Mereotopological Description of Product-Process Information and Knowledge for PLM

Frédéric Demoly, Aristeidis Matsokis, Dimitris Kiritsis, Samuel Gomes

HAL Id: hal-01526125
https://inria.hal.science/hal-01526125

Submitted on 22 May 2017

Mereotopological description of product-process information and knowledge for PLM

Frédéric Demoly[1] ⋆, Aristeidis Matsokis[2], Dimitris Kiritsis[2], and Samuel Gomes[1]

[1] IRTES-M3M, UTBM, 90010 Belfort cedex, France
`{frederic.demoly,samuel.gomes}@utbm.fr`
[2] STI-IGR-LICP, EPFL, CH-1015 Lausanne, Switzerland
`{aristeidis.matsokis,dimitris.kiritsis}@epfl.ch`

Abstract. This paper describes a description approach for modeling product-process information in the contexts of assembly oriented design and product lifecycle management (PLM). The growing evolution of models, methodologies, systems and tools over the entire product lifecycle has highlighted limits and difficulties – such as the awareness and understanding in engineering – that did not exist before. An emergent challenge remains in increasing awareness and understanding of actors in the management of product information and knowledge. This requires effort in new inspired approaches in the qualitative representation and reasoning of the product and processes, in ontological applications, knowledge-based approaches, models, etc. The main objective is to make assembly information consistent, accessible and exploitable by data management systems and computer-aided X tools by introducing a logical foundation. In this context, product-process relationships are considered and described in the part-whole theory supported by mereology and its extension, mereotopology, then implemented in an ontology.

Keywords: Product Lifecycle Management; Assembly modeling; Ontology; Mereotopology; Concurrent Engineering

1   Introduction

The growing evolution of models, methodologies, systems and tools over the entire product lifecycle has highlighted limits and difficulties – such as the awareness and understanding in engineering – that did not exist before. The 2D drawings of the product and face-to-face discussions between lifecycle actors have been revised for new way of working with people, tools, and managing/representing information and knowledge. In such a context, some approaches such as concurrent engineering, Design for X, 3D parametric design, ontology-based approaches, decision-making support, knowledge-based approaches, integrated data management have appeared to build the todays PLM strategy with a particular automation level, but an understanding layer is still missing. From

---

⋆ Corresponding author.

a system and tool point of view, the traditional segmentation of the digital chain – which includes CAD, CAE, PDM, ERP to name a few – has highlighted needs in hub, bridge, dashboard applications in order to enable the continuity and relationship of product lifecycle phases, but still limited to point-to-point interface.

An emergent challenge remains in increasing awareness and understanding of actors in the management of product information and knowledge. Indeed, due to the information overload (e.g. 3D large assembly model, multi-instances and parameters, etc.) this new way of working do not enable the full understanding of the lifecycle information and knowledge which is important in order to improve actors awareness and therefore quality in product lifecycle phases. This requires effort in new inspired approaches in the qualitative representation and reasoning of the product and processes, in ontological applications, knowledge-based approaches, models, etc.

The main objective is to make assembly information consistent, accessible and exploitable by data management systems and computer-aided X tools by introducing a logical foundation. Product information and knowledge as well as the related assembly sequence require a logical framework in order to be managed consistently and processed proactively. As a consequence, product-process relationships are considered in the part-whole mathematical and philosophical theory supported by mereology and its extensions with topology such as mereotopology primitives, theorems and axioms, and temporal relationships, then implemented in an ontology.

The paper briefly states, in section 2, the recent and relevant research works on assembly oriented design (AOD) approaches, assembly models and space-time mereotopological theories. In section 3, the proposed approach — called PRONOIA[3] (PROduct relatioNships description based On mereotopologIcAl theory) — for defining product relationships based on mereotopological theory and temporal relationships is presented. Finally, in section 4, an ontological implementation of PRONOIA by using Ontology Web Language - Description Logic (OWL-DL) and Semantic Web Rule Language (SWRL) is described, in order to demonstrate the application of such a qualitative description approach for PLM systems.

2   Literature review

From a recent past, the issue of concurrent product design and ASP [2, 3] has received much attention [4]. These efforts aimed at tackling difficulties and weaknesses discovered in Design for Assembly (DFA) and ASP approaches by introducing the concept of "Assembly-Oriented Design (AOD)" [2, 5]. The assembly oriented practice in the product development can be seen as a top-down vision

---

[3] PRONOIA was an Okaenid nymph of Mount Parnassos in Phokis (central Greece). She was the wife of the Titan Prometheus, and the goddess of foresight. Today this word means forethought and provident care.

by proactively considering the assembly related product design and their relationship issues in the early phases of the product development process. This emerging fashion highlights some challenges related to the recent shift in engineering design that promotes the relationships-based modelling and management paradigm [6]. Current engineering requirements consist of closer integration of product design and lifecycle models, better traceability on various abstraction levels of the product (i.e. functional, behavioural, structural, geometric, technological, etc.) and rational and consistent information management support with the concept of "relational design".

Based on this, recent assembly models relevant with the scope of the paper has been reported. Actually he significant growth of semantic value of information and knowledge models during the last decade has highlighted new contributions towards assembly semantic formalisms. Zha and Du promoted a knowledge-based system using multi-agent system and Petri Net to support assembly design and ASP by considering as input the part relational information [2]. Kim *et al.* proposed a spatial relationships-based assembly design formalism describing assembly relations, and an assembly model, in which relations are represented in XML format [9]. This work was only focused on the geometric aspect of the product and did not enable an efficient interpretation by CAD tools. Recently, Kim *et al.* have tried to demonstrate the feasibility of an ontological representation of assembly and associated constraints [10]. The same authors have proposed an ontology-based representation for assembly joints in collaborative product design by using mereotopological primitives and SWRL formalism [11]. Although this latter reported work is case study oriented, it can be echoed here with the scope of the paper. More recently and based on a multi-view model MUVOA (Multiple Viewpoints Oriented Assembly) [8], Demoly *et al.* have described a product-process data management approach by introducing the management of product relationships at various abstraction levels and in separate manner [7].

Other relevant research works can also be considered, especially in the field of formal ontology, which is intended to be the support of the proposed approach. Mereotopology can be considered as a theory derived from mereology which is the theory of part-whole relation [12]. Mereology enables the description of parthood relation which is a reflexive, antisymmetric and transitive primitive. An extension of this mathematical theory towards pointless topology is reported by the mereotopology, which provides a further first-order logic description of connections between entities such as spatial regions [13, ?,?]. Indeed, instead of focusing on set theory, experience in product modelling and design has emphasised on the need to work with a region-based theory by considering the product as it is perceived in the real world [16]. Built on this, many applications have been proposed: early product design stages [16], assembly joining technologies of mechanical parts[11], four-dimensionalism [19] in oil and gas industry [20], etc.

A more complete survey of these research domains can be found in [21, 22].

3   Mereotopological product relationship description approach

In this section, the proposed PRONOIA approach is described. It enables the qualitative description of the product relationships at the beginning of the product design process by incorporating information related to the early-defined assembly sequence.

3.1   Basic mereotopology

We introduce basic definitions of mereotopological primitives based on Smith's mereotopology [17], which are used as a starting point for the following subsections. Each primitive is written with a bold upper case between two entities as follows $x\mathbf{R}y$, where $\mathbf{R}$ describes a particular relation between two entities, and variables x, y, z, etc. are ranged over entities. Table 1 presents operators and symbols used for the description of mereotopological primitives and temporal relationships. All the mereotopological primitives described in this work are shown in Fig. 1 and Tab. 2.

Table 1. Fundamental mereotopological operators

Symbol	Name	Symbol	Name
$\vee$	Logical disjunction	$=$	Equality
$\wedge$	Logical conjunction	$\equiv$	Equivalence
$\rightarrow$	Logical implication	$\neg$	Logical negation
$\exists$	Existential quantifier	$\forall$	Universal quantifier
ϕ	Condition	ι	Definite descriptor
σ	Sum (fusion or join)	π	General product
$:=$	Definition	$\neq$	Difference

Firstly, the mereological theory enables the description of the parthood relation, where $x\mathbf{P}y$ means x *is part of* y and represents parthood for any application as shown in Fig. 1. Based on this primitive, other primitives can be further derived. Thus, the first derived mereological notion is that x *overlaps* y, denoted $x\mathbf{O}y$, when z is any part of x and y. The second primitive is that x *is discrete from* y, and is written $x\mathbf{D}y$, this primitive means that x does not overlap y. Last is the definition of x *is a point*, such as written $\mathbf{Pt}(x)$.

Mereotopological primitives can be introduced by considering topological means with the above described mereological primitives. A primitive derived from $\mathbf{P}$ is that $x\mathbf{IP}y$ and means x *is an interior part of* y. Another one is that x *crosses* y, and written $x\mathbf{X}y$. This primitive means that x is neither *a part of* y nor *discrete from* y. It also means that x *overlaps* both y and its complement ($\mathbf{1} - y$, with $\mathbf{1}$ stands for the Universe). The following primitive is x *straddles* y, denoted $x\mathbf{St}y$, and represents either a tangent or a boundary. Then it leads
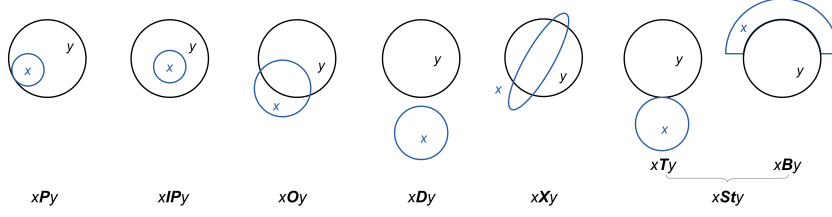
Fig. 1. Mereotopological primitives representation

Table 2. Mereotopological primitives description

Mereotopological description	Definition
$x\mathbf{P}y := \forall z(z\mathbf{O}x \to z\mathbf{P}y)$	x is part of y
$x\mathbf{O}y := \exists z(z\mathbf{P}x \land z\mathbf{P}y)$	x overlaps y
$x\mathbf{D}y := \neg x\mathbf{O}y$	x is discrete from y
$\mathbf{Pt}(x) := y(y\mathbf{P}x \to y = x)$	x is a point
$\sigma x(\phi x) := \iota y(\forall w(w\mathbf{O}y \equiv \exists v(\phi v \land w\mathbf{O}v)))$	sum of ϕers
$x\mathbf{X}y := \neg x\mathbf{P}y \land \neg x\mathbf{D}y$	x crosses y
For $y \neq \mathbf{1}, x\mathbf{X}y := x\mathbf{O}y \land x\mathbf{O}(\mathbf{1} - y)$	
$x\mathbf{St}y := \forall z(x\mathbf{IP}z \to z\mathbf{X}y)$	x straddles y
$x\mathbf{B}y := \forall z(x\mathbf{P}x \to z\mathbf{St}y)$	x is boundary of y
$x\mathbf{T}y := \exists z(z\mathbf{P}x \land z\mathbf{B}y)$	x is tangent of y
$x\mathbf{IB}y := x\mathbf{IP}y \land x\mathbf{B}x$	x is interior boundary of y
$x \subseteq_t y := \forall z(z \divideontimes x \to z \divideontimes y)$	x is temporally included in y
$x\mathbf{O}_t y := \exists z(z \subseteq_t y \land y \subseteq_t x)$	x temporally overlaps y

to the introduction of the boundary primitive, such as written $x\mathbf{B}y$ (means x *is boundary of y*), and tangent primitive, denoted $x\mathbf{T}y$ (means x *is tangent of y*). A further description of the related mereotopological axioms can be found in [17].

3.2   Overall description of the PRONOIA approach

Based on the mathematical and philosophical foundation described previously, we present the overall vision of the PRONOIA approach. To achieve this, the introduced mereotopological theory-based approach is combined with an assembly-oriented design framework for product design and assembly sequence planning called PASODE (Proactive ASsembly Oriented DEsign) [1]. Figure 2 illustrates the main steps of the proposed approach which are:

Step 1: Based on an initial engineering bill of material (eBOM), the product architect defines the part-to-part relationships (contact, precedence, kinematic and technological) (see Tab. 3) in order to fulfil already defined functional requirements. At this stage, the part relational information can be represented through a directed graph.

Step 2: An assembly sequence is then generated by the ASDA (Assembly Sequence Definition Algorithm) algorithm [1] which reasons on the early-defined part-to-part relational information. The assembly sequence provides information about selected sub-assembly types (i.e. serial, parallel, etc.), layers and base parts, and can be represented through gantt chart including temporal relations (left side of Fig. 2). At the same time, the part-to-part relationships are described with mereotopological primitives and temporal relations (right side of Fig. 2).

Step 3: A refined mereotopological description is produced based on the previous steps, i.e. the early-defined assembly sequence and the first mereotopological description. This enables the product structuring as well as the product relationships definition at the early design stage.

Step 4: The full mereotopological product relationships description is implemented with OWL-DL and SWRL rules in order to provide the product architect with a further insight of assembly and design intents from a qualitative point of view, and to make this information available for reuse in the product lifecycle.

In the following sub-sections, each aforementioned step of the proposed PRONOIA approach is described in detail.

3.3   Mereotopological description of part-to-part relationships

This first definition of part-to-part relationships (Tab. 3) produced by the product architect (step 1 of Fig. 2) can be expressed with mereotopological primitives and temporal relations (step 2 of Fig. 2). This information therefore requires the introduction of one topological primitive: *x is connected to y*, written $x\mathbf{C}y$ (from a spatial point of view) and a temporal precedence denoted $x \triangleleft y$, which means that *x temporally precedes y*.

At a more detailed level, the authors propose to consider the condition ϕ, such as defined by Smith [17] and recently reused by Kim *et al.* [11], and stated that a condition ϕ in a single variable x is satisfied if and only if the sentence ϕx is true

Table 3. Part-to-part relationships definition

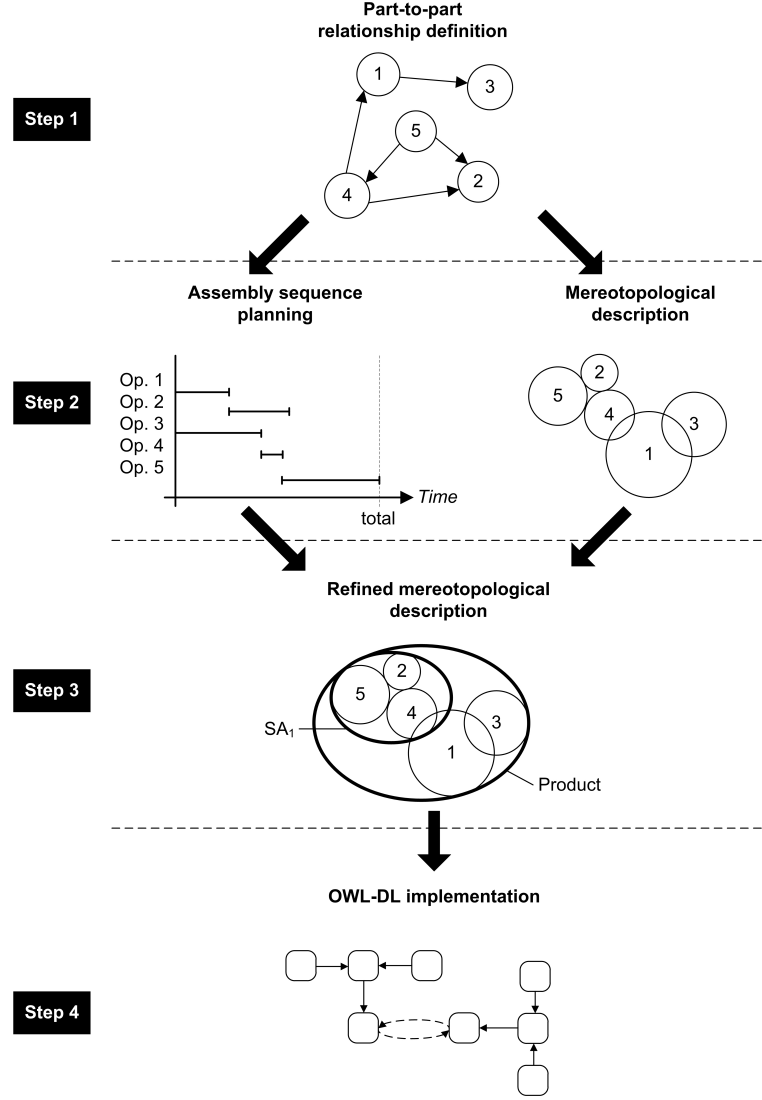| Relationship	Definition
Contact	Physical contact relation between two components.
Precedence	Assembly logical and temporal order for two components in contact or in non-contact.
Kinematic pair	Additional information on contact relation which enables the description of constrained degrees of freedom (rotation and translation) for each product component.
Technological pair	Additional information on contact relation which enables the definition of the assemblability of the product, and therefore on the joining relation between two components.

Fig. 2. Mereotopological product relationships description approach (PRONOIA)

for at least one value of x. Hence the sum of ϕers can be defined as the entity y which is such that, given any entity w, w overlaps y if and only if w overlaps with something that ϕ, such denoted $\sigma x(\phi x) := \iota y(\forall w(w\mathbf{O}y \equiv \exists v(\phi v \wedge w\mathbf{O}v)))$.

Based on this assumption, it is possible to introduce and define two entities (Fig. 3):

- k as any geometric skeleton entity for assembly positioning including point, line (also called straight), plane, curve of two product components x and y, and denoted $k := \sigma z(\phi z) \rightarrow \forall z(\phi z \rightarrow (z\mathbf{T}x \wedge z\mathbf{T}y) \wedge (z\mathbf{P}x \wedge z\mathbf{P}y))$.
- j as any geometric entity for mating boundary including point, line, plane, surface of two product components x and y, and denoted $j := \sigma z(\phi z) \rightarrow \forall z(\phi z \rightarrow z\mathbf{B}x \wedge z\mathbf{B}y)$.
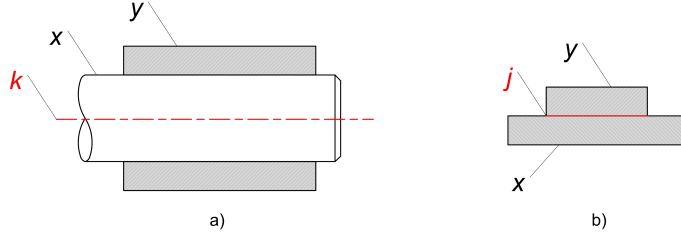

Fig. 3. Geometric representation of k a) and j b) entities

The entity k has been introduced to describe kinematic pairs, and j has been reused from Kim *et al.* definition for technological pairs [11]. Table 4 presents the mereotopological description of kinematic pairs. The geometric skeleton k is used as a neutral element between two product components in contact for assembly synchronisation (e.g. CAD models of parts and sub-assemblies) at the assembly modeling and design stage. For instance, a revolute pair allows only one rotation between two parts, therefore introducing two geometric skeletons: a plane and a line (straight) that are together perpendicularly constrained.

3.4   Mereotopological description of assembly sequence

From a four-dimensionalism point of view, the consideration of temporal aspect in mereotopology has led to the introduction of new primitives such as described in [18, 19]. These primitives enable the definition of temporal relations between spatial regions, especially the temporal inclusion (written $\subseteq_t$), the temporal overlaps (written $\mathbf{O}_t$) and the temporal precedence (written $\triangleleft$) based on the temporal connection (written $\divideontimes$) as infix operators. Figure 4 shows the representation of these temporal relations in order to facilitate understanding. In this field, literature has provided theories by highlighting connection among spatial, temporal, and spatio-temporal entities. However, only spatial regions are considered as entities in this paper since the application field is focused on product engineering.

Based on this assumption, the aim is to propose a mereotopological description of each sub-assembly type as described in [1], in order to build and describe the assembly sequence (steps 2-3 of Fig. 2). The ASDA algorithm — enabling the

Table 4. Mereotopological description of kinematic pairs

Kinematic pair	Mereotopological description	Comment
Rigid	$x\mathbf{K_{rig}}y := \exists x, y\{(x\mathbf{X}y \wedge x\mathbf{T}y) \wedge (x\mathbf{O}k \wedge y\mathbf{O}k)\}$	k is a point
Revolute	$x\mathbf{K_{rev}}y := \exists x, y\{(x\mathbf{X}y \wedge x\mathbf{T}y) \wedge (x\mathbf{O}k_1 \wedge y\mathbf{O}k_1) \wedge$ $(x\mathbf{T}k_2 \wedge y\mathbf{T}k_2)\}$	k_1 is a line, k_2 is a plane
Prismatic	$x\mathbf{K_{pri}}y := \exists x, y\{x\mathbf{X}y \wedge (x\mathbf{O}k \wedge y\mathbf{O}k)\}$	k is a line
Screw	$x\mathbf{K_{scr}}y := \exists x, y\{(x\mathbf{X}y \wedge x\mathbf{T}y) \wedge (x\mathbf{O}k_1 \wedge y\mathbf{O}k_1) \wedge$ $(x\mathbf{T}k_2 \wedge y\mathbf{T}k_2)\}$	k_1 is a line, k_2 is a plane
Cylindrical	$x\mathbf{K_{cyl}}y := \exists x, y\{x\mathbf{X}y \wedge (x\mathbf{O}k \wedge y\mathbf{O}k)\}$	k is a line
Spherical	$x\mathbf{K_{sph}}y := \exists x, y\{(x\mathbf{St}y \wedge x\mathbf{IB}y) \wedge (x\mathbf{O}k \wedge y\mathbf{O}k)\}$	k is a point
Planar	$x\mathbf{K_{pla}}y := \exists x, y\{x\mathbf{T}y \wedge (x\mathbf{T}k \wedge y\mathbf{T}k)\}$	k is a plane
Point-contact	$x\mathbf{K_{pt}}y := \exists x, y\{x\mathbf{T}y \wedge (x\mathbf{T}k \wedge y\mathbf{T}k)\}$	k is a point
Line-contact	$x\mathbf{K_{lin}}y := \exists x, y\{x\mathbf{T}y \wedge (x\mathbf{T}k \wedge y\mathbf{T}k)\}$	k is a line
Curve-contact	$x\mathbf{K_{cur}}y := \exists x, y\{x\mathbf{B}y \wedge (x\mathbf{B}k \wedge y\mathbf{B}k)\}$	k is a curve
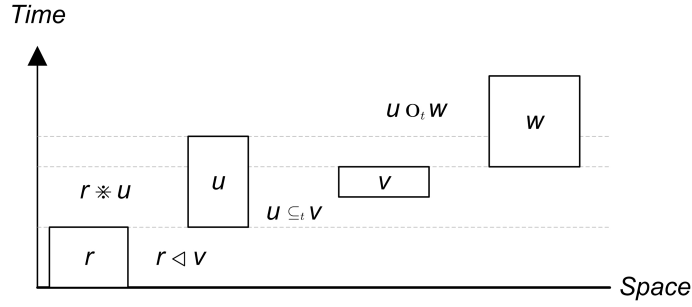
Fig. 4. Representation of temporal relations for regions r u, v, and w

assembly sequence definition from part-to-part relationships — is fully described in [1]. As a result, four theorems have been proposed below:

Theorem 1. *Any assembly of three components can be defined as serial assembly type if and only if:*
$$A_S := \exists x, y, z\{(x\boldsymbol{C}y \wedge x\boldsymbol{C}z \wedge y\boldsymbol{D}z) \wedge (x \triangleleft y \wedge y \triangleleft z)\}$$

Theorem 2. *Any assembly of three components can be defined as parallel assembly type if and only if:*
$$A_P := \exists x, y, z\{(x\boldsymbol{C}y \wedge x\boldsymbol{C}z \wedge y\boldsymbol{D}z) \wedge (x \triangleleft y \wedge x \triangleleft z)\}$$

Theorem 3. *Any assembly of three components can be defined as interconnected serial assembly type if and only if:*
$$A_{IS} := \exists x, y, z\{(x\boldsymbol{C}y \wedge x\boldsymbol{C}z \wedge y\boldsymbol{C}z) \wedge (x \triangleleft y \wedge x \triangleleft z \wedge y \triangleleft z)\}$$

Theorem 4. *Any assembly of three components can be defined as constrained serial assembly type if and only if:*
$$A_{CS} := \exists x, y, z\{(x\boldsymbol{C}y \wedge x\boldsymbol{C}z \wedge y\boldsymbol{D}z) \wedge (x \triangleleft y \wedge x \triangleleft z \wedge y \triangleleft z)\}$$

Built on this, the assembly sequence results from the concatenation of sub-assemblies descriptions including layers, and later on will influence the mereotopological description of the product relationships.

4   OWL-DL implementation

This section describes the last step (step 4) of Fig. 2 whose aim is to reuse the relational information along the product lifecycle. In this work the authors are defining an OWL-DL ontology information model. The concept behind developing an OWL-DL ontology model was to implement ontology advantages and features into the model which include: data structure layout, description logic (DL) [24] reasoning, semantic web rules including SWRL [25] and Jess rule engine [26]. The use of the ontology combined with Description Logics (DLs) allows the authors to define/figure out: equivalencies of classes, consistency of the classes and make re-classification of the classes according to the concepts that they represent. Moreover, DLs are also used to infer/categorise instances under classes according to rules and requirements. Thus, the information system is executable, dynamic and flexible. In the rest of this work, the following naming conventions are used: names of classes are written in capitalised/lower case Arial (i.e. Physical_Product). Names of attributes and relationships are capitalised /lower case Courier New (i.e. isParentOf) while names of instances are in italics Arial (i.e. *Physical_Part_1*).

The authors focus on the information exchange between product design and assembly process phases. This is a part of the BOL information. The aim is to make the information available along the product lifecycle. In order to describe the information in a generic and tangible manner, an ontology model has been developed, which could be then used as a part of a model describing the entire product lifecycle. The proposed PRONOIA ontology model is shown in Fig. 5. The model has been developed to describe the assembly properties as they have been defined in mereotopology. It consists of the following main classes: Kinematic_Pair, Physical_Contact, Relation, Physical_Product, Technological_Pair, Geometric_Entity. These classes consist of several sub-classes which are also shown in Fig. 5. The ontology model has been developed using the Protégé ontology editor (Protégé-OWL) and the classes have been defined using DL rules. The development process is the one described by Matsokis and Kiritsis [27]. The names of the relationships (objects properties) follow the Class_Name2Class_Name policy.

Fig. 6 shows the relationships between classes and those describing the mereotopological primitives. DL rules (restrictions) have been defined in order to give a logical meaning to the classes and to make them understandable/readable for the reasoner. The reasoner used in this case is Pellet 1.5.2 [28]. A comprehensive list of the added DL rules is presented in another paper [22]. Therefore, the reasoner can be used for supporting product architects by providing consistency check of the model, equivalencies among the classes, re-classification of the class-hierarchy and inference on the instances [29]. For example, the class Revolute of the PRONOIA model has the following restriction using standard DL notation

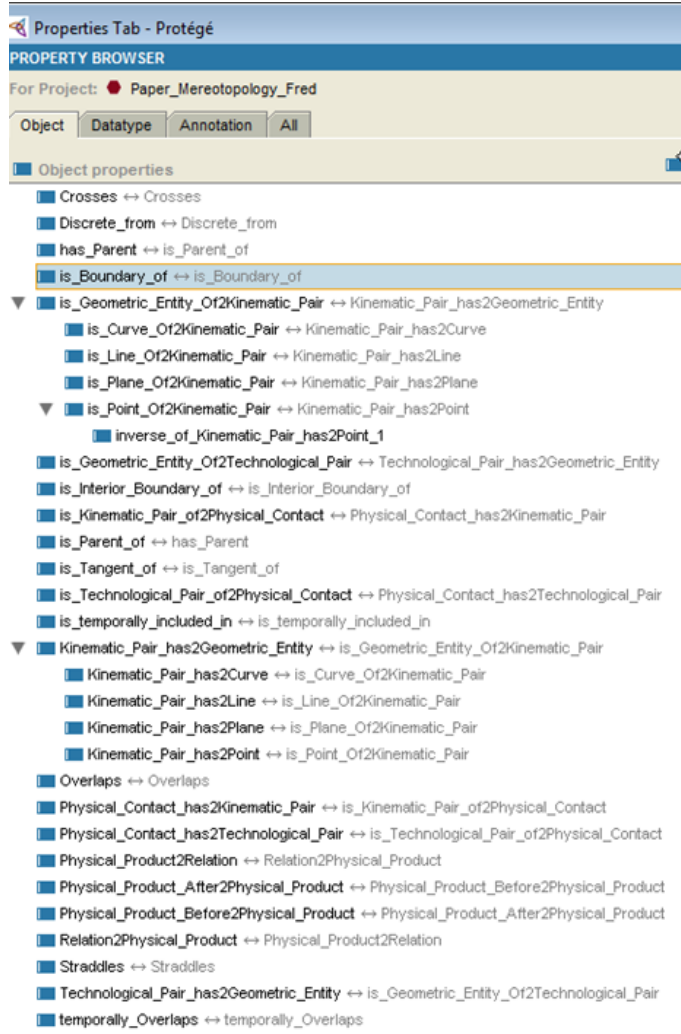**Fig. 5.** The PRONOIA ontology model in Protégé
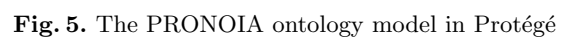
(see also Fig. 7):

Fig. 6. Relationships defined and browsed in Protégé properties view

$$\text{Revolute} \equiv \text{Kinematic\_Pair} \sqcap \forall \, \text{Kinematic\_Pair\_has2Geometric\_Entity}.(\text{Line} \sqcup \text{Plane})$$

This introduces the concept name Revolute, and asserts that its instances are just those individuals that are instances of Kinematic_Pair, and all of whose Geometric_Entity are instances of either Line or Plane. These restrictions rules are read by the reasoner and hence, the model is efficiently checked inconsistencies, inference and re-classification.
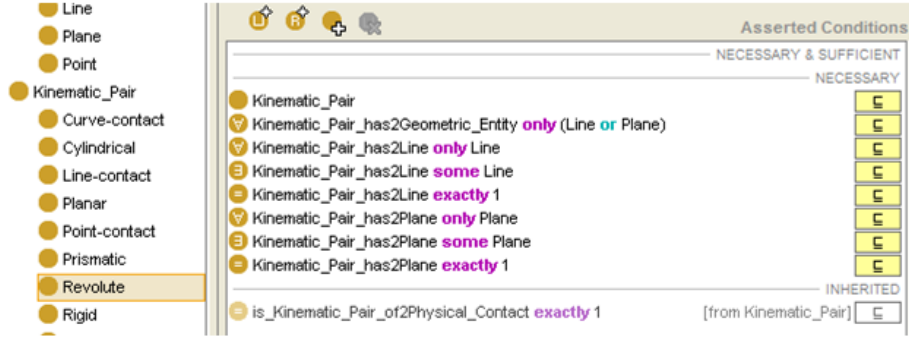
Fig. 7. Relationships defined and browsed in Protégé properties view

Moreover, SWRL rules have been added to the model in order to provide the mereological relationships between the different parts and geometric entities of the model (see [22]). In this way the parts are connected, and the SWRL and the Jess rule engine are used to provide the connections among the different parts, and between the parts and the geometric entities. For example, the SWRL rule related to the mereotopological description of the kinematic pair Revolute can be defined as:

Physical_Contact_has2Kinematic_Pair(?pcx, ?kpx) $\wedge$ Relation2Physical_Product(?pcx, ?ppx) $\wedge$ Relation2Physical_Product(?pcx, ?ppy) $\wedge$ differentFrom(?ppx, ?ppy) $\wedge$ Revolute(?kpx) $\wedge$ Kinematic_Pair_has2Line(?kpx, ?glx) $\wedge$ Kinematic_Pair_has2Plane(?kpx, ?gpx) $\rightarrow$ Crosses(?ppx, ?ppy) $\wedge$ is_Tangent_of(?ppx, ?ppy) $\wedge$ Overlaps(?ppx, ?glx) $\wedge$ is_Tangent_of(?ppx, ?gpx) $\wedge$ Overlaps(?ppy, ?glx) $\wedge$ is_Tangent_of(?ppy, ?gpx)

This kind of SWRL rule allows the product architect to reason about OWL individuals and to infer new knowledge. It contains an antecedent part (left side of $\rightarrow$) and a consequent part (right side of $\rightarrow$), both parts consist of logical conjunctions of atoms. In this way, a SWRL rule may be read as: if all the atoms in the antecedent part are true, then the consequent part is also true. Here the SWRL rule example expresses the necessary set of conditions dedicated to the generation of the mereotopological description (*i.e.* $x\mathbf{K_{rev}}y := \exists x, y\{(x\mathbf{X}y \wedge x\mathbf{T}y) \wedge (x\mathbf{O}k_1 \wedge y\mathbf{O}k_1) \wedge (x\mathbf{T}k_2 \wedge y\mathbf{T}k_2)\}$) of the kinematic pair Revolute with two geometric skeleton entities (*i.e.* k_1 as line and k_2 as plane). These kinds of query enable the product architect to have an overview of the influence of its part-to-part relations on the product mereotopological description, which will be reused by the designer during the geometric modelling stage in CAD tool.

The developed ontology model is generic and it can be used and extended for any type of application using its characteristics.

5   Conclusion and future work

Following a much larger project in the field of assembly oriented design and PLM [1, 22], the proposed PRONOIA approach consists in the description of product relationships from a qualitative point of view at the begining of the design process. This main objective has been covered with a large review of the literature of the fields and still represents a huge challenge to tackle in order to provide a new formal and machine-interpretable representation of AOD issue. Thus, the proposal has enabled the mereotopological description of product relationships at various abstraction levels, in order to provide a full insight and understanding of the product definition and assembly. The relationship-based modelling vision requires a qualitative description at the early stages of the product development process in order to promote a top-down and proactive support for product architects and designers. Moreover, an ontological implementation has been introduced using OWL-DL and SWRL in order to represent formally the assembly information and enable its reuse in the other lifecycle phases. Future work will address the application aspect of this theoretical basis, especially with an industrial case study in order to demonstrate its relevance in engineering.

References

1. Demoly, F., Yan X.-T., Eynard, B., Rivest, L. and Gomes, S., 2011. An Assembly-Oriented Design Framework for Product Structure Engineering and Assembly Sequence Planning. *Robotics and Computer-Integrated Manufacturing*, 27(1), 33–46.
2. Zha, X.F. and Du, H., 2002. A PDES/STEP-based model and system for concurrent integrated design and assembly planning. *Computer-Aided Design*, 34, 1087–1110.
3. Barnes, C.J., Jared, G.E.M. and Swift, K.G., 2004. Decision support for sequence generation in an assembly oriented design environment. *Robotics and Computer-Integrated Manufacturing*, 20, 289–300.
4. Wang, L., Keshavarzmanesh, S., Feng, H.Y. and Buchal, R.O., 2009. Assembly process planning and its future in collaborative manufacturing: a review. *International Journal of Advanced Manufacturing Technology*, 41, 132–144.
5. Mantripragada, R., 1998. Assembly oriented design: concepts, algorithms and computational tools. PhD thesis, Massachussetts Institute of Technology.
6. Brown, J., 2006. *Managing Product Relationships: Enabling Iteration and Innovation in Design*. Business Value Research Series, June, AberdeenGroup.
7. Demoly, F., Yan, X.-T., Eynard, B., Kiritsis, D., and Gomes, S., 2011. Integrated product rela- tionships management: a model to enable concurrent product design and assembly sequence planning, Journal of Engineering Design, Accepted, DOI:10.1080/09544828.2011.629317.
8. Demoly, F., Monticolo, D., Eynard, B., Rivest, L. and Gomes, S., 2010. Multiple viewpoint modelling framework enabling integrated product-process design. *International Journal on Interactive Design and Manufacturing*, 4(4), 269–280.
9. Kim, K.Y., Wang, Y., Muogboh, O.S. and Nnaji, B.O., 2004. Design formalism for collaborative assembly design. *Computer-Aided Design*, 36(9), 849–871.

10. 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, 1233–1250.
11. Kim, K.-Y., Yang, H., and Kim, D.W., 2008. Mereotopological assembly joint information representation for collaborative product design. *Robotics and Computer-Integrated Manufacturing*, 24, 744–754.
12. Leśniewski, S., 1982. On the foundations of mathematics. *Topoi* 2, 7–52.
13. Asher, N., and Vieu, L., 1995. Towards a geometry of common sense: A semantics and a complete axiomatisation of mereotopology. In Proc. IJCAI-95, Montréal, Québec, 846–852.
14. Varzi, A., 1996. Parts, wholes, and part-whole relations: The prospects of mereotopology. *Data and Knowledge Engineering*, 20, 259–286.
15. Borgo, S., Guarino, N., and Masolo, C., 1996. A pointless theory of space based on strong connection and congruence. In L. Aiello, J. Doyle, S.C. Shapiro (Eds.), Proc. Internat. Conference on Principles of Knowledge Representation and Reasoning (KR-96), Cambridge, MA, 220–229.
16. Salustri, F.A., 2002. Mereotopology for product modeling. A new framework for product modeling based on logic. *Journal of Design Research*, 2(1).
17. Smith, B., 1996. Mereotopology: a theory of parts and boundaries. *Data and Knowledge Engineering*, 20, 287–303.
18. Muller, P., 1998. A qualitative theory of motion based on spatio-temporal primitives. In A.G. Cohn, L. Schubert, and S. Shapiro (Eds.), Proceedings of KR-98, 131-141.
19. Stell, J.G., and West, M., 2004. A 4-dimensionalist mereotopology. In A.C. Varzi, L. Vieu (Eds.), Proceedings of formal ontology in information systems, IOS Press, 261–272.
20. ISO15926-2, 2003. Integration of lifecycle data for process plant including oil and gas production facilities: Part 2. Data model.
21. Demoly, F., Matsokis, A., and Kiritsis, D., 2011. A mereotopology-based approach for integrated assembly modeling and planning. Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conferences, IDETC/CIE 2011, August 28-31, Washington, DC, USA.
22. 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.
23. Demoly, F., Toussaint, L., Eynard, B., Kiritsis, D., and Gomes, S., 2011. Geometric skeleton computation enabling concurrent product engineering and assembly sequence planning. *Computer-Aided Design*, 43(12), 1654-1673.
24. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P.F., editors., 2003. The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press.
25. SWRL: A Semantic Web Rule Language Combining OWL and RuleML: http://www.w3.org/Submission/SWRL/. April 2011.
26. The Jess rule engine: http://herzberg.ca.sandia.gov/. April 2011.
27. Matsokis, A., and Kiritsis, D., 2010. An ontology-based approach for Product Lifecycle Management. *Computers in Industry*, 61(8), 787-797.
28. Pellet Reasoner: http://clarkparsia.com/pellet/ Last accessed April 2011.
29. Horrocks, I., 2007. Semantic Web: The story so far. ACM International Conference Proceedings vol. 225, pp. 120-125, Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A), Banff, Canada.