# On Enhancing Communication of Manufacturing Service Capability Information

Boonserm Kulvatunyou, Nenad Ivezic, Yunsu Lee

Systems Integration Division, National Institute of Standards and Technology
Gaithersburg, MD 20899, U.S.A.

*Abstract*

A *manufacturing service capability (MSC) model* is essential for correct communication of MSC information between customers and suppliers. MSC information elicits service details such as locations, areas of specialization, capacities, certifications, software capabilities, and material processing capabilities. This information is carried by semantics embedded in the MSC model's schema and MSC descriptions, which are the schema-compliant instances representing suppliers' capabilities. Presently, as in the case of web portal-enabled communications between customers and suppliers, this information is communicated using models that provide limited semantic precision. These models are also proprietary to specific industry communities; hence, access to MSC information across communities is also limited. In this paper, we describe deployment analysis of the merged-model-based semantic mediation approach to enhance access to and precision of proprietary MSC models that are encoded in relational databases (RDBs). The approach relies on a merged ontology, mappings between the Web Ontology Language- (OWL) encoded proprietary MSC models and the merged ontology, and OWL-enabled description logic inferences. We describe the analysis with an example manufacturing sourcing use case explaining the semantic mediation steps enabled by the approach. We characterize the first step that encodes the proprietary MSC model in OWL using three alternative encoding conventions and discuss their corresponding resulting behaviors associated with the semantic mediation. An analysis of the behaviors leads to a conclusion that the ontology-oriented encoding convention is most suitable of the three alternatives for semantic mediation in OWL. In the next step, development of the mapping ontologies between the OWL-encoded proprietary models and the merged ontology is discussed. Then, in the deployment stage, it is shown that the semantic mediation can be deployed in two ways, namely, single community, which shares same proprietary MSC model, and multi-community, which uses differing proprietary MSC models. The deployment analysis shows that the single community deployment introduces shared MSC semantics for precise and greater access to MSC information within the community while the multi-community deployment introduces shared MSC semantics for interoperable access to MSC information across the communities.

## 1 Introduction

One way in which small and medium enterprises (SMEs) can increase their capacity utilizations is to diversify their customer base. This approach, if successful, also increases business and supply chain resiliency to demand fluctuations and supply disruptions. A strategy that can help SMEs in diversifying their customer base is to enhance their abilities to communicate their manufacturing service capability (MSC) information. MSC information elicits service details such as locations, areas of specialization, certifications, software capabilities, and material processing capabilities. Customers and suppliers exchange MSC information by communicating MSC descriptions compliant to an MSC schema[1]. MSC descriptions allow for customers and suppliers to communicate their product, manufacturing requirements, and capabilities during the manufacturing sourcing processes.

Presently, alternative approaches for representing MSC information use MSC models that are suboptimal for potential customers to communicate efficiently with relevant suppliers. The reason is that these MSC models are proprietary to specific communities and have limited precision. In this paper, we address these issues, by describing a deployment analysis of the merged-model-based semantic mediation approach [Kulvatunyou et al. 2012] that enhances precision, access, and interoperability of proprietary MSC models that are encoded in relational databases (RDBs). The approach is based on the formal description logic represented in the Web Ontology Language (OWL) [W3C 2009a].

The rest of the paper is organized as follows. In the next section, we analyze three existing supplier portals in the context of a specific customer sourcing use case. The analysis illustrates the current state of practice in today's supplier portals and defines the target behaviors for MSC models to support. Then, we introduce two detailed proprietary MSC models represented as relational models and describe their semantic issues and differences. Then, we describe the merged-model-based semantic mediation approach and its deployment analysis to enrich the proprietary MSC models. We show the target behaviors by applying the semantic mediation approach on the examples of MSC models captured in the relational databases with varying schemas. The description of the application includes the steps and artifacts needed to deploy the semantic mediation approach. We present results that show that the single community deployment enables precise and higher access to MSC information within the community while the multi-community deployment enables an additional benefit of interoperable access to MSC information across the communities. Next, we provide a review of previous works in semantic mediation and those that complement our approach. Finally, we provide our conclusion.

## 2 Manufacturing Sourcing Use Case Analysis

Suppliers' manufacturing service capability (MSC) information represented by MSC models can be used to support communications of the MSC information in complex distributed manufacturing use cases such as manufacturing sourcing, supply chain planning, and supplier integration. Sourcing is a process of finding and qualifying suppliers for goods or services. Manufacturing sourcing, more specifically, focuses

---

[1] Since the semantics of MSC information is generally embedded in both MSC descriptions and their corresponding MSC schemas, for the purpose of discussion in this paper we use the term MSC model to refer to the combination of both MSC description(s) and their schemas. As a simple example, a supplier's MSC description may state the supplier to have the Wire EDM (Electro Discharge Machining) capability, which is a piece of MSC information. Then, the corresponding MSC model schema that states the Wire EDM to be a kind of EDM capability is an additional piece of MSC information. More complex axioms and rules may be contained in the MSC model schema to encode MSC information.

on qualifying suppliers for required manufacturing service capabilities (production capacity included). Motivated by existing commercial supplier information sharing portals which enable MSC information communication between customers and suppliers for manufacturing sourcing, we analyze the current state of practice in those portals in the context of a specific manufacturing sourcing use case. The purpose of the analysis is to discuss the semantic issues that hinder precise communication of MSC information and demonstrates the target behaviors that MSC models are required to support[2]. Three existing supplier portals, named Portal-A, B, and C, are analyzed in the context of the customer's use case requirements.

In the use case, a customer needs to send a request for a supplier to manufacture an injection mold. From the product perspective, the mold is a complex geometry part with sharp inside corners that requires tolerance tighter than 0.025 cm and that is made from pre-hardened stainless steel. From the process requirement perspective, the customer needs a supplier with Sinker EDM machining capability with a work envelop of up to 15 cm. The process must be able to achieve 0.025 cm tolerance and operate on a pre-hardened stainless steel work piece.

Next, we discuss the semantic issues resulting from an analysis of the publicly accessible information from three manufacturing service portals and in the context of the above use case requirements. The discussion is organized into three categories of semantic issues including 1) low fidelity MSC model; 2) semantic ambiguity; and 3) semantic modeling conflicts.

## 2.1 Low Fidelity MSC Model

On one hand, the manufacturing service capability (MSC) descriptions at these portals are insufficient for this broader domain, as the MSC models lack the attributes covering the manufacturing service capability criteria identified in our use case. On the other hand, the descriptions are mostly in textual formats, which makes it difficult to search and request for suppliers' services. In addition, most of the suppliers do not provide detailed MSC descriptions. The multitudes of these deficiencies constitute a low fidelity MSC model.

In Portal-A, the suppliers' MSC descriptions are either unstructured text or semi-structured text (i.e., in a formatted table). The only controlled vocabulary provided is the taxonomy of manufacturing service categories such as CNC Machining service, Drilling service, Electro Discharge Machining service (see Figure 1A for examples of manufacturing service categories). That is, suppliers can include (or exclude) whatever details they choose. A small portion of suppliers provide detailed MSC descriptions. For example, out of 1,693 suppliers registered in the portal under the EDM service category, only 31 provide detailed MSC descriptions (which is less than 2%)[3]. Nevertheless, these descriptions are either in a semi-

[2] In this paper, we focus only on engineering type of information within the MSC information and do not analyze other types of information (e.g., supplier's location or contact information) necessary to fully support customer-supplier interaction.

[3] In the deep hole drilling category, which is another category that we have investigated, only 3 out of 403, or less than 1%, of suppliers provide capability details. Upon discussions with the portal operator, a cited reason is that suppliers may not know how to describe their manufacturing capabilities, indicating that a richer manufacturing service model with guidance could help.

structured text or unstructured text format. Communication of MSC information using such representation will make it difficult for a supplier sourcing decision-making process when it needs to determine whether the supplier meets the required capability criteria. This is because an automatic determination can rely only on inaccurate keyword and string-based matching. For example, if one supplier indicates its tolerance capability in text as "+/- .0025" and another as "+/- 0.0025", a string-based matching approach may incorrectly determine that the first supplier cannot meet a tolerance requirement input of 0.0025 (such numerical text is also typically not indexed as a keyword). In addition, a string-based matching for the tolerance requirement input of 0.025 in the use case (which is achievable by the 0.0025 tolerance capability) will incorrectly determine that neither of the suppliers meets the required manufacturing service capability.

In Portal-C, suppliers' MSC descriptions are even more limited and text-based. We observed that suppliers at this portal specify little to no detail about their manufacturing service capabilities. Each supplier is allowed a single aggregated MSC description that can be associated to one or more manufacturing service categories. This single aggregated service description makes the capability details ambiguous. For example, if a particular MSC description is associated to both the Deep Hole Drilling and EDM Machining categories, it would be unclear whether a provided tolerance capability references the Deep Hole Drilling category or the EDM Machining category.

Portal-B's MSC model has higher fidelity than those of the previous two portals (see Figure 1B). That is, manufacturing service capabilities are represented with structured attributes that enable communication required for precise, multiple criteria sourcing decisions. As opposed to Portal-C, information precision is higher in Portal-B in that suppliers can have multiple MSC descriptions each of which is associated to a specific manufacturing service category. Nevertheless, Portal-B's manufacturing service categories are too broad. Its MSC model also lacks attributes for capturing important characteristics, such as tolerance for the EDM service. Even though some suppliers provide tolerance capability in the text of the MSC description, such communication is ineffective due to the string-based matching limitation described earlier.

The use case analysis indicates that the MSC information and its associated representations in today's supplier portals are insufficient for precise communication required in the sourcing use case. The impact of this is twofold. On one hand, a sourcing decision-making process may fail to identify situations where existing suppliers' capabilities in fact match the manufacturing requirements, resulting in missed opportunities. On the other hand, the sourcing decision-making process may incorrectly identify situations where existing suppliers' capabilities in fact do not match the manufacturing requirements, resulting in potential suboptimal and inefficient production solutions. Enrichment of MSC models and associated detailed MSC descriptions would help improve the situation.

## 2.2 Semantic Ambiguity

In Portal-A and Portal-C, manufacturing service categories are the primary mechanism to represent MSC information. They respectively have 2,683 and 496 categories to represent MSC information under the contract/custom manufacturing type of services. On the other hand, Portal-B relies on both manufacturing service categories and attributes (i.e., it is more feature-oriented than the other two portals). It has 125 searchable categories and an average of approximately 5 attributes per category for contract manufacturing. Each attribute has approximately 10 possible values on average. To convert the Portal-B representation to the equivalent of the Portal-A/C category-oriented representation, a classification system would be required with at least 270 million possible categories. Even though this indicates that Portal-B potentially has a higher fidelity of information than the other two portals, we found the semantics of these categories and attributes both by definitions and implementations to be ambiguous. This section discusses this issue.

Figure 1A shows a few common cases of semantic ambiguity in Portal-A. The first case is due to a *vague definition*. The `Complex & Difficult Machining` category is an example of such cases. It is a concrete category where suppliers can register their services. However, it is ambiguous what characteristics are necessary for suppliers to be a complex and difficult machining service provider.

The next semantic ambiguity case is due to *redundancy*. `Sinker EDM` and `Ram EDM` are two separate categories in Portal-A. However, the two terms have the same meaning according to Jameson (2001); and there is no explanation about the two categories within the portal. Suppliers who have the sinker EDM manufacturing service capability would need to register to both categories in order that they can be included for sourcing under either of the categories. We observed that there were 177 suppliers registered under the `Sinker EDM` category while there were 101 suppliers registered under the `Ram EDM` category. Clearly, some suppliers are not aware that the two terms are semantically the same.

Another case of semantic ambiguity in Portal-A is due to *lack of entailment* of the subcategory relationship with respect to the associated services. For example, `Sinker EDM` is a subcategory of the `EDM` category. However, if a portal user initiates a request to suppliers who have some manufacturing services registered (i.e., classified) under the `EDM` category, suppliers registered to the `Sinker EDM` category, but not directly to the parent `EDM` category, may not receive the request. This is also apparent from the difference between the number of suppliers shown under the EDM category (1,700) and the sum of suppliers shown under EDM children categories (2,500). In other words, suppliers registering to a subcategory are not automatically registered to the parent category.

**Figure 1 should be inserted about here**

Similar semantic ambiguities are observed in Portal-B (see Figure 1B which illustrates a portion of the Portal-B manufacturing service categories). For example, Portal-B has `CNC Machining Services` as well as `Machine Shop Services` categories that are similar in both the published descriptions and data structure, but our analysis found them to have no formal relationship. The `CNC Machining Services` category is listed as a subcategory of `Machine Shop Services`; hence, suppliers registered to `CNC Machining Services` should have been automatically registered to `Machine Shop Services`. However, similar to that of Portal-A, unless suppliers register their services to both categories they cannot be communicated for sourcing under the other category; and it is also apparent from the fact that the number of suppliers under the child `CNC Machining Services` (5,523) is larger than that of the parent `Machine Shop Services` (4,459).

Another semantic ambiguity issue in Portal-B is that a manufacturing service capability can be implied from both the service category semantics as well as from the manufacturing process capability property semantics. This issue is not present in Portal-A because its MSC model does not use properties to convey a manufacturing service capability. For example, suppliers are considered to have some EDM machining capabilities when they are registered to the `EDM Machining Services` category or when they are registered to a non-EDM service category but said to have EDM process capabilities (e.g., `Machine Shop Services` category → `Electrode EDM` property value in Figure 1B).

We have observed similar semantic ambiguity issues discussed in this section in Portal-C. We will demonstrate in our running example that Portal-A and Portal-B MSC models can be semantically enriched by using semantic mediation to remedy the semantic ambiguities discussed in this section. More importantly, this semantic enrichment is a modular addition that can be achieved without making any modification to the portals' manufacturing service models.

## 2.3 Semantic Modeling Conflict

We have observed that MSC models have differing representations across the portals and in some cases can be conflicting with each other. This issue makes it harder for suppliers and customers to communicate information across portals (i.e., submit a request and respond to a request). The semantic modeling conflicts are categorized into four types, and we discuss them respectively below. The four types of semantic modeling conflicts are: 1) expressivity conflict, 2) structural conflict, 3) generalization conflict, and 4) naming conflict [Sheth and Kashyab 1992, Park and Ram 2004]. These types of conflicts are not mutually exclusive though.

### 2.3.1 Expressivity conflict

The expressivity conflict means that each portal has differing sets of searchable concepts or combinations of concepts. As mentioned in the semantic ambiguity section, if we were to convert Portal-B's feature-oriented representation into Portal-A's and Portal-C's taxonomy-oriented representation, it would result in millions of categories. Therefore, we can say in general that Portal-B's MSC model is more expressive than those of Portal-A and Portal-C (for the contract manufacturing area). Figure 2A illustrates Portal-A's EDM subcategories that were not shown earlier in Figure 1A. Figure 2B shows Portal-B's possible EDM machining service capability descriptions. In Portal-A, the categories `Micro Hole EDM` and `High Speed Small Hole EDM` are available as specific EDM capabilities in addition to `Small Hole EDM`; while in Portal-B only `Small Hole EDM` is available. In the area of process capability related to the small hole EDM, Portal-A is more expressive than Portal-B. On the other hand, Portal-B has other features that allow the `Material` and (part) `Length Capacity` to be specified with choices of values such as `Stainless Steel`, `Hardened Metal` and `Less than 2.5cm, 2.5cm - 7.5cm, 7.5cm - 15cm`, respectively. Consequently, expressivity conflicts exist between the two portals. Expressivity conflicts may result in sourcing communications that use terms alien to another party. Making sourcing decisions based on such communications will likely be problematic. For example, the term Electrode EDM is formalized as an EDM capability only in Portal-B. If this term is used to communicate a request to suppliers in Portal-A, we have observed that the request is submitted to suppliers who sell EDM electrodes instead of suppliers who have an Electrode EDM manufacturing capability.

Another type of expressivity conflict may be viewed as the data value or data precision conflict. For example, we observed that most of the manufacturing capability details on Portal-A, although not represented in a structured format, describe the part length capability with only the maximum value. On the other hand, the part length capability is represented as multiple value ranges in Portal-B (Figure 2B).

**Figure 2 should be inserted about here**

### 2.3.2 Structural conflict

The structural conflict means that portals have differing ways of representing MSC descriptions. For example, the small hole EDM manufacturing service capability is represented as a manufacturing service category in Portal-A while it is an attribute value in the case of Portal-B. This is illustrated in Figure 2A and Figure 2B. The structural conflicts between portals would require suppliers and customers to understand different representations in order to communicate across the portals. Similarly, suppliers who wish to maintain their MSC descriptions at multiple portals need to understand these different ways of navigating and registering information to these portals. For example, a supplier who has registered to the `Exotic Metal EDM` category in Portal-A would need to register to the `EDM Machining Services` category in Portal-B with the `Materials` attribute pointing to `Precious Metal`. Clearly, this may cause significant cost inefficiencies for suppliers who interface to multiple portals to support multiple industries.

**Figure 3 should be inserted about here**

### 2.3.3 Generalization Conflict

The generalization conflict occurs when the same concept is represented in two models with different abstractions. Consider the injection-mold-making concept as an example. The concept has differing higher-level concepts across the three portals. It is represented as `Injection Molds` under the `Dies and Molds`, which is under `Machinery, Tools, and Supplies` category in Portal-A while in Portal-B it is represented as `Injection Molding` three levels down the hierarchy under the `Contract Manufacturing & Fabrication/Part Fabrication Services/Mold Making Services` categories (see Figure 3B). Moreover, in Portal-A the `Machinery, Tools, and Supplies` category is separate from the `Custom Manufacturing & Fabricating` category which semantically matches the `Contract Manufacturing & Fabrication` category in Portal-B. In Portal-C, the concept is modeled as the `Injection Mold Making` category two levels down the hierarchy under the `Manufacturing & Industrial Product/Tool, Die, and Mold Making` categories. The `Manufacturing & Industrial Product` seems to be a merger of Portal-A's `Custom Manufacturing & Fabricating` and `Machinery, Tools, and Supplies` categories.

The generalization conflict not only requires both the customer and supplier to communicate service requests and responses differently across portals, but the communication results will also differ across portals. For example, Portal-A and Portal-C have all dies and molds services under a single category. Figure 3 shows that all of them are grouped under the `Dies & Molds` and `Tool, Die, and Mold`

Making categories, respectively. On the other hand, Portal-B has similar dies and molds services scattered over two categories namely Mold Making Services and Tool and Die Makers.

#### 2.3.4 Naming Conflict

The naming conflict occurs when the same concept is labeled with differing terms. For example, the same type of EDM capability is called Sinker EDM as well as Ram EDM in Portal-A, while it is called Electrode EDM in Portal-B (Figure 1). In another example, the injection mold making service concept is called Injection Molds, Injection Molding, and Injection Mold Making in Portal-A, B, and C, respectively (Figure 3). Similarly, the extrusion die making service concept is called Extrusion Dies in Portal-A while it is called Extrusion and Extrusion Die Making in others. The variations of these terms make it harder for both suppliers and customers to communicate.

While semantic mediation cannot deal with the low fidelity manufacturing service model issue, it can remedy the semantic ambiguity and modeling conflict issues. We will demonstrate using our running example (i.e., the specific use case request for injection mold manufacturing in Section 2) that the semantic mediation can remedy semantic modeling conflicts. For example, suppliers indicating the electrode EDM capability either via the category association (Portal-A or Portal-C) or feature association (Portal-B) can be searched and communicated using either the category or feature-based approach without making any modification to proprietary models. Terms from any of the portals can be used. In another example, communication using the small hole EDM concept can also imply an interest in subsumed concepts (e.g., micro hole EDM, high speed small hole EDM) in another portal.

# 3 Proprietary MSC Models

In this section, we introduce two detailed proprietary MSC models represented as relational databases and describe their semantic issues and differences. The models formally represent real MSC information related to the EDM capability that we have observed from publicly accessible suppliers' data in Portal-A and Portal-B discussed in the previous section. Then, we specify the target behaviors for the semantic mediation in section 4 to resolve the semantic issues and differences between these models.

## 3.1 Portals' Database Schemas

Figure 4 below shows the relational schema of Portal-A's MSC model. In this model, MSC descriptions of any type of service are attached to the single, generic Capability table. In other words, data structures of different types of services are the same in Portal-A. This is not the case in the Portal-B's schema. Figure 5 shows the relational schema of Portal-B's MSC model related to the EDM machining service. For comparison purpose, additional Portal-B's schema related to the machine shop and mold making services are illustrated in Figure 21 and Figure 22 in Appendix A. It can be seen from Figure 5, Figure 21, and Figure 22 that differing types of services use differing sets of tables to capture MSC information resulting in differing data structures for each service type. In addition, the Capability table in Portal-A, which conceptually represents services, has a many-to-many relationship with the ProductOrServiceCategory table (through the CapabilityCategories intersection table). This means that a single service can be classified into multiple service categories. On the other hand, a service in Portal-B, which is represented by the ServiceDetails table, can be classified into only one service category. That is, the ServiceDetails table has a many-to-one relationship with the ServiceAndConsultantsCategory table.

**Figure 4 should be inserted about here**

**Figure 5 should be inserted about here**

## 3.2 Portals' Data

In this subsection, we illustrate sample data of Portal-A and Portal-B that will be used with the semantic mediation in section 4. Figure 6 and Figure 7 show the service category tables in Portal-A and Portal-B, respectively. These tables are relational representations of the taxonomies such as those shown in Figure 1. The Parent_ID and ParentID columns in the two tables reflect the subcategory relationship. Since these are foreign key relationships, they have no intrinsic subsumption logic between the parent and child categories within the model. It is up to the associated applications to implement these relationships; and we have observed in the use case analysis that such subsumption logic associated with the subcategory relationships are not carried in existing supplier portals, resulting in semantic ambiguities.

Notice from Figure 6 and Figure 7 and the database schemas in Figure 4 and Figure 5 that Portal-A does not have manufacturing process capability concepts such as the EDMCapability table (Figure 5) and MachiningCapability table (Figure 21) in Portal-B and that Portal-A relies solely on the more fine-grained service categories to convey the manufacturing process capability. For example, Figure 6 shows six subcategories (e.g., Ram EDM, Small Hole EDM) under the EDM category and four subcategories under the Dies & Molds category in Portal-A model as opposed to the Portal-B model in which neither the EDM Machining Service category nor the Mold Making Services category have any subcategory.

**Figure 6 should be inserted about here**

**Figure 7 should be inserted about here**

Figure 8 illustrates the list of materials used by Portal-A and Portal-B. Suppliers can use these values to describe the material capability for a particular service. Portal-A has a long list of materials that are more specific than that of Portal-B. For example, Portal-A has several types of stainless steel such as 15-4 Stainless Steel, 17-4 Stainless Steel, while Portal-B allows only the generic Stainless Steel. Both portals use terms that classify materials from multiple viewpoints, and hence, they are semantically overlapping or subsuming one another. For example, Portal-A provides the terms Ferrous, and Non-Ferrous, A-2 Tool Steel, A-6 Tool Steel, and Aluminum without any relationship. However, tool Steels such as A-2 Tool Steel and A-6 Tool Steel are Ferrous materials while Aluminum is a kind of Non-Ferrous material. In Portal-B, Titanium, Glass, Glass Ceramics are kinds of Ultra-hard Materials. Therefore, a supplier indicating that it can machine Aluminum will not be identified when a customer communicates a request to suppliers who can machine some Non-Ferrous materials. We will demonstrate that semantic mediation can be used for enhancing these semantic relationships and improving the MSC communication.

**Figure 8 should be inserted about here**

Figure 9 and Figure 10 illustrate example MSC descriptions from Portal-A and Portal-B, respectively. For ease of understanding of the data and semantic contrasts, these figures simplify the data structures into a form-like representation that is common in both portals, except for some names. Notice how the part size capability is represented differently in the two portals, and once again, the use of the single category and the process capability attribute in Portal-B vs. the use of extensive service categories in Portal-A.

**Figure 9 should be inserted about here**

**Figure 10 should be inserted about here**

## 3.3 Target Semantic Mediation Behaviors

In this subsection, we define target behaviors for the semantic mediation based upon semantic issues, database schemas, and data described in the previous subsections. The semantic mediation deployment analysis in the next section shall demonstrate these target behaviors.

Query requirements and their associated queries, which indicate services to be retrieved both within or across portals, are used as the specification of these target behaviors. A successful retrieval of expected data using their corresponding queries demonstrates that the semantic mediation has successfully occurred. Since each query *uses the structures and terminology only from a single portal and without adding any other portal-specific terms* to retrieve services across another portal that uses different structures and terminology, they show that the reconciliation between the differences is successful.

The first three columns in Table 1 describe the query requirements; and, the last two columns provide pointers to the corresponding executable queries. The executable expressions of queries will be shown in the deployment analysis section. The Expected Results column lists the individual services from Figure 9 and Figure 10 that the queries shall retrieve.

**Table 1 should be inserted about here**

We require that each query uses terms either only from Portal-A or only from Portal-B to return the same result for a given query requirement in order to conclude that bidirectional reconciliation between semantic issues occurs within the semantic mediation. Table 2 below summarizes types and resolutions of specific semantic issues captured by each query requirement.

**Table 2 should be inserted about here**

# 4 Semantic Mediation Deployment Analysis

This section describes the merged-model-based semantic mediation approach and its ability to enrich the proprietary MSC models described in the previous section. In particular, the deployment analysis subsection gives the detailed implementation of the approach. At the end, the subsection describing mediation results shows how the semantic mediation meets the target behaviors set forth in the previous section.

## 4.1 Merged-Model-Based Semantic Mediation Approach

The merged-model-based semantic mediation approach [Kulvatunyou et al. 2012] relies on the Web Ontology Language (OWL) and its underlying description logic (DL). In the first step of semantic mediation, proprietary MSC models are encoded in OWL. We will call an OWL encoded proprietary MSC model a *proprietary model*. The merged-model-based semantic mediation approach requires a *merged ontology* as an interlingua and source of semantics among participating proprietary models (for the purpose of semantic enrichment/disambiguation). The logical mappings, called *mapping ontologies*, between proprietary models and the merged ontology are then created. These linkages between proprietary models and the merged ontology form a mapping chain across all the proprietary models. After using OWL DL reasoner to perform inferences over the proprietary models, the merged ontology, and the mapping ontologies, reconciliation between terminological and structural differences as well as semantic enrichments occur across all the proprietary models. Definition 1 gives a formal definition for the merged-model-based semantic mediation.

> <u>*Definition 1:*</u> Merged-model-based approach is a 3-tuple $\chi = \{\Gamma, \Sigma, M'\}$ where
>
> - $\Gamma$ is a set of proprietary models participating in the semantic mediation, given by $\Gamma = \bigcup \gamma_i$ where $\gamma_i$ is a proprietary model, $i \in I, I = \{1,2,3,...,n\}$, and $n$ is the number of participating proprietary models;
> - $\Sigma$ is a merged ontology where $\Sigma \sqsupseteq_c \Gamma$, and $\sqsupseteq_c$ is conceptual (logical) superset . $\Sigma \sqsupseteq_c \Gamma$ means that every statement entailed in $\Gamma$ can be entailed by $\Sigma$. The concepts in $\Gamma$ may be modeled differently in $\Sigma$, however;
> - $M'$ is a set of mapping ontologies between each $\gamma_i$ and $\Sigma$, and given by $M' = \{\mu_{(i,\Sigma)}\}$, where
>   - $\mu_{(i,\Sigma)}$ is a set of two-way DL mapping statements between $\gamma_i$ and $\Sigma$,
>   - $|M'| = n$

## 4.2 Deployment Analysis

In this subsection, we show that the target semantic mediation behaviors of the exemplary proprietary manufacturing service models outlined in Section 3 are met by applying the merged-model-based semantic approach. We go through the steps and describe artifacts in the application. The first step is to encode the two relational proprietary MSC models into the proprietary models, $\gamma_1$ and $\gamma_2$, using OWL DL. This step is described first. Then, we describe the merged ontology, $\Sigma$, following with descriptions of the mapping ontologies $\mu_{(1,\Sigma)}$ and $\mu_{(2,\Sigma)}$. Finally, we discuss how the semantic mediation enables the target behaviors described in section 3.3.

### 4.2.1 Proprietary Models: Alternative OWL DL Encoding Conventions

The proprietary MSC models may be represented in a variety of syntaxes such as relational databases (RDBs), XML Schema and XML (Extensible Markup Language), RDF (Resource Description Framework) [W3C 2004a], or already in OWL. In our case, they are represented in RDBs. The first step in our semantic mediation approach is to encode these MSC models into OWL DL. We have found that there are multiple ways to encode artifacts from RDB into OWL DL. Below we outline three conventions.

#### 4.2.1.1 RDB-to-RDF Syntactical Encoding Convention

RDB-to-RDF syntactical encoding convention is a fully automated syntactical translation from RDB to RDF/OWL via the default RDB-to-RDF mapping profile. This default RDB-to-RDF mapping profile is provided within the D2RQ transformation framework and tools [D2RQ 2012]; however, that mapping profile uses only RDF vocabulary. The mapping profile can be customized to use OWL vocabulary in this convention such that tables are encoded as classes; non-foreign-keyed columns are encoded as datatype properties; foreign-keyed columns are encoded as object properties; rows are encoded as class instances; and column values are encoded as either datatype or object property values of the associated class instance. The issues with this encoding convention are that it is a pure syntactical translation where the OWL logical semantics are not utilized[4]; hence, increase in semantic precision is not accomplished. Figure 11 illustrates this encoding in the Manchester OWL Syntax [W3C 2009b]. It is a snippet of the Portal-A's `Supplier`, `Capability`, `ProductOrServiceCategory`, and `CapabilityCategories` tables shown in Figure 4 and the `Wire_EDM` service capability description shown in Figure 9 (the material capability is omitted at this point)[5]. Specifically to this model, the first issue is that the `db:Parent_ID` property would be better represented using the OWL's subclass axiom. By relegating a true generalization relationship to an object property, one introduces potentially significant loss in semantic precision, leading to semantic ambiguity, such as the cases discussed in section 3. The other issue is that the encoding does not clearly represent the purpose of the `CapabilityCategories` table which is to assign categories (`ProductOrServiceCategory` table) to a capability (`Capability` table). In other words, the `CapabilityCategories` table would be better encoded as an object property (of the `db:Capability`). By introducing inappropriate semantic constructs, one introduces more complicated semantic structures, leading to complexities when assuring mapping, reasoning, and desired query behaviors (i.e., information retrieval). This will be evident when we illustrate that the other two encoding conventions avoid these complicated semantic structures (e.g., meaningless instances).

**Figure 11 should be inserted about here**

#### 4.2.1.2 Object-oriented-modeling Encoding Convention

With the help of domain experts to tailor the D2RQ mapping profile for a specific relational schema, object-oriented-modeling encoding convention enhances semantics of the previous convention with object-oriented semantics (including the subclass and object associations). The typical results are that appropriate rows are represented as classes; parent-child relationships between rows are represented as subclasses; and intersection tables become object properties.

Figure 12[6] illustrates the object-oriented-modeling encoding convention of the same set of information as in Figure 11. A few key differences are 1) the `db:WireEDM`, `db:SinkerEDM` and `db:EDM` are now classes;

---

[4] In other words, only *assertion axioms* and *declaration axioms* are utilized in this transformation. Other types of axioms particularly *class expression axioms* and *property expression axioms* are not utilized. See W3C (2009c) for definitions of types of axioms.

[5] We have simplified the actual output from the D2RQ for readability and ease of understanding.

[6] In this and subsequent encoding illustration we left out for brevity purpose some columns that are not related to service information such as the `isConcrete` column in the `ProductOrServiceCategory` table. The purpose of

2) the first two classes are subclasses of the last as the db:Parent_ID property has morphed into the OWL subclass axiom; and 3) because the CapabilityCategories is an intersection table, the class db:CapabilityCategories and its object properties have morphed into a single db:hasProductOrServiceCategory object property. Note that for ease of understanding in this figure, we have changed the db:Supplier_ID property name to db:hasSupplier and truncated some properties such as db:ID and db:Category_Name that are irrelevant to the semantic mediation. It can be seen that this convention improves the semantic precision upon the previous one as the subcategory relationship is carried as the formal OWL subclass semantics. The change from the class db:CapabilityCategories and its object properties into the db:hasProductOrServiceCategory object property also brings about clearer and concise semantics of the relationship between the db:Capability and db:ProductOrServiceCategory classes.

**Figure 12 should be inserted about here**

#### *4.2.1.3 Ontology-oriented Encoding Convention*

Ontology-oriented encoding convention is guided by the general principle that all concepts have a corresponding class and uses of instances are minimized. The rationale for this principle is that 1) there are more mapping and semantic enhancement facilities for classes in OWL DL than for properties and instances; and 2) if every concept has a corresponding class then there is no need to map between differing types of entities. The principle necessitates that characteristics (of manufacturing service capabilities) are described by class and object/data property axioms as opposed to simply properties and values. To facilitate the discussion of this convention and to complete the encoding of the Wire_EDM service capability description in Figure 9, we add the material capability information to the snippet in Figure 12. This addition requires the encoding of the Material and CapabilityMaterials tables (Figure 4). Figure 13A and Figure 13B illustrate two alternative ways to encode these pieces of information in the object-oriented-modeling encoding convention. Records in the Material table represent types of materials such as alloy steel and carbon steel (Figure 8A). In Figure 13A, these are represented as instances (individuals) of the class db:Material. Then the material capability can be expressed as associations between a db:Capability instance and a db:Material instance using the db:hasMaterial object property such as this RDF triple (db:Wire_EDM, db:hasMaterial, db:AlloySteel). Alternatively, types of materials are represented as classes in Figure 13B. The material capability is then expressed as an associations between a db:Capability instance and an instance of the db:AlloySteel (or other material subclasses) such as the RDF triple (db:CNC, db:hasMaterial, db:AlloySteel_1). In this approach, the db:AlloySteel_1 seems extraneous as it has no real semantic meaning[7]. However, it is more convenient to provide additional formal semantics (including mapping semantics) to material concepts when they are modeled as classes. For example, statements like “alloy steel is a subclass of steel” would be possible. This is not the case for the approach in Figure 13A where specific types of materials are modeled as instances.

---

the isConcrete column is to support a business rule indicating which category can be used to categorize the capability. It has nothing to do with the service information.

[7] The instances db:AlloySteel_1 has no real semantic meaning in the sense that there is no difference if other instances such like db:AlloySteel_2 and db:AlloySteel_3 were created and used.

Figure 13 should be inserted about here

The model combined from snippets in Figure 12 and Figure 13B almost follows the ontology-oriented encoding convention. There are two issues left. First, the db:Max_Length property is a concept in and of itself (expressing the maximum part size capability), but it has no corresponding class while every other concept does. The other issue is that the model still uses the meaningless instances including db:AlloySteel_1, db:RamEDM_1, and db:WireEDM_1. Figure 14 illustrates the information in Figure 12 and Figure 13B that is fully encoded with the ontology-oriented encoding convention. The changes are discussed below.

First, the db:MaxPartLength class (and related object and datatype properties, db:hasMaxPartLength, db:hasMinLengthExclusive, db:hasMaxLengthInclusive) replaces the db:Max_Length datatype property. This addresses the first issue that the db:Max_Length datatype property deserves to be a class because it is a concept. Another way to view this is that the concept can still be broken down into two datatype properties, db:hasMinLengthExclusive, db:hasMaxLengthInclusive[8,9]. In addition, being a class gives access to more mapping functionalities in OWL DL than being a property. Because of reasons such as these, even if the original MSC models are already represented in OWL DL, it can still be beneficial for the purpose of semantic mediation in OWL DL to convert them to follow the ontology-oriented encoding convention.

The other change is that the usages of db:AlloySteel_1, db:RamEDM_1, and db:Wire_EDM_1 instances are replaced with class declaration axioms. This change eliminates the meaningless instances and addresses the second issue. Such uses of meaningless instances can cause confusions in the model, reduction in semantic precision, and more expensive mapping. For example, if db:AlloySteel_2 or others were created as instances of the db:AlloySteel class in addition to db:AlloySteel_1, confusion may occur because these instances are not different in this case. Semantic precision would also be reduced if they were not mapped or stated to be the same. Mapping statement would be needed for each additional instance created leading to more expensive mapping. The material capability and service category association are also modeled with the class declaration axioms using the rdfs:type predicate. The db:Wire_EDM instance is now declared with three types. First, it is a member of the db:Capability class. Second, it is a member of the db:SinkerEDM class. This is to say that it provides some sinker EDM services. Classification of the db:Wire_EDM instance into multiple service categories can be achieved by stating that it is a member of intersections of specific db:ProductOrServiceCategory classes, e.g., (db:SinkerEDM and db:WireEDM). Lastly, it is a member of an anonymous class, which necessarily has the db:hasMaterialCapability property whose value is a member of the db:AlloySteel class. This is to say that it can machine the alloy steel. Additional material capabilities can be expressed by stating that it is a member of intersections of other similar anonymous classes, e.g.,

[8] The mediation relating to data unit conflicts is outside the scope of this paper. The approach to deal with this within the realm of manufacturing sourcing using the manufacturing service models is deferred to future research. Interested readers are referred to existing works by Bijan and Smith (2008) and Hodgson and Keller (2011) that discuss the handling of units in OWL.

[9] The min value is defaulted to zero here. Portal-A engineer may default it to other values such as a part size known to be the smallest that can be handled by EDM machines in the market.

`(db:hasMaterialCapability some db:AlloySteel) and (db:hasMaterialCapability some db:CarbonSteel).`

**Figure 14 should be inserted about here**

Appendix B provides more complete OWL DL ontology-oriented encoding convention of Portal-A and Portal-B MSC models. The models contain only necessary definitions to capture the MSC descriptions delineated in Figure 9 and Figure 10. They are sufficient for verifying the target behaviors of these models after the semantic mediation discussed in subsequent sections.

### 4.2.2 Merged Ontology

The merged ontology serves as an interlingua to participating proprietary models. It is also a knowledge-base and source of semantics from which proprietary models inherit additional semantics. Such semantics enrich the semantics of proprietary models and also facilitate information accessibility across information sources. There are efforts to create such a reference model. In particular, Ameri and Dutta (2006) have defined an OWL-based manufacturing service ontology using the manufacturing-process-oriented view; and Jang et al. (2008) have defined an OWL-based manufacturing service ontology using the machining-feature-oriented view. Alternatively, manufacturing service capability can also be defined using the resource-oriented view such as that defined by Vichare et al. (2009). Defining a reference manufacturing service model, which necessarily covers broad manufacturing domain beyond machining, is beyond the scope of this paper. The authors are working with the Open Application Group standard consortium to begin such work within the Advancing Computer Interpretable Communication of Manufacturing Information work group. In addition to the aforementioned work, other existing research works and standards need to be taken into consideration in developing such reference model such as ISO 14649 (STEP-NC) which has standardized machining features [ISO 14649-1:2002], ISO 15331 which includes a standard for representing machining resources [ISO 15331-1:2003], ISO 13399 which includes a standard for representing cutting tool information [ISO 13399-1:2006], ASME B5.59-2 which is a standard for describing the performance and capabilities of milling and turning machines [ASME B5.59-2], and Ameri and Summers (2008) which provides an ontology for representation of fixture design knowledge.

For the purpose of this demonstration, the merged ontology currently used in this work reuses and extends parts of the Manufacturing Service Description Language (MSDL) [Ameri and Dutta 2006]. Figure 15 provides a high-level conceptual view of the merged ontology using the UML class diagram notation. `Process` and `Service` have several subclasses. These and other additional details are shown in Figure 16 that illustrates parts of the merged ontology used. Notice that the merged ontology not only has well-defined subclass hierarchy, it also contains defined class axioms such as the `mo:EDMService` class. The axiom has a class expression, `EquivalentTo: mo:hasProcess some mo:ElectroDischargeMachining`, establishing the semantic link between the notions of service categories and processes.

The `Material` concept in Figure 15 also has several subclasses and defined axioms which are major enhancements to MSDL. Parts of this prototyped material ontology are illustrated in Figure 17 and Figure 18. The material ontology facilitates the mapping and enhances the semantics of the disconnected list of materials in the proprietary models. Appendix C contains further details of the material ontology relevant to this semantic mediation. Figure 16 and Appendix C describe the merged ontology needed to verify the target semantic mediation behaviors outlined in section 3.3.

**Figure 15should be inserted about here**

**Figure 16 should be inserted about here**

Figure 17A shows the hierarchy of the material types in the material ontology with the focus on mo:Steel (which is the part that will be used in the semantic mediation)[10]. Figure 18A, B, C, and D show definitions of mo:Steel, mo:StainlessSteel, mo:MartensiticPrecipitationHardeningSS (SS stands for Stainless Steel), and mo:AusteniticSS, respectively. These classes are defined by the percentage ranges of element masses and by other characteristics such as the corrosion and hardening properties. The mo:AusteniticSS, mo:MartenisticSS, and mo:MartensiticPrecipitationHardeningSS are another way to categorize stainless steels in addition to by the series (e.g., the mo:StainlessSteel_300Series and mo:StainlessSteel_400Series class in Figure 17A). These categories initially have no subclasses. Figure 17B shows that specific stainless steels and series can be automatically classified under these categories by running an automated classifier within the Pellet OWL DL reasoner [Clark and Parsia 2012].

**Figure 17 should be inserted about here**

**Figure 18 should be inserted about here**

### 4.2.3 Mapping Ontologies

Two mapping ontologies, $\mu_{(1, \Sigma)}$ and $\mu_{(2, \Sigma)}$, are needed for this semantic mediation. Each mapping ontology is developed by domain expert owners of each proprietary model. $\mu_{(1, \Sigma)}$ is a mapping ontology from the Portal-A's proprietary model to the merged ontology; and $\mu_{(2, \Sigma)}$ is a mapping ontology from the Portal-B's proprietary model to the merged ontology. It should be noted that terminologies from Portal-A, Portal-B, and the merged ontology will be distinguished by the prefixes pa, pb, and mo, respectively.

Table 3 illustrates mapping axioms in $\mu_{(1, \Sigma)}$ (complete Portal-A proprietary model is in Figure 24 of Appendix B). For brevity, mappings for high-level classes (e.g., pa:Material) are not included because we do not need them to meet the target behaviors. The axioms are either equivalent property or inverse property axioms when the source (Portal-A term) is a property. When the source is a class, equivalent class axioms are used. In the table, both equivalent property and equivalent class axioms are denoted by the value E in the Axiom Type column, while the inverse property axiom is denoted by the value I. We make three noted observations about mapping axioms in this table as described below.

The first observation is that the axiom A2 states that the pa:hasSuppplier and mo:hasActor properties are equivalent. The mapping may look inaccurate in that the pa:hasSupplier should be just a subproperty of the mo:hasActor; however, this is valid in that it produces desirable mediation results.

[10] It should be noted that material hierarchy and definitions illustrated in this paper have not been reviewed by experts and hence should not be used as reference. The purpose of these figures is to illustrate the semantic mediation and that material ontology modeled in OWL DL can be useful.

The reason is two-fold: 1) in the related class mapping in axiom A15, it accurately stated that the class equivalence holds between the pa:Supplier and mo:Supplier and 2) in this use case we are not mapping any other class using the mo:hasActor property (e.g., pa:Customer and mo:Customer). In a more general situation, we can take advantage of mapping specialized object properties onto a more abstract object property and preserve the original semantics of the correspondingly mapped classes, as long as the ranges or domains of the corresponding specialized object properties are disjoint within the semantic mediation (or application) requirement context. (We will reference this general approach as 'abstract object property mapping' and use it throughout our discussion of property mapping in the rest of this section.) In addition, the subproperty mapping (supported by an OWL object subproperty axiom) would produce a poorer mediation result that does not meet the target behaviors. For example, let's assume that another portal has the same property as the pa:hasSupplier, say px:hasSupplier, which is also mapped as a subproperty of the mo:hasActor. In such a situation, Portal-A's instances, which have the pa:hasSupplier property, will not be inferred to have the property px:hasSupplier and vice versa. In other words, the translation between the Portal-A terms and the other portal will not occur[11].

The second observation is related to the part length capability mapping, axioms A3, A4, A5, and A16. The justification for the axiom A3 is the same as that in the first observation, based on the abstract object property mapping. The axioms A4 and A5 are straightforward data property mappings. The axiom A16 states the equivalence between the pa:MaxPartLength and mo:PartLengthCapability classes. Even though Portal-A only allows the maximum value of the part size/length capability to vary, the fundamental semantics and logical structure are the same as that of mo:PartLengthCapability. Therefore, it is reasonable to express the equivalence between the two classes. In addition, if the pa:MaxPartLength were expressed as the mo:PartLengthCapability class whose minimum length value is always fixed (to zero or another arbitrary value), then it would only be a subclass of the mo:PartLengthCapability. This would not yield the desirable target behaviors, because part length capability instances of Portal-B or that of other portals would not be translatable to the Portal-A's term.

**Table 3 should be inserted about here**

The third and last observation is related to the service category mapping in axiom A6 to A13. The merged ontology does not have comprehensive *categories of services*; however, it has comprehensive *categories of processes*. Therefore, the finer-grained service categories in Portal-A are mapped to the generic service class (mo:Service) with varying process capabilities in the merged ontology. Because the processes in the merged ontology do have subclass relationships, these semantics are transferred to the Portal-A service categories. Figure 19 shows that the pa:MicroHoleEDM service category, originally a sibling category, becomes a subcategory of the pa:SmallHoleEDM category after automated reasoning was completed over these mappings. This is because the mo:MicroHoleEDM process is a subclass of the mo:SmallHoleEDM process in the merged ontology (notice that the same terms refer to services in Portal-A vs. processes in the merged ontology). Effectively, the mapping axioms A6 to A13 enhance the

[11] Other alternatives may exist such as adjusting the merged ontology. We believe that there is a need for further research on guidance to design, usage, and mapping of OWL object and data properties. We are continuing to look into these issues in our research.

semantic precision of Portal-A service categories by inheriting the semantics from the process capabilities in the merged ontology. Next we discuss the Portal-B mapping ontology.

Figure 19 should be inserted about here

Table 4 illustrates mapping axioms in $\mu_{(2,\Sigma)}$ (complete Portal-B proprietary model is in Figure 25 and Figure 26 of Appendix B). Information in this mapping table should be read the same way as described earlier in Table 3. Only mappings of concepts necessary for the target behaviors are illustrated. Noted observations to mapping axioms in this table are as follows.

First, the properties `pb:hasMSProcessCapability` and `pb:hasEDMProcessCapability` are mapped to the same property, `mo:hasProcess`, in axioms `B1` and `B2`. The abstract object property mapping rationale, as in the first observation to Table 3, applies here, too. The respective classes representing the concepts provide precise semantic mapping including those in the mapping axioms `B13` to `B19`. Mapping the two properties as subproperties of `mo:hasProcess` would similarly result in degraded mediation results that do not meet the target behaviors.

Portal-B relies more on processes than service categories to convey its manufacturing capability similar to the merged ontology. Therefore, `B13` to `B19` are simple class equivalence mapping axioms between process concepts, while `B20` and `B21` are service categories to processes mappings. These mapping axioms enhance the semantics of the Portal-B model. For example, because the `pb:SmallHoleEDM` process is mapped to `mo:SmallHoleEDM` process and because the `mo:MicroHoleEDM` process is a subclass of the `mo:SmallHoleEDM` process, the reasoner can infer that `mo:MicroHoleEDM` is also a subclass of the `pb:SmallHoleEDM`. This will, for example, enable Portal-A services classified under the `pa:MicroHoleEDM` ***service category*** to be retrieved when using the `pb:SmallHoleEDM` ***process***.

Table 4 should be inserted about here

Table 5 shows mapping axioms that are also part of the Portal-B's mapping ontology. These equivalent class axioms enhance the Portal-B's model semantics related to the part length capability. Portal-B's part length capabilities are represented as a collection of value ranges. These mapping axioms connect the value ranges in the collection where possible and also translate them into the merged ontology terms. They use the notion of the *mapping class* [Kulvatunyou et al. 2012]. It is an OWL defined class with multiple class definitions (i.e., multiple `owl:equivalentClass` axioms). A mapping class is a virtual concept to provide links to those definitions. Each definition references terms from a single portal (notice that there are two rows for each mapping class in Table 5); therefore, translations occur between definitions. Note that for brevity, partial part length capability mapping classes are shown. Similarly defined mapping classes are additionally required for complete semantic enhancements including `PLCLessThan7.5`, `PLC2.5To15`, and `PLC2.5To120`.

Table 5 should be inserted about here

### 4.2.4 Mediation Results

The semantic mediation target behaviors identified in Table 1 of section 3.3 are verified in this section. First, we represent the queries `Q1.1`, `Q1.2`, `Q2.1`, and `Q2.2` in the OWL DL Query syntax [W3C 2009b].

The four query statements are shown below. Recall that `Q1.1` and `Q2.1` use only Portal-A's terms, while `Q1.2` and `Q2.2` use only Portal-B's terms. We discuss the semantic mediation results in two deployment scenarios, 1) single-community deployment and 2) multi-community deployment.

```
Q1.1: pa:Capability and pa:SinkerEDM and (pa:hasMaxPartLength some (pa:MaxPartLength
and ((pa:hasMinLengthExclusive value 0.0) and (pa:hasMaxLengthInclusive some
xsd:double[>= 15.0])))) and (pa:hasMaterialCapability some
pa:PreHardenedStainlessSteel)

Q1.2: pb:EDMMachiningServiceDetails and (pb:hasEDMProcessCapability some
pb:ElectrodeEDM) and (pb:hasPartLengthCapability some (pb:PartLengthCapability and
((pb:minPartLengthExclusive value 0.0) and (pb:maxPartLengthInclusive some
xsd:double[>= 15.0]))) and (pb:hasMaterialCapability some (pb:HardenedMetals and
pb:StainlessSteel)))

Q2.1: pa:Capability and pa:SmallHoleEDM and (pa:hasMaxPartLength some
(pa:MaxPartLength and ((pa:hasMinLengthExclusive some xsd:double[<= 3.0]) and
(pa:hasMaxLengthInclusive some xsd:double[>= 8.0]))))

Q2.2: pb:EDMMachiningServiceDetails and (pb:hasEDMProcessCapability some
pb:SmallHoleEDM) and (pb:hasPartLengthCapability some (pb:PartLengthCapability and
((pb:minPartLengthExclusive some xsd:double[<= 3.0]) and (pb:maxPartLengthInclusive
some xsd:double[>= 8.0]))))
```

#### 4.2.4.1 *Single-Community Deployment*

In the single-community deployment, only a single proprietary model is involved in the semantic mediation. The single community deployment can be, for example, an internal deployment within a large enterprise or a single information sharing portal (e.g., marketplace). The objective of this deployment is to enable the community sharing the same proprietary model to have access to more precise MSC information. To deploy the semantic mediation in this scenario, an OWL DL reasoner is executed over a proprietary model, the merged ontology, and the mapping ontology. This allows for the proprietary model to exploit the richer and more precise semantics of the merged ontology.

To verify the target behaviors on Portal-A's model, we execute `Q1.1` and `Q2.1` over Portal-A's proprietary model with and without semantic mediation. In the case of execution with semantic mediation, we first execute the Pellet OWL DL Reasoner on Portal-A's proprietary model (Appendix B), merged ontology (Figure 16 and Appendix C), and Portal-A's mapping ontology (Table 3). Executing `Q1.1` without semantic mediation returns both `pa:WireEDM` and `pa:CNC` services as expected; however, `Q2.1` does not return `pa:CNC` as expected. It is because within the Portal-A's proprietary model `pa:CNC` is not classified into the `pa:SmallHoleEDM` service category, a condition in `Q2.1`. This is in spite of its classification into another semantically more specific service category, `pa:MicroHoleEDM`. Because there is no logical relationship explicitly stated between `pa:MicroHoleEDM` and `pa:SmallHoleEDM`, the service cannot be classified as `pa:SmallHoleEDM`.

With the semantic mediation, `Q2.1` returns `pa:CNC`. This is because the `pa:CNC` service is originally classified as `pa:MicroHoleEDM` and the reasoner infers that `pa:MicroHoleEDM` is a subclass of the `pa:SmallHoleEDM` as discussed in section 4.2.3 and Figure 19. It follows that `pa:CNC` is also classified as (i.e., a member of) `pa:SmallHoleEDM`. The relationship between the `pa:MicroHoleEDM` and `pa:SmallHoleEDM` service categories occurs because of the mapping axioms `A12` and `A13` that link the two <u>*service categories*</u> to the merged ontology's `mo:SmallHoleEDM` and `mo:MicroHoleEDM` <u>*process*</u>

*capabilities*. Because mo:MicroHoleEDM is a subclass of the mo:SmallHoleEDM, the reasoner infers that pa:MicroHoleEDM is a subclass of the pa:SmallHoleEDM.

#### 4.2.4.2 Multi-community Deployment

In the multi-community deployment, two or more proprietary models are involved in the semantic mediation. The multi-community deployment can be, for example, a deployment across multiple enterprises or a deployment across information portals (marketplaces). The objective of this deployment is to enable communities using differing proprietary models to have interoperable communications of the MSC information across the models/portals in addition to having more precise accesses to the MSC information. To deploy the semantic mediation in this scenario, an OWL DL reasoner is executed over participating proprietary models, the merged ontology, and the mapping ontology between each participating proprietary model and the merged ontology. This allows for the proprietary models to exploit the merged ontology as an Interlingua in addition to its richer and more precise semantics.

To verify the target behaviors for the multi-community deployment, we execute all four queries on the deployment and verify that expected services from one portal are returned using queries of another portal. In such deployment scenario, the Pellet OWL DL reasoner is executed over Portal-A's and Portal-B's proprietary models (Appendix B), the merged ontology (Figure 16 and Appendix C), and the mapping ontologies (Table 3 to Table 5). Each of the four queries successfully retrieves expected services across the two proprietary models (see Table 1). Because Q1.1 and Q2.1 use only Portal-A's terminologies to retrieve also Portal-B's services and conversely Q1.2 and Q2.2 use only Portal-B's terminologies to retrieve also Portal-A's services, it is evident that reconciliation across the terminology sets has successfully occurred.

Next we take Q1.1 as an example and describe how the semantic mediation occurs such that it retrieves all the four services. First, the axioms A14 and B9 establish a mapping chain such that the pa:Capability and pb:ServiceDetails become equivalent classes. The result is that all individuals (i.e., class instances) of pa:Capability and pb:ServiceDetails are members of both classes. The first condition (the token pa:Capability) in the Q1.1 then retrieves all four service instances in both proprietary models.

The second condition (the token pa:SinkerEDM) in the Q1.1 calls for services in the pa:SinkerEDM service category. The pa:Wire_EDM service satisfies this condition because it is directly classified into the pa:SinkerEDM category. The pa:CNC service satisfies this condition because of the semantic enrichment by the mapping axioms A9 and A11 making the pa:SinkerEDM category equivalent to the pa:RamEDM category into which pa:CNC classified. The pb:S_4EDMService and pb:S_9MSService satisfy this condition because of the process to service category reconciliation between the pb:ElectrodeEDM and pa:SinkerEDM by the mapping axioms A9 and B16.

The third condition in Q1.1 (the part length expression token) calls for services that can accept parts with length between 0 and 15 cm. In this case, services that have their part length capabilities covering the value range (0, 15] cm. satisfy this condition. The pa:Wire_EDM and pa:CNC services satisfy this because it declares the part length capability value range (0, 30] cm and (0, 88], respectively. The pb:S_4EDMService satisfies this condition with the help of the mapping class PLCLessThan15 shown in Table 5 that connects all of its value range selections, (0, 2.5], (2.5, 7.5], (7.5, 15], into a single value range of (0, 15] cm exactly matching the condition. The mapping class PLCLessThan120 similarly enables the pb:S_9MSService to satisfy this condition. Without such semantic precision enhancements by the mapping classes, the two Portal-B's service would not be returned when using Portal-A's terminologies.

Finally, the last condition calls for services that can machine some pa:PreHardenedStainlessSteel. The pa:Wire_EDM service satisfies this condition with the exact match, while the pa:CNC service satisfies this condition because it expresses pa:StainlessSteel17-4 capability which is mapped to be equivalent to the mo:StainlessSteel_17-4PH. The mo:StainlessSteel_17-4PH is in turn inferred to be a subclass of the pa:PreHardenedStainlessSteel. This subclass relation is inferred primarily because mo:StainlessSteel_17-4PH class definition (Appendix C) and the pa:PreHardenedStainlessSteel mapping axiom A19 both contain the expression mo:StainlessSteel and (mo:isHardenedByHeatTreatment value true). Figure 20 shows the related inferred class hierarchy. The pb:S_4EDMService and pb:S_9MSService services satisfy this condition because they express their material capabilities as an intersection of anonymous classes having material capabilities pb:StainlessSteel, pb:HardenedMetals, and also (pb:StainlessSteel and pb:HardenedMetals), the result of which is a subclass of an anonymous class having material capability pa:PreHardenedStainlessSteel. The first two capabilities do not make the two Portal-B's services satisfying the condition, because neither of them is a subclass of the pa:PreHardenedStainlessSteel. However, the last capability does. This is because after replacing the two Portal-B's terms with their mapping axioms B24 and B25 in the class expression (pb:StainlessSteel and pb:HardenedMetals), we obtain (mo:StainlessSteel and mo:Metals and mo:isHardenedByHeatTreatment value true). Since mo:StainlessSteel and mo:Metals = mo:StainlessSteel, the expression can be reduced to (mo:StainlessSteel and mo:isHardenedByHeatTreatment value true), which means pa:PreHardenedStainlessSteel according to its A19 mapping axiom.

**Figure 20 should be inserted about here**

# 5 Related Work

Today's popular approaches to semantic mediation rely on procedural transformation languages such as XSL transformation language (XSLT) [W3C 1999] or XML Query Language (XQuery) [W3C 2010]. Other semantic mediation approaches exist. For example, Park and Ram (2004) suggested agent-based framework that uses its own formalism to model ontologies and mappings. Papakonstaninou et al. (1996) and Genesereth et al. (1997) suggested rule-based approach for information integration. The former introduced its own languages including the Object Exchange Message (OEM) and Mediator Specification Language (MSL), while the latter used Knowledge Interchange Format (KIF). Sciorer (1994) uses a contextualized data interchange scheme that gives complete semantics to the data being exchanged to enable communication between database systems. These works focus primarily on integration of relational databases with the aim to produce another relational view, enable federated queries using a global view, or exchange information between database systems. They lack the semantic enrichment of each data source shown in this paper. In addition, they do not rely on internet-based standards such like OWL and RDF (Resource Description Framework [W3C 2004a]) and consequently do not lend themselves to internet-based information integration. While KIF is a first order logic language, which is more expressive than OWL DL, it lacks built-in semantics that OWL has (e.g., subclass, subproperty, and their associated logical transitivity) making it less attractive to achieve semantic enrichment.

The merged-model-based semantic mediation approach requires three artifacts including the proprietary model, merged ontology, and mapping ontology. In rest of this section, we discuss works that are related to the creation of these artifacts.

We have shown that there are various ways to encode information in OWL DL and that it is more beneficial to semantic mediation as the encoding moves toward the ontology-oriented encoding convention. Works in Ontology Design Pattern (ODP) can be useful for development of detailed, practical ontology-oriented encoding convention. ODP is a modeling solution to solve a recurrent ontology design problem [Gangemi and Presutti 2009]. The W3C Semantic Web Best Practices and Deployment Working Group has published ODPs for OWL that are independent of concept [W3C 2005, W3C 2006]. For example, an ODP given by the Working Group is “Defining N-ary Relations on the Semantic Web.” Gangemi (2005) has also proposed another type of ODPs called conceptual ODPs and associated template for describing a conceptual ODP. A conceptual ODP is conceptual in a sense that it is a fragment extracted from a foundational or core ontology that provides the ODP’s background semantics. Conceptual ODPs proposed in Gangemi (2005) are however independent of a formal encoding. An ODPs repository [ODPs Repository 2012] has been established under the European NeOn research project [NeOn 2012]. Methodologies to create and use ODPs have also been proposed within the NeOn project [Presutti et al. 2008, Daga et al. 2010]. Investigation into how these methodologies and proposed ODPs may apply to the MSC information and its application should be performed.

When the information source is large, manually encoding the information in OWL DL is impractical. Technologies exist to automatically encode information from relational databases (RDBs) into OWL DL. Satya et al. (2005) have presented a survey on existing tools to transform information in RDB into RDF. They have found a number of tools supporting such a task including D2RQ, Oracle Database 11g, Virtuoso’s RDF View, Metatomix Semantic Platform, RDBtoOnto, SquirrelRDF, TopBraid Composor and Triplify. Based on Hert et al. (2011), which have presented an overview and classification of the RDB-to-RDF mapping languages proposed up to 2011, we have observed that it is not possible to encode fully in the ontology-oriented encoding convention with these technologies. One of the biggest issues is that these technologies do not support creating multiple classes from a single column based on the values of a related attribute. Therefore, the material concepts in section 4.2.1, for example, cannot be encoded as classes because they are originally records in a single table. A two-stage transformation can be used to mitigate such an issue. That is, information originally captured in an RDB (or other syntaxes) is first transformed into a common RDF graph syntax with a simple syntactical encoding convention, and then an ontology transformation is applied in the second stage to follow an ontology-oriented encoding convention. In the second stage, a pattern-based ontology transformation approach, which goes hand-in-hand with the usages of content ODPs, can be used. Such approach uses a graph and string-based ontology pattern definition to match the ontology artifacts in the source, which is then modified with another graph and string-based ontology pattern definition from the target ODP. Svab-Zamazal et al. (2011) has provided a promising methodology for pattern-based ontology transformation which enables modification to the whole or parts of an RDF/OWL ontology. They provide a well-defined XML schema to represent ontology pattern definitions and pattern transformation rules. A graphical user interface-based editor and a transformation engine have also been provided to create and execute these definitions and rules.

Creation, architecture, development, and management of the merged ontology are non-trivial tasks. A merged ontology may be created by unifying existing models. Hence, methodologies to create unified database views such as Navathe et al. (1986) and Hayne and Ram (1990) are relevant to the creation of the merged ontology. Ontology engineering processes and methodologies apply to any ontology development activity including the merged ontology development. Jones et al. (1998) have summarized the ontology engineering activities and identified the need for guidance on ontology reuse. Staab et al. (2001) have presented guidance for building ontologies either from scratch, reusing other ontologies as they are, or re-engineering them. Pinto et al. (2004) has suggested a distributed ontology engineering process. Merged ontology development should also rely on ODPs. As mentioned earlier, the NeOn project has delivered an initial and significant report on ontology development using ODPs [Daga et al. 2010]. In practice, merged ontology evolves over time. Noy and Klein (2004) have characterized the causes of evolution of ontologies, including changes in the domain, changes in conceptualization, and changes in the explicit specification. In addition, a merged ontology may be changed due to changes in participating proprietary models. Works related to ontology change management is well summarized in Flouris et al. (2008).

Mapping is one of the most difficult tasks in semantic mediation. It is also related to development and maintenance of the merged ontology in terms of gap analysis. Early works to automate or semi-automate ontology merging and mapping includes PROMPT [Noy and Musen 2003] and Chimaera [McGuiness et al. 2000]. Our evaluation of PROMPT has found that the system does not perform well when encountering with structural conflicts. Other research works in the past decade on algorithms to suggest mappings have been summarized in Shvaiko and Euzenat (2011). The authors have indicated that one of the open issues is to identify correspondences across classes and properties, i.e., when dealing with structural conflicts. We believe that the mapping task could be simplified if encoding of the proprietary MSC model follows the same ODPs used to develop the merged ontology, because the structural conflicts and the need to map between different types of entities would be minimized. Therefore, we believe that further research in ODPs is important for a productive deployment of the merged-model-based semantic mediation approach as it is related to all the three necessary artifacts.

# 6 Conclusion

Semantic precision and interoperability of proprietary manufacturing service capability (MSC) models can be enhanced with little to no change to them. This paper presents a deployment analysis of the merged-model-based semantic mediation approach that is based on a formal description logic model to achieve such enhancement. The approach requires three artifacts and starts with the encoding of the original/proprietary MSC model into the proprietary model represented in the OWL DL. In this step, we have introduced three encoding conventions. The first convention is the RDB-to-RDF syntactical encoding convention. This encoding convention can be fully automated. However, this OWL encoding is purely syntactical such that little OWL logical semantics are utilized; hence, increase in information precision is not accomplished. The second convention is the object-oriented-modeling encoding convention. This convention uses domain experts to identify object-oriented style semantics including subclass and object association and specify them in the RDB-to-OWL mapping; and it relies on class instances to capture MSC descriptions. The third convention is the ontology-oriented encoding convention. This convention calls for all concepts to be represented as OWL classes and eliminate the use of meaningless class instances. The transition from the first, to the second, and the third convention, shows that as OWL logical semantics are increasingly utilized, OWL mapping functionality is more accessible, and mapping is simplified. The two other artifacts required for the approach are the merged ontology and mapping ontologies. Observations associated with these artifacts are discussed. Finally, we illustrate that the deployment of these artifacts to enhance precision and access to MSC information can be done in two scenarios, namely single- and multi-community deployment. The single-community deployment enables precise and higher access to MSC information within the community; while the multi-community deployment enables precise and interoperable communications of MSC information across the communities.

## *DISCLAIMER*

Certain commercial software products are identified in this paper. These products were used only for demonstration purposes. This use does not imply approval or endorsement by NIST, nor does it imply these products are necessarily the best available for the purpose.

# Appendix A: Portal-B Database Schemas

**Figure 21 should be inserted about here**

**Figure 22 should be inserted about here**

**Figure 23 should be inserted about here**

# Appendix B: Proprietary Models

**Figure 24 should be inserted about here**

Portal-B's proprietary model is decomposed into two figures for readability. Figure 25 contains class and property definitions; and Figure 26 contains instance data.

**Figure 25 should be inserted about here**

**Figure 26 should be inserted about here**

## Appendix C: Additional Merged Ontology Artifacts

```
ObjectProperty: mo:hasHardness
ObjectProperty:
  mo:hasCorrosionResistance
ObjectProperty:
  mo:hasWearResistance
ObjectProperty:
  mo:hasMaterialContent
DataProperty:
  mo:hasCrystallineStructure
DataProperty:
  mo:hasContentPercentage
DataProperty:
  mo:isHardenedByHeatTreatment

Class: mo:Scale
Class: mo:Low SubClassOf: mo:Scale
Class: mo:Medium
  SubClassOf: mo:Scale
Class: mo:High SubClassOf: mo:Scale

Class: mo:MaterialContent
Class: mo:C_Content
  SubClassOf: mo:MaterialContent
Class: mo:Fe_Content
  SubClassOf: mo:MaterialContent
Class: mo:Ni_Content
  SubClassOf: mo:MaterialContent
Class: mo:Mn_Content
  SubClassOf: mo:MaterialContent
Class: mo:Cu_Content
  SubClassOf: mo:MaterialContent
Class: mo:Si_Content
  SubClassOf: mo:MaterialContent
Class: mo:P_Content
  SubClassOf: mo:MaterialContent
Class: mo:Ta_Content
  SubClassOf: mo:MaterialContent
Class: mo:Cr_Content
  SubClassOf: mo:MaterialContent
Class: mo:S_Content
  SubClassOf: mo:MaterialContent
Class: mo:Nb_Content
  SubClassOf: mo:MaterialContent

Class: mo:Metal
   SubClassOf: mo:Material
Class: mo:Titanium
   SubClassOf: mo:Metal
Class: mo:Ferrous
   SubClassOf: mo:Metal
```

```
Class: mo:StainlessSteel_17-4PH
  EquivalentTo:
    mo:StainlessSteel
    and (mo:hasCorrosionResistance some
    mo:High)
    and (mo:hasMaterialContent some
    (mo:C_Content and
    (mo:hasContentPercentage some
    xsd:double[<= 0.07])))
    and (mo:hasMaterialContent some
    (mo:Cr_Content and
    (mo:hasContentPercentage some
    xsd:double[>= 15.0 , <= 17.5])))
    and (mo:hasMaterialContent some
    (mo:Cu_Content and
    (mo:hasContentPercentage some
    xsd:double[>= 3.0 , <= 5.0])))
    and (mo:hasMaterialContent some
    (mo:Mn_Content and
    (mo:hasContentPercentage some
    xsd:double[<= 1.0])))
    and (mo:hasMaterialContent some
    (mo:Nb_Content and
    (mo:hasContentPercentage some
    xsd:double[>= 0.15 , <= 0.45])))
    and (mo:hasMaterialContent some
    (mo:Ni_Content and
    (mo:hasContentPercentage some
    xsd:double[>= 3.0 , <= 5.0])))
    and (mo:hasMaterialContent some
    (mo:P_Content and
    (mo:hasContentPercentage some
    xsd:double[<= 0.04])))
    and (mo:hasMaterialContent some
    (mo:S_Content and
    (mo:hasContentPercentage some
    xsd:double[<= 0.03])))
    and (mo:hasMaterialContent some
    (mo:Si_Content and
    (mo:hasContentPercentage some
    xsd:double[<= 1.0])))
    and (mo:hasMaterialContent some
    (mo:Ta_Content and
    (mo:hasContentPercentage some
    xsd:double[>= 0.15 , <= 0.45])))
    and (mo:hasCrystallineStructure
    value "Martenistic")
    and (mo:isHardenedByHeatTreatment
    value true)
  SubClassOf:
    mo:StainlessSteel
```

*(continue next page)*

```
Class: mo:Steel
  SubClassOf: mo:Ferrous
  EquivalentTo:
    (mo:hasMaterialContent some
    mo:Fe_Content)
    and (mo:hasMaterialContent some
    (mo:C_Content and
    (mo:hasContentPercentage some
    xsd:double[>= 0.2 , <= 2.1])))

Class: mo:AlloySteel
  SubClassOf: mo:Steel

Class: mo:ToolSteel
  SubClassOf: mo:AlloySteel
  EquivalentTo:
    mo:AlloySteel
    and (mo:hasCorrosionResistance
    some mo:High)
    and (mo:hasHardness
    some mo:High)
    and (mo:hasWearResistance
    some mo:High)

Class: mo:StainlessSteel
  SubClassOf: mo:AlloySteel
  EquivalentTo:
    mo:AlloySteel
    and (mo:hasCorrosionResistance
some
    mo:High)
    and (mo:hasMaterialContent some
    (mo:Cr_Content and
    (mo:hasContentPercentage some
    xsd:double[>= 10.5 , <= 28.0])))
```

```
Class:
  mo:MartensiticPrecipitationHardeningSS
  SubClassOf: mo:StainlessSteel
  EquivalentTo:
    mo:StainlessSteel and
    (mo:hasCorrosionResistance some
    mo:High)
    and (mo:hasMaterialContent some
    (mo:Cr_Content and
    (mo:hasContentPercentage some
    xsd:double[>= 14.0 , <= 17.5])))
    and (mo:hasMaterialContent some
    (mo:Cu_Content and
    (mo:hasContentPercentage some
    xsd:double[>= 2.5 , <= 5.0])))
    and (mo:hasMaterialContent some
    (mo:Nb_Content and
    (mo:hasContentPercentage some
    xsd:double[>= 0.15 , <= 0.45])))
    and (mo:hasMaterialContent some
    (mo:Ni_Content and
    (mo:hasContentPercentage some
    xsd:double[>= 3.0 , <= 5.5])))
    and (mo:hasMaterialContent some
    (mo:Ta_Content and
    (mo:hasContentPercentage some
    xsd:double[>= 0.15 , <= 0.45])))
    and (mo:hasCrystallineStructure
    value "Martenistic")
    and (mo:isHardenedByHeatTreatment
    value true)
```

## References


Ameri F. and Dutta D. (2006), An Upper Ontology for Manufacturing Service Description, ASME Conference Proceeding 2006, 651 (2006), DOI:10.1115/DETC2006-99600.

Ameri, F. and Summers, J.D. (2008), An ontology for representation of fixture design knowledge, *Computer Aided Design and Applications*, 5 (2008), pp. 601–611.

ASME B5.59-2. Information technology for machine tools - part 2: Data specification for properties of machine tools for milling and turning (draft), 2005.

Bijan, P., Smith M. (2008), Quantities in OWL, *OWLED 2008 OWL: Experiences and Directions*.

Clark and Parsia LLC (2012), Pellet: OWL 2 Reasoner for Java version 2.2.2, available online at http://clarkparsia.com/pellet/, accessed September 2012.

D2RQ framework version 0.8.1 (2012), Accessing Relational Databases as Virtual RDF Graphs, available online at http://d2rq.org/, accessed September 2012.

Daga, E., Blomqvist, E., Gangemi, A., Montiel, E., Nikitina, N., Presutti, V., and Villazón-Terrazas, B. (2010), *NeOn Project Delivery D2.5.2 - Pattern Based Ontology Design: Methodology and Software Support*.

Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., and Antoniou, G. (2008), *The Knowledge Engineering Review Journal*, Vol. 23(2), pp. 117-152.

Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., and Antoniou, G. (2008), Ontology Change: Classification and Survey (2008), *The Knowledge Engineering Review Journal*, Vol. 23(2), pp. 117-152.

Gangemi, A. (2005), Ontology Design Patterns for Semantic Web Content, *International Semantic Web Conference*, 2005, pp. 262–276.

Gangemi, A. and Presutti, V. (2009), Ontology Design Patterns, Handbook on Ontologies, 2nd Ed. *International Handbooks on Information Systems*, Springer, 2009.

Genesereth, M.R., Keller, A. M., and Duschka, O. M. (1997), Infomaster: An information integration System, *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data* (Tucson, AZ, May 13–15), J. M. Peckman Ed. ACM Press, New York, NY, 39–542.

Hayne, S., and Ram, S. (1990), Multi-user view integration system (MUVIS): An expert system for view integration, *Proceedings of the 6th International Conference on Data Engineering* (Los Angeles, CA, Feb. 5–9), pp. 402–409.

Hert, M., Reif, G., and Gall, H.C. (2011), A Comparison of RDB-to-RDF Mapping Languages, Proceedings of the International Conference on Semantic Systems, 2011.

Hodgson, R., Keller, P. (2011), QUDT – Quantities, Units, Dimensions and Data Types in OWL and XML, available online at http://www.qudt.org/.

ISO 13399-1:2006, Cutting tool data representation and exchange — part 1: Overview, fundamental principles and general information model.

ISO 14649-1:2002, Industrial automation systems and integration — physical device control — data model for computerized numerical controllers, part 1: Overview and fundamental principles.

ISO 15531-1:2003, Industrial automation systems and integration – industrial manufacturing management data – part 1: General overview.

Jang, J., Jeong, B., Kulvatunyou, B.S., Chang, J., and Cho, H.(2008), Discovering and Integrating Distributed Manufacturing Services with Semantic Manufacturing Capability Profiles, *International Journal of Computer-Integrated Manufacturing*, 21(6), pp 631-646, 2008.

Jameson, E.C. (2001), Electrical Discharge Machining, Society of Manufacturing Engineering (SME).

Jones, D., Bench-Capon, T., and Visser, P. (1998), Methodologies for Ontology Development, *Proceedings of the IT&KNOWS Conference of the 15th IFIP World Computer Congress*, 1998.

Kulvatunyou, B.S., Ivezic, N., Lee, Y., Shin, J. (2012), Semantic Mediation Approaches to Enhance Manufacturing Service Models, submitted to *International Journal of Computer Integrated Manufacturing*, available online at http://www.nist.gov/publication-portal.cfm#.

Kuokka, D. and Harada , L. (1996), Integrating information via matchmaking, Journal of Intelligent Information Systems, Vol. 6(2-3), pp. 261-279.

McGuiness, D.L., Fikes, R., Rice, J., Wilder, S. (2000), The Chimaera Ontology Environment, *Proceedings of AAAI Conference, 2000*.

Navathe, S. B., Elmasri, R., and Larson, J. A. (1986), Integrating user views in database design, *IEEE Computers*, Vol. 19(1), pp. 50–62.

NeOn (2012), NeOn (Networked Ontologies) Project, available online at http://www.neon-project.org/nw/, accessed September 2012.

Noy, N.F., and Klein, M. (2004), Ontology Evolution: Not the same as Schema Evolution, Knowledge and Information Systems, Vol. 6(4): 428-440, July, 2004.

Noy, N.F. and Musen, M.A. (2003), The PROMPT suite: Interactive Tools for Ontology Merging and Mapping, *International Journal of Human-Computer Studies*, Vol. 59(6), December 2003, pp. 983-1024, ISSN 1071-5819, 10.1016/j.ijhcs.2003.08.002.

ODPs Repository (2012), Ontology Design Patterns Repository, available online at http://www.ontologydesignpatterns.org, accessed September 2012.

Papakonstantinou, Y., Garcia-molina, H., and Ullman, J. (1996), Medmaker: A mediation system based on declarative specifications, *Proceedings of the 12th IEEE International Conference on Data Engineering* (New Orleans, LA, Feb. 26–March 1), S. Y. W. Su, Ed. 132–141.

Park, J. and Ram, S. (2004), Information Systems Interoperability: What Lies Beneath?, *ACM Transactions on Information Systems*, Vol. 22(4), pp. 595-632.

Pinto, H.S., Tempich, C., Staab, S. (2004), DILIGENT: Towards a Fine-Grained Methodology for DIstributed, Loosely-Controlled and Evolving Engineering of Ontologies, Proceedings of ECAI 2004, pp. 393-397, López de Mantaras, R., Saitta, L., Eds, IOS Press, Amsterdam.

Presutti, V., Gangemi, A., David, S., Aguado de Cea, G., Suárez-Figueroa, M.C., Montiel-Ponsoda, E., Poveda, M. (2008), *NeOn Project Delivery - D2.5.1. A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies*.

Satya, S.S., Halb, W., Hellmann, S., Idehen, K., Thibodeau, T., Auer, S., Sequeda, J., and Ezzat, A. (2005), *A Survey of Current Approaches for Mapping of Relational Databases to RDF, W3C RDB2RDF Incubator Group*, available online at http://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF_SurveyReport.pdf

Sciore, E., Siegel, M., and Rosenthal, A. (1994), Using semantic values to facilitate interoperability among heterogeneous information systems, *ACM Transaction on Database System*, Vol. 19(2), pp. 254–290.

Sheth, A. P., and Kashyap, V. (1992), So far (schematically), yet so near (semantically), *Proceedings of the IFIP WG2.6 Database Semantics Conference on Interoperable Database Systems* (DS-5, Lorne, Victoria, Australia, Nov. 16–20), Hsiao, D. K., Neuhold, E. J., and Sacks-Davis, R., Eds., 283–312.

Shvaiko, P. and Euzenat, J. (2011), Ontology matching: state of the art and future challenges, *IEEE Transactions on Knowledge and Data Engineering*, 99.

Staab, S., Schnurr, H.P., Studer, R., Sure, Y. (2001), Knowledge Processes and Ontologies, *IEEE Intelligent Systems*, 16(1), 26–34, 2001.

Svab-Zamazal O., Svatek V., Scharffe F., David J. (2011), Detection and Transformation of Ontology Patterns, *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, Revised Selected Papers from IC3K 2009, Springer CCIS no.128, 2011, pp. 210–223.

Vichare, P., Nassehi, A., Kumar, S. and Newman, S. (2008), A Unified Manufacturing Resource Model for Representation of CNC Machining Systems, *18th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM)*, 2008-06-30 - 2008-07-02, Skövde.

W3C – World Wide Web Consortium (1999), XSL Transformation (XSLT), November 16, 1999, available online at http://www.w3.org/TR/xslt.

W3C - World Wide Web Consortium (2004a), Resource Description Framework (RDF): Concepts and Abstract Syntax, February 10, 2004, available online at http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

W3C - World Wide Web Consortium (2005), Representing Specified Values in OWL: "value partitions" and "value sets", May 17 2005, available online at http://www.w3.org/TR/swbp-specified-values/

W3C - World Wide Web Consortium (2006), Defining N-ary Relations on the Semantic Web, April 12, 2006, available online at http://www.w3.org/TR/swbp-n-aryRelations/.

W3C - World Wide Web Consortium (2009a), OWL 2 Web Ontology Language, October 27, 2009, available online at http://www.w3.org/TR/owl2-overview/.

W3C – World Wide Web Consortium (2009b), OWL 2 Web Ontology Language Manchester Syntax W3C Working Group Note, October 27, 2009, available online at http://www.w3.org/TR/owl2-manchester-syntax/.

W3C - World Wide Web Consortium (2009c), OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax, October 27, 2009, available online at http://www.w3.org/TR/owl2-syntax/.

W3C – World Wide Web Consortium (2010), XQuery 1.0: An XML Query Language (Second Edition), December 14, 2010, available online at http://www.w3.org/TR/xquery/.

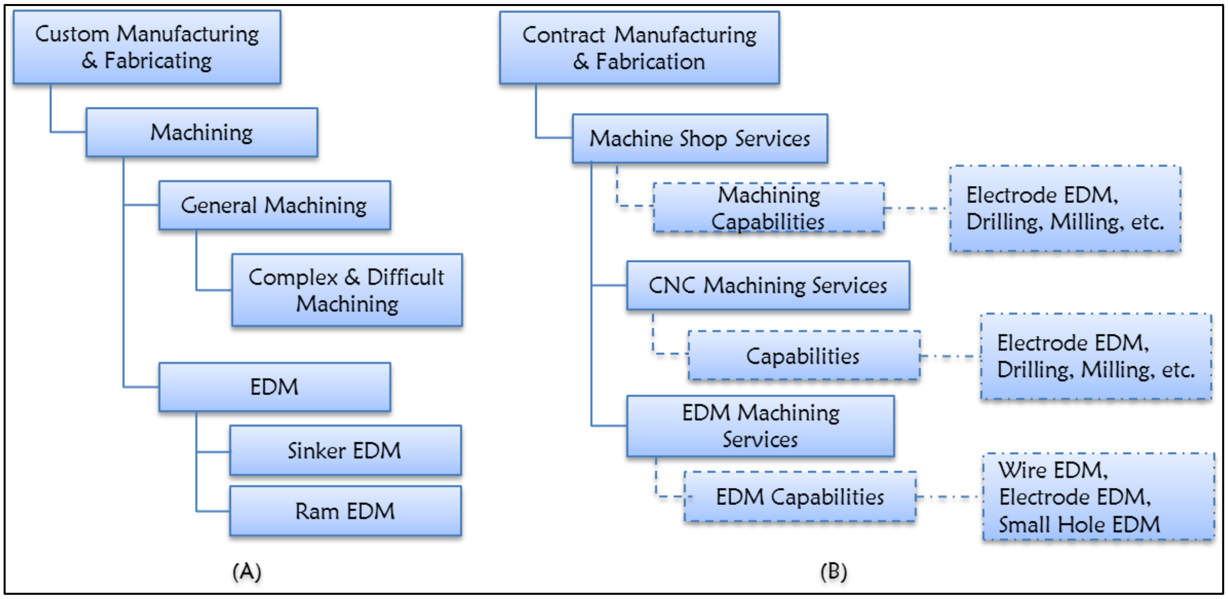



**Figure 1: Semantic ambiguities illustration - (A) Portal-A's model and (B) Portal-B's model (dashed boxes represent possible properties and dash-dot boxes represent possible property values)**

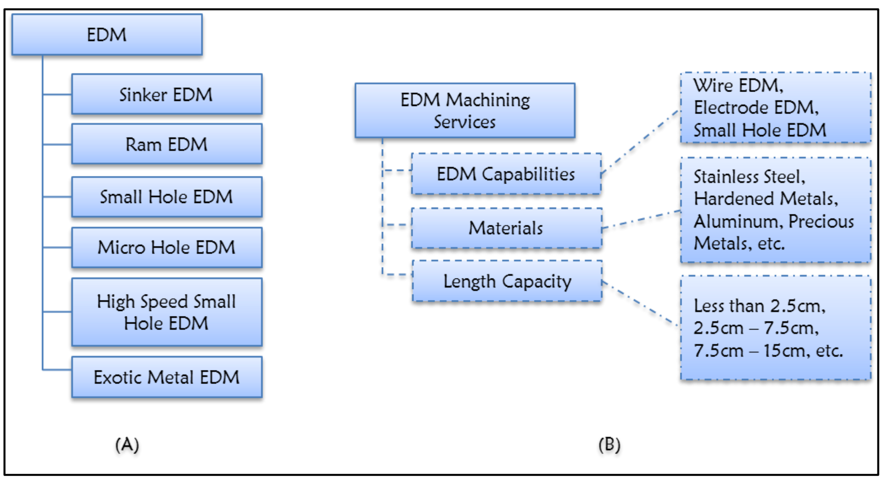



**Figure 2: Illustration of semantic modeling conflicts related to EDM – (A) Portal-A's model; (B) Portal-B's model**

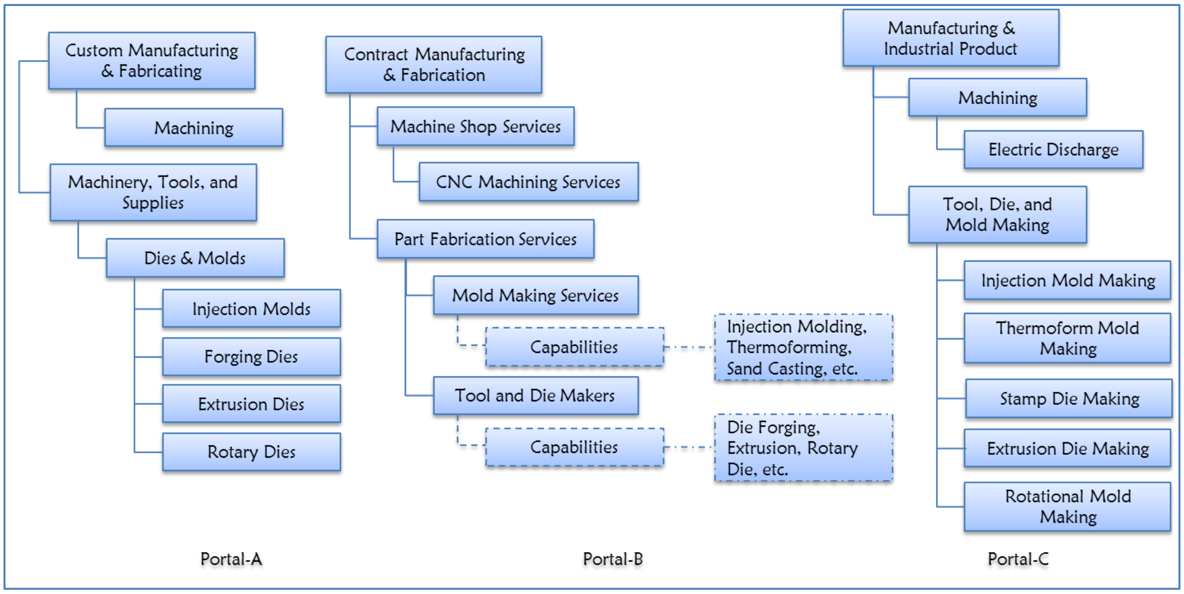



**Figure 3: A snippet of MSC models related to the dies and molding making capability to illustrate the generalization and naming conflicts across Portal A, B, and C**

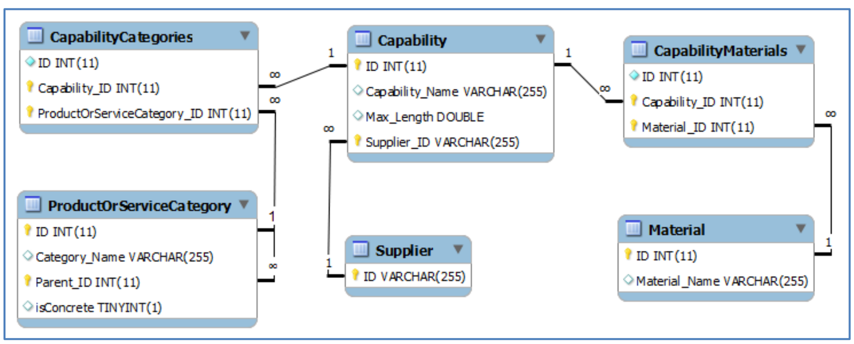



**Figure 4: Relational schema of Portal-A's MSC model**

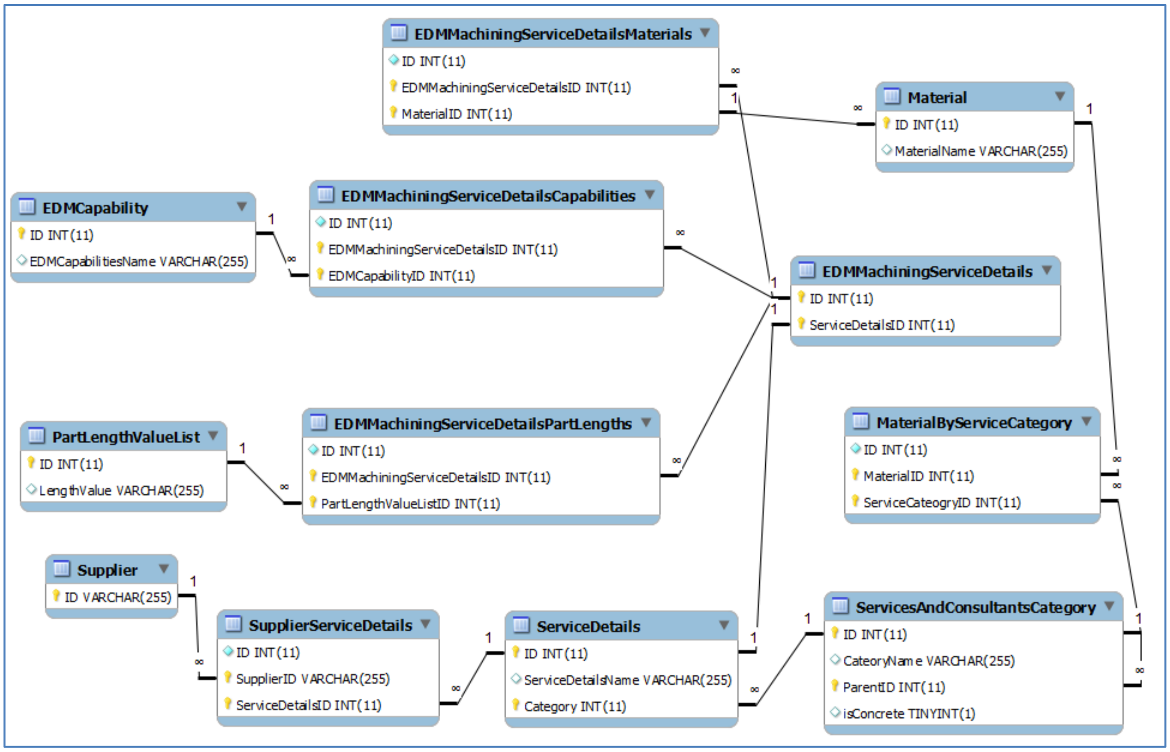



**Figure 5 Relational schema of Portal-B's MSC model related to the EDM machining service**

ProductOrServiceCategory

| ID | Category_Name | Parent_ID | isConcrete |
|---|---|---|---|
| 1 | Custom Manufacturing & Fabricating | | ☐ |
| 8 | Machinery Tools & Supplies | | ☐ |
| 2 | Machining | Custom Manufacturing & Fabricating | ☐ |
| 19 | Extrusion Dies | Dies & Molds | ☑ |
| 18 | Forging Dies | Dies & Molds | ☑ |
| 10 | Injection Molds | Dies & Molds | ☑ |
| 20 | Rotary Dies | Dies & Molds | ☐ |
| 17 | High Speed Small Hole EDM | EDM | ☑ |
| 16 | Micro Hole EDM | EDM | ☑ |
| 7 | Ram EDM | EDM | ☑ |
| 6 | Sinker EDM | EDM | ☑ |
| 15 | Small Hole EDM | EDM | ☑ |
| 14 | Wire EDM | EDM | ☑ |
| 4 | Complex & Difficult Machining | General machining | ☐ |
| 9 | Dies & Molds | Machinery Tools & Supplies | ☐ |
| 12 | CNC Machining | Machining | ☑ |
| 13 | EDM | Machining | ☑ |
| 3 | General machining | Machining | ☐ |

**Figure 6: Portal-A's `ProductOrServiceCategory` table data representing the taxonomy of service categories**

ServicesAndConsultantsCategory

| ID | CateoryName | ParentID | isConcrete |
|---|---|---|---|
| 1 | Contract Manufacturing And Fabrication | | ☐ |
| 6 | Machine Shop Services | Contract Manufacturing And Fabrication | ☑ |
| 2 | Part Fabrication Services | Contract Manufacturing And Fabrication | ☐ |
| 5 | EDM Machining Services | Machine Shop Services | ☑ |
| 3 | Mold Making Services | Part Fabrication Services | ☑ |

**Figure 7: Portal-B's `ServicesAndConsultantsCategory` table data representing the taxonomy of service categories**

| Material_Name | Material_Name |
|---|---|
| 15-4 Stainless Steel | Cobalt |
| 17-4 Stainless Steel | Columbium |
| 300 Stainless Steel | CR/HR Steel |
| 304 Stainless Steel | D2 Tool Steel |
| 316 Stainless Steel | Extrusions |
| 4130 Tool Steel | Ferrous |
| 4140 Alloy Steels | Forgings |
| A-2 Tool Steel | Fully Hardened Tool Steel |
| A-6 Tool Steel | H-13 Tool Steel |
| Alloy Steel | M2 Tool Steel |
| Aluminum | Non-Ferrous |
| Brass | Others |
| Carbon Steel | P-20 |
| Castings | Plastics |
| Chrome | Pre-hardened Stainless Steel |

(A)

| Material_Name |
|---|
| Aluminum |
| Brass |
| Stainless Steel |
| Steel / Steel Alloys |
| Titanium |
| Hardened Metals |
| Ultra-hard Materials |
| Glass |
| Glass Ceramics |
| Ceramics |

(B)

**Figure 8: Material table data representing the list of allowable specification of the material capability – (A) Portal-A's data; (B) Portal-B's data**

Supplier Capability Name: Wire_EDM
- Supplier: Supplier_5
- Service Category: Sinker EDM, Wire EDM
- Material Capability: Pre-hardened Stainless Steel, Alloy Steel, Ferrous, etc.
- Max Part Length: 30 cm

Supplier Capability Name: CNC
- Supplier: Supplier_1
- Service Category: CNC Machining, Ram EDM, Wire EDM, Micro Hole EDM
- Material Capability: Stainless Steel 17-4, Tool Steel, Titanium, etc.
- Max Part Length: 88 cm

**Figure 9: Example MSC descriptions of Portal-A**

Supplier Service Details Name: S_4EDMService
- Supplier: Supplier_4
- Service Category: EDM Machining Service
- Process Capability: Electrode EDM, Wire EDM, Small Hole EDM
- Material Capability: Stainless Steel, Hardened Metals, etc.
- Part Length: Less than 2.5 cm, 2.5 cm – 7.5 cm, 7.5 cm – 15 cm

Supplier Service Details Name: S_9MSService
- Supplier: Supplier_9
- Service Category: Machine Shop Service
- Process Capability: Milling, Drilling, Electrode EDM
- Material Capability: Stainless Steel, Hardened Metals, etc.
- Part Length: Less than 2.5 cm, 2.5 cm – 7.5 cm, 7.5 cm – 15 cm, 15 cm – 120 cm

**Figure 10: Example MSC descriptions of Portal-B**

```
Class: db:ProductOrServiceCategory
Class: db:Supplier
Class: db:Capability
Class: db:CapabilityCategories
DataProperty: db:ID
DataProperty: db:isConcrete
DataProperty: db:Category_Name
DataProperty: db:Max_Length
ObjectProperty: db:Parent_ID
ObjectProperty: db:Supplier_ID
ObjectProperty: db:Capability_ID
ObjectProperty:
   db:ProductOrServiceCategory_ID

Individual: db:Supplier_5
  Types: db:Supplier
  Facts:
    db:ID  "Supplier_5"

Individual: db:Wire_EDM
  Types: db:Capability
  Facts:
    db:ID  7,
    db:Supplier_ID  db:Supplier_5
    db:Max_Length  30.0

Individual: db:EDM
  Types: db:ProductOrServiceCategory
  Facts:
    db:ID  "13"^^xsd:int,
    db:Parent_ID  Machining,
    db:Category_Name  "EDM"
    db:isConcrete  true
```

```
Individual: db:SinkerEDM
  Types: db:ProductOrServiceCategory
  Facts:
    db:ID  "6"^^xsd:int,
    db:Parent_ID  EDM,
    db:Category_Name  "Sinker EDM",
    db:isConcrete  true

Individual: db:WireEDM
  Types: db:ProductOrServiceCategory
  Facts:
    db:ID  "14"^^xsd:int,
    db:Parent_ID  EDM,
    db:Category_Name  "Wire EDM",
    db:isConcrete  true

Individual: db:CapabilityCategories_11
  Types: db:CapabilityCategories
  Facts:
    db:Capability_ID db:Wire_EDM,
    db:ProductOrServiceCategory_ID
      SinkerEDM

Individual: db:CapabilityCategories_12
  Types: db:CapabilityCategories
  Facts:
    db:Capability_ID db:Wire_EDM,
    db:ProductOrServiceCategory_ID
       WireEDM
```

**Figure 11: OWL snippet illustrating the RDB-to-RDF syntactical encoding convention of Portal-A's data**

```
ObjectProperty: db:hasSupplier
ObjectProperty: db:hasProductOrServiceCategory
DataProperty: db:Max_Length
Class: db:ProductOrServiceCategory
Class: db:EDM
  SubClassOf: db:Machining
Class: db:SinkerEDM
  SubClassOf: db:EDM
Class: db:WireEDM
  SubClassOf: db:EDM
Class: db:Supplier
Class: db:Capability
Individual: db:Supplier_5
  Types: db:Supplier
Individual: db:SinkerEDM_1
  Types: db:SinkerEDM
Individual: db:WireEDM_1
  Types: db:WireEDM
Individual: db:Wire_EDM
  Types: db:Capability
  Facts:
   db:hasProductOrServiceCategory  db:SinkerEDM_1,
   db:hasProductOrServiceCategory  db:WireEDM_1,
   db:Max_Length  30.0,
   db:hasSupplier  db:Supplier_5
```

**Figure 12: OWL snippet illustrating the object-oriented-modeling encoding convention**

```
ObjectProperty: db:hasMaterial
Class: db:Material

Individual: db:AlloySteel
  Types: db:Capability

Individual: db:Wire_EDM
  Types: db:Capability
    Facts:
      db:hasProductOrServiceCategory
         db:SinkerEDM_1,
      db:hasProductOrServiceCategory
         db:WireEDM_1,
      db:Max_Length  30.0,
      db:hasSupplier  db:Supplier_5,
      db:hasMaterial  db:AlloySteel
```

(A)

```
ObjectProperty: db:hasMaterial
Class: db:Material
Class: db:AlloySteel
  SubClassOf: db:Material

Individual: db:AlloySteel_1
  Types: db:AlloySteel
Individual: db:Wire_EDM
    Types: db:Capability
    Facts:
     db:hasProductOrServiceCategory
        db:SinkerEDM_1,
     db:hasProductOrServiceCategory
        db:WireEDM_1,
     db:Max_Length  30.0,
     db:hasSupplier  db:Supplier_5,
     db:hasMaterial  db:AlloySteel_1
```

(B)

**Figure 13: OWL snippets illustrating two alternatives in object-oriented-modeling encoding convention – (A) instance-based and (B) class-based**

```
ObjectProperty: db:hasSupplier
ObjectProperty:
   db:hasMaterialCapability
ObjectProperty: db:hasMaxPartLength

DataProperty: db:hasMaxLengthInclusive
DataProperty: db:hasMinLengthExclusive

Class: db:ProductOrServiceCategory
Class: db:EDM
  SubClassOf: db:Machining
Class: db:SinkerEDM SubClassOf: db:EDM
Class: db:RamEDM SubClassOf: db:EDM

Class: db:Material
Class: db:AlloySteel
  SubClassOf: db:Material

Class: db:MaxPartLength
Class: db:Supplier
```

```
Class: db:Capability

Individual: db:Supplier_5
  Types: db:Supplier

Individual: db:Wire_EDM
  Types:
    db:Capability,
    db:SinkerEDM,
    db:hasMaterialCapability some
       db:AlloySteel
  Facts:
    db:hasMaxPartLength
         db:Wire_EDMMaxPartLength
    db:hasSupplier db:Supplier_5

Individual: db:Wire_EDMMaxPartLength
   Types: db:MaxPartLength
   Facts:
      db:hasMinLengthExclusive  0.0,
      db:hasMaxLengthInclusive  30.0
```

Figure 14: OWL snippet illustrating the ontology-oriented encoding convention of Portal-A

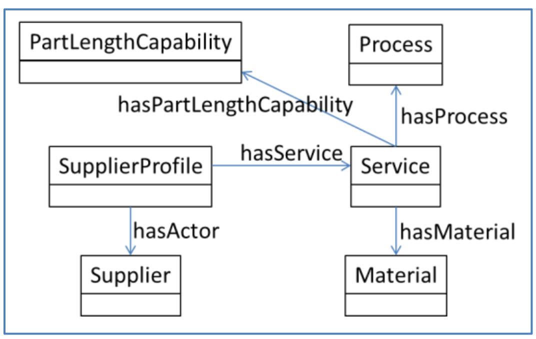
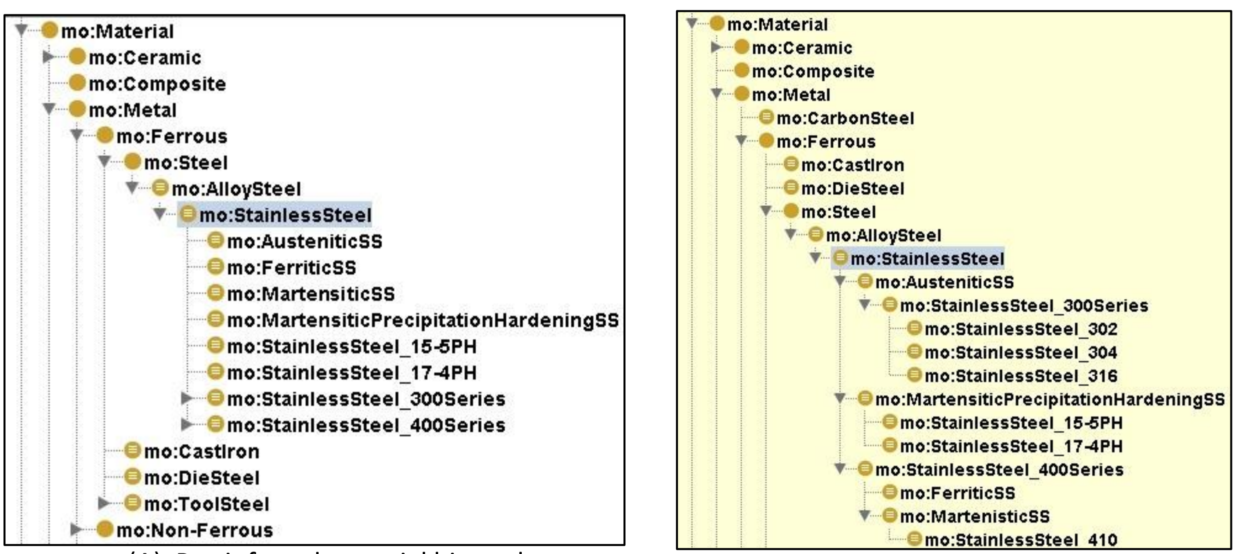



Figure 15: UML Conceptual model of the merged ontology

```
Class: mo:Service
Class: mo:MfgService
   SubClassOf:
      mo:Service
Class: mo:MachiningService
   SubClassOf:
      mo:MfgService
Class: mo:NonTraditionalMachiningService
   SubClassOf:
      mo:MachiningService
Class: mo:EDMService
   SubClassOf:
      mo:NonTraditionalMachiningService
   EquivalentTo: mo:hasProcess some
      mo:ElectroDischargeMachining

Class: mo:Actor
Class: mo:Supplier
   SubClassOf: mo:Actor
Class: mo:Customer
   SubClassOf: mo:Actor
Class: mo:PartLengthCapability
Class: mo:SupplierProfile

ObjectProperty: mo:hasActor
ObjectProperty: mo:hasMaterial
ObjectProperty: mo:hasProcess
ObjectProperty: mo:hasService
ObjectProperty:
   mo:hasPartLengthCapability
```

```
Class: mo:Process
Class: mo:MfgProcess
   SubClassOf: mo:Process
Class: mo:Machining
   SubClassOf: mo:MfgProcess
Class: mo:NonTraditionalMachining
   SubClassOf: mo:Machining
Class: mo:ElectroDischargeMachining
   SubClassOf:
      mo:NonTraditionalMachining
Class: mo:SinkerEDM
   SubClassOf:
      mo:ElectroDischargeMachining
Class: mo:WireEDM
   SubClassOf:
      mo:ElectroDischargeMachining
Class: mo:SmallHoleEDM
   SubClassOf:
      mo:ElectroDischargeMachining
Class: mo:MicroHoleEDM
   SubClassOf:
      mo:SmallHoleEDM

Class: mo:Material

DataProperty:
   mo:hasValueRangeMaxInclusive
DataProperty:
   mo:hasValueRangeMinExclusive
```

Figure 16: Parts of the merged ontology used in semantic mediation




(A) Pre-inferred material hierarchy

(B) Inferred material hierarchy

Figure 17: Examples of material class hierarchy


**Figure 18: Examples of definitions in materials ontology**

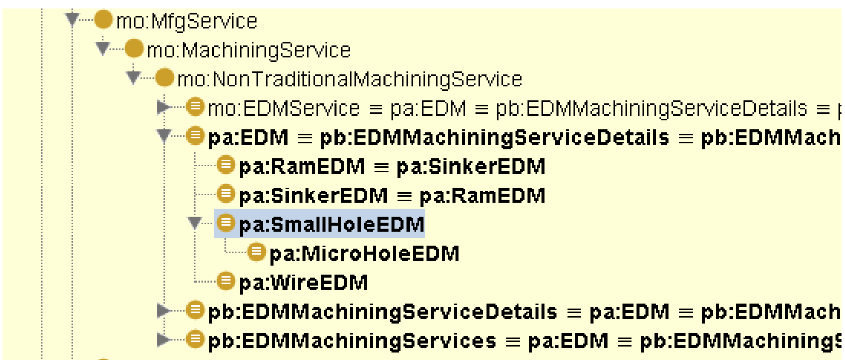



**Figure 19: Portal-A's hierarchy of service categories after the semantic mediation illustrating its semantic precision enhancement (before semantic mediation `pa:MicroHoleEDM` is a sibling of the `pa:SmallHoleEDM`)**

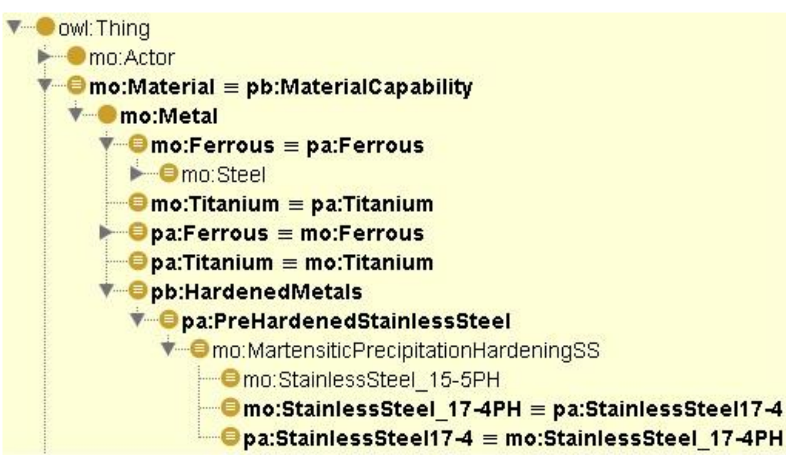



**Figure 20: Inferred material capability class hierarchy after taking into account mapping ontologies**

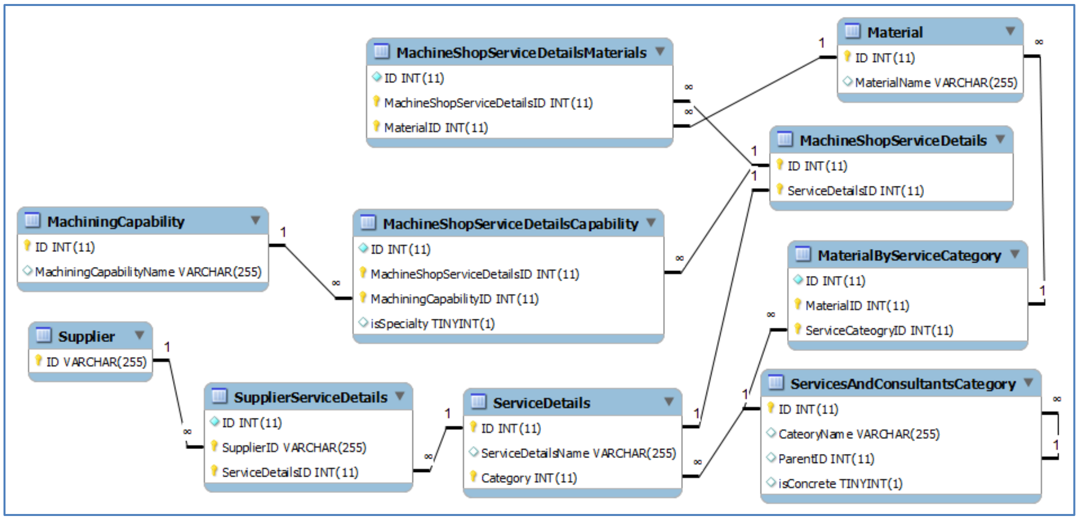



Figure 21: Portal-B's relational schema of its manufacturing service model related to the machine shop service

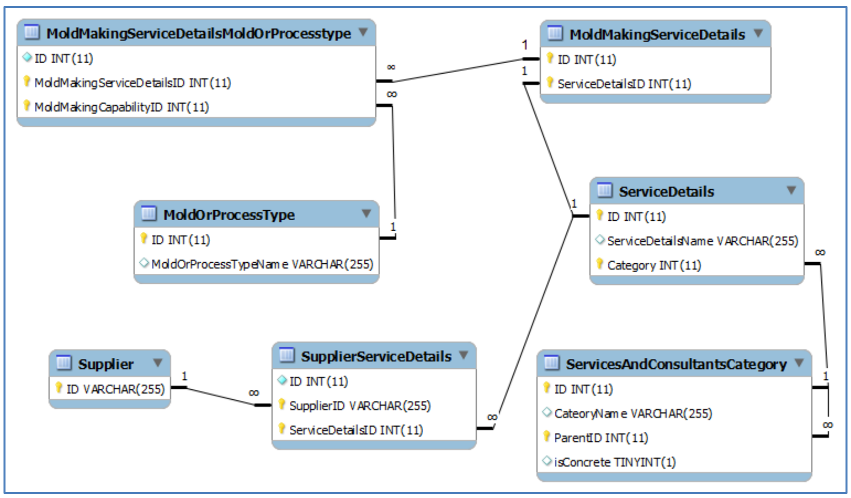
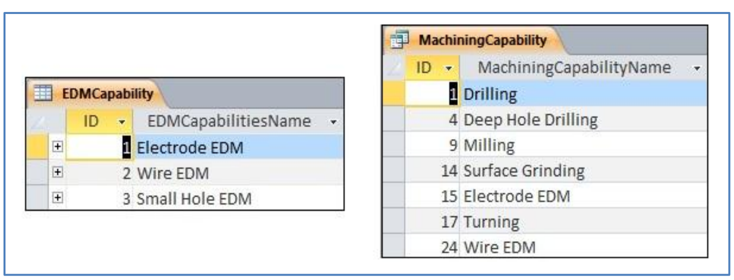



Figure 22: Portal-B's relational schema of its manufacturing service model related to the mold making service



EDMCapability

| ID | EDMCapabilitiesName |
|---|---|
| 1 | Electrode EDM |
| 2 | Wire EDM |
| 3 | Small Hole EDM |

MachiningCapability

| ID | MachiningCapabilityName |
|---|---|
| 1 | Drilling |
| 4 | Deep Hole Drilling |
| 9 | Milling |
| 14 | Surface Grinding |
| 15 | Electrode EDM |
| 17 | Turning |
| 24 | Wire EDM |

Figure 23: Example data of Portal-B's `EDMCapability` and `MachiningCapability` tables

```
Class: ProductOrServiceCategory
Class: CustomMfgAndFab
  SubClassOf: ProductOrServiceCategory
Class: Machining
  SubClassOf: CustomMfgAndFab
Class: CNCMachining
  SubClassOf: Machining
Class: EDM SubClassOf: Machining
Class: RamEDM SubClassOf: EDM
Class: SinkerEDM SubClassOf: EDM
Class: WireEDM SubClassOf: EDM
Class: SmallHoleEDM SubClassOf: EDM
Class: MicroHoleEDM SubClassOf: EDM

Class: Material
Class: PreHardenedStainlessSteel
  SubClassOf: Material
Class: StainlessSteel17-4
  SubClassOf: Material
Class: Ferrous SubClassOf: Material
Class: Titanium SubClassOf: Material
Class: ToolSteel SubClassOf: Material
Class: AlloySteel SubClassOf: Material

Class: Supplier
Class: Capability
Class: MaxPartLength

ObjectProperty: hasMaxPartLength
ObjectProperty: hasMaterialCapability
ObjectProperty: hasSupplier
DataProperty: hasMaxLengthInclusive
DataProperty: hasMinLengthExclusive

Individual: Supplier_1 Types: Supplier
Individual: Supplier_5 Types: Supplier
```

```
Individual: Wire_EDMMaxPartLength
  Types: MaxPartLength
  Facts:
    hasMinLengthExclusive  0.0,
    hasMaxLengthInclusive  30.0

Individual: CNCMaxPartLength
  Types: MaxPartLength
  Facts:
    hasMinLengthExclusive  0.0,
    hasMaxLengthInclusive  88.0

Individual: Wire_EDM
  Types:
    Capability, SinkerEDM and WireEDM,
    (hasMaterialCapability some
       Ferrous) and
    (hasMaterialCapability some
       AlloySteel) and
    (hasMaterialCapability some
       PreHardenedStainlessSteel)
  Facts:
    hasSupplier  Supplier_5,
    hasMaxPartLength Wire_EDMMaxPartLength

Individual: CNC
  Types:
    Capability, CNCMachining and RamEDM
       and WireEDM and MicroHoleEDM,
   (hasMaterialCapability some
      StainlessSteel17-4) and
   (hasMaterialCapability some
      Titanium) and
   (hasMaterialCapability some ToolSteel)
  Facts:
    hasMaxPartLength CNCMaxPartLength,
    hasSupplier  Supplier_1
```

**Figure 24: Portal-A's proprietary model**

```
#Service Categories
Class: ServiceConsultantsCategory
Class: ContractMfgAndFab
  SubClassOf:
     ServiceConsultantsCategory
Class MachineShopServices
  SubClassOf: ContractMfgAndFab
Class: EDMMachiningServices
  SubClassOf: MachineShopServices

Class: Supplier

#MS = Machine Shop
Class: ServiceDetails
Class: EDMMachiningServiceDetails
    SubClassOf: ServiceDetails
Class: MSServiceDetails
  SubClassOf: ServiceDetails

Class: Material
Class: StainlessSteel SubClassOf: Material
Class: HardenedMetals SubClassOf: Material

Class: PartLengthCapability
Class: MaterialCapability
```

```
#These are for classifying the Process
#Capability of a service
Class: MachiningProcessCapability
Class: EDMProcessCapability
Class: Milling
  SubClassOf: MachiningProcessCapability
Class: Drilling
  SubClassOf: MachiningProcessCapability
Class: ElectrodeEDM
  SubClassOf: EDMProcessCapability,
     MachiningProcessCapability
Class: SmallHoleEDM
  SubClassOf: EDMProcessCapability
Class: WireEDM
  SubClassOf: EDMProcessCapability

ObjectProperty: hasMaterialCapability
ObjectProperty: hasServiceDetails
ObjectProperty: hasPartLengthCapability
ObjectProperty: hasEDMProcessCapability
ObjectProperty: hasMSProcessCapability
DataProperty: minPartLengthExclusive
DataProperty: maxPartLengthInclusive
```

**Figure 25: Class and property declarations of Portal-B's proprietary models (Note: We use the '#' to denote commented lines instead of OWL annotation for brevity. This is not Manchester OWL syntax.)**

```
Individual: PLCLessThan2.5
  Types: PartLengthCapability
  Facts:
    maxPartLengthInclusive  2.5,
    minPartLengthExclusive  0.0
Individual: PLC2.5To7.5
  Types: PartLengthCapability
  Facts:
    maxPartLengthInclusive  7.5,
    minPartLengthExclusive  2.5
Individual: PLC7.5To15
  Types: PartLengthCapability
  Facts:
    maxPartLengthInclusive  15.0,
    minPartLengthExclusive  7.5
Individual: PLC15To120
  Types: PartLengthCapability
  Facts:
    minPartLengthExclusive  15.0,
    maxPartLengthInclusive  120.0
Individual: S_4EDMService
  Types:
    EDMMachiningServiceDetails,
    EDMMachiningServices,
    hasEDMProcessCapability some
       (ElectrodeEDM and
       SmallHoleEDM and WireEDM),
    (hasMaterialCapability some
       HardenedMetals) and
    (hasMaterialCapability some
       StainlessSteel) and
    (hasMaterialCapability some
       (HardenedMetals
       and StainlessSteel))
  Facts:
    hasMaterialCapability
       S_4EDMMtlCapability,
    hasPartLengthCapability
       PLCLessThan2.5,
    hasPartLengthCapability
       PLC2.5To7.5,
    hasPartLengthCapability
       PLC7.5To15,

Individual: S_4EDMMtlCapability
  Types:
    MaterialCapability,
    (hasMaterialCapability some
    StainlessSteel) and
    (hasMaterialCapability some
    HardenedMetals) and
    (hasMaterialCapability some
      (HardenedMetals and
    StainlessSteel))
```

```
Individual: S_9MSServiceMtlCapability
  Types: MaterialCapability,
    (hasMaterialCapability some
    StainlessSteel) and
    (hasMaterialCapability some
    HardenedMetals) and
    (hasMaterialCapability some
      (HardenedMetals and
    StainlessSteel))

Individual: S_9MSService
  Types: MSServiceDetails,
    MachineShopServices,
    hasMSProcessCapability some
       (Drilling and ElectrodeEDM
          and Milling),
    (hasMaterialCapability some
       HardenedMetals) and
    (hasMaterialCapability some
       StainlessSteel) and
    (hasMaterialCapability some
       (HardenedMetals and
       StainlessSteel))
  Facts:
    hasMaterialCapability
       S_9MSServiceMtlCapability,
    hasPartLengthCapability
       PLCLessThan2.5,
    hasPartLengthCapability PLC2.5To7.5,
    hasPartLengthCapability PLC15To120,
    hasPartLengthCapability PLC7.5To15,


Individual: Supplier_4
  Types: Supplier
  Facts: hasServiceDetails S_4EDMService

Individual: Supplier_9
  Types: Supplier
  Facts: hasServiceDetails S_9MSService
```

**Figure 26: Instance data of Portal-B's proprietary model**

**Table 1: Manufacturing service queries conveying semantic mediation goals**

| Query Requirement Name | Query Requirement | Expected Results (From Figure 9 and Figure 10) | Terminology Used | Query Name |
|---|---|---|---|---|
| Q1 | Identify services having the following manufacturing capabilities: has sinker EDM capability, can machine parts 15 cm and smaller, and can machine pre-hardened stainless steel. | • Wire_EDM<br>• CNC<br>• S_4EDMService<br>• S_9MSService | Portal-A | Q1.1 |
| | | | Portal-B | Q1.2 |
| Q2 | Identify services having the following manufacturing capabilities: has small hole EDM capability and can machine parts 3 cm to 8 cm. | • CNC<br>• S_4EDMService | Portal-A | Q2.1 |
| | | | Portal-B | Q2.2 |

Table 2: Semantic issues resolutions attested by queries

| Query Requirement Name | Semantic Issue Resolved | Semantic Issue Resolution Explanation |
|---|---|---|
| Q1 | Semantic Ambiguity | The Wire_EDM service in Portal-A is registered to the Sinker EDM service category. The CNC service of Portal-A is registered to the Ram EDM service category. The Sinker EDM and Ram EDM are independent service categories in Portal-A, yet they are semantically equivalent. After semantic mediation, both services are retrievable even when queried using only one of the terms. |
| | Naming Conflict | The sinker EDM manufacturing capability is identified by the term Sinker EDM in Portal-A, while it is identified by the term Electrode EDM in Portal-B. After semantic mediation, services registered using one term are also retrievable using the other term. |
| | Structural Conflict | The sinker EDM manufacturing capability is identified by the service category attribute in Portal-A, while it is identified by the process capability in Portal-B. After semantic mediation, services registered using one logical structure are also retrievable using the other logical structure. |
| | Generalization Conflict | The sinker/electrode EDM capability concept is modeled only under the EDM category in Portal-A, while it is modeled under both the EDM Machining Service category and the Machine Shop Service category in Portal-B (see Figure 5,Figure 6, Figure 21, and Figure 23). After semantic mediation, services registered to have the sinker/electrode EDM capability under any of the three categories are retrievable using any of the three categories. |
| | Data Value Conflict | Portal-A represents the part length capability with the maximum value, while Portal-B represents it with a set of value ranges. After semantic mediation, related services from both portals are retrieved when the query requirement uses the maximum value condition. |
| Q2 | Semantic Ambiguity | The CNC service in Portal-A is registered to the Micro Hole EDM service category. The Small Hole EDM and Micro Hole EDM are independent service categories in Portal-A, yet the former indeed subsumes the latter from the semantic mediation perspective. After semantic mediation, services registered to the Micro Hole EDM category including the CNC service are retrieved when queried for some services under the Small Hole EDM category. In other words, the Micro Hole EDM category becomes a subcategory of the Small Hole EDM category. |
| | Expressivity Conflict | The CNC service of Portal-A is registered to the Micro Hole EDM category – a concept which does not exist in Portal-B, yet it can be retrieved with the query Q2.2 that uses only Portal-B terms, specifically the Small Hole EDM process. This is because the small hole EDM concept subsumes the micro hole EDM concept. |
| | Structural Conflict | The small hole EDM manufacturing capability is identified by the service category in Portal-A, while it is identified by the process capability in Portal-B. After semantic mediation, services registered using one logical structure are also retrievable using the other logical structure. |
| | Data Value Conflict | Portal-A represents part length capability with the maximum value, while Portal-B represents it with a set of value ranges. After semantic mediation, related services from both portals are retrievable when the query requirement uses the value range condition. |

Table 3: Mapping ontology between Portal-A and the merged ontology, $\mu_{(1,\Sigma)}$

| Axiom ID | Axiom Type | Portal-A Term | Mapping in Merged Ontology Term |
|---|---|---|---|
| **A1** | E | pa:hasMaterialCapability | mo:hasMaterial |
| **A2** | E | pa:hasSupplier | mo:hasActor |
| **A3** | E | pa:hasMaxPartLength | mo:hasPartLengthCapability |
| **A4** | E | pa:hasMaxLengthInclusive | mo:hasValueRangeMaxInclusive |
| **A5** | E | pa:hasMinLengthExclusive | mo:hasValueRangeMinExclusive |
| **A6** | E | pa:ProductOrServiceCategory | mo:Service |
| **A7** | E | pa:CNCMachining | mo:Service and (mo:hasProcess some mo:Machining) |
| **A8** | E | pa:EDM | mo:Service and (mo:hasProcess some mo:ElectroDischargeMachining) |
| **A9** | E | pa:SinkerEDM | mo:Service and (mo:hasProcess some mo:SinkerEDM) |
| **A10** | E | pa:WireEDM | mo:Service and (mo:hasProcess some mo:WireEDM) |
| **A11** | E | pa:RamEDM | mo:Service and (mo:hasProcess some mo:SinkerEDM) |
| **A12** | E | pa:SmallHoleEDM | mo:Service and (mo:hasProcess some mo:SmallHoleEDM) |
| **A13** | E | pa:MicroHoleEDM | mo:Service and (mo:hasProcess some mo:MicroHoleEDM) |
| **A14** | E | pa:Capability | mo:Service |
| **A15** | E | pa:Supplier | mo:Supplier |
| **A16** | E | pa:MaxPartLength | mo:PartLengthCapability |
| **A17** | E | pa:Ferrous | mo:Ferrous |
| **A18** | E | pa:AlloySteel | mo:AlloySteel |
| **A19** | E | pa:PreHardenedStainlessSteel | mo:StainlessSteel and (mo:isHardenedByHeatTreatment value true) |
| **A20** | E | pa:ToolSteel | mo:ToolSteel |
| **A21** | E | pa:Titanium | mo:Titanium |
| **A22** | E | pa:StainlessSteel17-4 | mo:StainlessSteel_17-4PH |

Table 4: Mapping ontology between Portal-B and the merged ontology, $\mu_{(2, \Sigma)}$

| Axiom ID | Axiom Type | Portal-B Term | Mapping Axiom in Merged Ontology Term |
|---|---|---|---|
| **B1** | E | pb:hasMSProcessCapability | mo:hasProcess |
| **B2** | E | pb:hasEDMProcessCapability | mo:hasProcess |
| **B3** | E | pb:hasMaterialCapability | mo:hasMaterial |
| **B4** | I | pb:hasServiceDetails | mo:hasActor |
| **B5** | E | pb:hasPartLengthCapability | mo:hasPartLengthCapability |
| **B6** | E | pb:maxPartLengthInclusive | mo:hasValueRangeMaxInclusive |
| **B7** | E | pb:minPartLengthExclusive | mo:hasValueRangeMinExclusive |
| **B8** | E | pb:Supplier | mo:Supplier |
| **B9** | E | pb:ServiceDetails | mo:Service |
| **B10** | E | pb:MSServiceDetails | mo:Service and (mo:hasProcess some mo:Machining) |
| **B11** | E | pb:EDMMachiningServiceDetails | mo:Service and (mo:hasProcess some mo:ElectroDischargeMachining) |
| **B12** | E | pb:PartLengthCapability | mo:PartLengthCapability |
| **B13** | E | pb:EDMProcessCapability | mo:ElectroDischargeMachining |
| **B14** | E | pb:MachiningProcessCapability | mo:Machining |
| **B15** | E | pb:WireEDM | mo:WireEDM |
| **B16** | E | pb:ElectrodeEDM | mo:SinkerEDM |
| **B17** | E | pb:SmallHoleEDM | mo:SmallHoleEDM |
| **B18** | E | pb:Milling | mo:Milling |
| **B19** | E | pb:Drilling | mo:Drilling |
| **B20** | E | pb:EDMMachiningServices | mo:Service and (mo:hasProcess some mo:ElectroDischargeMachining) |
| **B21** | E | pb:MachineShopServices | mo:Service and (mo:hasProcess some mo:Machining) |
| **B22** | E | pb:Supplier | mo:Supplier |
| **B23** | E | pb:MaterialCapability | mo:Material |
| **B24** | E | pb:StainlessSteel | mo:StainlessSteel |
| **B25** | E | pb:HardenedMetals | mo:Metal and (mo:isHardenedByHeatTreatment value true) |

Table 5: Mapping class axioms to enhance semantics

| Axiom ID | Mapping Class | Mapping Class Definition |
|---|---|---|
| **B26** | PLCLessThan15 | mo:Service and<br>(mo:hasPartLengthCapability some<br>(mo:PartLengthCapability and<br>(mo:hasValueRangeMaxInclusive value 15.0) and<br>(mo:hasValueRangeMinExclusive value 0.0))) |
| **B27** | PLCLessThan15 | pb:ServiceDetails and<br>(pb:hasPartLengthCapability value pb:PLC2.5To7.5) and<br>(pb:hasPartLengthCapability value pb:PLC7.5To15) and<br>(pb:hasPartLengthCapability value pb:PLCLessThan2.5) |
| **B28** | PLC7.5To120 | mo:Service and<br>(mo:hasPartLengthCapability some<br>(mo:PartLengthCapability and<br>(mo:hasValueRangeMaxInclusive value 120.0) and<br>(mo:hasValueRangeMinExclusive value 7.5))) |
| **B29** | PLC7.5To120 | pb:ServiceDetails and<br>(pb:hasPartLengthCapability value pb:PLC15To120) and<br>(pb:hasPartLengthCapability value pb:PLC7.5To15) |
| **B30** | PLCLessThan120 | mo:Service and<br>(mo:hasPartLengthCapability some<br>(mo:PartLengthCapability and<br>(mo:hasValueRangeMaxInclusive value 120.0) and<br>(mo:hasValueRangeMinExclusive value 0.0))) |
| **B31** | PLCLessThan120 | pb:ServiceDetails and<br>(pb:hasPartLengthCapability value pb:PLC15To120) and<br>(pb:hasPartLengthCapability value pb:PLC2.5To7.5) and<br>(pb:hasPartLengthCapability value pb:PLC7.5To15) and<br>(pb:hasPartLengthCapability value pb:PLCLessThan2.5) |