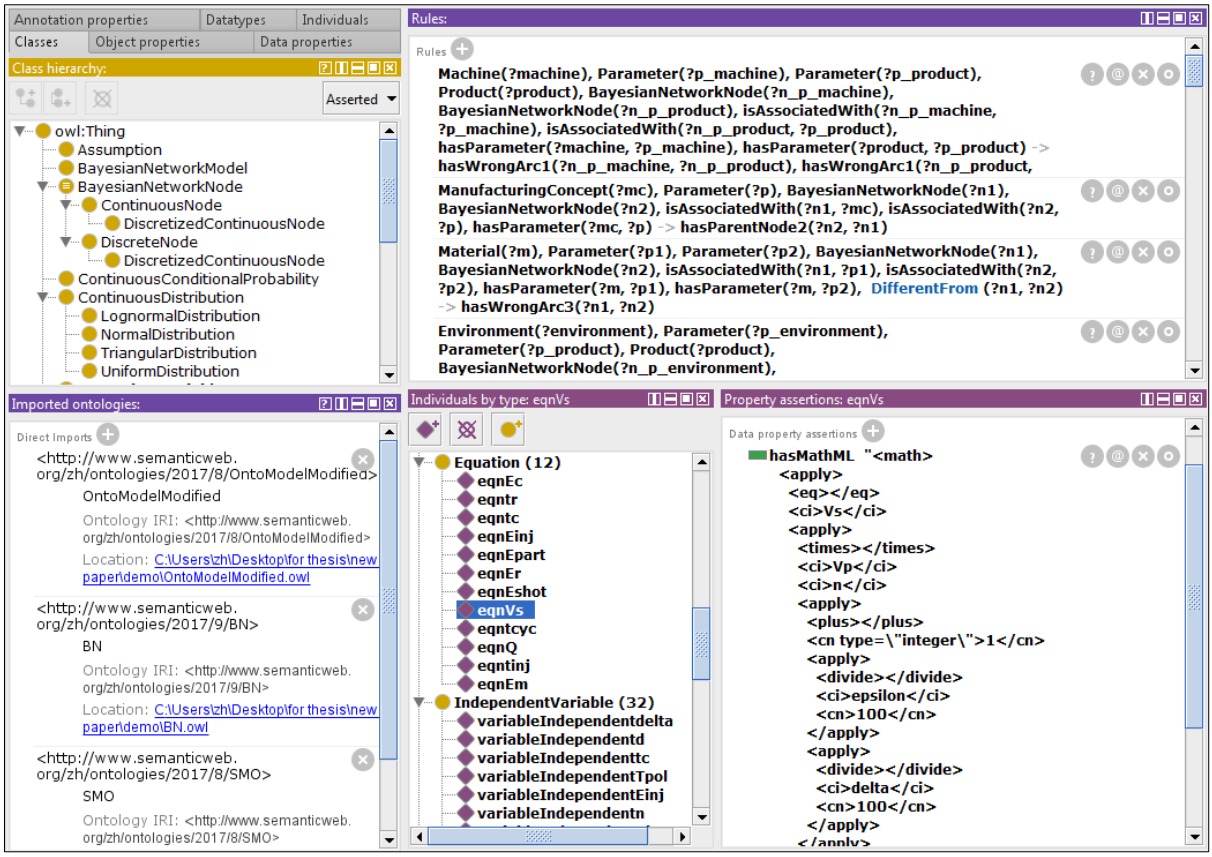

Figure 7. The Enriched BN Model in protégé 5.2

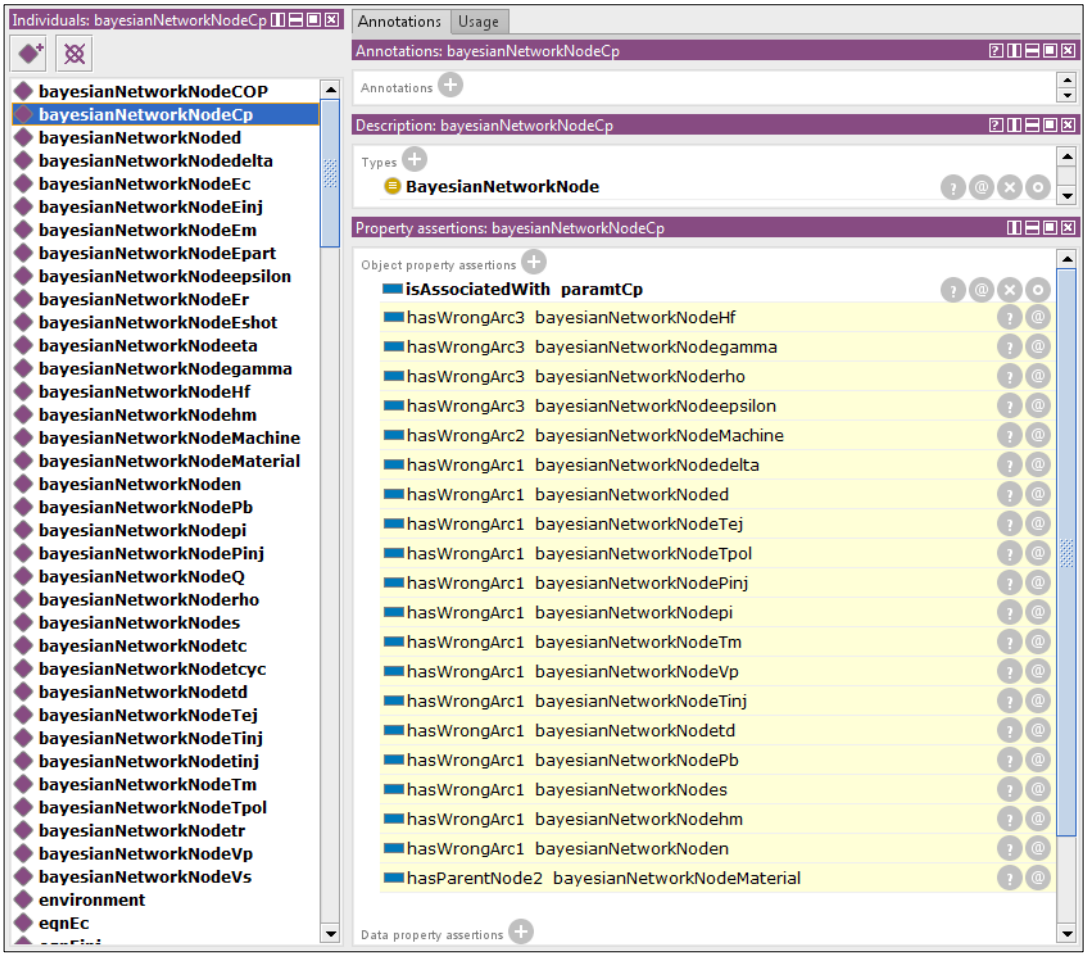

Figure 8. Inferred Whitelist and Blacklist Relationships in protégé 5.2

4.3 Utilization of the Enriched Analytics model

Following the development process (Figure 3), a BN is developed (Figure 9) by utilizing the enriched analytics model. In the development process, the enriched analytics model has been used to exchange information between a domain expert and a data analyst. The domain expert first models the domain knowledge (integrating the information model, adding the physics-based model, creating rationales) for the development of the BN. Then, the data analyst iteratively learns the BN structure from data with the whitelist and the blacklist, which are extracted from the enriched BN model through a parser (that has been developed for this purpose). Here, the enriched BN model is used to pass the BN along with its relevant domain knowledge between the domain expert and the data

analyst. After sending the enriched BN model with the learned structures, the domain expert can analyse the BN structure and add the corresponding rationales to improve the BN structure. The domain expert can directly add or modify domain knowledge on the enriched analytics model in GUI (Graphical User Interface) -based software tools like protégé. Figure 10 is a sequence diagram that shows the information exchange.

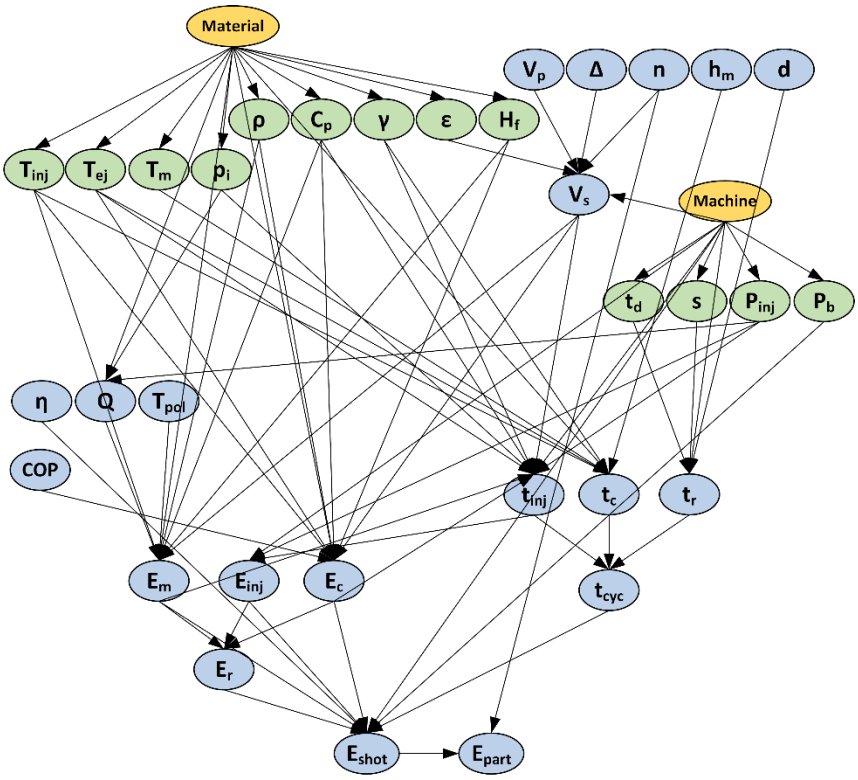

Figure 9. The final BN Structure

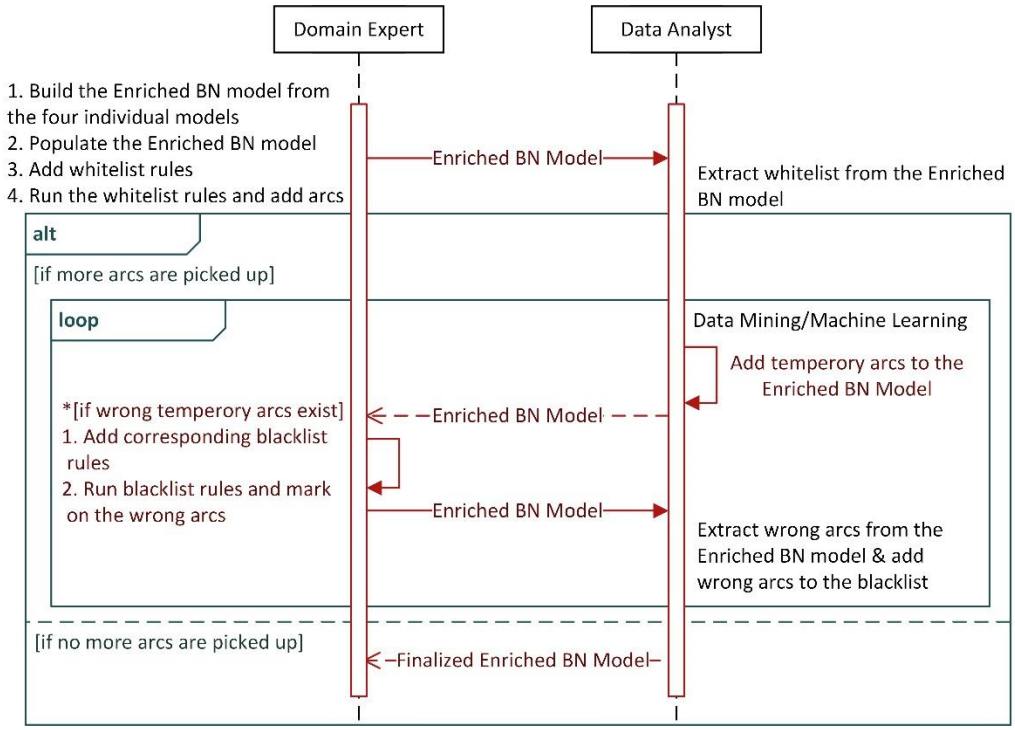

Figure 10. Information Exchange Using the Enriched Analytics model

4.4 Discussion

Two benefits have been identified during the development of the BN using the enriched analytics model: (1) Shortened cycle time for the analytics model development. In this case study, with the assistance of automatic processing and reasoning from the software tools, the development cycle time has been reduced from 2 hours to 15 minutes. With the formally defined rationales, the arcs captured by the whitelist and the blacklist can be generated automatically instead of working manually. (2) Eliminating human error. Through reasoning the formally defined rationales, more arcs in the blacklist have been identified. Some of these arcs can be easily missed by manual work.

These two direct benefits are brought by the enhanced interoperability and traceability of the enriched analytics model. From the perspective of enhancing interoperability, the enriched analytics model can explicitly represent the analytics models with their relevant domain knowledge through capturing their concepts, relationships, and rules, etc. without semantic ambiguity; the analytics models and their relevant domain knowledge are

formally represented, which enables the automatic processing through software tools. From the perspective of enhancing traceability, the entities of an analytics model can be directly traced to its corresponding domain concepts; the analytics models' structures can also be easily traced to the rationales which are used to create these structures. It can be expected that with the enhanced interoperability and traceability, more effective and efficient information exchange can be achieved by using the enriched analytics model in a distributed environment, where the domain experts and the data analysts are not necessarily in the same geographical area.

Further efforts need to be done to implement a platform to support the formal communications with the proposed enriched analytics model in real production applications. Technically, the parser of the OWL-based enriched analytics model is developed on top of OWL API (Horridge and Bechhofer, 2010). OWL API is a Java-based open-source OWL editor which also supports the same reasoning capability as protégé. So, the authors suggest to integrate OWL API-based parsers/reasoners into the industrial tools to support the parsing and reasoning of the OWL-based enriched analytics model. Additionally, domain experts can use protégé to directly edit an enriched analytics model since protégé has a friendly user interface for knowledge modelling.

5. Conclusion

This paper proposed a methodology to enrich analytics models with domain knowledge to facilitate data analytics in a Smart Manufacturing environment. The motivation for encapsulating an analytics model and its domain knowledge in a single model came from the need for interoperability and traceability. To model the SM domain knowledge, this paper proposed to explicitly and formally represent domain information models, physics-based models, and rationales. Information models are used to bridge the semantic gaps between the SM domain and the data analytics applications. By formally

representing the physics-based models, the knowledge captured in the physics-based models can be reused. The rationales, which are used to develop the analytics model, are digitally documented to enhance traceability. The formal representation of the targeting analytics model can be modelled based on the existing standards like PMML or PFA. To enrich an analytics model, the formally represented domain information models, physics-based models, and rationales should be semantically integrated with the analytics model.

A case study from a previous study, where a Bayesian Network model was developed to predict the energy consumption of the injection moulding process, was used to illustrate the development and utilization of the enriched analytics model. The components of the enriched analytics model are: a manufacturing domain information model, a set of mathematical equations as the physics-based models, the whitelist/blacklist rationales for constructing the BN, and the formally represented BN model. All these models are implemented using OWL. The case study demonstrates the benefits from utilizing the enriched analytics model: shortened cycle time for model development and eliminating human error. These two benefits come from the enhanced interoperability and traceability of the analytics model and its relevant domain knowledge.

Acknowledgement and Disclaimer

This work has been sponsored under the cooperative agreement between the U.S. National Institute of Standards and Technology (NIST) and Syracuse University.

Certain commercial systems are identified in this article. Such identification does not imply recommendation or endorsement by NIST; nor does it imply that the products identified are necessarily the best available for the purpose. Further, any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NIST or any other supporting U.S. government or corporate organizations.

References

ANSI/ISA. 2010. *ANSI/ISA-95.00.01-2010 (IEC 62264-1 Mod) Enterprise-Control System Integration - Part 1: Models and Terminology*. USA: ANSI/ISA.

Arena, D., Tsolakis, A. C., Zikos, S., Krinidis, S., Ziogou, C., Ioannidis, D., Voutetakis, S., Tzovaras, D., and Kiritsis, D. 2018. "Human Resource Optimisation through Semantically Enriched Data." *International Journal of Production Research* 56 (8): 2855-2877. doi: 10.1080/00207543.2017.1415468.

Boothroyd, G., Dewhurst, P., and Knight, W.A. 2011. *Product Design for Manufacture and Assembly*. 3rd ed. Boca Raton, FL: CRC Press.

Campos, L.M., and Castellano, J.G. 2007. "Bayesian Network Learning Algorithms Using Structural Restrictions." *International Journal of Approximate Reasoning* 45 (2): 233-254. doi: 10.1016/j.ijar.2006.06.009.

Data Mining Group. 2015. "Portable Format for Analytics (PFA)." Data Mining Group. Accessed October 11 2018. http://dmg.org/pfa/index.html.

Data Mining Group. 2016. "PMML Version 4.3." Data Mining Group. Accessed October 11 2018. http://dmg.org/pmml/pmml-v4-3.html.

Dotoli, M., Fay, A., Miśkowicz, M., and Seatzu, C. 2018. "An Overview of Current Technologies and Emerging Trends in Factory Automation." *International Journal of Production Research*. doi: 10.1080/00207543.2018.1510558.

Fourer, R., Gay, D.M., and Kernighan, B.W. 1990. "A Modeling Language for Mathematical Programming." Management Science 36: 519-554. doi: 10.1287/mnsc.36.5.519.

Hartmann, T., Moawad, A., Fouquet, F., Nain, G., Klein, J., Traon, Y. L., and Jezequel, J.M. 2017. Model-driven Analytics: Connecting Data, Domain Knowledge, and Learning. *arXiv:1704.01320*.

Hentenryck, P.V. 1999. The OPL optimization programming language. MIT Press Cambridge: MA, USA.

Horridge, M., and Bechhofer, S. 2010. “The OWL API: A Java API for OWL Ontologies.” *Semantic Web* 2 (1): 11-21. doi: 10.3233/SW-2011-0025.

IBM. 2018. “IBM ILOG CPLEX Optimization Studio.” IBM. Accessed November 22 2018. https://www.ibm.com/analytics/cplex-optimizer.

Johnson, I., Abécassis, J., Charnomordic, B., Destercke, S., and Thomopoulos, R. 2010. “Making Ontology-based Knowledge and Decision Trees Interact: An Approach to Enrich Knowledge and Increase Expert Confidence in Data-driven Models.” In *Knowledge Science, Engineering and Management. KSEM 2010. Lecture Notes in Computer Science, vol. 6291*, edited by Bi Y., Williams MA. Springer, Berlin, Heidelberg.

Kalet, A. M., Doctor, J. N., Gennari, J. H., and Phillips, M. H. 2017. “Developing Bayesian Networks from A Dependency-layered Ontology: A Proof-of-concept in Radiation Oncology.” *Medical Physics* 44 (8): 4350-4359. doi: 10.1002/mp.12340.

Kim, H., Vastola, J. T., Kim, S., Lu, J., and Grover, M. A. 2017. “Incorporation of Engineering Knowledge into the Modeling Process: A Local Approach.” *International Journal of Production Research* 55 (20): 5865-5880. doi: 10.1080/00207543.2016.1278082.

Kopanas, I., Avouris, N. M., and Daskalaki, S. 2002. “The Role of Domain Knowledge in A Large Scale Data Mining Project.” In *Methods and Applications of Artificial Intelligence. SETN 2002. Lecture Notes in Computer Science, vol. 2308*, edited by Vlahavas I.P., Spyropoulos C.D. Springer, Berlin, Heidelberg.

Kusiak, A. 2018. “Smart Manufacturing.” *International Journal of Production Research* 56 (1-2): 508-517. doi: 10.1080/00207543.2017.1351644.

Lechevalier, D., Narayanan, A., Rachuri, S., Foufou, S. & Lee Y.T. (2016). *Model-based engineering for the integration of manufacturing systems with advanced analytics*. In: Harik R., Rivest L., Bernard A., Eynard B., Bouras A. (eds) Product Lifecycle Management for Digital Transformation of Industries. PLM 2016. IFIP Advances in Information and Communication Technology, vol. 492. Springer, Cham.

Lee, Y.T. 1999. "Information Modeling: From Design to Implementation." Paper presented at the Second World Manufacturing Congress (WHC 1999), Durham, U.K.

Lemaignan, S., Siadat, A., Dantan, A.-Y., and Semenenko, A. 2006. "Mason: A Proposal for An Ontology of Manufacturing Domain." Paper presented at IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06), Prague, Czech Republic. doi: 10.1109/DIS.2006.48.

Li, Y., Zhang, H., Roy, U., and Lee, Y.T. 2017. "A Data-driven Approach for Improving Sustainability Assessment in Advanced Manufacturing." Paper presented at 2017 IEEE International Conference on Big Data (BigData 2017), Boston, MA, USA.

Madan, J., Mani, M., and Lyons, K. W. 2013. "Characterizing Energy Consumption of the Injection Molding Process." Paper presented at Proceedings from ASME 2013: Manufacturing Science and Engineering Conference, Madison, WI, USA. doi: 10.1115/MSEC2013-1222.

MESA International. 2013. "B2MML V0600. MESA International. Accessed October 11 2018. https://services.mesa.org/ResourceLibrary/ShowResource/0f47758b-60f0-40c6-a71b-fa7b2363fb3a.

Munger, T., Desa, S., and Wong, C. 2015. "The use of domain knowledge models for effective data mining of unstructured customer service data in engineering applications." Paper presented at the 2015 IEEE First International Conference on Big

Data Computing Service and Applications, Redwood City, CA, USA. DOI: 10.1109/BigDataService.2015.46.

Nannapaneni, S., Mahadevan, S., and Rachuri, S. 2016. “Performance Evaluation of a Manufacturing Process under Uncertainty Using Bayesian Networks.” *Journal of Cleaner Production* 113 (1): 947-959. doi: 10.1016/j.jclepro.2015.12.003.

Palmer, C., Usman, Z., Junior, O. C., Malucelli, A., and Young, R. I. M. 2018. “Interoperable Manufacturing Knowledge Systems.” *International Journal of Production Research*. doi: 10.1080/00207543.2017.1391416.

Perez-Rey, D., Anguita, A., and Crespo, J. 2006. “OntoDataClean: Ontology-based integration and Preprocessing of Distributed Data.” In *Biological and Medical Data Analysis. ISBMDA 2006. Lecture Notes in Computer Science, vol. 4345*, edited by Maglaveras N., Chouvarda I., Koutkias V., Brause R. Springer, Berlin, Heidelberg.

Pras, A., and Schoenwaelder, J. 2003. “On the Difference between Information Models and Data Models.” *RFC 3444*, Jan. 2003.

Ribeiro, I., Peças, P., and Henriques, E. 2012. “Assessment of energy consumption in injection moulding process.” In *Leveraging Technology for a Sustainable World*, edited by Dornfeld, D., and Linke B. Springer, Berlin, Heidelberg.

Scutari, M., and Denis, J. B. 2014. *Bayesian networks with examples in R*. Boca Raton, FL: Taylor & Francis Group.

Sinha, A.P., and Zhao, H. 2008. “Incorporating Domain Knowledge into Data Mining Classifiers: An Application in Indirect Lending.” *Decision Support Systems* 46 (1): 287-299. doi: 10.1016/j.dss.2008.06.013.

Suresh, P., Joglekar, G., Hsu, S., Akkisetty, P., Hailemariam, L., Jain, A., Reklaitis, G., and Venkatasubramanian, V. 2008. “Onto MODEL: Ontological Mathematical

Modeling Knowledge Management." *Computer Aided Chemical Engineering* 25: 985-990. doi: 10.1016/S1570-7946(08)80170-8.

Trappey, A. J. C., Trappey, C. V., Chiang, T., Huang, Y. 2013. "Ontology-based Neural Network for Patent Knowledge Management in Design Collaboration." *International Journal of Production Research* 51 (7): 1992-2005. doi: 10.1080/00207543.2012.701775.

W3C. 2000. "XEXPR - A Scripting Language for XML." W3C. Accessed October 11 2018. https://www.w3.org/TR/2000/NOTE-xexpr-20001121/.

W3C. 2004. "SWRL: A Semantic Web Rule Language Combining OWL and RuleML." W3C. Accessed October 11 2018. https://www.w3.org/Submission/SWRL/.

W3C. 2012. "OWL 2 Web Ontology Language Primer (Second Edition)." W3C. Accessed October 11 2018. https://www.w3.org/TR/owl2-primer/.

Wadhams, J. 2015. "JsonLogic - Build complex rules, serialize them as JSON, share them between front-end and back-end." JsonLogic. Accessed October 11 2018. http://jsonlogic.com/.

Witherell, P., Krishnamurty, S., and Grosse, I.R. 2006. "Ontologies for Supporting Engineering Design Optimization." *Journal of Computing and Information Science in Engineering* 7 (2): 141-150. doi: 10.1115/1.2720882.

Zhang, H., Zhu, B., Li, Y., Yaman, O., and Roy, U. 2015. "Development and Utilization of a Process-oriented Information Model for Sustainable Manufacturing." *Journal of Manufacturing Systems* 37 (2): 459-466. doi: 10.1016/j.jmsy.2015.05.003.