A First-Order Cutting Process Ontology for Sheet Metal Parts

Michael GRÜNINGER [a], Arnaud DELAVAL [b]
[a] *Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, Canada M5S 3G8*
[b] *IFMA Les Cezaux BP 265 63175 Clermont Ferrand, France*

Abstract. The semantic integration of manufacturing systems has been impeded by the lack of rigorous ontologies for specific domains of manufacturing processes and resources. In this paper we present a cutting process ontology for 2D shapes such as sheet metal parts, axiomatized in first-order logic. This ontology is an extension of the ontology of ISO 18629 (Process Specification Language) and an earlier shape ontology first used to support object recognition. The full ontology consists of an axiomatization of all possible ways to change a surface as the result of a cutting process and a taxonomy of cutting processes. All component ontologies are verified using representation theorems.

Keywords. manufacturing, ontologies, first-order logic, Process Specification Language

1. Motivation

Although 3D machining receives a great deal of attention, manufacturing of two-dimensional parts is also widespread. The cutting of sheet metal has applications in car bodies, airplane wings, medical tables, and construction materials. Similar processes are used not only in metal machining but also in the cutting of materials ranging from paper to fabric. Earlier work in the application of knowledge representation to sheet metal manufacturing processes has been restricted to approaches that are not based on a formal logic. The work in [6] uses a rule-based approach to capture the relationship between particular features of a sheet metal part and the possible operations that can be performed to produce the desired features. Approaches such as [7] specify grammar rules to define the slitting operations that can be done on any given sheet metal plate.

One drawback of this earlier work is the lack of sharability and reusability for the specifications of the manufacturing processes such as cutting or punching. Rather than a set of rules, we need a logical framework that can support both consistency-checking (to verify plans composed of cutting processes) as well as automated inference to reason about the consequences of particular cutting processes (for example, can one perform a different cutting process within a plan and still achieve a particular feature within the final product).

Another desirable application that is not supported by earlier work is the retrieval of cutting operations and partial plans from process repositories. Such repositories presume the existence of a classification of cutting processes. Moreover, this cannot be an ad hoc

classification; it should be one that is provably correct and complete with respect to the underlying definitions of cutting processes.

To address the above shortcomings and additional requirements, we propose an ontology for cutting processes in which the class definitions and other constraints are axiomatized in first-order logic. If we consider the class of manufacturing processes defined as activities that change shape, then cutting processes are the subclass of such shape-changing processes in which at least two new edges are created as a result of an occurrence of the process. Consequently, the cutting process ontology presented in this paper is based on two existing first-order ontologies – the ontology of 2D shapes introduced in the CardWorld Ontology [4], and the ontology of ISO 18629 (Process Specification Language [3], [2]). The Cutting Process Ontology consists of three sets of first-order axioms:

- Shape Ontology
- Shape Cutting Ontology
- Cutting Process Taxonomy

In the remainder of this paper, we will consider each of the first-order ontologies within the Cutting Process Ontology. We not only present the axioms within each ontology, but we also demonstrate the verification of the axioms with respect to their intended models.

2. PSL Ontology

The purpose of PSL-Core ([3], [2]) is to axiomatize a set of intuitive semantic primitives that is adequate for describing the fundamental concepts of manufacturing processes. Consequently, this characterization of basic processes makes few assumptions about their nature beyond what is needed for describing those processes, and the Core is therefore rather weak in terms of logical expressiveness.

Within PSL-Core [1], there are four kinds of entities required for reasoning about processes – activities, activity occurrences, timepoints, and objects. Activities may have multiple occurrences, or there may exist activities which do not occur at all. Timepoints are linearly ordered, forwards into the future, and backwards into the past. Finally, activity occurrences and objects are associated with unique timepoints that mark the begin and end of the occurrence or object.

Within the PSL Ontology, the theory $T_{occtree}$ extends the theory of $T_{pslcore}$[2]. An occurrence tree is a partially ordered set of activity occurrences, such that for a given set of activities, all discrete sequences of their occurrences are branches of the tree. An occurrence tree contains all occurrences of *all* activities; it is not simply the set of occurrences of a particular (possibly complex) activity. Because the tree is discrete, each activity occurrence in the tree has a unique successor occurrence of each activity. Every sequence of activity occurrences has an initial occurrence (which is the root of an occurrence tree).

[1] The axiomatization of PSL-Core in CLIF (Common Logic Interchange Formt) can be found at `http://www.mel.nist.gov/psl/psl-ontology/psl_core.html`

[2] The axioms of $T_{occtree}$ in CLIF can be found at `http://www.mel.nist.gov/psl/psl-ontology/part12/occtree.th.html`

Most applications of process ontologies are used to represent dynamic behaviour in the world so that intelligent agents may make predictions about the future and explanations about the past. In particular, these predictions and explanations are often concerned with the state of the world and how that state changes. The PSL core theory T_{disc_state} is intended to capture the basic intuitions about states and their relationship to activities[3].

Within the PSL Ontology, state is changed by the occurrence of activities. Intuitively, a change in state is captured by a state that is either achieved or falsified by an activity occurrence. We therefore use the $prior$ relation to specify the properties (known as fluents) that are intuitively true prior to an activity occurrence and also the $holds$ relation that specifies the fluents that are intuitively true after an activity occurrence.

Furthermore, state can only be changed by the occurrence of activities. Thus, if some state holds after an activity occurrence, but after an activity occurrence later along the branch it is false, then an activity must occur at some point between that changes the state. This also leads to the requirement that the state holding after an activity occurrence will be the same state holding prior to any immediately succeeding occurrence, since there cannot be an activity occurring between them.

3. Shape Ontology

The Shape Ontology (T_{shape}) is based on the CardWorld Ontology [4], which is a first-order ontology for 2D-object recognition in scenes with occlusion and images with noise. In the axiomatization of the Shape Ontology (see Figures 1 and 2), we focus on the mereotopological relations (i.e. parthood and connection) rather than geometric relations (such as relative alignment and length of segments, or the notions of curvature or surface area).

3.1. Axiomatization

There are three sorts of objects in the domain – surfaces, edges, and points. Every point is part of some edge and every edge is part of a unique surface. Every surface contains at least two edges and every edge contains at least two points. A vertex is a point that is part of two edges, and only two edges may meet at a point.

In the original CardWorld Ontology, the objects such as surfaces, edges, and points were represented as classes, and the relationships between these objects were represented by relations. On the other hand, we need to be able to represent how properties of surfaces, edges, and points can change as the result of activity occurrences. Moreover, we need to be able to capture the creation of new surfaces, edges, and points.

Within the PSL Ontology, properties of the world that change as the result of activity occurrences are represented as fluents. Consequently, all of the relations in the original CardWorld Ontology become fluents in the Shape Ontology. Thus, the three classes of objects become the unary fluent functions $surface(x)$, $edge(x)$, and $point(x)$, since new surfaces, edges, and points may be created by an activity. The binary fluent function $part(x, y)$ specifies the containment relationships between the elements of surfaces. The ternary fluent function $meet(e_1, e_2, v)$ specifies the relationship between two edges that

[3]The axioms of T_{disc_state} in CLIF can be found at `http://www.mel.nist.gov/psl/psl-ontology/part12/disc_state.th.html`

meet at a vertex (common point). The unary fluent function $outer(e)$ distinguishes edges that are part of a hole within a surface from those edges that are part of the outer boundary of the surface. Finally, the binary fluent function $connected(e_1, e_2)$ captures the relationship between edges that are part of the same boundary within a surface, whether this is the outer boundary or the boundary of a hole.

Within the Cutting Process Ontology, the axioms of the Shape Ontology are state constraints, that is, sentences that must be true prior to any activity occurrence. By the axioms of the PSL Ontology, these sentences also hold after any activity occurrence in the occurrence tree, since they cannot be falsified by any activity occurrence.

3.2. Verification of the Shape Ontology

The ontology is verified by providing a complete characterization of all models of the axioms up to isomorphism. One approach to this problem is to use representation theorems – we evaluate the adequacy of the ontology with respect to some well-understood class of mathematical structures (such as partial orderings, graph theory, and geometry) that capture the intended interpretations of the ontology's terms. Given the definition of some class of structures $\mathfrak{M}$, we prove that the class exists and is nonempty, which also provides a characterization of the structures in the class up to isomorphism. We prove that every structure in the class is a model of the ontology and that every countable model of the ontology is isomorphic to some structure in the class.

To formally capture these intuitions for the Shape Ontology, we first define a few classes of combinatorial structures [1] which will be the building blocks of models for the ontology.

Definition 1 *A tripartite incidence structure is a tuple* $\mathbf{G} = (\Omega_1, \Omega_2, \Omega_3, I)$*, where* $\Omega_1, \Omega_2, \Omega_3$ *are pairwise disjoint sets such that* $I \subseteq (\Omega_1 \times \Omega_2) \cup (\Omega_1 \times \Omega_3) \cup (\Omega_2 \times \Omega_3)$*. Two elements of* $\mathbf{G}$ *that are related by* I *are called incident. A flag of* $\mathbf{G}$ *is a set of elements of* $\Omega_1 \cup \Omega_2 \cup \Omega_3$ *that are mutually incident.*

Using these definitions, our intuitions tell us that scene elements should form tripartite incidence structures in which all maximal flags have three elements, since all objects should be part of some surface, i.e., there should not exist any isolated edges or points.

Definition 2 *A shape structure is a tripartite incidence structure*

$$\mathbb{S} = \langle S, E, P, \mathbf{part} \rangle$$

such that all elements of E *are elements of two maximal flags in* $\mathbb{S}$ *which contain a unique element in* S*.*

The existence of shape structures is established by the following theorem, which also provides a characterization up to isomorphism:

Theorem 1 *A tripartite incidence structure*

$$\mathcal{O} = \langle S, E, V, \mathbf{part} \rangle$$

is a shape structure iff it is isomorphic to the incidence structure $\langle P(G), G, V, \in \rangle$ *in which* G *is a set of cyclic graphs with vertices* V *and* $P(G)$ *is a partitioning of* G*.*

$$\forall x, o\ \neg(prior(point(x), o) \wedge prior(edge(x), o)) \quad (1)$$

$$\forall x, o\ \neg(prior(point(x), o) \wedge prior(surface(x), o)) \quad (2)$$

$$\forall x, o\ \neg(prior(edge(x), o) \wedge prior(surface(x), o)) \quad (3)$$

$$(\forall x, s, o)\ prior(part(x, s), o) \wedge prior(surface(s), o) \supset \neg prior(surface(x), o) \quad (4)$$

$$(\forall x, e, o)\ prior(part(x, e), o) \wedge prior(edge(e), o)$$

$$\supset \neg prior(surface(x), o) \wedge \neg prior(edge(x), o) \quad (5)$$

$$(\forall x, p, o)\ prior(part(x, p), o) \wedge prior(point(p), o)$$

$$\supset \neg prior(surface(x), o) \wedge \neg prior(edge(x), o) \wedge \neg prior(point(x), o) \quad (6)$$

$$(\forall x, s, o)\ prior(part(s, x), o) \wedge prior(surface(s), o)$$

$$\supset \neg prior(surface(x), o) \wedge \neg prior(edge(x), o) \wedge \neg prior(point(x), o) \quad (7)$$

$$(\forall x, e, o)\ prior(part(e, x), o) \wedge prior(edge(e), o)$$

$$\supset \neg prior(edge(x), o) \wedge \neg prior(point(x), o) \quad (8)$$

$$(\forall x, p)\ part(p, x) \wedge point(p) \supset \neg point(x) \quad (9)$$

$$(\forall e, s, v, o)\ prior(edge(e), o) \wedge prior(surface(s), o) \wedge prior(part(e, s), o)$$

$$\wedge prior(part(v, e), o) \supset prior(part(v, s), o) \quad (10)$$

$$(\forall x, o)\ prior(edge(x), o) \supset (\exists s)\ prior(surface(s), o) \wedge prior(part(x, s), o) \quad (11)$$

$$(\forall x, o)\ prior(point(x), o) \supset (\exists e)\ prior(edge(e), s) \wedge prior(part(x, e), o) \quad (12)$$

$$(\forall v, s_1, s_2, o)\ prior(part(v, s_1), o) \wedge prior(part(v, s_2), o) \wedge prior(point(v), o)$$

$$\wedge prior(surface(s_1), o) \wedge prior(surface(s_2), o) \supset (s_1 = s_2) \quad (13)$$

$$(\forall e, s_1, s_2, o)\ prior(part(e, s_1), o) \wedge prior(part(e, s_2), o) \wedge prior(edge(e), o)$$

$$\wedge prior(surface(s_1), o) \wedge prior(surface(s_2), o) \supset (s_1 = s_2) \quad (14)$$

$$(\forall s, o)\ prior(surface(s), o) \supset (\exists e_1, e_2, e_3)\ prior(edge(e_1), o) \wedge prior(edge(e_2), o)$$

$$\wedge prior(edge(e_3), o) \wedge (e_1 \neq e_2) \wedge (e_1 \neq e_3) \wedge (e_2 \neq e_3)$$

$$\wedge prior(part(e_1, s), o) \wedge prior(part(e_2, s), o) \wedge prior(part(e_3, s), o) \quad (15)$$

$$(\forall e, o)\ prior(edge(e), o) \supset (\exists p_1, p_2)\ prior(point(p_1), o) \wedge prior(point(p_2), o)$$

$$\wedge (p_1 \neq p_2) \wedge prior(part(p_1, e), o) \wedge prior(part(p_2, e), o) \quad (16)$$

Figure 1. T_{shape}: Shape axioms.

Two edges meet at a vertex iff they are distinct and the vertex is part of both edges.

$$(\forall e_1, e_2, v, o)\ prior(meet(e_1, e_2, v), o) \equiv$$

$$(prior(edge(e_1), o) \wedge prior(edge(e_2), o) \wedge prior(point(v), o)$$

$$\wedge prior(part(v, e_1), o) \wedge prior(part(v, e_2), o) \wedge (e_1 \neq e_2) \quad (17)$$

Every edge meets another distinct edge.

$$(\forall e_1, o)\ prior(edge(e_1), o) \supset (\exists e_2)\ prior(meet(e_1, e_2, v), o) \quad (18)$$

Exactly two edges meet at a vertex.

$$(\forall e_1, e_2, e_3, v, o) prior(meet(e_1, e_2, v), o) \wedge prior(meet(e_1, e_3, v), o) \supset (e_2 = e_3) \quad (19)$$

All outer edges are connected.

$$(\forall e_1, e_2, o) prior(outer(e_1), o) \supset (prior(outer(e_2), o) \equiv prior(connected(e_1, e_2), o)) \quad (20)$$

Figure 2. T_{shape}: Shape axioms.

We next define the class of structures that are isomorphic to the intended models of T_{shape}.

Definition 3 *Let $\mathfrak{M}^{shape}$ be the class of structures such that $\mathcal{M} \in \mathfrak{M}^{shape}$ iff*

1. *there exists a model $\mathcal{N}$ of $T_{disc\_state} \cup T_{occtree} \cup T_{pslcore}$ such that $\mathcal{N} \subset \mathcal{M}$;*
2. *each element of the occurrence tree in $\mathcal{N}$ is associated with a shape structure.*

The following representation theorem constitutes the verification of the Shape Ontology; as a consequence, it also demonstrates the consistency of the ontology.

Theorem 2 *$\mathcal{M} \in \mathfrak{M}^{shape}$ iff it is isomorphic to a countable model of $T_{shape} \cup T_{disc\_state} \cup T_{occtree} \cup T_{pslcore}$.*

4. Shape Cutting Ontology

Each model of the original shape axioms from the CardWorld Ontology corresponds to a different state within an occurrence tree in a model of $T_{shape} \cup T_{disc\_state} \cup T_{occtree} \cup T_{pslcore}$. On the one hand, changing a shape is equivalent to changing state within the occurrence tree; on the other hand, it is equivalent to a mapping between different models of the original shape axioms. Since we have already shown how the models of the Shape Ontology are isomorphic to a class of tripartite incidence structures, we can characterize all possible mappings between structures in this class. With respect to state, any property of a model that is not preserved by a mapping corresponds to a fluent that is either

achieved or falsified [4]. This correspondence forms the basis for the specification of the models of the axioms in the Shape Cutting Ontology.

4.1. Examples of Shape Cutting Activities

Suppose we begin with the surface depicted in Figure 3(a). With the cutting activity depicted in Figure 3(b), three new edges $\mathbf{e_5}, \mathbf{e_6}, \mathbf{e_7}$ are created; of these three, $\mathbf{e_7}$ is a modified edge, since all of its points were formerly parts of the edge $\mathbf{e_1}$. In Figure 3(c), two new edges $\mathbf{e_5}, \mathbf{e_6}$ are created, neither of which is a modified edge. In both of these cases, no new surface is created.

We next consider cases in which a new surface is created. In Figure 3(d), a new surface is created which contains two new modified edges $\mathbf{e_5}, \mathbf{e_6}$ as well as an existing edge $\mathbf{e_2}$ and a new edge $\mathbf{e_7}$. In addition, the original surface contains a new edge. In Figure 3(e), a new surface is created with two new edges, one of which is modified. In Figure 3(f), a new surface is created and each surface contains a new edge, but none of the new edges are modified.

In all of the above examples, the *outer* fluent was unchanged. In Figure 3(g), two new edges are created which form a hole in the surface. Suppose we begin with the surface in Figure 3(h); the surface in Figure 3(i) depicts the effect of an occurrence of the cutting activity that falsifies the *outer* fluent for all of the edges that were formerly part of the whole. One new outer edge and one new non-outer edge are modified. The activity whose occurrence is depicted in Figure 3(j) is a variant in which the *outer* fluent is falsified but only two edges are created, neither of which is modified.

4.2. Axiomatization

The Shape Cutting Ontology ($T_{cutshape}$ in Figures 4 and 5) begins with the definition of a cutting process to be any activity that creates at least two new edges in some surface. The remaining axioms explicate the different ways that a cutting process can possibly change the properties of a surface and its components. In particular, the axioms specify the conditions under which the fluents that specify the properties of surfaces ($part$, $meet$, and $outer$) can be achieved or falsified by occurrences of cutting processes. For example, the $part$ fluent changes as the result of edges or surfaces being created. New points are never created without an edge being created; if the edge already exists, some points simply become the vertices where other new edges meet.

4.3. Verification of the Shape Cutting Ontology

Models of $T_{cutshape}$ are based on the notion of partial automorphisms of a shape structure. A mapping $\varphi : \mathbb{S} \rightarrow \mathbb{S}$ is a partial automorphism iff it is an isomorphism between substructures of $\mathbb{S}$. We use the notation $dom(\varphi)$ to denote the set of elements in the substructure of $\mathbb{S}$ that is the domain of the mapping. The set of partial automorphisms of a shape structure forms an inverse semigroup, denoted by $PAut(\mathbb{S})$.

[4] The following definitions are used from the PSL Ontology:
$(\forall o, f)\ achieves(o, f) \equiv (\neg prior(f, o) \wedge holds(f, o))$
$(\forall o, f)\ falsifies(o, f) \equiv (prior(f, o) \wedge \neg holds(f, o))$
$(\forall o, f)\ changes(o, f) \equiv (achieves(o, f) \vee falsifies(o, f))$.

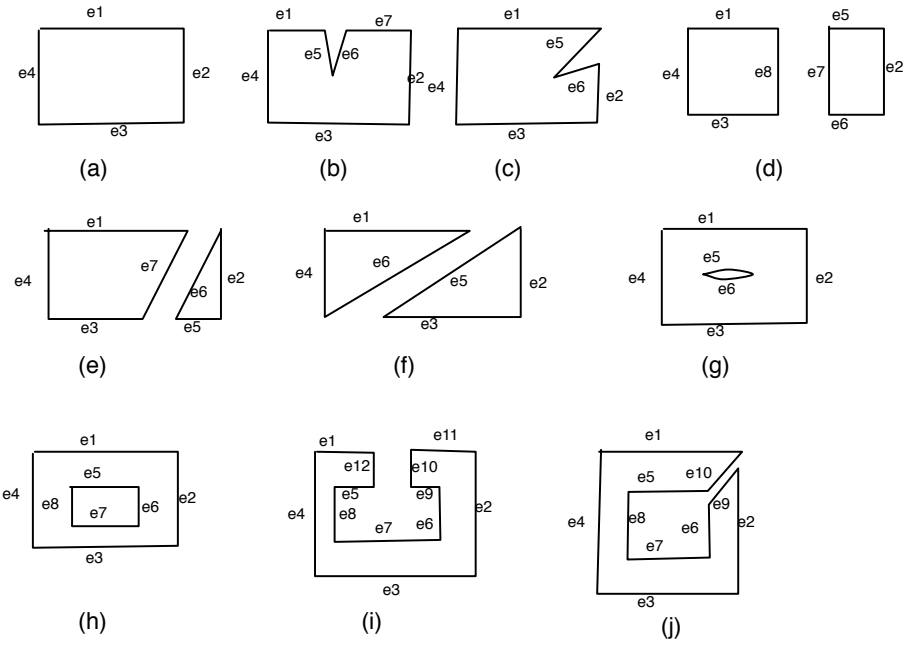

Figure 3. Examples of shape cutting processes.

Definition 4 *Let* $\mathfrak{M}^{cutshape}$ *be the class of structures such that* $\mathcal{M} \in \mathfrak{M}^{cutshape}$ *iff*

1. *there exists* $\mathcal{N} \in \mathfrak{M}^{shape}$ *such that* $\mathcal{N} \subset \mathcal{M}$*;*
2. *each element of the occurrence tree in* $\mathcal{N}$ *is associated with a unique* $\varphi \in PAut(\mathbb{S})$ *such that* $\mathbf{f} \in dom(\varphi)$ *iff* $\langle \mathbf{o}, \mathbf{f} \rangle \notin$ **changes***;*
3. $\langle \mathbf{o}, \mathbf{edge(e_1)} \rangle, \langle \mathbf{o}, \mathbf{edge(e_2)} \rangle \in$ **achieves**

Intuitively, the partial automorphisms capture the substructure of a shape structure that is not changed as the result of an occurrence of a cutting process.

Theorem 3 $\mathcal{M} \in \mathfrak{M}^{cutshape}$ *iff it is isomorphic to a countable model of* $T_{cutshape} \cup T_{shape} \cup T_{disc\_state} \cup T_{occtree} \cup T_{pslcore}$.

5. Cutting Process Taxonomy

The Cutting Process Taxonomy is a classification of cutting processes, that is, activities which satisfy Axiom 21. This is equivalent to characterizing all possible ways to change a surface that satisfies the axioms of T_{shape} in such a way that at least two edges are created. Since the axiomatization in $T_{cutshape}$ provides such a characterization, the classification of cutting processes corresponds to the classification of the models of $T_{cutshape}$.

5.1. Classes of Cutting Processes

Within the PSL Ontology, the taxonomy of activities arises from invariants that are used to classify the models of the core theories [5]. Invariants are properties of models that

A cutting process creates two new edges in some surface.

$$(\forall a)\, cutting(a) \equiv ((\forall o)\, occurrence\_of(o, a) \tag{21}$$

$$\supset (\exists e_1, e_2, s)\, holds(edge(e_1), o) \wedge holds(edge(e_2), o) \wedge (e_1 \neq e_2)$$

$$\wedge prior(surface(s), o) \wedge achieves(o, part(e_1, s)) \wedge achieves(o, part(e_2, s))$$

Surfaces, edges, and points are never destroyed.

$$(\forall x, o, a)\, cutting(a) \wedge occurrence\_of(o, a)$$

$$\supset \neg(falsifies(o, surface(x)) \vee falsifies(o, edge(x)) \vee falsifies(o, point(x))) \tag{22}$$

New surfaces contain existing edges.

$$(\forall s, o) achieves(o, surface(s)) \supset (\exists e) prior(edge(e), o) \wedge achieves(o, part(e, s)) \tag{23}$$

Existing edges never contain new points.

$$(\forall e, v, o, a)\, cutting(a) \wedge occurrence\_of(o, a)$$

$$\wedge prior(edge(e), o) \supset \neg achieves(o, part(v, e)) \tag{24}$$

We can never achieve *meet* for existing edges.

$$(\forall e_1, e_2, s, v, o, a)\, cutting(a) \wedge occurrence\_of(o, a)$$

$$\wedge prior(part(e_1, s), o) \wedge prior(part(e_2, s), o) \supset \neg achieves(o, meet(e_1, e_2, v)) \tag{25}$$

Outer edges are always preserved.

$$(\forall s, o, a)\, cutting(a) \wedge occurrence\_of(o, a) \supset \neg falsifies(o, outer(s)) \tag{26}$$

Every new outer edge contains an existing point.

$$(\forall e, o, a)\, cutting(a) \wedge occurrence\_of(o, a) \wedge achieves(o, edge(e))$$

$$\wedge achieves(o, outer(e)) \supset (\exists p)\, holds(part(p, e), o) \wedge prior(point(p), o) \tag{27}$$

Figure 4. Axioms of $T_{cutshape}$: Shape Cutting Ontology.

are preserved by isomorphism. For some classes of structures, invariants can be used to classify the structures up to isomorphism; for example, vector spaces can be classified up to isomorphism by their dimension. For other classes of structures, such as graphs, it is not possible to formulate a complete set of invariants. Nevertheless, even without a complete set, invariants can still be used to provide a classification of the models of a theory.

We use the following invariants to classify the models of $T_{cutshape}$:

A modified edge is a new edge that does not contain any new points.

$$(\forall e, o)\ modified\_edge(e, o) \equiv \tag{28}$$

$$(achieves(o, edge(e)) \wedge ((\forall p)\ holds(part(p, e), o) \supset prior(point(p), o))$$

A modified edge corresponds to a subset of points for an existing edge.

$$(\forall e_1, o, a)\ cutting(a) \wedge occurrence\_of(o, a) \wedge modified\_edge(e_1, o)$$

$$\supset (\exists e_2) holds(edge(e_2), o) \wedge ((\forall p) holds(part(p, e_1), o) \supset prior(part(p, e_2), o)) \tag{29}$$

Every new modified edge must meet an existing edge.

$$(\forall e_1, o, a)\ cutting(a) \wedge occurrence\_o(o, a) \wedge modified\_edge(e_1, o)$$

$$\supset (\exists e_2, v)\ achieves(o, meet(e_1, e_2, v)) \wedge prior(edge(e_2), o) \tag{30}$$

Figure 5. Axioms of $T_{cutshape}$: Shape Cutting Ontology.

1. number of surfaces that are created (which is either zero or one);
2. number of holes that are destroyed (which is either zero or one);
3. number of edges that are created (which is either two, three, or four);
4. number of pairs of existing edges that are changed (which is either zero, one, or two).

The axioms of $T_{cutprocess}$ in Figure 6 and Figure 7 explicitly define the classes of primitive activities that correspond to each of the possible values for the above invariants.

5.2. Verification of the Cutting Process Ontology

The correctness of $T_{cutprocess}$ is established by the following theorem, which has also been automatically derived using the Prover9 resolution theorem prover.

Theorem 4 *Let* $T_{cpo} = T_{cutprocess} \cup T_{cutshape} \cup T_{shape} \cup T_{disc\_state} \cup T_{occtree} \cup T_{pslcore}$. *The classes activities related to each invariant are disjoint:*
$T_{cpo} \models \neg(\exists a)\ (create\_surface(a) \wedge preserve\_surface(a))$
$T_{cpo} \models \neg(\exists a)\ (destroy\_hole(a) \wedge preserve\_hole(a))$
$T_{cpo} \models \neg(\exists a)\ (preserve\_meet(a) \wedge change\_one\_meet(a) \wedge change\_two\_meet(a))$
$T_{cpo} \models \neg(\exists a)(create\_two\_edge(a) \wedge create\_three\_edge(a) \wedge create\_four\_edge(a))$

6. Summary

In this paper we have introduced an ontology for cutting processes in the manufacturing domain of sheet metal parts. We have presented a first-order axiomatization of classes of intended models and verified the ontology by proving representation theorems with respect to these intended models.

Occurrences of activities in this class create a new surface.

$$(\forall a)\ create\_surface(a) \equiv \tag{31}$$

$$(\forall o)\ occurrence\_of(o, a) \supset (\exists s)\ achieves(o, surface(s))$$

Occurrences of activities in this class do not create a new surface.

$$(\forall a)\ preserve\_surface(a) \equiv \tag{32}$$

$$(\forall o)\ occurrence\_of(o, a) \supset \neg(\exists s)\ achieves(o, surface(s))$$

Occurrences of activities in this class destroy a hole by changing all non-outer edges to outer edges in the same surface.

$$(\forall a)\ destroy\_hole(a) \equiv \tag{33}$$

$$(\forall o)\ occurrence\_of(o, a) \supset (\exists e)\ prior(edge(e), o) \wedge achieves(o, outer(e))$$

Occurrences of activities in this class do not change any existing edges in a surface to be outer edges.

$$(\forall a)\ preserve\_hole(a) \equiv \tag{34}$$

$$(\forall o, e)\ occurrence\_of(o, a) \wedge prior(edge(e), o) \supset \neg achieves(o, outer(e))$$

Occurrences of activities in this class do not change the set of existing edges that meet other existing edges in the surface.

$$(\forall a)\ preserve\_meet(a) \equiv ((\forall o)\ occurrence\_of(o, a) \supset \tag{35}$$

$$\neg(\exists e_1, e_2, v)\ (achieves(o, meet(e_1, e_2, v)) \vee falsifies(o, meet(e_1, e_2, v)))$$

Two existing edges no longer meet after occurrences of activities in this class.

$$(\forall a)\ change\_one\_meet(a) \equiv \tag{36}$$

$$((\forall o)\ occurrence\_of(o, a) \supset (\exists e_1, e_2, v)\ falsifies(o, meet(e_1, e_2, v)))$$

Two pairs of existing edges no longer meet after occurrences of activities in this class.

$$(\forall a)\ change\_two\_meet(a) \equiv ((\forall o)\ occurrence\_of(o, a) \supset \tag{37}$$

$$(\exists e_1, e_2, e_3, e_4, v_1, v_2)\ falsifies(o, meet(e_1, e_2, v_1)) \wedge falsifies(o, meet(e_3, e_4, v_2)))$$

Figure 6. Axioms of $T_{cutprocess}$: Cutting Process Ontology.

Exactly two new edges are created by occurrences of activities in this class.

$$(\forall a)\, create\_two\_edge(a) \equiv ((\forall o)\, occurrence\_of(o, a) \supset \tag{38}$$

$$(\exists e_1, e_2)\, achieves(o, edge(e_1)) \wedge achieves(o, edge(e_2))$$

$$\wedge((\forall e)\, achieves(o, edge(e)) \supset ((e = e_1) \vee (e = e_2)))$$

Exactly three new edges are created by occurrences of activities in this class.

$$(\forall a)\, create\_three\_edge(a) \equiv ((\forall o)\, occurrence\_of(o, a) \supset \tag{39}$$

$$(\exists e_1, e_2, e_3)\, achieves(o, edge(e_1)) \wedge achieves(o, edge(e_2)) \wedge achieves(o, edge(e_3))$$

$$\wedge((\forall e)\, achieves(o, edge(e)) \supset ((e = e_1) \vee (e = e_2) \vee (e = e_3)))$$

Exactly four new edges are created by occurrences of activities in this class.

$$(\forall a)\, create\_four\_edge(a) \equiv ((\forall o)\, occurrence\_of(o, a) \supset \tag{40}$$

$$(\exists e_1, e_2, e_3, e_4)\, achieves(o, edge(e_1)) \wedge achieves(o, edge(e_2))$$

$$\wedge achieves(o, edge(e_3)) \wedge \wedge achieves(o, edge(e_4))$$

$$\wedge((\forall e)\, achieves(o, edge(e)) \supset ((e = e_1) \vee (e = e_2) \vee (e = e_3) \vee (e = e_4)))$$

Figure 7. Axioms of $T_{cutprocess}$: Cutting Process Ontology.

Several lines for future work are evident. First is the extension of the Shape Ontology to incorporate geometric fluents, such as relative alignment and the length of edges. The second is the extension of the Cutting Process Ontology to complex activities, which will support process planning with cutting processes.

References

[1] Buekenhout, F. (1995) *Handbook of Incidence Geometry*. North-Holland.

[2] Bock, C. and Gruninger, M. (2005) PSL: A semantic domain for flow models, *Software and Systems Modeling* 4:209-231.

[3] Gruninger, M. (2003) Ontology of the Process Specification Language, pp. 599-618, *Handbook of Ontologies and Information Systems*, S. Staab (ed.). Springer-Verlag, Berlin.

[4] Gruninger, M. (1993) Grouping Assumptions in Shape-Based Object Recognition, pp. 32-37, *Working Notes AAAI Spring Symposium Series 1993: AI and NP-Hard Problems*, Stanford.

[5] Gruninger, M. and Kopena, J. (2005) Semantic Integration through Invariants, *AI Magazine*, 26:11-20.

[6] de Sam Lazaro, A., and Engquist, D., and Edwards. D.B. (1993) An Intelligent Design for Manufacturability System for Sheet-metal Parts, *Concurrent Engineering: Research and Applications*,1:117-123.

[7] Soman, A. and Padhye, S. and Campbell, M. (2003) Toward an automated approach to the design of sheet metal components, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 17:187-204.