Towards an Ontology for Generative Design of Mechanical Assemblies

Bahar Aameri [a,*], and Hyunmin Cheong [b] and J. Christopher Beck [a]

[a] *Department of Mechanical and Industrial Engineering, University of Toronto, ON, Canada*

[b] *Autodesk Research, Toronto, ON, Canada*

Abstract.

In software-based generative design, a user specifies goals expressed as objectives and constraints to a software application and the application returns a set of feasible and/or optimal design solutions. For problems involving discrete design variables, such as in configuration design, searching the design space is often computationally intractable. Therefore, in the context of the configuration design of mechanical assemblies, we are investigating the use of ontologies to model and reason about designs while providing the ability to more efficiently prune infeasible designs. In this paper, we present an ontology to specify connection, parthood, and shapes in mechanical assemblies, so that the constraints of feasible configurations can be logically expressed and used during generative design. The ontology extends the Ground Mereotopology (MT) of Casati and Varzi to a multi-dimensional mereotopology and combines it with a qualitative shape ontology based on the Hilbert's axiomatic theory of geometry. Relationships between equi-dimensional individuals are captured by MT, while individuals with different dimensions are mereotopologically independent and are related by incidence and betweenness relations. The proposed ontology is a module in the larger effort to develop an overarching *PhysicalWorld Ontology*. We demonstrate the application of the proposed ontology in specifying properties of suspension systems and mechanical joints.

Keywords: assembly ontology, generative design, mereotopology, shape, boundary

1. Introduction

One of the emerging trends in manufacturing is the realization of software-based *generative design*: a user specifies design goals in the form of objectives and constraints and a software application returns a set of feasible and/or optimal design solutions. In the engineering design literature, notions such as “design optimization”, “design synthesis”, or “design automation” (Radford and Gero, 1987; Papalambros and Wilde, 2000; Antonsson and Cagan, 2005; Chakrabarti et al., 2011) have been used to describe research endeavors taken to enable such a step in a design work-flow.

While a wide variety of computational algorithms have been applied for generative design, they all share the common challenge. For almost all design problems, the number of solutions is extremely large and exploring even a small portion of the design space is computationally intractable. This challenge is especially prominent when the design problem involves discrete variables such as in a configuration design problem, where the goal is to compose a set of components via particular connections into an assembly that produces desired behaviors (Mittal and Frayman, 1989; Wielinga and Schreiber, 1997; Brown, 1998; Levin, 2009). In mechanical engineering, the challenge is even more significant because evaluation of a single candidate design solution may require physics-based simulation (e.g., running a multi-body dynamics simulation) which is expensive and represents a black-box objective function.

[*] Corresponding author: First Author, address of first author.

One of the key approaches taken to solve configuration design problems is to limit the search space to regions containing feasible solutions through the use of constraint programming techniques (e.g., O'Sullivan (1999); Fleishanderl et al. (1998)). In this approach, constraints corresponding to restrictions on the values that sets of variables can take on are modeled and imposed to reduce the number of possible designs. However, such constraints must be carefully formulated using quantitative variables with a predefined set of arithmetic and logical operators. An alternative, also well-known, approach is to use shape or generative grammars (Stiny, 1980), which specify rules to modify a given design, within a generate-and-test framework (Shea et al., 1997; Schmidt et al., 2000; McKay et al., 2012). While the rules are intended to produce only feasible solutions, they are not proven to be complete and so may result in a partial investigation of the design space.

In contrast, one may ignore feasibility and employ evolutionary search techniques, such as genetic algorithms, which have been successful in searching a large design space in some engineering applications (Deb and Goyal, 1996; Man et al., 1996). However, in our experience the ratio of feasible solutions to infeasible solutions in a typical mechanical configuration design problem is so small that extremely fast pruning operations are needed: it appears necessary, in fact, to reason directly about feasibility.

Considering these challenges, we are investigating the use of logical ontologies to model qualitative constraints and to prune infeasible designs during search. For example, a model of feasible designs could be generated based on logical axioms, which would represent a smaller sub-space that needs to be explored by a constraint programming solver. Feasible models could be defined as those satisfying logical statements related to physical structures such as “All components must be connected to each other” or physical behaviors such as “This particular component shall not move in the horizontal direction”, which could reduce the need to run physics-based simulation during search.

In this paper, we present an ontology, called the *Assembly Ontology*, a module of an overarching ontology called the *PhysicalWorld Ontology* which is under development. The Assembly Ontology is developed to specify connection, parthood, shape, and boundary constraints for a generative design software tool. It extends the Ground Mereotopology (MT) of Casati and Varzi (1999) to a multi-dimensional mereotopology and combines it with a qualitative shape ontology developed based on the Hilbert's axiomatic theory of geometry (Hilbert, 1902). In the current work, we evaluate the proposed ontology by representing the qualitative properties of feasible mechanical assemblies, using three types of suspension systems as examples.

The balance of the paper is organized as follows: we begin by a short overview of the the PhysicalWorld Ontology in Section 2 and then review related work in Section 3. Section 4 presents a motivating use case of the generative design of a suspension system and discusses the potential applications of the PhysicalWorld Ontology to this use case. In Section 5 we discuss the fundamental ontological commitments and choices for the Assembly Ontology. Section 6 presents the axiomatization of the Assembly Ontology based on the semantic requirements of Section 5. We conclude the paper by using the Assembly Ontology to axiomatize three types of suspension systems in Section 7.

2. An Overview of the PhysicalWorld Ontology

The PhysicalWorld Ontology is a suite of ontologies we are developing that aims to axiomatize concepts and properties required for representation and reasoning about physical domains (Aameri and Gruninger, 2017). It consists of three main modules, each with their own sub-modules and axiomatized in the language of first-order logic: the Assembly Ontology, the Occupation Ontology, and the Kinematics Ontology.

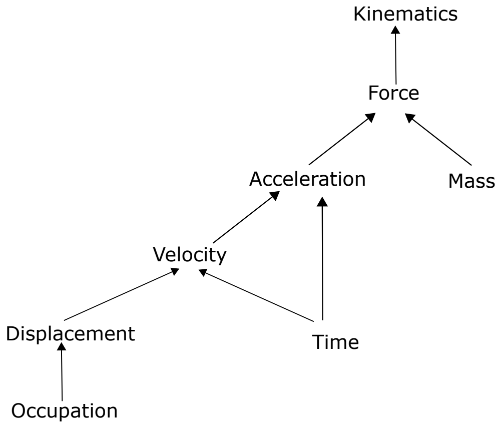

Fig. 1. Modules of the Kinematics Ontology. Solid lines denote conservative extension.[1]

The Assembly Ontology. The Assembly Ontology is the module of the PhysicalWorld Ontology that is presented in this paper and discussed in detail in Sections 5 and 6.

The Occupation Ontology. The Occupation Ontology specifies the relationship between a physical object and the space it occupies. It contains two distinct categories of entities: spatial regions and physical objects, where the former are considered as abstract entities in which physical objects are located. For example, the Eiffel tower is a physical object, which occupies a part of the three-dimensional region denoted as Champ de Mars in Paris. The entities in each category are of different dimensionality and the relationships between entities are axiomatized by a multi-dimensional mereotopology. However, entities in different categories are mereotopologically independent and physical objects are related to spatial regions only by occupation relations. The axioms of the ontology enforce that the mereotopological relations between physical objects are mirrored in the mereotopological relations between the corresponding spatial regions. That is, if a physical object a is part of (connected to) another physical object b, then the region occupied by a is part of (connected to) the region occupied by b.

The Kinematics Ontology. The Kinematics Ontology is an axiomatization of fundamental concepts required for qualitative representation of kinematics of physical systems. Accordingly, it includes six modules, each corresponding to a fundamental concept in describing kinematics of physical bodies: ontologies for Mass, Time, Displacement, Velocity, Acceleration, and Force. Figure 1 illustrates the modules of the Kinematics Ontology and their relationships.

Considering the quantitative formulation of these concepts, the Force Ontology depends on the Mass and Acceleration Ontologies; the Acceleration Ontology extends the Time and Velocity Ontologies; and the Velocity Ontology is an extension of the Time and the Displacement Ontologies. The Displacement Ontology itself depends on the Occupation Ontology since representing displacement requires a representation of physical location.

[1] An extension T_2 of a theory T_1 is conservative if for all sentences Φ in the language of T_1, Φ is entailed by T_2 iff Φ is entailed by T_1. T_2 is a non-conservative extension otherwise. A corollary of this definition is that T_2 is a conservative extension of T_1 if the language of T_2 extends the language of T_1.

3. Related Work

We categorize the existing related work in ontologies into three groups:

(1) ontologies of general concepts in assembly design and process
(2) feature ontologies that classify features of an assembly product
(3) ontologies that specify properties of a specific feature of assembly products (e.g., structure, joints, or components).

The present work falls under the third category.

General Concepts in Assembly Design and Process. There have been a number of efforts to develop general assembly ontologies. Dartigues et al. (2007)'s ontology, intended for interoperability among computer-aided design systems, extends NIST's Core Product Model (CPM) ontology (Fenves, 2001) with three generic classes: the Common Feature Ontology, the Design Feature Ontology, and the Process Planning Ontology. Borgo and Leitão (2007) present a first-order ontology for manufacturing scheduling and control operations based on DOLCE (Masolo et al., 2003). More recently a Common Logic based ontology for assembly design and process planning was developed by Imran and Young (2015, 2016). The ontology classifies and defines generic manufacturing concepts such as assembly features, tolerance, assembly fits, and assembly resources.

Mosca et al. (2009) propose an upper-lever OWL ontology for assembly process design based on Function-Behavior-Structure model (Gero and Kannengiesser, 2003). The ontology includes two disjoint super classes, $Function$ and $DesignObject$, and two binary relations, $needA$ and $partOf$, where $needA$ and $partOf$ define a mereology over elements of $Function$ and $DesignObject$, respectively. Members of $DesignObject$ and $Function$ are related by a binary relations $hasA$ and design objects are classified based on the role they play (i.e., their functions). More specifically, functions are represented through a functional decomposition captured by $needA$ and the mereology over design objects is identified with respect to the functional decomposition.

There have been efforts towards the standardization of manufacturing product data, including assembly products. The most widespread of them is STEP (STandard for the Exchange of Product data model) (ISO10303), an international standard describing product data spanning design to manufacturing. Within STEP, ISO10303-109 ("Kinematic and geometric constraints for assembly models") considers assembly models as a collection of positioned, oriented parts and describes different types of contact, as well we kinematic degrees of freedom between parts.

OntoSTEP (Barbau et al., 2012) is the OWL translation of STEP schema while the Open Assembly Model (OAM) (Fiorentini et al., 2007) is standardization effort extending NIST's CPM ontology by classes concerning relative position, orientation, location and connection between components of assemblies.

Feature Ontologies. The first systematic attempts to create an ontology for design features were made by Borst et al. (1997) and Horváth et al. (1998). Borst et al.'s Component Ontology is a module of the PhysSys ontology (Borst et al., 1997) for specifying the relationships between components of a physical system. It consists of a mereology and a topology over components and a system ontology that describes the relationship between a system, its components, and the external environment.

Based on an analysis of a friction pair, Horváth et al. (1998) suggest describing design features in terms of three fundamental concepts: (1) the set of components of the product (called *entities*); (2) the arrangements of the components (called *situations*); and (3) the physical environment the product is within (called *phenomena*). A situation has multiple aspects including involved components and their spatial interaction, relative position of the components, shapes and the morphological conditions of physical connections of the components, and conceivable movements of the components (e.g., the degree of freedom and orientation of movement).

Both the Component Ontology and that due Horváth et al. (1998) provide axiomatizations for some of their classes, however, the axiomatizations are incomplete.

Ontologies for Specific Properties of Assembly Products. Kim et al. (2006) suggest a vocabulary of assembly terms with an English description for each term which is used to derive a class hierarchy and a set of relationships between the classes that are specified by constraints in OWL and SWRL.

Kim et al. (2008) attempt to axiomatize assembly joining methods classified by Kim et al. (2006) using Smith's mereotopology (Smith, 1996). However, in their axiomatization they assume conditions about shapes of the components that are not explicitly axiomatized by their ontology (Aameri and Gruninger, 2017). Demoly et al. (2012) and Gruhier et al. (2015) take a similar approach in axiomatizing assembly joints, but their ontologies are also incomplete as they make implicit assumptions about shapes and dimensionality of assembly components.

Incompleteness (with respect to intended models of the ontology) is in fact the common shortcoming of all existing assembly ontologies.

4. Motivating Use Case – Feasibility of Mechanical Assemblies

This section describes a motivating use case of the generative design of a suspension system. Again, the main purpose of ontologies in the current work is to define feasible designs of a particular suspension system to help eliminate non-feasible choices during search.

In essence, a mechanical assembly such as a suspension system can be thought as a set of components that are interconnected to each other and behaving together as a whole to achieve a certain function. One could consider the necessary condition of interconnectedness as the primary criterion for feasibility: at minimum the components must be interconnected in certain ways for the system to exist in a meaningful form and to be capable of realizing the desired function. Then, whether and how well the system realizes the desired function could be either a secondary consideration of feasibility or the main objective criterion for optimization.

Figures 2-4 show typical suspension systems. In all designs, two wheels are shown and each wheel is connected to a shock via a ball joint. Figure 2 is a beam axle design, where the two wheels are connected laterally by a single axle and therefore their movements are coupled. Figure 3 is a swing axle design, where each wheel is connected to an axle and the two axles are connected via a ball joint, allowing independent movements of each wheel. Figure 4 is a double wishbone design, where each wheel is connected to two wishbone arms and each wishbone arm is connected to the main chassis via a ball joint, also allowing independent movements of each wheel.

4.1. Descriptions of a Feasible Suspension System

The following set of statements related to the interconnection of components describe what constitute a feasible suspension system.

- Every component must be a *proper* part of the system.[2]

[2] *a* is a *proper* part of *b* if and only if *a* is a part of *b*, but *b* is not a part of *a*.

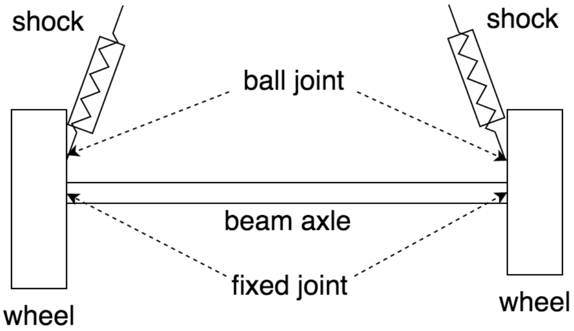

Fig. 2. A beam axle design.

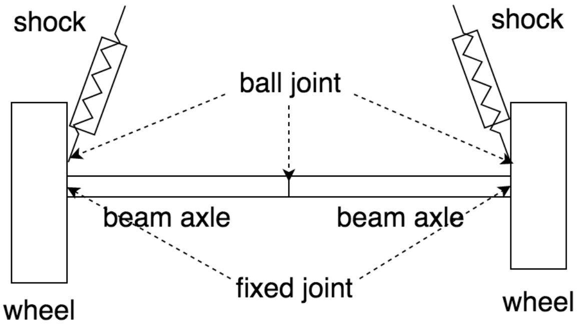

Fig. 3. A swing axle design.

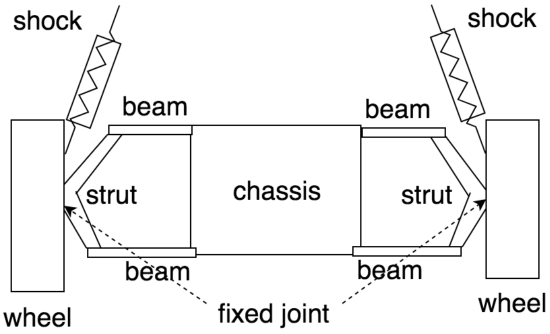

Fig. 4. A double wishbone design. All joints other than the specified ones are ball joints.

- Every component must be connected to all other components, either directly, via joints, or indirectly, via other components.
- Every structural component (all components other than wheels and shocks) must have at least two joints, in order to avoid having unnecessary “loose” components.
- Every joint must involve at least two components.

- Multiple components cannot occupy the same space except where they connect (i.e., at a joint).

Our goal is to use an ontology that captures the above statements as axioms to check the feasibility of a given design. A generative design problem could be posed as finding valid models of the axioms, guided by some search procedure that can find the optimal model with respect to chosen performance criteria.

4.2. Towards Qualitative Reasoning of Kinematics

Additional aspects of feasibility, related to the desired behavior and function of the system, can be defined based on the kinematics of the system. For example, one should expect that a working suspension system should have magnitudes of movements within certain ranges or that the system's degrees of freedom along certain directions are constrained.

As noted above, reasoning about kinematic behaviors requires representation of time, processes, and changes in position and orientation of entities. Such ontologies are currently being developed (Aameri and Gruninger, 2017) and there use in this context is part of future work.

Note, however, that ontologies for time, processes, and changes in position and orientation require the theories of mereotopology as the building block. For instance, reasoning about the relative movement between two components (e.g., a piston and a cylinder) requires determining the types of parthood or connection relations between the components (e.g., a cylindrical joint connection). Hence, the current work's contribution is an important and fundamental step toward the eventual goal.

4.3. Assumptions about Shapes and Joints

Besides the feasibility of the system, the following assumptions are made in representing suspension systems.

- All components are three-dimensional objects and have shapes.
- The shapes of components are idealized as geometric primitives (e.g., a shock is represented as a cylinder).
- Joints are idealized and can be represented with geometric entities such as points, lines, and surfaces.
- Two types of joints are considered: fixed (or welded) and ball (or spherical) joints. Fixed joints constrain the relative movements of paired components in all directions, both for translations and rotations. Ball joints constrain only the translations of paired components. Topologically, fixed joints and ball joints entail sharing of a common surface and a common point, respectively, between the two components involved.[3]

5. Ontological Choices for the Assembly Ontology

The key objective of the Assembly Ontology is to specify qualitative properties of mechanical assemblies such as the suspension system. In this section, we extract and delineate the semantic requirements

[3] We focus on representing the mereotopology of suspension systems, not relative movements, so different types of kinematic joints are not exhaustively defined. Other types of joints can be represented if the abstractions of fixed and ball joints are extended by a theory of relative orientation.

for the Assembly Ontology based on properties identified in the literature for representing such a product. To increase reuseability, we axiomatize the weakest theories that capture all the identified requirements. Section 7 demonstrates that these theories are sufficient for axiomatizing properties of suspension systems we described in Section 4. In Section 6, we discuss the possible extension of modules of the Assembly Ontology to stronger theories.

An assembly is a collection of components that are attached together by some sorts of mechanical joints (Popplestone et al., 1990; Gruhier et al., 2015). We base our ontological choices for mechanical assemblies on the following representation and reasoning requirements and challenges that have been identified in the literature.

- Part-whole relationships between an assembly and its components (Kim et al., 2008; Gruhier et al., 2015; Sanfilippo et al., 2016).
- Shapes and locations of, and spatial relations between, components (Popplestone et al., 1990).
- Shapes and morphological conditions of joints (Horváth et al., 1998).
- Relative position, orientation, and conceivable movement of components (e.g., degrees of freedom or the orientation of movement) (Horváth et al., 1998).
- Holes in surfaces (Sanfilippo et al., 2016). According to (Sanfilippo et al., 2016), there are two common treatments of holes: considering holes as entities on their own or as a quality of the objects that carry them. The former treatment requires a non-standard mereology to deal with the summation of material and immaterial entities, while the later needs a representation for shapes.
- Mating features (Papalambros and Wilde, 2000; Kim et al., 2006). When two components are joined, they do not make contact over their entire surface areas; rather, some geometric entities (i.e., faces, lines, or points) of the components are in contact. Notice that mating features have to be boundaries of components. That is, in addition to a three-dimensional representation of components that axiomatizes geometric entities and a connection relation, representing assembly joints requires a theory of boundaries.

In the present work we focus on axiomatizing part-whole and connection relationships; shape, relative position, and dimensionality of components; and the boundaries of components. Representation of the other features is part of future work.

Part-Whole and Connection Relationships. The most common way of formalizing connection and part-whole relations is to use mereotopologies (Kim et al., 2008; Gruhier et al., 2015; Sanfilippo et al., 2016). Given that different types of joints are abstracted as connection between mating features with different dimensionality (e.g., fixed and ball joints are respectively abstracted as connection between points and surfaces), and that the other modules of the ontology are required to be multi-dimensional, the Assembly Ontology needs to include a multi-dimensional mereotopology.

When developing a mereotopological theory for physical domains, we need to consider the fact that physical domains, and specifically assembly domains, are typically finite. Finiteness imposes two restrictions on the mereotopological theory appropriate for physical domains: first, the theory must not be atomless as such models are infinite; second, it must not be extensional since finite mereotopological configurations are not necessarily extensional (Aameri and Gruninger, 2017).[4] The existing multi-dimensional mereotopologies, such as Galton's Mereotopology (Galton, 2004), CODI (Hahmann, 2013)

[4] There are valid physical configurations that are *not* extensional, and so in representing a physical domain, the extensionality axiom should not be included but nor should its negation. Models of our theory can be either extensional and non-extensional.

and INCH (Gotts, 1996), are developed for the representation of spatial regions, and are too strong for representing spatial configuration between physical objects since they are either atomless or extensional (or both).

We therefore need to extend an existing equi-dimensional mereotopology to a multi-dimensional one. We choose the General Mereotopology (MT), the weakest theory among the mereotopologies presented by Casati and Varzi (1999). It has been shown by Gruninger and Aameri (2017) that MT has the same mereotopological properties as the RCC8 composition table (Randell et al., 1992). RCC8 has been successfully used in various physical domains for specifying spatial configurations of objects. It is also sufficient for stating all the mereotopological constraints for our use case.

Shape, Relative Position, and Dimensionality. For capturing three-dimensionality of assembly components and their shapes, we adopt and extend an ontology presented by Gruninger and Bouafoud (2011) for qualitative representation of three-dimensional shapes. The ontology is based on a subtheory of Hilbert's axiomatic theory of geometry (Hilbert, 1902) that deals with incidence and betweenness relations. The incident relation captures the relationship between entities with different dimensions, while the betweenness relation can be used to axiomatize relative position of components. In addition, the ontology is capable of axiomatizing holes in surfaces.

Shape Boundaries. There are two ontological viewpoints for the representation of boundaries. One considers a boundary as a region which has empty interior; that is, boundaries are *not* considered as lower-dimensional entities (Smith, 1996). The alternative approach, adopted by General Formal Ontology (GFO) (Baumann et al., 2016) and CODIB (Hahmann, 2013), considers a boundary as a lower-dimensional entity which is part of the bounded entity. The adoption of a multi-dimensional representation in order to represent joints appropriately, among other reasons (see above), requires also adopting this second view.

5.1. Ontological Commitments

The key ontological commitments we consider in axiomatizing the Assembly Ontology are the following. **(Req 1)** and **(Req 4)** to **(Req 6)** are adopted from Hilbert (1902) and Gruninger and Bouafoud (2011).

(Req 1) There are four dimensions.
For each dimension, there is a primitive predicate in the language of the ontology representing zero-, one-, two-, and three-dimensional objects.

(Req 2) A collection of three-dimensional entities is a physical entity.
There is a primitive predicate in the language of the ontology representing collections of three-dimensional objects.

(Req 3) Each physical entity is of exactly one of the four dimensions, or is a collection of three-dimensional objects.

(Req 4) Individuals with different dimensions are only related by incidence relations.
Each class of objects is mereotopologically independent of the other classes and there is no betweenness relationships between objects with different dimensions.

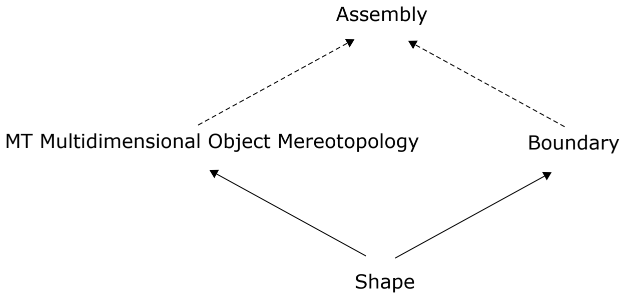

Fig. 5. Modules of the Assembly Ontology. Solid lines denote conservative extension and dashed lines denote nonconserative extension.

(Req 5) Zero- and one-dimensional entities do not exist independently of two-dimensional entities; two-dimensional entities do not exist independently of three-dimensional entities.

(Req 6) Relative positions of equi-dimensional individuals are captured by betweenness relations.

(Req 7) Spatial relationships between equi-dimensional individuals are captured by mereological and/or topological relations.

(Req 8) A boundary always bounds an entity with a higher dimension.
However, unlike Baumann et al. (2016), the dimension of an entity does not have to be exactly one dimension higher than its boundary. For example, a surface can have a boundary point and no boundary lines.

(Req 9) Every three-dimensional entity has at least one boundary, where the dimension of the boundary is exactly two-dimensional.

6. The Assembly Ontology

In this section we present the axioms of the Assembly Ontology. Considering the requirements discussed in Section 5, the ontology is designed to consist of three primary modules (Figure 5):

- *The Shape Ontology* based on the CardWorld and BoxWorld Ontologies (Gruninger and Bouafoud, 2011) which itself is based on Hilbert's axiomatic theory of geometry (Hilbert, 1902). The ontology specifies properties and relationships among five disjoint categories of entities that represent zero-, one-, two-, three-, and four-dimensional objects (four-dimensional objects are collections of three-dimensional objects).
- *The MT Multidimensional Object Mereotopology (MT MOM)* which captures mereological and topological relationships between individuals with zero-, one-, two-, and three-dimensions.
- *The Boundary Ontology* which extends the Shape Ontology with axioms that describe properties of physical boundaries.

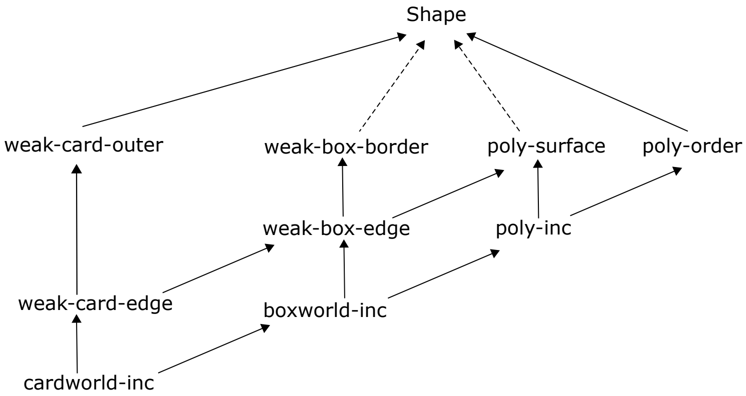

Fig. 6. Modules of the Shape Ontology. Solid lines denote conservative extension and dashed lines denote nonconservative extension.

6.1. The Shape Ontology

The CardWorld and BoxWorld ontologies axiomatize properties of a stand-alone two-dimensional and three-dimensional object, respectively. We extend these modules by another module called the PolyWorld, which specifies relationships between sets of three-dimensional objects. In axiomatizing PolyWorld, we relax and modify some of the assumptions originally made in developing CardWorld and BoxWorld.

Hilbert's theory consists of three subtheories: the first one axiomatizes properties of the incidence relation; the second one is a theory of betweenness; and the third one describes congruence relationships. The focus of the CardWorld and BoxWorld is on the incidence and betweenness relations, ignoring geometrical notions such as the length and relative alignment of lines or the curvature and areas of surfaces. The important advantage of the CardWorld and BoxWorld is that combining it with a multi-dimensional mereotopology is straightforward since it includes primitives for zero-, one-, two-, and three-dimensional objects.

Within the domain of a model of the Shape Ontology there are five disjoint categories, namely point, edge, surface, box, and poly, where a poly is a set of boxes, a box is set of surfaces, a surface is a set of edges, and an edge is a set of points.[5] Zero-, one-, two-, and three-dimensional objects respectively correspond with points, edges, surfaces, and boxes. The signature of the Shape Ontology includes a binary predicate, *in*, that captures the incidence relations between different categories of objects.

The axioms of the Shape Ontology are decomposed into nine modules, where three of them specify properties of two-dimensional shapes (*cardworld_inc*, *weak_card_outer*, and *weak_card_edge*), three specify properties of three-dimensional shapes (*boxworld_inc*, *weak_box_edge*, and *boxworld_border*), and three state properties of collections of three-dimensional shapes (*poly_inc*, *poly_surface*, and *poly_order*). The meta-logical relationships between these modules are shown in Figure 6. We first present the axioms of the modules corresponding with two-dimensional shapes and then discuss axioms representing three-dimensional shapes and their collections.

[5] The PhysicalWorld Ontology contains two other sort primitives, *physical_body* and *region*, representing physical objects and abstract space, respectively. The five dimension classes in the Shape Ontology are subcategories of *physical_body*.

6.1.1. Axiomatizing Two-Dimensional Shapes

$T_{cardworld\_inc}$ is the root theory of the CardWorld and guarantees disjointness between points, edges, and surfaces. The axioms of $T_{cardworld\_inc}$ also guarantee that *in* is an incidence relation; that is, it is reflexive, symmetric, and does not relate two distinct elements that are of the same sort.

Definition 1. *$T_{cardworld\_inc}$ is the theory axiomatized by the following sentences:*

$$(\forall x)\, point(x) \supset \neg edge(x) \wedge \neg surface(x). \tag{1}$$

$$(\forall x)\, edge(x) \supset \neg surface(x). \tag{2}$$

$$(\forall x,y)\, in(x,y) \supset in(y,x). \tag{3}$$

$$(\forall x)\, in(x,x). \tag{4}$$

$$(\forall x,y)\, in(x,y) \wedge point(x) \wedge point(y) \supset (x = y). \tag{5}$$

$$(\forall x,y)\, in(x,y) \wedge edge(x) \wedge edge(y) \supset (x = y). \tag{6}$$

$$(\forall x,y)\, in(x,y) \wedge surface(x) \wedge surface(y) \supset (x = y). \tag{7}$$

$T_{weak\_card\_edge}$ specifies properties of edges in a two-dimensional shape by stating that each point or edge is part of some surface (Axiom (8)) and an edge contains at most two vertices (Axiom (9)), where a vertex is a point where two edges meet (Axiom (11)).

Definition 2. *$T_{weak\_card\_edge}$ is the extension of $T_{cardworld\_inc}$ with the following sentences:*

$$(\forall x)\, (point(x) \vee edge(x)) \supset (\exists s)\, surface(s) \wedge in(x,s). \tag{8}$$

$$\begin{aligned}&(\forall e,v_1,v_2,v_3)\, edge(e) \wedge vertex(v_1) \wedge vertex(v_2) \wedge vertex(v_3) \wedge in(v_1,e) \wedge in(v_2,e) \wedge in(v_3,e)\\ &\supset (v_1 = v_2) \vee (v_1 = v_3) \vee (v_2 = v_3).\end{aligned} \tag{9}$$

$$\begin{aligned}&(\forall v,e_1,e_2)\, meet(e_1,e_2,v) \equiv\\ &(edge(e_1) \wedge edge(e_2) \wedge point(v) \wedge \neg(e_1 = e_2) \wedge part(v,e_1) \wedge part(v,e_2)).\end{aligned} \tag{10}$$

$$(\forall v)\, vertex(v) \equiv (\exists e_1,e_2)\, meet(e_1,e_2,v). \tag{11}$$

$T_{weak\_card\_edge}$ is a subtheory of the module *card_edge* of the CardWorld. The followings are the axioms of *card_edge* that are not in $T_{weak\_card\_edge}$, and the reason they are not included:

(1) **Each point is part of some edge.**
This axiom prevents the possibility of a point incident with a surface, without being part of an edge. Such a possibility is required for example for defining joints with only a single common point. Having points independent of edges may also be used to represent the location of center-of-mass on a three-dimensional object in future work.

(2) **Each surface contains an edge.**
A sphere has one surface and no edges.

(3) **Every edge contains at least two vertices.**
This axiom eliminates circular shapes as they only have one edge and no vertex. The base of a cylinder is for example circular, and in Section 7 we see that shocks are abstracted as cylinders.

$T_{weak\_card\_outer}$ is the theory obtained by extending $T_{weak\_card\_edge}$ with the module *cardworld_outer* of the CardWorld and is intended to represent holes within a surface. The edges in a surface can be abstracted as disjoint cyclic orderings. Intuitively, one of these orderings form the outer edges of the surface, while the others represent holes within the surface. *cardworld_outer* enforces that for every surface, there exists a unique set of outer edges that form a cyclic ordering. Axioms of *cardworld_outer* can be found in Gruninger and Bouafoud (2011).

6.1.2. Axiomatizing Three-Dimensional Shapes

$T_{boxworld\_inc}$ is the root theory of the BoxWorld, specifying the disjointness of boxes from the lower-dimensional entities in $T_{cardworld\_inc}$.

Definition 3. *$T_{boxworld\_inc}$ is the theory obtained by extending $T_{cardworld\_inc}$ with the following sentences:*

$$(\forall x)\, point(x) \lor edge(x) \lor surface(x) \supset \neg box(x). \tag{12}$$

$$(\forall x,y)\, in(x,y) \land box(x) \land box(y) \supset (x = y). \tag{13}$$

$T_{weak\_box\_edge}$ axiomatizes properties of edges of a box. Axioms (14) and (15) respectively define a ridge as an edge that is part of two surfaces and a border as an edge that is part of a unique surface. Axiom (16) states that a surface that is part of a box containing other surfaces also contains a ridge. If three distinct edges meet at the same vertex, then they cannot all be part of the same surface (Axiom (17)).

Definition 4. *$T_{weak\_box\_edge}$ is the extension of $T_{weak\_card\_edge} \cup T_{boxworld\_inc}$ with the following sentences:*

$$\begin{aligned}&(\forall e)\, ridge(e) \equiv \\ &(\exists s_1,s_2) edge(e) \land surface(s_1) \land surface(s_2) \land \neg(s_1 = s_2) \land in(e,s_1) \land in(e,s_2).\end{aligned} \tag{14}$$

$$(\forall e)\, border(e) \equiv edge(e) \land \neg ridge(e). \tag{15}$$

$$\begin{aligned}&(\forall x,s_1,s_2) box(x) \land surface(s_1) \land surface(s_2) \land \neg(s_1 = s_2) \land in(s_1,x) \land in(s_2,x) \supset \\ &(\exists e) ridge(e) \land in(e,s_1).\end{aligned} \tag{16}$$

$$\begin{aligned}&(\forall e_1,e_2,e_3,v,s)\, edge(e_1) \land edge(e_2) \land edge(e_3) \land \neg(e_1 = e_2) \land \neg(e_1 = e_3) \land \neg(e_2 = e_3) \land vertex(v) \\ &\land in(e_1,s) \land in(e_2,s) \land in(v,e_2) \land meet(e_1,e_2,v) \land meet(e_1,e_3,v) \supset \neg in(e_3,s).\end{aligned} \tag{17}$$

$T_{weak\_box\_edge}$ is a subtheory of the module *box_edge* of the BoxWorld. The following are the axioms of *box_edge* that are not in $T_{weak\_box\_edge}$, together with counter-examples that show why these axioms are not included.

(1) **An edge is part of at most two surfaces.**
Consider multiple sheets that share a single edge (like pages of a notebook, or turbine blades, idealized as sheets, that are connect to a shaft, idealized as a single edge). Then the common edge is part of more than two surfaces.

(2) **Every edge in a surface meets another distinct edge in that surface.**
An edge representing a circular hole on a surface does not meet any other edges.

(3) **Every border edge meets two distinct border edges** and **every border meets another unique border at a vertex.**
The border edge of a circular bowl, for example, does not meet any other border edges since a circular shape only has one edge.

$T_{weak\_box\_border}$ is the theory obtained by extending $T_{weak\_box\_edge}$ with the module *boxworld_border* of the BoxWorld and axiomatizes properties of border edges (i.e., edges that are part of exactly one surface) in a box. Axioms of *boxworld_border* state that if a box contains border edges, then they form a cyclic ordering.

6.1.3. Axiomatizing Relations between Three-Dimensional Shapes

T_{poly_inc} introduces a sort primitive, *poly*, and guarantees that polies are disjoint from other dimensional sorts in the ontology.

Definition 5. *T_{poly_inc} is the theory obtained by extending $T_{cardworld\_inc} \cup T_{boxworld\_inc}$ with with the following sentences:*

$$(\forall x)\, point(x) \vee edge(x) \vee surface(x) \vee box(x) \supset \neg poly(x). \tag{18}$$

$$(\forall x,y)\, in(x,y) \wedge poly(x) \wedge poly(y) \supset (x = y). \tag{19}$$

$T_{poly\_surface}$ axiomatizes the relationship between a box and its surfaces. Axioms (20) and (21) state that a surface does not exist independently of a box and is incident with at most two boxes (Axiom (21)). Axiom (22) guarantees that a box has at least one surface.

Definition 6. *$T_{poly\_surface}$ is the extension of $T_{poly\_inc} \cup T_{weak\_box\_edge}$ with the following sentences:*

$$(\forall s)\, surface(s) \supset (\exists b)\, box(b) \wedge in(s,b). \tag{20}$$

$$(\forall s,x_1,x_2,x_3) surface(s) \wedge box(x_1) \wedge box(x_2) \wedge box(x_3) \wedge$$
$$in(s,x_1) \wedge in(s,x_2) \wedge in(s,x_3) \supset (x_1 = x_2) \vee (x_2 = x_3) \vee (x_1 = x_3). \tag{21}$$

$$(\forall b)\, box(b) \supset (\exists s)\, surface(s) \wedge in(s,b). \tag{22}$$

T_{poly_order} define a betweenness relation over the set of boxes in a poly.

Definition 7. *T_{poly_order} is the extension of $T_{boxworld\_inc}$ with the following sentences:*

$$(\forall b_1,b_2,b_3)\, xbetween(b_1,b_2,b_3) \supset box(b_1) \wedge box(b_2) \wedge box(b_3). \tag{23}$$

$$(\forall b_1,b_2)\, xbetween(b_1,b_2,b_1) \supset (b_1 = b_2). \tag{24}$$

$$(\forall b_1,b_2,b_3,b_4)\, xbetween(b_1,b_2,b_3) \wedge \neg(b_2 = b_3) \wedge xbetween(b_2,b_3,b_4)$$
$$\supset xbetween(b_1,b_2,b_4). \tag{25}$$

$$(\forall b_1,b_2,b_3,b_4)\, xbetween(b_1,b_2,b_4) \wedge xbetween(b_2,b_3,b_4) \supset xbetween(b_1,b_2,b_3). \tag{26}$$

$$(\forall b_1,b_2,b_3)\, xbetween(b_1,b_2,b_3) \supset xbetween(b_3,b_2,b_1). \tag{27}$$

6.1.4. Axioms of the Shape Ontology

$T_{weak\_card\_edge}$ and $T_{poly\_surface}$ enforce that every point is part of some edge or surface, every edge is part of a surface, and every surface is part of some box. The axioms in T_{shape} strengthen this condition so that every box is part of a poly and that each point, edge, or surface or box is contained in a unique poly. Moreover, every poly must contain at least one box.

Definition 8. *T_{shape} is the extension of $T_{weak\_card\_outer} \cup T_{boxworld\_border} \cup T_{poly\_surface} \cup T_{poly\_order}$ with the following sentences:*

$$(\forall p)\, poly(p) \supset (\exists b)\, box(b) \land in(b,p). \tag{28}$$

$$(\forall b)\, box(b) \supset (\exists p)\, poly(p) \land in(b,p). \tag{29}$$

$$(\forall x_1,x_2,y)\, in(y,x_1) \land in(y,x_2) \land poly(x_1) \land poly(x_2) \land$$
$$(point(y) \lor edge(y) \lor surface(y) \lor box(y))) \supset (x_1 = x_2))) \tag{30}$$

6.2. The MT Multidimensional Object Mereotopology

The MT Multidimensional Object Mereotopology (MT MOM) extends the language of the Shape Ontology with two parthood predicates: *surface_part*, and *box_part* corresponding with surfaces and boxes; and four connection predicates: *point_C*, *edge_C*, *surface_C*, and *box_C*, respectively corresponding with points, edges, surfaces, and boxes. Mereotopological relationships between surfaces and boxes are axiomatized by the Ground Mereotopology (MT) (Casati and Varzi, 1999), while topological relations between points and edges are specified using the topological subtheory of MT. Individuals in different categories, however, are not related by topological or mereological relations.

Axioms (31) to (33) define *point_C* as a reflexive and symmetric relation over *point*, capturing connection relations between points.

Definition 9. *T_{point_mt} is the theory axiomatized by the following sentences:*

$$(\forall x,y)\, point\_C(x,y) \supset point(x) \land point(y). \tag{31}$$

$$(\forall x)\, point(x) \supset point\_C(x,x). \tag{32}$$

$$(\forall x,y)\, point\_C(x,y) \supset point\_C(y,x). \tag{33}$$

Axioms (34) to (36) define *edge_C* as a reflexive and symmetric relation over *edge*, capturing connection relations between edges.

Definition 10. *T_{edge_mt} is the theory axiomatized by the following sentences:*

$$(\forall x,y)\, edge\_C(x,y) \supset edge(x) \land edge(y). \tag{34}$$

$$(\forall x)\, edge(x) \supset edge\_C(x,x). \tag{35}$$

$$(\forall x,y)\, edge\_C(x,y) \supset edge\_C(y,x). \tag{36}$$

Axioms (37) to (44) axiomatize the mereotopology of surfaces.

Definition 11. *$T_{surface\_mt}$ is the theory axiomatized by the following sentences:*

$$(\forall x,y)\, surface\_part(x,y) \supset surface(x) \wedge surface(y). \tag{37}$$

$$(\forall x)\, surface(x) \supset surface\_part(x,x). \tag{38}$$

$$(\forall x,y)\, surface\_part(x,y) \wedge surface\_part(y,x) \supset (x = y). \tag{39}$$

$$(\forall x,y,z)\, surface\_part(x,y) \wedge surface\_part(y,z) \supset surface\_part(x,z). \tag{40}$$

$$(\forall x)\, surface(x) \supset surface\_C(x,x). \tag{41}$$

$$(\forall x,y)\, surface\_C(x,y) \supset surface\_C(y,x). \tag{42}$$

$$(\forall x,y)\, surface\_C(x,y) \supset surface(x) \wedge surface(y). \tag{43}$$

$$surface\_part(y,z) \wedge surface\_C(x,y) \supset surface\_C(x,z). \tag{44}$$

$surface\_part$ is defined over surfaces (Axiom (37)), and is a reflexive, antisymmetric, and transitive relation (Axioms (38) to (40)), while $surface\_C$ is a reflexive and symmetric relation over surfaces (Axioms (41) to (43)). Moreover, when a surface x is connected to another y, then x must be connected to all surfaces which y is part of (Axiom (44)).

Axioms (45) to (52) axiomatize the mereotopology of boxes.

Definition 12. *T_{box_mt} is the theory axiomatized by the following sentences:*

$$(\forall x,y)\, box\_part(x,y) \supset box(x) \wedge box(y). \tag{45}$$

$$(\forall x)\, box(x) \supset box\_part(x,x). \tag{46}$$

$$(\forall x,y)\, box\_part(x,y) \wedge box\_part(y,x) \supset (x = y). \tag{47}$$

$$(\forall x,y,z)\, box\_part(x,y) \wedge box\_part(y,z) \supset box\_part(x,z). \tag{48}$$

$$(\forall x,y)\, box\_C(x,y) \supset box(x) \wedge box(y). \tag{49}$$

$$(\forall x)\, box(x) \supset box\_C(x,x). \tag{50}$$

$$(\forall x,y)\, box\_C(x,y) \supset box\_C(y,x). \tag{51}$$

$$box\_part(y,z) \wedge box\_C(x,y) \supset box\_C(x,z). \tag{52}$$

box_part and box_C have the same properties as $surface\_part$ and $surface\_C$ except that they are defined over boxes.

T_{mt_mom} specifies the relationship between an object with the lower-dimensional entities of its parts and connected components.

Definition 13. *T_{mt_mom} is the extension of $T_{shape} \cup T_{point\_mt} \cup T_{edge\_mt} \cup T_{surface\_mt} \cup T_{box\_mt}$ with the following sentences:*

$$(\forall x,s_1,s_2)\, surface\_part(s_1,s_2) \wedge in(s_2,x) \wedge (box(x) \vee poly(x)) \supset in(s_1,x). \tag{53}$$

$$(\forall x,b_1,b_2)\, box\_part(b_1,b_2) \wedge in(b_2,x) \wedge poly(x) \supset in(b_1,x). \tag{54}$$

$$(\forall e_1,e_2)\, edge(e_1) \wedge edge(e_2) \wedge (\exists x) in(x,e_1) \wedge in(x,e_2) \wedge point(x) \supset edge\_C(e_1,e_2). \quad (55)$$

$$(\forall s_1,s_2)\, surface(s_1) \wedge surface(s_2) \wedge (\exists x) in(x,s_1) \wedge in(x,s_2) \wedge (edge(x) \vee point(x))$$
$$\supset surface\_C(s_1,s_2). \quad (56)$$

$$(\forall b_1,b_2)\, box(b_1) \wedge box(b_2) \wedge (\exists x) in(x,b_1) \wedge in(x,b_2) \wedge (surface(x) \vee edge(x) \vee point(x))$$
$$\supset box\_C(b_1,b_2). \quad (57)$$

$$(\forall e_1,e_2)\, edge\_C(e_1,e_2) \wedge \neg edge\_part(e_1,e_2) \wedge \neg edge\_part(e_2,e_1)$$
$$\supset (\exists p_1,p_2) point(p_1) \wedge point(p_2) \wedge in(p_1,e_1) \wedge in(p_2,e_2) \wedge point\_C(p_1,p_2). \quad (58)$$

$$(\forall s_1,s_2)\, surface\_C(s_1,s_2) \wedge \neg surface\_part(s_1,s_2) \wedge \neg surface\_part(s_2,s_1)$$
$$\supset (\exists x_1,x_2)(point(x_1) \wedge point(x_2) \wedge in(x_1,s_1) \wedge in(x_2,s_2) \wedge point\_C(x_1,x_2))$$
$$\vee (edge(x_1) \wedge edge(x_2) \wedge in(x_1,s_1) \wedge in(x_2,s_2) \wedge edge\_C(x_1,x_2)). \quad (59)$$

$$(\forall b_1,b_2)\, box\_C(b_1,b_2) \wedge \neg box\_part(b_1,b_2) \wedge \neg box\_part(b_2,b_1)$$
$$\supset (\exists x_1,x_2)(point(x_1) \wedge point(x_2) \wedge in(x_1,b_1) \wedge in(x_2,b_2) \wedge point\_C(x_1,x_2))$$
$$\vee (edge(x_1) \wedge edge(x_2) \wedge in(x_1,b_1) \wedge in(x_2,b_2) \wedge edge\_C(x_1,x_2))$$
$$\vee (surface(x_1) \wedge surface(x_2) \wedge in(x_1,b_1) \wedge in(x_2,b_2) \wedge surface\_C(x_1,x_2)). \quad (60)$$

Axioms (53) and (54) indicate that parts of a lower-dimensional entity of an object are also incident with that object. Axioms (55) to (57) state that if two objects share a common dimensional entity, then they are connected. Axioms (58) to (60) state that if two objects are connected, but are not part of each other, then each contain a lower-dimensional entity such that those entities are connected.

6.3. The Boundary Ontology

The Boundary Ontology is a conservative extension of the Shape Ontology that extends its language with binary predicates: $point\_bound(x,y)$, $edge\_bound(x,y)$, and $surface\_bound(x,y)$ respectively denoting that x is a point boundary, an edge boundary, or a surface boundary of y.

Definition 14. *$T_{boundary}$ is the extension of T_{shape} with the following sentences:*

$$(\forall x,y)\, surface\_bound(x,y) \supset in(x,y) \wedge surface(x) \wedge box(y). \quad (61)$$

$$(\forall x,y)\, edge\_bound(x,y) \supset in(x,y) \wedge edge(x) \wedge (surface(y) \vee box(y)). \quad (62)$$

$$(\forall x,y)\, point\_bound(x,y) \supset in(x,y) \wedge point(x) \wedge (edge(y) \vee surface(y) \vee box(y)). \quad (63)$$

$$(\forall x,y,z)\, surface\_bound(x,y) \wedge in(z,x) \wedge (edge(z) \vee point(z)) \supset$$
$$(edge\_bound(z,y) \vee point\_bound(z,y)). \quad (64)$$

$$(\forall x,y,z)\, edge\_bound(x,y) \wedge in(z,x) \wedge point(z) \supset point\_bound(z,y). \quad (65)$$

$$(\forall b)\, box(b) \supset (\exists s) surface\_bound(s,b). \quad (66)$$

Axioms (61) to (63) state that a boundary always bounds an entity with a higher dimension ((**Req 8**)). Axioms (64) and (65) state that lower-dimensional entities of a boundary entity are also boundaries. Axiom (66) captures **(Req 9)**.

6.4. Axioms of the Assembly Ontology

The Assembly Ontology extends MT MOM and the Boundary Ontology with statements about mereotopological relations between boundary entities.

Definition 15. *$T_{assembly}$ is the extension of $T_{mt\_mom} \cup T_{boundary}$ with the following sentences:*

$$(\forall b_1, b_2, x_1, x_2)(point\_C(x_1, x_2) \lor edge\_C(x_1, x_2) \lor surface\_C(x_1, x_2)) \land in(x_1, b_1) \land in(x_2, b_2)$$
$$\land box(b_1) \land box(b_2) \land \neg(x_1 = x_2) \land \neg(b_1 = b_2) \supset (point\_bound(x_1) \land point\_bound(x_2)) \lor$$
$$(edge\_bound(x_1) \land edge\_bound(x_2)) \lor (surface\_bound(x_1) \land surface\_bound(x_2)). \tag{67}$$
$$(\forall s_1, s_2)\, surface\_part(s_1, s_2) \land surface\_bound(s_2) \supset surface\_bound(s_1). \tag{68}$$

Axiom (67) enforces that only boundary entities of a box can be connected to entities of another box, while Axiom (68) states that surface parts of a boundary surface are also boundaries.

6.5. Discussion

The axiomatization we presented for the Assembly Ontology provides the weakest theory that is required for representing mechanical assemblies. Depending on domain-specific requirements, stronger theories may be needed. The modular design of the Assembly Ontology, as well as the separate, disjoint relations for each entity sort, enable extension of a module of the ontology with minimal effects on other modules. For example, in some application domains it is required that every pair of assembled components/subassemblies be represented as a distinct entity. In that case, the mereotopology of the three-dimensional objects must include an axiom stating that a mereotopological sum exists for every pair of three-dimensional entities. Considering that in the Assembly Ontology the mereotopology of boxes is independent of other entities, the only theory that has to be extended is T_{box_mt}.

Another important feature of the Assembly Ontology is the inclusion of elements of the class *poly* as collections of three-dimensional objects. Specifying an assembly product (such as a suspension system) as a *poly*, instead of a *box*, provides two important representational advantages: first, it enables distinguishing between subsystems and subassemblies of a physical product; second, it provides a way for describing objects that have the same mereotopological structure, but a different relative configuration of their components.

7. Applying the Assembly Ontology

In this section we discuss the application of the Assembly Ontology to the representation of the suspension systems described in Section 4.

In a beam axle suspension system (Figure 2), the system consists of two wheels, two shocks, and a beam. These components are connected to each other by fixed or ball joints. Before we specify the description of beam axle designs, we discuss axiomatization of properties of these components and the two types of joints, using the Shape Ontology.

7.1. Axiomatic Description of Components

The wheels have a torus shape, the shocks are cylindrical, and the beam is a cube. A torus is a single surface with a ring shape. In the language of the Shape Ontology, this is equivalent to say that a torus is a *box* that has exactly one boundary surface, and no edges or vertices:

$$(\forall x)\, torus(x) \equiv (box(x) \wedge (\exists s)\, bound\_surface(s) \wedge in(s,x) \wedge (\forall s_1) bound\_surface(s_1) \wedge in(s_1,x) \supset (s_1 = s)).$$

A cylinder can be defined as a *box* with exactly three boundary surfaces, where two of the surfaces have a circular shape, and the third one shares a common edge with each of the other two surfaces. A circular shape is a surface which has exactly one boundary edge:

$$(\forall s)\, circular(s) \equiv (surface(s) \wedge (\exists e)\, bound\_edge(e) \wedge in(e,s) \wedge (\forall e_1) bound\_edge(e_1) \wedge in(e_1,s) \supset (e_1 = e)).$$

Similarly, a cube can be defined as a box with exactly eight quadrilateral boundary surfaces and at least twelve boundary edges, where a quadrilateral shape is defined as a surface with four boundary edges and four vertices.

Having the axiomatic definitions of the shapes of the components, we can state the following:

$$(\forall x)\, beam(x) \supset cube(x).$$
$$(\forall x)\, wheel(x) \supset cylinder(x).$$
$$(\forall x)\, shock(x) \supset torus(x).$$

Additionally, we state that beams, wheels, and shocks are disjoint classes of components:

$$(\forall x)\, beam(x) \supset \neg wheel(x) \wedge \neg shock(x).$$
$$(\forall x)\, wheel(x) \supset \neg shock(x).$$

7.2. Axiomatic Description of Joints

As we explained in Section 4, we idealize a fixed joint as a shared surface between two components, and a ball joint as a shared point:

$$(\forall x,y)\, fixed\_joined(x,y) \equiv (box(x) \wedge box(y) \wedge (\exists z) surface(z) \wedge in(z,x) \wedge in(z,y)).$$
$$(\forall x,y)\, ball\_joined(x,y) \equiv (box(x) \wedge box(y) \wedge (\exists z) point(z) \wedge in(z,x) \wedge in(z,y)).$$

7.3. Axiomatic Description of Suspension Systems

We consider a beam axle suspension system as a *poly* that is incident with an element *a*, where two wheels, two shocks, and a beam are proper parts of *a*. Notice that in this case T_{mt_mdm} implies that the wheels, the shocks, and the beam are also incident with the suspension system. The beam and the shocks are connected to the wheels respectively by fixed and ball joints. Moreover, the shocks and the beam are between the two wheels:

$$(\forall x)\, beam\_axle(x) \supset poly(x) \wedge (\exists a, w_1, w_2, s_1, s_2, b) beam(b) \wedge wheel(w_1) \wedge wheel(w_2)$$
$$\wedge shock(s_1) \wedge shock(s_2) \wedge \neg(s_1 = s_2) \wedge \neg(w_1 = w_2)$$
$$\wedge box\_PP(b,a) \wedge box\_PP(w_1,a) \wedge box\_PP(w_2,a) \wedge box\_PP(s_1,a) \wedge box\_PP(s_2,a)$$
$$\wedge fixed\_joined(w_1,b) \wedge fixed\_joined(w_2,b) \wedge ball\_joined(w_1,s_1) \wedge ball\_joined(w_2,s_2)$$
$$\wedge xbetween(w_1,s_1,w_2) \wedge xbetween(w_1,s_2,w_2) \wedge xbetween(w_1,b,w_2) \wedge in(a,x).$$

where $box\_PP(x,y)$ denotes that x is a proper part of y and is defined by the following sentence:

$$(\forall x,y)\, box\_PP(x,y) \equiv box\_part(x,y) \wedge \neg box\_part(y,x).$$

Note that specifying a suspension system as a *poly*, instead of a *box*, enables us to represent it as a subsystem, rather than a subassembly, of a larger system.

The swing axle design can be described using a similar axiomatization, except that instead of one beam, there are two beams that are incident with the suspension system and are connected together by a ball joints (see Appendix A.1). Similarly, a double wishbone design is a *poly* with a similar description as *beam_axle*, except that it is incident with four beams, two struts, and a chassis, where the struts are idealized as cubes, and the chassis is a quadrilateral surface (see Appendix A.2).

To relate the presented ontology to the existing work and show that its capability of representing various design entities, we have provide axiomatic definitions of of assembly joining methods (presented by Kim et al. (2008)) in Section A.3.

8. Conclusion

We presented a first-order ontology, called the Assembly Ontology, for specifying properties of mechanical assemblies. The primary application for this ontology is to express qualitative design constraints for a generative design tool.

The Assembly Ontology contains three modules: the Shape Ontology, the MT Multidimensional Object Mereotopology (MT MOM), and the Boundary Ontology. The Shape Ontology is a multi-dimensional theory of shapes, developed based on the Hilbert's axiomatic theory of geometry, and is capable of representing shapes of objects with various dimensionality. The MT MOM extends the Shape Ontology to a multi-dimensional mereotopology capturing parthood and connection relations between equi-dimensional objects. The Boundary Ontology axiomatizes relations of boundary entities of an objects.

We demonstrated the application of the Assembly Ontology to specifying qualitative properties of suspension systems, as well as in defining different types of assembly joints.

In addition to subassemblies, the Assembly Ontology provides formalisms for representing subsystems of products and for formal description of assemblies with the same mereotopological structures but different relative configuration of components. The ontology provides an explicit axiomatization of its intended requirements, where the requirements are extracted based on the existing assembly formalizations, and with consideration of real-world manufacturing products such as suspension systems. The ontology is highly reusable as it modular design and use of sorted relations specific to particular classes of entities enable extension of a module of the ontology with minimal effect on the other modules.

There are three directions for future work:

- A formal verification of the presented ontology is required in order to demonstrate the correctness and completeness of the ontology with respect to its intended models (Aameri and Gruninger, 2015). Such a verification includes specifying the semantic requirements of the ontology by standard mathematical structures, characterizing the models of the ontology up to isomorphism, and showing that the models of the ontology are equivalent to the intended structures for the ontology.
- The axiomatic representation of dynamic behavior of mechanical assemblies needs axiomatic theories of kinematics concepts as well as ontologies for relative orientation. Development of such theories is part of future work.
- The ultimate goal of this work is to use ontologies in pruning the search space for a generative design tool by generating *all* structures, with a specific finite cardinality, that satisfy the axioms of the ontology. The existing automated theorem provers, however, are designed for consistency checking, and are not efficient in generating all possible models of a theory. Part of the future work is, therefore, to develop algorithms for automated construction of models of the Assembly Ontology.

References

Aameri, B. and Gruninger, M. (2015). A new look at ontology correctness. In *2015 AAAI Spring Symposium Series. AAAI Press*.

Aameri, B. and Gruninger, M. (2017). Encountering the physical world. In *Proceedings of the Joint Ontology Workshops 2017*.

Antonsson, E. K. and Cagan, J. (2005). *Formal engineering design synthesis*. Cambridge University Press.

Barbau, R., Krima, S., Rachuri, S., Narayanan, A., Fiorentini, X., Foufou, S., and Sriram, R. (2012). Ontostep: Enriching product model data using ontologies. *Computer-Aided Design*, 44(6):575–590.

Baumann, R., Loebe, F., and Herre, H. (2016). Towards an ontology of space for gfo. In *FOIS*, pages 53–66.

Borgo, S. and Leitão, P. (2007). Foundations for a core ontology of manufacturing. In *Ontologies*, pages 751–775. Springer.

Borst, P., Akkermans, H., and Top, J. (1997). Engineering ontologies. *International Journal of Human-Computer Studies*, 46(2-3):365–406.

Brown, D. C. (1998). Defining configuring. *AI EDAM*, 12(4):301–305.

Casati, R. and Varzi, A. (1999). *Parts and places: The structures of spatial representation*. MIT Press.

Chakrabarti, A., Shea, K., Stone, R., Cagan, J., Campbell, M., Hernandez, N. V., and Wood, K. L. (2011). Computer-based design synthesis research: an overview. *Journal of Computing and Information Science in Engineering*, 11(2):021003.

Dartigues, C., Ghodous, P., Gruninger, M., Pallez, D., and Sriram, R. (2007). Cad/capp integration using feature ontology. *Concurrent Engineering*, 15(2):237–249.

Deb, K. and Goyal, M. (1996). A combined genetic adaptive search (geneas) for engineering design. *Computer Science and informatics*, 26:30–45.

Demoly, F., Matsokis, A., and Kiritsis, D. (2012). A mereotopological product relationship description approach for assembly oriented design. *Robotics and Computer-Integrated Manufacturing*, 28(6):681–693.

Fenves, S. (2001). *A core product model for representing design information*. US Department of Commerce, Technology Administration, National Institute of Standards and Technology.

Fiorentini, X., Gambino, I., Liang, V., Foufou, S., Rachuri, S., Bock, C., and Mahesh, M. (2007). Towards an ontology for open assembly model. In *International Conference on Product Lifecycle Management, Milan, Italy*, pages 445–456.

Fleishanderl, G., Friedrich, G. E., Haselbock, A., Schreiner, H., and Stumptner, M. (1998). Configuring large systems using generative constraint satisfaction. *IEEE Intelligent Systems and their applications*, 13(4):59–68.

Galton, A. (2004). Multidimensional mereotopology. In *Proceedings of KR - Volume 4*, pages 45–54.

Gero, J. S. and Kannengiesser, U. (2003). Function-behaviour-structure: A model for social situated agents. In *Workshop on Cognitive Modeling of Agents and Multi-Agent Interactions, International Joint Conference on Artificial Intelligence*, pages 101–107.

Gotts, N. M. (1996). Formalising commonsense topology: The inch calculus. In *Proc. Fourth International Symposium on Artificial Intelligence and Mathematics*.

Gruhier, E., Demoly, F., Dutartre, O., Abboudi, S., and Gomes, S. (2015). A formal ontology-based spatiotemporal mereotopology for integrated product design and assembly sequence planning. *Advanced Engineering Informatics*, 29(3):495–512.

Gruninger, M. and Aameri, B. (2017). A new perspective on the mereotopology of RCC8. In *COSIT 2017*.

Gruninger, M. and Bouafoud, S. (2011). Thinking outside (and inside) the box. In *Proceedings of SHAPES 1.0: The Shape of Things. Workshop at CONTEXT-11*, volume 812. CEUR-WS.

Hahmann, T. (2013). *A reconciliation of logical representations of space: from multidimensional mereotopology to geometry*. PhD thesis, PhD thesis, Univ. of Toronto, Dept. of Comp. Science.

Hilbert, D. (1902). *The foundations of geometry*. Open court publishing Company.

Horváth, I., Vergeest, J., and Kuczogi, G. (1998). Development and application of design concept ontologies for contextual conceptualization. In *Proceedings of ASME Design Engineering Technical Conferences*, pages 1–30.

Imran, M. and Young, B. (2015). The application of common logic based formal ontologies to assembly knowledge sharing. *Journal of intelligent manufacturing*, 26(1):139–158.

Imran, M. and Young, R. (2016). Reference ontologies for interoperability across multiple assembly systems. *International Journal of Production Research*, 54(18):5381–5403.

Kim, K., Yang, H., and Kim, D. (2008). Mereotopological assembly joint information representation for collaborative product design. *Robotics and Computer-Integrated Manufacturing*, 24(6):744–754.

Kim, K.-Y., Manley, D. G., and Yang, H. (2006). Ontology-based assembly design and information sharing for collaborative product development. *Computer-Aided Design*, 38(12):1233–1250.

Levin, M. S. (2009). Combinatorial optimization in system configuration design. *Automation and Remote Control*, 70(3):519–561.

Man, K.-F., Tang, K.-S., and Kwong, S. (1996). Genetic algorithms: concepts and applications [in engineering design]. *IEEE transactions on Industrial Electronics*, 43(5):519–534.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., and Oltramari, A. (2003). Wonderweb deliverable d18–ontology library (final). *Technical Report, Laboratory for Applied Ontology, ISTC-CNR, Trento, Italy*.

McKay, A., Chase, S., Shea, K., and Chau, H. H. (2012). Spatial grammar implementation: From theory to useable software. *AI EDAM*, 26(2):143–159.

Mittal, S. and Frayman, F. (1989). Towards a generic model of configuraton tasks. In *IJCAI*, volume 89, pages 1395–1401. Citeseer.

Mosca, A., Palmonari, M., and Sartori, F. (2009). An upper-level functional design ontology to support knowledge management in sme-based e-manufacturing of mechanical products. *The Knowledge Engineering Review*, 24(3):265–285.

O'Sullivan, B. (1999). Constraint-aided conceptual design. *Engineering Research Series*, 9.

Papalambros, P. Y. and Wilde, D. J. (2000). *Principles of optimal design: modeling and computation*. Cambridge university press.

Popplestone, R., Liu, Y., and Weiss, R. (1990). A group theoretic approach to assembly planning. *AI magazine*, 11(1):82.

Radford, A. D. and Gero, J. S. (1987). *Design by optimization in architecture, building, and construction*. John Wiley & Sons, Inc.

Randell, D. A., Cui, Z., and Cohn, A. (1992). A spatial logic based on regions and connection. In *Proceedings of the KR92*, pages 165–176. Morgan Kaufmann.

Sanfilippo, E. M., Masolo, C., Borgo, S., and Porello, D. (2016). Features and components in product models. In *FOIS*, pages 227–240.

Schmidt, L. C., Shetty, H., and Chase, S. C. (2000). A graph grammar approach for structure synthesis of mechanisms. *Journal of Mechanical Design*, 122(4):371–376.

Shea, K., Cagan, J., and Fenves, S. J. (1997). A shape annealing approach to optimal truss design with dynamic grouping of members. *Journal of Mechanical Design*, 119(3):388–394.

Smith, B. (1996). Mereotopology: a theory of parts and boundaries. *Data & Knowledge Engineering*, 20(3):287–303.

Stiny, G. (1980). Introduction to shape and shape grammars. *Environment and planning B: planning and design*, 7(3):343–351.

Wielinga, B. and Schreiber, G. (1997). Configuration-design problem solving. *IEEE Expert*, 12(2):49–56.

Appendix A

A.1. Axiomatic Representation of Swing Axle Design

$$
\begin{aligned}
&(\forall x)\, \mathit{swing\_axle}(x) \supset \mathit{poly}(x) \wedge (\exists a, w_1, w_2, s_1, s_2, b_1, b_2)\mathit{beam}(b_1) \wedge \mathit{beam}(b_2)\\
&\wedge \mathit{wheel}(w_1) \wedge \mathit{wheel}(w_2) \wedge \mathit{shock}(s_1) \wedge \mathit{shock}(s_2) \wedge \neg(b_1 = b_2) \wedge \neg(s_1 = s_2) \wedge \neg(w_1 = w_2)\\
&\wedge \mathit{box\_PP}(b_1, a) \wedge \mathit{box\_PP}(b_2, a) \wedge \mathit{box\_PP}(w_1, a) \wedge \mathit{box\_PP}(w_2, a) \wedge \mathit{box\_PP}(s_1, a) \wedge \mathit{box\_PP}(s_2, a)\\
&\wedge \mathit{ball\_joined}(b_1, b_2) \wedge \mathit{fixed\_joined}(w_1, b_1) \wedge \mathit{fixed\_joined}(w_2, b_2)\\
&\wedge \mathit{ball\_joined}(w_1, s_1) \wedge \mathit{ball\_joined}(w_2, s_2)\\
&\wedge \mathit{xbetween}(w_1, s_1, w_2) \wedge \mathit{xbetween}(w_1, s_2, w_2) \wedge \mathit{xbetween}(w_1, b_1, b_2) \wedge \mathit{xbetween}(b_1, b_2, w_2) \wedge \mathit{in}(a, x).
\end{aligned}
$$

A.2. Axiomatic Representation of Double Wishbone Design

$$
\begin{aligned}
&(\forall x)\, \mathit{double\_wishbone}(x) \supset \mathit{poly}(x) \wedge (\exists a, w_1, w_2, s_1, s_2, b_1, b_2, b_3, b_4, st_1, st_2, ch)\mathit{beam}(b_1) \wedge \mathit{beam}(b_2)\\
&\wedge \mathit{beam}(b_4) \wedge \mathit{beam}(b_5)\mathit{wheel}(w_1) \wedge \mathit{wheel}(w_2) \wedge \mathit{shock}(s_1) \wedge \mathit{shock}(s_2) \wedge \mathit{strut}(st_1) \wedge \mathit{strut}(st_2)\\
&\wedge \mathit{chassis}(ch) \wedge \neg(b_1 = b_2) \wedge \neg(b_1 = b_3) \wedge \neg(b_1 = b_4) \wedge \neg(b_2 = b_3) \wedge \neg(b_2 = b_4) \wedge \neg(b_3 = b_4)\\
&\wedge \neg(st_1 = st_2) \wedge \neg(s_1 = s_2) \wedge \neg(w_1 = w_2)\\
&\wedge \mathit{box\_PP}(b_1, a) \wedge \mathit{box\_PP}(b_2, a) \wedge \mathit{box\_PP}(b_3, a) \wedge \mathit{box\_PP}(b_4, a) \wedge \mathit{box\_PP}(w_1, a) \wedge \mathit{box\_PP}(w_2, a)\\
&\wedge \mathit{box\_PP}(s_1, a) \wedge \mathit{box\_PP}(s_2, a) \wedge \mathit{box\_PP}(st_1, a) \wedge \mathit{box\_PP}(st_2, a) \wedge \mathit{box\_PP}(ch, a)\\
&\mathit{fixed\_joined}(w_1, st_1) \wedge \mathit{fixed\_joined}(w_2, st_2) \wedge \mathit{ball\_joined}(w_1, s_1) \wedge \mathit{ball\_joined}(w_2, s_2)\\
&\wedge \mathit{ball\_joined}(b_1, st_1) \wedge \mathit{ball\_joined}(b_2, st_1) \wedge \mathit{ball\_joined}(b_3, st_2) \wedge \mathit{ball\_joined}(b_4, st_2)\\
&\wedge \mathit{ball\_joined}(b_1, ch) \wedge \mathit{ball\_joined}(b_2, ch) \wedge \mathit{ball\_joined}(b_3, ch) \wedge \mathit{ball\_joined}(b_4, ch)\\
&\wedge \mathit{xbetween}(w_1, s_1, w_2) \wedge \mathit{xbetween}(w_1, s_2, w_2) \wedge \mathit{xbetween}(w_1, st_1, ch) \wedge \mathit{xbetween}(ch, st_2, w_2)\\
&\wedge \wedge \mathit{xbetween}(w_1, ch, w_2) \wedge \mathit{xbetween}(st_1, b_1, ch) \wedge \mathit{xbetween}(st_1, b_2, ch) \wedge \mathit{xbetween}(st_2, b_3, ch)\\
&\wedge \mathit{xbetween}(st_2, b_4, ch) \wedge \mathit{in}(a, x).
\end{aligned}
$$

A.3. Axiomatic Representation of Joining Methods

Kim et al. (2008) present a classification of assembly joining methods and use Smith's mereotopology (Smith, 1996) to axiomatically specify these classes. However, their axiomatization is incomplete in the sense that it does not capture the intended specifications of different types of joints (Aameri and Gruninger, 2017). This section demonstrates the axiomatization of three types of joining methods classified by Kim et al. (2008), namely fastening, welding, and spot welding, using the the Assembly Ontology.

When fastening two three-dimensional components, one of the boundary surfaces of one component is connected (in the topological sense) to one of the boundary surfaces of the other. However, if we weld two three-dimensional objects, the mating features of the objects will be transformed into one entity, and more importantly, the surface will not be a boundary surface anymore. In other words, two welded objects share a common geometric entity:

$$(\forall x,y)\, fastened\_box(x,y) \equiv (box(x) \wedge box(y) \wedge \neg(x=y) \wedge$$
$$(\exists z,w) \neg(z=w) \wedge surface\_bound(z) \wedge surface\_bound(w) \wedge in(z,x) \wedge in(w,y) \wedge surface\_C(z,w)).$$
$$(\forall x,y)\, welded\_box(x,y) \equiv (box(x) \wedge box(y) \wedge \neg(x=y)$$
$$\wedge (\exists z)(point(z) \vee edge(z) \vee surface(z)) \wedge in(z,x) \wedge in(z,y)).$$

Considering the above description of welding, two spot-welded objects share a point:

$$(\forall x,y) spot\_welded\_box(x,y) \equiv (box(x) \wedge box(y) \wedge \neg(x=y) \wedge (\exists z) point(z) \wedge in(z,x) \wedge in(z,y)).$$

Different types of fastening methods classified in Kim et al. (2008) (namely, fastening by threaded fasteners, fastening by rivets, and fastening by compression) can also be defined by using the Shape Ontology in describing shapes of the fastening objects.