AN ONTOLOGY ARCHITECTURE FOR STANDARDS INTEGRATION AND CONFORMANCE IN MANUFACTURING

Laurent Deshayes,
Lami, IFMA, Les Cézeaux, BP 265, 63175 Clermont Ferrand, France
Email: Laurent.Deshayes@ifma.fr

Sebti Foufou[1]
LE2i, UMR CNRS 5158, University of Burgundy, B.P. 47870, Dijon, France.
sfoufou@u-bourgogne.fr

Michael Gruninger
DPG Group, MSID Division, National Institute of Standards and Technology[2]
Mail Stop 8263, 100 Bureau Drive, Gaithersburg, MD 20899, U.S.A.
gruning@nist.gov

Abstract:

Standards reflect consensus on the semantics of terms. When used to communicate, whether between people or software systems, standards ensure the communication is correct. Different standards have different semantics for the same terms and express common concepts using different terms and in different ways. Communication between software systems based on different standards is sometimes difficult to achieve. Standards integration concerns the explicit representation of the overlapping sets of concepts in standards and the differences in their semantics to ensure that these standards are used consistently together. This in turn enables software that is based on integrated standards to interoperate, reducing the cost of software integration. Standards conformance determines whether the interpretation of the standardized terms used by software applications is consistent with semantics given by the standards. This paper proposes a general architecture to design ontologies for standards integration and conformance in manufacturing engineering. The ontology architecture is divided into four levels: vendor, standards, domain, and core. Manufacturing turning tools are used as a case study to illustrate the approach. Finally this paper offers some short examples of first order logic propositions.

Keywords: Product manufacturing, turning tools, standards integration, ontologies, first order logic, ontology architecture

1 Introduction

The manufacturing phase is a very important stage of any product lifecycle. To reduce production costs, companies need standardized information exchange that allows one to represent manufacturing processes capabilities as precisely as possible. In addition, knowledge for manufacturing engineering is a very complex set of information resources that generally crosses several engineering domains of competency. Standards are, in some way, consensual models created to reduce as much as possible semantic and inferential

[1] He is currently a guest researcher at MSID Division, NIST, Gaithersburg, MD 20899. sfoufou@cme.nist.gov

[2] Official contribution of the National Institute of Standards and Technology; not subject to copyright in the United States.

mismatches. In manufacturing engineering, most of these standards are decades old and are mainly based on mathematical or empirical models that presume sophisticated interpretation by highly skilled people and may confuse the non expert. To avoid semantic ambiguities across organization(s), most of these standards include natural language definitions and drawings illustrating the relevant information. A formal machine and human interpretable representation of implicit information in standards is intended to avoid defining and creating redundant standards and to add more expressiveness to facilitate their use and interpretation by automated information systems.

Various groups within industry, academia, and government [1] have been developing sharable and reusable knowledge bases known as ontologies [2-4]. The purpose of ontologies within engineering is to make explicit, for a given domain, the knowledge contained in engineering software and in business procedures [5]. In manufacturing, standards are used extensively in all phases of production, design and maintenance. The data, information and knowledge represented with standards need to be sharable across the enterprise. But, despite being written by people with a common background, the definitions and relations within standards have subtle inconsistencies, which may be exacerbated by the use of data from multiple standards or the implementation of the standard within manufacturing information systems. Furthermore, manufacturers frequently define their own extensions to the standard to satisfy specific needs not covered by the standards.

In this paper, we discuss the problems of standards integration and conformance checking for manufacturing applications. A four-level ontology architecture is proposed to deal with this problem, and is illustrated with axioms of standards for turning tools.

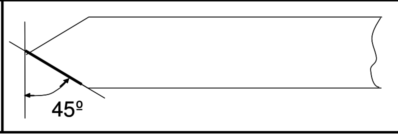
2 Problem of standards integration and conformance

According to the American Office of Management and Budget (OMB) Circular A-119 [6], the term "standard," or "technical standard" as cited in the National Technology Transfer and Advancement Act [6], includes all of the following:

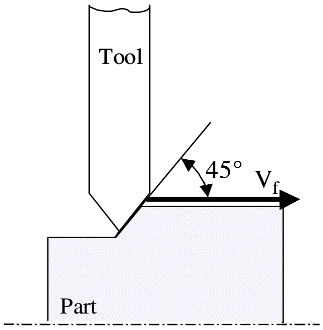
- Common and repeated use of rules, conditions, guidelines, or characteristics for products or related processes and production methods, and related management systems practices.
- The definition of terms; classification of components; delineation of procedures; specification of dimensions, materials, performance, designs, or operations; measurement of quality and quantity in describing materials, processes, products, systems, services, or practices; test methods and sampling procedures; or descriptions of fit and measurements of size or strength.

Standards are often locally extended, but they still form an important aspect of manufacturing knowledge representation as they reflect consensus on the semantics of terms for a wide variety of industries. But standards do not always present enough specific engineering knowledge, which drives vendors to develop their own extensions of the standards or sometimes their own standards. These user-defined extensions are generally required to fully capture the specific engineering knowledge of a company's intellectual capital. In addition, due to the globalization of markets, companies need new ways of translating their own data to their clients or subcontractors to avoid costly errors at the time of interpreting the encoded information. Ontologies are one way to cover such challenges by providing sharable domain concepts and precise relations between concepts.

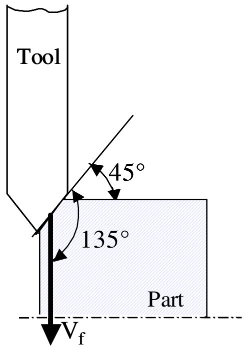
From the information technology point of view, standards may be seen as a list of well-defined concepts and relationships used to describe: i) a particular manufacturing domain (resource, process…) or ii) information about the product during its manufacturing. Nevertheless implicit information only known to the manufacturing specialists is needed for manufacturing applications to achieve full interoperability and to predict feasibility of design solutions [7].

Tool style symbol	Drawing	Process type	Cutting edge angle	Shank type
D		side cutting	45º	Straight shank

(a)



(b)



(c)

Figure 1: The semantic conveyed by drawings for tool style defined in the ISO 5608

Figure 1.a shows an example where adding illustrative drawings help to clarify the standard semantics. According to the ISO 5608 standard [8], the tool style is represented by a symbol and defines three characteristics of the tool: the type of cutting process that a tool can perform (Process Type in Figure 1.a), the value of the cutting edge angle, and the type of shank. From the figure in column two, one can understand much more easily the meaning of the information. One can even deduce more inferences (using personal interpretation) about the context, or process, in which each tool style may be used, or the kind of machining features that each tool style is able to produce, e.g., the semantics conveyed by a tool style of symbol D is that it can perform side turning kind of operations whose features require an angle less than or equal to 45º, such as illustrated in Figure 1.b. It can also be used to perform chamfers which generally require an angle of 45°, See Figure 1.c. But no indication is given in the standard about this knowledge. Also, the reference one can use to determine the cutting edge angle value is not given, which may lead to some confusion, e.g., if the feed speed is taken as a reference the cutting edge angle is 45°, in the case of an extern turning operation (Figure 1.b), and 135° in the case of a chamfering operation (Figure 1.c).

Due to such implicit knowledge contained in standards, recent works done by standardization committees used information technology tools, such as the EXPRESS language or the Extensible Markup Language (XML) [9; 10], to define their concepts. Figure 2, extracted from the ISO/FDIS 14649-121 [11], shows a data structure modelled in EXPRESS-G [12] for turning tools. Notice that most of the concepts are optional (dashed lines in Figure 2 within the domain of computerized numerical controllers; the only mandatory concept is **cutting_edge_location**. More interesting, **code 1** and **code 2** in Figure 2 are specified (in natural language) as being defined according to ISO 5608 [8] and ISO 1832 [13] respectively, each code is related to the cutting tool dimensions information. However, there is no clearly defined correspondence between the codes, the corresponding standards [8; 13], and the **Tool_dimension** entities of the model. Within other domains, such as cutting process modelling [14], other geometrical aspects of the cutting tool are required. Moreover, the concept of cutting edge angle is represented by two entities: the **end_cutting_edge_angle** and the **side_cutting_edge_angle**. The ISO 3200 [15] defines them respectively as the working cutting edge angle and the working minor cutting edge angle. Here again, slightly different implications, difficult to represent even for the expert, were added to the new standard.

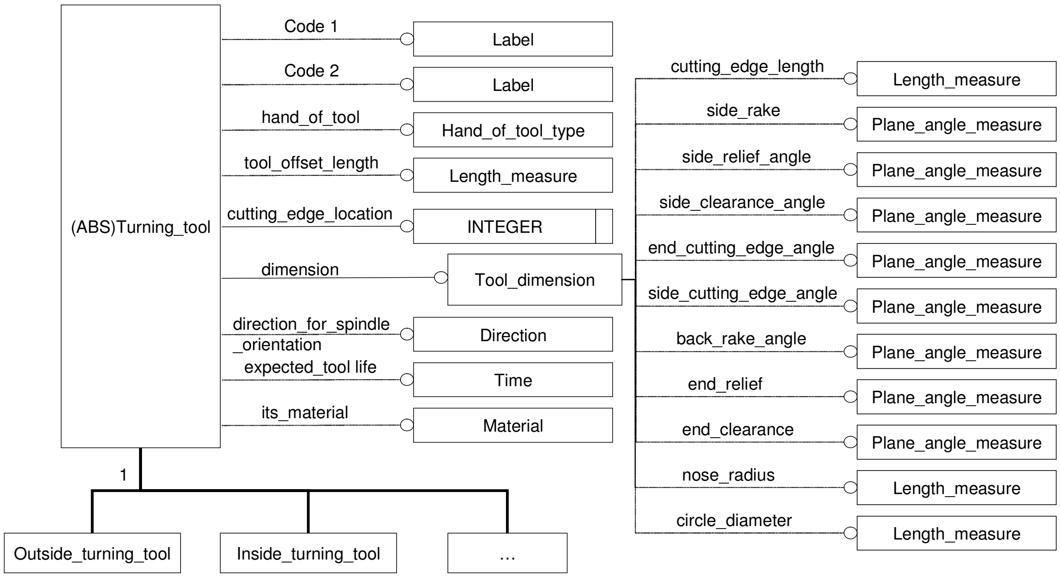


Figure 2: A cutting tool model in EXPRESS-G from ISO 14649-121

Since standard customizations (or extensions) and duplication of information in standards are unavoidable in practice, it is required to provide a modular and flexible approach to explicitly restrict the relations and definitions of domain-specific concepts. This needs to address two industrial needs for representing implicit knowledge: standards integration and standards conformance. **Standards integration** concerns the explicit representation of the overlapping sets of concepts in standards and the clear characterization of the differences in their semantics to ensure that these standards are used consistently together and with other systems. **Standards conformance** should determine whether the interpretation of the terminology used by software applications is consistent with semantics of the terminology given in standards. In practice both standards integration and conformance need engineering knowledge that over-crosses several manufacturing domains. Finally, mechanisms to incorporate domain knowledge are required to answer queries in a broader manufacturing level. These mechanisms include the representation of: i) implicit knowledge that may be derived from standards; and ii) the extensions defined by users of these standards. To answer these queries, a four-level ontology architecture is presented in the next section.

3 Proposed architecture

Recently, ontologies have been extensively used to represent semantics and knowledges. In particular an ontology makes explicit the knowledge that people often take for granted or as implicit knowledge in a domain. Although there is a wide variety of ontology approaches, all approaches agree that there are two essential components of any ontology: i) a vocabulary of terms that refer to the things (or concepts) of interest in a given domain; ii) some specifications of meaning for the terms, grounded in some form of logic. What distinguishes one ontology approach from another is the way the relationships among terms are specified. An ontology supports representation of a very rich variety of structural and non-structural relations such as generalization, inheritance, aggregation and instantiation. It can supply a precise model for software applications. Ontologies can be represented using a wide variety of logical languages which are understandable both by human beings and machines [16], such as propositional logic, first order logic and semantic web languages. Propositional logic lacks the expressive power to model concisely an environment with many objects and facts. First Order Logic (FOL) has much more expressivity and can represent much more complex relations between objects [16]. The Ontology Web Language (OWL) is the language widely used by the semantic web community [16-18]. In comparison to FOL, OWL is less expressive and based on a taxonomic model, which remains the core for many ontology tools. In this paper, due to the subtleties of concepts' semantic of the proposed ontology architecture, we specify the ontologies using the knowledge interchange format [19; 20], which is a first order language designed to support the interchange of knowledge among heterogeneous computer systems.

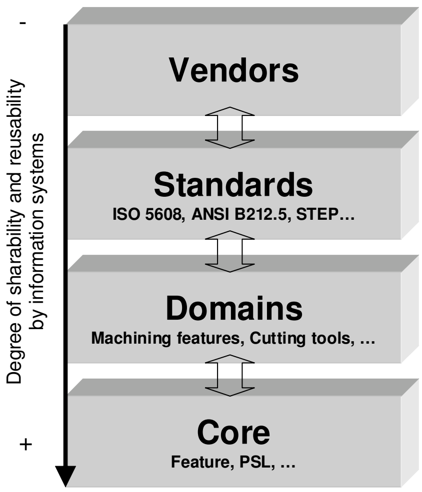


Figure 3: The proposed four-level ontology

Nelson and Schneider showed that product modelling architectures require ontology and interoperability standards to provide better functionalities to PLM systems within current commercial applications [21]. In [1] it was shown that enhanced ontology architectures will soon be required. Rather than developing large monolithic ontologies, complex systems will be supported by ontologies that can be decomposed into modules. This is particularly relevant for the problem of standards integration and conformance for manufacturing resources and processes. Modularization makes it easier to update the ontologies, since additional extensions are added without altering the existing ontology architecture. Moreover, by allowing different modules to provide alternative definitions, a modular organization makes differences in the semantics of overlapping concepts more explicit.

In Figure 3, we propose a simple architecture to overcome all these specifications for standards integration and conformance. It is divided into four levels; each level consists of different sets of ontologies. From bottom to top, ontologies of a given level are used to integrate those of the immediate upper level. This structure imposes rather strong requirements on the representational adequacy of the ontologies – they must not only be strong enough to capture the intended semantics of the terminology, but they must also be expressive enough to allow the generation of semantics-preserved mappings between the less generic ontologies.

Vendor ontologies are developed and used by providers of business applications or manufacturing resources. They are very specific to the provider's business activities. It is often the case that new terminologies are created by private companies and included with their products, such as the component technology called COM developed by Microsoft and largely used in the visual basic programming community.

Standards ontologies are created by consortia to standardize applications and provide uniform interfaces. Since a number of companies are involved in the development of these standards, terms and concepts are more generic than those defined at the vendor level. Nevertheless, they are the first level that presents some "common" agreement on the usage, definition or interpretation of terms or manufacturing systems.

Domain ontologies are used to check consistency between standards ontologies. In particular, standards are often developed for overlapping domains, which can lead to conflicting definitions for the shared terms; even if the terms used in different standards are closely related, subtle consequences can make it difficult for the terminology to be reusable and consistently interpretable. In addition, domain ontologies are used to provide rigorous definitions for the implicit assumptions made within the standards' ontologies. For example, many standards include visual drawings as part of their specification, but do not provide an explicit axiomatization of the intuitions behind these drawings; domain ontologies are used to provide such axiomatizations, which can then be used for automated reasoning such as conformance checking.

Core ontologies are the most generic concepts that cross multiple domains. They cover concepts such as process, product, resource, and geometry. In addition to providing formal specifications of the semantics of these generic concepts, core ontologies are also designed to maximize sharability and reusability, and hence do not make any ontological commitments that are not shared by all related domain ontologies.

4 Case study

Cutting tools are important resources of manufacturing processes since they affect the quality and the cost of manufactured parts. Due to the proliferation of cutting tools for mechanical machining, standardization has been required for several decades. Today, even if the use and the codification of cutting tools is well understood and standardized, there are still difficulties to really fully encapsulate in a consensual way all explicit and implicit concepts used to describe these resources. These difficulties are particularly important when terminologies are exchanged between world-wide applications. In this section, the four-level architecture introduced in section 3 is illustrated with standardized concepts used for turning tools encoding. Turning tools are a subclass of cutting tools used to machine rotational features on a class of machine tools known as lathe centers [11]. This study focuses only on insertable turning tools, although the approach can be generalized to other classes of manufacturing resources and processes. As showed in Figure 4, the cutting edge of an insertable turning tool is held by an interchangeable insert. An insertable turning tool is generally composed of two assembled parts: the tool body head, which holds the insert, and the tool body shank, which mounts the turning tool to the tool block.

The two standards considered in this study, ISO 5608 [8] and ISO 1832 [13], define codes and associated semantics for the turning tool holders and inserts respectively. The role of these codes is i) to classify those resources; and ii) to represent concisely their properties. Figure 5 shows an example of three codes, one for an insert and two for a tool holder. The insert code and the first tool holder code are based on their respective international standards, while the second tool holder code is obtained from a tool provider catalogue; the two tool holders have a similar geometry. Standardized as well as vendor specific codes are composed of compulsory and optional symbols. Compulsory symbols are the first height symbols for inserts, and the first nine symbols for tool holders. For the entire code to be syntactically valid and conform to the standard, its symbols have: i) to be placed following a specific order; ii) to respect a precise syntax; and iii) to correspond to a specific semantics. Each symbol represents particular properties (dimensional and non dimensional) of the resource and has a semantic

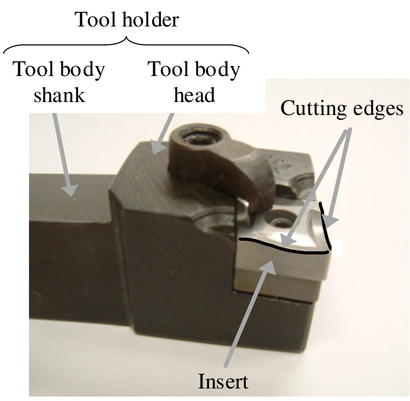


Figure 4. An insertable turning tool

meaning associated with it. More descriptions of tool holders and insert properties are given in standards [8; 13], specialized handbooks [21], or tool manufacturer catalogues.

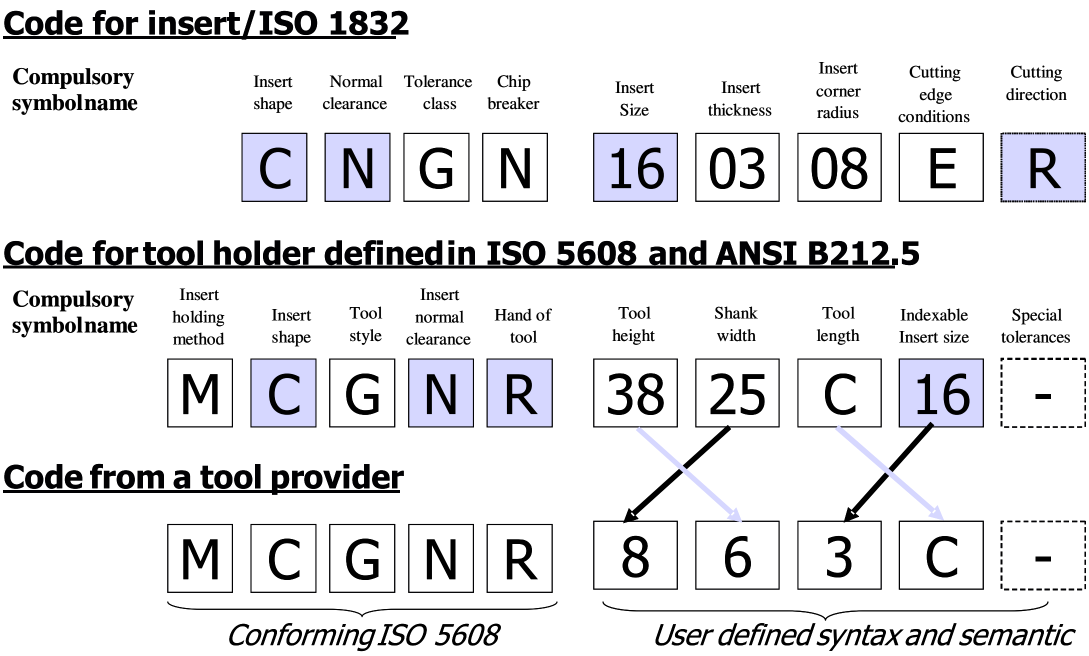


Figure 5. Standardized insert and tool holder codes as well as differences with an industrial tool holder's code

Subtle differences exist between the standardized code and the one provided by the tool provider. First, there is a difference on the dimension system, the ISO and ANSI standards use the metric, while the tool provider use the British system (inches). Therefore for a tool provider, the sixth and seventh symbols are described as: *"The sixth and seventh symbol shall be a significant two-digit number that indicates the holder cross section. For shanks 5/8''*

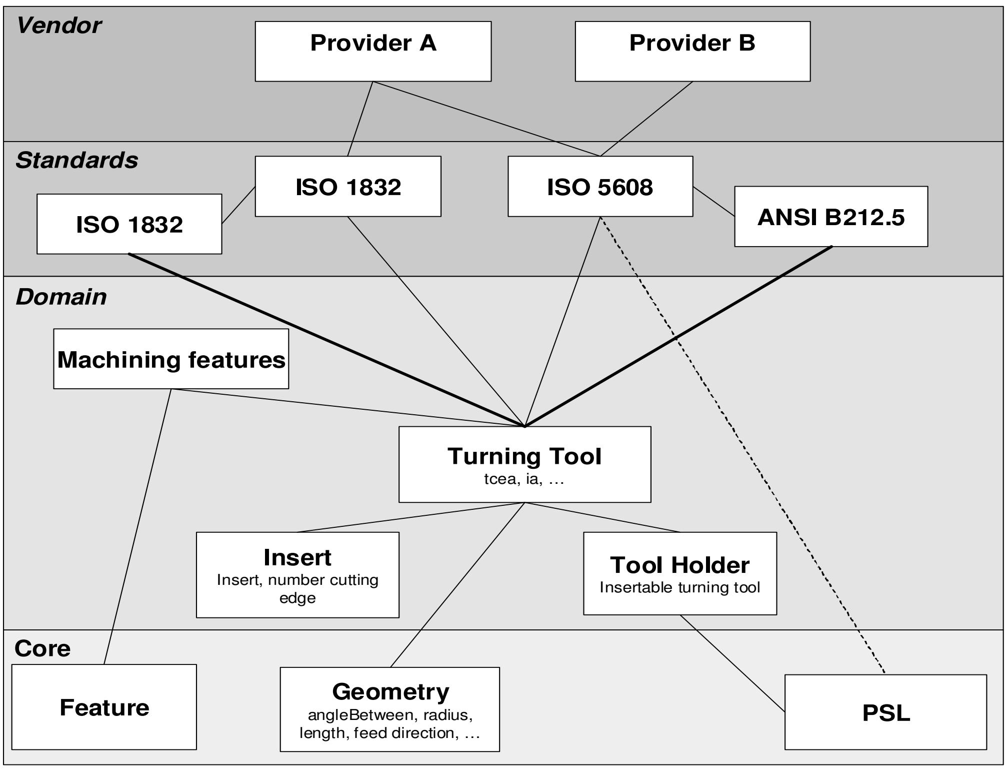


Figure 6. The four-level architecture for cutting tool ontology

square and over, the number will represent the number of sixteenth of width and height. For shanks less than 5/8'' square, the number of sixteenths of cross section will be preceded by a zero. For rectangular holder …", while for the metric standard we will find that these symbols are respectively the tool height and the shank cross section, see [8] for more details. In this example, not only the dimensional system is confusing, but the meaning of the symbols too. In some cases, this meaning seems to indicate that these two symbols were inverted. Concerning the eighth and ninth symbols, they are effectively inverted by the tool provider. Another aspect of these codes consists in determining whether a given insert can be mounted on a particular tool holder. In order to verify this relation, correspondences between the two codes are used. For the example of Figure 5, the encoded tool holders are mountable with the encoded inserts because symbols C, N, 16 and R (optional) in the tool holder's code correspond respectively to the same symbols in the code identifying the insert (dark symbols). These relations can be modelled easily using today's database systems. However, information such as the cutting processes that can be performed or the exact tool geometry is not explicitly represented. In fact, to be more accurate, such information as well as the symbols meaning is generally encoded in the application which does not facilitate systems interoperability.

The four-level ontology approach for the turning tool domain is illustrated in Figure 6. The top level concerns tool provider catalogues and concepts. The specific concepts used by tool vendors (Providers A and B) are defined in relation to the standardized concepts described at the standards ontology level. ISO 1832 [13] for inserts and ISO 5608 [8] for tool holders are internationally standardized concepts that may be encoded at this level. Notice that regional standards are also part of this level, such as the ANSI B212.4 and ANSI B212.5 that are the equivalent, in the United States, to the two previous ISO standards [22]. For standards integration purposes, i.e., for the standardized concepts to be more interpretable by other information systems, the domain level ontology is required to define the concepts and relationships that are generic to a wider manufacturing domain, such as turning tools, tool holders, inserts and machining features (features produced by a turning tool). Finally core ontologies include the most generic and interchangeable concepts. The design of well defined core concepts is a very intensive iterative process that may reach to standardize them, such as the PSL ontology [23].

The third symbol of the tool holder code defined in the standard ISO 5608 is the tool style.

(1)
```
(forall (?x ?y) (implies (ISO_5608_3 ?x ?y)
        (and   (toolStyle ?x ?y) (toolHolder ?x))))
```

Tool style is a property of a tool holder and includes the information about the tool cutting edge angle, shank type and cutting processes which, can be performed with the tool style.

(2)
```
(forall (?w)  (implies  (ISO_5608_3 ?x ?y)
        (exists ( ?u ?v ?w)
        (and   (toolHolder_tcea ?x ?u) (shankType ?x ?v) (cuttingProcess ?x ?w)))))
```

For all tool holder code ?x the third symbol ?y for tool holder implies that the tool holder requires performing a side cutting activity or a end cutting activity or both.

(3)
```
(forall (?x ?y) (implies  (ISO_5608_3 ?x ?y)
        (or  (requires sideCutting ?x) (requires endCutting ?x)
             (and (requires sideCutting ?x) (requires endCutting ?x)))))
```

Figure 7. Axioms for ISO 5608 ontology

The transition from vendor to standards ontology can be considered as a requirement for integrating specific knowledge with software applications. As a fact, it cannot be imagined that each company would build its own ontological theory without using existing standards. Therefore the mapping from vendor ontologies to the standards ontology may strongly reduce the amount of concepts to consensually define. Vendors may avoid the standards ontology level, but they risk introducing a vendor-specific ontology into the domain level. The transition with the standards ontology level guarantees that vendor specific knowledge does not cross the levels without being discussed by a wider community. In the remainder of the paper the next three levels (from standards to core ontologies) and their transitions are analyzed.

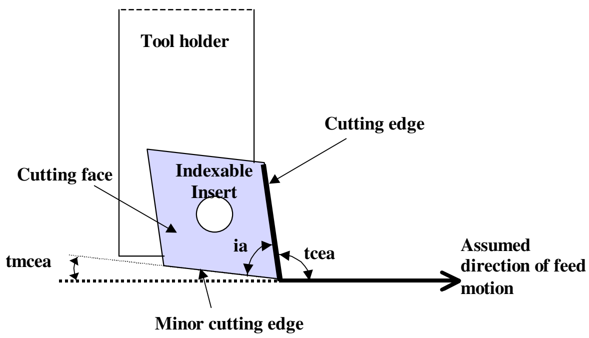


Figure 8. Characteristics of turning tool geometry

In the ISO 5608 standard, the tool style includes information about tool cutting edge angle, shank type, and the associated cutting processes (see Figure 1). Some relevant axioms for this concept are given in Figure 7. Notice that a full study case is out of the scope of this paper. The predicate **ISO_5608_3** represents the third symbol of the standardized code described in ISO 5608 and is defined in axiom (1) using the knowledge interchange format. Axiom (2) specifies the properties of **toolStyle**. The **toolHolder_tcea** predicate represents the tool cutting edge angle. Axiom (3) specifies the values that instances of **shankType** and **cuttingProcess** can have. This is also particularly useful for standard conformance purposes. The predicate **requires** is part of the PSL ontology [23]. Therefore the **cuttingProcess** concept is assumed to be a PSL activity. The use of PSL in describing cutting tool knowledge is particularly interesting to provide additional inferences on processes and improve standard integration. The use of such generic concepts with the domain ontology is detailed below. Axioms such as axiom (3) are particularly useful for standards conformance checking, to determine whether the tool style given by a tool provider conforms to ISO 5608. The tool provider can also define new axioms to describe its own specific knowledge and make it more interpretable by expert agents.

More generic ontology modules for the domain ontology level may be defined. The turning tool geometric concepts used in this example are represented in Figure 8. These concepts, selected from the turning tool standards [8; 13; 15], are: the turning tool holder; the indexable insert; the tool cutting edge; the tool minor cutting edge; the tool cutting edge angle, denoted by **tcea**; the tool included angle, denoted by **ia**; and the tool minor cutting edge angle, denoted by **tmcea**. The direction of feed motion (or the working plan) is a concept used in defining the angles and cutting edges.

Figure 9 shows part of an ontology used to describe turning tool components and geometric concepts. Axiom (4) specifies that a turning tool is composed of an insert and a tool holder. The predicates **insert** and **toolHolder** are primitives within the turning tool ontology. They will be defined in their respective domain ontology. Axioms (5) to (7) respectively define **tcea**, **ia**, and **tmcea**, which are the three major geometric angles considered for describing turning tools (see Figure). Notice that in these axioms predicates are primitives provided by domain and core ontologies. For example, predicates **angleBetween** and **feedDirection,** which represent the value of an angle included between two lines and the assumed direction of feed motion, have their definition provided in the geometry ontology.

The predicate **cuttingEdge**, which represents the tool cutting edge, may be a primitive of the turning tool ontology or defined in the insert ontology, depending on the context.

An insertable turning tool is the assembly (mounting) of an indexable insert and of a tool holder

```
(4)    (forall (?x ?y) (implies   (turningTool ?x ?y )
                          ( and  (toolHolder ?x) (insert ?y ) (mounting ?x ?y))))
```

The tool cutting edge angle, tcea, is the angle included between the cutting tool edge and the feed rate direction.

```
(5)    (forall (?x ?y ?w ) (iff (tcea ?x ?y ?w)
                          (exists (?v ?z)
                              (and  (turningTool  ?x ?y) (cuttingEdge ?x ?y ?z)
                                  (feedDirection ?x ?y ?v) (angleBetween ?z ?v ?w)))))
```

The included angle is the angle included between the cutting tool edge and the secondary tool edge.

```
(6)    (forall (?x ?y ?w) (iff (ia ?x ?y ?w)
                          (exists (?z ?v)
                             (and (turningTool ?x  ?y) (cuttingEdge ?x  ?y ?z)
                                 (minorCuttingEdge ?x ?y ?v) (angleBetween ?v ?z ?w)))))
```

The tool minor cutting edge angle, tmca, is the angle included between the secondary cutting tool edge and the feed rate direction.

```
(7)    (forall (?x ?v ?y) (iff   (tmcea ?x ?y ?v)
                          (exists (?w ?z)
                             (and (turningTool ?x ?y ) (minorCuttingEdge ?x ?y ?w)
                                 (feedDirection ?x ?y ?z) (angleBetween ?z ?w ?v)))))
```

Figure 9: Axioms for integrating turning tools with process information using PSL

Figure 7 and Figure 9 are representative of standards and domain ontologies. The transitions and the mapping between these two levels and the core ontology serve different purposes. Domain ontologies provide better reusability and consistency between the overlapping concepts and implicit assumptions made between standards. It reduces the number of translations from standards to core ontologies. For example both ANSI B212.5 and ISO 5608 concepts may be mapped to the turning tool domain ontology (axioms (4) to (7)), and the use of the tool holder domain ontology ensures stronger integrability of these two standards with the core ontology. In fact, the use of domain ontologies to make such transition provides a rigorous structuring and guarantees better completeness of the overall ontology. Some core concepts may be used at the different levels to facilitate the integration of specific concepts but at the risk of making the integration of standards more complicated. As an example, the turning tools domain ontology may use the **require** concept from PSL to indicate that a turning process requires a particular turning tool [24; 25]. The dashed line in Figure 6 represents this relationship. Indirect definition of this relationship is also given through the domain level ontologies with the links from standards to turning tools to tool holders to PSL. In this case, more properties (e.g. the machining feature, the turning tool, the tool holder and the insert) of the process maybe inferred. Oppositely, a direct relation (and modelling) between the ISO 5608 standards ontology level and PSL core ontology would impede such knowledge reasoning and discovery.

A tool holder is either a EndToolHolder, a SideToolHolder, or both

```
(8)        (forall (?x) (iff   (toolHolder ?x)
                               (or (EndToolHolder ?x ) (SideToolHolder ?x )
                                   (and (EndToolHolder ?x ) (SideToolHolder ?x )))))
```

An external turning activity occurs if and only if there exists a side tool holder or an end and side tool holder within the activity.

```
(9)        (forall (?x) (iff  (and (externTurning ?x ) (activity_occurence ?x))
                              (exists (?y) (9)
                                     (or  (SideToolHolder ?y)
                                          (and (EndToolHolder ?x ) (SideToolHolder ?x )))))
```

A tool holder is a reusable resource within a turning process activity

```
(10)       (forall (?x ?y) (implies  (and (toolHolder ?x) (turningProcess ?y))
                                     (reusable ?x ?y)))
```

An insert is a wearable resource within a turning process activity

```
(11)       (forall (?x ?y) (implies  (and (Insert ?x) (11) (turningProcess ?y))
                                  (wearable ?x ?y)))
```

Figure 10: Defining tool holders concepts using a core ontology defined in PSL

The inferences that can be done in the indirect representation are illustrated using the axioms of Figure 10, which are defined for the tool holder ontology using a core ontology defined in PSL. From Axiom (9), it may be inferred that if a process planner defines an external turning operation (i.e., an activity called external turning) then either a side tool holder or an end and side tool holder must be used to accomplish this operation. In addition, if an insertable turning tool is used, then it is automatically inferred that a tool holder and an insert are required (axiom (4)). Thanks to PSL theory it is also inferred that a tool holder does not need to be changed between activity occurrences, since axiom (10) defines a tool holder as a reusable resource for a turning process activity. Finally, using axiom (11) it can be inferred that the insert will need to be changed after some activity occurred. These reasoning can be mapped with standards terminology when required using the mapping between standards and the domain ontology level. In the first scenario it would be only inferred that **endCutting** is an activity; the tool holder needs to be used in an activity called **endCutting**, but nothing is deduced from the activity itself and the relations between the concepts. In addition, because of important number of standards, the use of core concepts at this level needs to be reduced in order to avoid replications of inferences.

5 Conclusion

In manufacturing, standards represent consensual definitions of the concepts within a particular domain; consequently organizations' specific knowledge is not represented, which leads manufacturing actors (such as tool providers) to define their own extensions of the standards. Semantic conflicts may appear during the mapping of terms between these extensions and the standards. The next generation of manufacturing systems needs more rigorous foundations of semantics than what is currently provided by data models and architectures. The four-level ontology architecture proposed in this paper for manufacturing resources standards integration and conformance checking may provide such a foundation. The generic core ontologies of the lower level enable full integration between the proposed architecture and other existing applications. In addition, due to the modularity of the proposed approach concepts from the different levels may be used, which facilitates the integration of new concepts. The domain ontology level is certainly the most important level since it can be used as a base to build more effective core ontology by categorizing and classifying manufacturing concepts and relationship in a coherent modular architecture. Of course, more efforts have to be done by manufacturers and application developers to provide such architecture and the consensus required by each level. The next step of this work is to develop a full application of the proposed architecture and validate it on a complete manufacturing scenario.

6 Disclaimer

No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. Certain commercial equipments, instruments, or materials are identified in this report in order to facilitate better understanding. Such identification does not imply recommendations or endorsement by the National Institute of Standards and Technology, nor does it imply the materials or equipment identified are necessarily the best available for the purpose.

7 References

[1] Ray, S., “Interoperability standards in the semantic web," *Journal of Computing and Information Science in Engineering*, Vol. 2, 2002, pp. 65-69.

[2] Gruber, T. R., "Towards principles for the design of ontologies used for knowledge sharing," Padova, Italy, 1993.

[3] Ciocoiu, M., Gruninger, M., and Nau, D., “Ontologies for integrating engineering applications," *Journal of Computing and Information Science in Engineering*, Vol. 1, 2001, pp. 45-60.

[4] Antoniou, G., and van Harmelen, F., *A Semantic Web Primer*, The MIT Press, Cambridge, Massachusetts, 2004, pp. 109-146.

[5] Uschold, M. and Gruninger, M., “Ontologies: Principles, Methods, and Applications," *Knowledge Engineering Review*, Vol. 11, No. 96, 1996, pp. 137.

[6] American Office of Management and Budget. http://standards.gov/standards_gov/ . 2005.

[7] Zhao, Y., Ridgway, K., and Al-Ahmari, A. M. A., "Integration of CAD and a cutting tool selection system," *Computers & Industrial Engineering*, Vol. 42, 2002, pp. 17-34.

[8] ISO 5608:1995, Turning and copying tool holders and cartridges for indexable inserts - Designation.

[9] ISO 10303-11: 1994, Industrial automation systems and integration -- Product data representation and exchange -- Part 11. Description methods: the EXPRESS language reference manual.

[10] Extensible Markup Language (XML). http://www.w3.org/XML/ . 2005.

[11] ISO/FDIS 14649-12, Industrial automation systems and integration - physical device control - Data model for computerized Numerical Controllers - Part 12: Process data for turning. 2005.

[12] Schenck, D., and Wilson, P. R., *Information modeling: the EXPRESS way*, Oxford University Press, New York, 1994.

[13] ISO 1832:1991, Indexable Inserts for cutting tools - Designation.

[14] Tlusty, G., *Manufacturing Processes and Equipment*, Prentice Hall 1999, pp. 415-610.

[15] ISO 3002-1:1982, Basic quantities in cutting and grinding - part 1: Geometry of the active part of cutting tools - General terms, reference systems, tool and working angles, chip breakers.

[16] Russell, S., and Norvig, P., *Artificial Intelligence A Modern Approach*, 2nd ed., Printice Hall 2003, pp. 193-319.

[17].Daconta, M. C., Obrst, L. J., and Smith, K. T., T*he Semantic Web*, Wiley Publishing 2003.

[18] Web Ontology Language (OWL). http://www.w3.org/2004/OWL/ . 2005.

[19] Menzel, C., and Hayes, P., "SCL: A Logic Standard for Semantic Integration," Sanibel Island, 2003.

[20] Genesereth, M. R., and Fikes, R. E., "Knowledge Interchange Format, Version 3.0, Reference Manual," Computer Science Department, Stanford University, Logic Group Report Logic-92-1, Stanford, California, 1992.

[21] Nelson, D. H., and Schneider, G. J., *Applied manufacturing process planning : with emphasis on metal forming and machining*, Prentice Hall 2001.

[22] ANSI B212.4 B2212.5. http://www.ccpa.org/pubs.html . 2005.

[23] Gruninger, M. and Menzel, C., “Process Specification Language: Principles and Applications," *AI Magazine*, Vol. 24, 2003, pp. 63-74.

[24] Process Specification language (PSL) . http://www.nist.gov/psl . 2005.

[25] Deshayes, L., El Beqqali, O., and Bouras, A., "The use of Process Specification Language for cutting processes," *International Journal of Product Development*, Vol. 2, No. 3, 2005, pp. 236-253.