A Systematic Approach to Developing Ontologies for Manufacturing Service Modeling

Farhad Ameri [1], Colin Urbanovsky, and Christian McArthur
Texas State University, Department of Engineering Technology
San Marcos, TX, U.S.A

Abstract. As engineering practices are increasingly becoming distributed and decentralized, formal engineering ontologies are becoming popular solutions for addressing the semantic interoperability issue in heterogeneous environments and bridging the gap between the legacy systems. Manufacturing Service Description Language (MSDL) is an ontology developed for formal representation of manufacturing services primarily in mechanical machining domain. In this paper, the metal casting extension to MSDL is introduced. This paper also introduces a systematic methodology for development of formal manufacturing ontologies that relies on incremental enhancement of explicit semantics. In particular, the proposed methodology focuses on the conceptualization phase and demonstrates how Simple Knowledge Organization System (SKOS) can be used early in the process for creating a controlled vocabulary, or thesaurus, in the domain of interest. The SKOS-based thesaurus helps identify the key concepts that will be used in an axiomatic ontology based on OWL-DL. Also, use of Semantic Web Rule Language (SWRL) for representation of constraint knowledge is discussed.

Keywords. Ontology, manufacturing supply chains, thesaurus, manufacturing service

1. Introduction

Manufacturing systems are under continuous transformation by the advances of cyber-enabled technologies such as cloud computing, wireless sensors, and web services. Automation technologies are transcending the borders of flexible and programmable automation and entering the intelligent automation area. In next generation automated manufacturing systems, planning and control are conducted in real-time by distributed software agents embedded in the hardware devices of manufacturing systems. The control units of future manufacturing systems have cognitive capabilities, such as learning, reasoning, and adapting to changes and they are integrated through a cohesive body of formal knowledge. In this context, formal representation of engineering knowledge is of utmost importance. In particular, there is an eminent need for development of various ontological models including product and process models. Ontologies play a key role in any distributed intelligent system as they provide a shared, machine-understandable vocabulary for information exchange among dispersed agents. In an environment in

[1] Corresponding Author: Assistant Professor, Department of Engineering Technology, 601 University Dr. San Marcos, TX 78666, E-mail: Ameri@txstate.edu

which agents have no previous knowledge of each other's type, capabilities, and interaction models, development of standard communication models with shared semantics is a necessity. Ideally, the common terminological system of an agent-based framework should provide the required building blocks for construction of a *shared body of knowledge* that can be understood and interpreted by all agents who subscribe to the terminology.

In the manufacturing domain, ontologies are in their early stage of development. Several ontologies have been proposed with the objective of facilitating knowledge management and information exchange across the extended enterprise. Some information models, such as Process Specification Language (PSL) [1], serve as neutral language for integrating several process-related applications (including production planning, process planning, workflow management and project management) throughout the product life cycle. Some others are aimed at providing a shared vocabulary for communication between machine control and process planning software applications [2]. Manufacturing ontologies vary with respect to the level formalism employed in the representation scheme. Some ontologies are mainly aimed at providing terminological means for information integration while some others are geared toward enabling advanced reasoning through providing sophisticated knowledge structures. It should be noted that heavier ontologies are not always preferred over lightweight ones due to the computational complexities associated with maintenance and management of heavily axiomatic ontologies. IEC 62264 standard [3], being developed by ISO TC 184/SC5 technical committee, is an example of a lightweight ontology that describes its domain through a set of object models. The purpose of this ontology is to facilitate the integration of business applications and manufacturing control applications within an enterprise. It mainly describes the attributes of the various objects in a manufacturing information model. Given the limited incorporation of explicit semantics in the model, it is placed at the lower end of the formality spectrum. ADACOR [4], on the other hand, is an example of heavyweight *domain ontology* based on a *foundational ontology* called DOLCE [5]. Foundational, or upper, ontologies are generic ontologies developed with the intention of formally describing various concepts that have similar interpretation across different domains. ADACOR is the ontology language of a holonic manufacturing system used for autonomous manufacturing control and it uses first-order logic as the knowledge modeling formalism. Most of the existing manufacturing domain ontologies are *descriptive* in nature in a sense that they provide the required means for describing manufacturing transactions and operations within a manufacturing system. However, there are few ontologies that deal with *characterization* of a manufacturing system itself with respect to *technological capabilities*. Capability characterization is increasingly becoming important as new manufacturing processes and technologies are being introduced and supply chains are becoming increasingly distributed. Manufacturing Service Description Language (MSDL) [6] is a formal domain ontology developed for representation of *capabilities* of manufacturing services. MSDL was initially designed to enable automated supplier discovery in distributed environments with focus on *mechanical machining services*. The objective of this paper is to introduce a structured procedure for developing ontologies for representing manufacturing *capability models*. Metal casting is selected as the domain of interest and MSDL is extended to include metal casting domain knowledge using the devised procedure.

There are several motivations for adapting a methodological approach to engineering ontology development. First, engineering knowledge models are often complex, multilayered, and highly interconnected models that need to go through a gradual and structured process of formalization and enrichment. Second, the knowledge users, who are typically not experts in knowledge representation and modeling, have to actively participate in knowledge modeling and validation in order to arrive at viable knowledge models. Without a well-defined and structured procedure, it is not easy to get all the ontology stakeholders involved effectively in the social process of knowledge capture and organization. Third, engineering ontologies that follow the same development path, lend themselves better to ontology mapping and merging.

This paper is organized as follows. A brief description of the ontology development methodology adopted in this work is described first. The next section provides an overview of the manufacturing capability model as conceptualized in MSDL. Various levels of capability model in MSDL as well as the core concepts are discussed later. The metal casting thesaurus is introduced afterwards followed by sections related to axiomatic casting ontology and casting rules.

2. Approach

The proposed methodology for ontology development in this work starts from a light-weight thesaurus, or controlled vocabulary, and guides the developers through gradual enrichment of the ontology by augmenting it with further semantics in the form of concept relationships, axioms, and rules. The proposed methodology uses Simple Knowledge Organization System (SKOS) [7] as a framework for creating a formal thesaurus. The created thesaurus helps ontology developers identify the key concepts of the domain of interest and also build partial taxonomies of the identified concepts and define some preliminary relationships, such as narrower and broader, between the concepts in the thesaurus. The identified concepts are further enhanced through introducing concept properties and imposing necessary and sufficient conditions on the concepts based on Description Logics (DL) [8] semantic model and Web Ontology Language (OWL) syntactic format. The output of this stage can be regarded as the *structural knowledge* of the domain of interest. The *constraint knowledge* is captured and formalized through introduction of rules modeled in Semantic Web Rule Language (SWRL), an extension of OWL that provides the ability to define complex rules and perform more advanced reasoning on the concepts in an ontology. As the ontology evolves, there is a need for continuous evaluation of the ontology with respect to the level of semantics incorporated in the ontology. Therefore, parallel to semantic evolution of the ontology, there is a need for ontology validation and verification with respect to accuracy and completeness using quantifiable metrics. Figure 1 demonstrates the major steps of the proposed procedure for engineering ontology development.

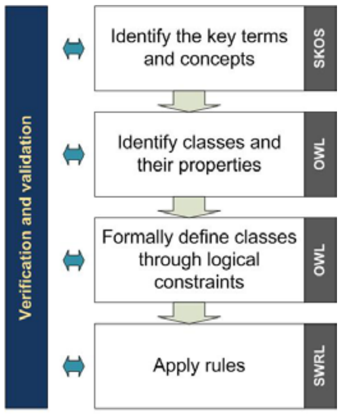

Figure 1 : The major steps of ontology development process

3. What is manufacturing capability model?

Since the proposed procedure is geared toward developing *capability* ontologies, it is in order to clearly define manufacturing capability early in this paper. For the purpose of this work, manufacturing capability is referred to as the limitations and the range of applicability of a manufacturing facility in transforming raw materials into products of increased value. More specifically, a capability model characterizes a manufacturing facility and its constituting elements including devices, machine, cells, operators, and processes with respect to the range of applicability, speed, cost, quality, and associated constraints and uncertainties. Based on this definition different dimensions of manufacturing capability include:

- *Technological capabilities* such as the resolution, accuracy, feed, speed, power, and automation level of the manufacturing equipment.
- *Operational capabilities* such as production capacity, throughput time, cost per unit, etc.
- *Geometric capabilities* such as shape producible, dimensions, wall thickness, work envelope, etc.
- *Quality capabilities* such as defect rate, surface finish, and tolerances.
- *Relational capabilities* that refer to interfaces with other systems and processes both hardware and software.
- *Stochastic capabilities* such as reliability, variations, etc.

The challenge in manufacturing capability modeling lies in developing conceptual capability models that characterize various facets of manufacturing capability in different levels of abstraction and also formalizing the semantics of the capability model in an unambiguous fashion.

Two example use cases for formal capability models include *autonomous design-to-fabrication* and *automated supply chain deployment*. Before introducing the metal casting thesaurus and ontology, a brief overview of MSDL and its core classes is provided next.

4. Manufacturing Service Description Language (MSDL)

As mentioned before, MSDL is a formal ontology since it is contains explicit semantics coded in a logic-based formalism. OWL-DL[2], a sub-language of OWL, is selected as the ontology language of MSDL. OWL is recommended by the World Wide Web Consortium (W3C) as the ontology language of the Semantic Web. OWL uses RFD/XML as the standard serialization; hence it has enough portability, flexibility, and extensibility for web-scale applications. Description Logic (DL) is supported by the Semantic Web meaning that OWL-based ontologies can be shared, parsed, and manipulated through open-source web-based tools and technologies, including multi-agent systems. The original purpose of MSDL was to serve as the ontology language of an agent-based framework for supply chain deployment.

[2] http://www.w3.org/TR/owl-guide/

4.1. Capability modem in MSDL

In MSDL, manufacturing capability is decomposed into five levels of abstraction, namely, and supplier-level, shop-level, machine-level, device-level, and process-level as shown in Figure 2. These five levels can collectively address the six dimensions of capability described earlier.

Supplier-level capability model deals with the capabilities of the supplier who runs a manufacturing facility. For example, expertise, skills, industry focus, product focus, and certifications are among the features of supplier-level capabilities. *Shop-level capability* describes the system-level capabilities of a manufacturing system owned by a supplier and described the system through its layout and material handling system and other supporting systems such as production planning and inventory control. Figure 3 shows the concept diagram of the Factory class used for describing shop-level capabilities.

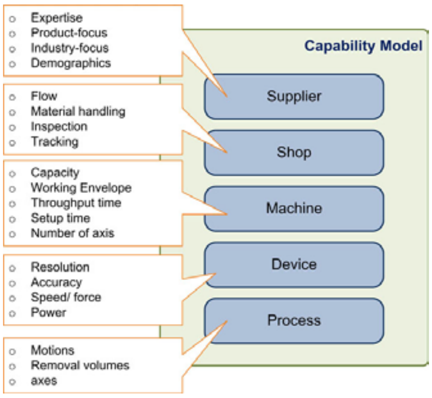

Figure 2 : Different Levels of the Manufacturing Capability Model

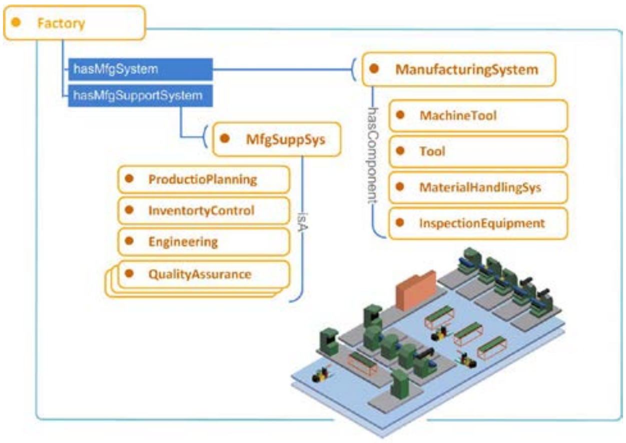

Figure 3: Factory class in MSDL is a sub-class of ProductionSystem

Machine-level Capability deals with characterization of the fabrication machines that are involved in conversion of the raw material into finished goods. Based on the proposed approach, manufacturing machines are represented through their components. Description of machines through their components is particularly beneficial in the context of Reconfigurable Manufacturing Systems (RMS) [9, 10] where conventional naming of machine tools is no longer applicable (Figure 4).

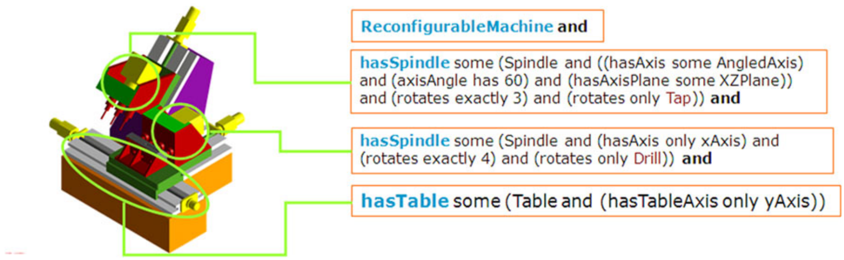

Figure 4: Ontological description of an RMS machine through its components

Device-level capability deals with characterization of devices, such as feed and spindle drives in a CNC machine, that are located at the lowest level of the hierarchy of the physical resources in any manufacturing system. In fact, the capabilities of the higher-level entities such as machine tools, and shop floor, can be inferred through aggregation of device-level capabilities. Therefore, the ontology should also cover the capabilities of the devices that form the basic building blocks of the physical factory. *Process-level capability* describes and characterizes manufacturing processes. Process is the most abstract entity in the capability model. The fundamental question in modeling process-level capability is how to describe the semantics of different manufacturing process such as mass change (either additive or subtractive), phase change, structure change, deformation, and assembly in a formal way. Different manufacturing processes call for different abstraction and conceptualization approaches.

4.2. Core Classes of MSDL

One of the core classes of MSDL is the *Service* class. Suppliers are the providers of manufacturing services and customers are the consumer of manufacturing services. In MSDL, supply and demand are represented by the *SupplierProfile* and *RFQ* (Request for Quote) classes respectively. As can be seen in Figure 5, a *Supplier Profile* has two major components, namely, the *Supplier* and the *Manufacturing Services* that the supplier provides. Services are further described through their associated processes, materials, resources, and supporting services. There are two primary methods for encoding further semantics (beyond concepts and properties) in MSDL. The first method is building taxonomies (i.e., explicit parent-child relationships) and the second method is axiomatic definition of classes. For example, the semantics of the *Industry* class are encoded in the form of an explicit taxonomy based on the North American Industry Classification System[3] (NAICS). Concepts such as *Process* and *Material*, on the other hand, are formally defined through *necessary and sufficient* conditions. Further constraints are applied on concepts using rules modeled in Semantic Web Rule Language (SWRL). SWRL rules are used by automated reasoners such as Pellet [11] and Hermit [12] to interpret the rules. For example, in a supply chain deployment scenario, supplier and customer agents can locally store instances of the MSDL concepts that pertain to their particular capabilities and needs.

[3] http://www.census.gov/eos/www/naics/

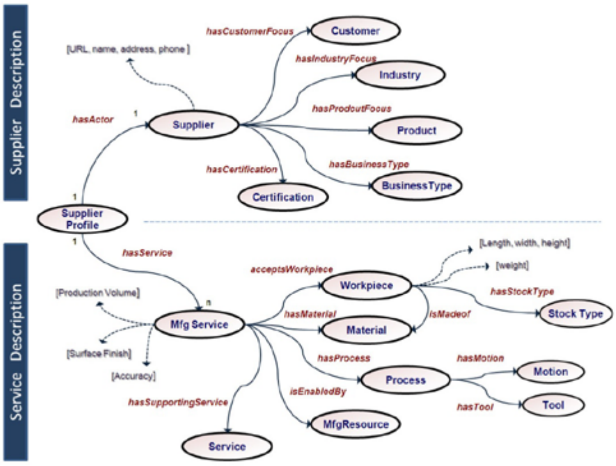

Figure 5: Concept diagram for the Supplier Profile class

Figure 6 shows the subclasses of the Process class in MSDL. As can be seen in this figure, the main subcategories of Process class in MSDL are addition processes, subtraction processes, consolidation processes, solidification processes, deformation processes, and property enhancing processes. The first revision of MSDL was limited to subtraction processes (i.e., conventional machining processes such as drilling, turning, and milling). This paper reports the metal casting extension of MSDL which is regarded as a solidification process. The metal casting ontology is developed based on a new methodology that starts with a semi-structured thesaurus. The casting thesaurus is discussed next.

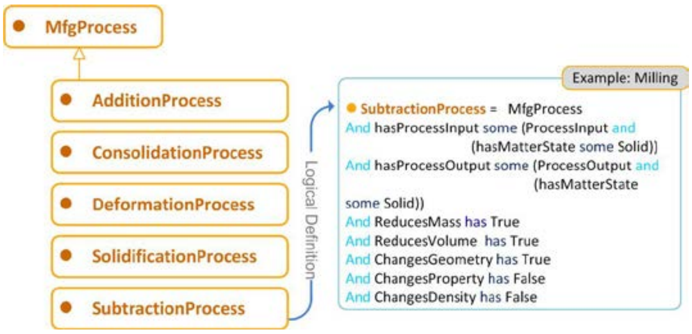

Figure 6: Manufacturing Process categorization in MSDL

5. Metal Casting Thesaurus

From a linguistic perspective, a thesaurus is a collection of terms connected through lexical relationships such as synonym, antonym, and metonym. International Standards Organization (ISO) defines thesaurus as “ the vocabulary of a controlled indexing language, formally organized with the aim of stating explicitly the relationships between the concepts" [13]. WordNet [7] is an example of a linguistic thesaurus developed for English terms. The process of integrating thesauri with information retrieval systems started in early 1990's and they gradually evolved from mere lexical resources towards powerful instruments for conceptual representation and knowledge organization [14].

A thesaurus improves the performance of electronic information retrieval systems through indexing documents by a controlled vocabulary in which terms and concepts are linked together through hierarchical relationships, associative relationships, and equivalence relationships. There exist several formal thesauri such as NAL Agricultural Thesaurus [15], Medical Subject Heading [16], and GEMET [17] (GEneral Multilingual Environmental Thesaurus) developed to support automated information retrieval in different application domains. However, in engineering domain, there are few thesauri that are specifically designed for information retrieval and knowledge organization. A lack of adaptation of controlled vocabulary in engineering can be attributed to the isolated nature of engineering activities, both in design and manufacturing, which has traditionally dominated the engineering realm. This has spawned a plethora of proprietary engineering information constructs that typically do not interoperate. Nevertheless, as engineering practices are increasingly becoming collaborative, interdisciplinary, and distributed, there is an eminent need for unifying frameworks, such as engineering thesauri and ontologies that can semantically connect apparently heterogeneous and disparate information models.

Although the need for developing comprehensive engineering thesauri endorsed by various stakeholders form government, industry, and academia, is a very real need that should be addressed eventually, this work is intended to explore how thesauri can be used for knowledge management in engineering domain. In other words, through developing a prototype thesaurus with a limited number of concepts, the authors investigate a systematic approach to engineering ontology development based on incremental enhancement of formal semantics embedded in the model. In a sense, a thesaurus can be regarded as a lightweight ontology that connects various concepts through elementary semantic relations. Since terms are regarded as the basic semantic units conveying abstract concepts, a thesaurus can be used for indentifying the core concepts and classes of a more complex ontology. The prototype thesaurus that is developed in this work helps in identification of the key concepts of the casting extension of the MSDL ontology. Since MSDL is an OWL-based ontology, SKOS (Simple Knowledge Organization System) modeling is used for thesaurus development. Similar to OWL, SKOS is based on Resource Description Framework (RDF), which allows concepts to be composed and published on the World Wide Web, linked with data on the Web and integrated into other concept schemes. SKOS provides a structured framework for creating different types of controlled vocabulary such as thesauri, concept schemes, and taxonomies. SKOS thesauri are concept-based, as opposed to term-based, in nature. In a term-based thesaurus, terms are directly connected together by semantic relationships whereas, in a concept-based thesaurus, semantic connection is at a concept level and terms are the lexical labels for the concepts, or units of thought, and may or may not have lexical relationships established among themselves. A SKOS thesaurus, like any other concept-based thesaurus, has a three-level structure (a) conceptual level, where concepts are identified and their interrelationships established; (b) terminological correspondence level, where terms are associated (preferred or alternative) to their respective concepts and (c) lexical level where lexical relationships are defined to interconnect the terms. The conceptual nature of SKOS is particularly useful in ontology development as it urges the developers to draw a distinction between terms and concepts and build a sound conceptual understanding of the domain of discourse.

To create the casting thesaurus, three main sources were utilized: 1) the casting textbooks 2) the web profiles of the providers of casting services and 3) DBpedia, the structured datasets gleaned from Wikipedia. DBpedia was used extensively to create the seed thesaurus early in the project by importing the relevant concepts and their associated sub-trees. Pool Party (PP), a thesaurus management system, was employed for creating the thesaurus. Figure 7 shows the concept diagram for the molding sand based on the SKOS terminology. Each concept in SKOS has exactly one preferred label (prefLabel) and can have multiple alternative labels (altLabel). For example, the sand that is used in casting is typically referred to as molding sand but foundry sand and casting sand are also used interchangeably to point to the same concept. In other terms, molding sand, casting sand, and foundry sand are synonyms in the casting thesaurus. The broader concept of the molding sand is sand. Silica sand and chromite sand are the narrower concepts; meaning that they are more specialized forms of the molding sand. Molding sand is also related to mold for example. Technically, all terms in the casting thesaurus can be related to one another. Therefore, broader, narrower, and related are the semantic relations used in any SKOS thesaurus. Also, each SKOS concept can have a definition provided in plain English or any other natural language.

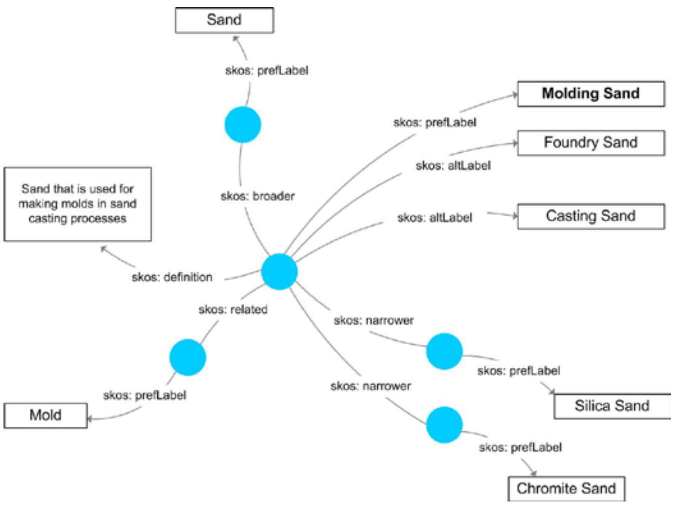

Figure 7: The concept diagram of the molding sand based on SKOS terminology.

One advantage of using SKOS is that any SKOS-based thesaurus can be connected to the Linked Open Data (LOD)[4] in order to reuse the existing datasets available on the LOD cloud. In fact, DBpedia, which was used for the purpose of creating the seed thesaurus in this work, is part of the LOD cloud currently containing more than 3.4 million concepts described by one billion relationships. A SKOS thesaurus can also be published and linked to the LOD cloud as RDF triples, thus allowing a larger community of users to validate and expand it. It should be noted that a SKOS-based thesaurus can serve as a self-sufficient ontology in many cases and adequately address the semantic needs of many knowledge organization and information retrieval systems. However, to enable more advanced reasoning capabilities, such as creating inferred taxonomies, the semantic

[4] http://linkeddata.org/

content of the thesaurus needs to be enriched by further constraining the identified concepts via logic-based restrictions.

6. Formal Ontology for Metal Casting

To further enhance the semantics of the created thesaurus and develop a formal axiomatic ontology, an OWL-based modeling is adapted in this work. A thesaurus can be evolved into an ontology by going through several formalization steps. In the first step of formalization, core concepts of the domain of interest, already identified in the thesaurus, are represented through formal classes with known properties. There isn't always a one-to-one mapping between the concepts in the thesaurus and the concepts in the ontology. Instead, a cluster of concepts in the thesaurus may define a single concept in the ontology.

The concepts in the casting thesaurus have no properties assigned to them but in the ontology, it is necessary to provide more details about each concept through introducing some attributes that describe each concept. For example, as can be seen in Figure 8, the weight and dimensions of the die casting machine are regarded as the properties of the machine with numeric values. The properties sometime take Boolean or literal values at their range. For instance, *isHotchamber* is a Boolean property used to determine if a die cast machine is hot chamber or cold chamber. At the next level of formalization, concepts are connected to one another through object properties. For example, the Die Casting Machine is related to the Die Casting Process through *hasProcess* relation or Sand Casting process is connected to Mold through *hasMold* property. The concepts, once connected, create a semantic network that defines the main structure of the ontology.

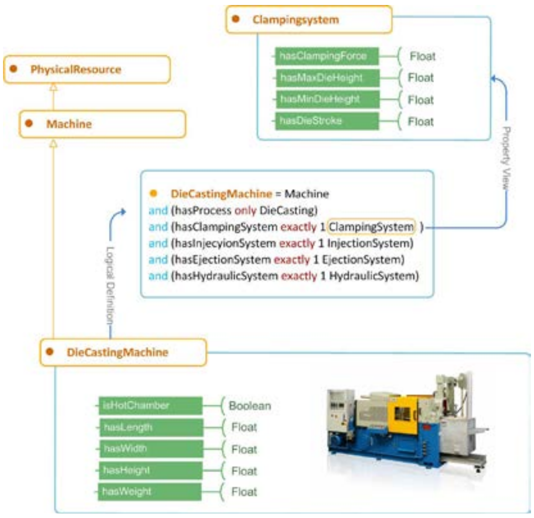

Figure 8: Logic view and property view for the Die Casting Machine in MSDL

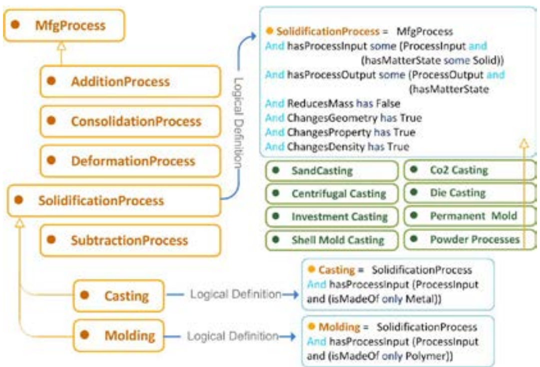

Figure 9: Formal definition of the Solidification Process in MSDL

At the third level of formalization, concepts are further annotated by axioms to form defined concepts. Defined concepts are basically formed through intersecting multiple conjuncts that collectively serve as a set of necessary and sufficient conditions that logically characterize the concepts. For example, concepts such as *Process* and *Material* are formally defined through necessary and sufficient conditions. Figure 9 provides the formal definition of the solidification process in MSDL. As the name implies, a solidification process is a *MfgProcess* that changes the state of its input material from either liquid or powder to solid. Casting, molding, and powder processes are examples of the solidification process. These processes do not reduce the mass of its input material but change the density and mechanical properties and typically change the geometry of the input material as well. Casting is a specific case of the solidification process in which the input material is a metal. The definitions of Sand Casting and Die Casting, as two sub-classes of the casting process, are provided in Figure 11 and Figure 10 respectively. The definition of sand casting implies that it is a casting process in which the mold is expendable *and* is made of sand *and* it is a gravity pouring process *and* the castable materials include cast iron, aluminum, bronze, brass, and stainless steel. The definition of the die casting process describes it as a casting process with a permanent mold made of steel. This process can be applied to nonferrous materials and does not use gravity for pouring. In this way, all casting processes can be uniquely defined using logical axioms.

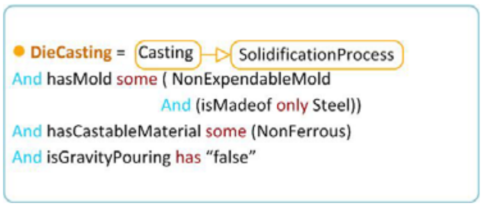

Figure 10: Formal definition of the Die Casting process in MSDL

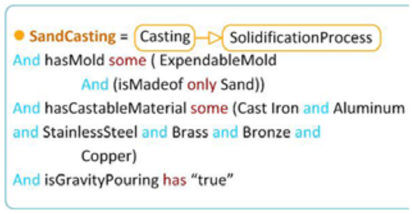

Figure 11: Formal definition of the Sand Casting process in MSDL

The concepts embedded within each definition may have formal definitions themselves. For example, *Aluminum* is not merely a string of characters but it is a subclass of nonferrous metals with known chemical and physical properties formally defined in the ontology. Figure 12 shows the formal definitions of aluminum and stainless steel in MSDL. DL reasoners, such as Racer [18] or Pellet [19] can be used to classify a flat set of defined classes and arrive at an inferred taxonomy. In other words, with an axiomatic approach for encoding semantics, there is no need for creating an explicit taxonomy of concepts from automated information processing standpoint. However, to make ontologies more readable and comprehensible for human developers, it is recommended to build explicit taxonomies while developing a formal ontology. Concept classification is one of the cornerstones of similarity measurements in formal ontologies.

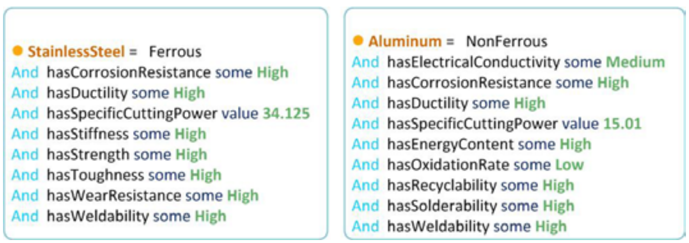

Figure 12: Formal definitions of Aluminum and Stainless Steel in MSDL

7. Metal Casting Rule Modeling

The next step of semantic enhancement of an ontology entails creation of the rules that convey further information about the concepts and their relations. In fact, the richness of a formal ontology depends on the level of details incorporated in the axiomatic definition of the concepts as well as the number and diversity of the rules encoded in the ontology. Rules are the main enablers of ontological reasoning and inference by machine agents. As the complexity of queries increases, so does the significance of knowledge-based reasoning and inference.

Human reasoning and cognition mechanism has been the subject of research in the Artificial Intelligence (AI) community for several decades now. Expert systems developed in AI domain are intended to imitate the way a human expert analyzes a particular situation by using different reasoning techniques such as rule-based, case-based, fuzzy logic, neural networks, and Bayesian networks [20]. Rule-based techniques, due to their structured nature, are the most common techniques adopted in expert systems [21].

OWL has the required level of expressivity for representing *structural knowledge* through concepts and the relationship between the concepts. Also it is possible to define concepts using different types of restriction such as *quantifier*, *cardinality*, and *hasValue*. However, for rule representation, OWL fails in providing the necessary building blocks especially when it comes to complex rules. To fill this gap, OWL was supplemented by a rule modeling language referred to as Semantic Web Rule Language (SWRL). SWRL is an extension of OWL that provides the ability to define complex rules and perform more advanced deductive reasoning about concepts in an ontology. SWRL rules are used by automated reasoners such as Pellet [19] and Hermit [22] to interpret the rules. SWRL is built on OWL DL and shares its formal semantics.

SWRL rules are composed of an antecedent (body) and a consequent (head). Both body and head are composed of positive conjunction of atoms. A SWRL rule follows an "if-then" logic. If the antecedent, or premise, holds true, the consequent must be true as well. For example, the flowing rule states that if a part is made of aluminum and its minimum wall thickness is greater than or equal to 3 mm, then it can be sand casted.

```
Part (?p) ^ isMadeOf (?p, ? m) ^ Aluminum (?m) ^ hasMinWallThickness (?
th )^ swrlb:greaterThan (?th, 3)
    -> SandCastAblePart (?p)
```

In essence, this rule creates a temporary class called *SandCastablePart* and any instance of the class Part that satisfies the conditions given in the body of the rule becomes the subclass of this temporary class. This classification utility is especially useful for narrowing down the search space when, for example, the goal is to find the parts that can be manufacturing using sand casting process. SWRL rules can be attached to the OWL ontology or they can be applied programmatically on the fly. It is recommended to apply the rules programmatically especially if the rules are parametric.

Rules can be used for multiple purposes in the casting ontology. For example, design validation can be conducted automatically using SWRL rules if the design itself is represented in OWL. Design validation in the context of an ontology can be translated into a consistency checking process. As another example, a rule-based approach can be adapted for finding the qualified suppliers for a particular casting service. The following rule describes a query for a casting service that accepts parts heavier than 100 pounds, with the tolerance of 0.01 inch or less, surface finish of 64 microinch or less, and production volume of 500 or more.

```
Service (?s)
^ hasProcess (?s, ?pr) ^ Casting(?pr)^ hasPart(?s, ?pt)
^ hasWeight (?pt, w?) ^ swrl:greaterThan (?w, 100)
^ hasAccuracy (?s, ?ac) ^ swrl:smallerThan (?ac. 0.01)
^ hasSurfaceFinish (?s, ?sf) ^ swrl:smallerThan (?sf, 64)
^ hasProductionVolume (?s, ?pv) ^ swrl:greaterThan (?pv, 500)
->DesirableService (?s)
```

This rule creates a temporary class called *DesirableService* that subsumes all instances of the Service class that satisfy the requirements. Another rule is required for identifying the suppliers who provide the described service. This rule is constructed as follows:

```
SupplierProfile (?sp) ^ hasService (?sp, ?s) ^ DesirableService (?s)
->  QulifiedProfile (?sp)
```

It should be noted that rules such as above can be expressed in OWL as class subsumption (e.g. SupplierProfile and (hasService some DesrirableService) subClassof QualifiedProfile). However, such expressions require addition of permanent classes such as QualifiedProfile or DesirableService to the ontology which will make the ontology more application-dependent and less generic. In general, with the aid of rules, the dynamic classes that have operational purposes can be kept separate from the conceptual and generic (static) classes that constitute the main body of the ontology. Although, SWRL is more expressive that OWL DL alone, this extra expressivity comes at the expense of risk of undecidability. Therefore, care should be taken when introducing SWRL rules. Especially one should avoid binding the rules to the individuals that are not known to the ontology as it renders the ontology undecidable.

8. Conclusions

The objective of this paper was two-fold: First, to report the metal casting extension of MSDL and second, to propose a systematic approach to developing manufacturing capability ontologies. The metal casting extension is currently limited to sand casting and die casting but in the future, it will be extended to all metal casting processes and equipment. The proposed approach for ontology development suggests breaking down the capability model into five distinct levels, namely, supplier-level, shop-level, machine-level, device-level, and process-level. Also, the proposed approach recommends identifying the concepts within the ontology through creation of a thesaurus early in development process. Simple Knowledge Organization System (SKOS) was used as the thesaurus modeling formalism. The adoption of SKOS as a common model to represent manufacturing thesaurus allows standard representation of conceptual thesauri. With a standard representation, linking of different manufacturing thesaurus is facilitated and therefore, multiple thesauri can be merged and combined to arrive at more comprehensive thesauri with wider scopes. The joint use of SKOS, OWL, and SWRL would offer a high level of flexibility with respect to arriving at a trade-off between expressivity requirements and computational complexity constraints. Future work in this area include enhancement of the developed thesaurus and ontology as well as and creating the necessary search tools that leverage the semantic structure of the developed knowledge model for different use cases.

References

[1] M. Gruninger, C. Menzel, The Process Specification Language (PSL) theory and applications, AI Mag., 24 (2003) 63-74.
[2] L.M. Deshayes, O.E. Beqqali, A. Bouras, The use of process specification language for cutting processes, International Journal of Product Development, 2 (2005) 236-253.
[3] I.S.O. (ISO), IEC 62264 -1 Enterprise Control Systems Integration - Part 1: Models and Terminology in, ISO TC 184/SC5 technical committee Geneva, 2003.
[4] S. Borgo, P. Leitao, The role of foundational ontologies in manufacturing domain applications, 2004.
[5] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, L. Schneider, Sweetening ontologies with DOLCE, in: A.B.V.R. GomezPerez (Ed.) Knowledge Engineering and Knowledge Management, Proceedings: Ontologies and the Semantic Web, 2002, pp. 166-181.
[6] F. Ameri, D. Dutta, An Upper Ontology for Manufacturing Service Description, ASME Conference Proceedings, (2006) 651-661.
[7] S.G. Kolte, S.G. Bhirud, WordNet: A Knowledge Source for Word Sense Disambiguation, International Journal of Recent Trends in Engineering, 2 (2009) 213-217.
[8] F. Badder, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, The description logic handbook: Theory, implementation and applications 2nd ed., Cambridge University Press, 2010.
[9] H.A. ElMaraghy, Flexible and reconfigurable manufacturing systems paradigms, International Journal of Flexible Manufacturing Systems, 17 (2005) 261-276.
[10] M.G. Mehrabi, A.G. Ulsoy, Y. Koren, Reconfigurable manufacturing systems: Key to future manufacturing, Journal of Intelligent Manufacturing, 11 (2000) 403-419.
[11] Pellet: OWL 2 Reasoner for Java, (2011).
[12] Hermit OWL Reasoner, (2011).
[13] Guidelines for the establishment and development of monolingual thesauri, in, nternational Organization for Standardization Geneve, 1986.
[14] J.A. Pastor-Sanchez, F.J.M. Mendez, J.V. Rodriguez-Munoz, Advantages of thesaurus representation using the Simple Knowledge Organization System (SKOS) compared with proposed alternatives, Inf. Res., 14 (2009) 31.
[15] Agricultural Thesaurus and Glossary Home Page, in, 2011.
[16] Medical Subject Heading, (2011).
[17] GEMET Thesaurus - Eionet, (2011).
[18] RACER DL Reasoner, in, http://www.sts.tu-harburg.de/~r.f.moeller/racer/, 2011.
[19] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur, Y. Katz, Pellet: A practical OWL-DL reasoner, Web Semantics, 5 (2007) 51-53.
[20] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 3 ed., Prentice Hall, Englewood Cliffs, New Jersey, 2007.
[21] J. Munguia, J. Lloveras, S. Llorens, T. Laoui, Development of an AI-based Rapid Manufacturing Advice System, International Journal of Production Research, 48 (2010) 2261-2278.
[22] Protege OWL, in, 2011.